



Test-Driven Development: Podstawy

Rob Myers

APN Promise
Warszawa 2026

Spis treści

Przedmowaxiii
Wstępxix
Podziękowaniaxxviii
O autorzexxx

Część I: Podstawowe techniki

Rozdział 1: Myślenie na sposób TDD	3
Siatka zabezpieczająca	4
Jednostki zachowania	4
Granice zachowania	6
Smak TDD	9
Przyszłość TDD	14
TDD oraz tworzenie oprogramowania oparte na zachowaniach	15
TDD i programowanie funkcyjne	16
Marzenia o TDD, komputerach kwantowych i braku implementacji ...	17
Rzeczywistość TDD, sztucznej inteligencji i „kodowania na wibracjach”	18
Podsumowanie	20
Rozdział 2: Podstawowe ruchy	21
Schemat blokowy TDD	21
Myślenie w kategoriach testów	23
Trzy podstawowe techniki	26
Zlecenia, zadania i lista rzeczy do zrobienia	26
Skromna lista rzeczy do zrobienia	28
Zaczynamy	28
Przewodnik po ćwiczeniach	31
Krok 1: Napisz test nowego zachowania	32

Krok 2: Spraw, aby wyraźnie się nie powiódł.	34
Krok 3: Niech to przejdzie	35
Krok 4: Refaktoryzacja	37
Powtarzanie cyklu	37
Kolejne ruchy	40
Podsumowanie	45
Rozdział 3: Wykorzystanie istniejących zachowań	47
Kolejne kroki.	47
Grupowanie testów według zachowania.	48
Udawanie	49
Triangulacja	49
Oczywista implementacja.	50
Refaktoryzacja: od oczywistego do lepszego	50
Ulepszanie testów.	53
Wezwij „stażystę z jetlagiem”	53
Refaktoryzacja testów	55
Usuwanie wady bezobjawowej	57
Eliminowanie skutków ubocznych metodą triangulacji.	60
Obraz jest wart tysiąca słów	62
Łączenie zachowań.	66
Optymalizacja z właściwych powodów	70
Podsumowanie	72
Najważniejsze wnioski	72
Co warto wypróbować dalej.	73
Rozdział 4: Zachowania związane z wyjątkami	75
Zgłaszanie/wyrzucanie	75
Test, który kończy się powodzeniem, gdy kod „zawodzi”	77
Wyjątki zewnętrzne	80
Podsumowanie	80
Rozdział 5: Utrzymanie praktyki opartej na testach.	81
Cechy dobrego testu jednostkowego	82
Szybkość	82
Przejrzysty.	83
Zwięzłość/koncentracja.	84
Niezależny.	85

Powtarzalny	85
Od czego zacząć	85
Karty CRC	86
Co testować	91
Testuj zachowania, a nie implementację	91
Testuj zachowania, a nie struktury danych	92
Testuj granice zachowań, a nie dane	95
Testuj tylko to, co jest znane	96
ZOMBIES	97
Gdy kod jest trudny do testowania	100
Testowanie zapachów i refaktoryzacje	101
Zapach: nieprzydatne nazwy testów	103
Zapach: nieprzydatne nazwy instancji	105
Zapach: kopiuj/wklej/modyfikuj	106
Zapach: zdublowany warunek	106
Zapach: nadmiar w warunku	109
Zapach: odrzucony warunek	111
Zapach: wymuszona wartość zwracana	112
Zapach: sprzężone asercje	116
Zapach: rozbudowana narracja	119
Zapach: niepowiązane asercje	121
Zapach: niezawodna asercja	122
Zapach: zbędna asercja	122
Zapach: powtarzające się asercje	123
Zapach: rozgałęziony kod	125
Zapach: obliczenia	126
Zapach: niestandardowe nadrzędne klasy testowe	127
Zapach: testy kompleksowe	128
Zapach: czyszczenie	130
Zapach: częste zmiany w warunku	130
Zapach: wyłącznie do celów testowych	133
Zapach: sporadyczne awarie	135
Zapach: rozproszone zmiany	136
Zapach: wiele dublerów testowych	141
Zapach: odkryta implementacja	141
Inne typowe wyzwania związane z testowaniem	142
Jak przetestować metodę prywatną	142

Jak przetestować klasę abstrakcyjną	142
Jak przetestować metodę, która tworzy własne zależności.	143
Jak przetestować lub zrefaktoryzować istniejący, nieprzetestowany kod	144
Podsumowanie	144

Część II: Praktyki pomocnicze

Rozdział 6: Dublery testowe.	147
Problem z zależnościami.	148
Taksonomia dublerów testowych.	149
Podróbki	150
Stuby	150
Imitacje	153
Szpieg.	155
Obiekty nullable	159
Dodatkowe zalecenia.	161
Preferowanie wstrzykiwania zależności	161
Opakowywanie zależności zewnętrznych	163
Wykorzystywanie prostych obiektów jako ich własnych dublerów testowych	166
Podsumowanie: Stosuj ostrożnie	168
Pamiętaj o wdrażaniu z rzeczywistymi zależnościami	168
Uzupełnianie o wąskie testy integracyjne.	168
Oszczędnie korzystaj ze środowisk imitujących	169
Rozdział 7: Testowanie starego kodu.	171
Testowanie charakteryzujące	173
Trzy pytania	174
Zaczynamy pisanie pierwszego testu	176
Dziel i rządź	177
Przyrostowe tworzenie testów	181
Inne kwestie do rozważenia	185
Testowanie wszystkich ścieżek i wyników	185
Maksymalizacja pokrywanego zakresu przy minimalnym wysiłku . . .	185
Testy charakteryzujące także trzeba refaktoryzować	186
Pamiętajmy, że nie jest to TDD	186
Gdy znajdziemy błąd.	187

Wprowadzenie proxy wirtualizacyjnego	188
Krok 1: tworzymy proxy wirtualizujące	190
Krok 2: zastępujemy oryginał przez proxy	191
Używanie w teście dublera testowego dla proxy	192
Podsumowanie: kompletny zestaw narzędzi	194

Część III: Zwrot z inwestycji

Rozdział 8: Czarne łabędzie	197
Dylemat zwinności	198
Praca bez siatki zabezpieczającej	198
Trzy poziomy wartości	200
Poziom 1: radykalna redukcja liczby błędów i poprawek	200
Poziom 2: krótszy czas wprowadzenia funkcji na rynek	201
Poziom 3: implementacja nieoczekiwanych pomysłów	201
Historie użytkowników z czarnymi łabędziami	202
Duża zmiana architektury	203
OTIS2	206
Proces sekurytyzacji pożyczek	209
Ujarmianie czarnych łabędzi	211
Podsumowanie	212
Dodatek: Ćwiczenia	215
Narzędzie do sprawdzania siły hasła	215
Narzędzie do sprawdzania siły hasła: część 1	215
Narzędzie do sprawdzania siły hasła: część 2	217
Salvo	218
Indeks	221