

УДК: 614.8:656.1

## ИСТОРИЯ РАЗВИТИЯ И БУДУЩЕЕ ПЛАТФОРМЫ .NET

Тимоходцев П.Е., студент группы ИТм-201, II курс

Протодьяконов А.В. к.т.н., доцент каф.ИиАПС

ФГБОУ ВО «Кузбасский государственный технический университет  
имени Т.Ф. Горбачева»  
г. Кемерово

Корпорация Microsoft развивает платформу .NET на протяжении уже двадцати лет. На сегодняшний день разработчики могут выбрать четыре возможных разновидности платформы .NET для создания своих программных продуктов: .NET Framework, .NET Core, .NET Standard. В данной статье будут описаны основных отличия этих платформ и рассказано, в каких случаях каждую из них использовать. Также будет рассказано про новые версии .NET 5.0 и .NET 6.0 и почему они заменяют все вышеперечисленные версии платформы целиком и полностью.

Для более полного понимания платформы .NET нужно окунуться в историю. В 2002 году корпорация Microsoft пообещала одну единую платформу для создания приложений на любом языке программирования: C#, Visual Basic, F# и многих других языках. Платформа предоставляла общий набор API, независимость от языка, совместимость с базовой операционной системой и другими приложениями, отличными от .NET, переносимость и множество интересных функций.

В долгосрочной перспективе планировалось сделать .NET кроссплатформенной инфраструктурой. Но .NET Framework работала только в ОС Windows.

Microsoft опубликовала спецификации Common Language Infrastructure (CLI), то есть среды выполнений .NET Framework и языка C#. Более того, они установили их как открытые стандарты ECMA-335 и ECMA-334. Таким образом, почти каждый может реализовывать среду выполнения .NET и язык программирования C# [1].

В 2007 году это проложило путь для проекта по переносу .NET на Linux: Mono Project. Позже были предприняты и другие попытки портировать .NET на Linux: среди прочих: DotGNU и CrossNET. Однако только проект Mono вызвал значительный интерес и создал готовый к производству фреймворк.

Однако, некоторые функции .NET Framework строго зависимы от Windows, поэтому перенести весь функционал фреймворка в Mono не представлялось возможным. Кроме того, версия Mono обычно не соответствовала актуальной версии .NET Framework. Это создавало серьезные трудности по переносимости .NET на Linux.

Открытые спецификации основных компонентов .NET стали первым шагом к быстрому распространению фреймворка. Microsoft вместе со сторонними разработчиками постепенно перенесли .NET Framework на разные платформы.

Когда появился .NET Framework, Microsoft также выпустила .NET Compact Framework. Это была урезанная версия фреймворка, предназначенная для работы на устройствах с ограниченной вычислительной мощностью: КПК, мобильные телефоны, телевизионные приставки и т.д.

В 2007 году технология Silverlight перенесла .NET в браузер, предоставив разработчикам основу для создания многофункциональных интернет-приложений, подобных Adobe Flash.

В 2009 году стартовал проект MonoTouch. Это была первая попытка портировать фреймворк Mono на iOS и Android. Через несколько лет, в 2013 году, проект MonoTouch стал основой для Xamarin – фреймворка для создания приложений для Android и iOS с помощью C#.

Уже в 2013 году .NET позволял создавать для компьютеров с Windows и Linux, для встроенных устройств, для браузера, а также для устройств iOS и Android. Можно было бы подумать, что цель достигнута, .NET стал мультиплатформенным. Но нет. Распространение различных сред привело к фрагментации самого фреймворка. Это не было большой проблемой для тех, кто разрабатывал приложения. Так как приложение обычно зависит от платформы. Эта фрагментация была большой проблемой для разработчиков библиотек. Вообще, сборку .NET можно загрузить теоретически в любой среде выполнения .NET. Но, к сожалению, она может сломаться в любой момент, если вызовет API, который не поддерживается платформой, на которой происходит загрузка [3].

Для решения этой проблемы в 2014 году Microsoft начало писать .NET Core с нуля. Первая версия платформы была готова в 2016 году. Новая структура представляла собой полную переборку спецификации .NET с учетом кроссплатформенной поддержки. Фактически, его среда выполнена работа в Windows, Linux и macOS.

У .NET Core не было ограничений обратной совместимости, которые были у .NET Framework. Его модульная архитектура обеспечивала уменьшение версии основных компонентов платформы, позволяя разработчику загружать дополнительные компоненты из репозитория NuGet. Этот рефакторинг позволил упростить развертывание и переносимость.

.NET Core был шагом в правильном направлении для межплатформенной поддержки. Но он принес еще одну реализацию в уже запутанную экосистему .NET. Другими словами, у разработчика библиотеки есть еще одна потенциальная проблема с переносимостью своей библиотеки.

Microsoft попыталась упорядочить разросшийся .NET, представив .NET Standard. Это не просто еще одна реализация фреймворка, а формальная спецификация .NET API, которую должны реализовывать несколько платформ .NET. Различные версии .NET Standard определяют разные наборы .NET API. Таким образом, реализация .NET объявляет, какую версию .NET Standard она поддерживает. Разработчик библиотеки должен ориентироваться на эту конкретную версию .NET Standard. Например, если разработчик хочет, чтобы его библиотека работала в .NET Framework 4.5, .NET Core 2.1 и Mono 4.6, он должен настроить свой проект на .NET Standard 1.1. Более подробно про .NET Standard можно посмотреть в официальной документации Microsoft.

В ноябре 2020 года был выпущен .NET 5. Это новая версия фреймворка отныне призвана стать единой платформой .NET. Больше нет необходимости в различных реализациях: .NET 5 – это общий набор API-интерфейсов, который работает в Windows, Linux, Mac, iOS и Android [Рисунок 1].

## .NET – A unified platform

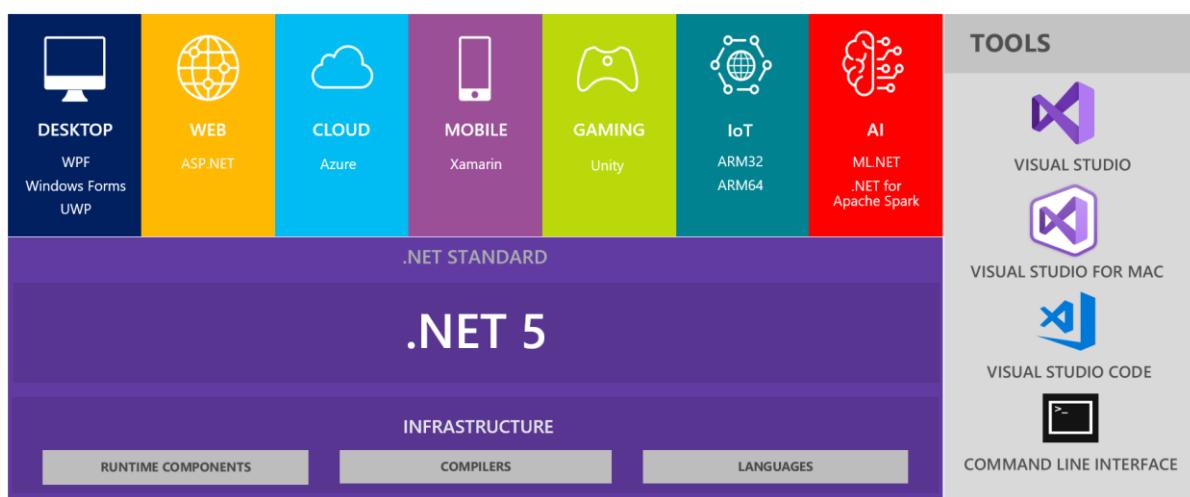


Рисунок 1 – Возможности платформы .NET 5.0

.NET 5 устраняет необходимость в .NET Standard. На самом деле можно думать о .NET 5 как об эволюции .NET Standard. Однако, .NET 5 – это не формальная спецификация, а фактическая реализация. Кроме того, выпуск .NET 5 знаменует собой конец .NET Framework [2]. Получается, что отныне в .NET больше не будет Framework, Core, Compact или Micro. Будущее .NET не будет иметь другого определителя, кроме как его версии.

.NET 5 – логическое продолжение .NET Core. Он нацелен на улучшение .NET в нескольких аспектах:

- Создание единой исполняющей среды и фреймворка, которые можно использовать везде, с одинаковым поведением в отладке.
- Расширение возможностей .NET за счет лучших аспектов .NET Core, .NET Core, Xamarin и Mono.
- Собрать единый продукт, над которым разработчики из Microsoft и сообщества будут вместе работать и расширять.

К основным нововведениям .NET 5 можно отнести:

- Однофайловые приложения (single file application). Все зависимости отныне помещаются в один файл с исполняющим файлом. И на целевом компьютере больше нет необходимости предварительно устанавливать фреймворк .NET.
- Улучшена поддержка и разработка десктоп-приложений (Windows Forms, WPF)
- Новая версия C# 9.0
- Поддержка Windows ARM64

В ноябре 2021 года Microsoft выпустила .NET 6.0, который продолжает развивать единую кроссплатформенную среду разработки.

Но выпуск новой единой платформы для разработки не означает, что .NET Framework исчезнет. На нем разработано множество программных продуктов, перенос которых на новую версию фреймворка будет очень дорогим и нецелесообразным. Но новые продукты предпочтительнее разрабатывать на актуальной версии .NET 6.0.

По итогу можно сказать, что Microsoft продолжает прогрессивно развивать платформу .NET и расширять ее возможности. За двадцать лет .NET стал полностью мультиплатформенным. Каждый год вносятся различные улучшения в платформу, сопутствующие фреймворки и языки программирования. С каждым годом возможности разработки на .NET расширяются, а сложность создания приложений уменьшается. Возможно, в будущем .NET сможет вытеснить с рынка такой фундаментальный язык, как Java.

### **Список литературы:**

1. .NET и .NET Framework для серверных приложений. [Электронный ресурс] URL: <https://docs.microsoft.com/ru-ru/dotnet/standard/choosing-core-framework-server>
2. Introducing .NET 5. [Электронный ресурс] URL: [https://devblogs.microsoft.com/dotnet/introducing-net-5/?ocid=AID742437\\_VK\\_oo\\_spl100000612090603](https://devblogs.microsoft.com/dotnet/introducing-net-5/?ocid=AID742437_VK_oo_spl100000612090603)

3. История и альтернативы платформы .NET. [Электронный ресурс]  
URL: <https://habr.com/ru/company/clrium/blog/466811/>