

//Импортозамещение, научно-техническая и экономическая безопасность: сборник статей V Международной научно-технической конференции "Минские научные чтения-2022", Минск, 07-09 декабря 2022 г. : в 3 т. Т.2. – Минск: БГТУ, 2022. С. 393–398.

2. Bierbrauer, J., Fridrich, J. Constructing Good Covering Codes for Applications in Steganography. In: Shi, Y.Q. (eds) Transactions on Data Hiding and Multimedia Security III. Lecture Notes in Computer Science, v. 4920. –Springer, Berlin, Heidelberg, 2008. DOI:10.1007/978-3-540-69019-1\_1.

3. Николайчук А. Н., Урбанович П. П. Стеганографический метод на основе использования особенностей отображения элементов в формате SVG // Труды БГТУ. Сер. 3, Физико-математические науки и информатика. 2023. № 1 (266). С. 64–70. DOI: 10.52065/2520-6141-2023-266-1-11.

УДК 004.4

**Н. Ю. Фрицлер, Н.А. Горбунова**

Карагандинский университет им. Е.А. Букетова  
Караганда, Казахстан

#### **РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ANDROID И IOS ПЛАТФОРМ**

*Аннотация.* В данной статье рассматриваются ключевые этапы разработки мобильного приложения для Android и iOS платформ. Уделено внимание процессу дизайна, программирования и тестирования, рассмотрены различия между нативной и кроссплатформенной разработкой, а также вопросы публикации приложений.

**N.Yu. Fritsler, N.A. Gorbunova**

Karaganda Buketov University  
Karaganda, Kazakhstan

#### **DEVELOPMENT OF A MOBILE APPLICATION FOR ANDROID AND IOS PLATFORMS**

*Abstract.* This article covers the key stages of developing a mobile application for Android and iOS platforms. It focuses on the design, programming and testing process, considers the differences between native and cross-platform development, and also discusses issues of publishing applications.

Сегодня мобильные приложения играют центральную роль в жизни пользователей по всему миру. От общения и развлечений до покупок и управления финансами - мобильные технологии стали неотъемлемой частью повседневности. Именно поэтому разработка приложений для мобильных платформ является одним из самых востребованных направлений в IT-индустрии. Разработка приложения сразу для обеих платформ позволяет охватить максимальное число пользователей и создать единый пользовательский опыт вне зависимости от типа устройства.

Целью данной статьи является предоставление структурированного руководства по разработке мобильного приложения, которое будет эффективно работать на платформах Android и iOS. Следует рассмотреть ключевые этапы разработки, особенности нативного и кроссплатформенного подходов, а также выбрать оптимальные инструменты и методы для создания качественного, удобного и безопасного мобильного продукта.

Первый и один из важнейших шагов в разработке мобильного приложения - это проведение анализа и составление плана проекта. Основой этого этапа является глубокое понимание целевой аудитории. Необходимо определить, кто будет использовать приложение, какие проблемы оно решает и какие потребности удовлетворяет. Исследование целевой аудитории помогает выявить ее возрастные, социальные и географические особенности, а также предпочтения и ожидания от будущего продукта. Такой анализ позволит создать приложение, максимально адаптированное под реальных пользователей.

Анализ конкурентов - еще один важный элемент начального этапа. Изучение уже существующих решений и их особенностей помогает понять, какие функции популярны, а какие не оправдали себя. Это не только помогает избежать ошибок конкурентов, но и выделиться за счет уникальных функций или улучшенного пользовательского интерфейса. Анализируя конкурентов, можно выявить их сильные и слабые стороны, понять, что работает, а что нет, и наметить направления для улучшений.

Следующим этапом планирования становится определение ключевых функций и пользовательских сценариев. Основные функции приложения должны отражать его уникальность и удобство, быть понятными и востребованными пользователями. Например, если создается приложение для управления задачами, ключевыми функциями могут быть добавление, редактирование, синхронизация и напоминания о задачах. Пользовательские сценарии определяют, как

люди будут взаимодействовать с приложением, и помогают сделать интерфейс интуитивным и удобным. Определение сценариев использования позволяет предугадать потенциальные проблемы, с которыми могут столкнуться пользователи, и найти пути для их решения еще на этапе планирования.

Также важно выбрать модель монетизации, так как это влияет не только на технические аспекты приложения, но и на его взаимодействие с пользователями. В зависимости от целей проекта и целевой аудитории можно выбрать модель разовых покупок, где пользователи платят только один раз за полное использование приложения, модель подписки, которая подходит для сервисов с постоянным обновлением контента, или модель с рекламой, которая позволяет предоставить бесплатный доступ, зарабатывая на просмотрах.

Выбор подхода к разработке мобильного приложения определяет, каким образом будет достигнута оптимальная производительность, доступ к функциям устройства и удобство поддержки кода. Одним из возможных подходов является нативная разработка, которая предполагает создание отдельных версий приложения для Android и iOS. Нативные приложения разрабатываются с использованием платформенных языков: Kotlin или Java для Android и Swift или Objective-C для iOS. Этот подход обеспечивает максимально точное взаимодействие с системой и доступ ко всем аппаратным и программным функциям устройства. Нативные приложения отличаются высокой производительностью, быстрой отзывчивостью и позволяют использовать все уникальные возможности каждой платформы, например, нативные интерфейсы, доступ к камере и датчикам. Однако этот подход требует разработки и поддержки двух отдельных кодовых баз, что может существенно увеличить затраты на разработку и время на запуск продукта [1].

В то же время кроссплатформенные фреймворки, такие как React Native, Flutter и Xamarin, предлагают альтернативное решение. Эти фреймворки позволяют писать код один раз и запускать его на обеих платформах, что сокращает время и затраты на разработку. React Native и Flutter стали популярными за счет высокого уровня производительности, особенно Flutter, который рендерит элементы интерфейса с использованием собственного графического движка. React Native, в свою очередь, предлагает широкую экосистему библиотек и инструментов, а также позволяет применять гибкие интерфейсы, похожие на нативные. Кроссплатформенные фреймворки упрощают поддержку приложения, поскольку изменения в одной

кодовой базе автоматически применяются для обеих платформ. Однако, несмотря на значительные преимущества, кроссплатформенные решения могут иметь ограничения по производительности и доступу к специфичным нативным функциям. Это может особенно проявляться при работе с графически насыщенными приложениями или когда требуется доступ к уникальным компонентам устройства.

Гибридный подход объединяет лучшие черты нативной и кроссплатформенной разработки, позволяя создавать ядро приложения, которое работает на обеих платформах, с возможностью добавлять нативные модули для тех частей, где требуется высокая производительность или особенные функции. Гибридный подход часто используется для сложных приложений, где важно сохранять баланс между универсальностью кода и производительностью. В этом случае кроссплатформенные части покрывают основную функциональность, а нативные компоненты позволяют внедрять уникальные функции для каждой платформы. Этот подход позволяет сократить затраты и время разработки, сохраняя при этом высокое качество работы приложения.

Разработка мобильного приложения начинается с этапа дизайна и прототипирования. На этом этапе создаются макеты экранов и UX/UI-решения, которые определяют внешний вид и удобство использования приложения. Дизайнеры прорабатывают интерфейс так, чтобы он был интуитивным и соответствовал ожиданиям пользователей, уделяя внимание мелким деталям, от расположения кнопок до выбора цветовой палитры [2].

После утверждения дизайна начинается этап разработки интерфейсов и логики приложения. Если приложение создается нативно, для Android используется язык Kotlin или Java, а для iOS — Swift или Objective-C. В кроссплатформенных приложениях, таких как те, что создаются с помощью Flutter или React Native, разработчики используют языки Dart или JavaScript соответственно. На этом этапе создаются основные экраны и их взаимосвязи, реализуется логика работы с данными, пользовательские сценарии и функциональные элементы, такие как кнопки, поля ввода, навигация. Для кроссплатформенных приложений разработка проходит в единой кодовой базе, что позволяет одновременно тестировать функциональность на обеих платформах. В то же время нативная разработка требует создания и синхронизации отдельных версий для Android и iOS, но при этом обеспечивает более высокий уровень интеграции с платформой [3].

Тестирование и отладка являются критически важными этапами,

которые обеспечивают стабильность и работоспособность приложения. Тестирование проводится как вручную, так и с помощью автоматизированных тестов, которые проверяют основные сценарии использования и отдельные функции приложения. Ручное тестирование позволяет выявить ошибки в реальных условиях использования, а автоматизация - проверять сложные или повторяющиеся процессы с минимальными затратами времени. Этот этап позволяет выявить недоработки и потенциальные проблемы, которые могли бы негативно повлиять на пользовательский опыт, а также оценить, как приложение реагирует на различные устройства и версии операционных систем.

Завершающим этапом разработки является оптимизация производительности и обеспечение безопасности. Оптимизация включает в себя работу над скоростью загрузки приложения, плавностью анимаций, временем отклика на действия пользователя и эффективным использованием ресурсов устройства, таких как память и процессор. Безопасность, в свою очередь, охватывает такие аспекты, как защита данных пользователя, шифрование передаваемой информации и предотвращение возможных уязвимостей. На этом этапе разработчики устраняют найденные проблемы, делают приложение более отзывчивым и безопасным для пользователя, что особенно важно в условиях растущего внимания к защите данных.

После завершения разработки и тестирования приложение готово к публикации в App Store и Google Play. Этот процесс включает подготовку необходимых материалов, таких как иконка, скриншоты, описание и ключевые слова, которые помогут пользователям найти приложение. Платформы предъявляют строгие требования к контенту и качеству приложений, поэтому перед публикацией проводится финальная проверка, чтобы убедиться, что все функции работают стабильно. Публикация приложения также предполагает учет правил каждой платформы и корректное заполнение форм, указание возрастных ограничений и методов монетизации. После одобрения приложение становится доступным для скачивания, и начинается этап взаимодействия с пользователями, сбора отзывов и анализа первых метрик.

Разработка мобильного приложения для Android и iOS - это сложный, но увлекательный процесс, который требует внимательного подхода на каждом этапе. От анализа целевой аудитории до выбора технологий, тестирования и публикации - все эти этапы важны для создания успешного продукта. Учитывая современные тенденции и инструменты, разработчики могут создавать приложения, которые

отвечают потребностям пользователей и выделяются на фоне конкурентов. Однако запуск приложения - это лишь начало, и для его успеха необходима постоянная поддержка, обновления и взаимодействие с пользователями. Такой подход обеспечивает долгосрочную привлекательность приложения на рынке.

#### **Список использованных источников**

1. McHugh, J. (2020). *iOS Development for Beginners: A Practical Guide to Creating iOS Apps*. San Francisco: App Development Publishing.
2. <https://habr.com/ru/articles/854988/>
3. Kottke, T. (2022). *Cross-Platform Mobile App Development: Building for Android and iOS*. Boston: O'Reilly Media.

УДК 621.396.9

**М. Халлыева, Г. Дурдыева, О. Мамедкулиева, С. Гуджанова**

Институт Телекоммуникаций и  
Информатики Туркменистана  
Ашхабад, Туркменистан

#### **ИСПОЛЬЗОВАНИЕ КОГНИТИВНОГО РАДИО ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ЧАСТОТНОГО СПЕКТРА**

***Аннотация.** В последние десятилетия наблюдается значительный рост потребления радиочастотного спектра, что приводит к его истощению. Когнитивное радио (КР) представляет собой перспективную технологию, способную повысить эффективность использования радиочастотного спектра за счет динамического доступа к неиспользуемым частотам. В данной статье рассматриваются основные принципы работы когнитивного радио, его архитектура, а также потенциальные приложения и вызовы, связанные с его внедрением.*

***Ключевые слова:** Когнитивное радио, эффективность частотного спектра, динамическое распределение спектра, спектральный доступ, распознавание радиосреды, пользователи с лицензией и без лицензии, вторичный пользователь.*

**M. Hallyyeva, G. Durdyyeva, O. Mammetkulyyeva, S. Guljanova.**

Institute of Telecommunications and  
Informatics of Turkmenistan  
Ashgabat, Turkministan