

Spis treści

Podziękowania (11)

Przedmowa (13)

Cytaty o TDD (15)

Wstęp (17)

Przewodnik po książce (19)

Dla kogo jest ta książka? (20)

W jakim stopniu muszę umieć programować? (20)

Nie jestem programistą C#. Czy ta książka ma jakąś wartość dla mnie? (20)

Umiem pisać testy jednostkowe i znam TDD. Czy ta książka jest dla mnie? (21)

Jestem menadżerem/dyrektorem/właścicielem (a więc nie programuję), a mój zespół chce wdrożyć TDD. Czy z tej książki dowiem się, czy warto? (22)

Jestem manualnym testerem. Czy powinienem znać TDD? (22)

Kontekst jest królem (23)

Atrapa to mock czy test double? (23)

xUnit a xUnit.net (24)

Funkcja a funkcjonalność oraz funkcyjny a funkcjonalny (24)

Synonimy (25)

Jak uczyć się metody Test-Driven Development? (25)

Dogmat (26)

Narzędzia użyte w tej książce (27)

Kod źródłowy do książki (28)

Rozdział 1. Wprowadzenie do TDD (29)

Błędy w oprogramowaniu (31)

Cykl Red-Green-Refactor (32)

Podsumowanie (33)

Rozdział 2. Co zyskujemy dzięki TDD? (35)

Wysoka jakość kodu (35)

Czy pisanie w metodyce TDD oznacza pisanie według zasad SOLID? (36)

Prostota kodu: YAGNI i KISS (37)

Żywa dokumentacja (38)

Lepsze zrozumienie wymagań biznesowych (38)

Automatyczna regresja (39)

Informacja zwrotna (41)

Mniej defektów (42)

Czas programowania jest krótszy (43)

Niższy koszt zmian (43)

Przypadki użycia (45)

Badanie Microsoft Research nad zespołami Microsoftu i IBM (45)

Pilotażowy projekt dla dużej firmy (45)

Mały projekt w parach (47)

Metaanaliza (47)

Podsumowanie (48)

Rozdział 3. Trudności przy wdrażaniu TDD (49)

Ścieżka nauki (49)

Dyscyplina (50)

Więcej narzędzi (50)

Początkowa percepcja dłuższego czasu potrzebnego do napisania kodu (51)

Jak "sprzedać" TDD kierownictwu? (51)

Musimy dostarczać szybko, a w naszym projekcie nie ma czasu na TDD (51)

Lista kontrolna: Definition of Done (52)

Podsumowanie (56)

Rozdział 4. Filozofia TDD (57)

Test-First czy Test-Last? (58)

Wszystkie testy naraz czy test po teście? (59)

Weryfikacja stanu czy zachowania? (60)

Jedna czy wiele asercji? (60)

Scenariusz 1.: Jedna oczekiwana zmienna wyjściowa (60)

Scenariusz 2.: Wiele oczekiwanych zmiennych wyjściowych (61)

Scenariusz 3.: Asercje pośrednie (62)

Wiele asercji w jednym teście (66)

Kiedy pisanie testów jednostkowych nie ma sensu? (67)

Czy należy pisać testy jednostkowe do bibliotek innych dostawców? (68)

Czy TDD sprawdza się dla małych aplikacji? (69)

Czy TDD sprawdza się dla dużych aplikacji? (70)

Czy testy jednostkowe zastępują testera? (70)

Jak pisanie testów jednostkowych wpływa na estymatę zadania? (71)

Czy testy jednostkowe można pisać w innym języku programowania niż pisany jest kod? (71)

Czy testy jednostkowe może pisać inna osoba? (72)

Czy system może mieć zbyt dużo testów jednostkowych? (72)

Czy testy jednostkowe to jedyne testy, jakie powinny znajdować się w aplikacji? (73)

Jakich testów powinno być najwięcej (piramida testów)? (74)

Podsumowanie (75)

Rozdział 5. Rodzaje testów (77)

Sposób wykonywania (77)

Wiedza na temat struktury systemu (test skrzynki) (77)

Poziom testowania (78)

Testowanie po zmianach dokonanych w systemie (80)

Testowanie нефunkcjonalne (80)

Test wydajnościowy (81)

Podsumowanie (82)

Rozdział 6. Test jednostkowy (83)

Struktura testu: Arrange-Act-Assert (83)

Alternatywne struktury testu (84)

Test jednostkowy a test integracyjny (85)

Myśl jak tester: ścieżki optymistyczne i przypadki brzegowe (86)

Jak nazywać klasy i metody testowe? (87)

Podział testów i projekty testowe (88)

Podsumowanie (89)

Rozdział 7. Nasz pierwszy test jednostkowy (91)

Wybór biblioteki do testowania (91)

Zanim zaczniemy... (92)

Dodanie solucji i projektów (92)

Dodanie biblioteki NUnit (93)

Etap red: pisanie testu do nieistniejącej metody (95)

Jak uruchomić test? (98)

Etap green: implementacja kodu (102)

Etap trzeci (i ostatni): refaktoryzacja kodu (103)

Podsumowanie (103)

Rozdział 8. Piszemy kolejne testy jednostkowe (105)

Drugi test jednostkowy (105)

Kolejne przypadki użycia (106)

Testy ułamków nieskończonych lub zaokrąglonych (108)

Testowanie wyrzucenia wyjątku (109)

Testowanie zdarzenia (111)

Podsumowanie (114)

Rozdział 9. Testowanie z NUnitem (115)

Asercje (116)

Model klasyczny i model oparty na twierdzeniach (118)

Operacja równości (120)

Porównanie dwóch typów wartościowych (121)

Porównanie dwóch typów referencyjnych (122)

Porównanie dwóch typów referencyjnych z nadpisanym operatorem porównania (122)

Tolerancja: delta i procent (124)

Tolerancja: czas (124)

Własna klasa obsługująca porównanie (125)

Metody pomocnicze (126)

Operacje porównania (126)

Własna klasa obsługująca porównanie (127)

Należy do zakresu (128)

Złożenia (128)

Testowanie typów (129)

Testowanie wyjątków (131)

- Testowanie, czy kod wyrzucił wyjątek (133)
- Testowanie, czy kod nie wyrzucił wyjątku (135)
- Testowanie parametru i komunikatu wyjątku (136)
- Testowanie wewnętrznego wyjątku (137)
- Typ tekstowy (137)
- Kolekcje (138)
- System plików (141)
- Komunikaty (142)
 - Własne komunikaty błędów (144)
 - Własne komunikaty informacyjne (145)
 - Komunikat a nazwa testu (146)
- Współdzielenie danych (147)
 - Kiedy korzystać ze współdzielenia danych? (148)
- Testy parametryzowane (150)
 - TestCase (151)
 - Values (152)
 - Range (154)
 - Random (154)
 - TestCaseSource (155)
 - ValueSource (160)
 - Testy oparte na zewnętrznych źródłach (160)
- Strategie łączenia wartości testowych (161)
 - Test kombinatoryczny (161)
 - Test sekwencyjny (162)
 - Test par (163)
- Teorie (166)
- Testowanie klas generycznych (170)

Zasada podstawienia Liskov (172)

Testowanie wywołań asynchronicznych (174)

Równoległe uruchamianie testów (178)

Poziom zrównoleglenia (179)

Kiedy zrównoleglić uruchamianie testów? (179)

Pozostałe atrybuty (179)

Sterowanie wątkiem (180)

Kategoria testu (180)

Atrybuty informacyjne (182)

Przekazywanie parametrów (182)

Ignorowanie testów (183)

Kolejność wykonywania testów (185)

Ustawienia regionalne (185)

Powtarzanie testu (188)

Czas wykonywania testu (189)

Platforma (190)

Atrybuty a testy parametryzowane (192)

Podsumowanie (192)

Rozdział 10. Testowanie zależności i atrapy obiektów (193)

Ręczne tworzenie atrapy (195)

Kryterium akceptacji nr 1: wiek klienta niższy niż 18 lat (196)

Kryterium akceptacji nr 2: wiek klienta większy bądź równy 18 lat (198)

Kryterium akceptacji nr 3: jeśli obiekt klienta jest nullem, to wyrzucić wyjątek (200)

Podsumowanie (200)

Wprowadzenie do frameworku Moq (201)

Składnia imperatywna i deklaratywna (202)

- Składnia imperatywna (203)
- Składnia deklaratywna (204)
- Wybór składni (205)
- Atrapa rekursywna (recursive mock) (205)
- Tryb zachowania właściwości (stubbing) (206)
- Zwracanie domyślnej wartości (207)
- Atrapa z sekwencyjnym rezultatem (208)
- Tryb zachowania atrapy (MockBehavior) (209)
- Przekazywanie parametrów w metodzie (argument matchers) (210)
 - Ponowne użycie matcherów (215)
- Weryfikacja wywołań (216)
 - Weryfikacja wywołania metody (217)
 - Weryfikacja dostępu i zapisu właściwości (219)
 - Testować stan czy zachowanie? (220)
 - Komunikat błędu (221)
 - Podsumowanie (222)
- Wywołanie zwrotne: Callback (222)
 - Podsumowanie (224)
- Wywołanie składowej bazowej: CallBase (224)
- Atrapa wyrzucająca wyjątek (226)
- Inne poziomy dostępności (227)
 - protected (228)
 - internal (229)
 - Podsumowanie (229)
- Klasyfikacja atrap (230)
 - Dummy (230)
 - Stub (232)

Fake (233)

Mock (235)

Spy (235)

Podsumowanie (236)

Ograniczenia Moqa (237)

Tworzenie atrap dla klas i metod statycznych (238)

Rodzaje bibliotek do tworzenia atrap (241)

Constrained (241)

Unconstrained (241)

Constrained czy unconstrained? (242)

Podsumowanie (243)

Rozdział 11. Dobre praktyki pisania testów jednostkowych (245)

Test powinien być szybki, bardzo szybki! (246)

Testy powinny być odizolowane i niezależne od siebie (247)

Test powinien być powtarzalny (247)

Test powinien być deterministyczny (248)

Test nie powinien mieć zależności zewnętrznych (248)

Test nie powinien mieć konfiguracji (249)

Wynik testu nie powinien być interpretowany (249)

Test nie powinien być pusty (250)

Załączek kodu powinien wyrzucać wyjątek (250)

Test powinien mieć jedną logiczną asercję (251)

Testy nie powinny być dyskryminowane (251)

Testy powinny być podzielone według kategorii (251)

Test powinien mieć strukturę Arrange-Act-Assert (251)

Test powinien obejmować ścieżki optymistyczne i przypadki brzegowe (252)

Test powinien mieć odpowiednią nazwę (252)

Testowane powinny być tylko publiczne składowe (252)

Test powinien oczekiwać konkretnego typu wyjątku (253)

Test powinien oczekiwać wyjątku w konkretnym wyrażeniu (253)

Test nie powinien zawierać instrukcji warunkowych i pętli (253)

Test powinien mieć wartości oczekiwane wpisane "na sztywno" (254)

Test powinien mieć asercję (255)

Test powinien być nieskomplikowany (255)

Test nie powinien być "przespecyfikowany" (256)

Test nie powinien zawierać metod z atrybutami SetUp i TearDown (257)

Klasa testowa powinna być bezstanowa (257)

Komunikaty asercji nie powinny być nadmiarowe (258)

Podsumowanie (259)

Rozdział 12. TDD i istniejący kod (261)

Refaktoryzacja bezpieczna i mniej bezpieczna (261)

Przykład bezpiecznej refaktoryzacji (264)

Dodawanie testów do istniejącego kodu (271)

Gdzie zacząć dodawać testy? (271)

Jak pisać testy? (272)

Narzędzia (274)

Podsumowanie (275)

Rozdział 13. Pokrycie kodu testami (277)

Co to jest pokrycie kodu testami? (277)

Narzędzia do mierzenia pokrycia kodu (278)

W ile procent pokrycia powinniśmy celować? (281)

Przykłady "fałszywych" testów o stuprocentowym pokryciu kodu (281)

Podsumowanie (283)

85%, 90% czy 100%? (283)

Pokrycie kodu jako narzędzie do identyfikowania brakujących testów (284)

Rozdział 14. Ciągła integracja (285)

Serwer ciągłej integracji (286)

Ciągła dostawa i ciągłe wdrażanie (288)

Podsumowanie (289)

Dodatek A. Biblioteki do testowania (291)

Dodatek B. Biblioteki do tworzenia atrap (293)

Dodatek C. Biblioteki do mierzenia pokrycia kodu testami (297)

Dodatek D. Testy z danymi zewnętrznymi - przypadek użycia (299)

Dodatek E. Rozszerzalność NUnita (303)

Atrybut informacyjny (305)

Atrybut umożliwiający współdzielenie danych (307)

Dodatek F. Bibliografia (311)

Źródła internetowe (314)

Skorowidz (319)