

# Indice generale

<b>Prefazione .....</b>	<b>xiii</b>
-------------------------	-------------

<b>Introduzione .....</b>	<b>xv</b>
---------------------------	-----------

Come è organizzato il libro.....	xvi
Che cosa c'è in un nome?.....	xvi
Codice sorgente e altre risorse.....	xvii
Feedback .....	xvii
Ringraziamenti per la seconda edizione .....	xvii

<b>Dall'Introduzione alla prima edizione .....</b>	<b>xix</b>
--	------------

Per chi è questo libro?.....	xx
Che cosa fa di un programmatore un "pragmatic programmer"? .....	xx
Pragmatici singoli, grandi squadre.....	xxi
È un processo continuo.....	xxii

<b>Capitolo 1 Una filosofia pragmatica .....</b>	<b>1</b>
--	----------

È la vostra vita .....	2
Il gatto mi ha mangiato il codice sorgente .....	3
La fiducia del team .....	3
Prendersi la responsabilità .....	4
Entropia del software .....	5
Primo, non nuocere .....	6
Zuppa di pietre e rane bollite .....	7
Dalla parte degli abitanti del villaggio.....	8
Software abbastanza buono.....	9
Coinvolgete gli utenti nel compromesso .....	9
Sapere quando fermarsi.....	10
Il portafoglio delle conoscenze .....	11
Il vostro portafoglio di conoscenze.....	11
Costruire il proprio portafoglio.....	12

Obiettivi.....	12
Occasioni di apprendimento .....	13
Pensiero critico.....	14
Comunicare!.....	15
Dovete conoscere il vostro pubblico.....	16
Dovete aver chiaro che cosa volete dire .....	16
Scegliete il momento giusto.....	17
Scegliete uno stile.....	17
Dategli un bell'aspetto .....	17
Coinvolgete il vostro pubblico .....	18
Imparate ad ascoltare.....	18
Rispondete alle persone.....	18
Documentazione .....	18
Riepilogo .....	19

## Capitolo 2 Un approccio pragmatico.....21

L'essenza del buon design.....	22
ETC è un valore, non una regola .....	22
DRY: i mali della duplicazione .....	24
DRY riguarda molto più che il codice .....	25
Duplicazione nel codice.....	25
Duplicazione nella documentazione.....	27
Duplicazione di rappresentazione.....	29
Ortogonalità.....	31
Che cos'è l'ortogonalità?.....	31
Benefici dell'ortogonalità.....	32
Progetto .....	33
Toolkit e librerie.....	35
Scrittura del codice.....	35
Test .....	36
Documentazione .....	36
Vivere con l'ortogonalità.....	37
Reversibilità.....	38
Reversibilità .....	39
Architettura flessibile .....	39
Proiettili traccianti .....	41
Codice che brilla nel buio .....	41
I proiettili traccianti non sempre colpiscono il bersaglio .....	44
Codice tracciante contro prototipazione.....	44
Prototipi e Post-it .....	45
Cose da prototipare .....	46
Come usare i prototipi.....	46
Prototipi di architettura.....	47
Come non usare i prototipi.....	47
Linguaggi di dominio .....	48
Qualche linguaggio di dominio del mondo reale.....	48

Caratteristiche dei linguaggi di dominio .....	50
Compromessi fra linguaggi interni ed esterni .....	51
Un linguaggio di dominio interno a basso costo .....	51
Stime .....	53
Quanto accurato è abbastanza accurato? .....	53
Da dove arrivano le stime? .....	54
Capire la domanda .....	54
Costruire un modello del sistema .....	54
Suddividere il modello in componenti .....	55
Dare a ciascun parametro un valore .....	55
Calcolare le risposte .....	55
Tener traccia della propria abilità nelle stime .....	56
Stimare i tempi di un progetto .....	56
Che cosa dire quando viene chiesta una stima .....	57
<b>Capitolo 3   Gli strumenti di base.....</b>	<b>59</b>
Il potere del puro testo .....	60
Che cos'è il puro testo? .....	61
Il minimo comun denominatore .....	63
Giochi di shell .....	63
Una shell tutta per sé .....	64
Power editing .....	66
Che cosa significa "competente"? .....	66
Il cammino verso la competenza .....	67
Controllo delle versioni .....	68
Tutto parte dal sorgente .....	69
Diramazioni .....	70
Controllo delle versioni come nodo centrale del progetto .....	71
Debug .....	72
Psicologia del debug .....	73
Una mentalità da debug .....	73
Da dove iniziare .....	73
Strategie di debug .....	74
Il programmatore in una terra straniera .....	75
La sezione binaria .....	76
Lista di controllo per il debug .....	79
Manipolazione del testo .....	80
Diari tecnici .....	82
<b>Capitolo 4   Paranoia pragmatica .....</b>	<b>83</b>
Progettazione per contratto .....	84
DBC .....	84
Invarianti di classe e linguaggi funzionali .....	87
DBC e crash precoci .....	88
Contratti dinamici e agenti .....	90
I programmi defunti non mentono .....	91

Catch and release è una tecnica per la pesca .....	91
Crash, non spazzatura.....	92
Programmazione assertiva .....	92
Asserzioni ed effetti collaterali.....	93
Lasciate attive le asserzioni .....	94
Come equilibrare le risorse .....	95
Annidare le allocazioni.....	98
Oggetti ed eccezioni.....	98
Equilibri ed eccezioni.....	98
Quando non si possono equilibrare le risorse .....	100
Controllare l'equilibrio .....	100
Non correte più dei vostri fari .....	101
Cigni neri.....	102

## Capitolo 5 **Piegarsi o spezzarsi .....**105

Disaccoppiamento .....	106
Disastri ferroviari .....	107
I mali della globalizzazione .....	110
L'ereditarietà aggiunge accoppiamento.....	111
Ancora una volta: tutto riguarda il cambiamento .....	111
Destreggiarsi nel mondo reale .....	112
Eventi.....	112
Macchine a stati finiti .....	113
Il pattern Osservatore .....	116
Publish/Subscribe (pubblicazione/abbonamento).....	117
Programmazione reattiva, flussi ed eventi.....	118
Gli eventi sono onnipresenti .....	120
Programmazione di trasformazioni.....	121
Trovare le trasformazioni.....	123
E la gestione degli errori? .....	127
Le trasformazioni trasformano la programmazione .....	130
Tassa sull'eredità.....	131
Un po' di informazioni di fondo.....	131
Problemi nell'uso dell'ereditarietà per condividere codice .....	132
Le alternative sono meglio .....	134
Raramente l'ereditarietà è la risposta.....	137
Configurazione.....	138
Configurazione statica .....	138
Configurazione come servizio .....	139
Non scrivete codice dodo.....	139

## Capitolo 6 **Concorrenza .....**141

Tutto è concorrente.....	141
Spezzare l'accoppiamento temporale .....	142
Alla ricerca della concorrenza .....	143
Opportunità di concorrenza .....	145

Occasioni di parallelismo .....	145
Identificare le opportunità è la parte facile .....	146
Uno stato condiviso è uno stato non corretto .....	146
Aggiornamenti non atomici .....	147
Transazioni con molte risorse .....	150
Aggiornamenti non transazionali .....	151
Altri tipi di accesso esclusivo .....	151
Dottore, fa male... ..	151
Attori e processi .....	152
Gli attori possono essere solo concorrenti .....	152
Un attore semplice .....	153
Nessuna concorrenza esplicita .....	156
Erlang imposta la scena .....	156
Lavagne .....	157
Una lavagna in azione .....	158
I sistemi di messaggistica possono essere come lavagne .....	159
Ma non è tanto semplice .....	159

## **Capitolo 7   Mentre scrivete il codice.....161**

Ascoltare il cervello rettiliano .....	162
Paura della pagina bianca .....	163
Combattere se stessi .....	163
Come parlare lucertolese .....	164
È ora di giocare! .....	164
Non solo il vostro codice .....	165
Non solo codice .....	165
Programmazione per coincidenza .....	166
Come programmare per coincidenza .....	166
Come programmare consapevolmente .....	169
Velocità degli algoritmi .....	171
Che cosa si intende per stima degli algoritmi? .....	171
Stima a buon senso .....	173
La velocità degli algoritmi in pratica .....	173
Refactoring .....	175
Refactoring: quando? .....	176
Refactoring: come si fa? .....	178
Test per scrivere il codice .....	179
Pensare ai test .....	179
Scrittura del codice guidata dai test .....	180
TDD: dovete sapere dove state andando .....	182
Ritorno al codice .....	183
Unit test .....	183
Test rispetto al contratto .....	184
Test ad hoc .....	185
Costruire una finestra di test .....	185
Una cultura del testing .....	185

Test basati su proprietà .....	187
Contratti, invarianti e proprietà .....	187
Generazione di dati di test .....	188
Trovare cattivi presupposti.....	189
I test basati su proprietà spesso vi sorprendono .....	192
I test basati su proprietà aiutano anche nella progettazione.....	192
Sicuri là fuori.....	193
L'altro 90 per cento.....	193
Principi base della sicurezza .....	194
Buon senso e crittografia .....	198
Dare un nome alle cose.....	199
Onore alla cultura.....	201
Coerenza.....	202
Cambiare nome è ancora più difficile.....	202

## **Capitolo 8 Prima del progetto.....205**

La fossa dei requisiti .....	206
Il mito dei requisiti .....	206
Programmazione come terapia .....	206
I requisiti sono un processo .....	208
Mettersi nei panni del cliente.....	208
Requisiti contro norme aziendali.....	209
Requisiti contro realtà .....	209
Documentazione dei requisiti .....	210
Solo una cosina ancora.....	211
Mantenere un glossario.....	211
Risolvere rompicapo impossibili.....	213
Gradi di libertà .....	213
Cambiate strada! .....	214
La fortuna aiuta la mente preparata .....	215
Lavorare insieme .....	215
Programmazione in coppia .....	216
Programmazione in gruppo .....	217
Che cosa devo fare? .....	217
L'essenza dell'agilità.....	218
Non può esserci mai un processo agile .....	219
E allora che cosa facciamo? .....	219

## **Capitolo 9 Progetti pragmatici .....221**

Team pragmatici .....	222
Niente finestre rotte.....	222
Rane bollite .....	223
Pianificate il vostro portafoglio di conoscenze .....	223
Comunicare la presenza del team .....	224
Non ripetetevi.....	224
Proiettili traccianti per il team.....	225

---

Automazione.....	226
Sapere quando smettere di aggiungere vernice .....	226
Le noci di cocco non bastano.....	227
Il contesto è importante.....	227
La taglia unica non è perfetta per nessuno .....	228
L'obiettivo reale .....	228
Starter kit pragmatico.....	230
Guidare con il controllo delle versioni .....	230
Test senza pietà e continui.....	231
Stringere la rete .....	234
Deliziate i vostri utenti.....	235
Orgoglio e pregiudizio.....	237
<b>Postfazione .....</b>	<b>239</b>
La bussola morale.....	240
Immaginate il futuro che volete .....	240
<b>Bibliografia .....</b>	<b>241</b>
<b>Risposte agli esercizi .....</b>	<b>243</b>
<b>Indice analitico.....</b>	<b>259</b>