

ӘОЖ 004.42

**Ш.Б. Шадықұл, П.А. Қожабекова, Н.С. Заурбеков, А.Б. Баймусаева**

магистрант, М.Әуезов атындағы ОҚУ, Шымкент, Қазақстан  
т.ғ.к., доцент, М.Әуезов атындағы ОҚУ, Шымкент, Қазақстан  
т.ғ.д., профессор, М.Әуезов атындағы ОҚУ, Шымкент, Қазақстан  
магистр, аға оқытушы, М.Әуезов атындағы ОҚУ, Шымкент, Қазақстан  
**\*Корреспондент авторы: pernesh-63@mail.ru**

## МОБИЛЬДІ ҚОСЫМШАЛАРДЫ ҚҰРАСТЫРУДЫҢ ТЕХНОЛОГИЯСЫ

### Түйін

Өздеріңіз білетіндей, мобильді қосымшаларды құрастыру процесіндегі маңызды элементтердің бірі – интеграцияланған құрастыру ортасын дұрыс таңдау, ол тек платформаға ғана емес, сонымен қатар оларды құрастыруға қатысатын мамандардың дайындық деңгейіне және біліктілігіне байланысты. Осы ғылыми мақаланың аясында мобильді қосымшаларды құрастырудың ең танымал және заманауи технологиясы қарастырылады. Бұл мақалада мобильді қосымшаларды құрастыруда қолданылатын жергілікті(нативный) және кросс-платформалық технологиялардың артықшылықтары мен кемшіліктеріне жеке - жеке талдау жасалады. Мақалада қарастырылған бағдарламалық жасақтаманы жасау құралдары құрылымының қажетті операциялық жүйесінде орнатылған тәуелсіз бағдарламалық өнімдерді жасауға мүмкіндік береді. Бұл орталар бағдарламалар жазуда арнайы дағдылар мен тәжірибені қажет етпейді және сонымен бірге әртүрлі бағдарламалау тілдерін пайдалана отырып, жоғары сапалы мобильді қосымшаларды жасау мүмкіндігін береді. Талданған мобильді қосымшаны құрастыру технологиялары ретінде келесі технологиялар таңдалды: жергілікті(нативный) және кросс-платформалық.

**Кілттік сөздер:** мобильді қосымша, жергілікті(нативный) технология, кросс-платформалық технология, iOS, Xcode, Swift, Android, React Native, Flutter, Ionic, Xamarin, PhoneGap.

**Кіріспе.** Мобильді қосымшаларды құрастыруда жергілікті(нативный) немесе кросс-платформалық құрастыру технологиялары қолданылады.

Жергілікті және кросс-платформалық құрастыру дегеніміз не? -деген сұрақты қарастырайық.

Жергілікті(нативный) технология (ағылшын тілінен native – родной дегенді білдіреді) арқылы мобильді операциялық жүйені құрастырудің түпнұсқалық тілдері мен құралдарын пайдалануды білдіреді.

iOS -та мобильді қосымшаларды құрастыру Xcode ортасының Swift тілін (бұрын Objective-C) қолдану арқылы жүзеге асырылады.

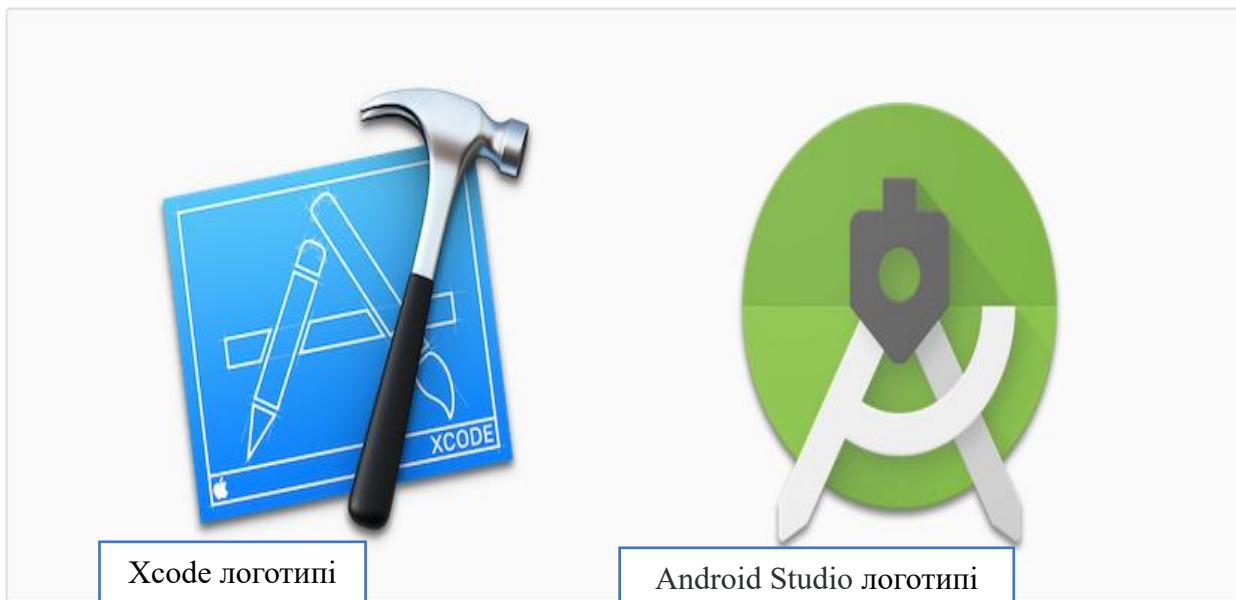
Android платформасында мобильді қосымшаларды құрастыру технологиясын қолдану барысында Android Studio ортасы және Kotlin тілі қолданылады (2018 жылға дейін негізгі тілі ретінде Java қолданылды).

Әрбір мобильді қосымшаны құрастыру ортасы код жазуға, интерфейстерді жобалауға, жөндеуге, профильдерді (бақылауға) және қолданбаларды құруға арналған утилиталардың тұтас жиынтығын қамтиды. Қоршаған орта да, сәйкес утилиталар жиынтығы да әрбір мобильді операциялық жүйе үшін арнайы жасалған және мобильді қосымшаларды құрастырудің ең ыңғайлы және қуатты құралдары болып табылады.

Мобильді қосымшаны құрастырудің кросс-платформалық технологиясы JavaScript тілдер тобына негізделген қосымшаны құру үшін арнайы фреймворктарды қолдануды қамтиды. Қосымшаның бүкіл құрылымы мен логикасы JavaScript-те келесі құралдардың көмегімен жасалады: React Native, Flutter, Ionic, Xamarin, PhoneGap және т.б. содан кейін жергілікті іске қосу элементіне оралады, яғни XCode немесе Android Studio үшін негізгі жобаға

біріктіріледі. Бұл бірден бірнеше операциялық жүйелер үшін бірдей логикамен жоба жинақтарын жасауға мүмкіндік береді.

**Теориялық талдау.** Қарапайым ұқсастықты дербес компьютерлер жағдайында көруге болады: MS Word, Skype, электрондық пошта агенттері, күнтізбелер – бұл жұмыс үстелінің операциялық жүйесі үшін жергілікті құрастырылған қосымшалар. Браузерде болып жатқан барлық нәрсе (веб-сайттар, онлайн мәтіндік және графикалық редакторлар, әлеуметтік желілер, чаттар, форумдар) кросс-платформалық технологиялар болып табылады.



Сурет 1. Xcode және Android Studio логотип көрінісі

### **Жергілікті(нативный) мобильді қосымшаларды құрастыру технологиясының артықшылықтары:**

Қазіргі таңда iOS және Android операциялық жүйелерінде өз ана тілдерінде мобильді қосымшаны құрастыру және жаңа технологияны қолдану келесі артықшылықтарға ие:

#### **-Мобильді қосымша жұмысының жылдамдығы**

Мобильді қосымша бастапқы құрастыру құралдарымен (Xcode, Android Studio) жасалғандықтан, жобаны құрастыру нәтижесінде алынған код осы платформа үшін оңтайлы болып табылады.

Мобильді қосымша құрылғының толық аппараттық қолдауына ие болады (бірдей кескіндерді өңдеуді жеке процессор жүзеге асырады, ол үшін арнайы - GPU қолданылады) және көп ағынды күрделі тапсырмаларды орындау, сонымен қатар фондық режимде мазмұнды жүктеу үшін қолданылады.

Қосымшаны құрастыру барысында бағдарламашылар кодтың барлық бөлімдерінің жылдамдығын өлшей алады және қажет болған жағдайда оларды оңтайландырады. Сондай-ақ олардың иелігінде жедел жадты пайдалануды бақылау, бақылаусыз ақпараттың тарап кетуін болдырмау және т. б. құралдар бар.

#### **-Іске асырудағы икемділік**

Интерфейсті құрудағы шектеулерден және жобаларды кросс-платформалық құрастыруға арналған фреймворктармен жүктелген визуалды эффекттердің күрделілігінен айырмашылығы, мобильді қосымшаларды құрастырудің жергілікті технологиясында белгілі

бір мобильді операциялық жүйенің технологиялары қабілетті барлық нәрсені жүзеге асыруға болады.

### **-Жаңа технологияларды қолдану және кросс-платформалық фреймворктарға тәуелділік**

Құрылғы мен операциялық жүйе өндірушілері ұсынатын жаңа бағдарламалық құрал мен аппараттық құрал сәйкес жаңартулар шығарылғаннан кейін бірден іске асыру үшін қолжетімді болады.

Мысалы, iOS 9-да қосымшалардың ішінен іздеу мүмкіндігі бар. Олардың әрқайсысында нәтижелерді белгілі бір іздеу сұранысы бойынша қайтаратын арнайы әдіс жүзеге асырылуы керек. Нәтижесінде, осы функционалдылық іске асырылған жергілікті iOS қосымшалары үшін iOS жүйесіндегі іздеу бөлімі арқылы мазмұнды іздеу мүмкіндігі қол жетімді болады және қосымшаларды, контактілерді, оқиғаларды және басқа ақпаратты іздеу жүзеге асырылады.

Мобильді қосымшаларды құрастыруға арналған кросс-платформалық технология жағдайында мұндай функционалдылықты жүзеге асыру үшін iOS 9 шығарылымын ғана емес, сонымен қатар сәйкес құрылымның жаңартылуын күтуге тура келеді және қашан қолдау көрсетілетінін болжау мүмкін емес.

### **-Тестілеудің жеңілдігі мен сапасы**

Қолданбаның құрылғының аппараттық ресурстарын пайдалануын бақылауға арналған жоғарыда аталған құралдардан басқа, құрастырушілер мен тестілеушілердің қарамағында технологиялардың толық спектрі бар.

Біріншіден, бағдарлама жұмыс істеп тұрған кезде жүйенің барлық параметрлері автоматты түрде басқарылады. Егер бағдарлама күтілгеннен көп жадты немесе орталық процессор ресурстарын қолдана бастаса, ол назардан тыс қалмайды.

Екіншіден, бірлік сынақтарын кеңінен қолдану мүмкіндіктері-қолданбадағы әрбір дерлік әдісті автоматты түрде тестілеу. Егер кодтың кез-келген өзгеруіне байланысты қосымшаның бір бөлігі дұрыс жұмыс істемей қалса, жаңа нұсқа жай жиналмайды, ал бағдарламашы оның себебін бірден көреді.

Үшіншіден, қателерді қашықтан бақылау жүйелерін біріктіруде кең мүмкіндіктер бар. Әрбір жергілікті жобада қатені және оның себебін кез-келген пайдаланушының құрылғысында көруге мүмкіндік беретін тиісті функционалдылық бар.

### **5. App Store және Google Play қосымшалар дүкенін толық қолдау**

Екі компания да қазіргі уақытта мүмкін болатын тиісті платформаларда қосымшаларды пайдалану кезінде пайдаланушыларға барынша оң тәжірибе алуға мүдделі.

Бұл қосымша мүмкіндігінше жоғары сапалы көрінуі керек дегенді білдіреді (егер экранның ажыратымдылығы жоғары болса және суреттер бұлыңғыр болса, қолданба App Store дүкенінде қабылданбайды), мүмкіндігінше жылдам жұмыс істейді (егер қолданба элементтердің шағын тізімін 20-30 секундта көрсетсе, ол да жіберіп алмайды) және тұтастай алғанда бәрі әдемі және ыңғайлы болуы керек.

Егер осы параметрлердің кез келгені тым төмен болса немесе мүлде орындалмаса, қолданба дүкенге жіберілмейді. Егер олар мобильді қосымшаларды жасауға арналған кросс-платформалық технологиялармен қол жеткізу өте қиын және әдетте мүмкін емес болса, олар сәйкес келмесе, сіздің өтініміңізді арнайы жарнама бөлімдерінде орналастыру үшін ешқашан тиісті компаниялар қарастырмайды(Featured).

Featured бөлімдеріндегі қосымшалардың ішінде интерфейсі жүйелік болып табылмайтын ойын жобаларын қоспағанда, App Store да, Google Play де кросс-платформалық технологиялардың көмегімен жасалмаған.

## **Мобильді қосымшаларды құрастырудің кросс-платформалық технологиясының артықшылықтары**

Кросс платформалық құрастыру ортасының келесі жағымды жақтары бар:

1. Бір уақытта бірнеше платформаға арналған қолданбаны енгізу үшін азырақ ресурстар қажет. Бұл, шын мәнінде, Android және iOS платформаларында Android және iOS үшін қосымшаларды құрастыруға арналған кросс-платформалық технологияның мәні - бірдей код екі платформада да жұмыс істейді. Жобада жұмыс істейтін бағдарламашылардың дәл жартысы қажет. Дизайнер графиканың бір жинағын ғана жасайды. Мұның бәрі жұмыс уақытының санын және жоба бюджетін қысқартады.

1. Қосымшаны іске асыру үшін бірден бірнеше платформаның астына енгізу үшін аз ресурстарды қажет етеді. Android Және iOS қосымшаларын құрастырудің кросс-платформалық технологиясының мәні де осында. Android және iOS платформаларында қосымшаларды құрастыруда екі платформада да бірдей код жұмыс жасайды. Бұл бағдарламашылардың жұмыс уақыты мен жоба бюджетін азайтуға септігін тигізеді.

2. Мобильді қосымшаларды құрастыруға аз уақыт қажет етеді. Бірегей интерфейс элементтерінің және кросс - платформалық қосымшаларды құрастырудың қарапайым технологияларының болмауына байланысты өнімдерді құрастыруға уақыт аз болады.

3. Өнімді жаңартудың жеңілдетілген циклі. Егер жобамызға өзгертулер енгізу керек болса немесе қандай да бір қатені түзету қажет болса, бұл жоба қамтылған барлық платформалар үшін бірден жасалады.

4. Сайттың мобильді нұсқасын пайдалану мүмкіндігі. Көп жағдайда мобильді қосымшаларды құрастырудің кросс-платформалық технологиясына арналған тілдер JavaScript тілдер тобына кіреді. Сондықтан, егер сізде жобаланып отырған сайттың мобильді нұсқасы бар болса, код пен ақпараттардың көпшілігін қосымша құрастыруда өзгеріссіз қолдануға мүмкіндік болады.

5. Бірыңғай қолданбалы логиканы қолдану. Қосымшаның жұмысына енгізілген Логика, барлық платформалар үшін бірдей жұмыс жасауға кепілдік беріледі. Бұл көбінесе операциялық жүйелердің әртүрлі архитектурасына байланысты кемшіліктердің бірі болуы мүмкін.

**Қорытынды.** Техникалық тұрғыдан және жасалған интерфейснің сапасы тұрғысынан мобильді қосымшаларды құрастыруда жергілікті технологияның артықшылықтары өте көп деуге болады. Дегенмен, кросс-платформалық технологияларға негізделген салалар да бар: оларға ойын секторларын және тестілеу жобаларын жатқызуға болады.

Қазіргі таңда ойындарды құрастыруда кросс-платформалық технологиялардың басым көпшілігін қолданады. Бұл сапаға нұқсан келтірместен ойын жобасын құрастыруды тездетеді, өйткені бұл жағдайда арнайы графикалық құрылымдар қолданылады (ең танымалы – Unity 3D).

Егер қандай да бір тестілеу өткізу үшін, кез-келген бір жобаны тез жасау шығу қажет болса, бұл жағдайда жобаның бір уақытта бірнеше платформада жұмыс істеуін талап етсе, кросс-платформаны іске асыру оңтайлы шешім болып табылуы мүмкін.

Егер құрастырылатын жобамыз ойын болмаса, ұзақ мерзімді дамуға бағытталған және пайдаланушылардан жағымды әсер алуды қажет етсе, онда мобильді қосымшаны жергілікті түрде жасау ұтымды болады. Мобильді қосымшаны құрастыру әдісі таңдалғаннан кейін, қосымшаны құрастыру құнын талқылау уақыты да келеді.

### **Әдебиеттер тізімі**

1. Голощапов А. Google Android программирование для мобильных устройств. Санкт-Петербург 2015-438 с.

2. Соснов А. Основы проектирование информационных систем—М.:ДМК Пресс,2016—1020 с.
3. Бурнет Э. Привет, Android! Разработка мобильных приложений. / Э. Бурнет. – Санкт-Петербург: Питер, 2018. – 256 с.
4. Бушуев А.П., Прожди Р.Г. Практичный планшет на Android. – СПб: Наука и Техника, 2015. - 336с.
5. Колисниченко Д. Программирование для Android. – СПб: БХВ-Петербург, 2017. - 272 с.
6. Анализ мобильных приложений и инструментальных средств их разработки[Электронный ресурс]. Режим доступа: <https://www.fundamental-research.ru/ru/article/view?id=39079> (дата обращения: 22.01.2025).

### References

1. Goloshhapov A. Google Android programmirovaniye dlja mobil'nyh ustrojstv. Sank-Peterburg 2015-438 s.
2. Sosnov A. Osnovy proektirovaniye informacionnyh sistem—M.:DMK Press,2016—1020 s.
3. Burnet Je. Privet, Android! Razrabotka mobil'nyh prilozhenij. / Je. Burnet. – Sankt-Peterburg: Piter, 2018. – 256 s.
4. Bushuev A.P., Prozhdi R.G. Praktichnyj planshet na Android. – SPb: Nauka i Tehnika, 2015. - 336s.
5. Kolisnichenko D. Programmirovaniye dlja Android. – SPb: BHV-Peterburg, 2017. - 272 s.
6. Analiz mobil'nyh prilozhenij i instrumental'nyh sredstv ih razrabotki[Jelektronnyj resurs]. Rezhim dostupa: <https://www.fundamental-research.ru/ru/article/view?id=39079> (data obrashhenija: 22.01.2025).

**Ш.Б. Шадькул, П.А. Кожобекова<sup>\*</sup>, Н.С. Заурбеков, А.Б. Баймусаева**  
магистрант, ЮКУ им. М. Ауэзова, Шымкент, Казахстан  
к.т.н., доцент, ЮКУ им. М. Ауэзова, Шымкент, Казахстан  
д.т.н., профессор, ЮКУ им. М. Ауэзова, Шымкент, Казахстан  
магистр, старший преподаватель, ЮКУ им. М. Ауэзова, Шымкент, Казахстан  
<sup>\*</sup>Автор для корреспонденции: [pernesh-63@mail.ru](mailto:pernesh-63@mail.ru)

### ТЕХНОЛОГИИ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

#### Аннотация

Как известно, одним из важных элементов в процессе сборки мобильных приложений является правильный выбор интегрированной среды сборки, которая зависит не только от платформы, но и от уровня подготовки и квалификации специалистов, участвующих в их сборке. В рамках данной научной статьи будет рассмотрена самая популярная и современная технология создания мобильных приложений. В этой статье будет проведен индивидуальный анализ преимуществ и недостатков нативных и кроссплатформенных технологий, используемых при разработке мобильных приложений. Рассмотренные в статье инструменты разработки программного обеспечения позволяют создавать независимые программные продукты, установленные в необходимой операционной системе устройства. Эти среды не требуют специальных навыков и опыта в написании программ и в то же время позволяют создавать высококачественные мобильные приложения с использованием различных языков программирования. В качестве анализируемых технологий разработки мобильных приложений были выбраны следующие технологии: локальные(нативные) и кроссплатформенные.

**Ключевые слова:** мобильное приложение, нативные технологии, кроссплатформенные технологии, iOS, Xcode, Swift, Android, React Native, Flutter, Ionic, Xamarin, PhoneGap.

**Sh.B. Shadykul, P.A. Kozhabekova, N.S. Zaurbekov, A.B. Baimusayeva**

Master's student, M.Auezov SKU, Shymkent, Kazakhstan

Cand.Tech.Sci., Associate Professor, M.Auezov SKU, Shymkent, Kazakhstan

Dr.Tech.Sci., Professor, M.Auezov SKU, Shymkent, Kazakhstan

Master, Senior Lecturer, M.Auezov SKU, Shymkent, Kazakhstan

\*Corresponding author's email: perlesh-63@mail.ru

## TECHNOLOGY OF DEVELOPING MOBILE APPLICATIONS

### Abstract

As you know, one of the important elements in the process of building mobile applications is the correct choice of an integrated build environment, which depends not only on the platform, but also on the level of training and qualifications of the specialists involved in their assembly. Within the framework of this scientific article, the most popular and modern technology for creating mobile applications will be considered. This article will provide an individual analysis of the advantages and disadvantages of local (native) and cross-platform technologies used in the development of mobile applications. The software development tools discussed in the article allow you to create independent software products installed in the necessary operating system of the device. These environments do not require special skills and experience in writing programs and at the same time allow you to create high-quality mobile applications using various programming languages. The following technologies were selected as the analyzed technologies for the development of mobile applications: native and cross-platform.

**Keywords:** mobile application, native technology, cross-platform technology, iOS, Xcode, Swift, Android, React Native, Flutter, Ionic, Xamarin, PhoneGap.