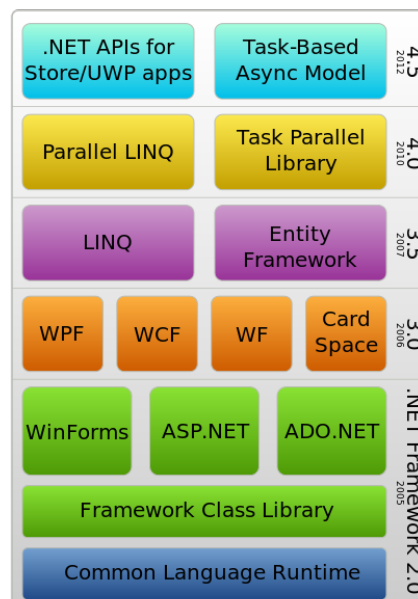


TERMINOLOGY

Due to the nature of this project and requirements by the industry partner Sivantos, there are a couple of technologies used one should have a good overview and understanding of.

- **.NET** is the name of a multitude of software frameworks and development platforms provided by Microsoft. It consists of libraries offering user interfaces, security / cryptography, database access, etc.



Different Components offered as .NET ¹

.NET comes in three different flavors: *.NET Framework*, *.NET Core* and *Xamarin*. All of these share same fundamental principles, but offer different class libraries as well as target platforms i.e. while .NET Framework is the oldest, historically grown project since 2002, it only runs on Windows but is perfectly suited for cases where legacy systems need to be supported as well. On the other hand, Xamarin is used for creating iOS / Android mobile applications with the support of a Microsoft Environment and usually C# as the programming language of choice. ²

- **.NET Core and Universal Windows Platform (UWP)**
UWP helps develop universal apps that run on both Windows 10 and Windows 10 Mobile without the need to be re-written for each. UWP provides a guaranteed core API layer across all devices that run Windows 10. UWP .NET apps use .NET Core underneath. UWP is only for the Windows ecosystem. *You can use .net core to create an app targeting UWP, but UWP isn't part of .net, it's part of Windows.* ³
- **Windows IoT**
(previously Windows Embedded) is a family of operating systems from Microsoft designed for use in embedded systems. ⁴ There are three different versions such as Enterprise, Mobile Enterprise, as well as Core. Windows IoT Core is basically a watered-down version of Windows, running on small, low-cost devices such as the Raspberry Pi. For

¹ https://en.wikipedia.org/wiki/.NET_Framework

² <https://blogs.msdn.microsoft.com/cesardelatorre/2016/06/27/net-core-1-0-net-framework-xamarin-the-whatand-when-to-use-it/#dotnet-whole>

³ <https://stackoverflow.com/questions/38680266/is-net-for-universal-windows-program-a-subset-of-net-core>

⁴ https://en.wikipedia.org/wiki/Windows_IoT



management of the Pi where IoT Core runs on, Microsoft offers Windows 10 IoT Core Dashboard⁵.

- **Visual Studio** is an integrated development environment (IDE) from Microsoft. It is mandatory to use this for this project, because UWP applications can only be coded on this platform, and UWP is what is used to make the code run later on the Raspberry Pi.

➔ **For an Introduction on how to get set up, refer to:**

*Getting Visual Studio (Windows IoT) with Raspberry Pi setted up.pdf*⁶

- **NuGet**

is a package manager that comes bundled with Visual Studio. When you use NuGet to install a package, it copies the library files to your solution and automatically updates your project (add references, change config files, etc.). If you remove a package, NuGet reverses whatever changes it made so that no clutter is left.⁷

- **Testing**

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.⁸ In the case of this project, a requirement is to use Travis CI to help with automating tests and integration.

```
using Xunit;

namespace Fibonacci.Tests
{
    public class Tests
    {
        [Fact] // fact is a keyword used by xunit
        public void Get_5th_number()
        {
            var generator = new FibonacciGenerator();

            Assert.Equal(generator.Fibonacci(5), 15);
        }

        [Fact]
        public void Get_6th_number()
        {
            var generator = new FibonacciGenerator();

            Assert.Equal(generator.Fibonacci(6), 8);
        }
    }
}
```

Simplest Form of a Test using Asserts

- **Continuous integration**

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.⁹

⁵ <https://developer.microsoft.com/en-us/windows/iot/docs/iotdashboard>

⁶ <https://drive.google.com/open?id=0BzaNmZTtJK4Tk9CaW1lQ0JTV0U>

⁷ <https://www.nuget.org/>

⁸ Kolawa, Adam; Huizinga, Dorota (2007). *Automated Defect Prevention: Best Practices in Software Management*. Wiley-IEEE Computer Society Press.

p. 75. ISBN 0-470-04212-5.

⁹ <https://www.thoughtworks.com/continuous-integration>

