

O autorze XV

O recenzencie technicznym XVII

Podziękowania XIX

Wprowadzenie XXI

Kto powinien przeczytać tę książkę? XXII

Jak zorganizowana jest ta książka? XXII

Część I: „Dlaczego” oraz „Co” XXIII

Część II: „Jak” XXIII

CZĘŚĆ I. „Dlaczego” oraz „co ”

1. Wartość automatyzacji testów

Dlaczego potrzebujemy automatyzacji testów?

Od modelu kaskadowego do zwinnego tworzenia oprogramowania

Koszt złożoności oprogramowania

Utrzymywanie stałego kosztu

Refaktoryzacja

Ciągłe doskonalenie

2. Od testowania ręcznego do automatycznego

Podejście pierwsze: nagrywanie i odtwarzanie

Uzyskiwanie maksimum korzyści z automatyzacji testów

Różnice pomiędzy testami manualnymi i automatycznymi

Testowanie eksploracyjne

Rozważania dotyczące testowania automatycznego

3. Ludzie i narzędzia

Wybieranie właściwych narzędzi

Kto powinien pisać testy?

Promowanie testerów manualnych lub niedoświadczonych deweloperów do rangi deweloperów automatyzacji

Dzielenie pracy między testerów manualnych i deweloperów automatyzacji

Korzystanie z dedykowanego zespołu automatyzacji

Dedykowany deweloper automatyzacji wewnątrz każdego zespołu

Dawanie deweloperom pełnej odpowiedzialności za automatyzację

Różnorodność narzędzi

Klasyfikacja narzędzi

IDE i języki programowania

Biblioteki testowania (jednostkowego)

Biblioteki w stylu BDD

Technologie zapewniające interakcję z testowanym systemem

Pakiety do zarządzania testami

Narzędzia kompilacji oraz potoki ciągłej integracji lub ciągłego dostarczania

Inne czynniki mające znaczenie przy wybieraniu narzędzi

4. Osiągnięcie pełnego pokrycia

W jaki sposób mierzymy pokrycie?

Procent przypadków testów manualnych pokrytych przez automatyzację

Procent pokrytych funkcji

Procent pokrycia kodu

Uzyskiwanie korzyści przed osiągnięciem pełnego pokrycia

Co robimy po osiągnięciu pełnego pokrycia?

W jaki sposób uzyskać 100% pokrycia?

Odwracanie koła

Mapa drogowa prowadząca do pomyślnego projektu automatyzacji

Kiedy rozpocząć pracę nad progresją?

Nadawanie priorytetu pracy w celu zlikwidowania luki w regresji

5. Procesy biznesowe

Regularne uruchamianie testów

Najprostsze podejście

Testowanie nocne

Obsługiwanie błędów wykrywanych przez automatyzację

Zachowywanie testów kończących się niepowodzeniem

Wykluczanie testów kończących się niepowodzeniem

Tworzenie obejść w teście

Traktowanie wszystkich niepowodzeń automatyzacji jako błędów krytycznych

Ciągła integracja

Tworzenie oprogramowania sterowane testami akceptacyjnymi

Ciągłe dostarczanie i ciągłe wdrażanie

Wydania kanarkowe

Podsumowanie

6. Automatyzacja i architektura testów

Założenia dotyczące do architektury testów

Poznanie architektury testowanego systemu

Powrót do podstaw: czym jest system komputerowy?

Czym jest test automatyczny?

Rzeczywiste systemy komputerowe

Alternatywy i założenia w architekturze warstwowej

Związki między zakresem a testem

Omówienie warstw

Alternatywne zakresy testowania

Rzeczywista architektura

Architektura planowana kontra architektura rzeczywista

Typowe warianty

Łączenie testów

Podsumowanie czynników

Co poza architekturą warstwową?

Podsumowanie: dokonywanie własnych wyborów

7. Izolacja i środowiska testowe

Stan

Problemy z izolacją i ich rozwiązania

Problem 1 – testy manualne i test automatyczny wykonywane w różnym czasie

Problem 2 – testy manualne i automatyczne wykonywane jednocześnie

Problem 3 – kolejność ma znaczenie

Problem 4 – testy automatyczne uruchamiane jednocześnie

Techniki izolacji

Korzystanie z oddzielnych kont

Osobne bazy danych dla testów manualnych i automatyzacji testów

Oddzielne środowisko dla każdego członka zespołu

Resetowanie środowiska przed każdym cyklem testowania

Tworzenie niepowtarzalnych danych dla każdego testu

Każdy test czyści wszystko, co utworzył

Współdzielone dane tylko do odczytu

Podsumowanie

8. Szersza perspektywa

Relacje między architekturą oprogramowania i strukturą biznesu

Prawo Conwaya

Zespoły pionowe kontra zespoły poziome

Zależności między architekturą oprogramowania i strukturą organizacyjną z automatyzacją testów

Dedykowany zespół automatyzacji

Deweloperzy automatyzacji w zespołach poziomych

Deweloperzy automatyzacji w zespołach pionowych

Elastyczna struktura organizacyjna

Ekspert ds. automatyzacji

Podsumowanie

CZĘŚĆ II. „Jak”

9. Przygotowanie do samouczka

Wymagania i założenia wstępne

Stosowanie procesu do istniejących systemów automatyzacji testów

Omówienie procesu

„Z dołu do góry” albo „z góry do dołu”

Proces

Poznanie testowanego systemu

Omówienie projektu MVCForum

Przygotowanie środowiska pod samouczek

Instalowanie Visual Studio w edycji Community

Pobieranie i instalowanie przeglądarki Chrome

Pobieranie i instalowanie bazy danych SQL Server Express

Pobieranie i budowanie aplikacji

Instalacja dodatku ReSharper (krok opcjonalny)

Korzystanie z narzędzia Git z poziomu Visual Studio

Przełączanie pomiędzy gałęziami

Podsumowanie

10. Projektowanie pierwszego przypadku testowego

Wybieranie pierwszego testu do zautomatyzowania

Wybieranie pierwszego przypadku testowego dla aplikacji MVCForum

Naukowa metoda projektowania przypadku testowego

Projektowanie kroków testu

Myślenie w kontekście obiektów i jednostek

Wzorzec obiektu strony

Podsumowanie

11. Kodowanie pierwszego testu

Tworzenie projektu

Modyfikowanie nazw klas, plików i metod testowych

Pisanie pseudokodu

Uwagi odnośnie do pseudokodu

Uzupełnianie kodu w celu jego skompilowania

Deklarowanie klasy LoggedInUser

Deklarowanie właściwości MVCForum

Deklarowanie metody RegisterNewUserAndLogin

Deklarowanie pozostałych klas i metod

Omówienie kodu modelu

Podsumowanie

12. Uzupełnianie pierwszego testu

Uruchamianie testu w celu znalezienia pierwszej metody do zaimplementowania
Dodawanie Selenium do projektu
Uruchamianie IISExpress
Implementowanie konstruktora MVCForumClient
Implementowanie metody RegisterNewUserAndLogin
Prośzenie dewelopera o dodanie unikalnego identyfikatora automatyzacji
Implementowanie metod ustawiających dla właściwości
Usuwanie duplikacji z metod ustawiających właściwości
Napotykanie błędu izolacji
Implementowanie metody CreateDiscussion i analizowanie niepowodzenia
Kończenie testu
Podsumowanie

13. Badanie niepowodzeń

Integrowanie z najnowszą wersją aplikacji MVCForum
Usprawnianie raportowania błędów
Unikanie debugowania
Badanie głównej przyczyny
Rozwiązywanie problemu
Więcej problemów...
Rejestrowanie oraz inne formy zbierania dowodów
Przechwytywanie ekranu
Rejestrowanie
Rejestrowanie zagnieżdżone
Rejestrowanie wizualne
Dodatkowe opcje rejestrowania i diagnozowania
Dodawanie zagnieżdżonego rejestratora wizualnego do testów aplikacji MVCForum
Badanie trudniejszych niepowodzeń
Niepowodzenia, które zdarzają się tylko na jednej maszynie
Badanie testów wpływających na inne testy
Badanie testów migoczących
Podsumowanie

14. Dodawanie kolejnych testów

Pisanie kolejnych testów
Planowanie kolejnych testów
Dodawanie testu: dyskusje mogą być filtrowane według kategorii
Podsumowanie procesu dodawania drugiego testu
Wprowadzanie dodatkowych usprawnień
Tworzenie bardziej zrozumiałych identyfikatorów
Organizowanie kodu w foldery
Wyodrębnianie klasy bazowej dla testów
Obsługa wielu użytkowników i przeglądarek
Wskazówki w zakresie korzystania z plików konfiguracyjnych testów
Obsługiwanie wielu przeglądarek
Dodatkowe możliwości usprawniania
Automatyczne ponowne tworzenie bazy danych
Oczyszczanie
Poprawienie wydajności
Dodawanie kolejnych testów
Testy sterowane danymi
Podsumowanie

15. Ciągła integracja

Czy to naprawdę konieczne?

Tworzenie procesu kompilacji testów

Planowanie procesu kompilacji testów

Tworzenie procesu automatycznego wdrażania

Dodawanie testów do kompilacji

Zmiana procesu tworzenia oprogramowania i kultury

Dążenie do „Świętego Graala”

Co jest potrzebne do zmiany kultury?

Określanie punktu wyjścia

Skracanie czasu wykonywania testów

Ulepszanie izolacji

Realizowanie wymagań wstępnych za pośrednictwem API

Równoległe wykonywanie i wirtualizacja

Uruchamianie wyłącznie testów poprawności w ramach ciągłej integracji

Dzielenie potoku CI na etapy

Pisanie głównie testów integracyjnych i jednostkowych

Uruchamianie testów wyłącznie dla konkretnych komponentów

Optymalizowanie wydajności testów

Pokrywanie szerszej macierzy

Podsumowanie

16. Tworzenie oprogramowania sterowane testami akceptacyjnymi (ATDD)

Omówienie metodyki ATDD

Bycie bardziej zwinnym

Dług techniczny

Co sprawia, że zespół jest zwinny?

Unikanie długu technicznego

Proces

Tworzenie historyjki użytkownika

Pisanie testów automatycznych

Dostarczanie aplikacji i zbieranie opinii na jej temat

Używanie testów akceptacyjnych jako dokumentacji

Wiązanie kroków zamiast testów

Kompromis między możliwością ponownego użycia, poziomem szczegółów i czytelnością

Wprowadzanie metodyki ATDD do istniejącego projektu

Rozpoczynanie bez testów automatycznych

Retrospektywna implementacja automatyzacji

Rozpoczynanie od naprawy błędów

Zwiększanie pokrycia regresji

Podsumowanie

17. Test jednostkowe i tworzenie oprogramowania sterowane testami (TDD)

Przyswajanie testów jednostkowych i TDD

Sposoby pisania testów jednostkowych

Mechanizm biblioteki testów jednostkowych

Sposób pisania testu jednostkowego

Testy jednostkowe i operacje wejścia/wyjścia

Mechanizm działania TDD

Czerwone-zielone-refaktoryzacja

Dlaczego najpierw powinniśmy pisać testy?

Prawdziwe wyzwania w testowaniu jednostkowym i TDD

Główne wyzwania związane z testowaniem jednostkowym
Główne wyzwania związane z podejściem TDD
Bardziej szczegółowe wyzwania
Opanowywanie czystego kodu i zasad SOLID
Opanowywanie umiejętności refaktoryzowania
Największe wyzwanie: co testować?
Używanie metodyki TDD w celach, do jakich była projektowana
Podejście „z zewnątrz do środka” kontra podejście „od środka na zewnątrz”
Podsumowanie

18. Inne rodzaje testów automatycznych

Testy wydajności
Mierzenie wydajności w środowisku produkcyjnym
Czego nie robić?
Definiowanie oczekiwanego rezultatu
Ponowne wykorzystywanie kodu pomiędzy testami funkcjonalnymi i testami wydajności
Badanie wąskich gardeł w wydajności
Wydajność postrzegana a wydajność rzeczywista
Testy obciążeniowe
Jak działają testy obciążeniowe
Definiowanie oczekiwanego rezultatu
Łączenie testów wydajności z testami obciążeniowymi
Uruchamianie testów w środowisku produkcyjnym
Testowanie wdrożenia
Testowanie stanu zdrowia środowiska produkcyjnego
Które testy uruchamiać?
Oczyszczanie danych testu
Testowanie wizualne
Przepływ pracy testowania wizualnego
Testowanie wizualne i testowanie w wielu przeglądarkach/na wielu platformach
Testy instalacji
Podejścia dla testów instalacji
Testowanie instalacji za pośrednictwem interfejsu użytkownika lub instalacji dyskretnej
Testowanie programu deinstalacyjnego
Testy aktualizacji
Podejścia dla testów aktualizacji
Testowanie algorytmów statystycznych, niedeterministycznych i sztucznej inteligencji
Sposoby testowania algorytmów statystycznych
Testowanie aplikacji, które wykorzystują liczby losowe
Testowanie aplikacji analityki biznesowej
Podsumowanie

19. Co dalej?

Popelniaj błędy
Słuchaj, konsultuj się i zasięgaj porad
Poznaj i dostosuj się do celów swojego biznesu
Poznaj swoje narzędzia
Doskonalenie umiejętności programistycznych
Doskonalenie umiejętności w zakresie zapewniania jakości
Poszerzaj swoje horyzonty
Dzielenie się wiedzą
Dziel się własnymi narzędziami

Bawmy się dobrze!

Dodatek A. Rzeczywiste przykłady

Przykład 1 – system monitorowania wodomierzy

Symulowanie serwera komunikacji

Praca z usługą Google Maps

Przykład 2 – system do handlu na rynku Forex

Rozwiązanie

Niestabilność powodowana przez CRM

Izolowanie środowisk

Testowanie aplikacji mobilnej z użyciem abstrakcyjnego zakresu testowania

Przykład 3 – zarządzanie sklepem detalicznym

Opis architektury

Wdrożenie minimalne

Struktura organizacyjna

Rozwiązania automatyzacji testów

Symulator daty i godziny

Testy dla trzech warstw

Testy kompleksowe

Dodatek B. Mechanizm oczyszczania

Wywołania zwrotne i delegaty

Budowanie mechanizmu oczyszczania

Problem

Proste rozwiązanie

Ponowne wykorzystywanie mechanizmu oczyszczania

Obsługiwanie zależności pomiędzy akcjami oczyszczającymi

Obsługiwanie wyjątków w akcjach oczyszczających

Podsumowanie

Dodatek C. Projekt „Test Automation Essentials”

Kontekst

Struktura projektu

Uwaga dotycząca testów jednostkowych i komentarzy XML

Pakiety NuGet

Funkcje i narzędzia

TestAutomationEssentials.Common

TestAutomationEssentials.MSTest

TestAutomationEssentials.CodedUI

TestAutomationEssentials.Selenium

Pomoc w tworzeniu projektu i przenoszenie na inne języki

Dodatek D. Wskazówki i praktyki zwiększające produktywność programisty

Preferuj korzystanie z klawiatury

Poka-Yoke

Unikaj wartości Null

Unikaj przechwytywania wyjątków

Wybieranie najbardziej odpowiedniego lokalizatora

Trwale zakodowane ciągi znaków w automatyzacji testów: za i przeciw

Indeks