

# Issues and Challenges in Selecting Content for Web Design Courses

*Azad Ali and Scott Mensch*  
*Indiana University of Pennsylvania*  
*Indiana, PA, USA*

[azad.ali@iup.edu](mailto:azad.ali@iup.edu); [smensch@iup.edu](mailto:smensch@iup.edu)

## Abstract

This paper discusses the pedagogical considerations and the steps that have been taken by faculty members within a technology program to update their web design courses. The department of Technology Support and Training (TST) – Eberly College of Business and Information Technology (ECOBIT) at Indiana University of Pennsylvania (IUP) has been updating their courses in response to technological changes and market demand. This study outlines the processes that were followed and the pedagogical considerations that have been taken into account when updating two of the TST's web design courses. The paper explains first the challenges that web design educators face when selecting content for their courses. The paper then outlines how these difficulties were addressed when designing the courses at the TST program.

**Keywords:** Web design course content; Web design curriculum, Courses in web design, Pedagogy of web design course, Content of web design courses

## Introduction

The rapid development of web based technologies has created excitement among educators of web design curriculum. This excitement may sometimes result in a rush to introduce the latest technologies into the classroom without fully studying the need and impact of the technology. While the inclusion of new technologies can be beneficial in many ways, the rush to update the curriculum may produce unforeseen and unplanned dilemmas. Educators may need to address the potential pitfalls associated with updating courses too rapidly to avoid costly mistakes. Goldweber, Impagliazzo and Bogoiavalenski (1997) noted about this rush of updating technology curriculum:

Rapid changes in computing often motivate educators to introduce innovations in the curriculum and the classroom. The haste to do something new or adopt some current fad can cause teachers to overlook possible adverse effects of these innovations on students and the profession. The deployment of

curricular or pedagogic innovations such as new languages and technologies may seem appropriate, but mistakes are costly. (p. 94)

---

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact [Publisher@InformingScience.org](mailto:Publisher@InformingScience.org) to request redistribution permission.

The process of selecting topics and coverage areas for web design courses differs with other technology courses in a number of ways. A study conducted by Vebyla and Roberts (1998) contrasted

the topic coverage areas between web design in Computer Science (CS) and web design in Information Systems areas (IS). This study defined the topics covered in IS and CS courses “single paradigm” and the topics in web design courses as “multi-faceted”. The same study explained the predicaments that web design instructors go through in selecting topics for their courses:

A topic area that necessarily embraces a more complex, multi-faceted execution model is that of web technology. For such a topic, the course designer must find a way of incorporating sufficient depth in all of the areas without generating too much content. The teacher of topic must ensure students gain sufficient grasp of the paradigms so that they can understand the workings of web design and how different components interact to produce the desired results ... In the traditional topics, it is generally possible to fairly rapidly focus in on a particular subject and treat it generally in isolation from other topics. ... Web technology topics cover such a complex multi-faceted subject area that such a single-focus, depth oriented approach does not work. This not only has ramifications for the design and teaching of such topics but also for the associated practical work (p. 27).

As Vebyla & Roberts explained, the process of selecting content for web design courses is quite different from other IT courses. This process requires web design educators to take special consideration when selecting the technology and content that will be included in their courses. The process becomes more complicated when the same faculty or department teaches more than web design courses. The core dilemma that this paper is attempting to address is that if a technology department teaches two web design courses, what are the suggested content of these courses? This paper illustrates how the contents of web design courses were planned, and the pedagogical consideration taken into account within the TST department in updating their two web design courses.

The remainder of this paper is divided into the following five sections: The first section describes the challenges that make web design a difficult task to practice. The second section contrasts the paradigms that pervade web design in general and web design technology education in particular. The third section explains about specific considerations that may be taken by colleges and universities in institutionalizing curriculum updates. The fourth section discusses additional factors that may be addressed when updating web design course content. The fifth section elaborates on the experiences of the TST faculty in updating the content of their web design courses followed by summary and concluding remarks.

### **Challenges for Web Design Practice**

Practitioners in the web design profession face certain challenges that are related to the nature of the job, but also to the evolving technology surrounding web page development. Writers in the technology field express different views about the factors that challenge web designers in their profession. Rode (2004) for example outlined four factors that make it difficult for web designer to create web applications:

- 1) Abundance of technologies and standards
- 2) Inadequate technologies and integration between technologies
- 3) Inconsistent implementation of standards
- 4) Differences in end user platforms

Organizations face similar difficulties related to web design introduction and adoption as they are also faced with challenges that are related to the technology itself. Nambisan and Wang (1999)

described three groups of factors that are considered “roadblocks” to adopting web technology in organizations:

- 1) Technology related
- 2) Project related
- 3) Application related

The challenges that web designers face and the roadblocks that impede the introduction of web technology in organizations are usually cascaded into the curriculum of web design also. A difficulty that web designers face in the adoption of new technology because of the changes of standards in the technology may need to be explained in the classroom to prepare students for the real work of web design. Difficulties with adopting web technology in organizations may need to be explained in web design courses. Students also must be aware of the potential roadblocks that they may face when introducing web technology in the workplace.

The remainder of this section explains the challenges that face web designers in their daily work of designing web pages and also the roadblocks that stand in the way of introducing the web technology into organizations. The section further explains how these challenges and roadblocks impact the selection of content in web design courses.

### ***The Multitude of Web Design Technologies***

The number of technologies that are used to design web pages has increased significantly as compared to the tools that were used initially to create such web pages. Horton (2006) noted that the tools that were used to design web pages were limited to using a simple text editing tool (like notepad) and a browser (like Internet Explorer). However this simple list was extended to a much longer record of software tools that are full of acronyms and that are sometimes hard to sift through and distinguish. Bardzell (2006) provided a brief list of these acronym-rich software tools that are used for designing web pages such as: HTML, CSS, Macromedia, Coldfusion, ASP, SQL, XHTML, DHTML, XML, ADO, CDO, JavaScript, Flash, PHP, Java, .NET, XSLT, WML, and WSDL.

The increase in software tools used when creating web pages is coupled with the diversification of the applications in which the Internet is used for. Nambisan and Wang (1999) explained that adopting web technology within an organization does not take place in one step. This adoption takes place at three distinct levels:

1. Information
2. Work collaboration
3. Core business transactions

At the information level, software tools or programming code can be used independently to generate the web pages. A web page at this level contains information for the visitor to view. However there are very few additions encompassed at this level. The interface with other software tools or with other functions of the organization is minimal at this level. As the organization moves to a higher level in their adoption of web technology, the interfaces of web applications increase. This involves other software tools as well as other functions within the organization. As this level of interaction increases, the number of problems and mismatches between the technologies also increases.

An example of the increase of interaction level might be a salesperson who wants access to sensitive data while on the road. He/She may need to access software tools that manage security (security software) and retrieve data from a database (through database software). The same salesper-

son may also need to access specific server applications (server software). In this example the selection of software is complicated by the various compatibility and interface issues as well as other factors such as the cost of installing and managing these software tools.

The selection of software is an issue itself due to the increasing number of software packages used and the multi-functionality that the software provides. Horton (2006) explained that although software tools have a primary purpose when created, the same software can be used for other functions and purposes as well.

Word processing software, such as MS Word, has a primary purpose to create and edit text. However, MS Word can also be used to create web pages. It can also be used for image editing through the picture toolbar, and can be used for simple calculation through the table options. It can further be used for data organization and retrieval.

Similar to word processing software, web page design tools are also multi-faceted also. An example is the Active Server Pages (ASP) tool, whose primary purpose is for server-side web development. However, it can be used for creating simple programs and is sometimes used in teaching introductory programming (Frost, Pike, & Huang, 2005).

Web design software or web authoring tools evolved at a similar pace with web technology. Initially, creating simple web pages was limited to coding HTML tags through simple text editor. Because of the proliferation of web pages, it became important is to find a software tool that simplified adding these tags and help in finding errors during the development stage of web pages. As web application diversified, the criteria for selecting web design tools did not remain limited to tags and simple error finding for the web authoring technology. Numerous other factors were added to the criteria for selecting web authoring software. Among these factors are ease-of-use, ability to include different objects, ability to upgrade, and a successful interface with other applications. Macromedia DreamWeaver for example has many interfaces to server side, programming languages, and databases. It also has interfaces to Flash and other tools. All of these interfaces may be needed as new web applications appear.

In terms of the tools mentioned above, similar difficulty arises when selecting a particular tool or programming language for faculty members teaching web design courses. The difficulty in selecting a new software tool requires not only learning the new features of the program, but also overcoming the associated difficulties with adopting the new technology. Difficulties such as incompatibility issues must be addressed when upgrading from one version to another or when choosing one application versus another.

### ***The Issue of Standards***

Writers in the information technology field contend that establishing standards for software and program development provide a number of advantages. Colet (2007) noted that standards are intended to impose order on developing software so they will be similar. Jain (2007) went a step further and noted a number of advantages that can be gained from following or adhering to standards in the software development industry as noted below:

This is based on the assumptions, subject to endless debate and supported by patient experience, that a methodical approach to software development results in fewer defects and, therefore, ultimately provides shorter delivery times and better value. The necessity of selecting and following a formal practice for software development is to provide desired discipline to deliver the quality expected for business success and avoiding wasting of time, squandering productivity, demoralization in developers etc. (p. 1).

Developing standards in the software development industry has been characterized as “lagging behind” in time (Colet, 2007). Some in the software industry believe that standards were developed as a consequence of increasing problems with software. Most of these software problems appeared mainly due to lack of standards in the software industry (Jain, 2007). Writers cite one notable problem that appeared as a result of not following standards in software development. This problem revolved around the notorious Y2K issues at the turn of this century. These issues appeared due to a lack of standards, or the failure to following the standards set at that time in respect to developing software (Ali & Kohun, 2005).

A similar discussion has been taking place regarding the slow pace of developing standards for web design and the effect it has on the quality of web page development. The multiplicity of applications involved in a particular web page may complicate the issue of standards further, but a brief history on the development of standards in web design may shed some light for the “slow pace” of development.

Web pages were initially developed for informational purposes. A web page was originally used for the purpose of displaying information. There was no reason for separating the code and the format of the data; instead everything was saved in one file. As formatting became complicated, and more formatting styles were introduced for various purposes, combining the code and the format in one file led to creating non-standard pages within the same web site. In other words, there were no uniform standards that were followed for the site. Instead one page used certain formats and others used different formats depending on the designer or the purpose of the page. As this increased, the new Cascading Style Sheets (CSS) were used more often and called increased for using the “two-tier” standards. In this two-tier approach, content was separated from the format. The content is in one file and the format in another. This leads to a web site will be more standardized as they follow the same format (Bardzell, 2006).

As web applications and the volume of data displayed on web pages increased, it became impractical to combine the content and the data in one file. Combining all the data within the same file on the web may jeopardize security of the data being presented. Jeopardizing security of the data takes place primarily because the source code of the web page is viewable from the browser to visitors of the web site.

This issue of increased access to data and the issue of data security on the web necessitated separating data from the content of the web page so the data will not be viewable from source code on the browser. It also necessitated retrieving data that can be accessed from databases which are not saved on the browser side. Instead these data database is usually saved remotely and the display of the data is controlled through programs that manage the presentation of data as well as the security issues associated with this display.

As noted above, these issues of web design did not become limited to only content and formatting. Many additional factors regarding database access and data security emerged. As a result, the demand for separating data from content and formatting increased. Furthermore, the demand to use a three-tier standards model has also increased (Bardzell, 2006). The three-tier model separates content from format as each in a separate tier (Bradley & Millsbaugh, 2003). In the first tier, the content is listed in one file such as the HTML file. In the second tier, the format and styles of the page is saved in another file as the case in CSS files. The third tier saves pages or programs that access data in databases.

The issues associated with web design standards do not stop here in the formation of multi-tier approach to handling files. Instead some web pages call for multiple standards. For example, the web usability guidelines were developed in order to make web pages compliant with standards so they can be accessed from different machines. A web page may contain images and multimedia, so it may have to follow standards regarding graphic and media design. In addition, a web page

may interface with a database and security software. Standards from both areas may need to be addressed when designing the web page.

The brief background on the development of standards listed above may give some understanding to the complexity and the effect that their absence may create on web pages. This of course cascades into the classroom. Faculty members teaching web design courses may face the dilemma of covering all these standards, or only a few of them. Covering all of the standards requires a lot of coverage and may overlap with other courses. An example would be covering standards for the web page interface with databases. This may create redundancy with similar coverage in a database course. Not covering these standards may compromise the depth in the course and the loss of viable information to the students.

Another example can be found in graphic design and the standards associated with using graphics on the web. Questions may arise over whether it would be more appropriate to cover these graphic design standards in a web design course or in a graphic design course. The answer to this question may not necessarily be dependent on whether a graphic design course is being taught in the institution. Even if a course in graphic design is not taught at the institution, the incorporation of graphics on web pages is becoming so prevalent and so necessary that faculty members of web design courses may ask themselves to what extent do they need to cover these standards in their web design courses.

### ***Differences in End User Platforms***

“End user platforms” may mean different things to different people. It may reference hardware, software, communication media or other platforms. End user platforms may also represent different versions of software and associated plug-ins. According to Buccino (2001), “End User platform” includes any of the following:

- Browsers and versions of browsers
- Platform plug-ins
- Screen sizes and resolution
- Network platform
- Client configuration

The magnitude of browser issues and their effect on web design profession may be understood by looking at the number of browsers that can be used to display web pages. Simply put, there are too many browsers that can be used to display web pages. Furthermore, the variety in the display among these browsers is significant. For example, the “Browser Statistics” web site [http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp) provides a list of the browsers most commonly used and their statistics of use by year. This web site lists seven browsers that were commonly used in 2007. It should also be noted that each of the seven browsers have different versions and the versions of the same browser are not necessarily compatible with each the other versions.

The number of browsers mentioned above does not give a clear picture of a bigger comprehensive list of browsers that are still in use. Veen (2001) noted that the number of browsers that are still in use exceeds one hundred and forty. Another project titled “The Browser List Project” (<http://browserlist.browser.org/browser-list.html>) provides a much longer list of browsers that were created since the introduction of the Internet. The list of browsers in this web site spans several pages and gives information into the variety of browsers used for the purpose of displaying web pages. This list was developed in 2000, and Veen’s book that mentioned above was written

in year 2001. Due to the age of these sources it is safe to assume that the number of browsers available in these days exceeds those listed in both sources.

Questions pertaining to the effects of this exhausting list of browsers on web design are very important. The answer to these questions is that most of these browsers are not fully compatible. A web page may be displayed differently depending on the browser being used. In addition, the numbers of plug-ins that are added to the browser complicate this further. A particular feature of a web page can be displayed appropriately with Flash or Java plug-in, but it may not be displayed on browsers without these plug-ins. Designers may need to account for the browsers with plug-ins and those without the plug-ins when writing programs for their web pages.

When discussing screen size and resolution, a simple list of the devices that are used to display web pages may help identify the difficulty associated with designing sites across varying platforms. Syngress Media (2001) noted that the display of web pages on standard computer monitors is dependent on the type of the monitor (CGA, EGA, VGA, SVGA). Equally important are the number of colors used and the screen resolution. The display on web pages is not limited to just simple standard monitors. Instead the list of devices used for displaying web pages is ever growing and includes: laptop monitors, Personal Data Assistants (PDA) cell phones, Liquid Crystal Display (LCD), PlasmaTV, and other devices. Each of these devices come in different sizes, number of colors used, and screen resolutions. As is the case with the increased number of browsers, the same can be said regarding the devices used to display web pages. The number of display devices is continually increasing. Web pages may not be displayed the same way on all of these devices. A web page may be displayed quite clearly on a large LCD screen, but may not be displayed as clearly on a PDA.

Similar issues in regards to network platforms and client configurations exist and pose potential difficulties when designing web pages. Different network platforms are in use, and what works on one network platforms may not work on another platform. Also, user computers are configured differently. What can be displayed according to one configuration may not be displayed the same way on a configuration for another user.

A question that continues to reappear in regards effect these various platforms have on web design profession in general and the teaching of web design courses in particular. The answer is once again is that addressing all of these platforms when designing web pages is too exhaustive for the designer. At the same time not taking these platform issues into consideration may result in the user's data displaying differently that what is intended by the designer.

The choice of platform used also affects the teaching of web design courses. Covering all the network platforms, the user display, the browsers, platform plug-ins, client configuration and other elements of web design is well beyond the content of one course. At the same time, not covering certain elements can lead to ignoring valuable information that may result in the objectives of the course not being met.

## Contrasting Paradigms

Teaching web design courses is subject to contrasting paradigms. These paradigms represent different yet overlapping technology platforms, styles of teaching, coverage areas, and other related subjects. Verbyla and Roberts (1998) explained the paradigms of web design and the challenges they add to selecting course content as:

The great diversity of paradigms which underpin the Web's functionality presents a challenge to the designer and teacher of any topic covering web technology. .... Students need sufficient depth of knowledge in several key areas in order to be able to become proficient in the relevant techniques. Not withstanding,

breadth of coverage is still required given the plethora of technologies associated with the web (p. 27).

The statement above indicate if anything the effect that paradigms have on designing or selecting content for web design courses. Furthermore, the depth of coverage is important for the student to have in order to be considered proficient in this field. In order to clarify this further and understand the various paradigms that are common in the web design field, a discussion of these paradigms may be necessary.

The remainder of this section outlines and explains about some of the different paradigms involved in teaching web design courses. It also discusses how these paradigms affect the selection of content for web design courses.

### ***Using Code versus Software Tools***

When the Internet was first introduced, academic programs sought to teach HTML tags as this was a hot industry topic. In some cases an entire course was designed to teach HTML tags.

Teaching HTML tags was initially accomplished through the use of a simple text editor such as notepad or any other simple text editing tools. Notepad can be used to create text files that are then used to display the pages in a browser.

Since these early days of the Internet, the range of functionality and applications covered on web pages has increased significantly. During this same period, the languages that interface with these applications have also increased. Because of these developments, browsers' programs like Javascript were developed to handle animation and interactivity. Also different languages were used to interface with server and database languages (like ASP, PHP, MySQL and others). Web developers can still code pages that incorporate the HTML tags and also the additional language codes that are needed to be included on the page to handle interactivity and the new features that emerged into the browsers.

During the development stages of these languages various web page creation software tools (like Microsoft FrontPage, Macromedia Dreamweaver and Adobe GoLive) were introduced. These software tools give the user options to "point-and-click" in order to "draw" web pages. The same software tools in turn generate the HTML tags and other codes necessary to display the web page that are "drawn" by the user.

Subsequent versions of these software tools provided an interface to various browser languages such as JavaScript. Macromedia Dreamweaver software for example provided also an interface to the different server side languages such as VBScript, ASP, PHP, and MySQL. Interfacing with these languages and applications did not become limited through programming languages. Instead these interfaces became available through "point and click" through the software tools that allowed the designer to work on them without an in-depth knowledge of the programming language. In some instances simple knowledge of the programming code can make it easier to interface the web page with the application (Bardzell, 2006), but in most cases, knowledge of the language may not help a lot in simplifying the interface with these applications.

To put it in simpler terms, the software tools for web design became "more sophisticated" as the time passed by and as the web applications increased. Also these software tools significantly simplified the process of creating web pages. Instead of memorizing syntax and tags for creating web pages, users can simply point, click, and draw the web page. The software will do the rendering of the code behind the scene to display the page. Questions arise over whether it would be more beneficial to teach students how to use code or software tools when creating web pages. Upon closer examination it is revealed that a number of complicated set of factors accompany each of the two options.



A study conducted by Hijazi (2003) investigated this issue of teaching HTML code in web design courses. The study searched the web for tutorials that teach HTML code. Hijazi found two hundred and nine tutorials existed in Google's directory that teaches HTML, and as many as ninety-five guides and tutorials were discovered in Yahoo's directory for the same topic. This study outlined the availability of tutorials, guides and help sessions on the Internet for teaching HTML. It also brought to question the feasibility of covering HTML code in depth for an entire course. Using all of these tutorials and directories, users can have access to an abundance of online tutorials which can be reviewed without taking courses. The question that may be brought up regarding this is that with the abundance of tutorials and help sessions, would it be necessary to teach HTML in any depth.

A study conducted by Rode (2004) compared the people who created web pages using software tools versus the people who code web pages. The study used the phrase "Nonprogrammer Web Application Developers" to describe individuals who design web pages without knowledge of HTML or scripting languages. The same study used the phrase "professional web programmers" to describe people who design web pages through coding. Rode further compared the two groups by saying:

Professional web programmers are at least able to create interactive applications while non-programmers are so far limited to creating static web sites. Our preliminary studies suggest that these limitations are not due to lack of interest but rather due to the difficulties inherent in interactive web development. Given the right tools and techniques even the nonprogrammers may become web application developers (p. 1055).

The sentiment that is derived from Rod's statement above is that the coder, or people who develop web pages using code, have more in-depth knowledge of the foundation of web design issues as compared to their counter-parts designers who use software applications to develop web pages. Rode also believed in the importance of understanding the foundation upon which web design is built. This foundation may be more understood when designer understand the "behind the seen code" that display the page.

In contrast to the view expressed above, proponents of using software tools describe the use of coding to create web pages as a process of "yesteryear" (Bardzell, 2006) or "old-fashioned way" (Koch, 2006). Even though both views agree that web design software tools such as FrontPage and Dreamweaver simplify the process of building web pages, the common point that was expressed in these studies emphasized the importance of learning the code that makes designing these web pages possible.

The difference between using code versus software has also been echoed by textbooks that teach HTML tags and scripting code when creating web pages (Evans, 2003; Hart & Geller, 2007). Questions of whether to teach students how to use code to create web pages or to use software tools for the same purpose may need to be addressed. Numerous other factors can be considered such as the background of the students and any prior knowledge of this technology. Students enrolled in technical courses or technical majors may have less difficulty understanding programming and using code to create web pages (Wiedenback, 2005). Additional factors regarding time and width of knowledge may also need to be taken into account.

### ***Static, Dynamic, and Interactive***

Web design course content has often been described as using three key words: static, dynamic, and interactive. These three words were meant to offer a clear distinction between one and another. Each term was supposed to describe the components that are specific to each key word.

However, upon further review of the same keywords it is possible for the three terms to overlap in multiple areas. Thus, a brief explanation of these terms is necessary.

Evans (2003) described a static web page as a page that “displays only information that does not change” (p. 5). Gosselin (2006) on the other hand associated static web pages with a “Lack of interactivity” that couldn’t change after they were rendered by the browser (p. 34). Boiko (2005) described static web sites as the “Simplest kind of a site, suitable for small sites that don’t change much and don’t require personalization” (p. 75).

Different terms have also been introduced to describe dynamic web pages. Hart & Geller (2007) described dynamic HTML as:

A combination of HTML enhancements and a scripting language that work together to include animation, interactive elements, and dynamic updating to web pages (p. 15).

Bardzell (2006) describes dynamic web sites as the separation of logic and presentation in the creation of web pages. Powers also associated dynamic web sites with “Database Driven Web-sites”. Boiko (2005) gave the following definition regarding dynamic web sites:

A dynamic Web site, sometimes known as a database-driven site, is a system for producing web pages on the fly as users request them. A data source (a relational database, or possibly an XML structure) on the Web server receives a query in response to a user clicking a link. The link activates a template page (p. 75).

Although interactive web pages can retrieve and update data, other layers of communication are involved in defining this term. Gosselin (2006) commonly refers to interactive web pages as those that “communicate with databases” (p. 34). Bardzell (2006) has further simplified this process by dividing web site design into three distinct layers: logic, presentation, and content.

Overlapping terms commonly lead to confusion, and may affect the selection of content when designing the structure of web design courses. Dynamic web design may include a wide range of topics that can be classified under “Interactive” and vice versa. In these instances the simple task of selecting a name for a web design course may lead to confusion as the possibility exists that it will not actively reflect the true content of the course. It is also possible that a course listed under one name may overlap with the content of a course listed under a different name.

The description of these three terms reveals that static refers to the simplest single-unit of coverage. However dynamic and interactive web design describes a wider range of topics that are inter-related with static web design. Dynamic web pages interact with animation, and multi-media elements. Interactive web pages involve the relationship with a database.

Educators may face dilemmas when they select one paradigm of web design courses over the others. Choosing one paradigm for a course may lead to the selection of a technology that is different than the others. Dynamic web sites for example require the teaching of interactivity, multimedia, content management systems, and news groups. On the other side, interactive web content may include connection to data sources, updates and retrieval.

### ***Server Side versus Client Side***

The terms “Server Side” and “client side” often referred to discussing web pages in general. However, these two terms normally refer to the dissimilar roles within the web design process. These terms also reference different sets of language code and interfaces associated with a specific web page. Gosselin (2006) explained the difference between client side and server side technologies as:

Client-Side scripting refers to a scripting language that runs on a local browser (on the client-tier) instead of on a web server (on the processing tier) (p.35).

Server-side scripting refers to a scripting language that is executed from a Web server. One of the primary reasons for using a server-side scripting language is to develop interactive web sites that communicate with a database (p. 36).

The available options within the client and server side may directly affect the selection of technologies. Bardzell (2006) describes technologies such as ASP, PHP, or ColdFusion as “designated” for server side web development. Gosselin (2006) on the other was more specific about the technologies involved and explained that client side web pages are those developed using HTML and Java Script languages. Although some programming languages can perform tasks pertaining to both the client and server sides, there are certain common software tools responsible for distinct functions during design process (Bardzell, 2006).

A study conducted by Sebastian specified the technologies involved in both the server side and the client side of web design (Sebastian, 2008). This study selected the following technologies as client-side “HTML and XHTML, CSS, Javascript, AML and XSLT, Applets”. The same study listed the following server-side technologies “Perl/CGI, Servlets and Java Server pages, PHP, ASP.NET, Database Access through the web, Ruby, Rails, and Ajax”. Although these terms describe specific technologies, the implication is that the content may require more in-depth knowledge of multiple and specific technologies. A particular technology such as ASP .NET covers the server side, but in order to understand this, a good knowledge of HTML and XHTML may be necessary.

Alternative names have been surfacing to describe the same paradigms associated with web design discussed here. These alternative terms are “front-end” and “back-end” paradigms. Graf (2006) outlined the difference between these paradigms as:

The front-end is the website – what the visitors and the logged-on users see. The back-end, on the other hand, contains the administration layer of the website for the administrator. Configuration, maintenance, clearing, creation of statistics, and new content creation are all done in the back-end (p. 9).

The selection of one teaching paradigm over another is not limited to the selection of the technology chosen. Instead it covers a much wider set of applications being used with the same technology. A student working on the server side may also need to be aware of database concepts in order to further understand this technology. By the same token, learning client side web design may necessitate teaching concepts of graphic design, multimedia, and others factors that affect the display of web pages on client browsers.

### ***Open Source versus Propriety Software***

The use of propriety or commercial software to create web pages has been in existence since the early days of the introduction of the Internet. Microsoft ASP for example was introduced in October 1996 (O’Reilly, 2006). The trend of using commercial software for web development continued until a new brand of web development tools emerged. This new brand of software was named Open Source Software (OSS). Asiri (2003) outlines the difference between propriety and open source software:

Open source refers to all software in which the source code is freely available for any person to view, amend, or adapt. This is distinguished from propriety software applications where the source code is secret, not made public or standardized, and closely protected through patents and intellectual property applications (p. 1).

Since the introduction of OSS, new terms and definitions have been introduced to describe this brand of software. Ye & Kishida (2003) provided the following definition for OSS:

Systems that give users free access to and the right to modify source code. OSS grants not only developers but all users, who are potential developers, the right to read and change its source code. Developers, users and user-turned developers from a community of practice. A community of practice is a group of people who are informally bounded by their common interest and practice in a specific domain (p. 419).

The origin of open source software dates back to 1991 when Linux was introduced. It is considered the first tangible achievement of the open source software movement (Asiri, 2003). Since that time the list of open source software has grown significantly and now covers a much wider range of applications (Dalziel, 2003). Open source coverage areas include many applications such as finance, mathematics, assistive technology, data mining, and document editing.

The list of open source software has increased at a speedy rate over the past years. Damicon (2007) provides a comparison list of commercial software categories and the open-source equivalents. This list includes more than sixty commercial software categories and hundreds of open-source equivalents. Open source software is not limited to web design; it includes also a wider range of courses such as networking, database, productivity software, and software used for teaching online courses.

The initial development of open source software was met with resistance from the industry and with skepticism from the public. This doubt was rooted over the impact it would have on pre-established commercial software. This parallels the public view of free technical products. The common perception that is often heard about free products casts doubt over such products. "After all, what do you get for free" or "you get nothing for free" are two statements that are commonly repeated among the public when describing free products. However, the quality of open source software and the impact on industry has proven many skeptics wrong. O'Reilly (1999) explains that the introduction of open source software had a greater impact than just providing the availability of code and stated that:

Some of the most significant advances in computing advances that are significantly shaping our economy and our future are the product of a little understood 'hacker culture'. It is essential to understand this culture and how it produces such innovative, high-quality software. What's more, companies large and small are struggling to understand how the ethics of free source code distribution affects the economic models underlying this business (p. 34).

The free availability of open source software and its stable platform has made this a viable option for web designers seeking an alternate solution (Bardzell, 2006; Gosselin, 2006). The availability and increased use of open source software has resulted in web designers and educators having to determine if open-source software such as PHP would be a feasible addition in their curriculum. Web design educators may need to weigh this option against vendor specific software such as Microsoft's ASP. Gosselin (2006) outlined a key advantage of open source software as its simplicity and availability that allows programmers to conduct further developments without restriction. Bardzell (2006) supports open source software but also details the difficulty associated in obtaining support as compared to vendor products. Bardzell noted that commercial software vendors often offer twenty-four hour help lines, online help, and training programs. The same can not be said for open source as users are forced to search through blogs, news groups, and other listings when seeking help.

Despite the initial skepticism that surrounded the initial development of open source software, but many still view this as a viable option from the beginning due to its stability and cost (Koochang & Hartman, 2005). The rapid expansion of open source software has forced web design educators to consider introducing this technology into their curriculum. At the heart of this discussion is PHP as a server side development tool, and MySQL as a database for the web. This compares to Microsoft's ASP .NET and Microsoft Access or Microsoft SQL server respectively. Comparison between these platforms is beyond the scope of this paper, but it worth mentioning that the development of these platforms parallel one other.

The consideration for introducing one platform versus another may not be limited to the use of the tool or the language itself. Instead, educators of web design courses may need to research industry demand when deciding whether or not to introduce a specific platform into their course. Web design educators may need to compare the options available between commercial and open source software.

### ***Theory versus Application Coverage***

An important question when teaching IT related courses is how much emphasis is to be placed on teaching theory (or concepts) versus the application and software tools. In many cases, the extent of coverage of the theory side of the topic distinguishes college courses from other technical or specialized schools. Colleges and universities traditionally place a greater emphasis on teaching theory as compared to the hands on approach of technical or specialized schools.

An emphasis on the application gives students more experience on the software features. Focusing on the application provides students with the opportunity to look at the features and tools provided. When emphasizing the theory of a web design course, students have more exposure to the concepts that exist, the correct use of the tools, and the conceptual framework of the technology.

The web design field offers an abundance of both concepts and applications. The concepts that may need to be covered in web design are directly related to the web page interfaces with and the paradigms each follows. On the client-side, students may need to learn about topics such as web-accessibility guidelines, images and multimedia, and other related concepts of web site design. On the server-side, students need to learn about database and security concepts as well as related network and interface applications.

Emphasizing the use of application side development tools broadens the options available when choosing content. The software tools used for developing web pages are abundant in features and capabilities. Programs such as Macromedia Dreamweaver suite offer five different software tools for developing portions and functionalities of the web pages. Each of these tools has a wide range of applications and capabilities that educators can utilize when teaching web design.

It is vital that web design educators find a balanced approach between theory and hands on instruction. In this balanced manner, students may build a solid understanding of both the theories of the technology behind web design. Denning (2001) noted that the balance of coverage between application and concepts in IT involves:

Learning the professional practices of a specialty of information technology is every bit as important as learning the intellectual core of computing. The mark of a well educated professional will be a balance of the two. The current academic inclination to disdain skill-specific training does not fit a profession. The education of computing professionals must count for practices as well as descriptive knowledge. It must include training as well as general education (p. 263).

The need for a balance between theory and hands on instruction in web design is not well marked and depends on the tools being used. This balance depends also on the field of study and the level

at which the course is being taught. Senior level courses may be more receptive to theoretical concepts as compared to a freshman or introductory classes. Students enrolled in more technical field of study may be more interested in practice while less technical field of study may emphasize more on the theory aspect of it. However, in all cases a mix of concepts and application is needed when selecting contents for web design courses. It is also may be necessary that web design educators make a personal judgment when deciding on the balance between the mix of application and theories to include in the courses.

### **Institutional Consideration**

The consideration of the specific institution and the regulations and procedures that they follow may vary from one institution to another. Specific considerations may need to be taken into account when dealing with courses that are contested among different departments. This form of contesting is heightened when departments compete for student enrollment and more specifically when enrollment is on its' down-tum in these institutions.

Additional factors that are specific to the institution may also be different depending on the college or within departments in the same college. All these factors may play a role when updating curriculum for particular courses. The remainder of this section outlines some of the institutional considerations that may be taken into account when updating web design courses.

### ***Domain of Study***

The web design coverage area (or domain of study) has widened and now includes numerous additional fields of study. A web design course may be taught in the traditional technology fields of Information Systems (IS) and Information Technology (IT). However, other fields of study may include elements of web design concepts in their curriculum. An example can be a computer science department that teaches web design from a technical perspective as compared to a marketing department that concentrates on teaching it from marketing or e-commerce perspective.

Because web design can be approached from many different views, it may not be easy to give sole ownership of web design curriculum to one single department. Frequently, competing departments may disagree on ownership of similar courses. In these instances it may be necessary to consult some standard curriculums that are designed to give guidance when selecting courses for particular programs.

The Association of Computing Machinery (ACM), the Institute for Electronic and Electrical Engineering (IEEE), and the Association for Information Systems (AIS) issue different documents regarding curriculum in various technology fields. Due to the popularity of these institutions, the documents produced by these organizations are usually considered as standard curriculum for IT academia. One of the most recent standard curriculums that is issued by these organizations is named the Computing Curricula 2005 The Overview Report (or CC2005).

CC2005 details the foundation that IT curriculum designer need to follow in five technology fields: Information Systems (IS), Information Technology (IT), Computer Science (CS), Computer Engineering (CE), and Software Engineering (SE). CC2005 suggests distinctive knowledge areas within each of the five Information Technology (IT) programs.

CC2005 provides a listing of different knowledge areas that the five categories of technology fields need to follow. It also provides a scoring mechanism for the coverage of each of the knowledge areas listed. This scoring mechanism range between 1 and 5, with 1 providing the minimum coverage and 5 provide the maximum coverage for the knowledge area (Ali, Kohun, & Wood, 2007).

CC2005 provides also a table suggesting relative performance capabilities for computing graduates by discipline for each of the five disciplines it discusses. This table lists the performance area, the performance capability and a suggested score for the capability/area designate for each of the five disciplines discussed in their document. The suggested score for the performance capability ranges from 1-5.

Among the knowledge areas listed in CC2005 is one area regarding “Digital media development” and is given a maximum score of 5 for the IT field. A further examination of the relative performance capability of CC2005 reveals more areas regarding web design covered in these performance capabilities. The performance capability table of CC2005 list one broad area named “Systems development through integration”. Under this category listed six additional performance capabilities that are all related in some form to the Internet and web design fields. These six performance areas are:

- Manage an organization’s web presence
- Configure and integrate e-commerce software
- Develop multi-media solutions
- Configure & integrate e-learning systems
- Develop business solutions
- Evaluate new forms of search engines

Scanning the scores given for each of the six performance capabilities listed above shows that the IT field receives a maximum score of points in five of the six categories listed above. The only one that IT received a score of 4 is the last one about the search engines. This indicates if any that web design and Internet is classified under the IT domain.

Another standard document developed by the body that developed CC2005 is named Computing Curricula Information Technology Volume (or IT2005). This document is more specific about the new Information Technology discipline. IT2005 lists several “Themes” for the Information Technology major that are considered as corner stones for selecting courses in this field. Two of the themes listed in IT2005 are directly related to the web design field. The first one is IPT integrative programming and technologies which discuss among other topics scripting techniques, and software security practices. The second theme is titled Web Systems and Technologies which discuss web technologies, digital media, web systems and others.

Combining both of these standard curriculum (CC2005 and IT2005) provides more evidence that web design is considered a domain of the IT field. Combining the same two documents suggests also that the web design topic may need to be covered in more than one course, as is the case in IT2005 two courses.

### ***First Course and Sequence of Courses***

One of the difficulties that face educators when teaching web design courses is that there is no designated first course for this field. Also, there is no established sequence of courses that the students take when enrolled in such programs. This is in contrast to other “well established” curriculums such as English or Math. In these curriculums there is usually a designated entry level course that the students complete such as “English 101” or “Math 101”. These entry courses are usually followed by a second course in the same field like “English 102” or “Math 102”. The same can not be said for web design courses. There are no designated courses, and no established sequence of courses has been constructed.

A similar problem faces programming curriculum in light of the increasing number of programming languages. In the early days of teaching programming in different curriculum, students used to be required to take BASIC programming language in their first programming course (Ali & Kohun, 2005). Regardless of the students' field of study, completing BASIC was a requirement as a first course. This was usually followed by other courses in the same field.

BASIC is no longer taught at academic institutions. The slow disappearance of BASIC is followed by designating Pascal as the first programming language to be taken by students. This designation did not last very long because technology changes opened a wide range of options to give educators and students a different set of entry courses.

The recent use of programming in education has shifted the focus of the discussion among faculty members teaching such courses. The point of discussion common among educators of introductory programming courses is which methodology of programming to teach. Faculty members may choose among object-oriented, structured, or a visual approach to programming when selecting introductory courses (Ali & Kohun, 2005).

Traditional "first course" in the web design field emphasized teaching HTML. As the Internet increased in use, so did the approaches to the introductory course in web design. To complicate the issue further, some of these courses are offered for "non-majors". A consideration that may need be taken into account for the students from other majors who want to take the course. In addition some web design tools can cover a wider range of applications that may cross into different fields of study such as marketing and management.

### **Teaching Styles**

Teaching style can vary from one course to another. But regards to technology teaching, two styles are common: Explorative and learning by doing. Ye and Kishida (2003) explained about these two styles of learning:

**Explorative learning-** This form of learning is similar to the most scientific research (e.g., scientists, practitioners) attempts to find new ways of doing things or of overcoming an existing problem. OSS systems viewed by GNU developers as scientific knowledge to be shared among mankind. Larry Wall started Perl because he ran into a problem he couldn't solve with existing tools, and he wanted to explore a way of doing things better.

**Learning by doing-** In this form of learning, the learners want to deepen their understanding of a certain tasks by actually engaging in practical tasks that allow them to apply their existing knowledge and to perfect their skills (p. 425).

The two styles of learning may apply to any learning environment in general and may not be limited to one field. Despite the generality of both learning styles, a decision can be made regarding their applicability. When discussing web design the preference of one learning style over the other is not clear cut. If teaching the course is accomplished through programming, then it may be argued that programming students traditionally learn more by doing hands on programs. On the other hand, using the software tool expose the students to few programming concepts thus more of teaching theory may be helpful in this case.

Questions may arise over the amount of hands-on practice as compared to theory needed when delivering a web design course. The web design field offers an abundant amount of materials that increases an explorative learning style. This is illustrated by giving students an ample amount of examples and explaining how they apply to meet the course objectives. It may need also to be stressed that "practice makes perfect" to deepen the understanding of the software tools chosen



for the class (Rode, 2004). At the same time, a minimum of concepts of theories may need to be explained in order to teach the “correct approach” for this field.

## Other Factors

This section explains about other additional factors that may influence the decision of selecting topics for web design courses. These factors may be common to other fields of study, but this section discusses them from the perspectives of web design field. These factors also do not fall within any the categories outlined above, thus they are grouped under the heading of additional or other factors.

### ***Additional Topics***

As the subjects that are covered in web design courses are increasing, additional topics are beginning to mesh together and increasingly are falling under the banner of web design courses. Because of this issue, new questions are beginning to merge as to the extent of additional topics that may need to be included in web design courses. This is not only restricted to the software and platform used, but also additional components such network security, image editing, e-commerce and others.

Various professional web design organizations such as the International Webmasters Association <http://www.iwanet.org/> and also the American Association of Webmasters <http://www.aa.webmasters.com/> offer courses for teaching web design topics. There are other web design teaching institutions, among them is such as Academy of Web Design <http://www.awdsf.com/>. In these institutions, additional topics such as image editing, animation, and the use of image editing software such as Adobe Photoshop and Flash are core elements within the courses offered.

Another reference that may be consulted in regards to additional topics in web design courses is the IT2005 standard curriculum. IT2005 list different additional topics that are suggested in the IT2005 web design suggested courses are digital media and Internet security.

### ***Reduce Redundancy and Duplicate of Materials***

Reducing redundancy is a common theme that is echoed in various computer technology courses. In database topics, some consider reducing redundancy as a goal that is sought when normalizing the database tables (Pratt & Adamski, 2002). Reducing redundancy is considered one of the steps that contribute to successful database design for business applications (Bast, Cygman, Flynn, & Tidewell, 2006). A similar sentiment is echoed in academia when designing courses. Redundant materials across courses are known to waste time and leads to inconsistencies. Moreover, duplication of course content leads to unfavorable attitude from students and teachers also (Smith & Hartley, 1954).

Technology fields are known to have redundant and duplicate materials among courses. A department for example that is teaching two programming courses may repeat concepts about design structure, sequence, selection, and looping concepts in each of the two courses they teach. A course in hardware design may repeat content regarding CPU, hard drives, and motherboard that may be covered in earlier computer literacy course. Redundancy in technology course seems to be inevitable.

Similarly, Web design course may have certain elements of redundancy and duplication among the materials covered. A course in server side web design may need to talk about HTML tags and the interactivity in JavaScript that is also covered in a client side web design course. Thus the same can be said as is inevitable in web design courses.

Reducing redundancy can be minimized if pre-requisite are established for courses. As an example, if students are required to complete client side web design before server side web design. Earlier concepts about HTML can be covered in the client side course and the second course can be dedicated to cover the server side technology. Planning the course content and putting in place steps to keep the redundancy minimal may be helpful. But in the absence of such pre-requisite, it seems that some redundancy in technology related courses is inevitable.

### ***Importance of Course Naming***

Giving a name to a course is important in many ways. First, it gives a glimpse of the content of the course to the students. A course name such as “Organic Chemistry” is helpful to students majoring in chemistry as the name indicates that it covers organic and it covers chemistry. Second importance of naming courses is that course names gives an impression to the level of the course and its’ pre-requisite. A course titled “Advanced Java Programming” provides understanding that it is a course involves advanced concepts in Java programming which requires the students to have knowledge of the language before taking it.

In the web design field, the naming conventions that followed include either naming a course after a specific technology or a filed name. Specific technology naming may include “Flash 8” or “Adobe CS3” that refer to specific technology name. One of the advantages of following this kind of naming conventions is that it draws attention to the technology itself. If the technology referenced in the name is more recent, chances are it may attract more students. However, the disadvantage is that if the technology changes then steps have to be taken to change the name of the course. In addition some of these steps may be administrative and bureaucratic that is specific to the institution.

The other naming convention refers to the subject or paradigm. For example, “Server side web design” is a name of course that teaches the server side paradigm of web design that was explained earlier in this paper. Another name, “Interactive web development” is a name that describes a course that covers topics related to the interactive web development paradigms. Naming a course haphazardly and giving names that do not reflect the content of the course may lead to a misunderstanding and disappointment among students.

A general step that provides certain categorizations of the course may lead to a better selection of names. Also, consulting standard curriculum may help in providing names for courses. The IT 2005 standard curriculum gives names for their themes like “Web Design, Scripting Languages, or Web Media”. Starting with these words and then modifying them according to the theme or category may help designers to select more appropriate names.

## **Web Design Courses in a Technology Support Program**

The Technology Support and Training (TST) – Eberly College of Business and Information Technology (ECOBIT) at Indiana University of Pennsylvania (IUP) offer two bachelor degrees and one associate degree. The first bachelor degree is in business technology support while the second is in business education. The associate degree is in business computing and information technology.

The TST department is one of three departments at IUP that offers a computer technology related degree. The other two are Computer Science (CS) and Management Information Systems (MIS). There are other programs that teach some components of technology such as Communication Media and Marketing. But in general, the three departments of CS, MIS, and TST offer degrees specifically in the computer related field.

A study conducted by Ali, Kohun, and Wood (2007) compared the content of the courses offered for the TST degrees. The study found that the TST program is more closely aligned with the IT field under the knowledge areas listed in CC2005. This study also found that the two web design courses offered within the TST program are compatible with the two specific knowledge areas about web that are listed in CC2005.

The remaining section explains about these two web design courses offered in the TST department: BTST401 and BTST402. Both of these courses are senior level (400-level) courses; thus one can assume that the students hold some familiarity with computer technology. Revisions of both courses are explained below.

To illustrate the different factors that are taken into account when selecting topics for both TST web design courses, table 1 below summarizes the pedagogical consideration that were discussed in this paper and how they are applied to each of the two courses that faculty members at the TST department attempted to address them in their two web design courses. The summary table is followed by a detailed description of the two courses.

Paradigm	TST401	TST402
Code Versus Software Tool	Software Tool	Code
Dynamic Versus Interactive	Dynamic	Interactive
Server Side/Client Side	Client Side	Server Side
Back-end/Front-end	Front-end	Back-end
Open Source/Commercial Software	Commercial Software	Open Source
Additional Coverage	Image editing tools	Database Technologies
Teaching Style	Learning by doing	Explorative
Theory/Application Coverage	Balanced moved more toward application	Balanced move more toward theory

BTST401 Web Design Dynamics - This course teaches web development through the use of software tools. Macromedia Dreamweaver has been selected as the main software tool for designing web pages in this class. The course also uses Macromedia Flash as the primary software for teaching image editing and graphic design and multi media concepts. A limited exposure to databases is included in this course. The course represents teaching topics related to creating "Client-Side", "Dynamic" and "Commercial Software" paradigms as outlined earlier in this paper. While this course strives for a balance between theory and practice, many of software features used in this class encompass a hands-on application approach. Additionally, this course is required for the business education major, thus less emphasis is placed on programming and the technical end of these topics. Regarding the conceptual part, emphasis is placed on web usability guidelines, web graphic optimization and web media usage also. Most of the exercises and projects given in this course follow the "learning by doing" paradigm explained above. Both software used for this course have many features and case projects that resemble real-life organizations are given so the students follow the steps to design web sites for these cases.

BTST402 Interactive Web Development - This course teaches web development using PHP programming language. This course includes also use of database retrieval/updates utilizing MySQL database language. Additional topics included the use of the Extensible Markup Language (XML) to interface with web pages through PHP. This course follows the “server-side” “interactive” and “open source” paradigms that were discussed earlier in this paper. Additional topics about database and web security are included in this course also. This course is taken mostly by students from the technology support major, thus more emphasis is placed on the conceptual part and the technical end of this topic. The conceptual part emphasizes on database concepts, security issues and various platforms of web interfaces with the server side of the technology. Most of the projects that are assigned in this course require learning by exploring approach that was discussed earlier. Although at the beginning of the course, students learn by practicing through specific designed steps, these steps get less specific and students are expected to learn more through exploring and looking at comparable programs.

Regarding naming the courses, the attempt was made to make the course names indicate their content. The first course the word “Dynamic” was included because the course teaches animation, creating dynamic web sites. The second course name used the word “Interactive” because it teaches concepts related to interactively updating databases. The other words in the title of the courses used two different words “Web design” versus “web development” so to draw a clearer distinction between the two courses.

Faculty members at the department attempted also to eliminate redundancy between the two courses as well with other courses at the department and the university in general. The content of both courses were reviewed and compared with the description of other course at the department. Words were used in defining the objectives of the course to make sure they do not describe the domain of other departments.

## Summary

This paper described the pedagogical consideration that were taken into account when selecting content for two web design courses within the Technology Support and Training) department – Eberly College of Business and Information Technology (ECOBIT) at Indiana University of Pennsylvania (IUP). The paper began by describing the factors that make it more difficult to practice web design profession in general and then to select content in web design courses. It covered these factors from various technological and academic perspectives. The paper also explained how these factors affect the teaching of web design courses and the selection of topics in this field.

The discussion of the paper then shifted toward describing the contrasting paradigms that are practiced under the web design field. These paradigms affect the selection of technology and also influence the decision regarding content to be covered for the web design courses and the approach of teaching them. The paper further then discussed additional factors that influence the selection of topics in web design courses. These factors do not fall within the general topics discussed earlier, thus they grouped under other topics.

Lastly, the paper explained how the TST department took steps to address these points of concern regarding the selection of topics for their two web design courses. The faculty members at the same department studied the factors and addressed them as they apply to each of the web design courses they teach in their department.

## References

- Ali, A., & Kohun, F. (2005). Suggested topics in introductory IS programming in Java. *Issues in Informing Science and Information Technology*, 2, 237-246. Retrieved from <http://proceedings.informingscience.org/InSITE2005/I19f28Ali.pdf>
- Ali, A., Kohun, F., & Wood, D. (2007). Comparing two programs with CC2005 Standard Curriculum. *Issues in Computer Information System*, VIII, 93-96. Available at [http://www.iacis.org/iis/2007\\_iis/index.htm](http://www.iacis.org/iis/2007_iis/index.htm).
- Asiri, S. (2006). *Open source software*. Retrieved October 26, 2006 from ACM Digital Library <http://www.acm.org/dl>
- Bardzell, J. (2006). *Macromedia Dreamweaver 8 with ASP, ColdFusion, and PHP, Training from the source*. Berkley, CA: Macromedia Press.
- Bast, K., Cygman, L., Flynn, G., & Tidewell, R. (2006). *Succeeding in business application with Microsoft Access 2003 – A problem solving approach*. Boston: Thomson, Course Technology.
- Boiko, B. (2005). *Content management bible* (2nd ed.). Indianapolis, IN: Wiley Publishing.
- Bradley, J. C., & Millspaugh, A. C (2003). *Advanced programming using Visual Basic .NET*. Boston: Irwin McGraw Hill.
- Buccino, A. (2001, January). E-pain relief. *Intelligent enterprise*, 4. Retrieved June 14, 2007 from <http://find.galegroup.com/ips/infomark.do>
- Colet, E. (2007). *The importance of standards*. Retrieved June 17, 2007 from <http://www.taborcommunications.com/dsstar/00/0118/101246.html>
- Computing Curricula Information Technology Volume. (2005). Retrieved March 4, 2007 from ACM Digital Library <http://www.sigite.org/>
- Computing Curricula 2005 The Overview Report. (2005). Retrieved October 2, 2006 from <http://www.computer.org/curriculum> .
- Dalziel, J. (2003, November 4). Open standards versus open source in e-learning. *Educause Quarterly*, pp. 4-7. Retrieved from <https://www.educause.edu/ir/library/pdf/eqm0340.pdf>
- Denning, R.J. (2001). Computing the profession: Practical experiences for undergraduate computer networking students. *Consortium for Computing Sciences in Colleges*, pp. 261 – 270.
- Damicon. (2007). *List of open-source software links*. Retrieved June 30, 2007 from <http://www.damicon.com/resources/opensource.html>
- Evans, J. (2003). *New perspectives: Microsoft FrontPage 2003 Comprehensive*. Boston: Course Technology.
- Frost, R. D., Pike, J. C., & Huang, W. (2005). Breaking down the blocking boundary of separated IS courses in IS curriculum: A case study. *Issues in Information Systems*, VI(1), 303-309. Retrieved from <http://www.business.pitt.edu/katz/phd/people/students/jacqueline-pike.html>
- Goldweber, M., Impagliazzo, J., Bogoiavlenski, L., & Clear, A. (1997). Historical perspectives on the computing curriculum. *ITCSE97 Working Group*. Retrieved October 10, 2006 from ACM Digital Library <http://www.acm.org/dl>
- Gosselin, D. (2006). *Web worrier series: PHP programming with MySQL*. Boston: Course Technology.
- Graf, H. (2006). *Building websites with Joomla*. Birmingham, UK: Packet Publishing.
- Hijazi, S. (2003). E-publishing-What is available? *Proceedings of the 2003 ASCUE conference*, retrieved October 10, 2006 from <http://fits.depauw.edu/ascue/proceedings/2003>
- Horton, S. (2006). *Choosing a web authoring tool. Web teaching guide*. Retrieved September 13, 2006 from <http://www.dartmouth.edu/~webteach/articles>

## Selecting Content for Web Design Courses

- Hart, K., & Geller, M. (2007). *New perspectives on Macromedia Dreamweaver 8 – Comprehensive*. Boston: Course Technology.
- Jain, D. (2007). *Importance of processes and standards in software development. The Code project*. Retrieved June 17, 2007 from <http://www.codeproject.com/useritems/Process.asp>
- Koch, P-P. (2006). *A history of web development*. Retrieved October 2, 2006 from [http://www.quirksmode.org/oddsandends/history\\_webdev.html](http://www.quirksmode.org/oddsandends/history_webdev.html)
- Koohang, A., & Harman, K. (2005). Open source: A metaphor for e-learning. *Informing Science Journal*, 8, 75-86. Retrieved from <http://inform.nu/Articles/Vol8/v8p075-086Kooh.pdf>
- Nambisan, S., & Wang, Y. (1999). Roadblocks to web technology adoption. *Communications of the ACM*, 42(1), 98-101. Retrieved June 14, 2007 from ACM Digital Library <http://www.acm.org/dl>
- O'Reilly, T. (1999). Lessons from open source development. *Communications of the ACM*, 42(4). Retrieved October 25, 2006 from <http://www.acm.org/dl>
- O'Reilly, T. (2006). What is ASP .NET. O'Reilly Windows devcenter .Com. Retrieved August 20, 2007 from <http://www.windowsdevcenter.com/>
- Powers, D. (2006). *PHP for Dreamweaver 8*. New York: Friends of ED Foundation.
- Pratt, P. J., & Adamski, J. (2002). *Concepts of database management*. Boston: Thompson Course Technology
- Rode, J. (2004). Nonprogrammer web application development. CHI 2004 *Doctoral Consortium*. Retrieved October 20, 2006 from ACM Digital Library <http://www.acm.org/dl>
- Sebestial, R. W. (2008). *Programming the World Wide Web*. Boston: Pearson Addison-Wesley.
- Smart, K., Rice, J. C., & Wood, L. E. (2000). Meeting the need of users: Toward a semiotics of the web. *Proceedings of the 18th Annual ACM International Conference on Computer Documentation*. Retrieved June 14, 2007 from ACM Digital Library, <http://www.acm.org/dl>
- Smith, H. F., Hartley, E., & Smith, O. (1954). A further study of student teacher' attitude on duplication of course content. *JSTOR: Peabody Journal of Education*, 38, 214-219.
- Syngress Media. (2001). *A+ certification study guide* (3d ed.). New York, NY: McGraw Hill Publishing
- Vebyla, J., & Roberts, G. (1998). *Web technology as curriculum*. ACM Press. retrieved September 15, 2007 from ACM digital library <http://www.acm.org/dl>.
- Veen, J. (2001). *The art and science of web design*. Indianapolis, IN: New Riders.
- Wiedenback, S. (2005). *Factors affecting the success of non-majors in learning how to program*. Retrieved October 20, 2006 from ACM Digital Library <http://www.acm.org/dl>
- Wikipedia. (2007). *List of open source software packages*. Retrieved June 30, 2007 from <http://en.wikipedia.org/wiki/>
- Ye, Y., & Kishida, K. (2003). *Toward and understanding of the motivation of open source software developers*. Retrieved March 20, 2007 from ACM Digital Library <http://www.acm.org/dl>

## Biographies



**Azad Ali, D.Sc.**, Associate Professor of Technology Support and Training at Eberly College of Business – Indiana University of Pennsylvania has 23 years of combined experience in areas of financial and information systems. He holds a bachelor degree in Business Administration from the University of Baghdad, an M.B. A. from Indiana University of Pennsylvania, an M.P.A. from the University of Pittsburgh, and a Doctorate of Science in Communications and Information Systems from Robert Morris University. Dr. Ali's research interests include object oriented languages, web design tools, and curriculum design. His community service and academic expertise gets him in the news on Pittsburgh television and in the newspapers.



**Scott Mensch, Ph.D.**, Assistant Professor of Technology Support and Training at Eberly College of Business Indiana University of Pennsylvania. Over the past 12 years he has held several positions in both IT and education and currently instructs and designs courses in both business and information technology in a traditional setting and online. Dr. Mensch completed his M.B.A. in 1998, and obtained two additional Associate Degrees in Computer Applications and Networking. In addition between 1999 and 2003 he also obtained his (CCNA) Cisco Certified Network Associate, (MCSA) Microsoft Certified 2000 Systems Administrator, (MCSE+I) Microsoft Certified Systems Engineer + Internet NT 4.0, A+ Computer Technician, and Server + Certifications. Dr. Mensch obtained his PhD in Organizational Management with a Specialization in IT Management, and is currently completing post doctoral studies in telecommunications and IT security.