

Indice generale

Prefazione	x1
Introduzione	xiii
Sul termine “craftsmanship”	xiii
Sull’unica vera via	xiii
Introduzione al libro	xiv
Per voi stessi	xiv
Per la società	xiv
La struttura di questo libro	xv
Una nota per i manager	xvi
Video di supporto	xvi
Ringraziamenti	xvi
L’autore	xvii
Capitolo 1	Craftsmanship: essere artigiani..... 1
Parte I	Le discipline.....7
XP: extreme programming	9
Il cerchio della vita	9
Test-Driven Development	10
Refactoring	10
Simple design	11
Programmazione collaborativa	11
Test di accettazione	12
Capitolo 2	Test-Driven Development 13
Panoramica	14
Software	15
Le tre leggi del TDD	16

La quarta legge	22
Le basi	24
Esempi semplici.....	24
Uno stack.....	24
Fattori primi.....	36
Il gioco del bowling.....	42
Conclusioni	55

Capitolo 3 TDD avanzato57

Sort 1	58
Sort 2	61
Rimanere bloccati.....	67
Arrange, Act, Assert	72
Arriva il BDD	73
Macchine a stati finiti.....	73
Ancora sul BDD.....	75
I test double.....	75
I dummy	77
Gli stub	80
Le spy.....	82
I mock	84
I fake	86
Il principio di indeterminazione del TDD.....	88
Londra contro Chicago.....	96
Il problema della certezza.....	97
Londra.....	97
Chicago.....	98
Sintesi.....	99
Architettura	99
Conclusioni	100

Capitolo 4 Progettazione dei test101

Test dei database	102
Test delle GUI	103
Input della GUI.....	105
Pattern di test	105
Test-Specific Subclass.....	106
Self-Shunt	107
Humble Object	108
Progettazione dei test.....	110
Il problema dei test fragili	110
La corrispondenza uno-a-uno.....	110
Spezzare ogni corrispondenza uno-a-uno.....	112
Il noleggio video	113
Specificità contro generalità	127

Presupposto della priorità delle trasformazioni.....	128
{} → Nil.....	129
Nil → Costante.....	129
Incondizionato → Selezione.....	131
Valore → Lista.....	131
Istruzione → Ricorsione.....	131
Selezione → Iterazione.....	132
Valore → Valore mutato.....	132
Esempio – Fibonacci.....	133
Il presupposto della priorità delle trasformazioni.....	135
Conclusioni.....	136
Capitolo 5 Refactoring.....	137
Che cos'è il refactoring?.....	138
Il toolkit di base.....	139
Rinominare.....	139
Estrarre i metodi.....	140
Estrarre le variabili.....	141
Estrarre i campi.....	142
Come un cubo di Rubik.....	151
Le discipline.....	152
I test.....	152
Test rapidi.....	152
Romperle corrispondenze uno-a-uno profonde.....	152
Refactoring continuo.....	153
Refactoring spietato.....	153
Continuate a superare i test!.....	153
Lasciatevi una via di fuga.....	154
Conclusioni.....	154
Capitolo 6 Simple design.....	155
YAGNI.....	158
Copertura dei test.....	159
Copertura.....	159
Un obiettivo asintotico.....	160
Design?.....	161
Ma c'è di più.....	161
Massima espressività.....	162
L'astrazione sottostante.....	163
I test: l'altra metà del problema.....	164
Riduzione al minimo delle duplicazioni.....	164
Duplicazioni accidentali.....	165
Riduzione delle dimensioni.....	165
Simple design.....	166

Capitolo 7	Programmazione collaborativa	167
Capitolo 8	Test di accettazione	171
	La disciplina	173
	La continuous build	174
Parte II	Gli standard	175
	Il vostro nuovo CTO	176
Capitolo 9	Produttività	177
	Non rilasceremo mai M***A	178
	Adattabilità non costosa	179
	Saremo sempre pronti	180
	Produttività stabile	181
Capitolo 10	Qualità	183
	Miglioramento continuo	184
	Competenza senza paura	184
	Qualità estrema	185
	Non scaricheremo le nostre responsabilità sul QA	186
	La malattia del QA	186
	Il QA non troverà nulla	187
	Automazione dei test	187
	Test automatizzati e interfacce utente	188
	Test dell'interfaccia utente	189
Capitolo 11	Coraggio	191
	Coprirsi a vicenda	192
	Stime oneste	192
	Saper dire di no	194
	Apprendimento aggressivo continuo	194
	Mentoring	195
Parte III	L'etica	197
	Il primo programmatore	198
	Settantacinque anni	199
	Da nerd a salvatori	201
	Da esempi a cattivi	203
	Governiamo il mondo	203
	Catastrofi	204
	Il giuramento	205

Capitolo 12 Danni.....207

Primo, non nuocere	208
Nessun danno alla società.....	208
Nessun danno funzionale.....	209
Nessun danno alla struttura del codice	211
Soft	212
I test.....	213
Il mio miglior lavoro.....	213
Farlo bene	214
Che cos'è una buona struttura?	215
La matrice di Eisenhower	216
I programmatori sono stakeholder.....	218
Il vostro meglio	218
Prova ripetibile	220
Dijkstra	220
Dimostrare la correttezza	220
La programmazione strutturata.....	222
Scomposizione funzionale.....	224
Test-Driven Development.....	224

Capitolo 13 Integrità.....227

Piccoli cicli.....	228
La storia del controllo del codice sorgente.....	228
Git	232
Cicli brevi	233
Integrazione continua	233
Branching e toggling	234
Deployment continuo.....	235
Build continue.....	236
Miglioramento incessante.....	236
Copertura dei test.....	237
Test delle mutazioni.....	237
Stabilità semantica.....	238
Pulizia	238
Creazioni.....	239
Mantenere un'elevata produttività	239
Viscosità	239
Gestione delle distrazioni	241
Gestione del tempo	243

Capitolo 14 Lavoro di squadra245

Lavorare in un team.....	246
Ufficio aperto/virtuale.....	246
Produrre stime oneste ed eque	247
Bugie	247

Onestà, accuratezza, precisione	248
Aneddoto 1: i vettori	249
Aneddoto 2: pCCU	250
La lezione	251
Accuratezza	251
Precisione	252
Aggregazione.....	253
Onestà.....	254
Rispetto	255
Mai smettere di imparare	255

Indice analitico	257
-------------------------------	------------