

В.В. Герасимов, С.С. Біловол, К.В. Іванова
**ПОРІВНЯЛЬНИЙ АНАЛІЗ XAMARIN ТА
PHONEGAP FOR .NET**

Анотація. Проведено порівняльний аналіз технологій Xamarin та PhoneGapfor .NET, що активно розвиваються останнім часом. Наведено особливості даних технологій, їх переваги та недоліки, розглянуто перспективи їх розвитку та сфери застосування.

Ключові слова: Xamarin, framework, .NET, C#, PCL, SAP, PhoneGap, архітектура, віртуальна машина, платформа, інтерфейс.

Вступ. Зі збільшенням популярності новітніх комп'ютерних та мобільних технологій бізнес спільнота все більше і більше звертає увагу на різноманітні програмні продукти. Можливість вести бізнес на ходу дозволяє суттєво збільшити прибутки, мінімізувавши при цьому втрати часу. Проте наявність різних платформ ускладнює та здорожує процес розробки додатку. Обумовлюється дана проблема наявністю кардинально різного інструментарію розробника ПЗ для відповідної платформи. Дану проблему намагаються вирішити фреймворки для кросплатформної розробки. Напочатку свого існування усі вони були створені для платформи Java. Проте після релізу проекту Mono, який дозволяє розробку програм для не Windows платформ мовою C#, було створено такі фреймворки як Xamarin та PhoneGapfor .NET, які складають основу крос-платформних рішень для платформи .NET.

Метою роботи є дослідження та проведення порівняльного аналізу кросплатформних технологій Xamarin та PhoneGapfor .NET.

Основна частина. Xamarin являє собою еволюцію проекту Mono з усіма його недоліками та перевагами. Порівнюючи його із найближчим міжплатформним фреймворком PhoneGap можна побачити, що вони притримуються різних парадигм, хоча й мають багато схожих елементів.

Для більшого розуміння усіх відмінностей необхідно розглянути кожний з фреймворків детальніше.

На початку свого існування PhoneGap не мав своєї студії розробки і був розширенням інфраструктури проекту з відкритим кодом під назвою ApacheCordova. Детально розглядаючи PhoneGapfor .NET можна побачити, що він майже не відрізняється від PhoneGap. Єдиною відмінністю є можливість роботи у середовищі Microsoft VisualStudio.

Сам PhoneGap-додаток являє собою звичайний сайт для мобільного пристрою, щовідображається у елементі WebView. Проте на відміну від звичайного мобільного сайту, такий додаток має можливість використовувати такі апаратні можливості телефону, як акселерометр, камера, медіа-функції, сховище даних тощо. PhoneGap не дозволяє розробнику використовувати засоби, що надає платформа напряму. Замість цього існують PhoneGapплагіни, що контактують з API мобільної операційної системи (рис. 1). Якщо розробник не може знайти необхідний плагін— він вимушений написати бібліотеки для кожної з платформ, для яких збирається створювати додаток.

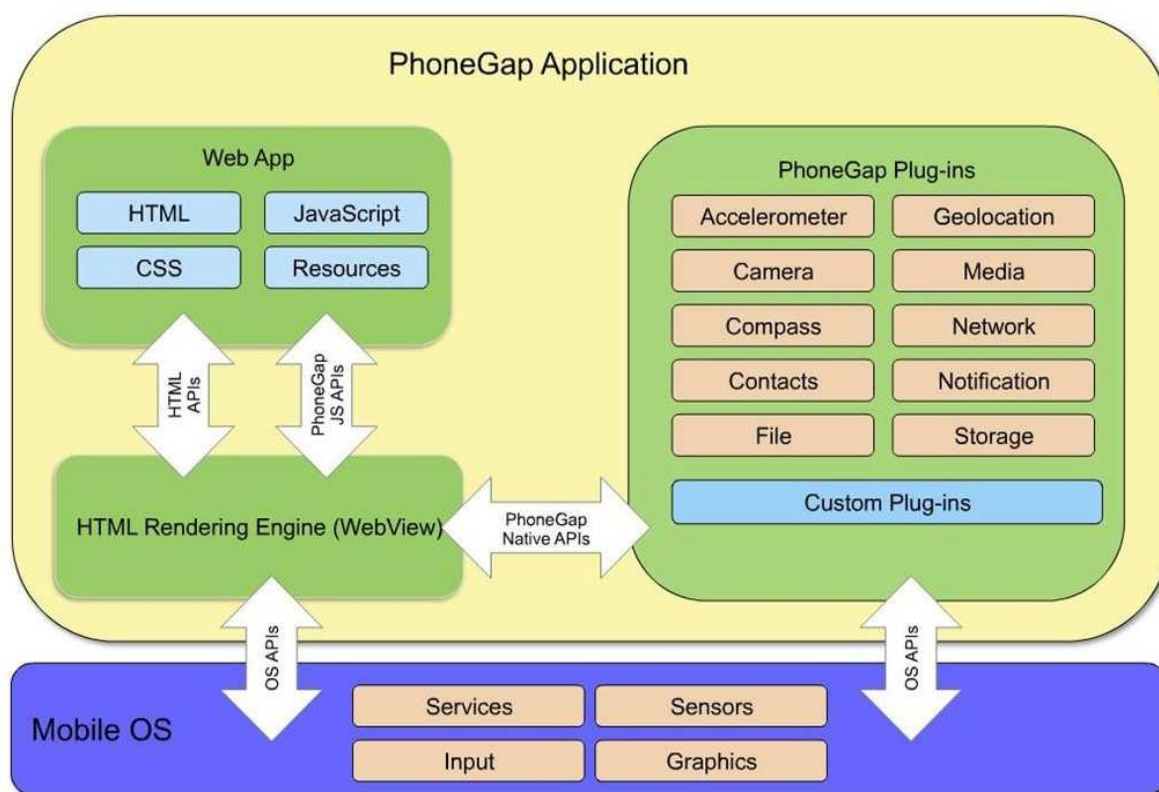


Рисунок 1 – Структура PhoneGap додатку

Інтерфейс створюється з використанням технологій HTML, CSS3, JavaScript і є універсальним для будь-якої мобільної операційної системи, для якої створюється додаток.

Основною перевагою даного підходу є повна абстракція від платформи, що дозволяє з легкістю зробити підтримку великої кількості операційних систем. PhoneGap підтримує Apple iOS, Android, Windows Phone, Blackberry, Symbian, Bada, webOS, Tizenі т.д. Проте на практиці велика кількість функціоналу операційних систем для деяких мобільних платформ не підтримується, а також у залежності від пристрою додаток може працювати дещо по різному.

Також єдиний інтерфейс для різних мобільних операційних систем значно спрощує і пришвидшує розробку самого додатку, проте PhoneGap-додаток, маючи структуру веб-сайту, працює помітно повільніше, ніж звичайний додаток для мобільної платформи.

Іншу парадигму кросплатформної розробки пропонує фреймворкXamarin. На відміну від PhoneGap, Xamarin надає більш тісний зв'язок із платформою, що допомагає розроблювати додатки, що не поступаються швидкодії додаткам, розробленим під конкретну платформу з допомогою нативних засобів.

З точки зору структури фреймворкуXamarin розподілено на декілька основних частин: MonoAndroid, MonoTouch та Mono. Найбільша відмінність MonoAndroid та MonoTouch від Mono полягає у тому, що Mono —це лише порт .NET Framework на Unix-подібні системи, що дозволяє розробляти програми для таких операційних систем як Mac та UbuntuLinux з використанням зручних засобів від Microsoft. Офіційно у рамках фреймворкуXamarinможлива розробка додатків лише для Mac OS, проте у рамках проекту Mono можлива розробка також і під UbuntuLinux.

Структура MonoAndroid та MonoTouch дещо складніша. Вони містять у собі набір компіляторів та обгортку нативного API конкретної платформи. Для кожної платформи створено свій компілятор, так як iOS та Android мають кардинально різні екосистеми.

Monoandroid та Monotouch, завдяки механізму обгортки, надають функціонал API платформ Android та iOS відповідно. Тісний зв'язок між Xamarin та нативним API означає також і тісний зв'язок із циклом роботи самого додатку.

Даний зв'язок можна побачити на прикладі роботи Android-додатку, створеного засобами Xamarin, де разом із самим додатком працює віртуальна машина Mono, яка обмінюється із віртуальною машиною Dalvik(або ART) зрозумілими для неї повідомленнями. На-

явність додаткової віртуальної машини суттєво збільшує розмір вихідного файлу додатку, проте ніяким чином не впливає на швидкість самого додатку. Пов'язано це із відмінною оптимізацією як фреймворку, так і самої віртуальної машини Mono.

Враховуючи тісний взаємозв'язок із мобільною операційною системою, кросплатформеність досягається не тільки за рахунок єдиної мови написання додатку, а також за рахунок структури самого додатку.

Додаток для різних мобільних операційних систем можна розбити на дві логічні частини (рис. 2):

1. Частина з платформи-залежним кодом;
2. Частина зі спільним у використанні кодом.

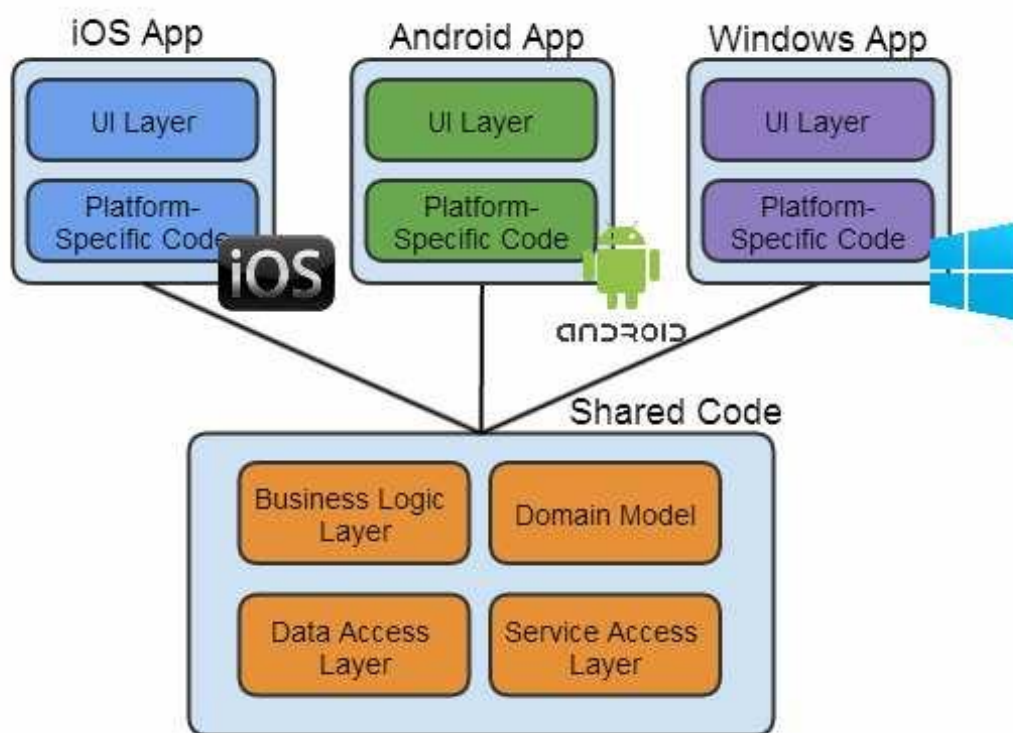


Рисунок 2 – Структура Xamarin-додатку

Платформи-залежний код містить специфічні для платформи механізми роботи з апаратними особливостями пристрою. До платформи-залежного коду також відноситься і механізм побудови та роботи користувацького інтерфейсу, що є платою за тісний зв'язок із нативним API.

Для збільшення коефіцієнту повторного використання програмного коду виділяють рівень спільного коду, який містить основну логіку роботи додатку, рівень доступу до сервісів, даних і т.д.

У рамках фреймворку Xamarin також було розроблено такі рішення як XamarinMobile та XamarinForms.

Основною проблемою подібних до Xamarin фреймворків є розробка уніфікованого інтерфейсу для усіх підтримуваних платформ без необхідності внесення змін до програмного коду у залежності від операційної системи пристрою. Для вирішення цього питання було створено XamarinForms — програмний шар, що абстрагується від реальної платформи та апаратно-залежного коду та уніфікує користувацький інтерфейс. Розробники Xamarin стверджують, що використання Xamarin.Forms допомагає збільшити об'єм спільного коду у додатках для трьох підтримуваних платформ до 90 відсотків.

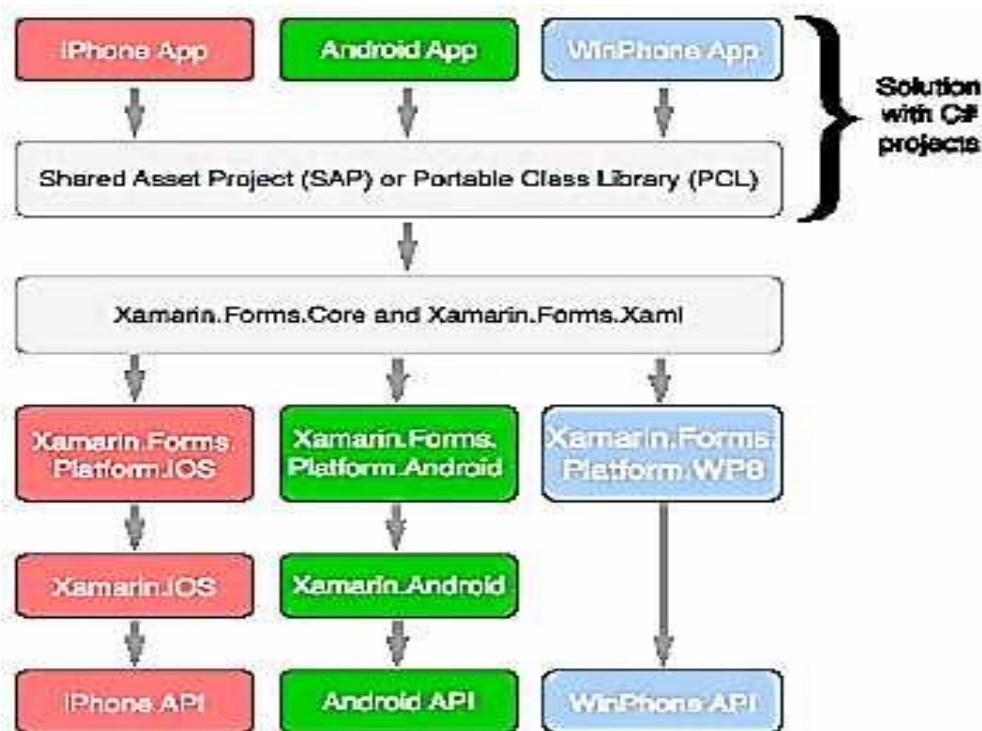


Рисунок 3 – Схема роботи XamarinForms

Як можна побачити зі схеми (рис. 3)— повної абстракції від платформи уникнути неможливо. Фреймворк повинен розуміти, для якої платформи додаток повинен бути створений. Далі виконання передається до бібліотеки PCL (PortableCommonLibrary) або SAP (SharedAsset Project), які і містять універсальний код для усіх платформ. PCL та SAP бібліотеки ставлять за мету збільшення об'єму спі-

льного коду, але механізми, завдяки яким досягається даний ефект, кардинально різні. У обох випадках створений проект стає системою з трьох додатків, проте працюватиме зовсім по-різному. У випадку PCL-підходу весь код буде зібраний до динамічної бібліотеки, до якої кожний проект посилається та зв'язує дані під час виконання. У випадку SAP усі спільні бібліотеки фактично включаються до складу усіх трьох додатків під час їх побудови.

Описаний рівнем вище інтерфейс аналізується ядром XamarinForms та через адаптери передається до API відповідної платформи. Не дивлячись на велику кількість проміжних етапів, швидкість Xamarin-додатку відповідна швидкості додатку, створеного нативними засобами.

Ще однією функціональною особливістю фреймворку є наявність XamarinMobile. Дана бібліотека надає можливість абстрагування від специфічного для платформи коду.

Кількість підтримуваних платформ Xamarin у порівнянні із PhoneGap значно менша. Підтримуються лише три найпопулярніші із них, а саме: Android, iOS та Windows Phone.

Не дивлячись на невелику кількість підтримуваних мобільних операційних систем, Xamarin має ряд переваг, а саме:

1. Швидкодія на рівні програм, створених нативними засобами розробки.
2. Повна підтримка функціоналу нативного API.
3. Повна підтримка усіх можливостей C#, таких як LINQ, lambdaexpressions, async/await і т.д.
4. Великий коефіцієнт повторного використання програмного коду.

Висновки. На даний момент ринок рішень для міжплатформної розробки для .NET Framework не дуже сильно розвинений. Існують лише PhoneGap for .NET та Xamarin. Обидві технології мають свої переваги та недоліки, і однозначного лідера обрати серед них практично неможливо.

Дешевий у розробці користувацький інтерфейс, широкий вибір доступних плагінів із різноманітною функціональністю у купі з непоганою швидкістю роблять PhoneGap непоганим засобом для створення невеликих кросплатформних мобільних додатків.

Проте проаналізувавши можливості Xamarinфреймворку та порівнявши його із перевагами, які надає PhoneGapfor .NET, можна дійти висновку, що найкращим вибором для створення великих між-платформних проектів є саме Xamarin. Не дивлячись на невелику кількість підтримуваних мобільних операційних систем, Xamarinмістить великий набір потужних засобів, які надають як .NET Framework, так і нативні бібліотеки. Додатково існує великий набір бібліотек, які пришвидшують розробку і зменшують вартість кінцевого продукту. Платою за можливість користуватись описаними вище перевагами є великий розмір додатків, але у більш ніж половині випадків розмір додатку не має критичного значення.

ЛІТЕРАТУРА

1. Cross-Platform MobileDevelopment: PhoneGapvsXamarin –
Режим Доступу: <http://www.justinshield.com/2014/05/cross-platform-mobile-development-PhoneGap-vs-xamarin/>
2. PhoneGapvsmonotouchfordataintensiveapp
Режим Доступу: <http://stackoverflow.com/questions/9191611/phone-gap-vs-monotouch-for-data-intensive-app>
3. Howtcreate HTML5 appson Windows PhonewithPhoneGap – режим
доступу: <https://msdn.microsoft.com/en-us/hh771462.aspx>
4. Разработкамобильных приложений на PhoneGap и jQueryMobile –
Режим Доступу: <http://habrahabr.ru/post/118059/>
5. Xamarin.Formsrecipes –
Режим Доступу:
<http://developer.xamarin.com/recipes/cross-platform/xamarin-forms/>
6. PhoneGapDocumentation –
Режим Доступу:
http://docs.PhoneGap.com/en/edge/guide_overview_index.md.html