Wprowadzenie

Istnieje wiele świetnych książek na temat automatyzacji testów, a w szczególności na temat najlepszych praktyk w tym zakresie. Jednak żadna z tych książek nie jest uniwersalna. Jak to ktoś kiedyś powiedział: „Te ‚najlepsze praktyki' są zawsze kontekstowe: nawet coś tak powszechnego jak oddychanie może mieć katastrofalne skutki, jeśli kontekstem będzie swobodne nurkowanie…".

Większość książek, które przeczytałem do tej pory na temat automatyzacji testów, skierowana jest w dużej mierze do deweloperów i skupia się głównie na testach jednostkowych lub pisanych przez deweloperów testach kompleksowych. Inne książki, które albo przeczytałem, albo o których słyszałem, poświęcone są konkretnej technologii automatyzacji testów, konkretnej metodyce, lub po prostu są już zbyt nieaktualne. Choć ogólnie zgadzam się z tym, że idea, zgodnie z którą to deweloperzy piszą testy, może być w wielu sytuacjach bardzo efektywna, to w rzeczywistości nie pasuje ona do wszystkich organizacji na wszystkich etapach. Co więcej, automatyzacja testów jest narzędziem, które służy i ma wpływ niemal na wszystkich interesariuszy organizacji tworzącej oprogramowanie, wliczając w to testerów, menedżerów produktu, architektów oprogramowania, ludzi z zespołów DevOps oraz menedżerów projektów, a nie tylko deweloperów. Ponieważ każda organizacja i każdy projekt oprogramowania jest inny, próba dostosowania technik, praktyk i narzędzi, które nie pasują do potrzeb lub umiejętności danego zespołu, może doprowadzić do niepowodzenia projektu automatyzacji testów, a w niektórych przypadkach nawet do upadku całego projektu oprogramowania.

Książka ta ma na celu zaprezentowanie szerokiego poglądu na temat automatyzacji testów, aby umożliwić czytelnikowi podejmowanie mądrych decyzji dotyczących jego konkretnego przypadku – biorąc przy tym pod uwagę jego ograniczenia i korzyści, jakie chce on uzyskać dzięki automatyzacji testów – ale również dostarczenie szczegółowych i praktycznych porad w zakresie efektywnej budowy automatyzacji testów, a przynajmniej dla większości przypadków.

Kto powinien przeczytać tę książkę?

Ponieważ automatyzacja testów wywiera wpływ na prawie wszystkich interesariuszy organizacji tworzącej oprogramowanie i w tej książce staramy się omówić prawie każdy aspekt automatyzacji testów, jest ona przeznaczona dla każdego, kto jest zaangażowany w proces tworzenia oprogramowania i chce dowiedzieć się, w jaki sposób można uzyskać więcej korzyści z automatyzacji testów. Do grona tych osób zaliczyć można: menedżerów zespołów zapewniania jakości, menedżerów zespołów deweloperów, deweloperów, testerów, architektów, menedżerów produktu (nazywanych również analitykami biznesowymi, analitykami systemu lub jeszcze inaczej), ludzi z zespołów DevOps itd. No i oczywiście deweloperów automatyzacji testów, których głównym zadaniem jest tworzenie testów automatycznych…

Znaczna część tej książki nie ma zbyt technicznego charakteru i skierowana jest do szerszego odbiorcy, jednak rozdziały od 11 do 14 są bardzo techniczne i skierowane do osób, które piszą kod i są dobrze zaznajomione z programowaniem obiektowym – w szczególności mam tu na myśli profesjonalnych deweloperów automatyzacji testów. Kod w tej części napisany został w języku C#, ale same koncepcje i pojęcia można z łatwością przenieść na inny obiektowy język programowania. Ponieważ języki C# i Java są do siebie podobne, programiści Java nie powinni mieć większego problemu ze zrozumieniem tego kodu. Jestem jednak przekonany, że również programiści innych języków będą w stanie łatwo go zrozumieć, a przynajmniej jego główne idee.

W szczególności mam nadzieję, że książkę tę przeczyta wielu menedżerów zespołów deweloperów i zapewniania jakości, ponieważ zwykle to oni mają największy wpływ na kształtowanie metodyki i procesów pracy w swojej organizacji, z którymi to automatyzacja testów powinna się integrować i wspomagać ich rozwój. Ponadto książka ta zawiera wskazówki i techniki przydatne dla osób niebędących menedżerami, pozwalające im usprawniać stosowane w organizacji metodyki i procesy pracy nawet bez żadnej formalnej władzy.

Jak zorganizowana jest ta książka?

Gdy po raz pierwszy usiadłem do pisania tej książki, starałem się myśleć o jej ogólnej strukturze, ale zorientowałem się, że będzie to bardzo trudne zadanie, ponieważ wygląda na to, że prawie każdy temat jest powiązany z wieloma innymi tematami. W tamtym czasie nie mogłem znaleźć przejrzystego i logicznego sposobu podzielenia jej treści na ogólne części, tak więc napisałem „gruntowny spis" tematów, które chciałem w tej książce omówić i po prostu zacząłem je pisać, przelewając swoją wiedzę bezpośrednio na papier (a mówiąc bardziej precyzyjnie, na klawiaturę…). Naturalnie rozpocząłem od najprostszych i najbardziej ogólnych rzeczy, a następnie stopniowo rozbudowywałem je o kolejne, bardziej zaawansowane i szczegółowe rozdziały. Ponieważ tematy te są ze sobą ściśle powiązane, często pisałem fragmenty odwołujące się do tematu, którego jeszcze nie napisałem, a przy bardziej zaawansowanych tematach odwoływałem się do wcześniejszych rozdziałów. Tak więc ostatecznie, niczym w dobrym projekcie Agile (a skoro już mowa o odwołaniach do innych rozdziałów, to zobacz rozdział 1, zawierający więcej informacji na temat metodyki

Agile), ogólna struktura tej książki zaczęła się stopniowo ujawniać. W pewnym momencie zdałem sobie sprawę, że książka przybrała dosyć logiczną strukturę złożoną z dwóch części: pierwsza część odpowiada bardziej na ogólne pytania typu „dlaczego" oraz „co", zaś druga część odpowiada na bardziej szczegółowe i techniczne pytania typu „jak".

Ogólnie zachęcam czytelników do przeczytania całej książki od początku do końca. Ponieważ jednak książka ta skierowana jest do szerokiego grona odbiorców o różnych problemach, umiejętnościach, zainteresowaniach, potrzebach itd., można również skupić się na lekturze wyłącznie konkretnych rozdziałów, przeglądając lub nawet pomijając pozostałe. Można przy tym też skakać w przód i w tył do innych rozdziałów wspominanych w aktualnie czytanym fragmencie, aby w razie potrzeby uzupełnić swoją wiedzę. Wreszcie warto zawsze trzymać tę książkę w pobliżu, aby skorzystać z niej później, gdy zastosowanie automatyzacji testów w organizacji wystarczająco dojrzeje i zacznie stawiać czoło nowym wyzwaniom.

Oto przegląd poszczególnych części i rozdziałów tej książki:

Część I: „Dlaczego" oraz „Co"

Ta część omawia temat automatyzacji testów pod kątem wielu różnych aspektów, ale w bardziej „ogólny" sposób. Ta część książki jest niezbędna dla tych, którzy nie mają dużego doświadczenia z automatyzacją testów i chcą się dowiedzieć, jak wpasowuje się ona w szeroki obraz tworzenia oprogramowania, oraz od czego można zacząć. Zawarte w niej rozdziały pomogą nam również zrozumieć to, czego możemy, a także czego nie powinniśmy oczekiwać od automatyzacji testów. Jest to szczególnie istotne dla menedżerów zespołów deweloperów i zapewniania jakości, ponieważ omawiają one takie aspekty jak struktura biznesu, procesy pracy, architektura itd. Ta część książki pomoże nam przy podejmowaniu wielu decyzji, jakie nas czekają (czego wiele osób nie bierze nawet pod uwagę!) i pokaże nam, jaki wpływ może mieć każda z nich. Nawet jeśli nie jesteśmy menedżerami i uważamy, że nie mamy żadnego wpływu na te rzeczy, powinniśmy przeczytać rozdziały z tej części, aby zrozumieć ograniczenia i zalety w naszej obecnej sytuacji, a także być w stanie lepiej komunikować je naszym menedżerom.

Jeśli mamy już doświadczenie z automatyzacją testów, to ta pierwsza część pomoże nam poszerzyć w tym temacie nasze horyzonty i pokaże nam opcje oraz konsekwencje związane z decyzjami, które wcześniej podjęliśmy w mniej świadomy sposób.

Część II: „Jak"

Po ogólnym zapoznaniu się z dziedziną automatyzacji testów, czas zakasać rękawy i zacząć pisać testy wraz z wymaganą infrastrukturą. Po napisaniu kilku testów wyjaśniamy, w jaki sposób możemy zrobić krok na przód i najbardziej efektywnie wykorzystać automatyzację testów w cyklu tworzenia oprogramowania.

Od strony merytorycznej rozdziały w tej części można podzielić na dwie grupy (przy czym podział ten nie jest nigdzie jawnie podany, z wyjątkiem tego miejsca): rozdziały od 9 do 14 pisane są jako praktyczny samouczek, w ramach którego projektujemy i tworzymy system automatyzacji testów wraz z kilkoma testami (z użyciem narzędzia Selenium) dla istniejącego projektu open source, zaś rozdziały od 15 do 19 stanowią przewodnik po wykorzystywaniu automatyzacji testów w najbardziej efektywny sposób, pokazując przy tym, jak wyciągnąć z niej maksimum korzyści.

Większość rozdziałów z tej pierwszej grupy ma bardzo techniczny charakter, w przeciwieństwie do rozdziałów drugiej grupy. Z tego powodu pierwsza grupa rozdziałów jest bardziej odpowiednia dla deweloperów, a w szczególności dla deweloperów automatyzacji testów posiadających umiejętności w zakresie programowania obiektowego, natomiast druga grupa rozdziałów może być użyteczna dla każdego. Doświadczonych deweloperów zachęcam do podążania za samouczkiem krok po kroku i wykonywania wszystkich kroków samodzielnie, aby mogli oni *doświadczyć* ich w lepszym stopniu. Osoby, które nie potrafią programować, powinny przejrzeć te bardziej techniczne rozdziały w celu zapoznania się z głównymi koncepcjami, które są w nich zawarte, nawet jeśli osoby te nie zamierzają implementować ich w swoim własnym projekcie.

Oto kompletny opis rozdziałów:

Część I:

- **Rozdział 1: Wartość automatyzacji testów** – w tym rozdziale wyjaśniono, dlaczego automatyzacja testów jest potrzebna i jakie są jej krótko- i długoterminowe korzyści.
- **Rozdział 2: Od testowania ręcznego do automatycznego** – ten rozdział zawiera omówienie różnic pomiędzy testowaniem ręcznym i automatycznym oraz początek nakreślenia realistycznych oczekiwań dotyczących automatyzacji testów, ponieważ znacząco różni się ona od zwyczajnie szybszych testów manualnych.
- **Rozdział 3: Ludzie i narzędzia** – w tym rozdziale wyjaśniono, kto powinien pisać testy i infrastrukturę automatyzacji, oraz jakie są konsekwencje stosowania alternatywnych rozwiązań. Dodatkowo omówiono sposób dobierania właściwych narzędzi w zależności od wybranej opcji.
- **Rozdział 4: Osiąganie pełnego pokrycia** – w tym rozdziale nakreślono realistyczne oczekiwania dla długoterminowej mapy drogowej projektu automatyzacji, a także pokazano, w jaki sposób możemy zacząć czerpać z niej korzyści jeszcze na długo przed tym, jak automatyzacja zastąpi większość manualnych testów regresji.
- **Rozdział 5: Procesy biznesowe** – w tym rozdziale wyjaśniono, w jaki sposób automatyzacja testów powiązana jest z procesami biznesowymi wytwarzania oprogramowania i podano ogólny zarys tematów, które omawiane są bardziej szczegółowo pod koniec tej książki.
- **Rozdział 6: Automatyzacja i architektura testów** – w tym rozdziale omówiono sposób, w jaki automatyzacja testów jest powiązana z architekturą testowanego systemu, oraz dlaczego ważne jest, aby były one do siebie dostosowywane.
- **Rozdział 7: Izolacja i środowiska testowe** – w tym rozdziale wyjaśniono, w jaki sposób należy planować automatyzację testów oraz jej środowiska wykonywania, aby

zagwarantować, że testy są wiarygodne i nie mają na nie wpływu żadne niepożądane efekty.

- **Rozdział 8: Szersza perspektywa** – w tym rozdziale omówiono wzajemne zależności pomiędzy wszystkimi tematami omawianymi w poprzednich rozdziałach – głównie między architekturą, strukturą biznesu, procesami biznesowymi i oczywiście automatyzacją testów. Omówiono również sposób, w jaki wszystkie te tematy odnoszą się do kultury biznesu.

Część II:

- **Rozdział 9: Przygotowanie do samouczka** – ten rozdział zawiera opis proces wykorzystywany w ramach samouczka, który ma również zastosowanie w większości projektów automatyzacji testów. W rozdziale tym pokazano również, jak można skonfigurować własną maszynę, aby móc samodzielnie wykonywać kolejne kroki tego samouczka.
- **Rozdział 10: Projektowanie pierwszego przypadku testowego** – w tym rozdziale uczymy się konkretnej techniki projektowania przypadków testowych w sposób najlepiej pasujący do testów automatycznych.
- **Rozdział 11: Kodowanie pierwszego testu** – w tym rozdziale pokazano, w jaki sposób możemy rozpocząć pisanie kodu dla pierwszego testu. Zaczynamy od napisania prostego szkieletu testu, w sposób, który pozwoli nam zaprojektować i utworzyć modułową infrastrukturę do wielokrotnego użytku. Pod koniec tego rozdziału nasz test będzie się kompilował, ale nie będzie on wykonywał jeszcze żadnej pracy.
- **Rozdział 12: Uzupełnianie pierwszego testu** – w tym rozdziale kończymy pracę, którą zaczęliśmy w rozdziale poprzednim. Pod koniec tego rozdziału będziemy mieć działający test oraz dobrze zaprojektowaną infrastrukturę, która będzie go obsługiwać.
- **Rozdział 13: Badanie niepowodzeń** – w tym rozdziale ćwiczymy sposób badania i radzenia sobie z rzeczywistym niepowodzeniem testu, które miało miejsce w nowej kompilacji testowanego systemu, oraz tworzenia raportu, który pomoże nam zbadać dodatkowe niepowodzenia w przyszłości.
- **Rozdział 14: Dodawanie kolejnych testów** – w tym rozdziale dodajemy jeden dodatkowy test. Ponadto omawiamy sposób dodawania coraz większej liczby testów przy jednoczesnym rozszerzaniu i usprawnianiu wspierającej ich infrastruktury, w tym obsługę testowania w wielu przeglądarkach, obsługę wielu środowisk i znacznie więcej.
- **Rozdział 15: Ciągła integracja** – w tym rozdziale (rozpoczynającym drugą grupę rozdziałów z części II) wyjaśniono, w jaki sposób możemy integrować testy do postaci kompilacji ciągłej integracji. Poza aspektami technicznymi, w rozdziale tym pokazano, jak zapewnić sukces ciągłej integracji jako narzędzia organizacyjnego i podano porady dla osób bez doświadczenia w programowaniu, jak stopniowo zmieniać na lepsze kulturę i procesy danej organizacji poprzez wykorzystywanie zalet ciągłej integracji.

- **Rozdział 16: Tworzenie oprogramowania sterowane testami akceptacyjnymi (ATDD)** – w tym rozdziale wyjaśniono korzyści ze stosowania oraz sposób implementacji metodyki tworzenia oprogramowania sterowanego testami akceptacyjnymi, która dzięki wykorzystaniu ciągłej integracji obejmuje cały cykl tworzenia oprogramowania i pomaga zespołowi efektywnie wykorzystywać metodykę Agile.
- **Rozdział 17: Testy jednostkowe i tworzenie oprogramowania sterowane testami (TDD)** – w tym rozdziale omówiono techniki, które tradycyjnie przypisywane są wyłącznie programistom aplikacji: testy jednostkowe i tworzenie oprogramowania sterowane testami, ale są tak naprawdę nieodłączną częścią automatyzacji testów.
- **Rozdział 18: Inne rodzaje testów automatycznych** – w tym rozdziale omówiono dodatkowe rodzaje testów automatycznych, w tym testowanie wydajności i obciążenia, testowanie w środowisku produkcyjnym, testowanie wizualne, testy instalacji, testowanie z wykorzystaniem sztucznej inteligencji i więcej.
- **Rozdział 19: Co dalej?** – w tym rozdziale podano pewne wskazówek dotyczące dalszego zdobywania i rozwijania umiejętności w zakresie automatyzacji testów.

Poza tymi rozdziałami, na końcu książki dostępne są również cztery dodatki:

- **Dodatek A: Rzeczywiste przykłady** – ten dodatek stanowi uzupełnienie rozdziału 6 („Automatyzacja i architektura testów") i zawiera cztery rzeczywiste przykłady architektur aplikacji oraz odpowiadające im rozwiązania automatyzacji.
- **Dodatek B: Mechanizm oczyszczania** – ten dodatek zawiera opis sposób budowy mechanizmu oczyszczania, przedstawionego w rozdziale 7 („Izolacja i środowiska testowe").
- **Dodatek C: Projekt „Test Automation Essentials"** – w tym dodatku opisałem stworzony przeze mnie projekt open source o nazwie Test Automation Essentials, zawierający wiele przydatnych narzędzi kodu (napisanych w C#) dla projektów automatyzacji testów.
- **Dodatek D: Wskazówki i praktyki zwiększające produktywność programisty** – ten dodatek stanowi uzupełnienie dla rozdziałów od 9 do 14 i zawiera wskazówki, które pozwolą nam zwiększyć produktywność podczas programowania. Choć wskazówki te mogą być przydatne dla dowolnego programisty, będą one szczególnie użyteczne dla programistów automatyzacji testów.

Przyjemnej lektury!