

Плюсы и минусы мобильной разработки на Xamarin

10.02.2023

Рассматривая разработку приложений для iOS или Android, большинство из нас в первую очередь думают о [Objective-C против Swift](#) и Java. Будучи нативными стеками технологий, они, естественно, являются наиболее часто используемыми инструментами мобильной разработки, когда дело доходит до разработки приложений для iOS и Android. Однако есть и другие способы создания высокопроизводительных и удобных для пользователя мобильных приложений. И Xamarin – один из них. В этой статье мы проводим углубленный обзор Xamarin и сравниваем его с нативными и кроссплатформенными решениями для мобильной разработки, представленными на рынке. Если вы знаете основы, подумайте о том, чтобы пропустить интересующие вас разделы.

Что такое Xamarin?

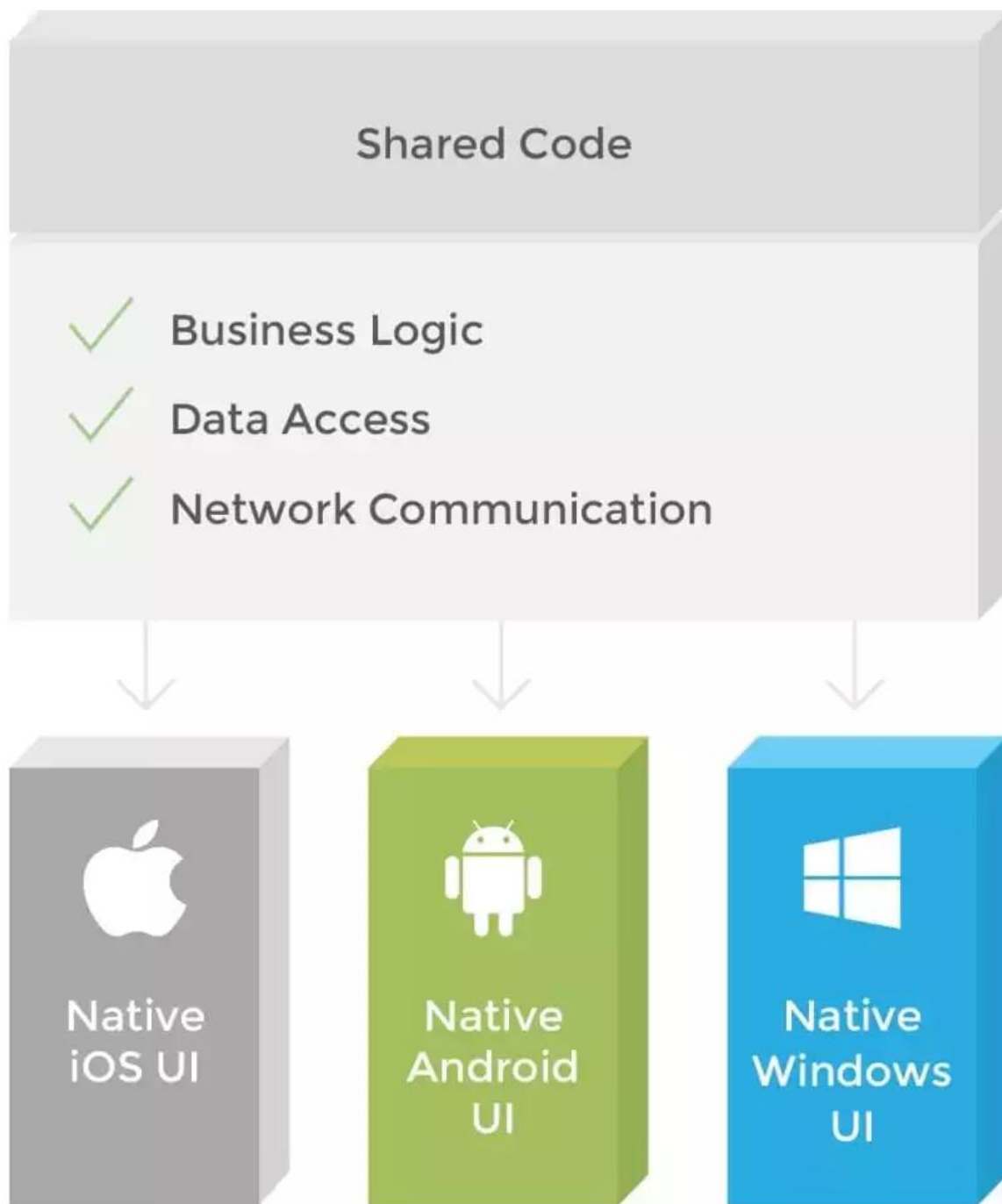
Невозможно игнорировать Xamarin, когда речь идет о [ключевых подходах к разработке мобильных приложений](#). Xamarin – это инструмент, используемый для кросс-платформенной разработки мобильных приложений, который позволяет инженерам совместно использовать до 90 процентов кода на основных платформах. Будучи сравнительно новым инструментом, он основан на стеке технологий Microsoft и уже насчитывает сообщество из более чем 1,4 миллиона разработчиков.

Платформа была создана разработчиками [Mono](#), платформы разработки с открытым исходным кодом, основанной на .NET Framework, во главе с Мигелем де Икасой и впервые представленной в 2001 году. Компания Xamarin была основана 16 мая 2011 года. Однако, в отличие от своего предшественника, Xamarin создавался как коммерческий проект, пока компания не была приобретена Microsoft в 2016 году. Xamarin стал

популярным кроссплатформенным продуктом для разработки мобильных приложений в экосистеме Microsoft. Это приобретение преодолело финансовый барьер для использования Xamarin. Поскольку Microsoft сделала Xamarin SDK открытым исходным кодом, он стал частью интегрированной среды разработки Xamarin Visual Studio ([ниже](#) мы обсудим лучшие IDE для разработки Xamarin). Каковы основные возможности Xamarin?

Язык C #. Xamarin использует один язык, C #, для создания приложений для всех мобильных платформ. В отличие от интерпретируемых решений, таких как Appcelerator Titanium, Xamarin изначально скомпилирован, что делает его универсальным вариантом для создания высокопроизводительных приложений с собственным внешним видом.

Основан на .NET framework. C # – это зрелый язык с надежной типизацией, которая предотвращает неожиданное поведение кода. Поскольку C # является одним из .NET Framework, его можно использовать с рядом полезных .NET функции, такие как лямбды, LINQ и асинхронное программирование (async / await). Технически говоря, Xamarin использует C # и собственные библиотеки, завернутые в .NET уровень для кросс-платформенной разработки приложений. Такие приложения часто сравнивают с нативными для мобильных платформ разработки iOS и Android с точки зрения производительности и удобства использования. Кроме того, Xamarin может использовать весь собственный и новейший доступ к API для использования базовых возможностей платформы в приложениях Xamarin, таких как [ARKit](#) на iOS или [Android Multi-Window](#).



Хотя код, относящийся к бизнес-логике, доступу к базе данных и сетевому взаимодействию, может использоваться совместно на всех платформах, Xamarin позволяет создавать уровень кода пользовательского интерфейса, зависящий от платформы. Таким образом, кроссплатформенные приложения Xamarin выглядят на 100% нативными на любом устройстве, обеспечивая лучший пользовательский интерфейс по сравнению с обычными гибридными приложениями.

Сборник. Платформа состоит из двух основных продуктов:

Xamarin.iOS и Xamarin.Android. В случае iOS исходный код компилируется непосредственно в собственный код сборки ARM (компиляция с опережением времени), в то время как приложения Android Xamarin сначала компилируются до промежуточного языка, а затем – в собственный код сборки во время выполнения (компиляция точно в срок). Однако в обоих случаях процесс автоматизирован и адаптирован для решения таких проблем, как выделение памяти, сборка мусора и совместимость платформы по умолчанию.

Xamarin.Формы. Xamarin.Forms – это отдельный продукт, предназначенный для создания прототипов или мобильных приложений, использующих до 100 процентов кода для iOS и Android.

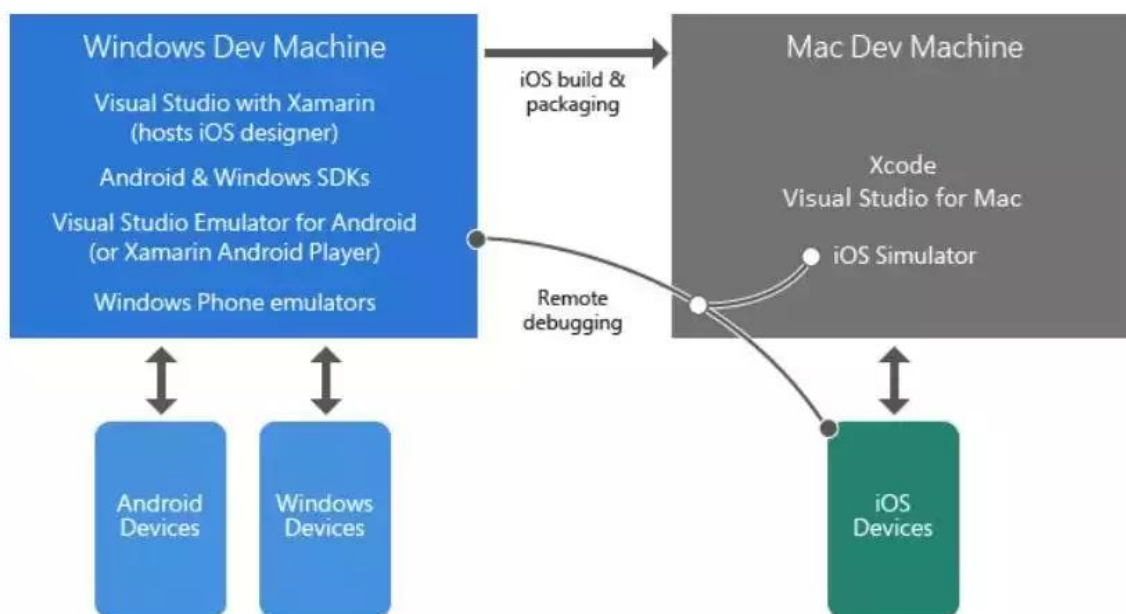
Xamarin.Mac. Кроме того, инженеры могут использовать Xamarin.Mac для создания приложений для Mac OS. Эти и некоторые другие продукты из стека Xamarin будут рассмотрены немного позже.

Пакеты SDK для платформы. Xamarin предоставляет доступ к специфичным для платформы SDK (Cocoa Touch для iOS и Android SDK) с помощью простого синтаксиса C #.

Привязки кода к конкретной платформе. В большинстве случаев инженеры смогут справиться с разработкой, сохраняя собственный внешний вид соответствующих платформ в среде Xamarin. Однако Xamarin позволяет вызывать существующий код платформы (например, Swift для iOS), если необходимо повторно использовать некоторые модули или реализовать некоторые функции, зависящие от платформы, такие как сканирование штрих-кодов, интеграция с PayPal или Google Analytics.

Поддержка носимых устройств. Xamarin поддерживает создание приложений для устройств Apple и Android Watch. Инженеры могут использовать Visual Studio как для Mac, так и для Windows для создания приложений Apple Watch для iOS. Кроме того, Xamarin полностью поддерживает Android Wear.

Структура проекта Xamarin. Ожидается, что большая часть работы, связанной с Xamarin, будет выполняться с помощью компьютера для разработки Windows с установленными Visual Studio и Xamarin. Приложения можно отлаживать прямо с рабочего стола или на устройствах и эмуляторах. Если вы планируете разрабатывать приложения для iOS в Windows, это также возможно, поскольку Visual Studio подключается к конструктору раскладки iOS и симулятору iOS. Существует также Visual Studio для Mac, которая позволяет запускать симулятор на Mac или непосредственно на привязанном iPhone. Отладка выполняется через Visual Studio.



Структура проекта Xamarin. Источник изображения: [блог поддержки разработчиков](#)

Как и у любой другой технологии, у Xamarin есть свои преимущества и недостатки, которые в последнее время стали предметом жарких дискуссий. Чтобы принять обоснованное решение о Xamarin против собственной разработки, давайте взвесим все плюсы и минусы разработки приложений с помощью Xamarin. Если у вас действительно мало времени, мы рекомендуем посмотреть это видео, в котором менее чем за 6 минут дается подробный обзор сильных и слабых сторон платформы Xamarin.

Плюсы использования Xamarin для разработки

Существуют веские причины, по которым [Xamarin используется многими компаниями](#), включая Trello, Slack и GitHub.

⊕ Единый технологический стек для кода для всех платформ

Xamarin использует C #, дополненный .Net framework для создания приложений для любой мобильной платформы. Таким образом, вы можете совместно использовать в среднем [80-90 процентов](#) кода на разных платформах, ускоряя цикл разработки. Xamarin также не требует переключения между средами разработки: вы можете создавать все свои приложения Xamarin в Visual Studio или Visual Studio Code. Кроссплатформенные средства разработки предоставляются как встроенная часть IDE без каких-либо дополнительных затрат.

⊕ Производительность близка к родной

В отличие от традиционных гибридных решений, основанных на веб-технологиях, кроссплатформенное приложение, созданное с помощью Xamarin, все еще может быть классифицировано как собственное. По мере развития платформы производительность Xamarin постоянно повышается, чтобы полностью соответствовать стандартам собственной разработки, при этом Microsoft дает [рекомендации](#) по методам оптимизации. Visual Studio также предлагает комплексное решение для создания, тестирования и отслеживания производительности приложения: [Visual Studio App Center](#) позволяет запускать автоматические тесты пользовательского интерфейса и выявлять проблемы перед выпуском. Однако эта услуга предоставляется за дополнительную плату. Ранее мы тестировали и сравнивали [производительность Xamarin](#) и собственных приложений. Результаты могут быть несколько устаревшими; однако они ясно демонстрируют достойный

уровень производительности приложений Xamarin.

⊕ Собственный пользовательский интерфейс

Xamarin позволяет создавать безупречные возможности, используя элементы пользовательского интерфейса, зависящие от платформы. С помощью Xamarin также можно создавать кроссплатформенные приложения для iOS, Android или Windows. Инструмент Forms, который преобразует компоненты пользовательского интерфейса приложения в элементы интерфейса, зависящие от платформы, во время выполнения. Как использовать Xamarin.Forms значительно увеличивает скорость разработки приложений, это отличный вариант для бизнес-ориентированных проектов. Тем не менее, может произойти небольшое снижение производительности из-за дополнительного уровня абстракции. Для пользовательского интерфейса приложения и повышения производительности вы все равно можете использовать Xamarin.iOS и Xamarin.Android отдельно для обеспечения отличных результатов. Традиционно считается, что Xamarin – не лучший вариант для приложений с тяжелой графикой. Тем не менее, существует ряд библиотек, таких как [OpenGL ES](#), и фреймворков, таких как [SkiaSharp](#) и [Urho](#), которые позволяют разрабатывать приложения с богатым пользовательским интерфейсом, сложными анимациями или играми даже в Xamarin.

⊕ Полная аппаратная поддержка

С помощью Xamarin ваше решение получает функциональность приложения собственного уровня. Он устраняет все проблемы с совместимостью оборудования, используя плагины и специальные API для работы с общими функциями устройств на разных платформах. Наряду с доступом к API, зависящим от платформы, Xamarin поддерживает связь с собственными библиотеками. Это позволяет улучшить настройку и функциональность на собственном уровне с минимальными затратами.

⊕ **Технология с открытым исходным кодом и мощной корпоративной поддержкой**

После того, как компания была приобретена Microsoft в феврале 2016 года, политика лицензирования Xamarin претерпела радикальные изменения. Xamarin SDK, включающий среду выполнения, библиотеки и инструменты командной строки, стал открытым исходным кодом и доступен для всех по лицензии MIT как часть Visual Studio. Устранив главный барьер – стоимость лицензии, Microsoft проложила путь для дальнейшего роста платформы. Кроме того, .NET Foundation, возглавляемая Microsoft и поддерживаемая Unity, JetBrains и Red Hat, в целом и Xamarin в частности, стала надежным и мощным технологическим стекком.

⊕ **Упрощенное обслуживание**

Благодаря кросс-платформенному приложению Xamarin упрощает обслуживание и обновления. Вы можете просто развернуть изменения или обновления в исходном файле, и они будут применены как к приложениям для iOS, так и к Android. Таким образом, это помогает вам экономить время и деньги, сохраняя ваши приложения в актуальном состоянии.

⊕ **Полная экосистема разработки**

Xamarin поставляется в одном пакете с полным набором инструментов разработки, включая собственную среду разработки. В апреле 2018 года множество инструментов разработки Xamarin, ранее известных как отдельные функции, стали частью [Центра приложений Visual Studio](#). В комплект входит сама платформа (Xamarin SDKs), тестирование (ранее Xamarin Test Cloud), распространение и аналитика (ранее Xamarin.Понимание) платформ и многое другое. Поскольку инструментарий остается бесплатным, нет необходимости вкладывать средства в дополнительные инструменты или интегрировать сторонние приложения для создания, тестирования

и развертывания приложений Xamarin.

⊕ **Xamarin.Формы: основа для создания собственных макетов пользовательского интерфейса**

Xamarin предлагает разработчикам создавать мобильные приложения двумя способами. Первый – использовать Xamarin.iOS и Xamarin.Android, которые появились раньше и изначально считались более мощными. Использование [Xamarin.Формы](#) – это второй подход. Это более продвинутая версия, которая позволяет быстро создавать прототипы или создавать приложения с менее специфичной для платформы функциональностью. Вот почему Xamarin.Forms лучше всего подходит для приложений, где совместное использование кода важнее пользовательского интерфейса. Это позволит вам повторно использовать до 99% кода, что означает, что вы сможете создать единый интерфейс и использовать его на разных платформах, не разрабатывая для каждой платформы по отдельности. Вы также можете создавать приложения, в которых некоторые части их пользовательских интерфейсов созданы с помощью Xamarin.Формы и другие разрабатываются с использованием инструментария native UI. Как Xamarin.Формы, разрабатываемые годами, стали универсальным инструментом и приобрели активное сообщество, которое уделяет ему внимание. Например, некоторые поставщики технологий предоставляют свои библиотеки, такие как [Telerik UI](#), которые расширяют технические возможности, предоставляя предварительно настроенные компоненты пользовательского интерфейса.

⊕ **Xamarin.Горячая перезагрузка форм XAML**

Как Xamarin.Forms нацелен на быструю и беспроблемную разработку, Microsoft предлагает использовать инструмент [горячей перезагрузки XAML](#). Он ускоряет разработку, позволяя изменять XAML во время отладки и сразу же видеть эти изменения в запущенном приложении, не требуя его остановки и перестройки. Кроме того, состояние навигации и данные будут

сохранены, чтобы вы могли быстро выполнять итерации в своем пользовательском интерфейсе. Таким образом, горячая перезагрузка XAML позволяет быстрее перестраивать и развертывать приложения для проверки изменений пользовательского интерфейса.

⊕ Создание приложений для Mac с помощью Xamarin.Инструмент для Mac

Xamarin.Mac позволяет разрабатывать полностью собственные приложения для Mac с использованием C # и .NET. Он интегрируется с теми же библиотеками, которые используются для разработки в Objective-C с помощью Xcode. С помощью Xamarin.Mac в сочетании с Xamarin.iOS и Xamarin.Android, разработчики могут совместно использовать до 90 процентов собственного кода для Android, iOS и Mac OS. Кроме того, Xamarin.Mac интегрируется с Xcode, поэтому разработчик может использовать конструктор интерфейсов Xcode для создания пользовательских интерфейсов приложения.

⊕ Учебные ресурсы Xamarin

[Microsoft Learn](#) – это онлайн-образовательный портал, который помогает людям развивать технические навыки, связанные с продуктами Microsoft, включая [Xamarin](#). Платформа предоставляет курсы как для начинающих, так и для опытных инженеров, позволяя учащимся подготовиться к сертификации Microsoft. Образовательные ресурсы включают бесплатное самостоятельное обучение, онлайн-занятия в режиме реального времени, лекции в офисе и гостевые лекции.

⊕ Поддержка Xamarin телевизоров, носимых устройств и Интернета вещей

Платформа Xamarin, благодаря своему Xamarin.Формирует, позволяет создавать переносимые версии для многих платформ. В дополнение к iOS, Android, Windows Phones и Mac вы можете

разрабатывать приложения для [Tizen](#) – операционной системы, используемой на некоторых устройствах Samsung, – для носимых устройств, телевизоров и Интернета вещей. Кроме того, Xamarin.iOS позволяет создавать приложения для [tvOS](#) для [Apple](#) и [watchOS](#).

Минусы Xamarin, которые следует учитывать

Тем не менее, все еще есть некоторые недостатки, которые могут бросить тень на платформу. Итак, почему бы не использовать Xamarin?

⊖ Немного отложена поддержка последних обновлений платформы

Это полностью зависит от команды разработчиков Xamarin. Сторонние инструменты не могут обеспечить немедленную поддержку последних версий iOS и Android, поскольку требуется некоторое время для внесения изменений и / или внедрения новых плагинов и т. Д.

⊖ Ограниченный доступ к библиотекам с открытым исходным кодом

В собственной разработке широко используются технологии с открытым исходным кодом. С помощью Xamarin вы должны использовать элементы, предоставляемые платформой, и некоторые другие ЧИСТЫЕ ресурсы с открытым исходным кодом, доступные как [разработчикам](#), так и [потребителям](#). Хотя выбор не так богат, как для [разработки мобильных приложений для Android и iOS](#), вы можете использовать пакеты [NuGet](#), которые теперь содержат все, что было в устаревших компонентах Xamarin.

0 Высокая стоимость для профессионального и корпоративного использования

Хотя Xamarin является бесплатной платформой с открытым исходным кодом для индивидуальных разработчиков, для корпоративных нужд эта платформа может стоить немалых денег. Вы потратите много денег на покупку лицензии для Visual Studio. Например, [Visual Studio Professional](#) с базовым набором инструментов разработки стоит 1199 долларов США за первый год и 799 долларов США за продление. В то же время [Visual Studio Enterprise](#), которая позволяет экспериментировать с дополнительными инструментами, будет стоить 5 999 долларов США за первый год и 2569 долларов США за продление. Функции для обоих мы рассмотрим ниже.

0 Кадровый резерв Xamarin и ограничения сообщества

Очевидно, что сообщество Xamarin значительно меньше, чем сообщество iOS или Android. Он также меньше, чем сообщества React и Ionic. Таким образом, найти опытного разработчика Xamarin может оказаться непростой задачей. Согласно [опросу Stack Overflow 2020](#) года, только 5,8% использовали Xamarin framework по состоянию на 2020 год, по сравнению с 11,5% у React Native и 7,2% у Flutter.



Использование Xamarin по сравнению с другими фреймворками. Источник изображения: [переполнение стека](#)

Несмотря на то, что количество инженеров Xamarin не идет ни в какое сравнение с сообществами разработчиков iOS или Android, платформа обеспечивает всестороннюю поддержку своим разработчикам. Как упоминалось выше, Microsoft Learn предлагает ресурсы и практическое обучение для тех, кто не знаком с этой технологией. При использовании этой поддержки процесс обучения для опытного инженера C # / .Net минимален.

Еще одной проблемой, с которой сталкиваются некоторые разработчики, является фрагментация и отсутствие стандартов в ресурсах для обучения и разработки, включая документы. Это влияет на качество поддержки, которую вы собираетесь получить.

0 Базовые знания родных языков

При использовании Xamarin.iOS и Xamarin.Android для создания мобильных приложений с действительно нативным внешним видом вам все равно нужно будет написать слой кода, зависящий от платформы, используя собственные API каждой платформы. Итак, хотя это и не обязательно, как раньше, было бы полезно иметь хотя бы базовые знания о собственных технологиях (Java / Kotlin для Android и Objective-C / Swift для iOS), чтобы иметь возможность понимать и извлекать максимальную пользу из специфичных для платформы документация. Кроме того, [Xamarin.Библиотека Essentials](#) предоставляет несколько кроссплатформенных API и может использоваться для доступа к собственным функциям. Это, однако, не относится к Xamarin.Формы.

0 Большой размер приложения

В зависимости от их типа и сложности приложения Xamarin обычно больше, чем собственные (последние могут быть вдвое меньше приложения Xamarin). Простое приложение “привет, мир” для Android может занимать до 16 МБ, большая часть которого используется связанными библиотеками, контентом, Mono runtime и сборками библиотеки базовых классов (BCL).



Размер приложения Xamarin. Источник изображения: developer.xamarin.com

Таким образом, приложения Xamarin обычно нуждаются

в [дополнительной оптимизации](#), такой как создание ссылок для удаления неиспользуемого кода из включенных библиотек и сохранения разумного размера файла приложения.

0 Xamarin.Формы скоро перестанут существовать

В мае 2020 года состоялась ежегодная конференция [Microsoft Build](#). Среди других новостей было объявлено, что Xamarin.Forms собирается объединиться с .NET и эволюционирует в новую кроссплатформенную платформу, называемую .NET [Multi-platform App UI](#), или MAUI. MAUI будет иметь весь собственный пользовательский интерфейс и предоставлять доступ ко всем собственным API. Кроме того, в рамках постепенного .NET унификация, Xamarin.Android и Xamarin.iOS в конечном итоге станут частью .NET 6 как .NET для iOS и .NET для Android и будут обновлены в соответствии с другими рабочими нагрузками .NET. Сроки для этого пока неизвестны, так что следите за обновлениями в будущем. Между тем, Xamarin.Миграция форм должна состояться в ноябре 2021 года. После выпуска MAUI Xamarin будет поддерживаться еще 12 месяцев. Итак, если в данный момент вы работаете с Xamarin или планируете создавать новые приложения Xamarin, стоит следить за датами и искать руководства Microsoft для плавного перехода.

0 Проблемы совместимости со сторонними библиотеками и инструментами

При интеграции сторонних ресурсов с вашим приложением Xamarin могут возникнуть некоторые проблемы. Хотя большинство инструментов и библиотек предлагают полную поддержку собственных технологий, поддержка Xamarin может не предоставляться поставщиком. Хотя у Xamarin есть собственное хранилище компонентов, всегда есть вероятность, что вам понадобятся определенные возможности или интеграция в вашем приложении, которые не предоставляются платформой. Итак, чтобы использовать другие сторонние пакеты, вы должны использовать

оболочки.

Xamarin Visual Studio IDE

Если у вас есть ощущение, что Xamarin подходит именно вам, вам будет интересно узнать основные IDE Xamarin, которые можно использовать, и как они сравниваются друг с другом.

Visual Studio (VS) – это интегрированная среда разработки Microsoft, или IDE, программное обеспечение, используемое для создания, отладки и публикации приложений на всех платформах и устройствах. Существует три основные версии Visual Studio для Xamarin – Visual Studio Community, Visual Studio Professional и Visual Studio Enterprise. Ключевыми факторами для выбора приемлемой версии являются опыт разработчиков (студентов или профессиональных инженеров) и размер их команды. Программисты могут разрабатывать программное обеспечение для веб, мобильных устройств, серверов и настольных компьютеров со всеми версиями. Узнайте больше о Visual Studio в нашей статье об [инструментах, используемых для разработки .NET](#).

Последняя версия – **Visual Studio 2019**. По сравнению с VS 2017, он содержит многочисленные улучшения, которые ускоряют разработку приложений, от совместной работы в режиме реального времени с использованием [Visual Studio Live Share](#) до нового окна запуска Visual Studio. Кроме того, он поддерживает разрешение на каждый монитор и улучшенные возможности отладки.

Рабочая нагрузка Xamarin теперь уменьшена примерно в 3 раза по сравнению с предыдущими версиями, что означает сокращение с 23 ГБ до 7 ГБ. Это значительно снижает нагрузку на диск и ускоряет установку. Кроме того, в новой версии были внесены значительные улучшения в управление Android SDK и эмуляторами. Он также приложил значительные усилия для улучшения разработки пользовательского интерфейса. Это включает в себя Xamarin.Улучшения форм благодаря IntelliCode для XAML, улучшениям AXML для Android, обновленной доставке Xamarin для iOS и многим другим функциям.

Сообщество Visual Studio хорошо подходит для студентов, проектов с открытым исходным кодом, небольших групп разработчиков до 5 пользователей. Пакет сообщества также хорош для команд, которые являются новичками в Visual Studio. Эта версия бесплатна для использования.

Visual Studio Professional – это набор инструментов и сервисов, созданных для профессиональных разработчиков, небольших команд и корпоративного использования. Эта версия предоставляет мощные функции для быстрого понимания вашего кода: отображение ссылок на код, изменений в коде, отображение того, кто последним модифицировал метод, или обнаружение прохождения тестов. Профессиональная версия включает [Azure DevOps](#), которая представляет собой набор функций, таких как гибкие инструменты планирования и тестирования проектов, [конвейеры CI / CD](#), решения для создания отчетов и многое другое. Годовая цена этой версии составляет 1199 долларов.

Visual Studio Enterprise имеет более [мощные инструменты и службы поддержки](#). Они позволяют управлять более сложными процессами между разработкой и ИТ-операциями; отслеживать качество с помощью ручных и автоматических инструментов тестирования; получать доступ к профессиональному обучению и всей экосистеме разработки и тестирования Microsoft. Корпоративная версия также имеет приоритетную поддержку, предложения партнеров и многое другое. Годовая цена Enterprise Studio составляет 5 999 долларов.

Для корпоративного использования мы также рекомендуем проверить [Центр приложений Visual Studio](#). Это инструмент, который организует рабочий процесс разработки приложений с использованием различных мобильных технологий, включая Xamarin. Здесь команды инженеров, работающие над несколькими приложениями, могут объединить свой набор инструментов. Он автоматизирует упаковку приложений с помощью GitHub или Bitbucket, обеспечивает интеграцию для различных служб тестирования, позволяет рассылать приложения по электронной

почте для тестирования, отслеживает сбои и анализирует.

Visual Studio Code, или [VS Code](#), была признана самой популярной средой разработки в 2019 году: [50,7%](#) разработчиков сообщили, что используют ее. VS Code – это бесплатный, легкий редактор кода с открытым исходным кодом, который предоставляет все основные инструменты, необходимые для простого цикла редактирования, сборки и отладки кода, но ему не хватает возможностей полнофункциональных IDE, позволяющих управлять более сложными рабочими процессами. Он работает на Mac, Linux и Windows и поддерживает такие базовые операции разработки, как выполнение задач, устранение неполадок и контроль версий. Он также имеет IntelliSense для разработки с использованием искусственного интеллекта, работает с GitHub и имеет богатую экосистему расширений. VS Code быстр, прост в использовании, легко настраивается и может быть хорошим выбором, если вам нужно что-то легкое, но с хорошей функциональностью

[Visual Studio Online](#), или GitHub Cloudspaces, представляет собой среду разработки на основе браузера, которая поддерживает VS 2019 и VS Code. Сейчас он доступен только в бета-версии и имеет ограниченную функциональность

Переход с Android Studio или Xcode на Visual Studio

Если ваша команда разработчиков рассматривает возможность перехода с Android Studio (AS) или Xcode на Visual Studio, кривая обучения не будет такой крутой. Сравнение функций между функциями показывает, что IDE очень похожи с точки зрения потоков разработки с небольшими нюансами для каждой из сред.

От Android Studio до Visual Studio. В то время как официальная поддержка Android Studio ориентирована на языки Java, C / C ++, Kotlin и XML, VS поддерживает все [основные языки программирования](#), включая AS. Это связано с широким спектром платформ, которые обслуживает IDE. VS предоставляет больше

возможностей для проектирования и позволяет работать с разными проектами из одной среды. Если вы используете облачную платформу Google, интегрированную с AS, для запуска виртуальных машин или просто управления проектами, Google любезно предоставит [расширение облачной платформы](#) и для VS. В принципе, вы можете использовать как облачную экосистему Microsoft, так и облако Google прямо из VS. Visual Studio может быть установлена на всех основных платформах: Windows, Android, Mac OS и Linux.

От Xcode до Visual Studio. Xcode IDE, аналогичная VS, поддерживает исходный код для всех основных языков программирования. Итак, после перехода не будет никаких сюрпризов. Опять же, если вы были привязаны к Mac OS во время использования Xcode, оборудование Apple для VS не требуется. Для тех, кто использует CloudKit для хранения на стороне сервера, Visual Studio и Xamarin обеспечивают [простую интеграцию CloudKit](#). Основным недостатком использования VS по сравнению с Xcode является его цена: как Android Studio, так и Xcode бесплатны для использования, в то время как VS требует довольно высокой абонентской платы.

По [оценкам разработчиков](#), Visual Studio и Android Studio имеют в основном одинаковый уровень простоты использования и качества поддержки. Сравнивая те же индексы с VS и Xcode, мы можем сказать, что оценки VS немного лучше. Кроме того, Visual Studio более популярна среди предприятий (> 1000 сотрудников), в то время как около 50 процентов пользователей Android Studio и Xcode – это малые предприятия (50 или менее сотрудников). Учитывая [рейтинг Strategy Analytics](#) в мобильных средах разработки, Xcode занял первое место по возможностям тестирования, как и Visual Studio. Android Studio отстала. Общее количество баллов: Xcode -22; Visual Studio -21; и Android Studio -13.

Рассматриваем другие варианты: Xamarin против гибридной разработки против родной iOS / Android против других кроссплатформенных фреймворков

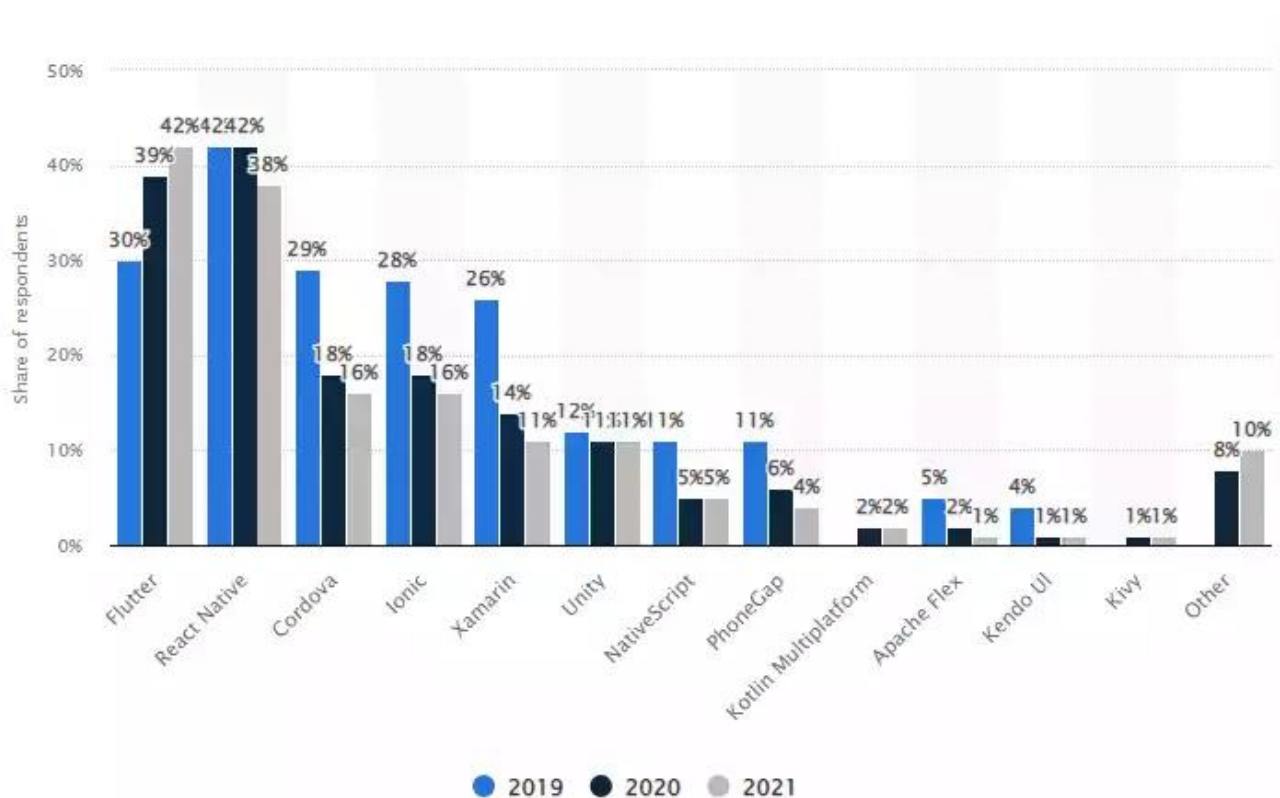
В последнее время многие разработчики склонны соглашаться с тем, что Xamarin можно считать “родным” инструментом разработки. Действительно, есть [мнение](#), что “все, что можно сделать в приложении для iOS с помощью Objective-C или Swift, и все, что можно сделать в приложении для Android с использованием Java, можно сделать на C # с использованием Xamarin”. Тем не менее, в дебатах о родном языке и Xamarin есть много подводных камней. Давайте посмотрим, как Xamarin сравнивается с собственными инструментами разработки и гибридными платформами разработки (Ionic, PhoneGap / Cordova).

	Xamarin	Native	Hybrid
Tech stack	One tech stack, single codebase (C#, .Net framework + native libraries)	Different tech stacks for each platform.	One tech stack, single codebase (Javascript, HTML5, CSS)
Code sharing	Yes (up to 96% with Xamarin.Forms)	No, different code bases	Yes, 100%
UI/UX	Complete UI customization is possible for each platform (with Xamarin.iOS and Xamarin.Android)	Completely platform-specific UI	Common UI for all platforms (limited customization capabilities)
Performance	Good, close to native	Excellent	Medium - Poor
Hardware capabilities	High - Xamarin uses platform-specific APIs, and supports linking with native libraries	High - Native tools have complete support for system capabilities out of the box.	Medium - The capabilities can be accessed through third-party APIs and plugins, although there are some risks due to the poor quality and unreliability of most of these tools.
Time to market	With Xamarin.Forms the time to market is fast due to the limited customization and extensive code sharing. Xamarin.iOS and Xamarin.Android require slightly more time as the amount of custom code increases.	The time to market for iOS or Android native app might equal that of Xamarin.Forms or Hybrid tools. Yet, building apps for multiple platforms, will require you to either prolong time to market or increase the number of developers involved.	Hybrid solutions offer the fastest time to market thanks to the single code base and minimum customization. These tools are even used for prototyping and proof of concept projects.

Xamarin против гибридной и нативной разработки в двух словах

Хотя гибридные инструменты мобильной разработки быстро развиваются, им все еще не хватает производительности и встроенных возможностей, которые предлагает Xamarin примерно за ту же стоимость. Однако более короткое время выхода на рынок остается их главным преимуществом. Что касается выбора между Xamarin или родной iOS / Android, вы должны учитывать доступное время и бюджет (собственная разработка обычно дороже и занимает больше времени) и тип приложения. Если вам нужна высокая производительность и идеально настроенный пользовательский интерфейс, стоит выбрать собственные приложения. По мере появления и развития все большего числа кроссплатформенных фреймворков Xamarin становится все труднее удерживать свои позиции на рынке. Сегодня ряд фреймворков уже опередили Xamarin по популярности и производительности. Лучшим выбором 2020 года был [признан React Native](#), за которым

следует [Flutter](#).



Динамика использования кроссплатформенных мобильных фреймворков. Источник: [Statista](#)

Каждый кроссплатформенный фреймворк приложения отличается функциональностью и структурой. Чтобы получить более полную картину, ознакомьтесь с нашими сравнительными статьями: [Xamarin vs Flutter](#) и [Xamarin vs React Native vs Ionic vs NativeScript](#). Последний, например, несмотря на меньшую популярность, был бы особенно полезен для тех разработчиков, которые используют [Angular](#) Framework, поскольку они тесно взаимосвязаны.

Совет

При сравнении плюсов и минусов перечисленные недостатки обычно рассматриваются как сопутствующий ущерб. Большинство владельцев бизнеса выбирают платформу [разработки мобильных приложений Xamarin](#), поскольку она сокращает время выхода на рынок и затраты на проектирование за счет совместного использования кода и использования единого технологического

стека. Тем не менее, назначение приложения и его целевая аудитория могут быть еще более важным фактором, который следует учитывать.

Исходя из опыта нашей команды, лучшим вариантом использования Xamarin являются **корпоративные мобильные решения**. Благодаря стандартному пользовательскому интерфейсу, который охватывает 90 процентов проектов, вся основная логика продукта может быть легко разделена между платформами. Следовательно, настройка платформы займет всего 5-10 процентов инженерных усилий.