

## **ASP.net VS PHP5**

*Von Äpfeln, Birnen und der Auswahl der geeigneten Plattform.*

### **Prolog**

„Ja, das waren noch Zeiten als das WWW aus rein statischen Seiten bestand. Man sprach bereits von Dynamik wenn die Webseite nervös zuckelnde animierte GIF-Animationen beinhaltete. Und wenn es nur ein Baustellenschild war. Sie erinnern sich?

Nun, Zeiten ändern sich, das Internet wird erwachsen. Webseiten werden zu richtigen Applikationen. „

### **Web, Web, Hurra!**

Als das WWW immer populärer wurde, merkte man schnell dass man mit reiner Statik nicht mehr vorankam. Die neuste Browsergeneration unterstützte interaktive Elemente wie Auswahlboxen und Eingabefelder. Es galt also die vom Benutzer eingegebenen Daten irgendwie zu verarbeiten. Dies konnte nur serverseitig, aufgrund des zugrunde liegenden Protokolls, geschehen. Es musste eine Technik geschaffen werden, um die vom Server entgegengenommenen Daten irgendwie zu verarbeiten. So entstand das CGI (=Common Gateway Interface) welche es ermöglichte kleine Programmfragmente (geschrieben in den unterschiedlichsten Sprachen) einzubinden. Anfänglich nahm man hierfür bereits existierende Programmiermöglichkeiten. Häufig waren Perl, C(++) und sogar Unix-Shell-Skripte als CGI Anwendungen anzutreffen. Doch diese brachten häufig nicht benötigte Flexibilität mit sich, kein Wunder, dafür waren diese Programmiersprachen nicht entworfen worden. So traten neue Programmiersprachen an, um das World Wide Web zu erobern.

In diesem Artikel wenden wir uns PHP, in der Version 5 und ASP.NET zu und vergleichen somit ganz klar Äpfel mit Birnen. Die Konzepte welche beiden Systemen zugrunde liegen, könnten unterschiedlicher nicht sein. Während es bei PHP in erster Linie darum geht, Webseiten Dynamik einzuhauchen (um z.B. Datenbanken abzufragen), sieht Microsoft ASP.NET als eine Technologie um den Spagat zwischen klassischen Applikationen und Web-Applikationen zu meistern.

Unabhängig davon wie die Entwickler Ihre Programmiersprachen sehen: Die meisten Aufgaben sind mehr oder weniger mit beiden Systemen zu lösen, die Frage ist, wie steinig der Weg jeweils ist.

### **Äpfel**

Lange nach dem Siegeszug von PHP schmiss Microsoft mit ASP eine moderne Sprache ins Wettrennen um auch auf diesem Gebiet eine entsprechende Technologie anbieten zu können. Kritiker werfen Microsoft vor, das Internet verschlafen zu haben und werten ASP als Bestandteil der Strategie das Internet nachträglich ebenfalls monopolisieren zu wollen. Die Nachteile der ersten Versionen von ASP waren die teilweise seltsam anmutende Syntax und die eindeutige Bevorzugung des Browsers aus dem Hause Microsoft. So galt lange die Devise: „ASP Anwendungen laufen nur auf dem Internet-Explorer gut“. Dies lag daran, dass ASP auch auf Client-seitige Skriptausführung setzte und hauptsächlich Internet-Explorer konformen HTML Code produzierte. Natürlich lag dieser Umstand auch am entsprechenden ASP-Entwickler, ASP förderte jedoch diesen Zustand.

Weiterhin wirkte ASP noch nicht ausgereift, an manchen Ecken und Kanten hatte man den Eindruck noch nicht mit einer Sprache „aus einem Guss“ zu arbeiten. Zusätzlich war die Syntax stark an Visual Basic angelehnt, sodass viele Entwickler (zu Recht) die Hände über dem Kopf zusammenschlugen.

Mit ASP.NET hat sich Microsoft manchen Kritiken angenommen und Abhilfe geschaffen.

ASP.NET ist momentan, wie die meiste Software aus dem Hause Microsoft, wen wundert es - nur unter Windows mit dem Internet Information Server lauffähig. Diese Aussage ist jedoch zeitlich begrenzt, da Microsoft mit dem Einreichen des .NET Konzeptes zur Standardisierung Open-Source Entwicklungen Tür und Tor geöffnet hat. Bei .NET an sich handelt es sich u.A. um ein Konzept wo der ausführbare Code in einem Zwischenschritt maschinenunabhängig übersetzt wird. Erst zur Ausführungszeit wird es im direkten Maschinencode auf der jeweiligen Plattform kompiliert. Weiterhin beinhaltet das .NET Konzept ein umfangreiches Framework (Klassen und Funktionssammlung) für alle erdenklichen Fälle.

Das .NET Konzept (Also auch ASP.NET) ist nicht mal abhängig von der Programmiersprache, es können zur Zeit sowohl Visual Basic.NET als auch C# eingesetzt werden, wobei prinzipiell jede .NET kompatible Sprache unterstützt wird.

Somit werden erste Vorteile sichtbar. Durch die Integration von ASP.NET im Gesamtkonzept von .NET wird dem Entwickler jede Menge Arbeit, zum Beispiel beim Zugriff auf Datenbanken, erspart. Weiterhin stehen aus dem Hause Microsoft umfangreiche Tools (z.B. Visual Studio) bereit um professionell Entwickeln zu können. Damit wird klar, dass ASP.NET durch seine Komplexität nicht jeder Manns Sache ist und ein entsprechendes (.NET) Know-how voraussetzt.

In Verbindung mit dem Internet Information Server skaliert ASP.NET besser, weiterhin ist die Ausführungsgeschwindigkeit um einige Prozentpunkte höher. Dies verwundert nicht wirklich, wenn man bedenkt das ASP.NET letztendlich kompiliert wird, während PHP immer erst zur Laufzeit interpretiert wird.

## **Birnen**

PHP geht hier einen anderen Weg. Angefangen vor langer Zeit als „Personal Homepage Tools“, ist es momentan in der Version 5 zu haben. Als Open Source Software sind keine Lizenzkosten oder Ähnliches notwendig. Vor allem in Verbindung mit dem Apache Webserver, der MySQL Datenbank und Linux (= LAMP, Linux / Apache / mySQL / PHP), hat es sich sehr schnell verbreitet. PHP ist aber nicht auf den Apache Webserver und auf Linux festgelegt. Es kann genauso gut und Windows mit dem Internet Information Server betrieben werden.

Weitere Vorteile liegen auf der Hand. Durch seine C ähnliche Syntax und der Gutmütigkeit wie PHP mit dem Code umgeht ist es sehr einsteigerfreundlich. Im Prinzip kann jeder sofort mit PHP loslegen ohne sich Gedanken um Initialisierung von Variablen und Typisierung zu machen. Damit fördert PHP jedoch den unerfahrenen Programmierer indirekt zum Pfuschen auf, dies ist auch einer der Hauptargumente den Gegner von PHP einbringen.

In der Version 5 ist PHP inzwischen professioneller geworden, so sind bereits Hauptbestandteile einer modernen Objektorientierten Sprache implementiert, die gesamte Sprache an sich wirkt durchdacht.

PHP steht im Gegensatz zu ASP.NET kein Gesamtkonzept zur Seite, jedoch können PHP-Befürworter auf eine umfangreiche Funktionssammlung verweisen. Neben diesen gibt es jedoch nur rudimentäre Ansätze einer kompletten Klassenbibliothek aus einem Guss. Der

aussichtsreiche Kandidat hierfür ist PEAR (=PHP Extension Application Repository), welches direkt von den PHP-Entwicklern empfohlen wird. Dieses Framework bringt bereits das Notwendigste mit (Datenbankabstraktion, Templatisierung, etc....) und sieht zurzeit sehr viel versprechend aus. Man darf also gespannt sein wie sich dies in Zukunft weiter entwickelt.

Auch die verfügbaren Entwicklungstools können momentan nicht mit Visual Studio mithalten, was jedoch eher an dem Gesamtkonzept von Open Source an sich liegen dürfte. Es entstehen schnell viele Projekte welche sich auf die Fahnen schreiben einen Anforderungsbereich abzudecken, jedoch kommen viele nicht über den Beta-Status hinaus. Der Kandidat der den meisten Erfolg verspricht, ist die IDE aus dem Hause Zend, welche mit einem Debugger und Features wie Codevervollständigung den Entwickler tatkräftig unterstützt.

An Visual Studio, der Entwicklungsumgebung Microsofts, kommt diese IDE noch nicht heran, die wichtigsten Features sind jedoch bereits enthalten sodass man angenehm damit arbeiten kann.

Die Tatsache dass PHP nicht kompiliert wird, sondern zur Laufzeit interpretiert wird trägt dazu bei dass man PHP allgemein den Ruf nachsagt, langsam zu sein. Diese Aussage kann man alleine so nicht stehen lassen, durch den Einsatz von Code-Optimierern und Caching-Systeme kann man diesen Nachteil sehr wohl wettmachen.

Allen Nachteilen zum Trotz, können mit PHP schnell und einfach umfangreiche Web-Applikationen erstellt werden, vor Allem wenn ein geringes Budget zugrunde liegt. Auch müssen dafür keine professionellen Programmierer beschäftigt werden, im Prinzip können versierte Homepagebastler bereits ahnsehnliche Ergebnisse abliefern.

### **Was tun, sprach Zeus?**

Der Vergleich zwischen Äpfeln und Birnen zeigt es schon, eine allgemein gültige Antwort zu treffen ist nicht möglich. Eher eine bei Juristen beliebte Antwort: „Es kommt darauf an“.

Wird Wert auf ein geringes Budget gelegt und soll es vor Allem schnell gehen ist sicherlich PHP die erste Wahl.

Ist abzusehen das das Projekt umfangreich zu werden droht, hochskalierbar sein muss und wird vor allem in der bisherigen Infrastruktur bereits auf Technologie aus dem Hause Microsoft gesetzt, sollte man ASP.NET in Erwägung ziehen. Vor Allem wenn die Gefahr besteht dass das Projekt irgendwann sehr umfangreich wird kann man insbesondere mit PHP aufgrund der „gepfuschten Programmierung“ sehr schnell in einer Sackgasse landen. Diese Gefahr ist jedoch durch den bloßen Einsatz von ASP.NET jedoch auch nicht ausgeschlossen.

### **Post Scriptum oder „ich mag aber gar kein Obst“**

Der Fairness halber sollte erwähnt werden das weder PHP noch ASP.NET das non plus ultra bei der WWW-Programmierung darstellen. Perl und weitere, teilweise exotische Programmiersprachen haben immer noch ihre Existenzberechtigung. Häufig, vor allem wenn es um Geschwindigkeit geht, können diese PHP und ASP.NET in die Schranken verweisen. Es kann sich also durchaus lohnen auch andere Systeme in Erwägung zu ziehen.

## **Über den Autor**

Felix-Gabriel Gangu ist Inf. für Softwareentwicklung und freier Dozent. Er beschäftigt sich seit 1999 mit PHP und ASP und hat an zahlreichen Projekten welche PHP und ASP.NET einsetzen, mitgewirkt. Sie erreichen ihn unter [felix@gangu.de](mailto:felix@gangu.de)