

## 2. Understanding .NET

### 1. Từ .NET Framework tới .NET CORE tới .NET

#### a. .NET framework:

- Version đầu tiên của .NET release năm 2002, để phát triển windows forms, windows desktop và web forms (phát triển web app)
- .NET framework version này chỉ chạy trên Microsoft Windows (không chạy được trên các platform khác)
- Sau này, Silverlight coi như một subset của .NET Framework - một runtime chạy tại browsers thông qua các add-in. Cùng thời điểm này công ty Ximian phát triển Mono runtime - giúp app chạy trên Linux và Android. Và Miguel de Icaza (the founder of Ximian) phát triển Xamarin và đưa các parts của .NET vào để phát triển app trên nền tảng Android & iOS.
- Microsoft mua Xamarin vào năm 2016

#### b. .NET Core

- Microsoft nỗ lực để tách các phần chính ra khỏi Core (lõi) làm sao để phát triển được các ứng dụng cross-platform
- .NET Core có nhiều thay đổi lớn so với .NET Framework. Code .NET ở dạng open-source và các app chạy đa nền tảng.
- Các code mới sẽ dựa trên các design pattern hiện đại
- NET Core tiếp cận theo dạng hướng module, framework được tách ra thành các module trong NuGet packages

#### c. .NET

- Sau .NET Core 3.1 là .NET 5 (phần Core bị bỏ) và version 4 cũng bị bỏ để chuyển sang 5 (NET 5) để phù hợp với .NET Framework 4.8
- Việc chuyển một project chạy trên .NET Core sang .NET rất dễ dàng nhưng .NET framework thì rất khó khăn, đòi hỏi nhiều thay đổi (về cấu hình, về code...)
- .NET Core 3.x có hỗ trợ WPF và Windows Forms applications (Cung cấp một loạt API hỗ trợ phát triển ứng dụng windows form) và cả Blazor server (để phát triển web app)
- Với .NET và NuGet packages Microsoft cho phép dự án tiếp cận nhanh các tính năng (được đóng gói) mới.
- Các công nghệ không còn được cung cấp trong .NET là ASP.NET Web forms, Windows Communication Foundation (WCF) và Windows Workflow Foundation (WF),
- Thay vì dùng ASP.NET web forms trong .NET dùng ASP.NET Blazor, thay vì dùng WCF ta dùng ASP.NET Core Web API hoặc gRPC, thay vì WF thì ta dùng Azure logic

#### d. .NET Standard

- .NET platform có 3 nền tảng để dùng trong các ngữ cảnh app khác nhau: .NET Core, .NET framework, Xamarin (cho mobile). Mỗi platform có điểm yếu và mạnh khác nhau và devs phải học 3 nền tảng này để phát triển,
- Vì thế, Microsoft cho ra .NET Standard với nhiều đặc tả (trừu tượng) các APIs mà tất các APIs của .NET có thể dùng để implements
- Với .NET Standard 2.0 +, Microsoft đã "tụ" cả 3 nền tảng trên lại và giúp dễ dàng chia sẻ mã nguồn của bất kỳ platform .net nào.

### 2. .NET TERMS

#### a. .NET SDK: .NET command-line interface (CLI), tools, libraries, và the runtime. CLI giúp devs tạo và build application. Và .NET SDK được cài sẵn trong Visual Studio 2019+. Lệnh sau show các version SDK (default latest version được dùng)

```
=====
```

```
>dotnet --list-sdks
```

```
=====
```

Lệnh check dotnet version hiện tại:

```
=====
```

```
>dotnet --version
```

```
=====
```

#### b. .NET RUNTIME

- Trong môi trường target system, SDK không cần thiết, chỉ cần setup .NET runtime, Nó bao gồm: các thư viện core và dotnet driver. Dotnet driver giúp run app. Thí dụ:

```
=====
```

```
>dotnet hell-app.dll
```

```
=====
```

- Để show các runtime dùng lệnh sau:

```
=====
```

```
>dotnet --list-runtimes
```

```
=====
```

#### c. Common Language Runtime (CLR)

- C# compiler compiles C# code sang Microsoft Intermediate Language (IL) code (là một phần của assembly code), chúng có nhiều tính năng theo dạng hướng đối tượng và IL run by CLR

- The IL code is compiled sang native code bởi CLR. IL code available in .NET assemblies is compiled by a Just-In-Time (JIT) compiler. This compiler creates platform-specific native code.

d. NET Compiler Platform

- C# compiler như một phần được cài đặt cùng với SDK và nó thuộc về .NET Compiler Platform

3. Danh sách các version của .NET

.NET CORE/.NET VERSION	RELEASE DATE	SUPPORT LEVEL	END OF LIFE
1.0	June 27, 2016	LTS	June 27, 2019
1.1	Nov. 16, 2016	LTS*	June 27, 2019
2.0	Aug. 14, 2017	Current	Oct. 1, 2018
2.1	May 30, 2018	LTS	Aug. 21, 2021
2.2	Dec. 4, 2018	Current	Dec. 23, 2019
3.0	Sep. 23, 2019	Current	Mar. 3, 2020
3.1	Dec. 3, 2019	LTS	Dec. 3, 2022
5.0	Nov. 10, 2020	Current	around Feb. 2022
6.0	Nov. 2021	LTS	Nov. 2024
7.0	Nov. 2022	Current	Feb. 2024 or earlier in case minor versions are released
8.0	Nov. 2023	LTS	Nov. 2026

4. Các kiểu ứng dụng và công nghệ

4.1. Data access ( phương pháp truy cập dữ liệu):

- Các files và directories được truy cập bởi các APIs (simple), và simple APIs không đủ linh hoạt cho một trường hợp. Vì thế dùng Stream APIs có nhiều sự linh hoạt và nhiều tính năng hơn như mã hóa và nén data.
- Để tương tác với database ta có thể sử dụng abstraction layer như Entity Framework Core

4.2. Windows Application:

- Ta có thể sử dụng new UI control WINUI 3.0 để tạo windows desktop application

4.3. Web application:

- Ta sử dụng ASP.NET Core MVC
- ASP.NET Core Razor Pages cung cấp lựa chọn để phát triển web app mạnh mẽ hơn
- ASP.NET Core Blazor là một new technology được sử dụng để loại JavaScript code

4.4. Services

- Trước đây, phát triển các dịch vụ thông qua SOAP và WCF
- Các ứng dụng hiện đại sử dụng dạng REST và API web. Với .NET sử dụng ASP.NET Core để tạo ra các Web APIs dễ dàng giao tiếp hơn cho các Apps rất nhiều trong các ứng dụng phân tán (90%)
- Với Web API là cách tiếp cận tốt để tạo ra các ứng dụng dạng Microservices (thay vì Monolithic)
- Để đặc tả các dịch vụ (services), một chuẩn mới đã được phát triển đó là OPENAPI (<https://www.openapis.org/>) có nguồn gốc từ Swagger (<https://swagger.io/>).

4.5. SignalR

- Sử dụng phát triển các kết nối real-time giữa client - server
- SignalR là một công nghệ của ASP.NET Core, SignalR được tạo ra bằng công nghệ Websocket để push các thông tin (data).

