**Следните основни цели и идеи са в основата на философията на .NET Framework:**

* **Платформена независимост** - Една от основните цели на .NET Framework е да даде възможност за създаването на платформено независими приложения, т.е. на приложения, които могат да работят на различни операционни системи и хардуерни конфигурации. До този момент Microsoft са създали имплементации на .NET Framework, които работят на x86 и x64 персонални компютри, Windows CE базирани устройства ([PDA](http://bg.wikipedia.org/wiki/PDA), [SmartPhone](http://bg.wikipedia.org/w/index.php?title=SmartPhone&action=edit&redlink=1), [Tablet PC](http://bg.wikipedia.org/w/index.php?title=Tablet_PC&action=edit&redlink=1)), както и игровата конзола [XBox 360](http://bg.wikipedia.org/w/index.php?title=XBox_360&action=edit&redlink=1). Спецификациите на CLI (*Common Language Infrastructure*, което обхваща BCL, CTS и CIL), както и езиците C# и C++/CLI са предадени на [ECMA](http://bg.wikipedia.org/w/index.php?title=ECMA&action=edit&redlink=1) и [ISO](http://bg.wikipedia.org/wiki/ISO) и са отворени стандарти. По този начин независими разработчици могат да поставят .NET Framework и прилежащите му езици на други софтуерни и хардуерни платформи.
* **Съвместимост с вече съществуващите технологии** - .NET Framework дава възможност за достъп до функционалност, реализирана в приложения извън .NET средата. Пространствата System.Runtime.InteropServices и System.EnterpriseServices дават възможност за достъп до [COM](http://bg.wikipedia.org/w/index.php?title=Component_Object_Model&action=edit&redlink=1) компоненти, докато с [P/Invoke](http://bg.wikipedia.org/w/index.php?title=Platform_Invocation_Services&action=edit&redlink=1) може да се извиква неуправляван код (*native code*).
* **Common Runtime Engine** - Програмните езици, които се поддържат от .NET Framework биват компилирани в [междинен код](http://bg.wikipedia.org/w/index.php?title=%D0%9C%D0%B5%D0%B6%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD_%D0%BA%D0%BE%D0%B4&action=edit&redlink=1) известен като [CIL](http://bg.wikipedia.org/w/index.php?title=CIL&action=edit&redlink=1) (*Common Intermediate Language*) а по-рано и като [MSIL](http://bg.wikipedia.org/w/index.php?title=MSIL&action=edit&redlink=1) (*Microsoft Intermediate Language*). При изпълнението си този междинен код не бива интерпретиран като при други виртуални машини, а вместо това компилиран по начин, известен като [JIT](http://bg.wikipedia.org/w/index.php?title=JIT&action=edit&redlink=1) (*Just In Time*) компилация в платформено-зависим машинен код (*native code*).
* **Езикова независимост** - Приложения за .NET Framework могат да бъдат създавани на няколко съвместими програмни езици, сред които [C#](http://bg.wikipedia.org/wiki/C_Sharp), [VB.NET](http://bg.wikipedia.org/w/index.php?title=VB.NET&action=edit&redlink=1) и [C++/CLI](http://bg.wikipedia.org/w/index.php?title=C%2B%2B/CLI&action=edit&redlink=1). Това е възможно благодарение на съвместимостта на типовете данни, които отделните езици поддържат. [CTS](http://bg.wikipedia.org/w/index.php?title=CTS&action=edit&redlink=1) (*Common Type System*) дефинира всички базови типове данни, както и начинът, по който те могат да бъдат конвертирани един в друг. Тези типове са споделени между всички .NET езици и са стандартизирани в CLI.
* **Улеснена инсталация** - Програмите, създадени на .NET Framework, както и техните компоненти, могат да бъдат инсталирани с просто копиране в желаната директория — процес, известен като *XCopy Deployment*.
* **Сигурност** - Програмният код, написан на .NET, се нарича *управляван код*, а също и *защитен код*, тъй като е изолиран от хардуерната среда, в която оперира, и това го предпазва от някои видове програмни грешки, които правят кода уязвим за атаки, например препълването на буфера (*buffer overflow*). Също така .NET Framework предлага система за сигурност, която може да се използва от всички .NET приложения.

**Компоненти на .NET Framework**

.NET Framework включва два основни компонента:

* *Common Language Runtime* (CLR) – програмната среда, в която се изпълнява кодът на .NET приложенията. Представлява [виртуална машина](http://bg.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D0%BD%D0%B0_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0), която компилира междинния код [CIL](http://bg.wikipedia.org/w/index.php?title=CIL&action=edit&redlink=1) (*Common Intermediate Language*) за конкретната хардуерна платформа и операционна система, с която работи потребителят. В допълнение средата осигурява редица услуги, като управление на паметта, вкл. автоматично "събиране на боклуци" ('garbage collection'), управление на прекъсванията, многонишковост ('multithreading') и други.
* *Framework Class Library* (FCL) – това е пълната колекция от класове, които .NET Framework предлага. Частта от FCL, известна като [BCL](http://bg.wikipedia.org/w/index.php?title=BCL&action=edit&redlink=1) (*Base Class Library*, 'стандартна библиотека от класове'), е разработената от Microsoft имплементация на стандарта [ECMA-335, Common Language Infrastructure](http://www.ecma-international.org/publications/standards/Ecma-335.htm). BCL осигурява голям набор базови функционалности, които програмистите могат да използват директно, например за [мрежова комуникация](http://bg.wikipedia.org/w/index.php?title=%D0%9C%D1%80%D0%B5%D0%B6%D0%BE%D0%B2%D0%B0_%D0%BA%D0%BE%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F&action=edit&redlink=1), [криптографски услуги](http://bg.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F) и други. В допълнение FCL предлага компоненти за изграждане на [потребителския интерфейс](http://bg.wikipedia.org/wiki/GUI) ([Windows.Forms](http://bg.wikipedia.org/w/index.php?title=Windows.Forms&action=edit&redlink=1) и [ASP.NET](http://bg.wikipedia.org/wiki/ASP.NET)), [уеб приложения](http://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) и [уеб услуги](http://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D1%83%D1%81%D0%BB%D1%83%D0%B3%D0%B0), достъп до [бази данни](http://bg.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D0%B8) и др.

Основни принципи и цели[[редактиране](http://bg.wikipedia.org/w/index.php?title=Microsoft_.NET_Framework&veaction=edit&vesection=2) | [редактиране на кода](http://bg.wikipedia.org/w/index.php?title=Microsoft_.NET_Framework&action=edit&section=2)]

Следните основни цели и идеи са в основата на философията на .NET Framework:

* **Платформена независимост** - Една от основните цели на .NET Framework е да даде възможност за създаването на платформено независими приложения, т.е. на приложения, които могат да работят на различни операционни системи и хардуерни конфигурации. До този момент Microsoft са създали имплементации на .NET Framework, които работят на x86 и x64 персонални компютри, Windows CE базирани устройства ([PDA](http://bg.wikipedia.org/wiki/PDA), [SmartPhone](http://bg.wikipedia.org/w/index.php?title=SmartPhone&action=edit&redlink=1), [Tablet PC](http://bg.wikipedia.org/w/index.php?title=Tablet_PC&action=edit&redlink=1)), както и игровата конзола [XBox 360](http://bg.wikipedia.org/w/index.php?title=XBox_360&action=edit&redlink=1). Спецификациите на CLI (*Common Language Infrastructure*, което обхваща BCL, CTS и CIL), както и езиците C# и C++/CLI са предадени на [ECMA](http://bg.wikipedia.org/w/index.php?title=ECMA&action=edit&redlink=1) и [ISO](http://bg.wikipedia.org/wiki/ISO) и са отворени стандарти. По този начин независими разработчици могат да поставят .NET Framework и прилежащите му езици на други софтуерни и хардуерни платформи.
* **Съвместимост с вече съществуващите технологии** - .NET Framework дава възможност за достъп до функционалност, реализирана в приложения извън .NET средата. Пространствата System.Runtime.InteropServices и System.EnterpriseServices дават възможност за достъп до [COM](http://bg.wikipedia.org/w/index.php?title=Component_Object_Model&action=edit&redlink=1) компоненти, докато с [P/Invoke](http://bg.wikipedia.org/w/index.php?title=Platform_Invocation_Services&action=edit&redlink=1) може да се извиква неуправляван код (*native code*).
* **Common Runtime Engine** - Програмните езици, които се поддържат от .NET Framework биват компилирани в [междинен код](http://bg.wikipedia.org/w/index.php?title=%D0%9C%D0%B5%D0%B6%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD_%D0%BA%D0%BE%D0%B4&action=edit&redlink=1) известен като [CIL](http://bg.wikipedia.org/w/index.php?title=CIL&action=edit&redlink=1) (*Common Intermediate Language*) а по-рано и като [MSIL](http://bg.wikipedia.org/w/index.php?title=MSIL&action=edit&redlink=1) (*Microsoft Intermediate Language*). При изпълнението си този междинен код не бива интерпретиран като при други виртуални машини, а вместо това компилиран по начин, известен като [JIT](http://bg.wikipedia.org/w/index.php?title=JIT&action=edit&redlink=1) (*Just In Time*) компилация в платформено-зависим машинен код (*native code*).
* **Езикова независимост** - Приложения за .NET Framework могат да бъдат създавани на няколко съвместими програмни езици, сред които [C#](http://bg.wikipedia.org/wiki/C_Sharp), [VB.NET](http://bg.wikipedia.org/w/index.php?title=VB.NET&action=edit&redlink=1) и [C++/CLI](http://bg.wikipedia.org/w/index.php?title=C%2B%2B/CLI&action=edit&redlink=1). Това е възможно благодарение на съвместимостта на типовете данни, които отделните езици поддържат. [CTS](http://bg.wikipedia.org/w/index.php?title=CTS&action=edit&redlink=1) (*Common Type System*) дефинира всички базови типове данни, както и начинът, по който те могат да бъдат конвертирани един в друг. Тези типове са споделени между всички .NET езици и са стандартизирани в CLI.
* **Улеснена инсталация** - Програмите, създадени на .NET Framework, както и техните компоненти, могат да бъдат инсталирани с просто копиране в желаната директория — процес, известен като *XCopy Deployment*.
* **Сигурност** - Програмният код, написан на .NET, се нарича *управляван код*, а също и *защитен код*, тъй като е изолиран от хардуерната среда, в която оперира, и това го предпазва от някои видове програмни грешки, които правят кода уязвим за атаки, например препълването на буфера (*buffer overflow*). Също така .NET Framework предлага система за сигурност, която може да се използва от всички .NET приложения.

**Версии**

Microsoft стартира разработката на .NET Framework в края на 90-те години, първоначално под името *Next Generation Windows Services (NGWS)*. В края на 2000 г. е публикувана първата бета версия на .NET 1.0.

**.NET Framework 1.0**

**.NET Framework 1.1**

Публикуван на 3 април 2003 г., това е първият сериозен ъпгрейд на .NET Framework. Може да бъде изтеглен като самостоятелен пакет или инсталиран с втората версия на Visual Studio .NET, [Visual Studio .NET 2003](http://bg.wikipedia.org/w/index.php?title=Visual_Studio_.NET_2003&action=edit&redlink=1). Това е и първата версия на .NET Framework, вградена директно в Windows операционна система (Windows Server 2003).

**Промени спрямо версия 1.0**

* Вградена поддръжка на мобилни [ASP.NET](http://bg.wikipedia.org/wiki/ASP.NET) контроли.
* Подобрена сигурност, [Code Access Security](http://bg.wikipedia.org/w/index.php?title=Code_Access_Security&action=edit&redlink=1) в ASP.NET приложенията.
* Вградена поддръжка на ODBC и клиент за Oracle бази данни.
* [.NET Compact Framework](http://bg.wikipedia.org/w/index.php?title=.NET_Compact_Framework&action=edit&redlink=1) - версия на .NET Framework за устройства с ограничени възможности.
* Поддръжка на Internet Protocol version 6 ([IPv6](http://bg.wikipedia.org/wiki/IPv6))

**.NET Framework 2.0**

Публикувана на 22 януари 2006, вградена във [Visual Studio .NET 2005](http://bg.wikipedia.org/w/index.php?title=Visual_Studio_.NET_2005&action=edit&redlink=1), [Microsoft SQL Server 2005](http://bg.wikipedia.org/w/index.php?title=Microsoft_SQL_Server_2005&action=edit&redlink=1) и [BizTalk 2006](http://bg.wikipedia.org/w/index.php?title=BizTalk_2006&action=edit&redlink=1), както и в операционната система [Windows Server 2003 R2](http://bg.wikipedia.org/w/index.php?title=Windows_Server_2003&action=edit&redlink=1). Достъпна за изтегляне от [сайта на Microsoft](http://www.microsoft.com/downloads/details.aspx?FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5&DisplayLang=en), откъдето може да се свали и [нейния SDK](http://www.microsoft.com/downloads/details.aspx?FamilyID=fe6f2099-b7b4-4f47-a244-c96d69c35dec&DisplayLang=en).

Следва да се отбележи, че версия 2.0 е последната, която поддържа [Windows 2000](http://bg.wikipedia.org/wiki/Windows_2000).

**Промени спрямо версия 1.1**

* Множество подобрения по базовите класове
* Поддръжка на 64-битови хардуерни платформи (x64 и IA64)
* Generics
* Много нови и подобрени ASP.NET контроли
* Declarative data binding
* Нови възможности за персонализация на ASP.NET страниците: теми (*themes*), скинове (*skins*) и *webparts*.
* [.NET Micro Framework](http://bg.wikipedia.org/w/index.php?title=.NET_Micro_Framework&action=edit&redlink=1) - [версия](http://www.aboutnetmf.com/entry.asp) на .NET Framework свързана с инициативата [Smart Personal Objects Technology](http://bg.wikipedia.org/w/index.php?title=Smart_Personal_Objects_Technology&action=edit&redlink=1).

**.NET Framework 3.0**

**.NET Framework 3.0**, известен първоначално като **WinFX**, включва набор нови възможности и библиотеки, които са в основата на новите операционни системи [Windows Vista](http://bg.wikipedia.org/wiki/Windows_Vista) и [Windows Server 2008](http://bg.wikipedia.org/wiki/Windows_Server_2008). .NET Framework 3.0 е достъпен и за операционните системи [Windows XP](http://bg.wikipedia.org/wiki/Windows_XP) SP2 and [Windows Server 2003](http://bg.wikipedia.org/w/index.php?title=Windows_Server_2003&action=edit&redlink=1). В новата версия липсват сериозни архитектурни нововъведения: използва се CLR на .NET Framework 2.0. За версия 3.0 няма еквивалентна версия на .NET Compact Framework.

.NET Framework 3.0 се състои от 4 основни нови компонента:

* [**Windows Presentation Foundation**](http://bg.wikipedia.org/w/index.php?title=Windows_Presentation_Foundation&action=edit&redlink=1) (WPF), известен по-рано като **Avalon** - нова система за [графичен интерфейс](http://bg.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D0%BD_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) и [API](http://bg.wikipedia.org/w/index.php?title=Application_programming_interface&action=edit&redlink=1) базиран на [XML](http://bg.wikipedia.org/wiki/XML) и [векторни графики](http://bg.wikipedia.org/w/index.php?title=%D0%92%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D0%B8_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B8&action=edit&redlink=1), която използва възможностите на 3D графичните карти и [Direct3D](http://bg.wikipedia.org/w/index.php?title=Direct3D&action=edit&redlink=1) технологии. Множество статии и пълна документация на WPF са налични в [WPF SDK](http://msdn2.microsoft.com/en-us/library/ms754130.aspx).
* [**Windows Communication Foundation**](http://bg.wikipedia.org/w/index.php?title=Windows_Communication_Foundation&action=edit&redlink=1) (WCF), известен по-рано като **Indigo** - сервизно-ориентирана система за съобщения (*service-oriented messaging system*), която позволява на приложенията да комуникират локално или отдалечено подобно на [уеб услуги](http://bg.wikipedia.org/wiki/%D0%A3%D0%B5%D0%B1_%D1%83%D1%81%D0%BB%D1%83%D0%B3%D0%B0).
* [**Windows Workflow Foundation**](http://bg.wikipedia.org/w/index.php?title=Windows_Workflow_Foundation&action=edit&redlink=1) (WF) - *Необходимо е описание на WF*.
* [**Windows CardSpace**](http://bg.wikipedia.org/w/index.php?title=Windows_CardSpace&action=edit&redlink=1), известен по-рано като **InfoCard** осигурява надеждното съхранение на "дигитални самоличности" (*digital identities*) и предлага унифициран интерфейс за избор на подходящата самоличност за извършването на определена задача, напр. влизане в даден уеб сайт.

**.NET Framework 3.5**

Версия 3.5 на .NET Framework използва същия CLR като версии 2.0 и 3.0. Тя автоматично инсталира .NET Framework 2.0 SP1, с което добавя някои нови функции в базовите класове на версия 2.0, необходими за работата на някои от новите компоненти, най-вече [LINQ](http://bg.wikipedia.org/w/index.php?title=Language_Integrated_Query&action=edit&redlink=1). Налице е пълна обратна съвместимост с версия 2.0, така че програмите, писани за версия 2.0 ще работят безпроблемно на версия 3.5. Пълна поддръжка на .NET Framework 3.5 е осигурена във [Visual Studio 2008](http://bg.wikipedia.org/w/index.php?title=Visual_Studio_2008&action=edit&redlink=1).

Пълният сорс код на BCL ще бъде публикуван от Microsoft под лиценз *Microsoft Reference License*.

Скоро ще бъде публикувана .NET Compact Framework 3.5, която ще направи достъпни LINQ и другите нововъведения в .NET Framework 3.5 за мобилни устройства.

**Промени спрямо версия 3.0**

Нови версии на най-популярните .NET езици - [C#](http://bg.wikipedia.org/wiki/C_Sharp) 3.0 и [VB.NET](http://bg.wikipedia.org/w/index.php?title=VB.NET&action=edit&redlink=1) 9.0

* Ламбда изрази
* Разширяващи методи (*extension methods*)
* Анонимни типове
* LINQ (*Language Integrated Query*), включително компонентите:
	+ LINQ to Objects
	+ LINQ to XML
	+ LINQ to SQL
* ADO.NET paging
* Методи за синхронизация на ADO.NET локални кешове със сървърни източници на данни
* Асинхронно мрежово I/O API
* Обвивки на интерфейсите на [WMI](http://bg.wikipedia.org/w/index.php?title=WMI&action=edit&redlink=1) и [Active Directory](http://bg.wikipedia.org/w/index.php?title=Active_Directory&action=edit&redlink=1)
* Подобрения по [WCF](http://bg.wikipedia.org/w/index.php?title=Windows_Communication_Foundation&action=edit&redlink=1) and [WF](http://bg.wikipedia.org/w/index.php?title=Windows_Workflow_Foundation&action=edit&redlink=1), които дават възможност на WCF да работи с [POX](http://bg.wikipedia.org/w/index.php?title=POX&action=edit&redlink=1) и [JSON](http://bg.wikipedia.org/wiki/JSON), както и да представя WF workflows като WCF услуги.
* Поддръжка на [HTTP pipelining](http://bg.wikipedia.org/w/index.php?title=HTTP_pipelining&action=edit&redlink=1) и [syndication feeds](http://bg.wikipedia.org/w/index.php?title=Web_feed&action=edit&redlink=1)
* Вградена поддръжка на AJAX

**.NET Framework 4.0**

Версия 4.0 се разпространява заедно с [Visual Studio 2010](http://bg.wikipedia.org/w/index.php?title=Visual_Studio_2010&action=edit&redlink=1). Версията на CLR е сменена на 4.0

**Промени спрямо версия 3.5**

* Parallel Extensions — вградена поддръжка на изпълнение на паралелни изчисления върху многоядрени процесори.
* dynamic ключова дума (при обръщения към COM обекти)
* Наименования и незадължителни параметри.