A review and analysis of technologies for developing web applications

Asha Mandava and Solomon Antony
Murray state University
Murray, Kentucky

Abstract
In this paper we review technologies useful for design and development of web-based applications. We also discuss about the technologies that are used at the client side and server side of web application. Next, we compare different web application development frameworks. In addition, we discuss life cycle model and framework of web application development.

Introduction
While developing business applications, the IT manager must make often make the important and irreversible decision of which platform to implement the solution (Taudes, 2000). The most common choices were desktop or the web platform. Web application development has been seeing increasing share in the overall IS projects. However, web applications do not render themselves well to application of the traditional systems development cycle model (Gellersen & Gaedke, 1999). The traditional approach also takes lots of time to complete particular web application. There are various types of web applications. Some web applications deliver organizational functionality, some are designed to interactive tools, some other are for communicational dialogue and others are for presentation of information and graphic design (Lang M., 2009). This paper deals with websites that cater to delivery of business functionality to the business users. We start with a description of components of web applications, followed by general trends in development methods. Next, study the components up close and report on popularity, usability and efficiency of the technologies. Finally, we compare three popular web application platforms and make recommendations for typical business applications.

Structure of web applications
Business applications are usually broken into logical chunks called "tiers" (Fernandes C., 2003). For every tier, there are some specific functions. Web applications tend to be multi-tiered by nature, with the most common structure being the three-tiered architecture. In its most common form, the three tiers are (i) presentation layer, (ii) application layer and (iii) storage layers. A web browser itself is the first tier that is presentation tier. By using some
dynamic Web content technologies like ASP, ASP.NET, CGI, ColdFusion, JSP/Java, PHP, Perl, Python, Ruby on Rails or Struts2 middle tier will host the application logic. Finally a database is the third tier storage. The rest of the paper presents and discusses the methods available to web application developers.

**Development practices**

In a 2002 survey of websites in the UK, it was found that most of them had links to backend databases (Taylor, 2002). It was also found that most development work was done in an *ad hoc* manner, without using any specific methodologies. However, later survey finds that most web application development projects use “clear tasks and phases” and use “explicitly documented processes” (Lang M. a., 2006). Before the development starts, modeling of the sites using flowcharts, screen mockups and storyboards is common. Some researchers found that in many projects, no methodologies were employed (Taylor, 2002).

In the 2006 survey, it was found that most web application projects comprised of small teams. More than 90% of the web projects had 10 or fewer developer teams (Lang M. a., 2006). With smaller teams, lighter more agile methods can be employed. It was found a most of them used in-house developed methods. Many of the in-house developed methods were hybrids of various development paradigms. However, popular object-oriented approach was not found to be widely used in the survey sample.

There are a number of benefits of using standards for web application development including (i) consistent design style across websites within the organization, (ii) standardization of web application requirements document, and (iii) guidance for testing the websites (Taylor, 2002). The need for model driven development of web applications is well recognized also (Valderas, 2011).

**Presentation layer**

At the presentation layer, the web pages are rendered to the browser. Traditionally, the webpages contain only HTML code. Now a days, the presentation layer of the web applications provide nearly the same user experience as desk top applications. Such interfaces employ a group of technologies, collectively called Rich Internet Application (RIA) (Driver, 2005). There are four categories of RIA technologies, namely (i) Script based, (ii) Plugin based, (iii) browser based and (iv) web based desk top technologies. In Script based RIA interfaces, scripting languages such as Java script (and its extensions AJAX and jQuery) and CSS are used. Many of the Google web applications use script based RIA techniques. Such client side code is executed or interpreted by the browser plat form. Client side scripts are generally viewable by any visitor who is using that application. (Hanley, 2011) In Plugin base RIA applications, pre-compiled objects written using Adobe’s Flash, Microsoft’s Silverlight, Active X, Java provide the rich user experience. In browser based RIA applications, the browsers natively support interface definition languages. In web based desktop technologies, application written in higher level languages directly interact with the web server without using a browser (Bozzon, 2006).
Application layer

There are two ways the application layer can be built – using server side scripting languages or compiled business objects. Server side scripting languages include ASP, PHP, JSP, ColdFusion, perl, Ruby, WebObjects and Python. The server side scripts are typically interpreted and run by the web server and are is not visible to clients (Hanley, 2011). PHP is a very common Server Side Scripting language. It is Linux / Unix based and Open Source with free redistribution, It is usually combined with MySQL DBMS. Zend Framework is a popular Object Oriented Web Application Framework that employs PHP. ASP is from Microsoft and runs on IIS web server. ASP.NET is the successor of ASP. ColdFusion is Adobe's Web Application Framework. Ruby on Rails is Ruby programming's Web Application Framework. It too allows free redistribution. Perl is a general purpose high-level programming language and Server Side Scripting Language. It lost its popularity to PHP. Python is a general purpose high-level programming language and Server Side Scripting language - free redistribution.

While both PHP and JSP are popular and efficient languages for coding the server side logic, JSP outperformed PHP in many e-commerce and banking application tests (Trent, 2008), however their performances were not significantly different on a bookstore simulation exercise (Cecchet, 2003). Between WebObjects and ASP, WebObjects technology is rated to be a better choice for development and maintenance, although it has a steeper learning curve (Al-Ghourabi, 2006).

The compiled objects that capture business logic are typically stored in program libraries and are very useful to the developer. Program libraries contain collection of commonly used functions, classes or subroutines. With these libraries, developers can easily add or edit functionalities to a frame worked or modular type application.

Comparison of three popular platforms in development:

Development of web application can be done in so many plat forms, next we compare the three most popular platforms namely Linux / Apache / MySQL / PHP (LAMP) vs. Microsoft’s ASP.NET vs. Sun’s Java 2 Enterprise Edition (J2EE) flat forms. The results of the comparison can be found in Table.1. (Hanley, 2011)

<table>
<thead>
<tr>
<th>Criteria</th>
<th>LAMP</th>
<th>ASP.NET</th>
<th>J2EE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Licensing cost</td>
<td>No licensing cost</td>
<td>Expensive licensing cost</td>
<td>No licensing cost</td>
</tr>
<tr>
<td>Platform(s)</td>
<td>Multiple</td>
<td>Only Windows</td>
<td>Multiples</td>
</tr>
<tr>
<td>Hardware cost</td>
<td>Runs on very inexpensive servers</td>
<td>Requires slightly more expensive servers</td>
<td>Requires expensive Servers</td>
</tr>
<tr>
<td>Staffing</td>
<td>Somewhat difficult to find qualified people in this domain</td>
<td>Very easy to find qualified people in this domain</td>
<td>We can find qualified people reasonably</td>
</tr>
<tr>
<td>Security</td>
<td>Good</td>
<td>improved</td>
<td>Good</td>
</tr>
<tr>
<td>Performance</td>
<td>Very good</td>
<td>Often requires more expensive hardware to perform well</td>
<td>Often requires substantial configuration and expensive hardware</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------</td>
<td>-------------------------------------------------------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>Scalability</td>
<td>Scales very well</td>
<td>Difficult to scale</td>
<td>Scales well when configured properly</td>
</tr>
<tr>
<td>Administration</td>
<td>Difficult: Often requires reading documentation and editing text files</td>
<td>Easy: Often can be done through point and click interface</td>
<td>Moderate: Sometimes can be done visually</td>
</tr>
<tr>
<td>Configuration flexibility</td>
<td>Can be difficult to configure properly</td>
<td>Easy to configure</td>
<td>Moderately difficult to configure</td>
</tr>
<tr>
<td>Frameworks</td>
<td>Many available often difficult to choose</td>
<td>One standardized framework</td>
<td>One standardized framework</td>
</tr>
<tr>
<td>Components</td>
<td>Widely available</td>
<td>Widely available</td>
<td>Widely available</td>
</tr>
<tr>
<td>Compatibility</td>
<td>Very good: New versions usually backwards compatible</td>
<td>Moderate: New versions often break functionality</td>
<td>Bad: Many problems between old and new versions</td>
</tr>
</tbody>
</table>

Table.1

From the contents in Table 1, we can make the following recommendations

1. If the project has short budget, then use LAMP platform.
2. If security of application is most important, then use LAMP or J2EE platform.
3. If the project has to be in more compatible mode, then use LAMP platform.
4. If the project is large, then use ASP.NET platform.
5. If clients want configuration flexibility, then use ASP.NET platform.
6. If client wants high performance project, then use ASP.NET platform
7. If the project has more time and high budget, then use J2EE.

**Conclusion**

Object oriented approach build web applications very efficient when one can accomplish more in less time. Because it uses modern processes, by this both developers and clients can benefit. To develop these types of applications there are so many scripting languages and new technologies are there we don’t have to stick to one. Comparison of most popular plat forms in Table.1 give good knowledge to the developers as well as clients in choosing of web application platform.
Bibliography


