

Benefits of .NET

White Paper

Abstract

This paper describes the in brief the values of the Microsoft .NET initiative from a development perspective.

Benefits of .NET

Revised by Chris Rindal
QA Technician, Tometa Software, Inc.

Tometa creates custom software for you

Tometa Software designs and develops robust software solutions for virtually all industries including in-house (vertical market) and retail software, some of which is on the shelves at your local software store. We focus our unique combination of creative, technical, and problem-solving skills on meeting our client's objectives. Because of our clarity of purpose, commitment to process, and broad professional skill sets, we are able to provide our clients with world-class solutions that are functionally superior and fully aligned with our client's strategic focus.

Balancing development speed, quality and cost is what we are all about. Tometa combines agile development practices with fixed pricing, so you know what the cost, end product, and delivery time table look like—up front. If we underestimate the effort, we complete the overrun on our dime. Simple as that. That's why large enterprise firms like Alcoa and NASDAQ choose Tometa.

Tometa's agile development expertise and low-overhead US location keep our prices comparable to offshore vendors – without offshore challenges. Using a fixed pricing model, we provide upfront visibility into a project's ultimate costs, end product and delivery schedule. Our clients like knowing that we have “skin in the game” – a fixed price that aligns our goals with yours, incenting us to get the job done right and fast.

Lastly, as a Microsoft Certified Gold Partner, Tometa Software, can customize its products or create custom web, client/server, and traditional applications. With programming experience in C#, C++, Visual Basic, CGI, HTML, RPG, Delphi, Java and many others; Tometa Software is uniquely positioned to meet your needs as a development firm.

[Check us out today](#)

What are the benefits of the .NET Framework? Put simply, faster time to market, easier deployment and administration, and improved performance. Here are some of the key benefits:

- **Any Language.** The .NET Framework enables developers to use any programming language, and for applications written in any programming languages to integrate deeply with each other, enabling current development skills to go forward without retraining.
- **Less Code.** The .NET Framework uses a highly componentized, plumbing-free design that enables developers to focus on writing business logic. Developers don't need to write IDL or Registry code, and ASP .NET, for example, includes dozens of controls that encapsulate common programmer tasks such as a shopping cart.
- **Reliability.** The .NET Framework includes technologies to make applications more reliable. For example, memory, threads, and processes are managed by the .NET Framework to ensure that memory leaks don't occur. And ASP .NET monitors running Web applications and can automatically restart them at administrator-defined intervals.
- **Mobility.** The .NET Framework makes large advances with regards to mobile devices. Integrated support for standards such as HDML and WML mean that applications built on the .NET Framework can target even cellular phones.
- **Manageability.** The .NET Framework goes to incredible lengths to make it easy to deploy, run, and manage applications. From the "no-touch" deployment features that eliminate "DLL Hell," to the instrumentation built into every application designed for the .NET Framework, never has it been easier to keep applications alive and well.
- **Security.** The .NET Framework includes an evidence-based security system designed for today's Internet environments. By collecting evidence about where an application came from, who created it, its digital signature, as well as what the application is trying to do and combining that evidence with a security policy, the .NET Framework's runtime environment can make very fine-grained decisions about whether to run an application. It can even "negotiate" with the application, for example, denying it the right to write to a protected directory and enabling the application to choose whether it will run without that permission.
- **Internet.** The .NET Framework was designed from the ground up to be an Internet computing environment. Its deep support for Web services using SOAP gives unparalleled interoperability with non-Windows environments. Microsoft is committed to the vision of delivering software as a service, and Web services are central to that vision. Web services are applications that expose their features over the Internet (or intranet) using standard Web protocols such as HTTP and XML. The .NET Framework makes exposing an application's features like this as easy as writing a single word in a developer's source code.
- **Usability.** Windows 2000 focused on usability by end users. The .NET Framework brings that usability to all developers. It supports the integration of multiple programming languages in a way unimagined previously, enabling developers to choose the right programming language for the task at hand. All programming

languages target a single, extensive, and extensible set of class libraries and are therefore able to integrate with each other as never before.

- **Data Access.** As the rest of the .NET Framework was redesigned for the Internet, so was the data access system. ADO.NET, a significant rework of the ADO model, is designed for today's Web-based style of data access using XML and prepared for "disconnected" use with a high-speed local cache.
- **Hardware.** The .NET Framework is designed to support not just the hardware of traditional Windows operating systems, but, with its sibling the .NET Compact Framework, also other smart devices such as mobile phones, enhanced televisions, and retail devices such as registers. The .NET Compact Framework is designed for downloading and running applications securely on a variety of CPUs and operating systems.