

The Difficulties of Incorporating Web Development into a University Curriculum

Jill E. Courte
Miami University, Hamilton, OH
courteje@muohio.edu

Abstract

Can web development be a viable part of a university curriculum today and in the foreseeable future? Several factors make developing and promoting a web programming curriculum a daunting task. Perhaps the most difficult obstacle is the ongoing evolution of the practice of web development. What might have been a reasonable curriculum for web development even two or three years ago would not be altogether viable today.

Most universities deal with comparatively static topics; significant change can come slowly to most university curriculums. This is due in large part to the universities' teaching goals of providing information of lasting value, avoiding topics of fleeting importance. For example, an educator is fairly safe developing a programming curriculum using languages such as C, C++, or Java. The basic skills needed for traditional programming have not changed dramatically even though the software industry evolves in terms of new programming languages or paradigms. You can safely teach students how to solve problems using a programming language without fear that you'll need to dramatically change how you think about the topic within a short window of time.

On the other hand, web development is presently an extremely dynamic discipline. For instance, the skills and tools used in web development can involve a wide variety of disciplines. Where a software engineer might be proficient in one or two languages and environments, and have experience with a few tools, web development professionals may be called upon to be familiar with a wide variety of languages, environments, tools, and techniques. In addition, web development often entails some synthesis of programming, business, and the arts. This paper is a case study of the difficulty of designing a curriculum to meet the needs of present and future web developers in an evolving discipline.

Introduction

A year ago I moved from industry to the academic world and was given the task of developing a web programming curriculum for a two-year degree track within an information technology program. The program is relatively young, formed by spinning off a group previously included in a computer science and systems analysis program. At the time, my primary experience was in C and C++ software development and engineering. I had some very limited experience with web development in industry, primarily in a production environment that created dynamic web pages using ASP and SQL databases on windows platforms. The primary developers on these projects were the old-fashioned types who hand-coded everything using a simple editor and shunned any kind of integrated development tools or frameworks. I had also taken classes on HTML and JavaScript, and had experimented a little with ASP.NET. So, to use a tried and true phrase, I knew enough to be dangerous.

I began a mission to figure out exactly what should be in a web curriculum for a two-year program, looking at what other institutions offered and more importantly, trying to get a sense of what was actually used in industry. I consulted with professionals currently working in the field regarding the basic skills that they felt were important for web developers. Step one was to define what web programming is and what philosophy should guide a curriculum.

Defining Web Programming

Web programming for our purposes was defined as a means to achieve networked delivery of information. In practice, it can mean a wide variety of things, including but not limited to, web page construction, multimedia production, server side scripting, client applications, and server applications. The field is relatively young and is rapidly evolving; what a business did to create a presence on the web just 5 years ago is likely to be different than what a business does today.

In addition, the business and employment opportunities for information technology overall are in an uncertain state. As McGee and Chabrow [2] recently noted,

Joblessness has nearly doubled in the last three years, while the number of Americans calling themselves IT professionals has decreased by nearly 160,000.

Outsourcing is often stated as a cause for this downswing, but a greater impact has probably been made by the overall maturation of the software industry. Companies no longer need to create original computer software as a wide range of sophisticated software has become commercially available. Companies that just a few years ago would have hired programmers to write proprietary software can now purchase what they need. More emphasis is on hiring technology professionals who can help integrate purchased or pre-existing software into their processes as well as support users of purchased software.

Where web development is concerned, programming has evolved very rapidly from the days of hand-coding HTML to use of high-level development tools with a focus on integration of varying components, for instance, bringing together text, graphics, sound, and animations. Web page elements like shopping carts or forums are readily available. There are a great many integrated development environments and programs designed to generate useful web pages. With the wide range of applications and the dynamic nature of web programming, it became clear that it would be a challenge to define a curriculum that was relevant, useful, and acceptable for a traditional academic environment.

In general, traditional computer science departments in many universities have been slow to offer courses in web programming and development. Education in this area has primarily come from technical schools, short course training, information technology departments, or in a great many cases, self-teaching. I believe that the primary reason for this is that traditional computer educators view web programming more as an application than a foundation or discipline. In addition, as companies jumped on the bandwagon to get a presence on the web, they utilized and retrained their existing programmers.

Curriculum Philosophy

After consultation with web professionals and other academics, an initial philosophy was developed, with the expectation was that students should experience the following to at least an introductory level.

- Foundation elements in computer programming and problem solving
- Basics of web page development directly using HTML/XHTML, JavaScript, etc.

- Use of at least one web page production package, such as Macromedia Dreamweaver
- Integration of purchased or pre-existing software into a cohesive client interface, such as integration of a shopping cart or user forum into a web page
- Production and/or integration of interactive media, such as flash animations, graphics, video, or musical elements
- Server-side application development, including scripting
- Applets
- Database fundamentals
- Web page design and usability concepts
- Accessibility issues and alternatives
- Business knowledge in electronic commerce and information management
- Web server administration and security fundamentals
- Information architecture and Web project management
- Awareness of standards, such as those moderated by W3C
- Cross-platform issues and alternatives
- Life cycle development processes from requirements gathering to design to implementation to testing to release

Our target student populations included all the normal suspects.

- First-year students
- Professionals in related disciplines, for instance, graphics designers who wish a more technical background
- Other non-traditional students upgrading or changing job skills

This translated to an initial set of courses that could be added to our core programming curriculum to complete an Associates Degree in Web Programming.

- Web Client Programming
- Database Design and Development
- Web Server Programming
- Systems Analysis and Design
- Survey of Operating Systems
- Business and the Internet
- Web Site Practicum

Electives included a variety of multimedia courses, object-oriented programming, information studies, and basic art.

Reality

After defining the curriculum philosophy and initial set of courses, the time came to plan courses, justify the need for a web programming concentration, and submit everything for approval. At that point, it became unclear if a web programming concentration was completely viable. Students would leave the program with rudimentary knowledge in preparing web pages and some server-side applications, including basic database operations. From the standard core courses, they would also have some basic programming knowledge.

More discussion brought doubts that emphasizing delivery of web pages would be good preparation for future jobs. The numbers of companies that provide those kinds of services have dwindled, at least in our area. Many smaller businesses either don't utilize the web or have a relatively static web presence. A very informal survey of jobs available in our area for a 3 month period on monster.com showed a relatively low number (29) that fit into a web development category.

Table 1 shows the skills that employers were looking for under the umbrella of web development or programming. Of the posted opportunities, 63% required HTML skills, and only 43% were looking for JavaScript skills. On the other hand, 80% were looking for database skills and 63% were looking for general programming experience. In general, there were a wide range of skills that employers listed as requirements for positions related to web development. That was not a surprise, as previously noted, web development can involve a wide range of activities. However, it was somewhat surprising that from an employment perspective, general programming and database skills were as important as any specific web-based skills.

Skill	% Required	% Suggested
ASP	33	3
.NET	27	13
SQL Server	33	3
Oracle	7	0
Other DB	40	3
HTML/XML	63	0
DHTML/JavaScript	43	0
WML	3	0
CSS	10	0
SOAP	10	0
Programming (C++, C# Java)	60	3
COM/Services	23	0
Design/development/support tools	33	10
Web development tools	30	3
GUI design	27	0
Graphics	7	3
Flash	10	3
Systems	20	7

Table 1 Required and suggested skills for recent job postings

Also somewhat surprising was the number of years of experience employers were looking for, with 34% requiring 3 – 6 years of experience. In addition, stated degree requirements for web development positions show that only 35% requiring some kind of advanced degree. This suggests more emphasis on experience over education where employers are concerned, which I believe is true in general based on my own industry experience. However, where web programming is concerned, I have to ask if employers are looking for experience simply because new graduates are not likely to have much web programming experience.

Years Experience	% Required
0	28
1 or more	10
2 or more	28
3 or more	14
5 or more	17
6 or more	3

Table 2 Required experience for recent job postings

Degree	% Required
None	62
High School	3
Associate	7
Bachelor	28

Table 3 Required degrees for recent job postings

Shifting Problems

When I first started to look at what should be required for a web programming curriculum, I identified several areas that I thought would be problematic. These included the wide range of sophistication with technology seen in beginning students, the dynamic nature of the web in general, selection of hardware platforms for laboratories and classrooms, and finding potential industry partners for practicum courses.

After trying to determine the potential employability of students granted an Associates Degree with a web programming concentration, one conclusion might be that such an option is not viable. As noted by authors looking at the state of technology-related employment [1, 2], future job growth may only be available in highly skilled sectors. For the present, the need to educate workers specifically to do web development may have passed. It appears that education in basic programming skills with some options for web application development may be sufficient if recent job postings are indicative of a general climate.

Conclusions

A year ago I started the task of developing a web programming curriculum for a two-year degree track within an information technology program. A year later the only thing that seems clear is that web development and its associated employment options may be changing too rapidly to capture in a university curriculum, and I'm left with more questions than conclusions.

Is it enough for applied programs to teach students basic programming and database skills, and attempt to add in whatever web programming skills are currently in demand? The real key is anticipating what technologies and practices will be in use, and the development of courses and programs at universities seems to move too slowly to keep up with something that is currently as dynamic as web programming. How can an information technology department at a university develop relationships with industry such that they can anticipate and prepare students for future employment?

References

[1] Levy, F. and Murnane R.J., *The new division of labor*. Princeton University Press and Russell Sage Foundation, 2004.

[2] McGee, M.K. and Chabrow, E., "Tech-job upheaval", *informationweek.com*, Aug 2, 2004, pp 20-22.