

Temat: „Automatyzacja testów oprogramowania dla urządzeń mobilnych”

Urządzenia mobilne z roku na rok stają się coraz powszechniejsze. Telefony, GPS-y, tablety i wiele innych można spotkać niemal wszędzie, na każdym rogu ulicy miasta, w którym żyjemy. Wzrost popularności technologii mobilnej zaowocował wzrostem zapotrzebowania na różnego rodzaju aplikacje zarówno do obsługi samych sprzętów, jak również na codzienne potrzeby ich posiadaczy.

Początkowo każde urządzenie posiadało swój własny, dedykowany system operacyjny, projektowany i wytwarzany tylko i wyłącznie dla jednej rodziny produktu. Również oprogramowanie tworzone było w sposób ukierunkowany na konkretne rozwiązania. Wraz z postępem opracowano nowe podejście, które oparło się na stworzeniu miniaturowego systemu operacyjnego, który podobnie do odpowiedników z PC-ów, będzie w stanie obsłużyć wiele standardów sprzętowych. Producenci mobilnych systemów operacyjnych dostarczają razem z nimi narzędzia developerskie. Dzięki temu niezależni programiści mogą dostarczać dowolne narzędzia na daną platformę. Jednak również prowadzi to do pewnego problemu, który wiąże się z jakością rozwijanych programów. Jak testować? Czy testować?

Czym różni się testowanie mobilnego oprogramowania od zwykłego?

Pytanie postawione przed chwilą wydaje się być banalne, dla każdego kto miał choć trochę styczności z testowaniem. W końcu zadaniem dla testera jest sprawdzenie czy dany produkt nie posiada błędów i spełnia swoją rolę. Jednak po głębszym zastanowieniu można natrafić na znaczącą grupę różnic, dzielącą podejście do testowania aplikacji z komputerów lub internetu od aplikacji mobilnych. Podstawowych przyczyn można dopatrywać się w samej charakterystyce systemów przenośnych:

- Opierają się na zdarzeniach – konkretnych akcjach podejmowanych przez użytkownika
- Stanowią zamknięte środowiska – bez wsparcia licznych producentów sterowników, peryferii lub alternatywnych rozwiązań
- Muszą spełniać wyższe standardy jakości – wynika to z mentalności i podejścia użytkowników końcowych, chcących cieszyć się produktem, który w każdych okolicznościach spełni swoje zadanie
- Obsługują wiele zdarzeń i protokołów jednocześnie – np. smartfon oprócz monitorowania usług telekomunikacyjnych, interakcji z userem za pomocą widgetów, musi także obsługiwać procesy wykonujące się w tle, utrzymując przy tym wysoką wydajność na znacznie bardziej ograniczonych zasobach

Oczywiście nie są to wszystkie przyczyny. Jednak opierając się na nich można przyjrzeć się bliżej poszczególnym typom testów. W typowym cyklu wytwarzania oprogramowania pierwsza faza opiera się na sporządzeniu wymagań i założeń dotyczących programu. Następnie rozdzielane są role w zespole developerskim, tak by w zależności od stosowanego modelu, w określonym momencie otrzymać funkcjonujący komponent. W całym procesie wykorzystywane jest wsparcie testerów w celu zapewnienia jak najwyższej jakości i niezawodności końcowej aplikacji. W przypadku urządzeń mobilnych poza przeglądami formalnymi tworzonego kodu źródłowego, we wczesnych fazach projektu wsparcie zespołu jakościowego ogranicza się do sprawdzania kolejnych działających release'ów, zdolnych do funkcjonowania na sprzęcie docelowym.

Można śmiało stwierdzić, że testowanie oprogramowania mobilnego skupia się na czterech podstawowych obszarach:

- Funkcjonalności
- Interfejsów
- Wydajności
- Stabilności

Zakres pierwszy jest najbardziej powszechnym rodzajem testów bez względu na rodzaj środowiska i przeznaczenie. W rozpatrywanym przypadku funkcjonalności są ściśle zależne od akcji podejmowanych przez

użytkownika. Muszą one przejść weryfikację pod względem zgodności z różnymi elementami składowymi urządzeń. W przypadku telefonów (popularnych smartfonów) analizowane są różne wielkości ekranów, klawiatury itd. Dodatkowo funkcjonalność musi być jasna, klarowna, nastawiona na realizowanie przyjętego celu w najprostszy możliwy sposób. Surowość kryteriów akceptacji wynika z mentalności użytkownika końcowego. Od urządzeń mobilnych wymaga się niezawodności, realizowania zadań w każdych warunkach, wysokiego standardu wykonania i możliwości prostej konfiguracji.

Ze względu na wysoko zawieszoną poprzeczkę w stosunku do funkcjonalności badanego systemu, równie kluczowymi elementami stają się interfejsy. Wysoki nacisk jest przykładany do przejrzystości i ergonomiczności. Bez poprawnie zdefiniowanych elementów, uwzględniających różnorodność dostępnych urządzeń na rynku oraz odzwierciedlających stany i podejmowane akcje w programie, żadna aplikacja nie ma szansy zaistnieć. Dodatkowe ograniczenie stanowi ilość dostępnych zasobów, czyniąc testy wydajności i stabilności niezwykle istotnymi.

Automatyzacja testów

Automatyzacja testów aplikacji na urządzenia mobilne jest zagadnieniem stosunkowo nowym, ze względu na dynamiczny rozkwit samej technologii mobilnej w ostatnim czasie. Z całą pewnością każdy z opisanych we wcześniejszym akapicie obszarów można zautomatyzować. Podstawowe pytanie „Jak tego dokonać?” pojawia się, gdy stajemy przed wyborem metody. W zależności od dostępnych zasobów, czasu oraz możliwości można obrać jedną ze ścieżek:

- Symulowanie procesów – na potrzeby automatyzacji można wykorzystać wirtualne środowisko, które w większości wypadków dostępne jest razem ze środowiskiem developerskim. Daje to możliwość przeprowadzenia testów zarówno systemowych jak i funkcjonalnych na aplikacji uruchamianej bez urządzenia mobilnego. Rozwiązanie zaoszczędza czas oraz redukuje koszty projektu, jednak niesie ze sobą ryzyko związane z różnicami pomiędzy emulowanym systemem, a tym realnie instalowanym i operującym na konkretnych rozwiązaniach sprzętowych.
- Wykorzystanie dostępnych urządzeń – często zdarzają się przypadki, gdzie firma na potrzeby projektu otrzymuje urządzenia od zleceniodawcy lub posiada własne zaplecze sprzętowe. Pozwala to skonfrontować produkt ze światem realnym. Podejście jest kosztowe, ponieważ baza dostępnych urządzeń musi być stale aktualizowana. Dodatkowo ograniczona jest ilość dostępnych urządzeń, na których można jednocześnie pracować i testować, co może istotnie wpłynąć na czas przejścia przez wszystkie etapy cyklu wytwarzania.
- Wynajęcie bazy urządzeń – na potrzeby realizacji zadań projektu można pokusić się o wynajęcie bazy urządzeń. Wśród firm oferujących swoją pomoc, zajmujących się kolekcjonowaniem wszystkich możliwych urządzeń i wypożyczaniem ich do dowolnego użytku, znajdują się też takie, które dają możliwość podłączenia zdalnego konkretnej grupy produktów.
- Wykorzystanie zewnętrznej firmy – na rynku międzynarodowym istnieją firmy wyspecjalizowane w testowaniu zarówno urządzeń mobilnych, jak i aplikacji tworzonych pod nie. Posiadają one odpowiednie zasoby do realizacji wszystkich powierzanych zadań. Wynajęcie zasobów w postaci testerów i urządzeń nie jest tanie, ale gwarantuje przeprowadzenie pełnego zestawu testów, bez konieczności uszczuplania własnych zasobów.

Odpowiednia strategia testów wymaga dokładanego przeanalizowania jakie zasoby będą niezbędne do ich wykonania. Wynajęcie bazy urządzeń lub wykorzystanie zewnętrznej firmy wiąże się z dodatkowymi kosztami, które niejednokrotnie przewyższają budżet projektu. Jeśli jednak posiadamy urządzenie, na którym chcemy sprawdzić działanie tworzonego programu, automatyzacja procesu będzie wymagała zdecydowanie większego

wkładu pracy. Wiąże się to z koniecznością podłączenia urządzenia do środowiska testowego, w taki sposób, aby móc wysyłać zapytania i odbierać reakcje urządzenia.

W tym celu można posłużyć się specjalnymi narzędziami, pozwalającymi na zdalnie sterowanie obiektem. Wykorzystując dedykowane dla danego systemu operacyjnego kontrolery oraz sterowniki producenta możemy z platformy PC stworzyć małe centrum zarządzania testami. Skrypt ze standardowych narzędzi automatyzacji, będzie miał możliwość wysłania kroku testowego do urządzenia mobilnego, po czym odebrania reakcji. Zazwyczaj operacja wykonywana jest za pomocą specjalnego interfejsu odzwierciedlający aktualny stan np. telefonu. Chcąc rozszerzyć zakres analizy trzeba wziąć pod uwagę konieczność indywidualnego konfigurowania środowiska testowego pod każdy nowy sprzęt. Jak tego uniknąć? Lub raczej:

Czy potrzebne jest urządzenie z zainstalowanym systemem operacyjnym do testowania aplikacji?

Otóż nie! Tak jak zostało już wcześniej wspomniane, każdy producent dostarcza razem ze środowiskiem odpowiedni emulator na platformę PC. Poniżej przedstawiono najpopularniejsze emulatory wraz z krótką charakterystyką:

1. Google Android

Wymagania:

Android SDK

System operacyjny:

Windows/Linux/Mac

Obsługiwane urządzenia:

wszystkie z zainstalowanym systemem w wersjach 1.1, 1.5, 1.6, 2.0, 2.1, 2.2, 2.3, 3.0,3.2

Opis:

Stworzony w oparciu o emulator QEMU, symulujący architekturę ARM. Pozwala na uruchomienie całego systemu Android.



2. Apple iOS

Wymagania: *iOS SDK*

System operacyjny: *Mac*

Obsługiwane urządzenia:

iPhone 3GS, iPod Touch, iPhone 4, iPad

Opis:

Pozwala na tworzenie i uruchamianie aplikacji na urządzenia z dwóch rodzin: iPhone i iPad



3. Microsoft Windows Phone

Wymagania:

Windows Phone SDK

System operacyjny:

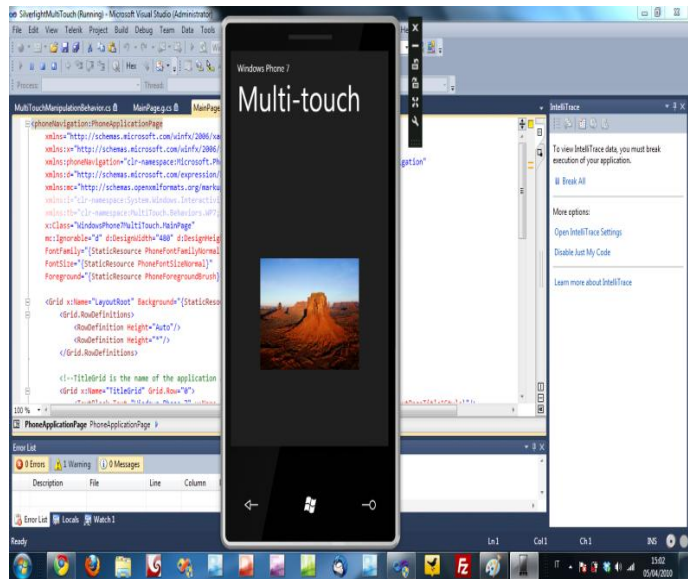
Windows (z wyłączeniem Windows XP, Server oraz maszyn wirtualnych)

Obsługiwane urządzenia:

Wszystkie pod kontrolą Windows 7

Opis:

Pozwala na tworzenie i uruchamianie aplikacji na urządzeniu z zainstalowanym Windows Phone 7. Domyślnie zintegrowane środowisko z Visual Studio.



Korzystając z emulatorów można podejść do automatyzacji testów w dokładnie taki sam sposób, jak do automatyzacji w przypadku standardowych aplikacji na platformy PC. Posiadamy konkretny program, który ma reagować w określony sposób. Akcje użytkownika są przekazywane za pomocą standardowych akcji systemu (kliknięcia myszki, przerwania z klawiatury). Reakcje badanego środowiska można dokumentować za pomocą screen shotów.

W tabeli poniżej ujęto wszystkie podstawowe zalety i wady podejścia opierającego się o pracę na emulatorach.

Zalety	Wady
<ul style="list-style-type: none"> • Oszczędność – nie trzeba posiadać urządzeń z danym systemem operacyjnym, zarówno do rozwoju aplikacji jak i testowania • Możliwość pracy nad aplikacjami na różne platformy • Projektowanie/testowanie aplikacji dla różnych modeli urządzeń – bez dodatkowych kosztów • Wierne oddanie zachowania systemu urządzenia mobilnego i interakcji z użytkownikiem 	<ul style="list-style-type: none"> • Czasami występują niezgodności zachowania aplikacji na urządzeniu i emulatorze • Aby mieć pewność poprawności działania aplikacji, finalne testy należy przeprowadzić na docelowym urządzeniu • Wolniejsze działanie niż rzeczywiste urządzenie • Nie wszystkie zasoby są dostępne (np. GPS, akcelerometri, GSM, kamera)