

SPIS TREŚCI

	Przedmowa	15
	Wstęp	19
	Podziękowania	25
	O autorze	27
Rozdział 1	Rzemiosło	29
CZĘŚĆ I	Procedury	39
	Programowanie ekstremalne	41
	Krag rozwoju	42
	Programowanie sterowane testami	43
	Refaktoryzacja	44
	Prostota projektu	45
	Programowanie zespołowe	46
	Testy akceptacyjne	47

Rozdział 2	Programowanie sterowane testami	49
	Ogólny zarys	50
	Oprogramowanie	53
	Trzy prawa TDD	54
	Czwarte prawo	65
	Podstawy	67
	Proste przykłady	67
	Stos	68
	Czynniki pierwsze	83
	Gra w kręgle	91
	Zakończenie	109
Rozdział 3	Zaawansowane techniki TDD	111
	Sortowanie — podejście 1.	112
	Sortowanie — podejście 2.	116
	Utknięcie	124
	Przygotuj, działaj, sprawdź	131
	Wprowadzenie do BDD	132
	Automaty skończone	133
	Znowu o BDD	135
	Dublery testowe	136
	Atrapa	139
	Zaślepka	142
	Szpieg	145
	Imitacja	147
	Podróbka	150
	Zasada niepewności metodyki TDD	153
	Londyn kontra Chicago	165
	Problem pewności	166

Londyn	167
Chicago	168
Synteza	169
Architektura	170
Zakończenie	172
Rozdział 4 Projektowanie testów	173
Testowanie baz danych	174
Testowanie interfejsów G U I	176
Dane wprowadzane z interfejsu G U I	179
Wzorce testowe	180
Podklasa specyficzna dla testów	181
Samopodstawienie	182
Skromny obiekt	183
Projektowanie testów	187
Problem kruchych testów	187
Zgodność jeden do jednego	188
Zrywanie zgodności	189
Wypożyczalnia filmów	191
Szczegółowość kontra ogólność	208
Domniemane pierwszeństwo przekształceń (Transformation Priority Premise)	209
{} •» Nil	212
Nil -> stała	212
Stała -> zmienna	212
Bezwarunkowość -> wybór	213
Wartość -> lista	214
Instrukcja -> rekurencja	214
Wybór -> iteracja	215
Wartość -> zmieniona wartość	215

SPIS TREŚCI

	Przykład: Fibonacci	216
	Domniemane pierwszeństwo przekształceń	220
	Zakończenie	221
Rozdział 5	Refaktoryzacja	223
	Czym jest refaktoryzacja?	225
	Podstawowy zestaw narzędziowy	226
	Zmiana nazwy	226
	Wyodrębnianie metody	227
	Wyodrębnianie zmiennej	229
	Wyodrębnianie pola	230
	Kostka Rubika	242
	Procedury	243
	Testy	243
	Szybkie testy	243
	Zerwij z głęboką zgodnością jeden do jednego	244
	Stale refaktoryzuj	244
	Refaktoryzuj bezwzględnie	244
	Niech wyniki testów będą stale pozytywne!	245
	Pozostaw sobie wyjście	246
	Zakończenie	246
Rozdział 6	Prostota projektu	247
	YAGNI	251
	Kod pokryty testami	253
	Pokrycie	254
	Cel asymptotyczny	256
	Projekt?	256
	To jednak nie wszystko	257

	Zwiększenie wyrazistości	257
	Bazowa abstrakcja	259
	Testy: druga część problemu	260
	Ograniczenie duplikacji	261
	Przypadkowa duplikacja	262
	Zmniejszanie	263
	Prosta konstrukcja	264
Rozdział 7	Programowanie zespołowe	265
Rozdział 8	Testy akceptacyjne	271
	Procedura	274
	Ciągła budowa	275
CZĘŚĆ II	Standardy	277
	Twój nowy dyrektor techniczny	278
Rozdział 9	Produktywność	279
	Nie będziemy nigdy wciskać badziewia	280
	Możliwość niedrogiej adaptacji	282
	Będziemy zawsze gotowi	284
	Stabilna wydajność	285
Rozdział 10	Jakość	287
	Ciągłe ulepszanie	288
	Odważna fachowość	289
	Wyjątkowa jakość	290
	Nie będziemy zrzucali pracy na dział zapewniania jakości	291
	Przypadłość działu zapewniania jakości	292

SPIS TREŚCI

	Dział zapewniania jakości niczego nie znajdzie	292
	Automatyzacja testów	293
	Testowanie automatyczne a interfejsy użytkownika	294
	Testowanie interfejsu użytkownika	296
Rozdział 11	Odwaga	297
	Zastępujemy się nawzajem	298
	Rzetelne oszacowania	300
	Musisz mówić „NIE”	301
	Ciągłe aktywne uczenie się	303
	Mentorowanie	304
CZĘŚĆ III	Etyka	307
	Pierwszy programista	308
	Siedemdziesiąt pięć lat	309
	Ofermy i wyzwoliciele	314
	Wzory osobowe i czarne charaktery	317
	Rządzimy światem	318
	Katastrofy	319
	Przysięga	321
Rozdział 12	Szkody	323
	Po pierwsze, nie szkodzić	324
	Nie szkodzić społeczeństwu	325
	Uszczerbek w funkcjonowaniu	327
	Nieszkodzenie strukturze	330
	Elastyczność	331
	Testy	333

Najlepsza praca	335
Jak zrobić to dobrze	336
Czym jest dobra struktura?	337
Macierz Eisenhowera	339
Programiści są interesariuszami	341
Dokładanie wszelkich starań	343
Powtarzalny dowód	345
Dijkstra	345
Udawnianie poprawności	346
Programowanie strukturalne	349
Dekompozycja funkcyjna	351
Programowanie sterowane testami	353
 Rozdział 13 Integralność	 357
Krótkie cykle	358
Historia kontroli kodu źródłowego	358
Git	364
Krótkie cykle	365
Ciągła integracja	366
Gałęzie kontra przełączniki	367
Ciągłe wdrażanie	370
Ciągła budowa	371
Bezwzględne ulepszanie	372
Pokrycie testami	373
Testowanie mutacyjne	374
Stabilność semantyczna	375
Oczyszczanie	375
Wytwory	376

SPIS TREŚCI

Utrzymywanie wysokiej wydajności	376
Lepkość	377
Radzenie sobie z rozproszeniami	380
Zarządzanie czasem	383
Rozdział 14 Praca zespołowa	387
Praca w zespole	388
Otwarte/wirtualne biuro	388
Rzetelne i uczciwe oszacowania	390
Kłamstwa	391
Uczciwość, dokładność, precyzja	392
Historia nr 1: wektory	394
Historia nr 2: pCCU	396
Nauczka	398
Dokładność	398
Precyzja	400
Łączenie	402
Uczciwość	403
Szacunek	406
Nigdy nie przestawaj się uczyć	407