# Hiểu về phát triển ứng dụng di động đa nền tảng (Cross-Platform)

Cho đến hiện nay, trên thị trường có rất nhiều nền tảng điện thoại thông minh như: Android, iPhone, BlackBerry, Nokia, Windows Phone (tại thời điểm này gồm có: Windows Phone 7 và Windows Phone 8), và WebOS. Gần đây với sự ra mắt của các nền tảng mới hơn như Samsung Bada và Meego, cũng đang có xu hướng phát triển mạnh và ngày càng có thế trên thị trường. Với một số lượng lớn các nền tảng điện thoại thông minh như vậy, đã tác động đáng kể và đòi hỏi quan điểm vững chắc của các nhà phát triển ứng dụng trên di động.

Trong năm 2000, chúng ta đã từng biết một hiện tượng tương tự trong thế giới máy tính để bàn (Desktop), với sự có mặt của các nền tảng như Windows (Microsoft), Mac (Apple) cùng nhiều phiên bản khác nhau của Linux và UNIX. Vào thời điểm đó, rất khó để xây dựng một sản phẩm mà có thể chạy trên mọi nền tảng. Kết quả hướng giải quyết được thực hiện trong nội bộ bằng cách xây dựng các Framework C++ và tích hợp sẳn các mô-đun. Đi cùng với đó, Sun đã xây dựng một nền tảng chung (Java Platform) và vấn đề đã dần được làm nhẹ đi.

Từ những năm 2004 đến năm 2008, cộng đồng các nhà phát triển đã nhìn thấy một loại phân mãnh khác đang diễn ra trong thế giới tình duyệt. Và dễ dàng nhìn thấy hơn ở đây đó là giữa Internet Explorer 6 với Firefox và Safari. Sau đó, cùng với sự ra đời của Chrome và các trình duyệt khác thì sự phân mãnh này ngày càng diễn ra mạnh, cụ thể hơn nữa. Bản chất của sự phân mãnh này chủ yếu là do một số trình duyệt không tuân thủ theo một số thông số kỹ thuật được đặt sẳn của W3C (World Wide Web Consortium). Và vì vậy, điều này không quá khó để giải quyết. Rất nhiều thư viện JavaScript đã giúp và giải thoát tình trạng Cross-Browser trên.

Sự phân mãnh trong thế giới điện thoại thông minh Mobile OS lại phức tạp hơn, bởi vì không có một đặc điểm kỹ thuật hay một chuẩn nhất định nào trong lĩnh vực phát triển này. Năm 2007, Apple và Google đều tung ra những nền tảng di động của thế hệ điện thoại thông minh. Và cho đến năm 2008 thì cả hai công ty này đều có thị trường phát triển riêng với số lượng người dùng tăng dần theo cấp số mũ.

Các nhà phát triển ứng dụng di động nhanh chóng tiếp cận với các nền tảng mới và xây dựng lại những ứng dụng tương tự trên iPhone, Android, BlackBerry, WebOS, Symbia và giờ đây lại phải thêm Samsung Bada vào danh sách này. Sự thách thức cũng được làm rõ như khi chúng ta phát triển ứng dụng cho iPhone thì cần MAC, cho BlackBerry thì lại cần Windows.

## Phân loại các ứng dụng di động

Ngày nay, lĩnh vực phát triển ứng dụng di động tồn tại ba khái niệm là: Native Application, Web Application và Hybird Application.

* **Native Application:** Là những ứng dụng được xây dựng theo cách chính thống với các ngôn ngữ lập trình, nền tảng do các nhà sản xuất di động quy định (ví dụ: Objective-C cho iOS, Java cho Android, .NET cho Windows Phone,… ).
  + Ưu điểm của Native Application là nhà phát triển có thể thoải mái truy cập vào hệ thống, phần cứng của thiết bị (như hệ thống tập tin, camera, microphone, cảm biến gia tốc,… ).
  + Nhược điểm là bị bó buộc với công nghệ và nền tảng mà nhà sản xuất đưa ra.
* **Web Application:** Là các ứng dụng được xây dựng trên nền tảng web (mà điển hình là HTML5), chỉ hoạt động trên trình duyệt của điện thoại. Tiền thân của ý tưởng này là những trang web có giao diện tùy biến cao, chạy được trên nhiều độ phân giải màn hình, về sau phát triển mạnh và hình thành nên một hướng đi mới cho việc phát triển ứng dụng di động.
  + Ưu điểm là có thể thoải mái phát triển ứng dụng mà không cần quan tâm tới nền tảng vì nó Cross-Platform.
  + Nhược điểm là không có được sức mạnh truy cập sâu vào hệ thống và thiết bị như native application.
* **Hybird Application:** Là con lai giữa hai khái niệm ***Native Application*** và ***Web Application*** kia. Có thể hiểu nôm na các ứng dụng này là: một native application chỉ có một đối tượng *webview* trên màn hình, dùng để hiển thị nội dung trang web application, và web application giao tiếp với native application thông qua một cầu nối (bridge) để mang lại sức mạnh của native application cho web application.

## Hiểu về Web Service

Là một nhà phát triển, Khi bạn nhìn vào các ứng dụng web ta cần phải hình dung hai lại web application đó là:

1. Web application truy cập thông qua trình duyệt.
2. Web application truy cập thông qua protocols như RESTful web services.

Tất cả các ứng dụng web phổ biến như Google, Facebook, Twitter, LinkedIn, MySpace, Flickr, và Picasa cung cấp một giao diện tĩnh cho các dịch vụ của họ. Nhiều công ty muốn phát triển các ứng dụng di động đa nền tảng với các web service của họ hoặc thông qua các web service khác. Trong khi đó, PhoneGap có thể làm việc cho các ứng dụng một cách độc lập với các tính năng của web application được mở rộng và khả năng tương tác với thiết bị.

## Tổng quan về Mobile Application

Điều đầu tiên cần lưu ý mobile application không phải là web application. Sự khác biệt được thể hiện trong cả bản chất và các tính năng được cung cấp.

Mobile application không phải là một web application

Cách người dùng tương tác với web application và mobile application cũng hoàn toàn khác nhau. Người dùng tương tác với điện thoại thông minh thông qua màn hình cảm ứng sinh động hơn và với nhiều phương thức khác.

Sự khác nhau về phương thức giao tiếp giữa ứng dụng trên điện thoại thông minh và web application

Sự khác biệt đó còn thể hiện ở trong việc nhận dạng vị trí thiết bị khả năng Push Notification (thông báo).

## Những thách thức trước khi phát triển ứng dụng đa nền tảng ra đời

Trong khi phát triển ứng dụng di động đang là xu hướng hiện nay, với số lượng ngày càng tăng của các hệ điều hành (OS) di động, có rất nhiều thách thức liên quan đến việc phát triển các ứng dụng di động.

### Sự phân mảnh của các nền tảng hệ điều hành

Xu hướng phân mảnh với số lượng ngày càng tăng của các nền tảng di động. Đầu tiên, đó là BlackBerry và điện thoại thông minh Symbian, sau đó đến sự phát triển mạnh mẽ của iPhone, Android. Chắc chắn rằng, nền tảng di động không dừng lại ở đó khi HP đến với WebOS, Microsoft đã giới thiệu Windows Phone, và giờ đây, lại đến Samsung với Bada.



Sự phân mảnh phát triển song song với số lượng của các hệ điều hành di động

### Tiêu hao nguồn nhân lực

Nếu chúng ta chọn việc xây dựng một ứng dụng di động trên từng nền tảng thì chúng ta phải đối mặt với nhiều vấn đề nan giải. Cần một số lượng đội (Team) phát triển ứng dụng tương ứng với mỗi nền tảng di động đó, đi kèm với đó sẽ là các mức chi phí, người quản lý (Team Leader) với nhiều rủi ro có thể xãy ra.



Sự gia tăng nguồn nhân lực cho các nền tảng phát triển khác nhau

### Yêu cầu về kinh nghiệm với một nền tảng di động nhất định

Sự đòi hỏi kinh nghiệm chuyên môn của một nhà phát triển ứng dụng di động là điều bắt buộc. Điều đó đảm bảo mức độ chuyên nghiệp trong việc xây dựng ứng dụng di động ở một nền tảng nào đó.



Phát triển ứng dụng riêng biệt trên các nền tảng tạo sự trải nghiệm khác nhau ở người dùng đầu cuối

### Sự phân mảnh theo tính năng

Khả năng phân mảnh theo tính năng của các mobile application cũng xãy ra cao, khi khả năng đáp ứng các chức năng cơ bản của hệ điều hành lẫn phần cứng tương thích.



Phân mảnh theo tính năng giữa sự khác nhau các hệ điều hành di động

### Phân mảnh trong môi trường phát triển ứng dụng

Môi trường phát triển ứng dụng là một trong những phân mảnh quan trọng. Chúng ta phải cần ít nhất hai hệ điều hành Windows và Mac để phát triển ứng dụng cho các nều tảng sau đây.

1. iOS
2. Android
3. BlackBerry
4. WebOS
5. Symbian
6. Windows Phone

Hơn thế nữa, chúng ta sẽ phải sử dụng một loạt các ngôn ngữ lập trình, chẳng hạn như Java, C++ và Objective C. Ngoài ra, ta còn phải sử dụng một số IDE, chẳng hạn như Xcode, Eclipse.

Bảng yêu cầu môi trường phát triển

|  |  |  |  |
| --- | --- | --- | --- |
| **Mobile OS** | **Operating System** | **Software/IDEs** | **Programming Language** |
| iOS | Mac only | Xcode | Objective C |
| Android | Windows/Mac/Linux | Eclipse/Java/Android Development Tool (ADT) | Java |
| BlackBerry | Windows mainly | Eclipse/JDE, Java | Java |
| Symbian | Windows/Mac/Linux | Carbide.c++ | C++ |
| WebOS | Windows/Mac/Linux | Eclipse/WebOS plugin | HTML/JavaScript/C++ |
| Windows Phone | Windows mainly | Visual Studio 2010/2012 | C#, .NET, Silverlight or WPF |

## Hướng phát triển ứng dụng đa nền tảng của PhoneGap

PhoneGap ra đời từ những đặc tính chung và phổ biến của các nền tảng di động.

### Browser Component là một nền tảng chung

Như đã được đề cập, từ vài năm trước các trình duyệt đã ngày càng bị phân mảnh do mức độ tuân thủ chuẩn W3C khác nhau giữa các nhà sản xuất trình duyêt. Tuy nhiên cho đến hiện tại, các trình duyệt đã có cải thiện tốt hơn trong các phiên bản mới của mình. Đặc biệt, điều đó thể hiện rõ hơn trong thế giới trình duyệt của các nền tảng di động. Bởi vì hầu hết các trình duyệt trên nền tảng di động hiện đại đều dựa trên cùng một **webkit-based browser**.

Cũng mới đây, sự phân mảnh của các trình duyệt cũng dần giảm khi tuân thủ theo một chuẩn công nghệ mới hơn đó là HTML5/CSS3. Xem bảng dưới đây cung cấp thông tin giữa các nền tảng di động tương ứng với các nền tảng trình duyệt. Qua đó ta thấy, ngoại trừ Windows Phone 7 thì tất cả các nền tảng khác đều tuân thủ theo chuẩn **Webkit-based**. Và cũng may mắn thay, khi các trình duyệt đó cũng đang tuân thủ các chuẩn của HTML5/CSS3. Trong thời gian qua việc tuân thủ, thống nhất các qui chuẩn cũng dần được cải thiện.

Nền tảng di động và trình duyệt

|  |  |
| --- | --- |
| **Mobile OS** | **Browser** |
| Android | Webkit-based |
| iOS | Webkit-based |
| BlackBerry 6.0 + | Webkit-based |
| Windows Phone 7 | IE 7-based |
| WebOS | Webkit-based |
| Nokia | Webkit-based |
| BADA | Webkit-based |

PhoneGap sử dụng các trình duyệt hiện đại và xây dựng ứng dụng trên chuẩn HTML5/CSS3 nhằm tạo ra sự tương đồng giữa các nền tảng di động khác nhau.

### Mobile Application Webview

Tất cả các nền tảng động đều hỗ trợ nhúng một trình duyệt vào trong ứng dụng. Điều đó có nghĩa là trên trên một ứng dụng di động ta hoàn toàn có thể hiển thị một trang web HTML - thành phần được nhúng vào đó được gọi là Webview. Như vậy, trong một ứng dụng ta có thể hiển thị nội dung chẳng hạn như thông tin về sản phẩm, hướng dẫn sử dụng,… thông qua một trang web tĩnh. Hơn nữa, các Webview có thể tương tác với nền Ajax thông qua một web service.

### Khả năng khai thác tối ưu tới thiết bị

Về mặt kỹ thuật, tất cả các trình duyệt được nhúng vào đều có thể sử dụng JavaScript để sử dụng các module để gọi các mã Java/C++/Objective C của một ứng dụng thông thường (Native Application).

### Sử dụng chuẩn HTML5/CSS3 để viết ứng dụng

Sự bùng nổ của công nghệ HTML5/CSS3 đã làm tăng khả năng tương tác và làm giàu tính năng của các ứng dụng web. HTML5 không chỉ cung cấp khả năng định dạng hiển thi trang web mới, mà hơn thế nữa là tính năng hỗ trợ đa phương tiện, chạy nền, khả năng hỗ trợ cơ sở dữ liệu và còn nhiều hơn thế nữa. CSS3 cung cấp những khả năng mà bây giờ có thể cạnh trang với Flash. Đó là một thực tế mà các trình duyệt di động phải sớm chấp nhận các tiêu chuẩn W3C, và cũng đồng nghĩa là các ứng dụng trên điện thoại di động cũng dần phù hợp với ứng dụng HTML5/CSS3.

### Kết luận

Ứng dụng viết bằng PhoneGap sử dụng HTML5, Javascript và CSS3. Sử dụng PhoneGap, Các lập trình viên không cần biết nhiều đến ngôn ngữ lập trình ứng dụng di động truyền thống tương thích với các nền tảng tương ứng.

Mặc dù PhoneGap sử dụng những qui chuẩn được cũng cấp để xây dựng ứng dụng, nhưng khả năng mở rộng của nó rất linh hoạt dựa vào những plug-in được cung cấp sẳn.

PhoneGap đang là một hướng đi phát triển cho việc xây dựng ứng dụng đa nền tảng hiện nay.