

Jak skutecznie wdrożyć automatyzację testów funkcjonalnych

Automatyczne testy funkcjonalne optymalizują jakość oprogramowania i budują wartość biznesową

Streszczenie

Obecnie firmy z rynku IT stają przed potrójnym wyzwaniem. Muszą optymalizować jakość aplikacji o wciąż rosnącym poziomie złożoności – do tego jeszcze szybciej i taniej niż kiedykolwiek przedtem – aby dostarczać konkurencyjne rozwiązania z godziwym zyskiem. Firma Borland wykazuje praktyczne zrozumienie wyzwań stojących przed dzisiejszym rynkiem rozwoju oprogramowania, oferując rozwiązania wspierające cały cykl dostarczania aplikacji dzięki dostosowanym do specyfiki branży narzędziom.

Większość z firm odnoszących sukcesy w obszarze dostarczania rozwiązań software'owych odpowiedziało na to wyzwanie: „Odpowiedzią jest automatyzacja!” Po raz kolejny czołowi przedstawiciele biznesu IT z sukcesem pokazali skuteczność automatyzacji testów funkcjonalnych, drastycznie zwiększając szybkość produkcji oprogramowania przy jednoczesnej redukcji kosztów dostarczania. Wciąż jednak 85% organizacji ma kłopoty z wdrożeniem automatyzacji¹. Tak wysoki odsetek niepowodzeń wynika głównie z postrzegania automatyzacji jedynie jako procesu wdrożenia wybranego rozwiązania. Powyższe podejście charakteryzuje się oczekiwaniem natychmiastowego rezultatu w zmniejszeniu kosztów oraz przyspieszeniu produkcji w zasadzie jednocześnie z zakupem dowolnego narzędzia.

¹ International Institute for Software Testing

Czym jest testowanie funkcjonalne?

Testy funkcjonalne, lub inaczej testy black-boxa, to integralna część procesu Zapewniania Jakości stosowana w celu zweryfikowania czy obszar funkcjonalności oferowanej przez aplikację użytkownikowi działa zgodnie z przyjętymi wcześniej założeniami. Testowanie funkcjonalne może obejmować zarówno testy manualne i automatyczne. W każdym przypadku jest to ciąg testów emulujących interakcje pomiędzy użytkownikiem a aplikacją w celu zweryfikowania czy dana aplikacja robi to, do czego została zaprojektowana.

Po co automatyzować testy funkcjonalne?

Testowanie manualne jest dobrą praktyką w niektórych przypadkach. Wymaga jednak czasu by przejść przez żmudny proces weryfikacji funkcjonalności aplikacji, co dziś często nie idzie w parze z coraz krótszymi cyklami wytwarzania oprogramowania. W rezultacie możemy mieć do czynienia z powstrzymywaniem procesu produkcji oraz zbyt pobieżnym testowaniem aplikacji, co skutkować może przeniknięciem usterek do wersji finalnej. W dodatku, jeśli aplikacja ma działać poprawnie na różnych platformach, wysiłek testowania manualnego rośnie do potęgi wynikającej z liczby wspieranych środowisk. W dodatku testy ręczne są bardziej podatne na ludzkie błędy i niekonsekwencje, które mogą zaważyć na wyniku testów.

Testy automatyczne pozwalają skutecznie przyspieszyć cykl testowania i polepszyć jakość dostarczanego oprogramowania. Dzięki objęciu automatyzacją testów regresyjnych, działy jakości są w stanie osiągać lepiej widoczne wyniki. Oznacza to zwiększenie stopnia pokrycia testami dzięki rozszerzeniu automatyzacji o te części aplikacji, które mogły nie być uprzednio dokładnie testowane w poprzednich wersjach.

Automatyzacja testów dodatkowo optymalizuje jakość oprogramowania i efektywność procesu testowania poprzez dostarczanie poniższych długofalowych rezultatów:

Możliwość ponownego wykorzystania – Automatyzacja testów ułatwia działom jakości lepsze dopasowanie do napiętych harmonogramów prac dzięki wielokrotnemu wykorzystaniu raz zdefiniowanych testów, jako alternatywę dla każdorazowego tworzenia nowych skryptów. Doświadczenie wykazuje, że testy wielokrotnego użycia są uruchamiane częściej, umożliwiając testerom odnalezienie i usunięcie większej liczby usterek na wczesnych etapach projektu. Możliwość ponownego wykorzystania testów umożliwia również działom jakości tworzenie całych bibliotek skryptów testowych gotowych do wykorzystania – a w efekcie, na przetworzenie każdego testu w długoterminową wartość intelektualną.

Przewidywalność i metodyczność – Stosując automatyzację działy QA mogą przeprowadzać testy w sposób bardziej konsekwentny i usystematyzowany. Staje się to szczególnie istotne, gdy dział programistów dostarcza nowy build. Realizując testy regresyjne dział jakości może szybko zweryfikować, czy wszystkie dotychczasowe funkcjonalności działają wciąż poprawnie w najnowszej wersji i szybko przekazywać rezultaty z powrotem. Konsekwentność ma również wpływ na sam proces testowania. Powtarzając te same czynności testowe QA może

sprawniej odtwarzać i weryfikować usterki, tym samym przyspieszając cały proces wytwarzania.

Produktywność – Automatyzacja testów pozwala na stworzenie wysokowydajnego środowiska pracy, dzięki któremu organizacja może zwiększyć efektywność testowania bez konieczności wykorzystania nowych zasobów. Przykładowo, stosując automatyzację firma może prowadzić nie nadzorowany proces testowania 24 godziny na dobę testując aplikacje na różnych platformach, pod różnymi przeglądarkami i środowiskami jednocześnie. Umożliwia to pracownikom skupić się na innych działaniach jakościowych. Uzyskiwany w ten sposób zysk produktywności skutkuje w dwójnasób: skróceniem cyklu testowego oraz zwiększeniem możliwości zoptymalizowania jakości oprogramowania.

Podsumowując, wymienione zalety automatyzacji pozwalają działom jakości osiągać coraz lepsze rezultaty jakościowe, podejmować szybkie i trafne decyzje oraz minimalizować ryzyko.

Jak zautomatyzować proces testowania funkcjonalnego?

1. Ustalenie, testowanie których aplikacji powinno podlegać automatyzacji

Najlepszym kandydatem do zastosowania automatyzacji testowania jest oprogramowanie wymagające tworzenia wielu wersji przez cały cykl jego życia, przykładowo ze względu na wciąż rozszerzaną lub zmienianą funkcjonalność. Aplikacje, które muszą produkować stały zestaw wyników bazując na relatywnie stałych danych również są świetnym przykładem. Tego typu oprogramowanie w pełni pozwoli wykorzystać korzyści wynikające z przewidywalności oraz wielokrotnego użycia automatycznych testów funkcjonalnych.

2. Wybranie właściwego podejścia testowego

Istnieje kilka metodologii automatyzacji testów funkcjonalnych.

Testy modularne – to podejście dzielenia testowanej aplikacji na skrypty komponentowe lub moduły. Używając języka skryptowego tester buduje abstrakcyjną warstwę dla każdego komponentu – w efekcie, ukrywając go przed resztą aplikacji. Podejście to ułatwia zarządzanie i skalowanie zestawów testowych dzięki ochronie pozostałej części aplikacji przed efektem zmian w obrębie pojedynczego modułu

Biblioteka testów – to kolejne podejście skryptowe dzielące aplikację na moduły używane do budowania testów. W przeciwieństwie do podejścia modularnego, biblioteka testów opisuje moduły jako procedury i funkcje, a nie jako skrypty – umożliwiając jeszcze większą modularność, zarządzanie i wielokrotne wykorzystanie.

Testowanie po słowach kluczowych – zwane również testowaniem tablicowym, jest podejściem niezależnym od aplikacji, bazującym na danych w tabelach oraz łatwych do zrozumienia „słowach kluczowych”, opisujących czynności testowe dla aplikacji. Dane

tablicowe i słowa kluczowe są niezależne od rozwiązania automatyzującego testy oraz samych skryptów testowych. Testowanie po słowach kluczowych umożliwia mniej technicznemu personelowi tworzenie testów dzięki używaniu znanych terminów zamiast tworzenia skryptu czy programowania. To podejście wymaga uprzedniego zdefiniowania odpowiednich słów kluczowych w postaci tabel i słownika, podobnie jak samych modułów czy komponentów skryptowych określonych przez słowa kluczowe. Pomimo czasu jaki jest wymagany do przygotowania, testowanie w oparciu o słowa kluczowe sprawdziło się jako bardzo wydajne i efektywne rozwiązanie.

Testowanie według danych – to podejście polegające na przechowywaniu danych w osobnym pliku (np. arkuszu kalkulacyjnym) zamiast na kodowaniu ich bezpośrednio w skrypcie. Dzięki temu pojedynczy skrypt może przetestować wszystkie wymagane przez nas wartości danych. Ponieważ korzystamy w tym przypadku z zewnętrznego dla skryptu źródła danych, staje się on odporny na zmiany w tychże. Modyfikacja wartości wymaga od nas jedynie wykonania zmian w pliku przechowującym nasze dane, podczas gdy skrypt bazowy pozostaje ten sam. Takie podejście zwiększa produktywność inżynierów testowych dzięki uproszczeniu zarządzania skryptami oraz umożliwieniu wysokopoziomowego wielokrotnego użycia skryptów bazowych. Opisywana metoda często jest stosowana w połączeniu z wyżej wymienionymi podejściami.

Nagrywanie/odtworzenie testów stanowi najwygodniejszą metodę tworzenia skryptów testowych. Wykonuje się je przez mechanizm nagrywania interakcji użytkownika aplikacji. Wynikiem nagrania jest wygenerowanie automatycznego skryptu testowego, który może być następnie uruchomiony i odtworzony w dowolnym momencie. Pomimo iż to podejście jest najłatwiejszym ze wszystkich opisanych, stworzone w ten sposób testy są raczej dość sztywne i rzadko nadają się do wielokrotnego użycia. Poważniejsze zmiany w aplikacji przeważnie będą wymagały ponownego nagrania kroków testu, co z kolei nie przyczyni się do przyspieszenia procesu testowania. Największą korzyścią ze stosowania podejścia nagraj/odtwórz jest niski koszt, łatwość użycia w większości aplikacji z interfejsem użytkownika oraz szybkość przygotowania testu.

3. Stworzenie planu testowania aplikacji

Plan testowania aplikacji jest dokumentem opisującym zakres, podejście, zasoby oraz czas wszystkich automatycznych i manualnych czynności zaangażowanych w proces testowania danej aplikacji. W szczególności plan testu obejmuje wszystkie funkcjonalności aplikacji, które mają zostać przetestowane, zadania testowe, osoby odpowiedzialne za poszczególne z nich, środowisko testowe, wybrane podejście, platformy oraz środowiska systemowe objęte działaniem aplikacji jak również metryki raportowe.

4. Stworzenie i wdrożenie testów automatycznych

Plan testowania aplikacji stanowi dokładną mapę dla tworzonych testów automatycznych. Bazując na nim, testy działu jakości tworzą pojedyncze testy korzystając z obecnego w firmie rozwiązania automatyzującego w oparciu o wybrane podejście (np. testy modułowe, testy

oparte na słowach kluczowych, testy oparte na danych czy metodę nagraj/odtwórz). Kiedy wszystkie testy zostaną połączone z poszczególnymi wymaganiami, zdefiniowane i stworzone, automatyzacja jest gotowa by zacząć dla nas pracować.

Przydatne wskazówki w podejściu do automatyzacji

Zawsze traktuj projekt automatyzacji testów jak proces tworzenia oprogramowania

Tak jak w przypadku wytwarzania oprogramowania, proces automatyzacji testów musi być odpowiednio zaprojektowany, przygotowany, udokumentowany oraz podlegać przeglądom. Doświadczenie jasno pokazuje, że wszelkie odstępstwa od powyższych standardów grożą porażką projektu.

Firmy mogące pochwalić się sukcesem w dziedzinie automatyzacji patrzą na zagadnienia jakości wskroś całego procesu wytwarzania oprogramowania i stosują sprawdzone praktyki podczas wdrażania automatycznych testów funkcjonalnych. Przykładem niech będzie przesuwanie testowania do bardzo wczesnych etapów projektu – nawet do momentu definiowania i zbierania wymagań. W takim przypadku świetnie sprawdza się stosowanie modularyzacji kodu. Dodatkowo tworzone są wówczas zasoby testowe, które z powodzeniem można wykorzystywać ponownie w późniejszych fazach projektu oraz podczas pracy nad kolejnymi wersjami oprogramowania.

Wdrożenie automatyzacji testów funkcjonalnych nie jest krótkoterminowym projektem. Ci, którym się to udało, doskonale wiedzą, że automatyzacja testowania jest pełnoetatowym zadaniem wymagającym na początku znacznej inwestycji czasu oraz zasobów. Jednocześnie należy mieć na uwadze długofalowe cele i oczekiwania kierownictwa. Kolejnym aspektem, o którym należy pamiętać, jest fakt, że automatyczne testy nie zastąpią w pełni człowieka. Wskazany wydaje się zatem odpowiednie szkolenie i budowanie kompetencji przy wykorzystaniu najlepszych praktyk w metodologiach automatyzacji procesu testowania.

Optymalizacja jakości to wspólny wysiłek

Poziom jakości oprogramowania nie zależy wyłącznie od działu QA - to najczęściej wynik działań i kultury pracy całego przedsiębiorstwa. Dzięki stworzeniu odpowiedniego środowiska współpracy wszystkich członków projektu oraz działów biznesowych, ze szczególnym uwzględnieniem programistów, dział jakości jest w stanie zoptymalizować jakość dostarczanych produktów upewniając się, że spełniają one wszystkie biznesowe wymagania.

Jednym ze sposobów osiągnięcia takiego poziomu współpracy jest stworzenie stanowiska Architekta Automatyzacji. Taka osoba współpracuje bezpośrednio z odbiorcami końcowymi, aby uzyskać pełny obraz funkcjonalności przyszłej aplikacji w oczach użytkowników. Na bazie tej wiedzy Architekt Automatyzacji może następnie zaprojektować cały proces automatyzacji testów pod kątem oczekiwań użytkowników i wymagań biznesowych.

Kolejną metodą na poprawę wewnętrznej współpracy jest wybór właściwego dostawcy odpowiedniego rozwiązania wspierającego automatyzację pod kątem silniejszego współdziałania zespołów w całym procesie dostarczania oprogramowania. Rozwiązania tej klasy skutecznie potrafią „natchnąć” do większej kooperacji w zakresie zapewnienia jakości, od pierwszych faz projektowych aż po produkcję i wdrożenie. Jednocześnie zapewniają kierownictwu pełen wgląd w działania QA umożliwiając lepszą ocenę znaczenia działań jakości w procesie produkcji.

Zarządzanie stale zmieniającymi się aplikacjami

Zmiany są nieuniknione – pojawiają się też w oprogramowaniu wraz z rozwojem biznesu, wymaganiami użytkowników oraz postępem technologii. W praktyce zarządzanie aplikacją i jej utrzymanie to nakład ponad 70% wszystkich kosztów wytwarzania².

Firmy mogące pochwalić się sukcesami na polu utrzymania oprogramowania zmniejszają te wydatki dzięki „wbudowaniu” elementów zarządzania w automatyzację testów. Przykładowo, stosuje się modularyzację testów przez rozdzielanie i izolowanie funkcjonalności aplikacji. W przypadku wystąpienia zmiany po stronie funkcjonalnej, zachodzi potrzeba modyfikacji jedynie wybranego komponentu lub komponentów zamiast całościowej edycji skryptu testowego. Dzięki temu możliwe są szybkie i łatwe zmiany testów odzwierciedlające zmiany w aplikacji.

Łatwość zarządzania i możliwość ponownego użycia to dwa podstawowe obszary, w których automatyzacja testów ma wpływ na zwrot z poniesionych inwestycji (ROI). Najlepszym podejściem do otrzymania tych atrybutów w automatyzacji jest jej zaadaptowanie przy użyciu architektury zorientowanej obiektowo. Pozwala ona bowiem na wykorzystanie wybranej metodologii (np. testów modularnych, testów opartych na słowach kluczowych czy nagraj/odtwórz). A co najistotniejsze, podejście obiektowe zapewnia najbardziej odczuwalną redukcję czasu i zaangażowanych w zarządzanie aplikacją zasobów.

Na co zwrócić uwagę przy wyborze narzędzia do automatyzacji?

Zintegrowane środowisko do automatyzacji testów – Zapewnia maksimum wydajności przy coraz krótszym cyklu produkcyjnym. Działy jakości potrzebują automatycznego rozwiązania pozwalającego na budowanie modułowych i gotowych do wielokrotnego użycia zasobów testowych. Dzięki przyspieszeniu procesu testowania przez ponowne wykorzystanie skryptów, możliwe staje się lepsze zoptymalizowanie jakości oprogramowania przez zwiększenie przewidywalności poszczególnych testów.

Lepsza adaptacja do ciągłych zmian wymagań biznesowych – Nowe wymagania ze strony biznesu mogą pojawiać się niemal z dnia na dzień, tworząc potrzebę natychmiastowego odzwierciedlenia ich po stronie aplikacji. Ale czy równie szybko możemy aktualizować związane z nimi przypadki testowe? Odpowiedzią jest odpowiednio zautomatyzowane

² Guelph Natural Computational Research Group, University of Guelph

rozwiązanie minimalizujące nakłady pracy związanej z zarządzaniem życiem oprogramowania. Wbudowane możliwości wielokrotnego wykorzystania skryptów i wygodne zarządzanie zorientowanym obiektowo rozwiązaniem zapewni możliwość szybkiej reakcji na potencjalne zmiany z rozszerzeniem na nowe technologie włącznie.

Wsparcie wielu technologii – Nowoczesne rozwiązania muszą wspierać użytkownika w szerokim zakresie różnorodnych platform, środowisk i dostępnych technologii. Należy unikać sytuacji, kiedy potrzebne będzie inwestowanie w dodatkowe rozwiązania do automatyzacji testów, w tym celu dobrze jest zwrócić szczególną uwagę na wspierane przez dane rozwiązanie środowiska i konfiguracje.

Optymalizacja zapewnienia jakości w całym cyklu tworzenia oprogramowania – Pomimo iż testy funkcjonalne mogą początkowo stanowić priorytet, zapewnienie jakości nie zaczyna się i nie kończy w ich ramach. Docelowo zakresem działań QA powinny być objęte pozostałe obszary, jak testowanie jednostkowe, wydajnościowe i obciążeniowe, zarządzanie testami i wydajnością aplikacji jeszcze w fazie produkcji.

Rozwinięta społeczność użytkowników – W dzisiejszym napędzanym przez rozwój technologii świecie biznesu, oprogramowanie korporacyjne może przyczynić się do sukcesu bądź porażki firmy. Kluczowym staje się zatem zmniejszenie ryzyka dysfunkcji aplikacji przez wybranie sprawdzonego rozwiązania, popartego sukcesami szerokiej liczby obecnych użytkowników.

Odpowiedni poziom wsparcia i usług dodatkowych – Wdrożenie automatyzacji testów wymaga początkowo dodatkowego czasu i zasobów – czynników dość ograniczonych w dzisiejszych procesach wytwórczych. Istotne zatem staje się wybranie rozwiązania oferowanego przez dostawcę wraz z dodatkowymi usługami wdrożeniowymi, szkoleniami, konsultacjami oraz pełnym wsparciem technicznym w celu bezproblemowej implementacji i uruchomienia nowych procesów w firmie.

Borland[®] SilkTest[®] – rozwiązanie do pełnej automatyzacji testów funkcjonalnych

Borland[®] SilkTest[®] jest wiodącym w branży, wielokrotnie nagradzonym rozwiązaniem do kompletnej automatyzacji wszystkich testów funkcjonalnych aplikacji stworzonych w dowolnej technologii jak .NET, Web, Java[™] czy w architekturze klient-serwer. SilkTest umożliwia firmom poprawną weryfikację, czy biznesowe aplikacje w pełni pokrywają się ze stawianymi im wymaganiami oraz oczekiwaniami. Działając w oparciu o najświeższe zdobycze technologii testowania SilkTest pozwala na zwiększenie dokładności, poprawności oraz zmniejszenie nakładów pracy w odniesieniu do uzyskiwanych wyników. SilkTest wspiera również wszystkie główne metodologie testowe, w tym modularność, testy biblioteczne, testy w oparciu o słowa kluczowe, testy w oparciu o dane czy nagraj/odtwórz. Dodatkowo dzięki wykorzystaniu zalet Unicode[®] SilkTest zapewnia wsparcie dla aplikacji wielojęzycznych.

Zaprojektowany w celu zmaksymalizowania prostoty użycia przy najwyższej wydajności, SilkTest zapewnia rekordową szybkość i produktywność procesu testowania. Przykładowo, użytkownicy mogą testować aplikacje wieloplatformowe, w technologii cienkiego i grubego klienta, w wielu środowiskach używając pojedynczego skryptu polegając jedynie na niezwykle elastycznym, wydajnym i stworzonym specjalnie na potrzeby testerów zorientowanym obiektowo języku skryptowym Borland® 4Test®. Co więcej, wbudowany w produkt SilkTest konfigurowalny recovery system pozwala prowadzić automatyczne testy bez obsługi 24 godziny na dobę, 7 dni w tygodniu. Unikając konieczności powtarzania wciąż tych samych zadań, dział jakości może objąć testami więcej elementów i wyłapać więcej usterek w procesie produkcji oprogramowania, tym samym optymalizując jakość dostarczanych aplikacji.

W zakresie weryfikacji funkcjonalności w wersjach roboczych, użytkownicy rozwiązania SilkTest mogą testować wpływ nowych elementów na dotychczasowe funkcjonalności dzięki ponownemu wykorzystaniu dotychczasowych przypadków testowych. Gotowe do wielokrotnego użycia zasoby testowe minimalizują wkład pracy niezbędny do zarządzania jakością aplikacji, pozwalając użytkownikom dokonywać jedynie małych modyfikacji wcześniej stworzonych skryptów.

Architektura produktowa rozwiązania SilkTest pozwala organizacjom na efektywne kosztowo zarządzanie złożonymi testami nawet bardzo rozbudowanych aplikacji. Wykorzystując technologię rozproszonych agentów SilkTest pozwala zarządzać wszystkimi testami razem oraz osobno wykorzystując biznesowe scenariusze wykorzystania oprogramowania. Dodatkowo, moduł SilkTest Runtime, będący okrojona wersją narzędzia SilkTest, pozwala działom IT na dystrybucję testów na różne platformy i środowiska, zwiększając produktywność i minimalizując koszty związane z obsługą testów.

Zarządzanie jakością w cyklu życia aplikacji

Rozwiązanie Borland Lifecycle Quality Management (LQM) umożliwia przedsiębiorstwom produkcję wysokiej jakości oprogramowania, redukując ryzyko oraz zwiększając zwrot z inwestycji. Wiodące firmy z całego świata, wliczając wiele wyszczególnionych w ramach listy Fortune 500, polegają na innowacyjności produktów z rodziny Borland Silk, chroniąc poziom swojego biznesu, budując przewagę konkurencyjną oraz reputację własnej marki. A ponieważ jakość to nie tylko wdrożenie odpowiedniego narzędzia, firma Borland oferuje pełen zakres usług konsultingowych, szkoleniowych i wspierających oferowane produkty – wszystko by zapewnić swoim klientom sukces wprowadzenia automatyzacji testów funkcjonalnych.

Dodatkowo, jako rozwiązania współpracujące z narzędziem SilkTest, Borland oferuje:

Borland® SilkCentral® Test Manager do zarządzania całym procesem testowym

Borland® SilkPerformer® modelowe testy wydajnościowe i obciążeniowe

Borland® SilkPerformer® Component Test Edition do wczesnych testów zdalnych komponentów współdzielonego dostępu

Borland® SilkCentral® Performance Manager do monitorowania aplikacji z punktu widzenia użytkownika końcowego

Podsumowanie

Dzięki dobrej organizacji procesu testowania oraz automatyzacji testów funkcjonalnych firmy mogą zwiększyć jakość tworzonych aplikacji dzięki weryfikacji dokładności i pewności działania tworzonych aplikacji we wczesnych fazach produkcji. Testy automatyczne kreują dodatkową wartość w postaci zapewnienia kadry zarządzającej, że nawet złożone i rozbudowane aplikacje są dostarczane i wdrażane na czas i w założonym budżecie. Obecnie firmy na całym świecie czerpią powyższe korzyści używając rozwiązania SilkTest, nagradzanego produktu do automatyzacji testów funkcjonalnych firmy Borland. SilkTest oferuje powtarzalne, przewidywalne metody testowania funkcjonalności oprogramowania w ramach kolejnych powstających jego wersji, przez cały czas ich życia. W rezultacie SilkTest zapewnia szybkość i dokładność w całym procesie testowania optymalizując jakość dzięki przeniesieniu działań QA ze żmudnych zadań testowania manualnego do obszaru zarządzania pokryciem testami i proaktywnego wyłaniania usterek i rozbieżności już we wczesnych fazach procesu wytwarzania oprogramowania.

Borland is the leading vendor of Open Application Lifecycle Management (ALM) solutions - open to customers' processes, tools and platforms – providing the flexibility to manage, measure and improve the software delivery process.



Copyright © 2007 Borland Software Corporation. Borland and all other brand and product names are service marks, trademarks, or registered trademarks of Borland Software Corporation or its subsidiaries in the United States and other countries. All other marks are properties of their respective owners. 24479
www.borland.pl