

# Web Engineering Curriculum: A Case Study of an Evolving Framework

Yogesh Deshpande

University of Western Sydney  
Locked Bag 1797, Penrith South DC  
NSW 1797, Australia  
[y.deshpande@uws.edu.au](mailto:y.deshpande@uws.edu.au)

**Abstract.** In their comprehensive review of computing disciplines, the Joint (ACM and IEEE-CS) Task Force on Computing Curricula identified a group of subjects as Net-centric, to be taught at under-graduate level. Web Engineering was still in its infancy at the time. We started a Web Engineering specialization at master's level in 1999 and have recently reviewed it comprehensively. Based on our experience in teaching different aspects of Web development at both under-graduate and graduate levels, this paper proposes a framework to design curricula for Web Engineering that can evolve in tandem with the evolution of the Web. The framework helps to dovetail the knowledge areas in a coherent manner avoiding a smorgasbord approach to curriculum design.

## 1 Introduction

The need for disciplined approaches and new methods and tools for development has been well recognised by Web site and application developers and expert commentators [2, 4, 10-12]. Originating from Document Engineering [13] and Web Site Engineering [10] Web Engineering continues to develop as a specific area of work, as attested to by workshops, special tracks and international conferences, along with two new journals [8, 9]. Web Engineering arose out of widespread and genuine concerns about how Web sites and applications were, and are, being developed. They led to a working definition of Web Engineering as “the application of systematic, disciplined and quantifiable approaches to development, operation and maintenance of Web-based applications” [3].

The phenomenal growth of the Web created a matching demand in courses that taught Web site and application development. Commercial training institutions were the first to offer courses in Web technologies and at the height of the dot com boom numbered in tens of thousands across the world. Universities were generally uninterested in offering such short training courses but dealt with the new technologies by initiating or modifying undergraduate courses in specific areas, such as programming, networking, human-computer interaction and so on. At a more generic level, the ACM and IEEE-CS, in their most recent review of computing curricula [1], published in 2001, identified net-centric computing as one of the 14 areas to represent the body of knowledge for computer science. The development of Web Engineering has been too recent for an explicit mention as a body of knowledge.

The first Workshop on Web Engineering at WWW7 devoted half a session to the discussion of appropriate graduate curriculum. As a result, at the University of Western Sydney (UWS), we offered Web Engineering as a specialization at master's level in 1999. Elsewhere, Greenspun has described how he and his colleagues taught Web development at MIT without dealing with curriculum design at a more general level. Whitehead [14] has specifically addressed the formulation and issues of Web Engineering curriculum at graduate level by identifying the relevant knowledge areas.

The master's course in Web Engineering at UWS has just undergone a full review. Since Web Engineering was not covered by the Task Force, we did not have a specific and relevant framework for us to work with. However, their recommendations have been useful in the exercise as also the references cited above.

During our deliberations, it became clear that the identification of knowledge areas may turn out to be too specific and time-bound. Web Engineering is still young. The constituent body of work is rapidly building up and, like the Web itself, will continue to evolve. The teaching of Web Engineering, therefore, requires a blend of currently established body of work and ongoing research. Graduate students in Web Engineering have to be prepared for a level of enquiry bordering on research work and experiments with emerging technologies. Designing an appropriate curriculum for Web Engineering thus involves more than identifying knowledge areas.

Consequently, we had to construct a framework for the Web Engineering curricula which would be flexible enough to accommodate both the existing and evolving knowledge areas. This short paper broadens the framework and develops the underlying principles, to facilitate further discussions by specifically addressing the curricular issues. It does not cover details of our implementation and revisions. Section 2 describes and briefly discusses the initial framework and aims and objectives of the course. Section 3 presents the revised framework. Section 4 concludes the paper with recommendations for further work and discussions.

## 2 Initial Framework for Web Engineering Curriculum

One way to arrive at a curriculum would be to answer the following, sequential questions.

1. What are the aims of Web Engineering?
2. What are the activities comprising Web Engineering?
3. What are the knowledge areas that will facilitate the activities?
4. How can these knowledge areas be combined to build a curriculum?

These are broad questions that do not consider specific, pedagogical and institutional concerns, in the same way that the Task Force left the responsibility of tailoring degree courses to individual institutions. The four questions, in their logical order, are predicated on a relatively stable field of study. They assume that a field of study such as Web Engineering 'exists' and that there are 'activities' in this field that people agree upon. It is then a question of identifying the knowledge areas relevant to these activities in order to build a curriculum.

However, Web Engineering did not evolve so logically nor has its evolution stopped. The limited role of activities in defining a curriculum in this context becomes obvious as the early concerns with Document Engineering and Web Site Engineering were overtaken by the growing complexity of Web development and the

experiences of Web developers and researchers during that time. Figure 1 from [3], encapsulates those concerns in six rising levels of complexity.

<b>6. Web Project Planning and Management</b>
<b>5. Web-based System</b>
<b>4. Web Site Construction</b>
<b>3. Web Site Design</b>
<b>2. Web Page Design</b>
<b>1. Web Page Construction</b>

**Fig. 1.** Levels of complexity of Web development [11]

We first started teaching Web development at the undergraduate level, in 1997. Initially, the subjects built up the students’ skills and knowledge to level 4, viz. Web Site Construction, appropriate for the undergraduates. However, such a focus does not fully address a whole gamut of issues, such as life-long learning (i.e. developing capabilities to master new technologies and developments) or to discharge their social, legal, ethical and professional responsibilities that come to the fore more forcefully when dealing with the Web than is the case with normal software development. Consequently, we decided to offer a specialization in Web Engineering at the master’s level to tackle two additional questions:

- 5. How much of the complexity, technical and non-technical, should be addressed directly by the curriculum and indirectly through student projects?
- 6. How can future developments be accommodated in building a curriculum?

The specialization in Web Engineering was structured according to Figure 1, enrolled computing graduates, and addressed the six questions in its delivery through case studies, active discussions, assignments and feedback on students’ projects.

**2.1 Aims and Objectives of a Course in Web Engineering**

Based on the previous discussion, we started the master’s course with the aims to:

- 1. provide graduates with a critical appreciation of IT/IS methodologies, the Web technologies, innovation, and their roles within an organisation;
- 2. enable graduates to build Web applications using multimedia and hypermedia based upon sound methodologies in project management and people management;
- 3. prepare graduates to fulfil their social, legal, ethical and professional obligations through case studies and on-going analysis of current developments;
- 4. develop graduates’ expertise in new technologies used in intranets, extranets and the Internet;
- 5. enable graduates to administer Web and application servers and manage content and information;
- 6. provide a global business and IT perspective to the graduates.

### 3 Revised Framework for Web Engineering Curriculum

Figure 1, over time, proved to be too specific (e.g. in its attention to page construction and design). The students also started to acquire some of these skills at undergraduate levels. Hence, we began the review by revising Figure 1 to Figure 2.

<b>4. Project Management</b>
<b>3. Web Applications</b>
<b>2. Web sites</b>
<b>1. Internet Technologies</b>

**Fig. 2.** A Framework for Web Engineering Curriculum

The revised model facilitated further analysis of where each relevant technology, protocol and standard fits in, what issues (social, ethical, legal and others) need to be addressed at each layer and how to understand and manage each layer in curricular terms. However, neither figure properly accommodates the recent arrivals of Web Services, Mobile Computing and component-based application development. These are more advanced topics with a degree of complexity and the interdependence not found in the early Web applications. Hence, we extended the model as in Figure 3.

<b>Project Management</b>		
<b>Web Services</b>	<b>Mobile Computing</b>	<b>... (advanced apps)</b>
<b>Web Applications (incl Server Management)</b>		
<b>Web sites</b>		<b>Non-Web site Internet (component-based applications)</b>
<b>Internet Technologies</b>		

**Fig. 3.** A Revised Framework for Web Engineering

Figure 3 is an abstraction (and a superset of Figure 2) that allows for identification and development of other layers and sub-layers. The model combines the treatment of knowledge areas and complexities associated with Web developmental activities, and consciously attempts to address the issues of future developments as they may affect the curricula. The Web Engineering curriculum can now be formulated as part core and part optional or elective to suit local conditions.

### 4 Recommendations for Future Work

This short paper has presented a framework for a Web Engineering curriculum and its evolution. The major advantage of the framework is that it separates out levels of complexity in the activities that constitute Web Engineering, making it flexible and adaptable to the changing circumstances. It also clarifies the place of relevant knowledge areas. It then becomes possible to think of ‘flavours’ of curriculum to suit the local, institutional conditions, strengths and weaknesses, pedagogical issues and

research strengths. Examples of these variations can be cited in terms of student demographics and expectations, demand for more 'topical' courses, local industry participation through project work, available expertise, and research trends. Furthermore, 'the proof of the pudding' means that all these courses must be rigorously evaluated from the students' and employers' points of view and the actual outcomes. These issues will be the subject of another paper.

Finally, it needs to be acknowledged that this framework has not taken on board what might be regarded as 'information systems', i.e. more sociological and/or management, point of view. More work is needed to identify and analyse these issues and then solve the concomitant problems.

**Acknowledgments.** The author would like to thank all the members of the School of IT and other colleagues who participated in the development of this framework.

## References

1. CC01 (2001) The Joint Task Force on Computing Curricula, IEEE Computer Society and Association for Computing Machinery, *Computing Curricula 2001*, vols 1 and 2, Ironman Draft, February 2001, <http://www.acm.org/>
2. Cutter Consortium, *Research Briefs*, 7 Nov 2000
3. Deshpande, Y., Murugesan, S., Ginige, A. Hansen, S., Schwabe, D., Gaedke, M. and White, B. (2002) Web Engineering, *Journal of Web Engineering*, vol 1, no. 1, 3-17
4. Glass, R. (2001) Who's Right in the Web Development Debate?, *Cutter IT Journal*, vol 14, no. 7, pp 6-10
5. *IEEE Multimedia*, Special issues on Web Engineering, vol 8, nos 1 and 2, Jan-Mar 2001 and Apr-Jun 2001
6. *International Journal of Web Engineering and Technologies (IJWET)*  
<http://www.inderscience.com/ijwet/>
7. *Journal of Web Engineering*, Rinton Press, <http://www.rintonpress.com/journals/jwe>
8. Murugesan, S., Deshpande, Y., Hansen, S. and Ginige, A. (1999) Web Engineering: A New Discipline for Development of Web-based Systems, *Proceedings of the First ICSE Workshop on Web Engineering*, International Conference on Software Engineering, Los Angeles, May 1999.  
<http://aeims.uws.edu.au/WebEhome/ICSE99-WebE-Proc/San.doc> (shortly to be restored)
9. Murugesan, S. and Deshpande, Y. (2001) *Web Engineering*, Lecture Notes in Computer Science- Hot Topics, vol 2016, Springer Verlag, 2001
10. Powell, T.A. (1998) *Web Site Engineering*, Prentice-Hall, Upper Saddle River, NJ
11. Pressman, R.S. (1998) Can Internet-Based Applications Be Engineered? *IEEE Software*, September/October 1998
12. Pressman, R.S. (2001) Web Engineering: An Adult's Guide to Developing Internet-Based Applications, *Cutter IT Journal*, vol 14, no. 7, pp 2-5
13. White, B. (1996) Web Document Engineering, SLAC-PUB-7150, May 1996, <http://www.slac.stanford.edu/pubs/slacpubs/7000/slac-pub-7150.html> (based on a tutorial presented at the World Wide Web Conference WWW5, Paris, 1995)
14. Whitehead, J. (2002) A Proposed Curriculum for a Masters in Web Engineering, *JWE*, vol 1, no. 1