Web Engineering

Article

1 author:

Sven Ziemer
Bauhaus Luftfahrt e.V.
23 PUBLICATIONS **92** CITATIONS

SEE PROFILE

Web Engineering

Sven Ziemer
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, sven.ziemer@idi.ntnu.no

Abstract

WebEngineering is a relatively new discipline in the field of Software Engineering. This paper describes the special characteristics of Web Application and on the different Web Engineering practices used to develop Web Applications. Based on the work in the WebSys project some suggestions about how to improve the Web Engineering practices are given.

1. Introduction

The growth of the Internet and the world wide web (WWW) has resulted in a large number of Web Systems and Web Applications. Today many people are relying on Web applications like Internetbanking, entertainment and E-commerce applications. They expect these applications to be available, safe, secure and reliable.

The development of Web Applications and the practices used to develop these applications are different from traditional software development (see e.g. [15], [12] and [4]). The development of Web Applications has not been without problems, as the phrase "Web Crisis" [12] shows. A survey on Web Application development by the Cutter Consortium [3], highlights serious problems plaguing large Web-based projects:

- Delivered systems did not meet business needs 84 % of the time.
- Schedule delays plagued the projects 79 % of the time.
- Projects exceeded the budget 63 % of the time
- Delivered system did not have required functionality 53 % of the time
- Deliverables were of poor quality 52 % of the time.

It is therefore an interest in how to develop Web Applications, and to understand where Web Application Development differs from traditional software development. Today Web Engineering has emerged as a discipline in its own, right as the field that studies at the "application of systematic, disciplined and quantifiable approaches to development, operation, and maintenance of Web-based application" [4].

The purpose of this paper is to give an understanding of the special characteristics of Web Applications and of the Web Engineering practices used to develop these applications. Based on this understanding the paper will discuss some approaches from the WebSys project to support the development of reliable Web Applications within the given Time-to-Market requirement. We review textbooks, journals and conference proceedings published about Web development. In section 2 special characteristics for Web Applications and different approaches to categorize them are presented. Section 3 is looking at the development practices for Web Applications and section 4 is presenting some of the approaches that will be further investigated by the WebSys project. Conlusions are given in section 5.

2 Web Applications

Web Applications are software applications deployed by the World Wide Web. They use a single client-server model, and run in a Web browser on the client computer. Once a new release of a Web Application is installed on the server, this release is available to all users. This immediate deployment characteristic is probably one of the most powerfull characteristics of a Web Application.

There are different names in use for what here is called a Web Applications. Names in use are Web Sites, Web-based applications and Web Applications [4]. Some authors are also using different names to indicate different types of Web Applications [1]. In this article the term Web Application is used to represent all types.

A definition of the term Web Application is given in [9]:

A Web Application is a software application

that depends on the specifications of the World Wide Web Consortium (W3C) and that offers web-specific resources like contents and services that can be used by a Web browser.

2.1 Categorization of Web Applications

There are several ways to categorize Web Applications. The first categorization is found in [1].Web Applications are divided into two groups: Web Application that have state, and that use some server-side logic and Web Sites that only have client-side logic. According to the terminology used in this paper, both are Web Applications. This two categories are rather broad and do not describe different feature of the large number of different types Web Applications with much precision. What are different types of Web Applications in other categorizations, will be the same type of Web Application in this categorization.

Another categorization is given in [8]. Here Web Applications are divided along two dimensions:

- The amount of control logic, and
- the amount of data prosessed.

The author of this categorization uses four different categories, but the number of categories could easily be extended.

- Brochure – No control logic, and no data prosessed. An example for this category is a simple homepage.
- Service oriented applications – Some control logic, and a small amount of data prosessed. These sites are dedicated to provide services to its users, like email on the web (e.g. Hotmail).
- Data intensive applications – Web Applications that provide an interface to browse and query large quantities of data. An example is google.com.
- Information system applications – A mix of Service oriented applications and Data intensive applications (e.g. Amazon.com).

The last categorization used for Web Applications is given in [4] and with some changes also in [7]. The order of these categories roughly illustrates the evolution of Web applications and thereby also the growing complexity of Web Applications. These categories are shown in table 1. This list of categories will be extended when new types of Web Applications are invented.

Depending on the category of a Web Application, different resources, techniques and methods are required to develop them. Also, the purpose of Web Applications can change with the category ([15] and [4]). With this in mind the categorization given in table 1 is probably the most helpful. Implicitly it uses the same two dimensions as mentioned in the approach from [8]. The complexity of each new category is growing. This approach can be extended, and it is possible to link resources such as methods, processes, and best practices to each category.

2.2 Evolution of Web Applications

Web Applications are dynamic applications, they are changing and evolving constantly. This is a result of the Word Wide Web environment and of the market environment that the applictions are a part of. As Web Applications grow, they can also change the category they are belonging to. The evolution of Web Applications is not only an issue of the categorizations, but also an issue about the perception of the Web Development, about the motivation for Web Development and about the general change in the use of Web technology.

- **Perception** There are several perceptions of Web Development (see figure 2). These levels show the maturity of an organisation with Web Development. Each level also requires some special skills to succeed on this level. The first three levels require skills in Human-Computer Interaction (HCI) (level 1 and 2) and Information Architecture (level 3). Levels 4 to 6 deals with processes of interest to software engineering ([4] and [6]).
- **Motivation** The motivation depends upon the initial purpose of using Web Applications (Web 'presence' or becoming a Web-based organisation), the customers' expectations and the competitive environment of the Word Wide Web ('keeping up with competitor A') [4]. The drive to systemise development is subject to overall perception of the Web. A company who enters Web Engineering at stage 3, will view the development effort mainly in terms of "publishing" or "brand building/reinforcement". Lessons learnt from software engineeering are regarded as irrelevant og simply ignored. A low level perception of the Web is likely to lead to an ad hoc, sporadic development effort.
- **Change in the use of the WWW** Currently we are using the Web quite differently than the purpose for which it was originally conceived – sharing scientific information among a few scientists. Not only does the scope and complexity of current Web Application vary

Table 1: Categories of Web Applications

Category	**Examples**
Informational	Online newspapers, product catalgues, newsletters, service manuals, classifieds, e-books
Interactive	Registration forms, customized information presentations, games
Transaction	E-shopping, ordering goods and services, banking
Workflow	Planning and scheduling systems, inventory management, status monitoring
Collaborative work environments	Distributed authoring systems, collaborativ design tools
Online communities, marketplaces	Chat groups, recommender systems, marketplaces, auctions
Web Portals	Electronic chopping malls, intermediaries
Web Services	Enterprise applications, information and business intermediaries

Table 2: Levels of perception in Web Development

6. Web project planning and management
5. Web-based System
4. Web Site Construction
3. Web Site Design
2. Web Page Design
1. Web Page Construction

widely (from small-scale, short-lived services to large-scale enterpirse applications distributed across the Internet and corporate intranets and extranets), but in addition there has been a substantial evolution in the complexity of Web Applications from early and simple Web-Based Systems toward todays Advanced Web-Based Systems. Some characteristics of simple and advanced Web-Based Systems are shown in table 3.

As Web applications have evolved, the demands placed on Web-based systems and the complexity of designing, developing, maintaining, and managing these systems also have increased significantly.

3 Web Development

This section looks at the differences between Web Development and the development of traditional software. First, we give a presentation of what most authors considers the most typicall differences (section 3.1). Section 3.2 will look at the different development practices resulting from this differences, and finally section 4 will describe some approaches from the WebSys project to improve Web Engineering practices.

3.1 Differences in Web Engineering

An overview over differences in Web Engineering compaired to traditional software engineering is given in table 4. Some imortant issues are:

- **Rush to marked** Web Applications have short Time-to-Market (TTM) requirement (see e.g. [15] and [9]). The average project time for Web Projects is under 3 months. The reason for this is partly the competetive environment on the Web. “Presence is everything” [15]. This desperate rush to market is supported by the technology for developing Web applictions. This is further intensified by the fact that application deployment is matter of a few seconds and that the internet, the World Wide Web and the availability of Web browsers created an international market with a breadth and scale of nearly inestimable potential.
- **Evolutionary development** As a result of the desperate rush to marked, we see a sense of extreme urgency in Web Engineering: obvious opportunities that some one else will grab unless the market is captured first and held against all newcommers that follow. The flexibility offered by the Internet environment makes it possible to deliver the desired functionality in chunks

Table 3: Characteristics of simple and advanced Web-based systems

Simple Web-Based Systems	**Advanced Web-Based Systems**
Simple Web pages primarily presenting textual information	Complex Web pages
Information content doesn't change–fairly static	Information is dynamic–changes with time and users' need
Simple navigation	Difficult to mavigate and find information
Stand-alone systems	Integrated with database and other planning, scheduling, and tracking systems
High performance isn't a major requirement	Required high performance and continuous availability
Developed by a single indivitual or by a small team	Requires a large development team with expertise in diverse areas
Used for information dissemination in non-core application	Deployed in mission-critical applications

Table 4: Major Differences between Web Applications and Conventional Software

1. compressed development schedules	2. constant evolution with shortened revision cycles
3. "Content is king", i.e. it is integrated inextricably with procedural processing	4. Insufficient requirement specifications
5. small teams working to very short schedules	6. emerging technologies/methodologies
7. lack of accepted testing processes	8. user satisfaction and the threat from one's competition
9. minimal management support	10. criticality of performance
11. evolving standards to which Web applications should or must comply, depending on the specific circumstances (for example accessibility standards for government sites or IEEE or W3C standards for technological reasons).	12. understanding of additional disciplines required for Web Applications, such as hypertext, graphic design, information architecture.
13. security considerations	14. legal. social and ethical issues
15. Variety of backgrounds of developers	16. Rapidly evolving implementation environment, encompassing various hardware platform

of successive releases. Given that the market is willing to accept features in "installments", the cycle time for Web Applications is further compressed into short release cycles. This way of handeling Web Development is called Web Gardening by some authors [12].

- **Requirement Management** This issue is related to the previous one. The requirements for Web Applications are elicitated during the development of the application. This is partly due to the changing market environment, which the customers have to respond to. The technology that is used can also change [4]. The results is that there is no formal requirement phase in many web development projects. Instead, prototypes are used to get feedback from both the customer and the user. Prototypes of Web Applications are also published to collect user feedback [5]. As a result thus we need developers with an understanding of the applications problem domain. This understanding is needed when there are no formal requirement specification.

- **Different kind of market environment** The new market environment offers the software products a new degree of flexibility in terms of requirements and quality factors. Web Applications are in most cases not mission-critical or life-critical. Especially in the early stages of a products introduction, organizations are willing to sacrifice some aspect of product quality for shorter Time-to-Market. Requirements are negotiable from release to release in a market defined process.

- **Multi disciplinarity** The development of Web Applications requires project teams with several skills. Activities involved in building Web Applications include content managment, graphic design and Human-Computer Interaction. Also legal and ethical issues have to be considered. In addition comes the technical activites we also find in traditional software engineering [4]. This multidisciplinarity results in some challenges for the development team. An important issue is the communication between team members with such a diversity of backgrounds, especially in the light of the rush to market and the short development cycles [13].

- **The nature of End-users** Web Applications may address users anywhere in the world. They are in general not confined to specific user groups within an organisation. When Web Applications go beyond intranets, strategies and policies must be developed to better understand the potentially unknown, and perhaps unknowable, users in order to establish the quality parameters of the applications so that they can deliver quality systems, test sites and applications and maintain security [4].

3.2 Web Development Practices

The differences listed in the previous section, result in some development practices that are special for Web Engineering. There are conducted two surveys about Web Development practices ([15], [10] and [11]).

- **Parallel development** To meet the short Time-to-market requirement, developers have to try to shorten the time needed to develop a new release. This can only be done by parallel development, meaning that the development team is working on two different releases simultaneously. Releases may be developed in parallel, or staged onto the market such that design, development, and quality assurance are all taking place simultaneously, but sequentially on different releases. This poses interesting challenges in staffing the project phases. Some development personnel will work on one specific release from start to finnish, others participate in one or more phases across several releases. In the extreme, even several phases of the development occur in parallel. Sometimes, coding begins even before the requirements have been fully understood [15]. The Scrum development process is developed for this purpose [16].

- **Release orientation** In the early phases of a Web Application, releases are made in short cycles. The release cycle can be between 2 and 15 days [15]. A new release is defined by its deadline, and not by its content. It is more important to keep the deadline than to fulfill the expected requirements for a new release. Functionality that misses that deadline is easily transferred to the next release. The same is true for quality attributes. A future release can improve on the poor quality.

- **Tool dependence** Many Web Application development organizations make heavy use of development tools to speed up the design and coding process.

- **Customer involvement** Since the requirements are evolving during the development, customers are involved intimately in the development effort. Customers are often co-located with the develoment team, and participate closely in all phases of development [15]. The customer involvement can give them immediate feedback on costs and schedule implications of their requirements. Prioritization of features can be based on the customer's demand. There are some problems associated with this kind of customer involvement, as the customer often have a short time focus, and there is no focus on the long term business architecture.

- **Prototyping** Prototypes are used to deal with the unstable and evolving requirements. They are used to agree on requirements, and to receive feedback both from the customer and from the end users. This is possible since the prototypes are published. Prototyping is used as a way to communicate with the customers to validate and refine requirements [15].

- **Tailored methodology** The processes and methods used in Web Engineering are often taken from traditional software engineering and tailored for the special need of Web Engineering [5]. Some organisations have also developed an overall framework within which individual projects are allowed to tailor their own methodologies [15].

- **Web Crisis** Development approaches used for Web development have been ad hoc, with no or litte attention paid to development methodologies, measurement and evaluation techninges, applications qulity and project management. The result are Web Applications with a high probability of failure. This situation is described by the term Web Crisis [12].

- **Document orientation** Many developers, clients and managers still consider Web development primarily as an authoring activity rahter than an application development activity, to which some of the well-known software engineering and amangement principles and practices coult apply.

- **Evolving development processes** The development process used to develop a Web applications is evolving in the same way that a Web Application evolves itself. A release in an early stage of a Web application is developed using a different process than a release at a later stage. The reason for this is that early releases have to attract and catch potential users, whereas later releases have to defend this user base. In the early phases the requirements are not well understod and it can be accepted to not addres quality in a formal way. At the same time the rush to marked is strongest due to the competitive web environment. This makes it difficult to use a formal development process. At a later phase a Web Application will be more mature and quality issues become more important. The development process is evolving to address both the quality issues and the requirement changes in a more formal way.

4 How to improve Web Engineering Practices

The WebSys project [2] tries to find methods, guidelines, practices and processes to help balancing the conflicting requirements of short Time-to-market and reliability.

The desperate rush to market comes at the expence of software with reduced quality [15]. Some approaches which are investigated by the WebSys project are:

- The use of a standard architecture or a product line architecture for Web Applications should help starting new Web Applications projects with some proven components. This way, the quality of proven applications can be reused in new applications, and in the best case, TTM can be even shorter. However, these components have to fit into the rather ad hoc development practices used when developing new Web Applications.

- Another topic investigated by the WebSys project is the use of patterns for Web Development.

- The use of Model-Driven Development can also speed up the development time of Web Applications, while addressing quality issues. There exists some Web Application development environments who lets the developer specify Web Applications by using UML models [14].

- Another way to improve Web Engineering Practices would be to have a process modell for Web Engineering, where the evolving process can be managed in a controlled way. Such a process should also support the special requirements [5] of a Web Development process. The WebSys project is working with a Web Development process that uses several Trade-off techniques to speed up decision making and to enhance the communication among team members [17].

5. Summary and Conclusions

This paper has shown the special characteristics of Web Applications and Web Engineering Practices. Web Applications need to be developed in a disciplined way, and thus to apply traditional software engineering methods to Web Engineering. Still, the main source for methods in Web Engineering is Software Engineerig, but these methods must be adapted to the environment of Web Engineering projects.

A note on the special characteristics of Web Applications: not every Web Applications has these rather extrem charac-

teristics. There are also Web Applications which are developed the same way as traditional software applications. But the competitive environment of the Web, and the need to react to events in the market place can result in these rather extrem Web Development projects.

Ideas and approaches from the WebSys project [2] to improve on the Web Engineering practices has been presented. These approaches has to be tested in an commercial environment.

References

[1] J. Conallen. *Building Web Applications with UML, Second Edition*. Addison-Wesley, 2003.

[2] R. Conradi, T. Stålhane, T. Dybå, and D. Sjœberg. Websys: Web-based systems – time-to-market vs. reliability. Technical report, Norwegian University of Science and Technology (NTNU), 2000.

[3] C. Consortium. Research briefs, November 2000.

[4] Y. Deshpande, S. Murugesan, A. Ginige, S. Hansen, D. Schwabe, M. Gaedke, and B. White. Web engineering. *Journal of Web Engineering*, 1(1):3–17, October 2002.

[5] G. Engels, M. Lohmann, and A. Wanger. *Entwiklingsprozess von Web-Anwendungen*, chapter 10, pages 239–263. dpunkt, 2004.

[6] A. Ginige. Web engineering: Managing the complexity of web systems development. In *Proceeding*, pages 721–729. ACM, 2002.

[7] A. Ginige and S. Murugesan. Web engineering: An introduction. *IEEE Multimedia*, 8(1):14–18, 2001. Special issues on Web Engineering.

[8] A. Hassan. Architecture recovery of web applications. Master's thesis, University of Waterloo, 2001.

[9] G. Kappel, B. Pröll, S. Reich, and W. Retschitzegger, editors. *Web Engineering*, chapter Web Engineering - Die Disciplin zur systematischen Entwicklung von Web-Anwendungen. dpunkt.verlag, 2004.

[10] A. McDonald and R. Welland. A survey of web engineering in practice. Technical Report TR-2001-79, Departmend of Computing Science, University of Glasgow, 2001.

[11] A. McDonlad and R. Welland. Web engineering in practice. In *Proccedings of the fourth WWW10 Workshop on WebEngineering*, pages 21–30, 2001.

[12] S. Murugesan, Y. Deshpande, S. Hansen, and A. Ginige. A new discipline for web-based system development. In *Proceedings of the Workshop on Web Engineering, ICSE99*, 1999.

[13] K. Norton. Applying cross-functional evolutionary methodologies to web development. In S. Murugesan and Y. Deshpande, editors, *WebEngineering*, volume 2016 of *LCNS*, pages 48–57. Springer Verlag, 2001.

[14] J. Pleumann and S. Haustein. A model-driven runtime environment for web applications. In *UML 2003 – The Unified Modeling Language*, pages 190–204, 2003.

[15] B. Ramesh, J. Pries-Heje, and R. Baskerville. Internet software engineering: A different class of processes. *Annals of Software Engineering*, 14(1-4):169–195, 2002.

[16] L. Rising and N. Janoff. The scrum software development process for small teams. *IEEE Software*, 17(4):26–32, July-August 2000.

[17] T. Stålhane and S. Ziemer. A trade-off toolbox. Technical report, Department of Computer and Information Science, 2003.