Spis treści

Przedmowa 9

O autorach 11

O korektorze merytorycznym 12

Wprowadzenie 13

Rozdział 1. Dlaczego TDD jest ważne? 17

Najpierw trochę o nas 18
Historia Johna 18
Historia Claytona 18
Czym jest TDD? 19
Podejście do TDD 19
Podejście alternatywne 20
Proces 20
Po co zawracać sobie tym głowę? 21
Argumenty przeciwko TDD 21
Testowanie wymaga czasu 21
Testowanie jest kosztowne 22
Testowanie jest trudne 22
Nie wiemy jak 22
Argumenty za TDD 23
Mniejsza pracochłonność testowania manualnego 23
Mniej błędów 23
Pewien poziom poprawności 23
Brak strachu przed refaktoryzacją 24
Lepsza architektura 24
Szybsza praca 24
Różne rodzaje testów 25
Testy jednostkowe 25
Testy akceptacyjne 25

Testy integracyjne 25
Testy typu end-to-end 26
Liczba testów poszczególnych rodzajów 26
Części testu jednostkowego 26
Aranżacja 26
Akcja 26
Asercja 27
Wymagania 27
Dlaczego wymagania są ważne? 27
Historie użytkownika 27
Gherkin 29
Nasze pierwsze testy w C# 31
Rozwijanie aplikacji z testami 33
Nasze pierwsze testy w JavaScripcie 34
Dlaczego to ma znaczenie? 37
Podsumowanie 37

Rozdział 2. Przygotowanie środowiska testowego w .NET 39

Instalacja SDK .NET Core 39
Przygotowanie VS Code 40
Tworzenie projektu w VS Code 44
Przygotowanie Visual Studio Community 45
Pobieranie Visual Studio Community 46
Instalacja Visual Studio Community 46
Przesiadka na xUnit 46
Programistyczne kata 47
Stworzenie projektu 47
Czym jest Speaker Meet? 50
Projekt Web API 51
Podsumowanie 55

Rozdział 3. Przygotowanie środowiska testowego w JavaScripcie 57

Node.js 57
Czym jest Node? 58
Po co nam Node? 58
Instalacja Node 58
NPM 61
Szybkie wprowadzenie do IDE dedykowanych dla JavaScriptu 62
Visual Studio Code 63
WebStorm 64
Create React App 65
Czym jest Create React App? 66
Instalacja modułu globalnego 66
Tworzenie aplikacji za pomocą Reacta 66
Mocha i Chai 67
Szybkie kata sprawdzające środowisko 72
Wymagania 72
Wykonanie 72
Rozpoczęcie kata 73
Podsumowanie 76

Rozdział 4. Co należy wiedzieć przed rozpoczęciem pracy? 77

Nietestowalny kod 78
Wstrzykiwanie zależności 78
Wyodrębnianie oprogramowania zewnętrznego 79
Sobowtóry testowe 79
Frameworki imitujące 80
Zasady SOLID 80
Powitanie zależne od czasu 83
Kruche testy 84
Rodzaje sobowtórów testowych 86
Przykład wielopoziomowy 93
Podsumowanie 100

Rozdział 5. Tabula rasa — podejście do aplikacji na sposób TDD 101

Gdzie zacząć? 101
Golenie jaka 102
Duży projekt od razu 103
Czysta kartka 103
Po jednym kawałku 103
Minimalny wykonalny produkt 104
Inny sposób myślenia 104
Nie będziesz tego potrzebować 104
Małe testy 105
Adwokat diabła 106
Najpierw testy ścieżek negatywnych 109
Kiedy testowanie jest bolesne 113
Symulacja 113
Najpierw asercja 114
Bądź zorganizowany 114
Rozbicie aplikacji Speaker Meet 114
Prelegenci 114
Społeczności 115
Konferencje 115
Wymagania techniczne 115
Podsumowanie 115

Rozdział 6. Podejście do problemu 117

Zdefiniowanie problemu 117
Przetrawienie problemu 118
Epiki, funkcje i historie — ojej! 118
Problem Speaker Meet 120
Architektura heksagonalna wielowarstwowa 126
Architektura heksagonalna 127
Podstawowe, ale wydajne podziały wielowarstwowe 127
Kierunek testowania 130
Od tyłu do przodu 130
Od przodu do tyłu 137
Od wewnątrz na zewnątrz 144
Podsumowanie 148

Rozdział 7. Sterowanie testami aplikacji C# 149

Przegląd wymagań 149
Lista prelegentów 150
API 150
Testy API 151
Usługa 156
Testy usługi 156
Czyste testy 160
Repozytorium 161
Wykorzystanie fabryki z FakeRepository 163
Szczegóły prelegentów 165
API 165
Testy API 165
Usługa 169
Testy usługi 169
Czyste testy 172
Coś więcej z repozytorium 172
Dodatkowa praca związana z fabryką 173
Testowanie przypadków wyjątkowych 174
Podsumowanie 176

Rozdział 8. Wyodrębnianie problemów na zewnątrz 177

Odseparowanie problemów 177
Gravatar 178
Planowanie na przyszłość 185
Abstrahowanie warstwy danych 185
Rozszerzanie wzorca repozytorium 186
Zapewnienie funkcji 191
Tworzenie generycznego repozytorium 210
Krok pierwszy: abstrahowanie interfejsu 210
Krok drugi: abstrahowanie klasy konkretnej 211
Krok trzeci: zmiana testów, aby wykorzystywały repozytorium generyczne 215
Entity Framework 219
Wstrzykiwanie zależności 222
Podsumowanie 223

Rozdział 9. Testowanie aplikacji napisanej w JavaScripcie 225

Tworzenie aplikacji za pomocą Reacta 226
Wyodrębnienie aplikacji 226
Konfiguracja bibliotek Mocha, Chai, Enzyme i Sinon 226
Plan 228
Komponent React 228
Rzut oka na testowalność Reduxa 229
Testy jednostkowe usługi API 230
Lista prelegentów 230
Imitacja usługi API 231
Akcja pobierania wszystkich prelegentów 235
Reduktor dla pobierania wszystkich prelegentów 239
Komponent listy prelegentów 240

Szczegóły prelegenta **246**
Rozbudowa imitacji usługi API 246
Akcja pobierania prelegenta 248
Reduktor dla pobierania prelegenta 254
Komponent dla szczegółów prelegenta 257
Podsumowanie **260**

Rozdział 10. Kwestia integracji **261**

Implementacja rzeczywistej usługi API **261**
Zamiana imitacji API na prawdziwą usługę 262
Wykorzystanie biblioteki Sinon do imitacji odpowiedzi Ajaxa 264
Konfiguracja aplikacji 273
Testy integracyjne od początku do końca **273**
Zalety 273
Wady 274
Ile testów end-to-end trzeba mieć? 274
Konfiguracja API **274**
Projekt dla testów integracyjnych 275
Gdzie zacząć? 275
Weryfikacja odwołań repozytorium do bazy 275
Weryfikacja, czy usługa odwołuje się do bazy przez repozytorium 278
Weryfikacja odwołań API do usługi 280
Podsumowanie **284**

Rozdział 11. Zmiany w wymaganiach **285**

Witaj, świecie **286**
Zmiana wymagań 286
FizzBuzz **287**
Nowa funkcja 287
Aplikacja TODO **289**
Oznaczanie jako wykonane 289
Dodanie testów 289
Kod produkcyjny 290
Dodanie testów 290
Kod produkcyjny 292
Zmiany w aplikacji Speaker Meet **293**
Zmiany po stronie serwera 293
Zmiany po stronie interfejsu 295
Co teraz? **296**
Przedwczesna optymalizacja 296
Podsumowanie **297**

Rozdział 12. Problem z kodem zastanym **299**

Czym jest kod zastany? **299**
Dlaczego kod staje się zły? 300
Kiedy projekt staje się projektem zastanym? 300
Co możemy zrobić, aby zapobiec powstawaniu kodu zastanego? 301
Typowe problemy wynikające z kodu zastanego **302**
Niezamierzone skutki uboczne 302
Nadmierna optymalizacja 303

Zbyt sprytny kod 304
Ścisłe łączenie z kodem zewnętrznym 304
Problemy przeszkadzające w dodawaniu testów 305
Bezpośrednia zależność od frameworka lub kodu zewnętrznego 305
Prawo Demeter 306
Praca w konstruktorze 306
Globalny stan 307
Metody statyczne 307
Duże klasy i funkcje 307
Radzenie sobie z problemami wynikającymi z kodu zastanego 308
Bezpieczna refaktoryzacja 308
Pierwsze testy 310
Idąc dalej 312
Naprawa błędów 313
Niebezpieczna refaktoryzacja 313
Podsumowanie 313

Rozdział 13. Sprzątanie bałaganu 315

Dziedziczenie kodu 315
Gra 316
Prośba o zmianę 316
Czasami dostajesz od życia cytryny 317
Zaczynamy 317
Abstrakcja klasy zewnętrznej 320
Niespodziewane dane wsadowe 324
Szukanie sensu w szaleństwie 329
Końcowe upiększanie 335
Gotowy na ulepszenia 337
Podsumowanie 349

Rozdział 14. Pokaż się z najlepszej strony 351

Co omówiliśmy? 351
Idąc naprzód 352
TDD to osobisty wybór 352
Nie potrzebujesz pozwolenia 353
Rozwijaj aplikacje poprzez testy 353
Wprowadzanie TDD do Twojego zespołu 353
Nie zmuszaj nikogo do TDD 354
Grywalizacja TDD 354
Pokaż zespołowi zalety 354
Kontroluj rezultaty 355
Powrót do świata jako ekspert TDD 355
Poszukaj mentora 355
Zostań mentorem 356
Praktykuj, praktykuj, praktykuj 356
Podsumowanie 357

Skorowidz 358