



To .NET or not .NET

White Paper

Abstract

This paper discusses the advantages and draw backs of the .NET technology

To .NET or not .NET

Index

- ❖ Introduction and Overview
- ❖ .NET – Definition and Back ground
- ❖ Advantages of .NET.
 - Interoperability and Mobility
 - Reliability and Security
 - Opening the World of Wireless
- ❖ .NET from a Developers Viewpoint
 - Flexibility of Language and Development Protocols
 - Security and Accessibility
 - Implementation and Deployment
 - Code Reuse and Scalability
- ❖ Disadvantages of .NET.
 - Performance and Integration
 - Operational Issues
 - Support
- ❖ Determining Factors in the Decision

Introduction and Overview

The internet and e-business has touched on almost every aspect of modern life. From entertainment to research to telecommunications and even into the realm of our personal lives, the computer and its connection to the internet has become a critical part of almost every modern home. When one thinks of the internet, it seems to be an almost inexhaustible place of complexity and wonder. The scope of the size of the internet is vast but it is not a stagnant place. How the internet works and interacts with its large audience and clientele is continuously evolving. That complexity is a challenge to all who hope to participate in the cyber marketplace for profit.

Within the technology and I.T. industries the impact of the internet is even more profound. It is out of the question any longer to design a new application or software package without having the web presentation and interfaces to the internet as a cornerstone of the design. It is a small wonder that when the landscape of the internet changes significantly, the need for developers and planners to adapt to that change is critical.

In this paper we will discuss the change in philosophy that is underway in internet development currently and the tools and resources that are available to exploit this change and adapt to it successfully. One such tool is the simply named development tool set called .NET (pronounced Dot Net).

This document will examine .NET as a potential solution for developing in the new internet environment. This white paper will define what .NET exactly is and then go on to examine its strengths as a development tool. From there we will discuss the weaknesses of .NET as well as some other alternatives and attempt to provide as much information as possible to influence the decision on whether or not to use .NET.

.NET – Definition and Background

Before moving very far, it is necessary to get a feel for what .NET is. As a Microsoft product, the product literature is helpful. Microsoft describes .NET as a *“platform for XML Web services, the next generation of software that connects our world of information, devices and people in a unified, personalized way.”* This description doesn't really provide a definition that can be useful in a pragmatic way.

The change that is underway in how the internet fundamentally works can be summed up in the word “interoperability”. Strangely the concept of interoperability did not begin with Microsoft but with Sun Microsystems which gave us the Java development tool that has had such a profound affect on web development. The principle of interoperability changes the fundamental transaction that occurs on the internet.

The fundamental transaction in the past has been the download. In essence, every time a web page is addressed, the page and all the objects you need to “surf” the site are downloaded to the local computer. Data that might be needed by the user is downloaded as it is viewed.

However, in the world of interoperability, the transaction depends not on the user computer but on the site. Instead of the browser bringing the programs and data to the PC, the user executes the programs and views the data on the hosting site. The outcome is profound. Because the “work” of the visit to the site is done on the host computer and the stress is taken off of the user equipment, the customer is free to utilize the web site from any location. As distributed computing becomes a crucial model for accessing the web utilizing remote systems, PDA devices, managed hot spots and laptops, this approach is perfectly poised for the way the new internet audience uses the web. Some of the ramifications of this approach are:

1. The Internet becomes an extension of the customer’s computer
2. Using the web becomes network centered rather than desktop centered
3. Software will not be sold any longer. Instead it will be marketed by subscription.
4. Your email, your information, your data and your profiles will all be hosted on the web so you can access them from anywhere.
5. Everything including storage will become a web service.

It is within the context of this change that the discussion of not only whether to use .NET but how to take advantage of .NET becomes relevant. But all of this does not clearly define what .NET is. Further information from the Microsoft web site is of some value in which Microsoft defines the tool in this way:

Microsoft® .NET is a set of Microsoft software technologies for connecting information, people, systems, and devices. It enables a high level of software integration through the use of Web services—small, discrete, building-block applications that connect to each other as well as to other, larger applications over the Internet.
<http://www.microsoft.com/net/basics/whatis.asp>

.NET utilizes the power of XML which is a sophisticated version of the well known HTML language. XML makes it possible to move the majority of the data, links, objects and interfaces needed to run a modern web site to network centric locations. While XML has been around for a long time, it has been the integration resources that .NET brought to the table that made the realization of interoperability possible.

Advantages of .NET

Interoperability and Mobility

There are two types of advantages to using .NET, the advantages to the computer user and the advantages to the developers, i.e. the suppliers of services to the internet population. To the users of the internet .NET enables the user to evolve in their use of the internet away from a single desktop orientation to true interoperability and mobility. Because the user can access any of their accounts, any of their preferences, email accounts, data and common web services from any platform at any time, the user is freed to use technology that drives today's computing marketplace e.g. mobile communications technology.

Reliability and Security

Any time such a change might come along, reliability and security are a primary concern. .NET has mitigated that concern by building a secure layer at the .NET utility level thus relieving the application level of security concerns. In this way .NET permits users to sustain a high level of comfort and confidence in their on-line transactions while using their web services in a mobile fashion. Such a seamless migration of a large customer base with no anxiety or drop in customer confidence is of immeasurable value to the business entities servicing that clientele.

On Line Forms

.NET has made possible a move away from paper toward the "paperless office" that has been talked about so often. Many of these advantages have already become a reality as the computer user has become accustomed to online filing of government forms and the like. These services and innovations can be made available through the power of standardization and the array of programs and functions within .NET that empower such development. *Winforms applications* that are a standard part of the .NET environment are what make these kinds of services possible for the developer to offer.

Opening the World of Wireless

Wireless applications have exploded in popularity since 2000. An amazing variety of applications have become available including handheld messaging, SMS, digital transmissions, laptops, Instant messaging plus the expansion of the concept of internet usage from the traditional home or office computer to distributed access in airline terminals, retail outlets, even outdoor venues.

.NET is an ideal tool to take advantage of this market. Because the communication and integration between application level and the constantly changing internet customer level is handled by .NET, the developer can design and implement services with the security that the application will be accessible to all compliant wireless systems today and that future introduction of technology will simply merge into the communications channel with no need for change at the application point of service.

.NET from a Developers Viewpoint

Flexibility of Language and Development Protocols

.NET delivers to internet application developers significant advantages and resolves headaches that have been troublesome to developers for some time. At the top of the list is that .NET facilities *seamless use of any language or protocol for development*. Because all applications communicate through the .NET gateway, the programming engine that delivers those standardized message packets to .NET is not important. Therefore the developers are freed to use whatever tools and languages are most beneficial to each project rather than be constrained to tools that service the web interface.

Security and Accessibility

The framework of .NET provides resolution to an array of common other development concerns. Security issues are less troublesome at the applications level because they have been verified as sufficient at the .NET level of interaction. By opening the door to all manner of workstation or destination hardware, .NET vastly expands the accessibility that an application might enjoy thus maximizing the development investment in ways never before possible. Overall the flexibility that .NET offers combined with the power that the standardization that .NET introduces makes .NET an outstanding developer's resource for the introduction of applications that will truly serve the modern internet environment.

Implementation and Deployment

Because .NET has been gaining momentum through the internet industry for a sufficient period of time much is known of the values it brings to the development process. By all measurements of .NET deployments and project completions compared to comparable projects under other environments, .NET projects are completed on or before schedule and on or under budget. Further due to standardization that is inherent in the tool,

deployment complexities and 11th hour crisis that so often plague applications roll outs are significantly reduced in .NET projects.

A new developer resource that has changed the landscape of implementation and deployment for designers is the .NET innovated development tool of C#. C# brings with it the development orientation of managed code and metadata which is a concept that predates .NET but comes into its full utilization in the .NET environment. Metadata introduces an interpretation layer between code and data. In this way .NET can control the code. Self describing metadata which is the structure of C# makes it possible for code to be deployed in a diverse setting of applications, environments and operating systems and it makes code immune to obsolescence due to technology evolutions because the onus of tracking such change is within .NET not in the code itself. This is an extraordinarily efficient design principle which is the heart of the power that .NET brings to the development, implementation and deployment setting.

The .NET file and code organization system working within the metadata parameters is ideally suited for using InstallShield as a deployment device. InstallShield has become a de facto standard in internet installations and one that has become commonplace to customers. That good will and trust can be capitalized on by any .NET deployment and the history and state of the art design of the InstallShield methods reduces or virtually eliminates the risk of a wide spread deployment of an application. By integrating InstallShield, the old barriers to widespread deployment of an application across platforms and operating systems is eliminated which is an unheard of strategic advantage for any product development undertaking.

Code Reuse and Scalability

Prior to the .NET change, the most powerful overhaul of the computing paradigm was the concept of object oriented systems design and programming. .NET with the use of metadata and self defining code takes that concept to a more sophisticated level. With object orientation, code itself was reused because it was deployed in modules in which the entry point and exit point of the module was standardized so any new application could utilize that code as long as the standardized "hooks" into the object are observed and utilized correctly.

.NET expands that efficiency to the code level so the environment is so empowered to manage and utilize the code because the data and coding itself is designed for reuse. How this is achieved is opposite from the traditional thinking of systems design. Instead of an application being developed with one or many platforms in mind, the code itself is neutral

and the interface, .NET itself, facilitates the integration to many environments.

Disadvantages of .NET

Performance and Integration

As with any new technology, there are still performance and integration issues that are still being worked out. However as .NET deployment becomes more widespread, integration will become more and more streamlined. If the planning for integration with in house and third party applications has a priority in the preparation stage of the project, the risks during implementation can be managed.

Operational Issues

From an operational perspective, the capacity and ongoing cost to support .NET at the server level will by nature be higher. .NET moves the center of processing from the desktop to the web server of the service provider to facilitate open communication to mobile devices and greater interoperability. But placing that new load on existing servers does not come without a cost. Early in the decision making process, a capacity analysis and forecast is in order to ascertain to some level of reliable detail the impact the move to .NET will have on existing server load and if there are upgrades needed, just what the new technology will cost.

Support

Finally a pragmatic look at *support and ongoing operational issues* in a .NET environment is in order before the decision to move to that technology is implemented. Application development is measurably more complex in the .NET world. The benefits of .NET far outweigh the difficulties of the transition but the impact on development costs and time frames must be factored in. From an operational support point of view, because Web Services is so much more distributed and carries with it the baggage of a higher applications interdependency, performance problems and problem resolution becomes significantly more difficult. The result on up-time percentage, expectation of swift problem resolution from a help desk perspective and on Service Level Agreements cannot be overlooked even as early as the design stage of application planning.

Determining Factors in the Decision

There is no question that a move to .NET holds promise of great rewards to the developers. The commitment Microsoft and other key industry players have made to the technology broadcasts that this will be the model for future internet application functionality. If that forecast is accurate, and all indications are that it is, the decision to move to .NET development is more of a “when” question rather than an “if” question.

However, each of the levels of concern just expressed should undergo a complete and “brass tacks” review from management before the decision is made to move forward. It pays to “count the costs” and go into this new venture with everyone’s eyes wide open rather be surprised by difficulties that such a shift in design philosophy will most certainly bring to a shop.

Once the analysis and cost benefit studies have been done, the decision to “go or no go” with .NET will become more clear. That level of analysis takes a time and effort to complete. *The resources of Tometa Software can provide assistance* in moving that study in the right direction as well. However, once the effort has been made and if the analysis was done properly and thoroughly, all concerned with the final decision will have a clear understanding both of the real world costs and problems that will have to be conquered and of the tremendous benefits a well executed migration to .NET will return and continue to return in the months and years ahead. It will take bold and skilled leadership to take an application base in the direction of .NET but the rewards and promise of greater market growth and penetration will almost certainly be greater than the investment and the business entity will benefit from the decision to begin now developing in .NET as it truly begins to spread and become the operating environment for the internet of the future.

This document is for informational purposes only. Tometa Software, Inc. MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

© 2004 Tometa Software, Inc. All rights reserved.