

Analiza możliwości skrócenia czasu tworzenia aplikacji mobilnej na systemy Android oraz iOS przy użyciu technologii Xamarin

Daniel Molenda*, Maria Skublewska-Paszowska

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Artykuł przedstawia porównanie czasu tworzenia aplikacji mobilnej na system Android i iOS wykonanej przy pomocy wieloplatformowej technologii Xamarin oraz wykonanej natywnie dla obu systemów osobno. Do badań wykorzystano autorską aplikację zaimplementowaną w trzech środowiskach programistycznych: VisualStudio (przy użyciu technologii Xamarin), Android Studio oraz XCode. Wyniki przeprowadzonych badań wykazały, że utworzenie aplikacji w technologii Xamarin trwa o około połowę mniej czasu.

Słowa kluczowe: aplikacja mobilna; aplikacja wieloplatformowa; Android; iOS; Xamarin

*Autor do korespondencji.

Adres e-mail: daniel.m.92@wp.pl

Analysis of the possibility of shortening the time of creating a mobile application for Android and iOS systems using Xamarin technology

Daniel Molenda*, Maria Skublewska-Paszowska

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. The article presents a comparison of times needed for creating a mobile application for Android and iOS system using multi-platform Xamarin technology and respective native technologies for both systems. The authors' application was implemented in three programming environments: VisualStudio (using Xamarin technology), Android Studio and XCode. The results of the conducted research have shown that the creation of an application in Xamarin technology takes approximately half the time.

Keywords: mobile application; cross-platform application; Android; iOS; Xamarin

*Corresponding author.

E-mail address: daniel.m.92@wp.pl

1. Wstęp

Optymalizacja czasu tworzenia oprogramowania jest niezwykle istotna nie tylko z punktu widzenia inwestorów, którzy jak najmniejszym kosztem chcą otrzymać gotowy produkt, ale także dla twórców tego oprogramowania. Bardzo dobrze jest to widoczne na przykładzie aplikacji mobilnych, gdzie występuje różnorodność systemów operacyjnych, na które są one przeznaczone. W tym przypadku producenci chcąc dotrzeć do jak największej grupy odbiorców, muszą liczyć się ze wzrostem kosztów jakie muszą ponieść, aby to osiągnąć. Natomiast twórcy tych aplikacji często stają przed zadaniem napisania tej samej funkcjonalności kilkukrotnie w różnych językach programowania. Z tego powodu powstają różnego rodzaju technologie, które pozwalają w mniejszym lub większym stopniu na „ujednoczenie” tego procesu.

Jedną z takich technologii jest Xamarin, który posiada wspólny interfejs do tworzenia aplikacji na najpopularniejsze systemy mobilne: Android oraz iOS [1]. Technologia ta została wykorzystana do zbadania w jakim stopniu przy jej użyciu można skrócić czas wytworzenia aplikacji mobilnej na te dwa systemy.

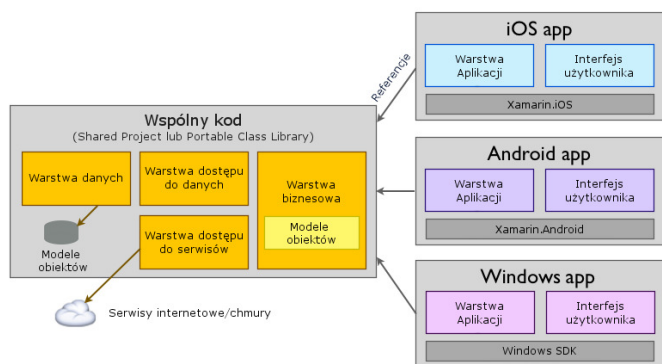
Istnieje wiele publikacji dotyczących porównań technologii Xamarin z tradycyjnym podejściem do budowania aplikacji mobilnych. Większość z nich porusza tematykę porównania wydajności aplikacji, jak np. pozycja [2], w której autor porównuje kilka w różnym stopniu skomplikowanych aplikacji – od „Hello World”, aż po zaawansowane funkcjonalnie aplikacje pobierające dane z baz SQL oraz łączące się z serwisami internetowymi. Często publikowane są artykuły przedstawiające wady i zalety użycia technologii wieloplatformowych [3]. Nigdzie natomiast nie odnaleziono publikacji dotyczących porównania czasu tworzenia aplikacji mobilnej przy użyciu natywnych narzędzi (Android Studio, XCode) oraz aplikacji utworzonej w wieloplatformowej technologii Xamarin (Visual Studio). Być może jest to spowodowane trudnością oszacowania czasochłonności zbudowania takiej aplikacji, chociażby ze względu na wykorzystanie różnych języków programowania (Java, Swift, C#) oraz na znaczne różnice w dostępnych narzędziach ułatwiających ten proces.

2. Charakterystyka technologii Xamarin

Xamarin jest ujednoczoną oraz uproszczoną platformą do tworzenia aplikacji mobilnych na systemy Android oraz iOS, która wykorzystuje język programowania C# oraz platformę

.NET do dostarczania uniwersalnych rozwiązań na oba te systemy operacyjne. Platforma ta stworzona została w 2011 roku przez twórców Mono (zestawu narzędzi umożliwiających uruchomienie aplikacji stworzonych dla platformy Microsoft.NET w wielu środowiskach – m.in. w systemie Linux oraz macOS [4]), w której skład wchodzi Xamarin.Android (formalnie Mono for Android) oraz Xamarin.iOS (formalnie MonoTouch). Początkowo była to platforma posiadająca swoje własne środowisko programistyczne – Xamarin Studio, jednak po wykupieniu jej przez firmę Microsoft, stała się integralną częścią rozbudowanego środowiska programistycznego Visual Studio, które pozwalało na utworzenie osobnych projektów dla każdej z obsługiwanych platform (Android, iOS). Każde z utworzonych rozwiązań posiadało dedykowane narzędzia do tworzenia m.in. interfejsu użytkownika. Poza tym programista miał możliwość wyboru metody udostępniania wspólnego kodu (w postaci dodatkowego projektu) dla tych platform (Rys. 1):

- 1) Shared Project – którego kompilacja odbywała się przy wykorzystaniu bibliotek aktualnie wybranej platformy (Android lub iOS);
- 2) Portable Class Libraries – projekt kompilowany niezależnie, zawierający wspólne biblioteki dla obu platform.



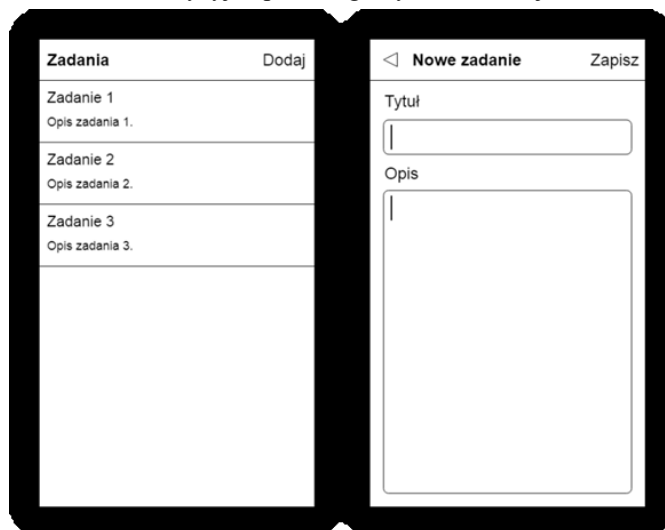
Rys. 1. Architektura udostępniania wspólnego kodu w platformie Xamarin

Dążąc do jeszcze większego ujednoczenia procesu tworzenia aplikacji mobilnych na wiele systemów operacyjnych, 28 maja 2014 roku twórcy platformy Xamarin opublikowali nowe narzędzie o nazwie Xamarin.Forms – dzięki któremu umożliwiono tworzenie jednego wspólnego interfejsu użytkownika dla wszystkich obsługiwanych platform. Jest to biblioteka składająca się z około 40 komponentów (układów oraz kontrolki), z których każdy ma swój natywny odpowiednik [5]. Początkowo możliwości tego narzędzia były ograniczone, jednak w miarę jego rozwoju twórcy zdecydowali się udostępnić możliwość dowolnego konfigurowania wyglądu oraz zachowania kontrolki na docelowych platformach za pomocą tzw. rendererów [6].

3. Aplikacja mobilna

W celu przeprowadzenia badań stworzona została aplikacja mobilna „Zadania”. Umożliwia ona tworzenie listy zadań, które określane są poprzez tytuł oraz opis. Aplikacja posiada opcje wyświetlania oraz dodawania zadań na liście. Składa się ona z dwóch okien (Rys. 2):

- 1) Lista zadań – okno składające się z przewijanej listy zadań
- 2) Utworzenie nowego zadania – okno zawierające kontrolki edycyjne poszczególnych informacji o zadaniu.



Rys. 2. Prototyp okien aplikacji „Zadania”. Po lewej – lista zadań, po prawej – tworzenie nowego zadania

Ten z pozoru prosty projekt realizuje wiele zagadnień związanych z funkcjonowaniem aplikacji mobilnych, takich jak: zarządzanie nawigacją pomiędzy oknami aplikacji oraz przesyłanie danych między nimi, oprogramowanie listy przewijanej wykorzystującej niestandardowy wygląd wierszy (spotykanej niemal w każdej aplikacji), czy wykorzystanie podstawowych kontrolki do wyświetlania oraz edycji tekstu.

Aplikacja zaimplementowana została na dwa systemy mobilne: Android oraz iOS w trzech różnych środowiskach programistycznych:

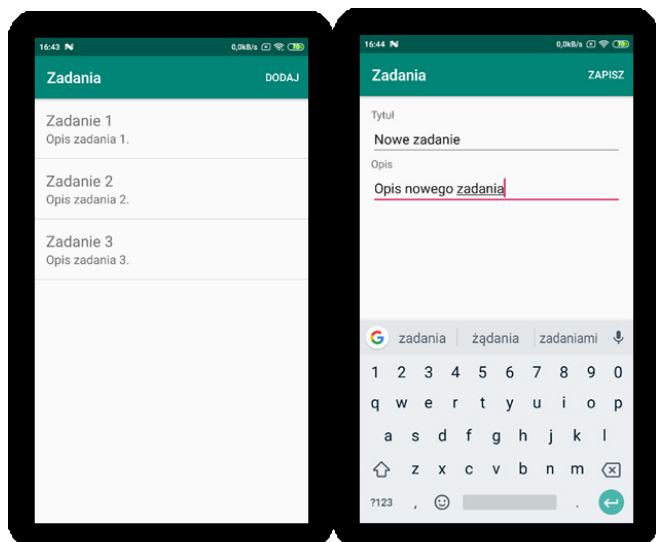
- 1) Android Studio przy użyciu języka programowania Java – system Android (Rys. 3);
- 2) XCode przy użyciu języka Swift 4 – system iOS (Rys. 4);
- 3) Visual Studio przy użyciu języka C# oraz technologii Xamarin. – wieloplatformowa aplikacja na systemy Android (Rys. 5) oraz iOS (Rys. 6).

Podczas badań porównywane będą aplikacje napisane natywnie – w środowiskach przeznaczonych wyłącznie do tworzenia aplikacji na jeden system (Android Studio, XCode) – oraz aplikacja napisana na obie platformy jednocześnie (Xamarin).

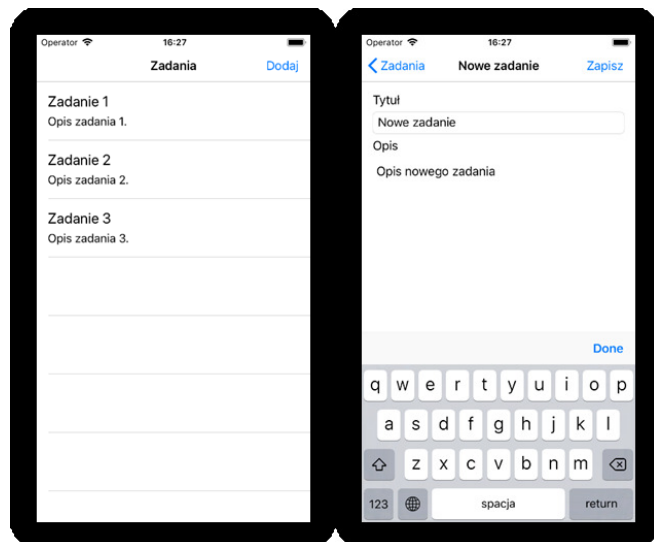
4. Przeprowadzone badania

Badania zostały podzielone na dwa etapy. Pierwszym z nich było porównanie czasów kompilacji aplikacji dla trzech wariantów:

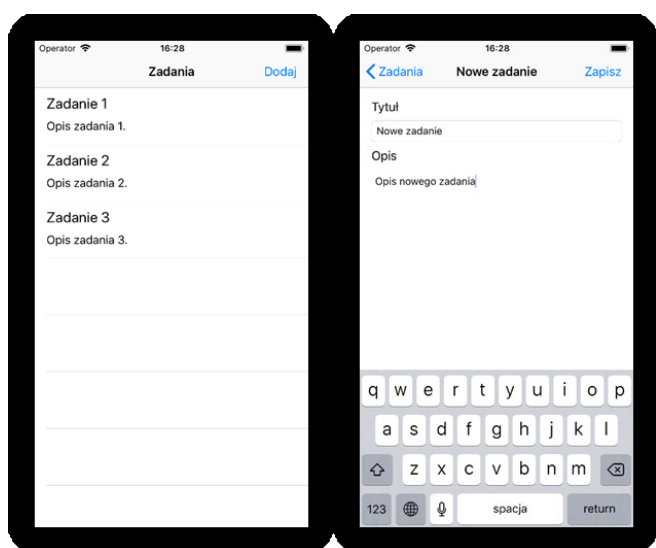
- 1) Pierwsza kompilacja – zmierzenie czasu kompilacji nowo utworzonego projektu;
- 2) Kompilacja po wprowadzeniu zmiany w implementacji aplikacji;
- 3) Kompilacja wynikowej aplikacji przeznaczonej do dystrybucji (np. w sklepie Google Play lub AppStore).



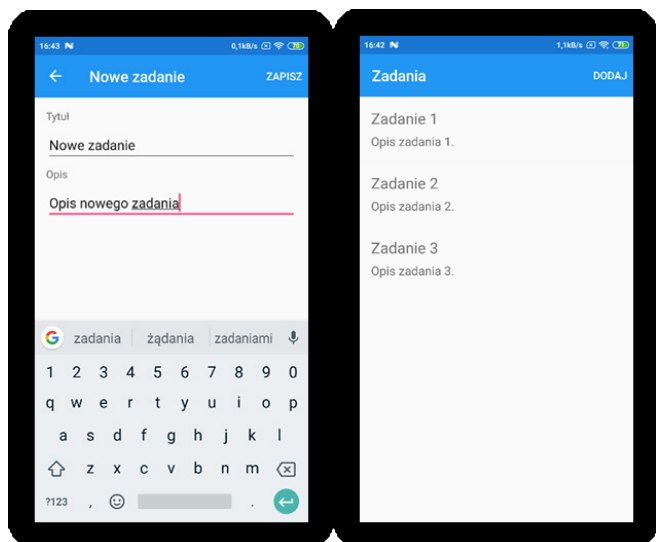
Rys. 3. Wygląd aplikacji mobilnej zadania na systemie Android utworzonej w środowisku Android Studio



Rys. 6. Wygląd aplikacji mobilnej „Zadania” w systemie iOS utworzonej w środowisku Visual Studio przy użyciu technologii Xamarin



Rys. 4. Wygląd aplikacji mobilnej zadania na systemie iOS utworzonej w środowisku XCode



Rys. 5. Wygląd aplikacji mobilnej „Zadania” na systemie Adnroid utworzonej w środowisku Visual Studio przy użyciu technologii Xamarin

Kolejny etap to porównanie czasu utworzenia samej aplikacji. Spotkano się tutaj z problemem oszacowania czasu ze względu na dość istotne różnice w wykorzystanych środowiskach programistycznych. Niektóre z nich to:

- wykorzystanie różnych języków programowania i związane z nimi możliwości,
- znaczne różnice w budowaniu interfejsu graficznego aplikacji,
- różnorodność narzędzi wspomagających tworzenie aplikacji, takich jak generowanie szablonów oraz automatyczne uzupełnianie kodu.

Z tego powodu postanowiono dokładnie opisać proces tworzenia przykładowej aplikacji „Zadania” na każdym z omawianych środowisk, a następnie odtworzenie go w rzeczywistości i zmierzenie czasu jego trwania poprzez nagranie ekranu w trakcie jego realizacji. Proces ten podzielono na trzy części:

- 1) Utworzenie i uruchomienie projektu;
- 2) Utworzenie oraz oprogramowanie okna listy zadań;
- 3) Dodanie okna oraz funkcji tworzenia nowych zadań.

Dzięki takiemu podziałowi, wraz z wynikami z poprzedniego etapu badań, utworzono dwa uproszczone procesy utworzenia oraz dystrybucji aplikacji „Zadania” na dwa systemy operacyjne Android oraz iOS: jeden zakładający wykonanie aplikacji tradycyjnie w dwóch osobnych środowiskach (Android Studio, XCode) oraz drugi – przy użyciu wieloplatformowej technologii Xamarin w środowisku Visual Studio (Tabela 1).

Mając na uwadze przytoczone wcześniej różnice w wykorzystanych do utworzenia aplikacji „Zadania” środowiskach, podjęto następujące kroki, aby zaprezentowany sposób porównania czasu tworzenia tej aplikacji był w jak największym stopniu wiarygodny:

- 1) Dokładnie sprecyzowano krok po kroku każdą wykonaną czynność podczas tworzenia aplikacji;

- 2) Proces tworzenia został w maksymalnym stopniu zoptymalizowany dla każdego z wykorzystanych środowisk;
- 3) Aplikacja wykonana została przez programistę z 4 letnim doświadczeniem w tworzeniu aplikacji mobilnych na systemy Android ora iOS, w każdym z odmawianych środowisk;
- 4) Starano się zachować jak największą płynność w trakcie tworzenia aplikacji.

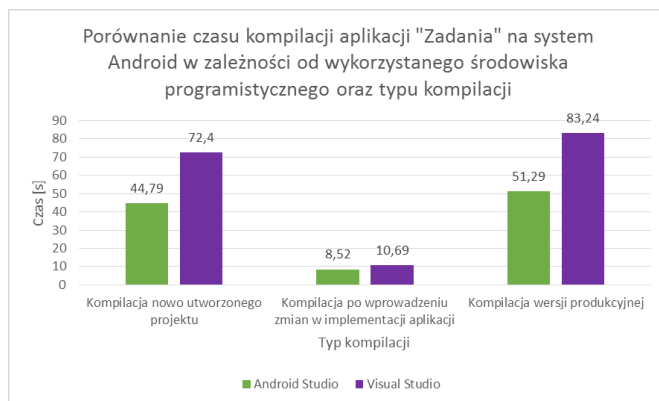
Tabela 1. Procesy tworzenia aplikacji mobilnej „Zadania” z podziałem na wykorzystane środowisko programistyczne

Lp.	Etap	
	Android Studio, XCode	Visual Studio
1	Utworzenie i uruchomienie projektu w środowisku Android Studio	Utworzenie i uruchomienie projektu w środowisku Visual Studio
2	Utworzenie oraz oprogramowanie okna listy zadań	Utworzenie oraz oprogramowanie okna listy zadań
3	Kompilacja nowo utworzonego projektu	Kompilacja nowo utworzonego projektu na system Android
4	Dodanie okna oraz funkcji tworzenia nowych zadań	Kompilacja nowo utworzonego projektu na system iOS
5	Kompilacja i uruchomienie aplikacji po wprowadzeniu zmian w implementacji	Dodanie okna oraz funkcji tworzenia nowych zadań
6	Kompilacja wynikowej aplikacji na system Android przeznaczonej do dystrybucji	Kompilacja i uruchomienie aplikacji po wprowadzeniu zmian w implementacji na systemie Android
7	Utworzenie i uruchomienie projektu w środowisku XCode	Kompilacja i uruchomienie aplikacji po wprowadzeniu zmian w implementacji na systemie iOS
8	Utworzenie oraz oprogramowanie okna listy zadań	Kompilacja wynikowej aplikacji na system Android przeznaczonej do dystrybucji
9	Kompilacja nowo utworzonego projektu	Kompilacja wynikowej aplikacji na system iOS przeznaczonej do dystrybucji
10	Dodanie okna oraz funkcji tworzenia nowych zadań	
11	Kompilacja i uruchomienie aplikacji po wprowadzeniu zmian w implementacji	
12	Kompilacja wynikowej aplikacji na system iOS przeznaczonej do dystrybucji	

5. Wyniki badań

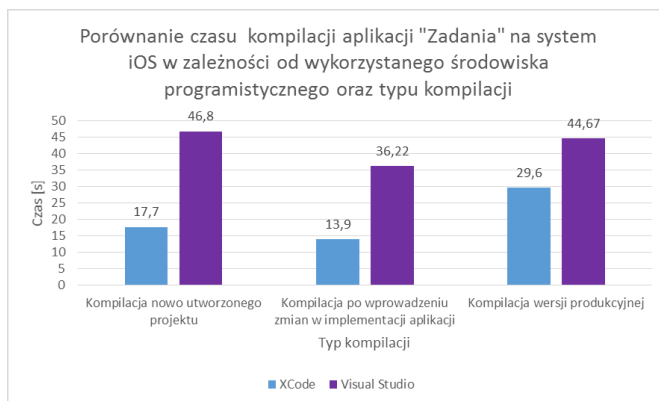
Na Rys. 7 oraz Rys. 8 przedstawiono wyniki badań polegających na zmierzeniu czasu trwania kompilacji aplikacji mobilnej na systemy Android (Rys. 7) oraz iOS (Rys. 8) ze względu na użyte środowisko programistyczne oraz warunki w jakich kompilacja została przeprowadzona. Zbadano kolejno: czas kompilacji nowo utworzonego projektu, czas kompilacji po wprowadzeniu zmiany w implementacji aplikacji oraz czas kompilacji aplikacji w wersji produkcyjnej przeznaczonej do dalszej dystrybucji. Na podstawie tych danych na pierwszy rzut oka można stwierdzić, że niezależnie od systemu operacyjnego oraz warunków przeprowadzenia kompilacji w każdym przypadku kompilacja aplikacji wykonanej przy użyciu technologii Xamarin trwa znacznie dłużej.

W przypadku systemu Android (Rys. 7) kompilacja nowo utworzonego projektu w środowisku Visual Studio trwała o około 61% dłużej czasu niż w środowisku Android Studio. Podobnie więcej czasu zajęła także kompilacja wersji produkcyjnej aplikacji (około 62%). Warto natomiast przyrzeć się czasom kompilacji aplikacji po wprowadzeniu zmian w implementacji, która jest wykonywana stosunkowo częściej niż dwa przeanalizowane powyżej typy, ponieważ czynność tę powtarza się przy każdej okazji poprawiania błędów w implementacji, czy dodawania nowych funkcji aplikacji. Różnice w tym przypadku są niewielkie, ponieważ jest to tylko o około 25% więcej czasu w przypadku środowiska Visual Studio.



Rys. 7. Porównanie czasu kompilacji aplikacji "Zadania" na system Android w zależności od wykorzystanego środowiska programistycznego oraz typu kompilacji

W przypadku systemu iOS (Rys. 8) różnice dla tych samych typów kompilacji są znacznie wyraźniejsze niż dla systemu Android. Największą różnicę stanowi tutaj kompilacja nowo utworzonego projektu, ponieważ o około 164% więcej czasu zajęła ona w środowisku Visual Studio. Około 160% więcej czasu zajęła także kompilacja po wprowadzeniu zmiany w implementacji aplikacji w środowisku Visual Studio. Najmniejszą różnicę w czasach można zaobserwować podczas kompilacji wersji produkcyjnej – zajęło to o około 50% więcej czasu w przypadku środowiska Visual Studio.



Rys. 8. Porównanie czasu kompilacji aplikacji "Zadania" na system iOS w zależności od wykorzystanego środowiska programistycznego oraz typu kompilacji

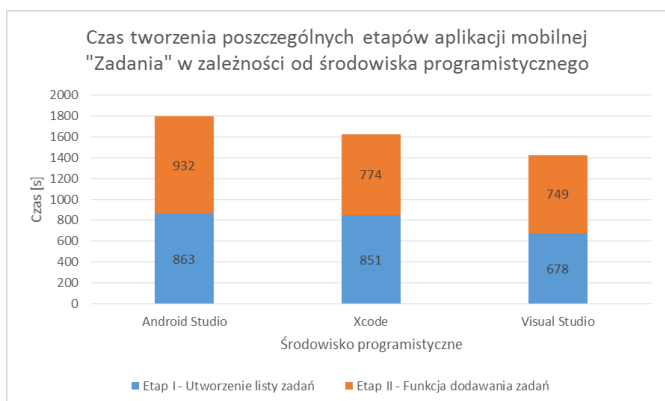
Na Rys. 9 przedstawiono czas utworzenia i inicjalizacji projektu aplikacji „Zadania”, w każdym z trzech

wykorzystanych środowisk programistycznych: Android Studio, XCode oraz Visual Studio. Warto zwrócić uwagę na środowisko Android Studio, które potrzebowało znacznie więcej czasu na inicjalizację i przygotowanie projektu do pracy. W stosunku do środowiska XCode było to o około 174% więcej czasu, natomiast w porównaniu do Visual Studio – około 95% więcej czasu.



Rys. 9. Czas utworzenia i inicjalizacji projektu aplikacji mobilnej w zależności od wykorzystanego środowiska programistycznego

Rys. 10 przedstawia porównanie czasów trwania poszczególnych etapów implementacji aplikacji mobilnej „Zadania” w zależności od wykorzystanych środowisk programistycznych. Na podstawie przedstawionych wyników można stwierdzić, że najmniej czasu zajęło zaimplementowanie aplikacji w środowisku Visual Studio. Należy tutaj także wspomnieć, że implementacja ta jednocześnie kierowana jest na dwa systemy operacyjne (Android oraz iOS), ponieważ wykonana została przy użyciu wieloplatformowej technologii Xamarin. W stosunku do implementacji w środowisku Android Studio czas ten był krótszy o około 20% oraz o około 12% – w porównaniu ze środowiskiem XCode.



Rys. 10. Czas tworzenia poszczególnych etapów aplikacji mobilnej "Zadania" w zależności od środowiska programistycznego

W Tabeli 2 oraz Tabeli 3 przedstawiono czasy trwania poszczególnych etapów tworzenia aplikacji „Zadania” w zależności od wybranej metody tworzenia: w dwóch osobnych środowiskach (Android Studio, XCode) (Tabela 2) oraz w wieloplatformowej technologii Xamarin przy użyciu środowiska programistycznego Visual Studio (Tabela 3). Podsumowanie tych czasów jako sumaryczny czas tworzenia aplikacji mobilnej „Zadania” na system Android oraz iOS

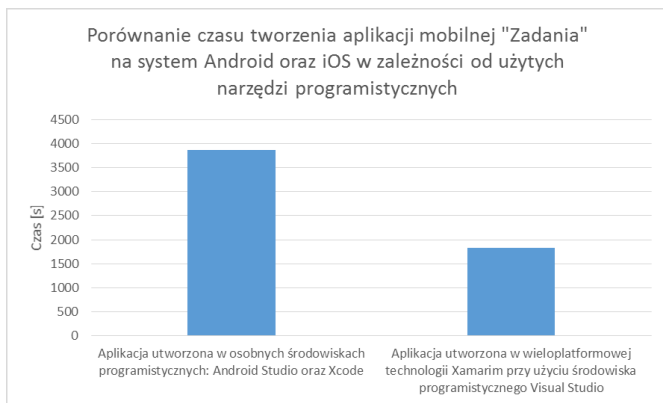
w zależności od użytych narzędzi przedstawiono na Rys. 11, z którego wynika, że czas potrzebny do utworzenia aplikacji mobilnej „Zadania”, przy użyciu wieloplatformowej technologii Xamarin oraz środowiska programistycznego Visual Studio, jest krótszy o około 53% w porównaniu z utworzeniem tej samej aplikacji używając dwóch oddzielnych środowisk: Android Studio oraz XCode.

Tabela 2. Czas trwania poszczególnych etapów tworzenia aplikacji w dwóch osobnych środowiskach programistycznych Android Studio oraz XCode

Lp.	Etap	Czas trwania [s]
1	Utworzenie i uruchomienie projektu w środowisku Android Studio	211
2	Utworzenie oraz oprogramowanie okna listy zadań	863
3	Kompilacja nowo utworzonego projektu	44,79
4	Dodanie okna oraz funkcji tworzenia nowych zadań	932
5	Kompilacja i uruchomienie aplikacji po wprowadzeniu zmian w implementacji	8,52
6	Kompilacja wynikowej aplikacji na system Android przeznaczonej do dystrybucji	51,29
7	Utworzenie i uruchomienie projektu w środowisku XCode	77
8	Utworzenie oraz oprogramowanie okna listy zadań	851
9	Kompilacja nowo utworzonego projektu	17,7
10	Dodanie okna oraz funkcji tworzenia nowych zadań	774
11	Kompilacja i uruchomienie aplikacji po wprowadzeniu zmian w implementacji	13,9
12	Kompilacja wynikowej aplikacji na system iOS przeznaczonej do dystrybucji	29,6
Suma		3873,8

Tabela 3. Czas trwania poszczególnych etapów tworzenia aplikacji w wieloplatformowej technologii Xamarin przy użyciu środowiska programistycznego Visual Studio

Lp.	Etap	Czas trwania [s]
1	Utworzenie i uruchomienie projektu w środowisku Visual Studio	108
2	Utworzenie oraz oprogramowanie okna listy zadań	678
3	Kompilacja nowo utworzonego projektu na system Android	72,4
4	Kompilacja nowo utworzonego projektu na system iOS	46,8
5	Dodanie okna oraz funkcji tworzenia nowych zadań	749
6	Kompilacja i uruchomienie aplikacji po wprowadzeniu zmian w implementacji na systemie Android	10,69
7	Kompilacja i uruchomienie aplikacji po wprowadzeniu zmian w implementacji na systemie iOS	36,22
8	Kompilacja wynikowej aplikacji na system Android przeznaczonej do dystrybucji	83,24
9	Kompilacja wynikowej aplikacji na system iOS przeznaczonej do dystrybucji	44,67
Suma		1829,02



Rys. 11. Porównanie czasu tworzenia aplikacji mobilnej "Zadania" na system Android oraz iOS w zależności od użytych narzędzi programistycznych

6. Wnioski

Podsumowując zebrane wyniki badań można stwierdzić, że użycie technologii Xamarin w kontekście czasu tworzenia aplikacji na dwa systemy mobilne (Android oraz iOS), jest jak najbardziej uzasadnione, ponieważ pozwala to na zaoszczędzenie ponad połowy czasu w stosunku do wytworzenia tej samej aplikacji osobno na każdy z systemów. Warto jednak mieć na uwadze wyniki z pierwszego etapu badań (dotyczącego czasów kompilacji aplikacji), gdzie wskazano znaczne różnice przemawiające na niekorzyść technologii Xamarin. Różnice te wynikają z dodatkowych zadań jakie są wykonywane podczas kompilacji aplikacji w środowisku Visual Studio przy użyciu technologii Xamarin. Z uwagi na to, że kod pisany jest w języku C# (innym niż natywny dla danej platformy), to musi on być najpierw odpowiednio przetworzony (w zależności platformy) do postaci natywnej [7]. Dodatkowo, w trakcie tego przetwarzania, wykonywany jest proces linkowania – usuwania z wynikowego kodu nieużywanych klas, w celu

zmniejszenia objętości aplikacji [8] [9]. Niemniej jednak pomimo, iż kompilacje aplikacji w procesie jej tworzenia są dość często wykonywaną czynnością (choćby przy wprowadzaniu drobnych poprawek, czy podczas poszukiwania i poprawiania błędów w implementacji), to w stosunku do sumarycznego czasu potrzebnego do jej wytworzenia jest to niewielki ułamek.

Literatura

- [1] <http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201903-201904-bar> [15.05.2019].
- [2] <https://www.altexsoft.com/blog/engineering/performance-comparison-xamarin-forms-xamarin-ios-xamarin-android-vs-android-and-ios-native-applications> [15.05.2019].
- [3] Corral L., Janes A., Remencius T.: Potential advantages and disadvantages of multiplatform development frameworks – A vision on mobile environments. *Procedia Computer Science* Volume 10, 2012, s. 1202-1207.
- [4] [https://pl.wikipedia.org/wiki/Mono_\(oprogramowanie\)](https://pl.wikipedia.org/wiki/Mono_(oprogramowanie)) [15.05.2019].
- [5] <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/app-fundamentals/custom-renderer/renderers> [15.05.2019].
- [6] Ch. Petzold, *Cross-platform C# programming for iOS, Android, and Windows*, Microsoft Press, Redmond, 2016.
- [7] <https://docs.microsoft.com/en-us/xamarin/cross-platform/app-fundamentals/building-cross-platform-applications/understanding-the-xamarin-mobile-platform> [15.05.2019].
- [8] <https://docs.microsoft.com/en-us/xamarin/ios/deploy-test/linker> [15.05.2019].
- [9] <https://docs.microsoft.com/en-us/xamarin/android/deploy-test/linker>. [15.05.2019].