

INDICE

Prefazione all'edizione del ventesimo anniversario	xi
Prefazione alla prima edizione	xiii
Nota all'edizione italiana	xv
Capitolo 1 – La pozza di bitume	1
Il prodotto di sistemi di programmazione	1
Le gioie del mestiere	4
I dolori del mestiere	5
Capitolo 2 – Il mitico mese-uomo	7
Ottimismo	8
Il mese-uomo	9
Test di sistemi	12
Stime senza spina dorsale	14
Disastro di pianificazione rigenerativo	14
Capitolo 3 – Il team chirurgico	19
Il problema	19
La proposta di Mills	21
Come funziona	24
Estensione	25
Capitolo 4 – Aristocrazia, democrazia e progettazione di sistemi	27
Integrità concettuale	27
Ottenere l'integrità concettuale	28
Aristocrazia e democrazia	29
Che cosa fa l'implementatore nell'attesa?	32
Capitolo 5 – L'effetto del secondo sistema	35
Disciplina interattiva per l'architetto	35
Autodisciplina: l'effetto del secondo sistema	36

Capitolo 6 – Passare parola	41
Specifiche scritte: il manuale	41
Definizioni formali	42
Incorporamento diretto	45
Riunioni e tribunali	45
Implementazioni multiple	47
Il registro delle telefonate	47
Test del prodotto	48
Capitolo 7 – Perché la Torre di Babele è stata un fallimento?	49
Una verifica manageriale del progetto Babele	49
Comunicazione nel grande progetto di programmazione	50
La guida di lavoro del progetto	51
Che cosa	51
Perché	51
Meccanica	52
Come si farebbe oggi?	53
Organizzazione nel grande progetto di programmazione	54
Capitolo 8 – Prendere le decisioni	59
I dati di Portman	61
I dati di Aron	61
I dati di Harr	62
I dati dell'OS/360	64
I dati di Corbató	64
Capitolo 9 – Dieci chili in un sacco da cinque	65
Lo spazio del programma come costo	65
Controllo delle dimensioni	66
Tecniche di spazio	68
La rappresentazione è l'essenza della programmazione	69
Capitolo 10 – L'ipotesi documentale	71
Documenti per un prodotto hardware	71
Documenti per un dipartimento universitario	73
Documenti per un progetto software	74
Perché avere documenti formali?	74
Capitolo 11 – Pianificate di buttarne via uno	77
Impianti pilota e scala industriale	77
L'unica costante è il cambiamento stesso	78
Pianificate il sistema in vista del cambiamento	79
Pianificate l'organizzazione in vista del cambiamento	79
Due passi avanti e uno indietro	81
Un passo avanti e uno indietro	83
Capitolo 12 – Utensili affilati	85
Macchine target	86
Che tipo di struttura target?	86
Distribuzione del tempo	87

Macchine veicolo e servizi di dati	89
Simulatori	89
Veicoli per compilatori e assembler	89
Librerie e contabilità di programma	90
Strumenti di programmazione	91
Sistema di documentazione	91
Simulatore di prestazioni	91
Linguaggi di alto livello e programmazione interattiva	92
Linguaggio di alto livello	92
Programmazione interattiva	93
Capitolo 13 – L'intero e le parti	95
Progettare per escludere gli errori	95
Definizioni a prova di bug	95
Test delle specifiche	96
Progettazione top-down	96
Programmazione strutturata	97
Debug dei componenti	98
Debug in macchina	98
Dump di memoria	98
Istantanee	99
Debug interattivo	99
Casi di test	100
Debug di sistema	101
Usare componenti sottoposti a debug	101
Costruire molto scaffolding	102
Controllare i cambiamenti	102
Aggiungere un componente alla volta	103
Quantizzare gli aggiornamenti	103
Capitolo 14 – Nascita di una catastrofe	105
Pietre miliari o macine da mulino?	106
“L'altro pezzo è in ritardo, comunque”	107
Sotto il tappeto	108
Attenuare il conflitto di ruolo	108
Spostare il tappeto con la forza	109
Capitolo 15 – L'altra faccia	113
Quale documentazione è necessaria?	114
Per usare un programma	114
Per credere in un programma	115
Per modificare un programma	116
La maledizione del diagramma di flusso	116
Programmi che si autodocumentano	120
Un approccio	121
Qualche tecnica	122
Perché no?	124

Capitolo 16 – Non ci sono pallottole d’argento.	
Sostanza e accidente nell’ingegneria del software	127
Abstract	127
Introduzione	128
Deve per forza essere una cosa difficile? Difficoltà essenziali	129
Complessità	130
Conformità	131
Modificabilità	131
Invisibilità	132
Svolte del passato hanno risolto difficoltà accidentali	133
Linguaggi di alto livello	133
Partizione di tempo	134
Ambienti di programmazione unificati	134
Speranze per l’argento	135
Ada e altri passi avanti nei linguaggi di alto livello	135
Programmazione a oggetti	136
Intelligenza artificiale	137
Sistemi esperti	138
Programmazione “automatica”	140
Programmazione grafica	141
Verifica dei programmi	142
Ambienti e strumenti	142
Stazioni di lavoro	143
Attacchi promettenti all’essenza concettuale	143
Acquistare o costruire	144
Perfezionamento dei requisiti e prototipazione rapida	146
Sviluppo incrementale: far crescere il software, non costruirlo	147
Grandi designer	148
Capitolo 17 – “Non ci sono pallottole d’argento” rivisitato	151
Licantropi e altri terrori legendari	151
Esiste una pallottola d’argento: eccola qui!	152
Ciò che è scritto in modo oscuro verrà frainteso	152
Accidente	153
Una questione di fatto	153
Le difficoltà essenziali quindi sono senza speranza?	154
La complessità è a livelli	154
L’analisi di Harel	156
Pessimismo, ottimismo o realismo	156
Temi “tetri”	157
L’esperienza ideale di Harel	158
Ecco la pallottola d’argento	159
Invisibilità	159
Il punto di Jones: la produttività segue dalla qualità	160
Che cosa è accaduto allora alla produttività?	161
I numeri della produttività	161
Software confezionato: acquistare, non costruire	161
Strumenti avanzati per la mente	162
Programmazione a oggetti: basterà una pallottola d’ottone?	163

Costruire con componenti più grandi	163
Perché la tecnica a oggetti si è sviluppata lentamente?	163
Costi a monte, benefici a valle	164
E il riutilizzo?	165
Come si presenta oggi il riutilizzo a livello aziendale?	166
Imparare grandi vocabolari: un problema prevedibile ma non previsto per il riutilizzo del software	167
Il risultato sulle pallottole: posizione immutata	168
Capitolo 18 – Le affermazioni di questo libro: vere o false?	171
Capitolo 1. La pozza di bitume	171
Capitolo 2. Il mitico mese-uomo	172
Capitolo 3. Il team chirurgico	173
Capitolo 4. Aristocrazia, democrazia e progettazione di sistemi	174
Capitolo 5. L'effetto del secondo sistema	175
Capitolo 6. Passare parola	175
Capitolo 7. Perché la Torre di Babele è stata un fallimento?	176
Comunicazione	176
Guida di lavoro del progetto	177
Organizzazione	178
Capitolo 8. Prendere le decisioni	178
Capitolo 9. Dieci chili in un sacco da cinque	179
Capitolo 10. L'ipotesi documentale	181
Capitolo 11. Pianificate di buttarne via uno	182
Pianificate l'organizzazione in vista del cambiamento	183
Due passi avanti e uno indietro: la manutenzione del programma	184
Un passo avanti e uno indietro: l'entropia del sistema aumenta nel corso della sua vita	185
Capitolo 12. Utensili affiliati	185
Linguaggi di alto livello	186
Programmazione interattiva	187
Capitolo 13. L'intero e le parti	187
Capitolo 14. La nascita di una catastrofe	188
Capitolo 15. L'altra faccia	190
Epilogo originale	192
Capitolo 19 – Il libro vent'anni dopo	193
Perché un'edizione per il ventesimo anniversario?	193
L'argomentazione centrale: l'integrità concettuale e l'architetto	194
Integrità concettuale	194
L'architetto	195
Separazione dell'architettura da implementazione e realizzazione	196
Ricorsione degli architetti	196
Oggi ne sono più convinto che mai	196
L'effetto del secondo sistema: featurite e frequenze ipotetiche	197
Progettare per grandi gruppi di utenti	197
Featurite	197

Definire l'insieme degli utenti	198
Frequenze	198
E che cosa dire dell'“effetto del secondo sistema”?	199
Il trionfo dell'interfaccia WIMP	199
Integrità concettuale attraverso una metafora	200
Emissione di comandi e il problema dei due cursori	201
Una soluzione brillante	201
Potere per gli utenti o facilità d'uso	202
Transizione incrementale: da neofita a utente avanzato	202
Il successo dell'incorporamento diretto come strumento per far rispettare l'architettura	203
Il destino di WIMP: l'obsolescenza	203
Non costruitene uno da buttare: il modello a cascata è sbagliato!	204
Deve esserci movimento a monte	205
Un modello a costruzione incrementale è meglio: il perfezionamento progressivo	206
Costruire un sistema scheletro dall'inizio alla fine	206
Le famiglie di Parnas	207
L'approccio “costruzione ogni notte” di Microsoft	208
Costruzione incrementale e prototipazione rapida	209
Parnas aveva ragione e io torto a proposito dell'incapsulamento	210
Quanto è mitico il mese-uomo? Il modello e i dati di Boehm	212
Quanto è vera la Legge di Brooks?	213
Le persone sono tutto (beh, quasi tutto)	214
Peopleware	214
Progetti in movimento	215
Il potere di cedere potere	215
Qual è la sorpresa più grande? Milioni di computer	217
La rivoluzione dei microcomputer ha cambiato come vengono usati i computer	218
La rivoluzione del microcomputer ha cambiato il modo in cui si costruisce il software	219
Una nuova industria del software: il software preconfezionato	220
L'industria classica del software	221
Il mondo dei sistemi operativi si è compattato	221
L'industria del software preconfezionato	222
Acquistare e costruire: i pacchetti preconfezionati come componenti	223
Metaprogrammazione	223
Questo attacca realmente l'essenza	223
Che cosa serve?	224
Lo stato e il futuro dell'ingegneria del software	225
Epilogo – Cinquant'anni di meraviglia, entusiasmo e gioia	227
Note e riferimenti bibliografici	229