RAPIDSOFT SYSTEMS

Rapidsoft Systems
redefining business relationships

A Rapidsoft Systems' White Paper

© September, 2008

On Web Development Technologies: .NET or PHP

Anyone, who works in the web development or manages web development, knows the hottest, and perhaps never ending debate is PHP vs. Microsoft's .NET. It is like religious divide that is hard to overcome – both sides are adamant that they have the best technologies for the web development. As someone deeply involved in the development of a number of Web Application, at Rapidsoft Systems we have to deal with this question almost on the weekly basis. So, here are our thoughts on the issue.

But, before we delve deeper in more details – it is important to understand the core features of each technology.

If you have no familiarity with .NET, not a problem. No one seems to exactly know why even it as called .NET except that in the days of Internet boom of early 2000, having anything with dot (.) and NET in the name of company or product made a lot of people very rich. So, we can't really blame Microsoft for it to cash on it.

The first, Active Server Pages (ASP) was a scripting technology for building dynamic web pages on Windows Server. It was moderately successful in its days. In 2002, Microsoft revamped ASP, and added a lot of new functionality, made it object oriented and called it ASP.NET. ASP.NET relies on a broad web development framework called .NET. In the technology Industry, the word "framework" means that developers of a technology are either way smarter than average "Joe" and want to emphasize this very boldly by calling it a "framework", or they don't really have a clue what they were trying to develop. In a layman's term, we can say that "Framework" means a collection of "diverse and varied" tools and technologies that cannot be given one name, but are collectively designed to solve a particular set of problems. Each tool or technology can be called as a Framework "Component". If this definition doesn't suit to describe any "Framework" you deal with, then assume the second rationale for calling it a "framework". Let us now go back to the .NET framework.

Using our definition of the Framework in the preceding paragraph, we can deduce that the .NET framework refers to all the tools and technologies for building Windows Desktop Applications, Web Applications and Web Services on a Windows Platform. Here are the key features of ASP.NET:

1. ASP.NET supports a wide variety of languages – Some say over 40 languages. The most popular choices are C# (C 'Sharp') and Visual Basic (VB). We don't care what remaining 38 are.

2. ASP.NET pages are “compiled” not interpreted. In ASP.NET, the server converts the code to a compiled Microsoft Intermediate Language (MSIL) code (similar to Java runtime byte code) after the first run, this allows subsequent calls the run much faster. Microsoft's .NET Common Language Runtime (CLR) provides the runtime environment to execute the intermediate code.

3. ASP.NET has full functionality of the .NET Framework. What it means that basic functionality for handling XML, Web Services, database integration, email handling, and dealing with regular expression for forms validation is all provided.

4. ASP.NET is an Object oriented technology with a number of useful classes already provided. This means that learning curve is a bit longer, but once you know the pre-built functionality of the framework, the development work can move very fast.

5. ASP.NET is full functionality Web Design framework with support for Web Services, AJAX, Remoting, WCF and Generics. And, with new release you can even use SilverLight. Oh –well this is too much jargon in one sentence so we will try to explain these fancy terms further.

Remoting: NET Remoting provides a way for application in different machines/domains to communicate with each other. Remoting provides a powerful yet an easy way to communicate with object in different app. Any object which executes outside the app domain can be considered as Remote. This enables your applications to take advantage of remote resources in a networked environment.

Remote objects are accessed via Channels, Channels can be thought of as Transport mechanism to pass messages to and from remote objects. All the method calls along with the parameters are passed to the remote object thru the channel viz. HTTP or TCP.

Web Services: The Web services technology enables cross-platform integration by using HTTP, XML and SOAP for communication thereby enabling true business-to-business application integrations across firewalls. Because Web services rely on industry standards to expose application functionality on the Internet, they are independent of programming language, platform and device.

Windows Communication Foundation (WCF) Services: The Windows Communication Foundation, which was code-named Indigo, provides .NET class libraries, tools and hosting facilities for enabling software entities to communicate using any protocols, including those used by Web services.

Generics: This is more to do with programming flexibility. We can refer to a class, where we don't force it to be related to any specific Type, but we can still perform work with it in a Type-Safe manner. A perfect example of where we would need Generics is in dealing with collections of items (integers, strings, Orders etc.). We can create a generic collection than can handle any Type in a generic and Type-Safe manner. For example, we can have a single array class that we can use to store a list of Users or even a list of Products, and when we actually use it, we will be able to access the items in the collection directly as a list of Users or Products, and not as objects (with boxing/unboxing, casting).

Now that we have give a lot of details about ASP.NET, we will move our focus too PHP, the Open Source Queen (or the King) of the Web development.

PHP is primarily a scripting language with layers of functionality added to develop web application over the years with each release. It started as a custom scripting language to handle Web Transactions, but has grown into a powerful tool with objected oriented features added later on.

One main feature of PHP is that it is an untyped language. This means that a variable can hold any datatype. It also means that a variable may be assigned data of one particular datatype and then later be given data of a different datatype. For example:

```
var x = 'Shoes';
x = true;
x = 1;
```

This can be a great feature, especially for small projects, since a developer has tremendous freedom in developing applications. However, this can be a great pain to debug in a large project if the things go wrong.

One major issue with PHP, and that can drive some people nuts is that names of functions are not case-sensitive but variables are; built-in functions are not consistently named; and no real structure is enforced on PHP developers, making it easy to write messy code. On the other hand, lack of this structure can be seen as a quick and easy way to achieve what you want to do and move on to the next task. If you are coming from PERL or UNIX scripting world, PHP is easy to pick up in a few days (or at least, reasonable set of its core functionality).

Another, great strength of PHP is its strong function libraries and modules. If there's something you want to do—such as creating PDFs, Flash SWFs, and many image formats and handling e-mail—libraries are likely already available to help you do it.

Arguments For and Against

Let us first summarize our description of two web technologies. ASP.NET is:

Strongly Typed, Object Oriented, Sandboxed, Multi-Syntax, Component Centric, Event Driven, forms oriented, pre-compiled experience.

PHP is:

Loosely typed, objects optional, fixed syntax, component-less, runtime interpreted, structured programming model.

Arguments in Favor of ASP.NET

There's no doubting ASP.NET, and the .NET framework as a whole is a very robust and consistent framework, with a choice of languages and consistent object set. If you're a programmer by mindset and have been in using OO programming, you are going to love the extensive capabilities of ASP.NET. There is nothing that you can't do with .NET that you can do with any other Framework or tool for web development.

- .NET is excellent for enterprise-level web and application development, particularly for large

corporations and complex multi-layered transactions oriented applications (e.g., major banks, insurance companies etc.). It allows better separation of design and application logic using of code-behind pages and user-controls.

- Needless to say, advanced functionality of ASP.NET takes some time to learn and master. Therefore, it comes with substantial learning curve, architectural complexity.

- ASP.NET and PHP are based on completely different architectural philosophy. Beside the oft-repeated scripted vs. compiled argument, ASP.NET (WebForms) is event-driven. Objects have events and code can handle those events. For example, when a page is loaded an event is fired. When a button is pressed an event is fired. The entire page lifecycle is made up of events. In addition, the ASP.NET Web Forms environment is an abstraction made up entirely of events. This abstraction attempts to hide HTTP from the developer.

-There are a lot of dull tasks that Microsoft has made easier. Think of .NET as a set of readymade widgets for the web developments. Somebody wrote for you and made it all available at one place. You can create/enable a login system within five minutes (database included).

- Performance is very good. One can build very efficient large sites (myspace.com and monster.com are two that use .NET).

- While not an open-source project, community support is excellent and Microsoft does a great job promoting their product. They've also create video tutorials for people that would rather see than read. (http://www.asp.net/learn/).

- ASP.NET has a strong provider model. Don't like how they handle authentication, site maps or profiles? That's fine. Create your own or just inherit from theirs to modify a few small things.

- You can write ASP.NET applications in quite a few languages. Microsoft supports their languages out of the box (VB.NET, C#, etc..), but you can use others. Python, is around. MS is working on a Ruby implementation for .NET. Somebody has also created a PHP compiler. Most of this is Microsoft gimmickry and for bragging rights because most .NET web applications, any way, are actually in C#.

-ASP.NET has some extremely powerful developer friendly features like Postback, Reuseable components, Master pages and Caching etc. Just check any .NET book or tutorial to know what these terms mean. This makes development go very fast for certain complex applications.

- There are a lot of big corporation, specially banks and governments that use ASP.NET extensively. This makes finding a paying job a lot easier. PHP, being a choice of education sector and small businesses that don’t want to spend or have too much money on web efforts, offers jobs that likely to be not that well paid.

ASP.NET Negatives:

- ASP.NET is a single platform environment. You have to buy Windows Servers. This means that your deployment costs are higher than LINUX environment. That could be a major factor when you take cost of deployment especially for a large infrastructure.

- There is big and long learning curve, especially if the developers haven't previously worked in OO environment, and even otherwise. With that comes the cost of developers.

Arguments for PHP

- PHP is easy and free to get the basic tasks done quickly and easily. As I mentioned earlier, you can learn PHP in a few days due to its basic scripting language structure and built in functionality, especially if you have a back ground in UNIX or PERL scripting.

- For simple sites – PHP provides easy way to handle forms and generate HTML pages from database queries without the need to learn a complex proprietary framework. It was primarily designed to provide interface between HTML forms and the backend database, so programming requirements for simple database driven web operations are minimal. You can quickly finish the tasks with PHP.

- Multi-platform – you can run PHP on LINUX, or on Windows Server. That gives tremendous flexibility to change your hosting as needed.

- For PHP Development, you can make use of many free IDEs available that are very impressive and well supported (e.g. Eclipse)

- PHP compilations involve the PHP engine changing it to the executable form of the page (similar to how Java uses byte code in its "compiled" state). Furthermore, the only way to then achieve similar speed to ASP.NET is to CACHE the "compiled" version of the PHP scripts. This is how the Zend Accelerator works.

- If you prefer Object oriented programming, PHP 5 adds that functionality too.

- Error handling in PHP is greatly improved in PHP 5 with exception handling is quite similar to any other programming language.

Negatives of PHP

- Many developers, especially with .NET expertise and experience, think doing large projects with PHP can be an issue. This is very subjective judgment because there are many large sites that have been developed with PHP too.

- Some developers consider PHP, a bit weak in the areas of Web Services support compared to ASP.NET. PHP 5 has added Web services support, but perception remains in the mind of some.

- Most PHP IDEs require lots of add-ons in-order to add similar functions to Microsoft Visual Studio for debugging Web Applications.

- No built in support for AJAX like ASP.NET. Requires add-ons and client side programming. This is an area where PHP indeed cannot do much being a purely server side scripting language. But, you can develop the AJAX based applications with PHP also quite easily.

Contrasting PHP with ASP.NET

PHP, as mentioned, is certainly a very good web programming language as long as you program in PHP 5 and up. PHP, until release 5, lacked in the error reporting department as many times when source code had errors, PHP simply error-ed out and died. With .NET Technology as well as with PHP 5, you can now gracefully error out as custom error mechanisms work before the standard handler. So, both are now equal.

.NET technology also has many integrations between different platforms and many libraries that will take integration tasks easy. There is heavy integration between SharePoint and other Microsoft platforms which makes application development changes very easy to deploy system wide. On the other hand, PHP too has numerous scripts readily available for various tasks in the open source world. However, finding suitable scripts, testing them and modifying for your own use takes some efforts. I would say ASP.NET wins here.

As far as security goes, .NET makes an attempt to help developers create more secure applications as input processing is handled by the framework. You can also add to the input processing routines to attempt to filter additional input. PHP has a number of input validation frameworks but lack consistency unlike ASP.NET.

PHP 5 has added a lot of the class and object security features, so in this regard, they are now very equal. It just comes down to the developer, as each platform has its ups and downs.PHP has become an Enterprise language over time and many enterprises (including Yahoo! and Google) are creating custom platforms using PHP as their base. However, .NET has many more libraries so developers aren't reinventing the wheel. Many of the libraries are very stable, whereas PHP's frameworks are scattered with no conformity. That is true for any open source software tool where you really have to work hard to search and find that is not part of the basic environment.

Common Sense Approach to Web Development

There are many factors that must be considered while making decision about which web application framework to use. What kind of developers are available in the company, size of the project, budget availability for hosting, your desire to use open source software and save on host charges etc. So, don't be bogged down by pure technology debate as both can get the work done.

Remember that it is hard to match the ASP.NET in depth and breadth of features including the versatility of Microsoft Visual Studio IDE. There is nothing that you can't do with .NET with utmost ease; in the hands of knowledgeable developers, it can facilitate very rapid development of complex web applications, but it costs more money to deploy ASP.NET solutions. And, you have to buy multiple Microsoft Server Licenses (multiple development, staging and deployments servers) and Visual Studio Licenses for your developers. These licenses aren't cheap and Microsoft keeps changing software every so often forcing you to buy upgrades every few years. Also, note once a .NET application – forever a .NET application (except doing a total rewrite). Are you ready to deal with this recurring cost of Microsoft products? Does revenue generated by your web applications or other business operations makes this cost to be irrelevant? If not, we suggest stick with PHP.

PHP which is in its 5th revision now, is an Open Source web development language that also facilitates the creation of feature rich, dynamic websites that can use all kind of databases. There are tons of readymade scripts that you can download and customize for your own use. This makes PHP extremely attractive to the independent web developers and for small and medium businesses that are very cost conscious.

All in all, ASP.Net and PHP are both excellent options, offering basically the same functionality but somewhat different way of doing things. No matter how you reach your decision whether based on the cost of initial investment, or flexibility of open sources, the end result depends upon the quality of engineers that write the code. We, at Rapidsoft Systems, have excellent teams of engineers comfortable both in ASP.NET and PHP, and we have done complex portals in both technologies. After all, who cares if the site works as designed.

For more information and specific questions, please contact us at:

Rapidsoft Systems, Inc,
Mailing Address: 7 Diamond Court, Princeton Junction,
New Jersey 08550, USA

www.rapidsoftsystems.com

Phones: **1-609-439 9060 (US East Coast, NJ Office)**
1-408-829-6284 (US West Coast, San Jose Office)
Toll Free (North America): 1-800-946-5490
Fax: 1-831-855-9743

Email: info@rapidsoftsystems.com