ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

# Methods of Cross-Platform Development Mobile Applications.

**Pavel Sergeyevich Ptitsyn*.**

Research Institute of Semiconductor Engineering, JSC, Russian Federation, 394033, Voronezh, Leninsky Prospekt, 160a.

**ABSTRACT**

In the design and development of applications based on mobile technologies there are significant technical and technological difficulties associated with the integration of these applications into a single information landscape of the enterprise, and support for various versions of mobile operating systems that these applications were developed. Because of the above-identified difficulties, significantly increases the time and budgets of developing mobile applications, the cost of implementation and ownership of these systems. In addition, there is no quick response to new market challenges, jeopardizing perspectives for business development companies. To solve the above-identified issues, it used specialized cross platform technologies, which provide the transfer of the program code on various versions of operating systems and devices. The aim of this work was the systematization and analysis of proven methods of cross-platform development mobile application, including such approaches as mobile web applications, hybrid mobile applications, and native stand-alone runtime mobile applications, including the description of the technical features and benefits of using of each of these methods. As well as analysis of existing software solutions implementing these methods of the cross-platform development mobile applications.

**Keywords:** software development, mobile applications, cross platform approaches

***Corresponding author***

ISSN: 0975-8585

## INTRODUCTION

Currently, mobile devices in terms of functionality are not inferior to personal computers, which opens up a wide field of application for mobile devices in various spheres of human activity. For efficient use of mobile devices in specific spheres of life need to develop appropriate mobile applications, implementing the desired functionality. This circumstance gives rise to a demand for the development of various mobile applications. Currently, however, mobile devices are manufactured based on a variety of mobile operating systems, such as Android, iOS, Windows Phone. Each of mobile operating systems in the mobile application development requires using of their own development tools and programming languages. Development of mobile applications separately for each operating system is a fairly time-consuming and expensive solution to this problem.

To solve the above-identified problems using specialized development environment, providing the transfer of the program code on various versions of operating systems and devices. The main advantages of the technology cross-platform application development are:

* Code reuse in applications.
* Access to application software modules and plug-ins, which simplifies their integration into other services or tools.
* Ability to develop applications using web technologies.
* No needs additional knowledge and experience of technology to create applications for different mobile platforms.
* Reduction of time debugging and publishing applications.
* Absence of complexity with control version of applications, and adding a new functionality in the project.

According to consulting firm Gartner in 2016 more than half the applications developed for mobile devices, will be implemented based on cross-platform mobile application development technologies [1].

Given work contains descriptions the main methods of cross-platform development mobile applications, including their features, as well as advantages and disadvantages given approaches.

## METHODS

In works [2,3] described the development of cross-platform technologies for mobile applications. There are three main approaches:

- The development of mobile web applications. These applications are built using web technologies, and as a runtime using the web browser of mobile operating system. Access to the application by using URL- address on the Internet.
- The development of hybrid applications. This approach involves the development of applications using web technologies, but allows the application to access the main hardware features of the device where it runs. The application is installed directly on the device through the application store the operating systems.
- The development of applications for stand-alone runtime. Stand-alone runtime environment is a set of libraries with a single software interface and implemented for each target operating system. Applications written in a programming language, but the library is used for different operating systems can be run on any operating system for which there is a realization of the selected stand-alone runtime. The application is installed directly on the device through the application store the operating systems.

The details of the technical architecture technologies of cross-platform mobile applications development shown in Figure 1 [4].

ISSN: 0975-8585

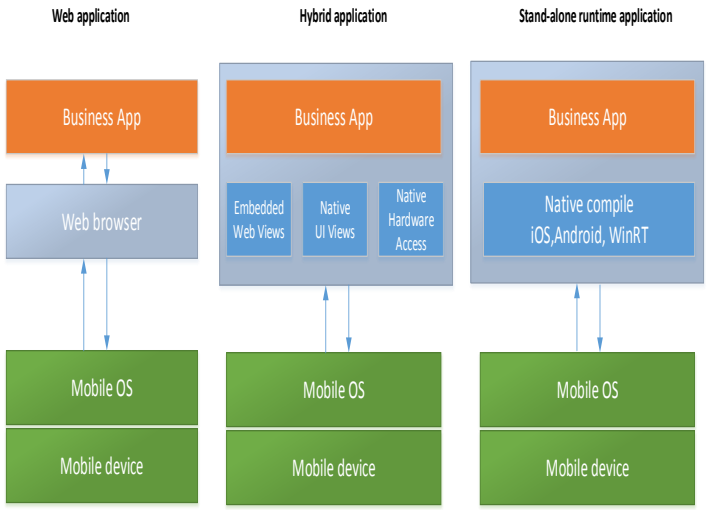


**Fig.1. The technical architecture technologies of cross-platform mobile applications development**

The table 1 shown the matrix of comparing the main technical specifications of cross-platform technologies for the mobile application development [4,5].

**Table 1. Main technical specifications of cross-platform technologies for the mobile application development**

| Technical specification | Web appli-cations | Hybrid Appli-cations | Stand-alone (native) appli-cations |
|---|---|---|---|
| Programming languages | HTML5 JavaScript | HTML5 JavaScript Native languages (option) | C++ Java C# Other modern languages |
| Code portability | High | High | High |
| Support features of mobile devices | Low | Average | High |
| Access to the system API | No | Average | Average |
| Extended graphics features | Average | Average | High |
| UI\UX | Average | Average | Average |
| Flexible updates | High | Average | Average |
| Application deployment | Low | High | High |

## RESULTS AND DISCUSSION

### Mobile web applications

The mobile web application is a website adapted for use in the browsers of mobile operating systems. Mobile web application may exist both independently and be an adapted version of the general website. The choice of tools for developing web applications is very wide. For the development of server-side, applications can be used development technologies: PHP, .NET, Java, and others. To create a complex GUI client application used modern web technologies [6]:

* HTML (the current version of HTML5) - a markup language of web documents.

- CSS (the current version of CSS3) - a formal language for describing the appearance of a document written using HTML.
- JavaScript programming language and related extension libraries, including jQuery, facilitating interaction JavaScript and HTML.

The usage of these technologies allows creating mobile applications with a complex graphical user interface and rich functionality. Web application hosted on the web servers on the Internet, and accessed through a browser on a mobile device.

In work [7, 8] noted the widespread use of HTML5 technology to develop mobile web applications because given technology supports the following features:

- Open geolocation API.
- Touch-sensitive devices.
- WebGL technology for 3D-graphics rendering.
- Vector graphics (Canvas).
- Video streaming.
- Optimize the display of content on mobile browsers (Viewport).
- Real-time communication (WebRTC).
- Work with multithreading (WebWorkers).
- Browser templates.

Furthermore, to simplify the process of creating mobile applications using specialized sets of components (frameworks), such as [8]:

- JQuery Mobile - contains a set of components to optimize the web application to run on a variety of mobile platforms based on HTML5 technology.
- Sencha Touch - contains a set of components that allow creating graphical interfaces mobile web applications designed to manage the touch of a finger. The library is compatible with all modern mobile browsers.
- Kendo UI - provides a set of components for creating mobile applications based on server-side technologies: PHP, ASP.NET MVC, and JSP.

In work [9] noted the approach to the development of mobile web applications based on visual designer of web forms that allows to implement a graphical user interface and the business logic of the application.

The typical architecture of the mobile web application implemented using components jQuery Mobile shown in Figure 2.

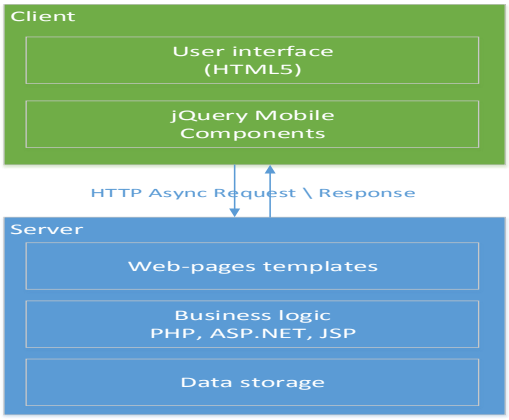


**Fig.2. The architecture of mobile web applications jQuery Mobile**

ISSN: 0975-8585

One of the approaches for development of mobile web applications, based on the idea of setting up mobile applications using pre-written templates (for example iBuildApp). The graphical user interface (designer) implemented many typical elements such as widgets to display RSS, view photos, play audio & video stream, communication in the social networks, etc. This approach allows to create many types of applications, but is not suitable for building complex applications with unique behavior [9]. Figure 3 shows the graphical user interface of the designer.

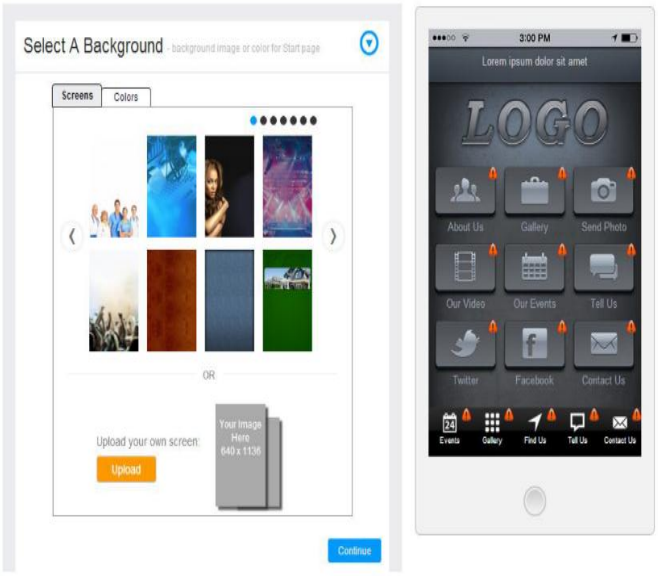


**Fig.3. iBuildApp designer for building mobile applications**

In addition, it should be noted Codiqa software is a service for creating HTML5 mobile application interfaces, based on a library of interface elements jQueryMobile. Creating an interface based on the principle dragging of the elements from the palette to the canvas applications and editing the properties of these elements. The application logic defined using the integrated JavaScript code editor. The graphical designer of Codiqa application shown in Figure 4.

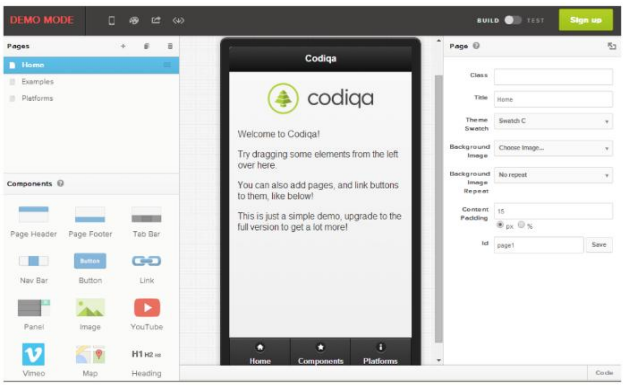


**Fig.4. Codiqa designer for building mobile applications**

The work [10] identified the following features of the web approach to mobile application development:

* The single server for all of the mobile platforms.
* Guaranteed work in web browsers of modern mobile operating systems.
* Ability to adjust the appearance of the application, depending on the target operating system.
* No need to install the application on the devices, the application accessed through a hyperlink.

* Graphical user interface created using web technologies, similar to its own graphical user interface of the operating system, but does not repeat it in full.

The significant disadvantage of web approach to the development of cross-platform mobile applications are the following [10,11]:

* Low performance of web application (JavaScript & DOM).
* There are no support for store -based applications.
* There are no access to most of the hardware capabilities of the mobile device (geolocation, accelerometer, and other), and operating system features (contacts, calendar, and other).
* There are no possibility to work in off-line mode.
* The complexity of creating a rich user interface.
* There are no support to send push- notifications.

**Hybrid mobile applications**

The hybrid approach of developing applications for mobile operating systems combines the method of developing mobile web applications, and the method of development of native applications. This approach based on web mobile application wrapped into a native application, which has access to the capabilities of a mobile devices, such as a camera, a gyroscope, local data storage, and other [12].

The development of hybrid mobile applications based on the specific software frameworks, such as PhoneGap and Titanium, which allow access to the hardware capabilities of the mobile device and operating system functions. Using web technologies to create hybrid applications allows applying the same technologies as in the development of common mobile web applications (HTML5, CSS3, and JavaScript).

The technical architecture of a hybrid approach to the development of mobile applications, as an example of the PhoneGap framework [13], shown in Figure 5.

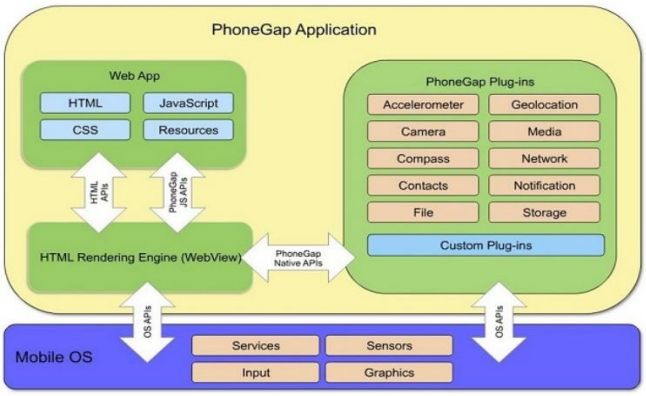


**Fig.5. Technical architecture of a hybrid approach to development mobile applications**

In the works [14, 15] identified the following features of a hybrid approach to mobile application development:

* The unified process of development and testing for multiple target mobile platforms. The application built and tested in different browsers, and then assembled for each mobile platform in the form of native applications.
* The usage of web technology can guarantee to run on most modern mobile platforms. Also eliminates the need to learn different programming languages and technologies for each mobile platform separately.
* Access to the capabilities of mobile devices provided by a software interface, which implemented in JavaScript libraries like PhoneGap and Titanium.
* Display the application in full screen mode using the built-in controls of operating system.

* The application deployment using the appropriate operating system mobile application store (App Store, Microsoft Marketplace, and Google Play).

In [16] marked the most popular technologis for the development of mobile applications based on a hybrid approach, such as PhoneGAP (34% market share), Appcelerator Titanium (21% market share), Adobe AIR (19% market share).

PhoneGap is open source software framework, also known under the name Apache Cordova. It enables developers to gain access to the hardware capabilities of the device on which the application is running, using JavaScript programming language. In addition, it is worth noting that for PhoneGap has developed more than 200 plug-ins, greatly simplifying the process of application development [17].

Another well-known platform for hybrid applications is the Appcelerator Titanium Mobile. The platform includes a library for each target operating system that it provides as an intermediate layer for the application that developed in JavaScript language. Therefore, the application does not address directly to the target operating system libraries, and through the Titanium library. This allows writing the code once and moving it to all supported platforms [17].

Building applications based on the Appcelerator Titanium platform divided into three conceptual steps [18]:

* Precompiling. The role of Titanium precompiler is to optimize of JavaScript code: reducing the number of spaces, character size, and so on. Moreover, the creation of a hierarchy of dependencies of all API functions used in the project.
* Front-end compilation. Its role is to generate the appropriate platform native code and, if necessary, establishing native construction project and the specific code necessary to the compiler of the platform.
* Compiling platform and packaging. Each platform has a set of appropriate tools (e.g., Xcode for iOS) for the final compilation of the application. After compiling an application is packaged, and can be to run the emulator, or tested on the device.

In addition, it is important that Titanium allow expanding the possibilities, the writing of third-party modules on the Objective-C language and C for iOS, and Java language for Android. In particular, this means that if a certain class is not ported to the Titanium API from iOS SDK, the developer has the opportunity to do it himself, if he knows related programming languages.

Technical Architecture of Appcelerator Titanium shown in Figure 6.

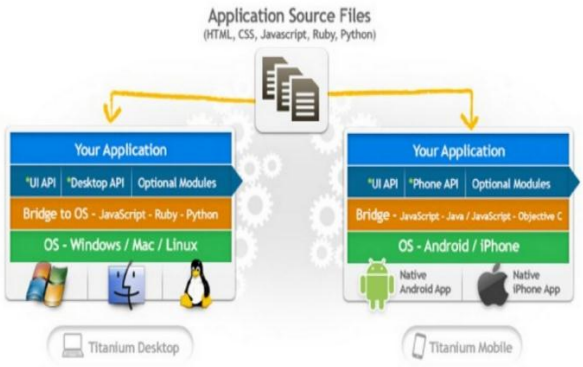


**Fig.6. Technical architecture of Appcelerator Titanium**

In addition, it should be noted that there is a commercial platform Appcelerator Platform Enterprise, which implements the following functional:

- Analysis, including monitoring application data in real-time, performance monitoring, analysis of the logs, analysis of the processes of creating and deploying applications.
- Integration, including built-in connectors to popular industrial platforms (Salesforce, SAP, Oracle, Microsoft Dynamics and SharePoint). As well as the ability to create their own connectors to any services.

Another example of a hybrid approach to the development of mobile applications is described in [19] based on integrated development environment Adobe Flash Builder. This environment implemented on IDE Eclipse development tool, and designed to create a multi-functional cross-platform web applications using open environment Flex. To create a GUI using the internal editor and language MXML, for writing code language ActionScript. The result of the compilation is a SWF- file that, for example, can be used to perform in the web browser (using Adobe Flash Player), or as a standalone using AIR middleware. In addition, during compilation automatically generated HTML-page that contains code to load and run the SWF- file.
Technical architecture of Adobe Flash Builder shown in Figure 7.

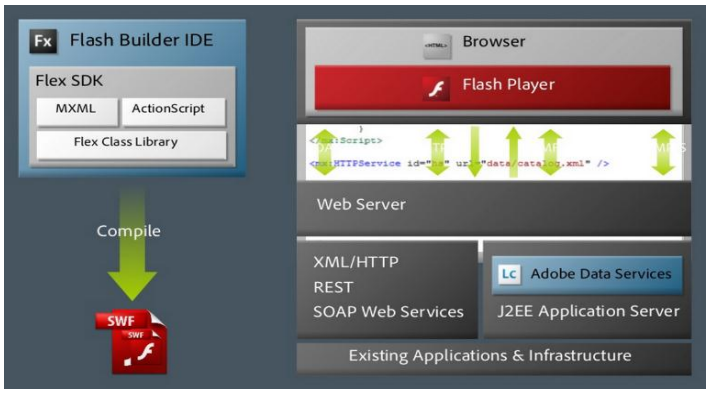


**Fig.7. Technical architecture of Adobe Flash Builder**

Adobe Flash Builder has the following distinctive functionality [19]:

- Powerful tools generated code based IDE Eclipse, includes editors for MXML, Adobe ActionScript, and CSS, supports syntax highlighting, auto-completion, code folding, interactive step-through debugging, and automatic generation of common code.
- Visual layout, the design of application created visually using a large number of built-in components and preview user interface layout, design and behavior.
- Development of data-oriented applications using protocols based REST \ SOAP.
- Interactive data visualization, the ability to create data dashboards and interactive data analysis carried out by placing the selected chart type and tying it to a data source using the Flex library charts.
- Native support for Adobe AIR, contains all the tools necessary to build, debug, package, and signing the AIR-applications.

According to [19], the segment of hybrid solutions cannot be considered a monolithic, the developers offer very different architectures, targeting different business challenges. On one side are the designers of applications, offering a set of standard functional blocks that can quickly assemble a complete application. On the other side of the platform with its own middleware, which remains universal, allow to design unique applications that rely on different mobile platforms.

The common architecture of hybrid solution is to use native libraries and JavaScript to implement the user interface. Of these platforms can be mentioned, for example, PhoneGap. The native shell acts as a proxy, allowing the interface in JavaScript language calls related API to access the mobile OS devices, which can be done directly from web browser.

It may also include hybrid interpreted by applications that use native API, but the business logic described in its abstract level. The interface is also often built on JavaScript or Lua. Members of this family are Appcelerator, iPFaces, JMango, Octopod and Prhomobile.

According to Gartner analysis [1], the hybrid applications that offer a balance between HTML5 web apps and native developments, will be used for more than half of mobile devices by 2016.

**Native applications based on stand-alone runtime**

By developing applications for stand-alone runtime understood the technology of writing applications with the possibility of the creation of assemblies for multiple operating systems. This source code is directly responsible for the business logic of the application, and has to be identical for all target operating systems. Access to operating system features and functions of a mobile device takes place through specific runtime libraries, which provide a single programming interface to access all target operating system. Accordingly, the application can be created for any operating system, which provides implementation of related runtime libraries [20].

The development of mobile applications for stand-alone runtime has the following features [20]:

* A single server for all of the mobile platforms.
* Ability to write a single graphical interface, interpreted for each target operating system.
* The performance of developed applications are comparable with native applications.
* Ability to placing applications in app stores for the relevant operating system.
* The maximum possible access to the hardware capabilities of the mobile devices.

One of the most well-known platform for the development of cross-platform mobile applications using stand-alone runtime is Xamarin platform. The platform based on the Xamarin Mono project is a free software for developing applications on .NET Framework that provides implementation for various operating systems. Xamarin allows developing applications in C# language for Windows Phone, Android and iOS, which will use MonoTouch and MonoDroid libraries, supplied with the Xamarin platform [21].

The technical architecture of Xamarin platform shown in Figure 8, and contain the following layers [21]:

* Data Layer (DL) – the layer of data storage.
* Data Access Layer (DAL) – the layer provides access to the data storage.
* Business Layer (BL) - the layer describes the business logic of the application.
* Service Access Layer (SAL) - the layer is responsible for interaction with remote services based on common communication standards such as REST, JSON, and WCF.
* Application Layer (AL) - the layer contains the platform-specific code to the target operating system that is associated with a specific library implementation of access to operating system resources.
* User Interface Layer (UI) – the layer containing implementation of the user interface. It can be written separately for each target operating system.


**Fig.8. Technical architecture of Xamarin platform**

ISSN: 0975-8585

The process of compiling native Xamarin assemblies shown in Figure 9. From the standpoint of application performance between iOS and Android there is a key difference - the way they are precompiled. To run the application on the Android uses a virtual Java-machine Dalvik. Native applications written in Java, compiled into a certain intermediate byte code that interprets the commands in Dalvik processor at the time of execution of the program. This so-called Just-in-time compilation (compilation on the fly). In iOS, another compilation model used - Ahead-of-Time (compilation before execution). Xamarin considers this distinction by providing separate compilers for each of these platforms, which allows the output is native applications that can use all the hardware and software platform resources [21].

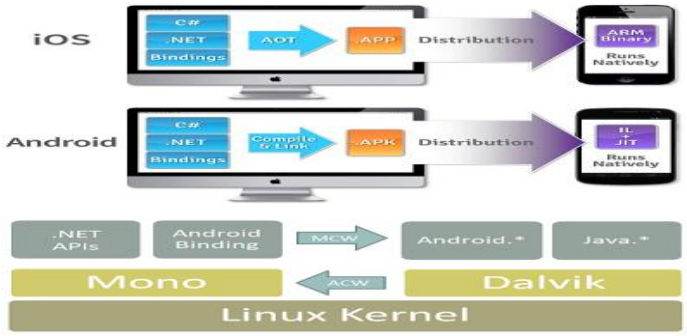


**Fig.9. The process of compiling native Xamarin assemblies**

The development of applications on Xamarin platform are possible in an integrated environment Xamarin Studio or the popular development environment Microsoft Visual Studio, which supplied to a special plug-in. More work with Xamarin Studio is considered in [21] is an example of creating a simple application for iOS. In [21] describe examples of interaction with the hardware capabilities of the mobile device through Xamarin, shows significant differences in access to the Bluetooth protocol stack on operating systems iOS and Android, which indicates the existence of the likelihood of having to write different code to perform the same tasks on different operating systems.

Xamarin platform has a store of components where there are hundreds of free and paid extensions, including for solving business problems and system integration:

* SAP Mobile Platform SDK - provides integration connectors to access ERP SAP.
* Azure Mobile Services - a library for integration with Azure cloud storage and cloud services.
* SQLCipher - a library for access to databases.
* JSon.NET - a library for converting and receiving data in JSON format.

Other examples of the implementation of this approach to the development of cross-platform mobile applications are presented in the publication [22, 23], such as J2ME Polish (development language Java), Marmalade (development language C ++), iFactr (development language C #), SIO2 (development language C / C ++), Corona (development language Lua).

## CONCLUSIONS

The result of this work is research of the technologies of cross-platform development mobile applications, including such approaches as, mobile web application, hybrid application, and native stand-alone runtime application. Identified the strengths and weaknesses of given approaches to cross-platform development, including such aspects as performance, time of development, implemented functionality, application deployment, interaction with corporate information systems. However, each of the given approaches provide significant advantages over the standard mechanisms of development mobile application including the following:

Code reuse in applications.

- Using specified software modules and plug-ins, which simplifies their integration into other services and tools.
- The possibility for developers to develop applications using familiar technologies and programming languages.
- The debugging and publishing applications much faster.
- There are no problems with version control of applications, and the adding of new changes in the developing project.
- Reduce the time and cost of developing mobile applications in 2-5 times compared to using the standard development tools.
- Reduce the cost of system integration of enterprise applications in 2-3 times compared to using the service bus solutions.
- Reduce the cost of ownership of mobile applications is 2-4 times compared to using standard application support.

## ACKNOWLEDGMENTS

The Ministry of Education and Science of the Russian Federation supported the work (Agreement #14.576.21.0067, unique identifier agreement RFMEFI57614X0067).

## REFERENCES

[1] Gartner, “By 2016 more than half of all mobile apps will be hybrid”. - Stamford, CT: Gartner. Inc., 2013.
[2] VDC Research Group, “Native VS Cross platform development”. - Boston, MA: VDC Research Group Inc., 2010.
[3] F. Gavalas, “Development Platforms for Mobile Applications: Status and Trends”, IEEE Software magazine, vol. 28 (1), 2011.
[4] M. Palmieri, “Comparison of cross-platform mobile development tools”, 16th International Conference on Intelligence in Next Generation Networks, Berlin, 2012.
[5] J. Ohrt, O. Turau, “Cross-Platform Development Tools for Smartphone Applications”, Computer, vol. 45 (9), 2012.
[6] S. Allen, G. Vidal, “Pro Smartphone Cross-Platform Development: iPhone, Blackberry, Windows Mobile and Android Development and Distribution”. - New York: Apress, 2010.
[7] A. Suyesh, “Cross-Platform Mobile Development: An Alternative to Native Mobile Development”. - Kalmar, Sweden: Linneuas University, 2013.
[8] S. Kaltofen, M. Milrad, A. Kurti, “A Cross-Platform Software System to Create and Deploy Mobile Mashups”, Lecture Notes in Computer Science, vol. 6189 (1). - Berlin, Germany: Springer Berlin Heidelberg, 2010.
[9] P. Sharon, E. Anderson, “Operating systems for mobile computing”, Computer Science, vol. 25 (2), 2009.
[10] G. Hohpe, B. Woolf, “Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions”. - Boston, MA: Addison-Wesley Longman Publishing Co. Inc., 2003.
[11] I. Adusei, K. Kyamakya, K. Jobmann, “Mobile positioning technologiesin cellular networks: an evaluation of their performance metrics”, In MILCOM2002. Proceedings, vol. 2 (1), 2002.
[12] P. Dharma, Q. Zehg, “Introduction to Wireless and Mobile Systems”. - Boston, MA: Cenage Learning, 2011.
[13] S.Z. Asif, “Next Generation Mobile Communications Ecosystem: Technology Management for Mobile Communications”. - Hoboken, NJ: John Wiley & Sons, 2011.
[14] M. Firtman, “Programming the mobile web”. - New York, NY: O’Reilly Media, 2010.
[15] VisionMobile, “Cross-platform developer tools 2012, Mobile Industry Altlas”. - London, UK: VisionMobile Ltd., 2012.
[16] J. Spurlock, “Bootstrap. Responsive Web Development”, New York, NY: O'Reilly Media, 2013.
[17] A. Charlesworth, “The ascent of smartphone”, Engineering Technology, vol. 4 (3), 2009.
[18] C. Duarte, A. Afonso, “Developing once, deploying everywhere”, Computer Science, vol. 5(1), 2011.
[19] B. Fling, “Mobile Design and Development: Practical Concepts and Techniques for Creating Mobile Sites and Web Apps”. - San Francisco, CA: O'Reilly Media, 2009.

[20] A. Redda, “Cross platform Mobile Applications”. - Tronheim, Norway: Norwegian University of Scietnce and Technology, 2012.
[21] Xamarin, “Building Cross Platform Applications”. - San Francisco, CA: Xamarin, 2014.
[22] R. Want, “When cell phones become computers”, Pervasive Computing, vol. 8(2), 2009.
[23] M.H. Goadrich, M. P. Rogers, “Smart smartphone development: iOS versus Android”, Proc. SIGCSE, 2011.