Chapter I

Web Engineering: Introduction and Perspectives

San Murugesan
Southern Cross University, Australia

Athula Ginige
University of Western Sydney, Australia

Abstract

Web-based systems and applications now deliver a complex array of functionality to a large number of diverse groups of users. As our dependence and reliance on the Web has increased dramatically over the years, their performance, reliability and quality have become paramount importance. As a result, the development of Web applications has become more complex and challenging than most of us think. In many ways, it is also different and more complex than traditional software development. But, currently, the development and maintenance of most Web applications is chaotic and far from satisfactory. To successfully build and maintain large, complex Web-based systems and applications, Web developers need to adopt a disciplined development process and a sound methodology. The emerging discipline of Web engineering advocates a holistic, disciplined approach to successful Web development. In this chapter, we articulate and raise awareness of the issues and considerations in large, complex Web application development, and introduce Web engineering as a way of managing complexity and diversity of large-scale Web development.

Introduction

Within a decade, the World Wide Web has become ubiquitous, and it continues to grow unabated at exponential rate. Web-based systems and applications now deliver a complex array of varied content and functionality to a large number of heterogeneous users. The interaction between a Web system and its backend information systems has also become more tight and complex.

As we now increasingly depend on Web-based systems and applications, their performance, reliability and quality have become paramount importance, and the expectations of and demands placed on Web applications have increased significantly over the years. As a result, the design, development, deployment and maintenance of Web-based systems have become more complex and difficult to manage.

Though massive amounts of Web development and maintenance continue to take place, most of them are carried out in ad hoc manner, resulting in poor quality Web systems and applications. Problems such as outdated or irrelevant information, difficulties in using the Web site and finding relevant information of interest, slow response, Web site crashes, and security breaches are common. We encounter these kinds of problems because Web developers failed to address users' needs and issues such as content management, maintenance, performance, security, and scalability of Web applications. They also often overlook important non-technical considerations such as copyright and privacy.

Many Web developers seem to think that Web application development is just simple Web page creation using HTML or Web development software such as *Front Page* or *Dreamweaver* and embodying few images and hyperlinking documents and Web pages. Though certain simple applications such as personal Web pages, seminar announcements, and simple online company brochures that call for simple content presentation and navigation fall into this category, many Web applications are complex and are required to meet an array of challenging requirements which change and evolve. There is more to Web application development than visual design and user interface. It involves planning, Web architecture and system design, testing, quality assurance and performance evaluation, and continual update and maintenance of the systems as the requirements and usage grow and develop.

Hence, ad hoc development is not appropriate for large, complex Web systems, and it could result in serious problems: the delivered systems are not what the user wants; they are not maintainable and scalable, and hence have short useful life; they often do not provide desired levels of performance and security; and/or most Web systems are often much behind schedule and overrun the budget estimates.

More importantly, many enterprises and organisations cannot afford to have faulty Web systems or tolerate downtime or inconsistent or stale content/information. The problems on the Web become quickly visible and frustrate the users, possibly costing the enterprises heavily in terms of financial loss, lost customer and loss of reputation. As is often said, "We cannot hide the problems on the Web."

Unfortunately, despite being faced with these problems and challenges, most Web application development still continues to be ad hoc, chaotic, failure-prone, and unsat-

isfactory. And this could get worse as more inherently complex Web systems and applications that involve interaction with many other systems or components pervade us and our dependence on them increases.

To successfully build large-scale, complex Web-based systems and applications, Web developers need to adopt a disciplined development process and a sound methodology, use better development tools, and follow a set of good guidelines.

The emerging discipline of Web engineering addresses these needs and focuses on successful development of Web-based systems and applications, while advocating a holistic, disciplined approach to Web development.

Web Engineering uses scientific, engineering, and management principles and systematic approaches to successfully develop, deploy, and maintain high-quality Web systems and applications (Murugesan et al., 1999). It aims to bring Web-based system development under control, minimise risks and improve quality, maintainability, and scalability of Web applications.

The essence of Web engineering is to successfully manage the diversity and complexity of Web application development, and hence, avoid potential failures that could have serious implications.

This chapter aims to articulate and raise awareness of the issues and considerations in large-scale Web development and introduce Web engineering as a way of managing complexity and diversity of large-scale Web development.

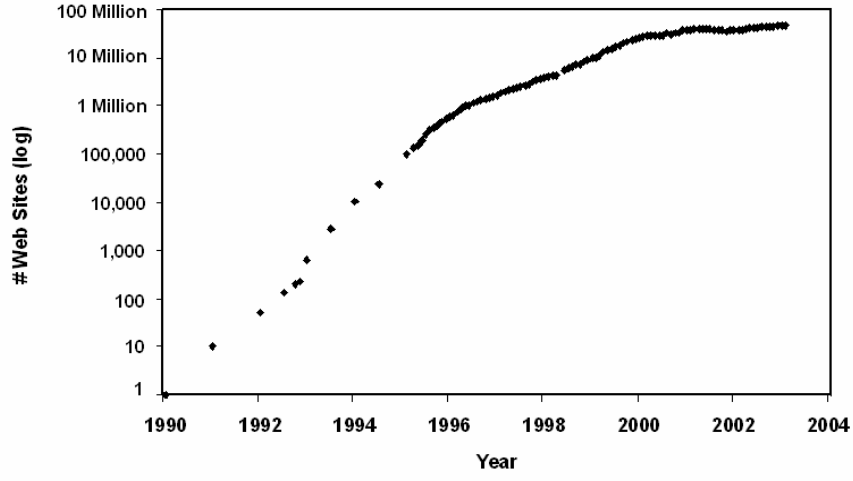
Following a brief outline of the evolution of the Web and the categorisation of Web applications based on their functionality, this chapter examines current Web development practices and their limitations, and emphasises the need for a holistic, disciplined approach to Web development. It then presents an overview of Web engineering, describes an evolutionary Web development process, discusses considerations in Web design and recommends ten key steps for successful development. In conclusion, it offers perspectives on Web Engineering and highlights some of the challenges facing Web developers and Web engineering researchers.

Evolution of the Web

The Web has become closely ingrained with our life and work in just a few years. From its initial objective of facilitating easy creation and sharing of information among a few scientists using simple Web sites that consisted primarily of hyperlinked text documents, the Web has grown very rapidly in its scope and extent of use, supported by constant advances in Internet and Web technologies and standards. In 10 years, the number of Web sites dramatically has grown from 100 to over 45 million (Figure 1).

Enterprises, travel and hospitality industries, banks, educational and training institutions, entertainment businesses and governments use large-scale Web-based systems and applications to improve, enhance and/or extend their operations. E-commerce has become global and widespread. Traditional legacy information and database systems are being progressively migrated to the Web. Modern Web applications run on distributed

Figure 1. Growth of Web sites



Note: Web Sites = Number of Web servers; one host may have multiple sites by using different domains or port numbers.

Source: Hobbes' Internet Timeline, 2004, www.zakon.org/robert/internet/timeline/

hardware and heterogeneous computer systems. Furthermore, fuelled by recent advances in wireless technologies and portable computing and communication devices, a new wave of mobile Web applications are rapidly emerging. The Web has changed our lives and work at every level, and this trend will continue for the foreseeable future.

The evolution of the Web has brought together some disparate disciplines such as media, information science, and information and communication technology, facilitating easy creation, maintenance, sharing, and use of different types of information from anywhere, any time, and using a variety of devices such as desktop and notebook computers, pocket PCs, personal digital assistants (PDAs), and mobile phones. Contributions of each of these disciplines to the evolution and growth of the Web are:

- **Media:** integration of different types of media such as data, text, graphics, images, audio and video, and their presentation (animation, 3D visualisation); different types of interaction and channels of communications (one-to-one, one-to-many, many-to-one, and many-to-many).
- **Information science:** information organisation, presentation, indexing, retrieval, aggregation, and management; and collaborative and distributed content creation.
- **Information and communication technology and networking:** efficient and cost-effective storage, retrieval, processing, and presentation of information; infrastructures that facilitate transfer and sharing of data and information; wired and wireless Internet communication; and personalised and context-aware Web applications.

Table 1. Categories of Web applications based on functionality

Functionality/Category	Examples
Informational	Online newspapers, product catalogues, newsletters, manuals, reports, online classifieds, online books
Interactive	Registration forms, customized information presentation, online games
Transactional	Online shopping (ordering goods and services), online banking, online airline reservation, online payment of bills
Workflow oriented	Online planning and scheduling, inventory management, status monitoring, supply chain management
Collaborative work environments	Distributed authoring systems, collaborative design tools
Online communities, marketplaces	Discussion groups, recommender systems, online marketplaces, e-malls (electronic shopping malls), online auctions, intermediaries

Many new Web technologies and standards have emerged in the last couple of years to better support new, novel Web applications: XML, Web services, the Semantic Web, Web personalisation techniques, Web mining, Web intelligence, and mobile and context-aware services.

The advances in Internet and Web technologies and the benefits they offer have led to an avalanche of Web sites, a diverse range of applications, and phenomenal growth in the use of the Web.

Categories of Web Applications

The scope and complexity of Web applications vary widely: from small scale, short-lived (a few weeks) applications to large-scale enterprise applications distributed across the Internet, as well as via corporate intranets and extranets. Web applications now offer vastly varied functionality and have different characteristics and requirements. Web applications can be categorised in many ways — there is no unique or widely accepted way. Categorisation of Web applications based on functionality (Table 1) is useful in understanding their requirements and for developing and deploying Web-based systems and applications.

Web Development Practices

Web development has a very short history, compared to the development of software, information systems, or other computer applications. But within a period of few years, a large number of Web systems and applications have been developed and put into widespread use.

The complexity of Web-based applications has also grown significantly — from information dissemination (consisting of simple text and images to image maps, forms, common gateway interface [CGI], applets, scripts, and style sheets) to online transactions, enterprise-wide planning and scheduling systems, Web-based collaborative work environments, and now multilingual Web sites, Web services and mobile Web applications.

Nevertheless, many consider Web development primarily an authoring work (content/page creation and presentation) rather than application development. They often get carried away by the myth that "Web development is an art" that primarily deals with "media manipulation and presentation." Sure, like the process of designing and constructing buildings, Web development has an important artistic side. But Web development also needs to follow a discipline and systematic process, rather than simply hacking together a few Web pages.

Web applications are not just Web pages, as they may seem to a causal user. The complexity of many Web-based systems is often deceptive and is not often recognised by many stakeholders — clients who fund the development, Web development managers and Web developers — early in the development.

Several attributes of quality Web-based systems such as usability, navigation, accessibility, scalability, maintainability, compatibility and interoperability, and security and reliability often are not given the due consideration they deserve during development. Many Web applications also fail to address cultural or regional considerations, and privacy, moral and legal obligations and requirements. Most Web systems also lack proper testing, evaluation, and documentation.

While designing and developing a Web application, many developers fail to acknowledge that Web systems' requirements evolve, and they do not take this into consideration while developing Web systems. Web-based systems development is not a one-time event as perceived and practiced by many; it is a process with an iterative lifecycle.

Another problem is that most Web application development activities rely heavily on the knowledge and experience of individual (or a small group of) developers and their individual development practices rather than standard practices.

Anecdotal evidence and experience suggest that the problems of ad hoc development (outlined above and in the Introduction section) continue to be faced by developers, users, and other stakeholders. As a result, these are increasing concerns about the manner in which complex Web-based systems are created as well as the level of performance, quality, and integrity of these systems.

> *"Many organisations are heading toward a Web crisis in which they are unable to keep the system updated and/or grow their system at the rate that is needed. This crisis involves the proliferation of quickly 'hacked together' Web systems that are kept running via continual stream of patches or upgrades developed without systematic approaches." (Dart, 2000)*

Poorly developed Web-based applications have a high probability of low performance and/or failure. Recently, large Web-based systems have had an increasing number of

failures (Williams, 2001). In certain classes of applications such as supply-chain management, financial services, and digital marketplaces, a system failure can propagate broad-based problems across many functions, causing a major Web disaster. The cost of bad design, shabby development, poor performance, and/or lack of content management for Web-based applications has many serious consequences.

The primary causes of these failures are a lack of vision, shortsighted goals, a flawed design and development process, and poor management of development efforts — not technology (Ginige & Murugesan, 2001a). The way we address these concerns is critical to successful deployment and maintenance of Web applications.

Therefore, one might wonder whether development methodologies and processes advocated over the years for software or information systems development and software engineering principles and practices could be directly used for developing Web applications. Though the valuable experiences gained and some of processes and methodologies used in software engineering (and other domains) could be suitably adapted for Web development as appropriate, they are not adequate, as Web development is rather different from software development in several aspects.

Web Development is Different

It is important to realise that Web application development has certain characteristics that make it different from traditional software, information system, or computer application development (Deshpande et al., 2002; Deshpande & Hansen, 2001; Ginige & Murugesan, 2001a, 2001b; Glass, 2001; Lowe 2003; Murugesan et al., 1999; Pressman, 2001 and 2004).

Web applications have the following characteristics:

- Web applications constantly evolve. In many cases, it is not possible to fully specify what a Web site should or will contain at the start of the development process, because its structure and functionality evolve over time, especially after the system is put into use. Further, the information contained within and presented by a Web site will also change. Unlike conventional software that goes through a planned and discrete revision at specific times in its lifecycle, Web applications continuously evolve in terms of their requirements and functionality (instability of requirements). Managing the change and evolution of a Web application is a major technical, organisational and management challenge — much more demanding than a traditional software development.
- Further, Web applications are inherently different from software. The content, which may include text, graphics, images, audio, and/or video, is integrated with procedural processing. Also, the way in which the content is presented and organised has implications on the performance and response time of the system.
- Web applications are meant to be used by a vast, variable user community — a large number of anonymous users (could be many millions like in the cases of eBay and the 2000 Sydney Olympics Web site) with varying requirements, expectations, and

skill sets. Therefore, the user interface and usability features have to meet the needs of a diverse, anonymous user community to whom we cannot offer training sessions, thus complicating human-Web interaction (HWI), user interface, and information presentation.

- Nowadays, most Web-based systems are content-driven (database-driven). Web-based systems development includes creation and management of the content, as well as appropriate provisions for subsequent content creation, maintenance, and management after the initial development and deployment on a continual basis (in some applications as frequently as every hour or more).
- In general, many Web-based systems demand a good "look and feel," favouring visual creativity and incorporation of multimedia in presentation and interface. In these systems, more emphasis is placed on visual creativity and presentation.
- Web applications have a compressed development schedule, and time pressure is heavy. Hence, a drawn-out development process that could span a few months to a year or more is not appropriate.
- Ramifications of failure or dissatisfaction of users of Web-based applications can be much worse than conventional IT systems.
- Web applications are developed by a small team of (often young) people with diverse backgrounds, skills, and knowledge compared to a team of software developers. Their perception of the Web and the quality of Web-based systems also differ considerably, often causing confusion and resulting in misguided priorities.
- There are rapid technological changes — constant advances in Web technologies and standards bring their own challenges — new languages, standards, and tools to cope with; and lots of errors and bugs in early versions of new mark-up languages, development tools, and environments (technology instability).
- Web development uses cutting-edge, diverse technologies and standards, and integrates numerous varied components, including traditional and non-traditional software, interpreted scripting languages, HTML files, databases, images, and other multimedia components such as video and audio, and complex user interfaces (Offurt, 2002).
- The delivery medium for Web applications is quite different from that of traditional software. Web applications need to cope with a variety of display devices and formats, and supporting hardware, software, and networks with vastly varying access speeds.
- Security and privacy needs of Web-based systems are more demanding than that of traditional software.
- The Web exemplifies a greater bond between art and science than generally encountered in software development.

These unique characteristics of the Web and Web applications make Web development different and more challenging than traditional software development.

Web Engineering

Web engineering is way of developing and organising knowledge about Web application development and applying that knowledge to develop Web applications, or to address new requirements or challenges. It is also a way of managing the complexity and diversity of Web applications.

A Web-based system is a *living* system. It is like a garden — it continues to evolve, change, and grow. A sound infrastructure must be in place to support the growth of a Web-based system in a controlled, but flexible and consistent manner. Web engineering helps to create an infrastructure that will allow evolution and maintenance of a Web system and that will also support creativity.

Web engineering is application of scientific, engineering, and management principles and disciplined and systematic approaches to the successful development, deployment and maintenance of high quality Web-based systems and applications (Murugesan et al., 1999).

It is a holistic and proactive approach to the development of large Web-based systems, and it aims to bring the current chaos in Web-based system development under control, minimise risks, and enhance the maintainability and quality of Web systems.

Since its origin and promotion as a new discipline in 1998 (Deshpande, Ginige, Murugesan & Hansen, 2002; Murugesan, 1998), Web engineering is receiving growing interest among the stakeholders of Web-based systems, including developers, clients, government agencies, users, academics, and researchers. In addition, this new field has attracted professionals from other related disciplines such as multimedia, software engineering, distributed systems, computer science, and information retrieval.

Web Engineering is Multidisciplinary

Building a large, complex Web-based system calls for knowledge and expertise from many different disciplines and requires a diverse team of people with expertise in different areas. Web engineering is multidisciplinary and encompasses contributions from diverse areas: systems analysis and design, software engineering, hypermedia/hypertext engineering, requirements engineering, human-computer interaction, user interface, information engineering, information indexing and retrieval, testing, modelling and simulation, project management, and graphic design and presentation.

"Contrary to the perception of some professionals, Web Engineering is not a clone of software engineering, although both involve programming and software development" (Ginige & Murugesan, 2001a). While Web Engineering uses software engineering principles, it encompasses new approaches, methodologies, tools, techniques, and guidelines to meet the unique requirements of Web-based systems. As previously stated, development of Web-based systems is much more than traditional software development. There are subtle differences in the nature and lifecycle of Web-based and software systems, as well as the way in which they're developed and maintained. "Web development is a mixture between print publishing and software development, between

marketing and computing, between internal communications and external relations, and between art and technology" (Powell, 2000).

Evolution of Web Engineering

Web Engineering is progressively emerging as a new discipline addressing the unique needs and challenges of Web-based systems development. Since 1998, when the First Workshop on Web Engineering was held in Brisbane, Australia, in conjunction with the World Wide Web Conference (WWW7), there has been series of workshops and special tracks at major international conferences (WWW conferences 1999-2005, HICS 1999-2001, SEKE 2002 and 2003 and others), and a dedicated annual International Conference on Web Engineering (ICWE) 2002-2005.

There also have been a few special issues of journals on topics related to Web Engineering. There are two new dedicated journals, *Journal of Web Engineering* (www.rintonpress.com/journals/jweonline.html) and *Journal of Web Engineering and Technology* (www.inderscience.com), as well as an edited book, *Web Engineering: Managing Diversity and Complexity of Web Application Development* (Murugesan & Deshpande, 2001).

The bibliography at the end of this chapter gives details of special issues, conferences, books, and journal articles on Web engineering and other related areas.

New subjects and courses on Web engineering are now being taught at universities, both at undergraduate and postgraduate levels, and more research is being carried out on various aspects of Web engineering. Also, not surprisingly, there is growing interest among Web developers in using Web engineering approaches and methodologies.

Evolutionary Web Development

Web-applications are evolutionary. For many Web applications, it is not possible to specify fully what their requirements are or what these systems will contain at the start of their development and later, because their structure and functionality will change constantly over time. Further, the information contained within and presented by a Web site often changes — in some applications as often as every few minutes to a couple of times a day. Thus, the ability to maintain information and to scale the Web site's structure (and the functions it provides) is a key consideration in developing a Web application.

Given this Web environment, it seems the only viable approach for developing sustainable Web applications is to follow an evolutionary development process where change is seen as a norm and is catered to. And, this also mandates adoption of a disciplined process for successful Web development.

Web Development Process

A Web development process outlines the various steps and activities of Web-based systems development. It should clearly define a set of steps that developers can follow and must be measurable and trackable (Ginige & Murugesan, 2001c).

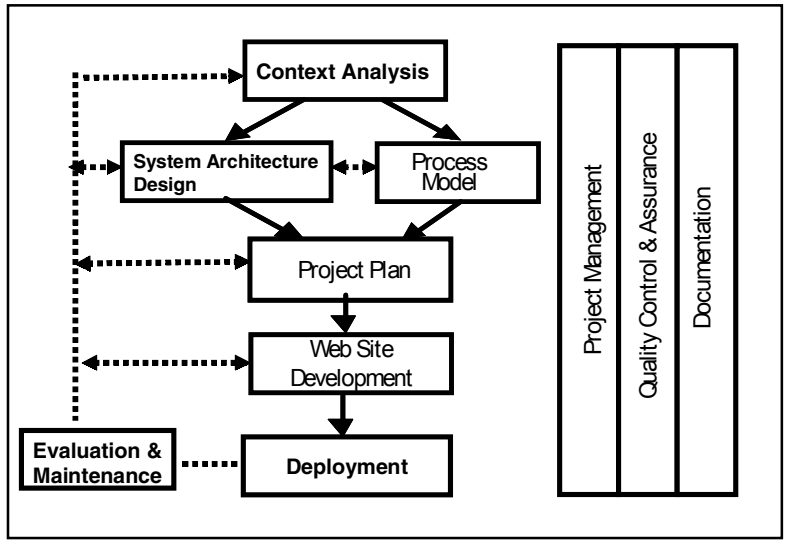
Characteristics of Web applications that make their development difficult — and uniquely challenging — include their real-time interaction, complexity, changeability, and the desire to provide personalised information. In addition, the effort and time required to design and develop a Web application is difficult to estimate with a reasonable accuracy.

Based on our practical experience in building Web applications, we recommend an evolutionary process for Web development, shown in Figure 2. This process assists developers in understanding the context in which the application will be deployed and used; helps in capturing the requirements; enables integration of the know-how from different disciplines; facilitates the communication among various members involved in the development process; supports continuous evolution and maintenance; facilitates easier management of the information content; and helps in successfully managing the complexity and diversity of the development process (Ginige & Murugesan 2001c).

Context Analysis

The first essential step in developing a Web-based system is "context analysis," where we elicit and understand the system's major objectives and requirements, as well as the

Figure 2. Web development process



needs of the system's typical users and the organisation that needs the system. It is important to realise at this stage that requirements will change and evolve — even during system development and after its deployment. It is also important to study briefly the operation for which a Web application is to be developed, and the potential implications of introduction of the new system on the organisation. This study should normally include: how information (to be made available on the Web) is created and managed; organisational policy on ownership and control (centralised or decentralised) of information; its current and future plans and business objectives; possible impact of the introduction of Web-based applications on the organisation; the resulting changes in its business and business processes; and emerging trends in the industry sector.

As the Web applications evolve and need to be modified to cater to new requirements — some of which arise from changes or improvements in the business process as a result of deployment of the new Web-based system — an understanding of a big picture about the organisation and its information management policies and practices is a prerequisite for successful design, development, and deployment of Web-based applications.

Before starting Web development, therefore, developers need to elicit and understand the system's major objectives and requirements, gather information about the operational and application environment, and identify the profile of typical system users.

In addition to the functional requirements, potential demands on the scalability, maintainability, availability, and performance of the system need to be specifically elicited and understood by the developers at the beginning of the development process. Based on this information, developers then arrive at the system's functional, technical, and non-technical requirements, which, in turn, influence the system's architectural design.

For instance, if the information content and the system's functions are going to evolve considerably, like in most e-business systems, the system needs to be designed for scalability. On the other hand, if the information changes frequently — like in weather reports, special sales offerings, job vacancies, product price list, brochures, and latest news or announcements — to keep the information current and consistent, the system needs to be designed for easy information maintainability (Merialdo et al., 2003). Moreover, where the application demands very high availability and needs to cater for high peak or uncertain demands, the system may be required to run on multiple Web servers with load balancing and other performance enhancement mechanisms (Almedia & Menasce, 2002; Menasce & Almedia, 2002; Oppenheimer & Patterson, 2002). Examples of this category of applications are online stock trading, online banking, and high volume near-real-time sports and entertainment Web sites such as the Olympics, Wimbledon, and Oscar Web sites.

Thus, it is very important to recognise that scalability, maintainability, and/or performance need to be built into the initial system architecture. It would be very hard, or impossible, to incorporate these features if the initial architecture is not designed to support them. To illustrate this, consider an e-business Web site that provides product information, such as price and availability, which appears on many different pages and changes frequently. If the Web site is designed as static Web pages, then every time a product's information changes, one has to incorporate the change in every page that contains this information. This is a cumbersome and laborious task, and often changes are only made to a few pages, instead of all relevant pages. As a consequence of this, the same information appearing on different pages will be inconsistent.

A better approach to ensure consistency of information across all Web pages is to automatically retrieve the information, when and where needed, from a single information source. If product information is stored in a single central database, then by extracting the relevant information from this database, we can dynamically create various Web pages that contain this information. In the database-driven approach, we need to change the information only in one place: the database. Further, the database-driven Web sites can have a back-end system to allow an authorised person, who may not be skilled in Web page development, to make information changes easily through a Web interface, from anywhere. A database-driven Web site requires a completely different architecture than a Web site that has only static Web pages. Hence, an appropriate architecture that would meet the system's requirements needs to be chosen early in the system development.

Thus, as highlighted in Table 2, the objective of context analysis is to capture and derive the key information required to develop the Web application. In addition, it can also identify non-technical issues that have to be addressed for successful implementation and application of the system. These may include reengineering of business processes where required, organisational and management policies, staff training, and legal, cultural and social aspects.

Context analysis can minimise or eliminate the major problems plaguing large Web-based system development. But, many developers and project managers overlook this essential first step in Web system development and face the problems later when it is hard to correct them.

Based on the context analysis, we then arrive at the system's technical and non-technical requirements (Lowe, 2003), which, in turn, influence the system architecture design.

Architecture Design

In system architecture design, we decide on various components of the system and how they are linked. At this stage, we design:

Table 2. Objectives of context analysis of Web applications

The objectives of context analysis, the first step in Web development, are to:
▪ Identify the stakeholders and their broader requirements and experiences.
▪ Identify the functions the Web site needs to provide (immediately, and in the short, medium, and long term).
▪ Establish what information needs to be on the Web site, how to get this information, and how often this information may change.
▪ Identify the corporate requirements in relation to look and feel, performance, security, and governance.
▪ Get a feel of the number of users (typical and peak) and anticipated demands on the system.
▪ Study similar (competitive) Web sites to gain an understanding of their functionalities, strengths, and limitations.

Table 3. Means of fulfilling the requirements of Web application

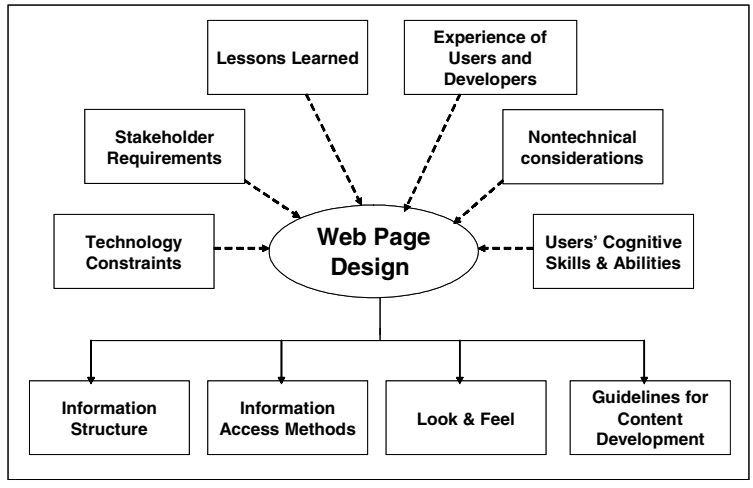
Requirement	Means of Fulfilment
Uniform look and feel across all Web pages that can easily be modified	Creation of Web pages using templates and style sheets
Consistency of information that may appear in different places or pages	Storing information in a single place (such as in a database or as an XML file), without duplication of information in different places or databases, and retrieving the required information for presentation where and when needed
Ease of information update and maintenance	Provision of a back-end system to edit information in a data repository; could have Web interface for easy access from anywhere
Ability to add new Web pages easily	Dynamic generation of navigational links, rather than predetermined static navigational links
Decentralised system administration	Provision of a multi-user login system to access back-end systems and inclusion of a "user administration system" that can assign specific functions and data sets to content managers and other developers/administrators
Mechanisms for quality control and assessing the relevance of information	Inclusion of metadata for Web pages; use of a Web robot for gathering salient information, processing the information gathered and taking appropriate action(s) for ensuring quality or relevance of information presented.
Increased probability of being found through search engines	Using meta tags and registering with search engines

- An overall system architecture describing how the network and the various servers (Web servers, application servers and database servers) interact;
- An application architecture depicting various information modules and the functions they support; and
- A software architecture identifying various software and database modules required to implement the application architecture.

Table 3 summarises the means of fulfilling some of the requirements of Web-based applications (Ginige & Murugesan, 2001c).

We then decide on an appropriate development process model (Uden, 2002; Pressman, 2004) and develop a project plan. To successfully manage Web development, a sound project plan and a realistic schedule are necessary. Progress of development activities must be monitored and managed. Project planning and scheduling techniques that are commonly used in other disciplines can be used for Web development. Following this, the various components of the system and Web pages are designed, developed and tested.

Figure 3. Web page design



Web Page Design

Web page design is an important activity; it determines what information is presented and how it is presented to the users. A prototype usually contains a set of sample pages to evaluate the page layout, presentation, and navigation (within and among different pages). Based on the feedback from the stakeholders, the page design is suitably modified. This process may go through a few iterations until the stakeholders and designers are satisfied with the page layout, presentation and the navigation structure.

Web page content development needs to take into consideration the stakeholders' requirements, users' cognitive abilities (Cloyd, 2001), technical issues and considerations, nontechnical issues, earlier experiences of developers and users, and lessons learned from similar Web applications (Figure 3).

If the Web system is intended for global use, by users from different countries, the Web content and presentation may have to be localised; there also may be a need for multilingual Web sites (for details, see Becker & Mottay, 2001; Collins, 2002). Also, the Web site's content and usability have to be designed from a global perspective and be responsive to cultural sensitivity in language along with appropriate use of colour, presentation, and animation (Becker & Mottay, 2001).

Web Maintenance

After a Web-based system is developed and deployed online for use, it needs to be maintained. As outlined earlier, content maintenance is a continual process. We need to formulate content maintenance policies and procedures, based on the decision taken at the system architecture design stage on how the information content would be main-

tained, and then we need to implement them. Further, as the requirements of Web systems grow and evolve, the system needs to be updated and also may be redesigned to cater to the new requirements.

It is important to periodically review Web-based systems and applications regarding the currency of information content, potential security risks, performance of the system, and usage patterns (by analysing Web logs), and take suitable measures to fix the shortcomings and weaknesses, if any.

Project Management

The purpose of project management is to ensure that all the key processes and activities work in harmony. Building successful Web-based applications requires close coordination among various efforts involved in the Web development cycle. Many studies, however, reveal that poor project management is the major cause of Web failures both during development and subsequently in the operational phase. Poor project management will defeat good engineering; good project management is a recipe for success. Successfully managing a large, complex Web development is a challenging task requiring multidisciplinary skills and is, in some ways, different from managing traditional IT projects.

Quality control, assurance and documentation are other important activities, but they are often neglected. Like project management, these activities need to spread throughout the Web development lifecycle.

Steps to Successful Development

Successful development of Web systems and applications involves multiple interactive steps which influence one another. We recommend the following key steps for successful development and deployment of Web applications (Ginige & Murugesan, 2001c):

1. Understand the system's overall function and operational environment, including the business objectives and requirements, organisation culture and information management policy.
2. Clearly identify the stakeholders — that is, the system's main users and their typical profiles, the organisation that needs the system, and who funds the development.
3. Elicit or specify the (initial) functional, technical, and nontechnical requirements of the stakeholders and the overall system. Further, recognise that these requirements may not remain the same; rather, they are bound to evolve over time during the system development.
4. Develop overall system architecture of the Web-based system that meets the technical and nontechnical requirements.

5. Identify subprojects or subprocesses to implement the system architecture. If the subprojects are too complex to manage, further divide them until they become a set of manageable tasks.
6. Develop and implement the subprojects.
7. Incorporate effective mechanisms to manage the Web system's evolution, change, and maintenance. As the system evolves, repeat the overall process or some parts of it, as required.
8. Address the nontechnical issues, such as revised business processes, organisational and management policies, human resources development, and legal, cultural, and social aspects.
9. Measure the system's performance, analyse the usage of the Web application from Web logs, and review and address users' feedback and suggestions.
10. Refine and update the system.

Web System Design: Challenges

The Internet is an open platform that provides unparalleled opportunities. But it has virtually no control over visitor volume, or when and how they access a Web system. This makes developing Web applications that exhibit satisfactory performance even under a sudden surge in number of users a nebulous and challenging task.

Satisfying the expectations and needs of different types of users with varying skills is not easy. When users find a site unfriendly, confusing, or presented with too much information, they will leave frustrated. Worse yet, these frustrated users may spread the bad news to many others. Web site usability factors include good use of colours, information content, easy navigation, and many more. They also include evaluation from an international perspective so that you can reach a global audience. Web usability factors that impact the Web user experience are (Becker & Berkemeyer, 2002): page layout, design consistency, accessibility, information content, navigation, personalisation, performance, security, reliability, and design standards (naming conventions, formatting, and page organisation).

A Web-based system also has to satisfy many different stakeholders besides the diverse range of users, including: persons who maintain the system, the organisation that needs the system, and those who fund the system development. These may pose some additional challenges to Web-based system design and development.

Today's Web-savvy consumers do not tolerate much margin of error or failure. Web system slow down, failure, or security breach may cause a loss of its customers — probably permanently. A whopping 58 percent of first time customers would not return to a site that crashed (Electronic Hit and Run, USA Today, 10 Feb 2000). According to a study (Inter@ctive Week, 6 Sep 1999), US$4.35 billion may be lost in e-business due to poor Web download speeds alone.

As Web applications are becoming mission-critical, there is greater demand for improved reliability, performance, and security of these applications.

Poor design and infrastructure have caused many Web applications to be unable to support the demands placed on them, so they have therefore failed. Many Web sites have suffered site crashes, performance failures, security breaches, and outages — resulting in irate customers, lost revenue, devalued stocks, a tarnished reputation (bad publicity, lack of customer confidence), permanent loss of customers, and law suits (Williams, 2001). Stock prices have become inextricably linked to the reliability of a company's e-commerce site.

The recent major failures and their impact on enterprises have served as a forceful reminder of the need for capacity planning, and improved performance, quality, and reliability. Successful Web application deployment demands consistent Web site availability, a better understanding of its performance, scalability, and load balancing. Proactive measures are needed to prevent grinding halts and failures from happening in the first place.

Large-scale Web system design is a complex and a challenging activity as it needs to consider many different aspects and requirements, some of which may have conflicting needs (Ivory & Hearst, 2002; Siegel, 2003; Cloyd, 2001).

We use terms like scalability, reliability, availability, maintainability, usability, and security to describe how well the system meets current and future needs and service-level expectations. These *-ilities* characterise (Williams, 2000) a Web system's architectural and other qualities. In the face of increasingly complex systems, these system qualities are often more daunting to understand and manage.

Scalability refers to how well a system's architecture can grow, as traffic, demand for services, or resource utilisation grows. As Web sites grow, small software weaknesses that had no initial noticeable effects can lead to failures, reliability problems, usability problems, and security breaches. Developing Web applications that scale well represents one of today's most important development challenges.

Flexibility is the extent to which the solution can adapt as business requirements change. A flexible architecture facilitates greater reusability and quicker deployment.

Thus, the challenge is to design and develop sustainable Web systems for better:

- Usability — interface design, navigation (Becker & Mottay 2001),
- Comprehension,
- Performance — responsiveness,
- Security and integrity,
- Evolution, growth, and maintainability, and
- Testability.

Web Testing and Evaluation

Testing plays a crucial role in the overall development process (Becker & Berkemeyer, 2002; Hieatt & Mee, 2002; Lam, 2001). However, more often than not, testing and evaluation are neglected aspects of Web development. Many developers test the system only after it had met with failures or limitations have become apparent, resorting to what is known as *retroactive testing*. What is desired in the first place is *proactive testing* at various stages of the Web development lifecycle. Benefits of proactive testing include assurance of proper functioning and guaranteed performance levels, avoidance of costly retroactive fixes, optimal performance, and lower risk.

Testing and validating a large complex Web system is a difficult and expensive task. Testing should not be seen as a one-off activity carried out near the end of development process. One needs to take a broad view and follow a more holistic approach to testing — from design all the way to deployment, maintenance, and continual refinement.

The test planning needs to be carried out early in the project lifecycle. A test plan provides a roadmap so that the Web site can be evaluated through requirements or design stage. It also helps to estimate the time and effort needed for testing — establishing a test environment, finding test personnel, writing test procedures before any testing can actually start, and testing and evaluating the system.

Lam (2001) groups Web testing into the following broad categories and provides excellent practical guidelines on how to test Web systems:

- Browser compatibility
- Page display
- Session management
- Usability
- Content analysis
- Availability
- Backup and recovery
- Transactions
- Shopping, order processing
- Internalisation
- Operational business procedures
- System integration
- Performance
- Login and security

Experience shows that there are many common pitfalls in Web testing and attempts should be made overcome them (Lam, 2001). Testing and evaluation of a Web application may be expensive, but the impact of failures resulting from lack of testing could be more costly or even disastrous.

Knowledge and Skills for Web Development

The knowledge and skills needed for large, complex Web application development are quite diverse and span many different disciplines. They can be broadly classified as:

- Technologies supporting and facilitating Web applications
- Design methods
  - Design for usability — interface design, navigation
  - Design for comprehension
  - Design for performance — responsiveness
  - Design for security and integrity
  - Design for evolution, growth and maintainability
  - Design for testability
  - Graphics and multimedia design
  - Web page development
- System architecture
- Web development methods and processes
- Web project management
- Development tools
- Content management
- Web standards and regulatory requirements

Web Development Team

As previously mentioned, development of a Web application requires a team of people with diverse skills and backgrounds (Hansen, 2004). These individuals include programmers, graphic designers, Web page designers, usability experts, content developers,

database designers and administrators, data communication and networking experts, and Web server administrators. A Web development team is multidisciplinary, like a film production team, and must be more versatile than a traditional software development team.

Hansen et al. (2001) presents a classification of the participants in a Web development team and a hierarchy for their skills and knowledge. This classification helps in forming a team and in devising a strategy for successful reskilling of the development team.

Conclusion

Web engineering is specifically targeted toward the successful development, deployment and maintenance of large, complex Web-based systems.

It advocates a holistic and proactive approach to developing successful Web applications. As more applications migrate to the Web environment and play increasingly significant roles in business, education, healthcare, government, and many day-to-day operations, the need for a Web engineering approach to Web application development will only increase. Further, as we now place greater emphasis on the performance, correctness, and availability of Web-based systems, the development and maintenance process will assume greater significance.

Web Engineering is an emerging discipline having both theoretical and practical significance. It is gaining the interest among researchers, developers, academics, and clients. This is evidenced by increased research activities and publications in this area, hosting of dedicated international conferences and workshops, publication of new journals devoted to Web Engineering, and universities offering special courses and programmes on the subject. It is destined for further advancement through research, education, and practice.

> *"To advance Web engineering, it is essential to define its core body of knowledge, to identify the areas in need of greater research and to develop a strategy to tackle the new technologies, new applications and the various technical, methodological, and societal issues that arise in tandem with such developments." (Deshpande, Olsina & Murugesan, 2002)*

Some of the areas that need further study, in no particular order, include:

- Web application delivery on multiple devices — desktop and pocket PCs, mobile phones, PDAs, TVs and refrigerators
- Context-aware Web applications and context-sensitive responses
- Device-independent Web access and content presentation

- Modelling and simulation of Web applications and systems
- Performance evolution and enhancement
- Testing and validation of systems
- Effort and cost estimation
- Web personalisation
- Quality control and assurance

No Silver Bullet!

Web Engineering will not make the problems and the risks go away. But, it can help you plan, monitor, control, and cope with the challenging task of developing large, complex Web applications. It will also facilitate making more informed decisions and developing better quality and better-engineered Web systems and applications.

It is important to understand the wider context in which a Web-based system or application will be used, and design an architecture that will support the development, operation, and maintenance as well as evolution of the Web application in that context, addressing the key issues and considerations. We strongly recommend that Web developers and project managers move away from an ad hoc, hacker-type approach to a well-planned, systematic, and documented approach for the development of large, high-performance, evolutionary, and/or mission-critical Web sites and applications.

Our key recommendations for successfully developing and implementing large, complex Web application are to:

- Adopt a sound strategy and follow a suitable methodology to successfully manage the development and maintenance of Web systems.
- Recognise that, in most cases, development of a Web application is not an event, but a process, since the applications' requirements evolve. It will have a start, but it will not have a predictable end as in traditional IT/software projects.
- Within the continuous process, identify, plan, and schedule various development activities so that they have a defined start and finish.
- Remember that the planning and scheduling of activities is very important to successfully manage the overall development, allocate resources, and monitor progress.
- Consider the big picture during context analysis, planning, and designing a Web application. If you do not, you may end up redesigning the entire system and repeating the process all over again. If you address the changing nature of requirements early on, you can build into the design cost-effective ways of managing change and new requirements.

- Recognise that development of a large Web application calls for teamwork and shared responsibility among the team members, so motivate a team culture.

Web engineering has been successfully applied in a number of Web applications. A well-engineered Web system is:

- Functionally complete and correct
- Usable
- Robust and reliable
- Maintainable
- Secure
- Perform satisfactorily even under flash and peak loads
- Scalable
- Portable, where required perform across different common platforms; compatible with multiple browsers
- Reusable
- Interoperable with other Web and information systems
- Universal accessibility (access by people with different kinds disabilities)
- Well-documented

Time to deploy an online Web system, though still important, is no longer a dominant process driver, as more emphasis is now placed on quality Web systems in terms of functionally, usability, content maintainability, performance, and reliability.

Web engineering can help enterprises and developers to convert their Web systems and applications from a potential costly mess into powerful resource for gaining sustainable competitive advantage.

Acknowledgments

The authors would like to thank Yogesh Deshpande and Steve Hansen, both from University of Western Sydney, Australia, for their contribution in origination and development of the Web engineering discipline and for their input on various aspects of Web development reported in this chapter which evolved through our collaborative efforts over the years. We would also like to thank our graduate students Anupama Ginige and Indra Seher who contributed to formulation and presentation some of the ideas presented in this chapter.

References

Almeida, V.A.F., & Menasce, D.A. (2002). Capacity planning for Web services: An essential tool for managing Web services. *IT Professional*, (July-August), 33-38.

Becker S., & Berkemeyer, A. (2002). Rapid application design and testing for usability. *IEEE Multimedia*, (Oct-Dec), 38-46.

Becker, S., & Mottay, F. (2001). A global perspective of Web usability for online business applications. *IEEE Software, 18*(1), 54-61.

Cloyd, M.H. (2001). Designing user-centered Web applications in Web time. *IEEE Software, 18*(1), 62-69.

Collins, R.W. (2002). Software localization for Internet software: Issues and methods. *IEEE Software,* (March/April), 74-80.

Dart, S. (2001). *Configuration management: A missing link in Web engineering*. Norwood, MA: Arttech House.

Deshpande, Y. et al. (2002). Web engineering. *Journal of Web Engineering, 1*(1), 3-17.

Deshpande, Y., Ginige, A., Murugesan, S., & Hansen, S., (2002). Consolidating Web engineering as a discipline. *SEA Software*, (April), 32-34.

Deshpande, Y., & Hansen, S. (2001). Web engineering: creating a discipline among disciplines. *IEEE Multimedia,* (April - June), 82-87.

Deshpande Y., Olsina, L., & Murugesan, S. (2002). Web engineering. Report on the *Third ICSE Workshop on Web Engineering, ICSE2002,* Orlando, FL, USA.

Ginige, A., & Murugesan, S. (2001a). Web engineering: An introduction. *IEEE Multimedia, 8*(1), 14-18.

Ginige, A. & Murugesan, S. (2001b). The essence of Web engineering: Managing the diversity and complexity of Web application development. *IEEE Multimedia, 8*(2), 22-25.

Ginige, A., & Murugesan, S. (2001c). Web engineering: A methodology for developing scalable, maintainable Web applications. *Cutter IT Journal, 14*(7), 24-35.

Glass, R. (2001). Who's right in the Web development debate? *Cutter IT Journal, 14*(7), 6-10.

Hansen, S. (2002). Web information systems: The changing landscape of management models and Web applications. *Proceedings of the 14th international conference on software engineering and knowledge engineering* (pp. 747-753). ACM.

Hansen, S., Deshpande, Y. & Murugesan S. (2001). A skills hierarchy for Web-based systems development. In S. Murugesan & Y. Deshpande (Eds.), *Web Engineering – Managing Diversity and Complexity of Web Application Development* (LNCS Vol 2016, pp. 223-235). Berlin: Springer.

Hieatt, E., & Mee, R. (2002). Going faster: Testing the Web application. *IEEE Software*, (March - April), 60-65.

Ivory, M.Y, & Hearst, M.A. (2002). Improving Web site design. *IEEE Internet Computing*, (March - April), 56-63.

Lam, W. (2001). Testing e-commerce systems: A practical guide. *IT Professional, 3*(2), 19-27.

Lowe, D. (2003). Web system requirements: An overview. *Requirements Engineering, 8*, 102-113.

Menasce, D.A, & Almeida, V.A.F. (2002). *Capacity planning for Web services: Metrics, models, and methods*. Upper Saddle River, NJ: Prentice Hall.

Merialdo, P. et al. (2003). Design and development of data-intensive Web sites: The Araneus Atzeni. *ACM Transactions on Internet Technology, 3*(1), 49-92.

Murugesan, S. (1998). *Web engineering*. Presentation at the *First Workshop on Web Engineering, World Wide Web Conference (WWW7),* Brisbane, Australia.

Murugesan, S. et al. (1999). Web engineering: A New Discipline for Development of Web-based systems. In *Proceedings of the First ICSE Workshop on Web Engineering*, Los Angeles (pp. 1-9).

Murugesan, S., & Deshpande, Y. (Eds) (2001). *Web engineering: Managing diversity and complexity of Web application development*. Lecture Notes in Computer Science – Hot Topics, 2016. Berlin: Springer Verlag.

Offutt, J. (2002). Quality attributes of Web software applications. *IEEE Software, Special Issue on Software Engineering of Internet Software, 19*(2), 25-32.

Oppenheimer, D., & Patterson, D.A. (2002). Architecture and dependability of large-scale Internet services. *IEEE Internet Computing*, September-October, 41-49.

Pressman, R.S. (2001). What a tangled Web we weave. *IEEE Software, 18*(1), 18-21.

Pressman, R.S. (2004). Applying Web Engineering, Part 3. *Software Engineering: A Practitioner's Perspective* (6th ed.). New York: McGraw-Hill.

Reifer, D.J. (2000). Web development: Estimating quick-to-market software. *IEEE Software, 17*(6), 57-64.

Siegel, D.A. (2003). The business case for user-centered design: Increasing your power of persuasion. *Interactions, 10*(3).

Uden, L. (2002). Design process for Web applications. *IEEE Multimedia,* (Oct-Dec), 47-55.

Williams, J. (2000). Correctly assessing the "ilities" requires more than marketing hype. *IT Professional, 2*(6), 65-67.

Williams, J. (2001). Avoiding CNN moment. *IT Professional, 3*(2), 68-70.

Bibliography on Web Engineering

For further information on many different aspects of Web development and Web Engineering, we have listed below some useful resources such as books, special issues, journal articles, and Web sites.

Books

Burdman, J. (1999). *Collaborative Web development: Strategies and best practices for Web teams.* Addison-Wesley.

Dart, S. (2001), *Configuration management: A missing link in Web engineering.* Norwood, MA: Arttech House.

Dustin, E., Rashka, J., & McDiarmid, D. (2001). *Quality Web systems: Performance, security, and usability.* Reading, MA: Addison-Wesley.

Friedlein, A. (2000). *Web project management: Delivering successful commercial Web sites.* Morgan Kaufmann.

Friedlein, A. (2003). *Maintaining and evolving successful commercial Web sites.* Morgan Kaufmann.

Gerrad, P. & Thompson, N. (2002). *Risk-based e-business testing.* Artech Publishers.

Hackos, J.T. (2002). *Content management for dynamic Web delivery.* John Wiley & Sons.

Lowe, D. & Hall, W. (1999). *Hypermedia and the Web: An engineering approach.* New York: John Wiley & Sons.

Menasce, D.A. & Almeida, V.A.F. (2002). *Capacity planning for Web services: Metrics, models, and methods.* Upper Saddle River, NJ: Prentice Hall.

Nakano, R. (2002). *Web content management: A collaborative approach*. Boston: Addison Wesley.

Nguyen, H. Q. (2001). *Testing applications on the Web: Test planning for Internet-based systems.* John Wiley.

Nielsen, J. (1999). *Designing Web usability: The practice of simplicity.* Indianapolis, IN: New Riders Publishing.

Powell, T.A. (1998). *Web site engineering: Beyond Web page design.* Upper Saddle River, NJ: Prentice Hall.

Powell, T.A. (2000). *Web design: The complete guide.* New York: McGraw-Hill.

Pressman, R.S. (2004). Applying Web engineering. In *Software engineering: A practitioner's perspective*. New York: McGraw-Hill.

Rosenfeld, L. & Morville, P. (2002). *Information architecture for the World Wide Web: Designing large-scale Web sites.* O'Reilly & Associates.

Scharl, A. (2000). *Evolutionary Web Development*. Springer.

Shklar, L. & Rosen, R. (2003). *Web application architecture: Principles, protocols and practices.* John Wiley & Sons.

Stottlemyer, D. (2001). *Automated Web testing toolkit: Expert methods for testing and managing Web applications.* John Wiley.

Vidgen, R. et al (2002). *Developing Web information systems: From strategy to implementation.* Butterworth Heinemann.

Wodtke, C. (2002). *Information architecture: Blueprints for the Web*. New Riders.

Journals

IEEE Internet Computing. www.computer.org/internet

IEEE Software. www.computer.org/software

Journal of Web Engineering, Rinton Press. www.rintonpress.com/journals/jwe

Journal of Web Engineering and Technology. www.inderscience.com

Web Information Systems Engineering. http://www.i-wise.org

World Wide Web, Kluwer Academic Publishers. http://www.kluweronline.com/issn/1386-145X

Special Issues

Engineering Internet Software, *IEEE Software*, March-April 2002.

Testing E-business Applications, *Cutter IT Journal,* September 2001.

Usability and the Web, *IEEE Internet Computing*, March-April 2002.

Usability Engineering, *IEEE Software*, January-February 2001.

Web Engineering, *Cutter IT Journal*, 14(7), July 2001.

Web Engineering, *IEEE MultiMedia*, Jan.–Mar. 2001 (Part 1) and April–June 2001 (Part 2).

Journal Articles

Almedia, V.A.F., & Menasce, D.A. (2002). Capacity planning for Web services: An essential tool for managing Web services. *ITPro*, July-August 2002, 33-38.

Arlitt, M., et al. (2001). Characterizing the scalability of a large Web-based shopping system. *ACM Transactions on Internet Technology, 1*(1), 44-69.

Barnes, S. & Vidgen, R. (2002). An integrative approach to the assessment of e-commerce quality. *Journal of Electronic Commerce Research, 3*(3). http://www.webqual.co.uk/papers/jecr_published.pdf

Baskerville. et al. (2003). Is Internet-speed software development different? *IEEE Software*, Nov-Dec, 70-77.

Becker, S. & Mottay, F. (2001). A global perspective of Web usability for online business applications. *IEEE Software, 18*(1), 54-61.

Brewer, E.A. (2002). Lessons from giant-scale services. *IEEE Internet Computing,* July, 46-55.

Cardellini, V. et al. (1999). Dynamic balancing on Web server systems. *IEEE Internet Computing,* May-June, 2839.

Ceri, S., Fraternali, P., & Bongio, A. (2000, May). Web modelling language (WebML): A modelling language for designing Web sites. *Proceedings of the World Wide Web WWW9 Conference,* Amsterdam.

Cloyd, M.H. (2001). Designing user-centered Web applications in Web time. *IEEE Software, 18*(1), 62-69.

Collins, R.W. (2002). Software localization for Internet software: Issues and methods. *IEEE Software*.

Davison, B.D. (2002). A Web catching primer. *IEEE Internet Computing.*

Deshpande et al. (2002). Web engineering. *Journal of Web Engineering, 1*(1), 3-17.

Deshpande, Y. et al. (2002). Consolidating Web engineering as a discipline. *SEA Software.*

Deshpande, Y. et al. (2002, July). Web site auditing – The first step towards reengineering. *Proc 14th International Conference on Software Engineering and Knowledge Engineering,* Italy, 2002, pp. 731 – 737.

Deshpande, Y. & Hansen, S. (2002). Web Engineering: Creating a discipline among disciplines. *IEEE Multimedia,* 82-87.

Fewster, R. & Mendes, E. (2001, April 4-6). Measurement, prediction and risk analysis for Web applications. *IEEE Seventh International Software Metrics Symposium* London, England, pp. 338-348.

Ginige, A. & Murugesan, S. (2001) Web engineering: An introduction. *IEEE Multimedia*, *8*(1), 14-18.

Ginige, A. & Murugesan, S. (2001). Web engineering: A methodology for developing scalable, maintainable Web applications. *Cutter IT Journal, 14*(7) 24–35.

Ginige, A. & Murugesan, S. (2001). The essence of Web engineering: Managing the diversity and complexity of Web application development. *IEEE Multimedia, 8*(2), 22-25.

Glass, R. Who's right in the Web development debate? *Cutter IT Journal, 14*(7), 6-10.

Goeschka, K.M. & Schranz, M.W. (2001). Client and legacy integration in object-oriented Web engineering. *IEEE Multimedia, Special issues on Web Engineering, 8*(1), 32-41.

Hieatt, E. & Mee, R. (2002). Going faster: Testing the Web application. *IEEE Software*, 60-65.

Ingham, D.B., Shrivastava, S.K., & Panzieri, F. (2000). Constructing dependable Web services. *IEEE Internet Computing*, 25-33.

Isakowitz, T., Stohr, E. & Balasubmmnian, P. (1995). RMM: A methodology for structured hypermedia design. *Comm A CM, 38*(8), 35-44.

Ivory, M.Y & Hearst, M.A. (2002). Improving Web site design. *IEEE Internet Computing*, 56-63.

Kirda, E., Jazayeri, M., Kerer, C. & Schranz, M. (2001). Experiences in engineering flexible Web services. *IEEE Multimedia, Special issues on Web Engineering, 8*(1), 58-65.

Lam, W. (2001). Testing e-commerce systems: A practical guide. *IT Professional, 3*(2), 19-27.

Liu, S., et al. (2001). A practical approach to enterprise IT security. *IT Professional, 3*(5) 35-42.

Lowe, D. (2003). Web system requirements: An overview. *Requirements Engineering, 8,* 102-113.

Lowe, D. & Henderson-Sellers, B. (2001). OPEN to change. *Cutter IT Journal, 14*(7), 11-17.

Maurer, F. & Martel, S. (2002). Extreme programming: Rapid development for Web-based applications. *IEEE Internet Computing*, 86-90.

Menasce, D.A. (1993). Load testing of Web sites. *IEEE Internet Computing,* 89-92.

Merialdo. P. et al. (2003). Design and development of data-intensive Web sites: The Araneus Atzeni. *ACM Transactions on Internet Technology, 3*(1), 49-92.

Mich, L. et al. (2003). Evaluating and designing Web site quality. *IEEE Multimedia*, 34-43.

Offutt, J. (2002). Quality attributes of Web software applications. *IEEE Software, Special Issue on Software Engineering of Internet Software, 19*(2), 25-32.

Olsina, L., Lafuente, G. & Rossi, G. (2001). Specifying quality characteristics and attributes for Websites. In S. Murugesan & Y. Deshpande (Eds), *Web engineering – managing diversity and complexity of Web application development* (pp. 266-278). Berlin: Springer.

Oppenheimer, D., & Patterson, D.A. (2002). Architecture and dependability of large-scale Internet services. *IEEE Internet Computing*, 41-49.

Perlman, G. (2002). Achieving universal usability by designing for change. *IEEE Internet Computing,* 46-55.

Powel, T.A. (1998). *Web site engineering: Beyond Web page design.* Prentice Hall.

Pressman, R.S. (2001). What a tangled Web we weave. *IEEE Software, 18*(1), 18-21.

Pressman, R.S. (2001). Can Internet-based applications be engineered? *IEEE Software*, *15*(5), 104-110.

Reifer, D.J. (2000). Web development: Estimating quick-to-market software. *IEEE Software*.

Roe, V. & Gonik, S. (2002). Server-side design principles for scalable Internet systems. *IEEE Software*, 34-41.

Scalable Internet Services (2001). *Internet Computing*.

Schwabe, D. & Rossi, G. (1998). An object oriented approach to Web-based application design. *Theory and Practice of Object Systems (TAPOS), special issue on the Internet, 4*(4), 207-225.

Schwabe, D., Esmemldo, L., Rossi, G. & Lyardet, F. (2001). Engineering Web application for reuse. *IEEE Multimedia, 8*(1), 20-31.

Scott, D., & Sharp, R. (2002). Developing secure Web applications, *IEEE Internet Computing*, 38-45.

Siegel, D.A. (2003). The business case for user-centred design: Increasing your power of persuasion. *Interactions, 10*(3).

Upchurch, L. et al. (2001). Using card sorts to elicit Web page quality attributes. *IEEE Software*.

Williams, J. (2000). Correctly assessing the "ilities" requires more than marketing hype. *IT Professional, 2*(6), 65-67.

Web sites

ACM SIGWEB: www.acm.org/sigweb

Jakob Nielsen's Website: www.useit.com

NIST Web Usability: zing.ncsl.nist.gov/WebTools/index.html

Universal Usability Guide: www.universalusability.org

Usability Professional Association: www.upassoc.org

Usable Web: www.usableweb.com

Web Engineering Resources, R.S. Pressman and Associates: www.ispa.com/spi/index.html#webe

Web Engineering.org Community Homepage: www.webengineering.org

Web Information System Development Methodology: www.wisdm.net

Web Information Systems Engineering: http://www.i-wise.org

Web Quality: www.webqual.co.uk

World Wide Web Consortium: www.w3.org

Conferences

International Conference on Web Engineering (ICWE) 2004 and 2005. www.icwe2004.org; www.icwe2005.org

Web Information Systems Engineering Conference. http://www.i-wise.org/

World Wide Web Conference. www.www2004.org; www.www2005.org