

Другой значимой тенденцией является интеграция генеративного искусственного интеллекта в системы распознавания речи. Большие языковые модели и технологии синтеза речи обеспечивают создание виртуальных ассистентов и голосовых ботов, способных к более естественному диалогу и уже применяемых в службах поддержки и цифровых помощниках [5].

### Заключение

Проведённый анализ показал, что технологии распознавания речи эволюционировали от простых акустических моделей к интеллектуальным системам, учитывающим контекст и намерения пользователя. При этом сохраняются проблемы, связанные с точностью распознавания, обработкой специализированной лексики и интерпретацией эмоциональных характеристик речи.

Дальнейшее развитие голосовых технологий обусловлено углублением интеграции искусственного интеллекта и расширением сфер их применения, при обязательном учёте пользовательского опыта, этических требований и профессиональной специфики.

### Список литературы

1. Шматов В. Н., Бонковски П., Медведев Д. С., Корзунин С. В., Голендухин Д. В., Спыну С. Ф., Муромцев Д. И. Взаимодействие с устройствами интернета вещей с использованием голосового интерфейса // Научно-технический вестник информационных технологий, механики и оптики. 2019. Т. 19, № 4. С. 714–721. DOI: 10.17586/2226-1494-2019-19-4-714-721.
2. Баляба Я. В., Рычка О. В. Разработка приложения для распознавания речи // Информатика и кибернетика. 2023. № 1 (31). С. 5–11. EDN CMCNTY.
3. Нурмухаметов Р. К., Торин С. С. Цифровое доверие (digital trust): сущность и меры по его повышению // Известия Тульского государственного университета. Экономические и юридические науки. 2020. № 1. С. 32–39. EDN IDNRCS.
4. Шерман В. А. Анализ существующих систем управления содержанием // Студенческий научный форум: материалы XVIII Международной студенческой научной конференции (Москва, 20 октября 2024 г. – 15 марта 2025 г.). М.: ЕНПП, 2025. С. 117–119. EDN MOGAVJ.
5. Чепурной М. П., Дяченко А. С. Сравнительный анализ фреймворков для фронтенд-разработки // Международный студенческий научный вестник. 2025. № 1. С. 12. EDN COGVEA.

## СРАВНИТЕЛЬНЫЙ АНАЛИЗ НАТИВНОЙ И КРОССПЛАТФОРМЕННОЙ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Гринин А. О.

ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,  
e-mail: anna.dyachenko597@mail.ru

Научный руководитель: Дяченко А.С.

В современной индустрии мобильных приложений существуют разные программные платформы, но на рынке уже долгое время

остаются две доминирующие операционные системы – Android и iOS. Наличие нескольких операционных сред ставит перед инженерами проблему выбора оптимальной технологической стека. На сегодняшний день сформировались два основных вектора разработки мобильных приложений. Первый – это классический нативный подход, требующий создания и поддержки уникальной кодовой базы для каждой операционной системы. Второй – кроссплатформенный, позволяющий использовать единый программный код для разных платформ. Существенные различия в архитектуре, стоимости и производительности этих методов обуславливают необходимость их детального изучения и сравнительного анализа [1].

### Материалы и методы исследования

Актуальные научные работы в области мобильной разработки демонстрируют смещение исследовательского фокуса с чисто технических параметров производительности на комплексную оценку жизненного цикла программного продукта. Согласно выводам авторов [1], нативный подход остается приоритетным для проектов, требующих низкоуровневого управления ресурсами, в то время как кроссплатформенные решения позволяют значительно оптимизировать процесс разработки. Современные фреймворки, такие как Flutter и React Native, рассматриваются как эффективный инструмент, предлагающий баланс между кроссплатформенной совместимостью и скоростью вывода продукта на рынок [2].

Целью исследования является проведение сравнительного анализа нативного и кроссплатформенного подходов для определения наиболее эффективной стратегии разработки мобильных приложений в условиях дуополии платформ Android и iOS, учитывая баланс между качеством продукта и затратами на его производство.

### Результаты исследования и их обсуждение

В современную технологическую эпоху смартфоны стали наиболее распространенными электронными устройствами, интегрированными практически во все сферы человеческой деятельности. Современный смартфон характеризуется не только его базовыми коммуникационными возможностями (звонками и сообщениями), но и расширенным функционалом: наличием систем глобального позиционирования (GPS), модулей биометрической аутентификации, высокотехнологичных камер и развитых операционных систем, обеспечивающих работу сложного программного обеспечения. Согласно статистическим данным на 2025 год, доминирующими платформами остаются Android с долей рынка 71,9% и iOS, занимающая 27,67% [3].

## Сравнительный анализ подходов к разработке мобильных приложений

Критерий	Нативный подход	Кроссплатформенный подход
Языки программирования	Swift, Objective-C (iOS); Kotlin, Java (Android)	Dart (Flutter), JavaScript (React Native), C# (.NET MAUI), Kotlin (KMP)
Производительность	Прямой доступ к ресурсам процессора и GPU	Затраты ресурсов на слой абстракции
Доступ к API и железу	Полный доступ ко всем датчикам и системным функциям	Поддержка новых функций ОС внедряется с задержкой
Пользовательский интерфейс	Идеально плавный	Высокая схожесть, но возможны трудности с кастомной анимацией и специфическими жестами
Сложность поддержки	Необходимо обновлять и тестировать два независимых приложения	Изменения вносятся в один проект и применяются ко всем платформам сразу
Объем приложения	Минимальный (нет лишних библиотек и движков)	Значительный (из-за встроенного веса кроссплатформенного движка)
Безопасность	Максимальная за счет встроенных средств защиты конкретной ОС	Зависит от безопасности стороннего фреймворка
Целевой сегмент	Высоконагруженные системы, приложения со сложной графикой	MVP, корпоративные приложения

При наличии двух основных операционных систем имеет смысл ориентироваться на обе платформы при разработке приложения, предназначенного практически для всех пользователей.

Существуют два основных подхода к созданию мобильных приложений: нативная и кроссплатформенная. Нативная разработка подразумевает использование языков и инструментов, которые созданы непосредственно для конкретной операционной системы. Так, для Android это Java или Kotlin, а для iOS – Objective-C или Swift. Этот подход позволяет приложению напрямую обращаться к функциям смартфона, обеспечивая максимальную скорость работы, но требует написания отдельного кода для каждой платформы. В случае кроссплатформенного решения разработчики используют единый исходный код, который запускается на разных системах благодаря специальному промежуточному слою – абстракции. Этот слой переводит универсальные команды в понятный для конкретной ОС вид, что значительно экономит ресурсы при разработке. Среди наиболее известных инструментов такого типа выделяются: React Native, Flutter, .NET MAUI и Kotlin Multiplatform [4]. Подробное сравнение рассматриваемых подходов представлено в таблице.

Для масштабных и долгосрочных проектов приоритетным остаётся нативный подход к разработке. Он обеспечивает полное соответствие требованиям конкретной платформы, доступ к её аппаратным ресурсам и специализированным API, а также высокую производительность и качество пользовательского опыта. Приложения со сложной логикой, интерфейсом и анимацией требуют стабильности и масштабируемости, которые в полной мере реализуются

именно в нативной разработке [5]. Опыт крупных компаний показывает, что попытки перехода на кроссплатформенные решения нередко заканчиваются возвратом к нативным технологиям из-за ограничений по производительности и сложности масштабирования.

Кроссплатформенные технологии целесообразны для создания прототипов, пилотных решений и простых приложений со стандартизированным функционалом, где скорость разработки и снижение затрат являются приоритетом. Согласно данным Statista, около трети мобильных разработчиков используют кроссплатформенные фреймворки, однако большинство специалистов по-прежнему отдают предпочтение нативным технологиям, особенно при реализации сложных и ресурсоёмких проектов [6]. Таким образом, анализ существующих практик демонстрирует, что выбор подхода должен определяться не модными тенденциями, а целями проекта, его масштабом и требованиями к качеству конечного продукта, что создаёт основу для дальнейших обобщений и выводов.

### Заключение

Проведённый анализ показал, что выбор между нативной и кроссплатформенной разработкой зависит от целей проекта. Нативный подход обеспечивает максимальную производительность, безопасность и качество пользовательского опыта и оптимален для ресурсоёмких и масштабируемых приложений. Кроссплатформенная разработка эффективна для быстрого запуска и снижения затрат, но имеет ограничения по производительности. Оптимальный выбор определяется балансом между качеством и скоростью разработки.

**Список литературы**

1. Кодитувака С., Пасиллоб Н. Разработка мобильного программного обеспечения в цифровую эпоху: сравнительная оценка кроссплатформенных фреймворков // JPO. 2025. Т. 8, № 2. С. 9–17.
2. Чепурной М. П., Дяченко А. С. Сравнительный анализ фреймворков для фронтенд-разработки // Международный студенческий научный вестник. 2025. № 1. С. 12. EDN COGVEA.
3. Mobile Operating System Market Share Worldwide // Statcounter Global Stats. [Электронный ресурс]. URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide> (дата обращения: 25.12.2025).
4. Kotlin Multiplatform for cross-platform mobile development // Kotlin Documentation. [Электронный ресурс]. URL: <https://kotlinlang.org/docs/multiplatform/cross-platform-mobile-development.html> (дата обращения: 27.12.2025).
5. Дяченко А. С. Использование динамических элементов в дизайне сайтов для улучшения пользовательского опыта // Материалы XXV Всероссийской научно-практической конференции молодых ученых, аспирантов и студентов с международным участием, посвященной 80-летию Победы в Великой Отечественной войне 1941–1945 гг. (Нерюнгри, 23–25 октября 2025 г.). Якутск: Северо-Восточный федеральный университет им. М. К. Аммосова, 2025. С. 352–357. EDN ОКРПЖК.
6. Statista. Worldwide software developer working hours [Электронный ресурс]. URL: <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/> (дата обращения: 26.12.2025).

**АСПЕКТЫ ИСПОЛЬЗОВАНИЯ  
МУЛЬТИМЕДИЙНЫХ  
ТЕХНОЛОГИЙ В ИСКУССТВЕ**

Гриценко М. С., Окулова Е. А.

*ФГБОУ ВО «Мелитопольский государственный университет», Мелитополь,  
e-mail: mmaria.gritsenko@yandex.ru*

*Научный руководитель: Окулова Е. А.*

**Введение**

В современном мире культурное пространство характеризуется неразрывным слиянием искусства и технологий, что дает повод для образования новейших форм художественного выражения. Мультимедиа становится частью художественного прогресса и включает в себя цифровые среды, программные платформы, НТТР. Классические формы искусства основаны на линейных и статичных носителях, например, живопись, скульптура, традиционный кинематограф. А мультимедийные технологии обладают динамикой и интерактивной природой, что кардинально меняет взаимосвязь между автором и зрителем. Актуальность нашего исследования обусловлена бурным развитием технологий и их внедрением в сферу искусства. В последние годы произошел определенный прорыв в области программного обеспечения и инфраструктуры интернет, что позволило авторам и художникам создавать новые формы художественного выражения. В сетевом искусстве применение интернета как платформы и поиска материала, компьютерная анимация, достигшая невероятного уровня реалистичности, а также

передовые технологии видеомонтажа – все это создает новую художественную эпоху, в которой размываются границы между дисциплинами и медиа. В связи с этим необходимо провести исследование применения мультимедийных технологий в искусстве, которые затрагивают не только технические, так и эстетические, философские и культурные измерения. Исключительно качественный подход даст описать не только современные тенденции, а и спрогнозировать будущее развитие цифровых художественных практик.

**Анализ последних исследований  
и публикаций**

Исследования цифрового искусства чаще опираются на то, как технологии меняют художественные формы. Зачастую упуская при этом из виду более глубокий смысл между понятием об авторстве технической части, самой эстетике в целом. В публикациях об net art рассматривается критика цифровой среды. Многие статьи описывают яркие проекты, при этом нет объяснения как такие формы влияют на эстетическую и даже политические возможности сетевого искусства. В сфере анимации отечественные исследователи акцентируют внимание на коммерции. И редко затрагивают тему как игровые движки и программы с открытым доступом влияют на новое восприятие анимации в целом. Что касается видеомонтажа, не смотря на активные программные обеспечения крайне мало уделяется внимание созданию реалистичной графики и анимации. И все чаще приходим к коротким горизонтальным видео, которые построены не на основах кинематографа. Одним из сложных вопросов остается искусственный интеллект. Нейросеть участвует в создании картин, дизайна и музыки. Известные художники применяют инструменты ИИ, а известные дизайнеры экспериментируют. Крупные компании заключают договоры для расширения возможностей, при этом сохраняя свои права на уникальность. При всем этом нет четкого ответа об авторстве.

Цель исследования – системный анализ аспектов использования мультимедийных технологий в современном искусстве на примере четырех ключевых направлений: сетевого искусства, компьютерной анимации, видеомонтажа, актуальное развитие искусства и нейросети. Особое внимание уделяется трансформации художественных форм, эволюции зрительского восприятия и влиянию цифровой среды на креативные процессы.

**Материал и методы исследования**

В ходе исследования использовались следующие методы:

- изучение фундаментальных трудов по теории медиа, цифровой эстетике и современным