Global Co-Operation in the New Millennium
The 9th European Conference on Information Systems
Bled, Slovenia, June 27-29, 2001

DESIGN OF WEB-BASED INFORMATION SYSTEMS - NEW CHALLENGES FOR SYSTEMS DEVELOPMENT?

Peter H. Carstensen, Lasse Vogelsang

The IT University of Copenhagen, Denmark, Glentevej 67, DK-2400 Copenhagen NV, Denmark
Tel.: +45 38 16 88 30, Fax: +45 38 16 88 99
carstensen@it-c.dk, vogelsang@it-c.dk

ABSTRACT

The web-technology is going through major changes these years, both with respect to types of systems based on web-technology, organization of the development work, required approaches and competencies, etc. We must rethink the organization of the development work. This requires a deeper and coherent understanding of the nature of web-development. This paper presents findings from a field study undertaken in a web-development company. From these findings we characterize web-application development and discuss some major challenges to cope with and to find proper support for in the future. We found that web-development is characterized by involvement of many expertise groups with little training and experience in information systems design, that some of the existing modeling and communication tools introduce severe problems, and that the pace of the introduction of new tools and features causes development and management problems. A further complication factor is that web-applications of today often are business critical systems.

1. INTRODUCTION

As a basis for designing complex information systems the Web-technology has matured a lot over the last few years. The technology is still fairly simple with a number of unsolved problems, but the advantages and potentials are so significant that most of today’s design of information systems to some extent is based upon web-technology. Organizations increase their investment in and usage of web-based technology. The scope of web-based application has grown enormously and has moved to become a platform that can support all facets of organizational work (Isakowitz et al., 1998). Furthermore, the web-technology differs from the traditional information technology in that "it might be labeled as a new type of information system, but [...] it is fundamentally a new medium of human communication" (Turoff and Hiltz, 1998, p. 116).

In parallel with this trend the activities to be IT supported become increasingly knowledge demanding, and the actors are confronted by increasing demands for improved quality products or services, improved complexity of the products and services, higher flexibility, shorter lead-times, etc. To cope with these demands work is often undertaken by large groups including people with different background and perspective. More actors become involved and an abundance of decisions have to be made by mutually interdependent actors. Actors involved in complex cooperative activities need support for communicating, coordinating their activities, keeping track of state of affairs in the field of work, sharing information, etc. To complicate it further an increasing part of the work that needs IT support is conducted in virtual organizations, meaning that the control structure and the spatial and functional arrangements differ from situation to situation (Mowshowitz, 1997). We need a re-orientation of the philosophy of management, IT usage, etc.

The trends mentioned above have serious implications for the development of information systems: New groups of expertise must be involved, the technology is rapidly changing and well established products and standards are rare (Burdman, 1999), the roles of and interplay between the developers and users are dramatically changing. These challenges become even greater due to the fact that most web-application development has been started as ad-hoc based 'quick and dirty' development of small sets of web-pages mainly used for 'toy purposes', information publishing or advertising. To phrase it differently: Many of the at last adapted traditions within traditional software development on careful analysis and modeling, punctual establishment of conform architectures, attempts to estimate, etc. are absent in most web-application development.

In order to come up with ideas, recommendations, tools, techniques, concepts and methodologies supporting the processes of developing web-based information systems we need a better understanding of what characterizes development of large scale web-applications today, who are involved, what expertise is needed, what are the essential problems, etc. This paper presents a first set of findings from an ongoing project that follows design, development and use of interactive web-applications (we stress the interactive aspect of the application to put explicit focus on web-applications that mediate interactions among multiple distributed collaborating actors, cf., DIWA, 2000). We present findings from a field study of web-design undertaken in a web-development company. From these findings we characterize web-application development and discuss some major challenges to cope with and to find proper support for in the future.

Much of the literature on web-design concerns the new potentials, possibilities and challenges for business and user organizations. Examples are articles on the potentials of the new technology in terms of 'perfect communication and information transmission' (e.g., Turoff and Hiltz, 1998), challenges when setting up the e-businesses or changes in the supply-chain relationships as a result of web-based e-business (e.g., Baron et al., 2000). Of course are these aspects essential when discussing web-based information systems. They are however considered out of scope.

The literature on design and development of web-based information systems seems to agree that development of web-based systems is different from development of 'traditional' IT-based systems. Kristin Braa et al. argue that the technology mainly is an interaction medium and that in a www-environment new applications will be developed and assembled by cloning existing components (Braa et al., 2000). Thus the notion of tinkering is more important in web-application development than 'rational' design decisions. From this they argue that the key-words of the information systems discipline will be "prototyping, object orientation, reuse and bricolage, 'quick and dirty ethnography', networking, redundancy, plug-ins, innovations, customer focus and time to market." (ibid., p. 27). Others argue that development of web-sites (a set of web-pages) differs a lot from development of web-based information systems. Tomás Isakowitz et al. state that the latter "supports work, and is usually tightly integrated with other non-WISs such as databases and transaction processing systems" (Isakowitz et al., 1998, p. 79). Web-based information systems are, however different from traditional information systems in the sense that they require new approaches and often are results of grass-root efforts. Some authors also stress the differences in terms of the speed of change in the technological basis. The pace at which the continuous evolvement of tools and features are running at the web-technology is extreme even compared to the rest of the IT-area, and that these tools "have lured people away from recognizing the need for a systematic design approach" (Balasubramanian and Bashian, 1998, p. 113).

The need for new competencies, new roles, new approaches and methodologies, new ways of organizing the development work, etc. has been widely recognized. The aim of this paper is to contribute to our understanding of the nature of development of web-based information systems. We do this by reporting from a 'real-life study'.

2. THE APPROACH TAKEN

In order to obtain a coherent understanding of complex engineering and development work settings and the work conducted, field studies are essential means (cf. e.g., Yin, 1989; Orlikowski, 1993). This paper is based on data collected in an empirical study of web-application development efforts. The study focused on roles, competencies, communication structures, use of tools and techniques, etc. in a large project designing and implementing a large web-site. Besides addressing activities in the one project in focus we also discussed more general themes related to the introduction of new tools and methodologies in the company. The work setting is described in further detail in the next section.

First phase of the field study and the data analysis was conducted over a period of four months and was mainly based on qualitative interviews (Patton, 1980), and observation and active participation in a number of project meetings. Two months after version 1 of the product was launched, we conducted a second series of interviews with the project participants. These were retrospective reflections on the experiences gained from the project work. Furthermore we dug into documents, models and specifications produced by the project. 10 semi-structured interviews have been conducted.

Our approach had a structured and targeted orientation. Although we did not start out with a strict set of hypotheses, we did bring an articulated perspective. Explicitly we addressed roles, competencies and the interaction between the involved designers. The data was analyzed and structured according to a number of topics that were identified from a first rough analysis. This resulted in a first very descriptive portraiture of the work. Then this was discussed and validated with some of the project participants, and it was presented to and discussed with a large group of researchers who have made similar studies in other web-development companies (see DIWA, 2000 for further details). In the light of the lessons learned from the first phase the data from both series of studies were analyzed. This has been done by going through the interview data, structure it according to the topics identified and a synthesis of the core problems or lessons that can be identified. Then these were written up in a second version of the working and discussed and validated with some of the project participants and the group of research colleagues.

The research approach taken can be characterized as qualitative research heavily inspired by theories and conceptualizations within Work Analysis (Schmidt and Carstensen, 1990) and from other comparable studies of software engineering work (e.g., Carstensen et al., 1995; Kraut and Streeter, 1995; Grinter, 1997).

The purpose of research must be to provide both the richness of detail and relevance of the problems studied, as well as a certain tightness of control or rigor. We do not believe that one empirical effort necessarily needs to encompass both aspects. Of course we do recognize that since the results reported in this paper are drawn from a single field study, we can only make weak claims as to the generality of the findings. We are, however, confident that our results have some validity since there seems to be a large overlap with results found in similar studies by our colleagues in the DIWA project. DIWA is an acronym for Design of Interactive Web-Applications (DIWA, 2000).

3. THE WORK SETTING

Our studies took place in an IT-development company (hereafter called the Company). Most of the activities in the Company have to do with development of software for a large Danish pharmaceutical company. The Company develops business systems, productions systems, web-based systems, and supports the IT-systems. The Company has developed these systems for the last 30 years. Of course development of web-based systems is of a more recent date. We will return to this issue and some of the problems related to this activity. Our studies have taken place in the web-development department (hereafter called WebSystems) in the Company.

Until recently WebSystems has been a minor department in the Company with 15-20 employees developing web-sites basically for information publishing for different departments in an pharmaceutical organization. The requirements for documentation have been minor and were not taken very seriously. This has changed within the last two years because the developed systems have become more critical to business and the size of the projects has grown significantly. Today WebSystems builds web-applications for e-commerce, document handling, community service, etc.

A number of new employees with various backgrounds have been hired to meet the new and increased requirements, which have emerged due to the new type of developed systems. It used to be mainly engineers that were hired in WebSystems. Today WebSystems hire people with a background in the arts, graphical design, etc. We will elaborate this issue in the following sections.

The size of WebSystems increases dramatically. When WebSystems was established five years ago only with a few people developing web-related systems. Today the number of employees is approximately 70. The number of employees has doubled within the last year and is planned to double again within the next year. This has significant impact on how WebSystems is organized.

Historically the demand for documentation of the software development has been high in the Company because the software is used in the pharmaceutical industry. Errors can cause serious problems for the customers' health and pharmaceutical company's business. Hence, the Company has had a formalized development methodology since 1994, where the Company became ISO9000 certified. The used methodology is based on techniques, tools and standard documents for different activities such as data modeling, and project planning, testing, etc., and includes guidelines for the use of the mentioned techniques, tools, and documents dependent on the project type. The methodology follows the waterfall model. According to key developers and managers in WebSystems this is problematic. WebSystems uses object-oriented technology to develop their systems, which is poorly supported by traditional structured analysis and a waterfall approach. Therefore is the existing methodology assessed as unsuitable for the web-development.

We have followed a specific project in WebSystems, the Zyme project developing a large web-site, including a web-portal with approximately 10 sub-sites, a content management system, and facilities for e-commerce and dedicated community support. The sub-sites address very different user groups, for example investors, the press, education institutions, customers (including possibilities of purchasing enzymes). Thus, the demands for usability and flexibility are high. WebSystems was responsible for the design and development of all aspects of the site, the information architecture, user interface and navigation, technical infrastructure, etc.

18 people were involved in the project, most of them working full-time. The project had a product manager and two project managers, one from Web-Systems and one from the customer. The rest were divided into four groups: Information architects, Designers, Developers, and Hardware Architects equally distributed. The groups have quite different educational backgrounds. The information architects typically had a background in the arts and some training in the basic web-technology (i.e., HTML). The designers' background was more diverse. They had a background in chemistry, graphical design, psychology, business and computer science. All the developers and hardware architects had a traditional technical background (programming, computer science or engineering).

The information architects' task was to determine the user groups for the portal and sub sites, categorize the intended and required information for each user group, and the functionality to navigate through the information. This did this in cooperation with some of the employees from the customer. The designers designed the graphical design. The developers and hardware architects implemented the technical part of the web-site, i.e., programming and configuration of the software and hardware of the system.

The project was divided into different phases. The first phase was inspired by the elaboration phase in the Rational Unified Process (cf., Jacobson et al., 1999) and included a series of meetings between a few people from WebSystems and representatives for the customer. The result was a number of general use case descriptions (Ibid.), a preliminary information architecture and tender material. The products the first phase was made by the product manager, an information architect and the project manager from WebSystems. In the next phase did the information architects establish a detailed information architecture, the designers made a 'look and feel' description of the site (defining pictures, fonts, etc.), and the developers prepared the technical development (they were basically trained in the tool the site was going to be implemented in). In the third phase the site was implemented and content was established and uploaded. Lastly the site was launched on time and meets the deadline.

4. OBSERVATIONS

This chapter briefly illustrates and comments on some of the observations made during our study.

4.1 Development of Web-Information Systems of Today

One of our basic assumptions was that web-applications are becoming increasingly business critical, and that this would be reflected in the organization of the development work. The Zyme project was an example of a web-application that was considered strategically important for the customer. It was a very large project and the customer required a thorough pre-analysis resulting in tender documents to ensure that different web-development companies could bet on the implementation. Furthermore, the deadline for version one was considered vital to the customer, and the site was considered the most important public relation activity. Thinking of IT as essential for public relation is new to most developers. The term 'branding' became important in the Zyme project. The designers and developers were informed that the sites should signal the attitude of the organization and high quality. These requirements must still be combined with the traditional requirements, such as informative, easy to use, quick to glance, etc. This was new to the actors, and obviously they had a very abstract and uncertain understanding of what it meant for their application. As one of the information architects phrased it:

> "One of the ideas is that the site should contain something with 'a kick' - you know - some energy! It is important for them [the customers] that this brand is pushed in the head of the user. One of our solutions will thus be that beside the main navigation there must be room for the system to present interesting stuff from one of the sub-sites on the portal entry and thereby push information into the face of the users" (our translation).

The changes in the importance of the applications have implications for the requirements to documentation and systems stability, security, etc. In the Zyme project much effort was spend on designing a proper and useful hardware and software platform to ensure high stability and good opportunities for expanding the system.

Another basic assumption for our study was that we expected to find indications that web-based applications become more 'interactive' in the sense that they support interaction among different users of the application (cf., DIWA, 2000). The Zyme site is an example of interaction in the sense of electronic commerce. But apart from that we found less usage of web-applications as means for interaction among collaborating actors than we expected. Most web-usage is still organized mainly in terms of publishing information.

The experienced problems in WebSystems are not only related to the used methodology (or lack of), to the introduction of new competencies, etc. Also WebSystems have to cope with a fast changing technology. In the Zyme project it was decided to use a completely new platform to support the development of the web-site. The hardware was new and the product to which the server-software should be developed was new. Two of the developers in the Zyme project had a one-week course in the product. The remaining developers and hardware architects had to setup and program the server-product during the project without training. To support the process consultants from the supplier were allocated to the project. It was, however soon realized that the consultant had very limited knowledge of the product and could spend very little time at WebSystems due to limited resources at the product provider. It was the first Danish implementation of the product, and it was used for implementation of a strategic application!

When it was decided to use the new product it was already known that a new and better version would be released before the programming was about to start. It was not clear for the project participants, which features the new release would provide, but they knew the web-site should be ported to the new version shortly after the first version was released. The designers had more or less to design without knowing the technical possibilities. According to the involved actors this was one of the biggest problems in the project.

The limited knowledge of the product caused frustration in all the involved groups and not only within the developers. The developers found that it would have been easier to implement the system in the products they usually used, and the information architects and designers encountered problems because they did not know the technical limitations of the new product. Consequently the information architects had to redesign part of the information architecture resulting in a new design of the user interface, which again involved the designers and a the creation of a new graphical design. This example illustrates the extraordinary pace at which new tools and techniques are invented creating an 'interoperability nightmare' for application developers and users and makes it difficult to manage the development process (cf., DIWA, 2000).

Another key aspect in the project was the customer's project manager. It was important for the customer to have a representative in the project, especially to ensure a good 'branding' the and get the right ‘look and feel’. To ensure this the customer's project manager was located together with the rest of the project members during the development. He collaborated with most of the project members in the different phases of the project. He was involved in making the information architecture, gave feedback to the designers, discussed functionality with the developers, and contributed to the preliminary and ‘wild’ ideas for the branding. Involvement in the ‘brainstorm phase’ made it possible for him to come up with new ideas and request changes in the original design described in the tender material and use cases.

Most of the project members from WebSystems stated that the influence of the customer's project manager was far too high and difficult to control. Often the project got a long series on comments to preliminary ideas that had not yet been carefully considered by the information architects and designers themselves, and many of the request established this way were either not in the tender materiel (i.e., not in the contract) or impossible to implement in the tools selected. One developer phrased the problem as follows:

> “... it is the first time that the customer is so close to a project. They are actually positioned right in the middle of the project and are able to take part in everything and — they do it!” (our translation)

And a designer stated the following about the customer’s project manager;

> “The customer has been allover the project: They participated in the analysis and they intervened the development of the information architecture, and more or less also in the design. Thus he has been playing back and forth with the information architects and the designers. The developers have been focusing on their own problems and have not all the time been involved in the design. So they have not been aware of some changes for instance if we have moved some graphics or something like that. The customer has been involved in this and has in practice been the first to see all the things we have suggested. This is a consequence of the structure we have in the project with a customer as project manager. It has made it hard to manage the customer’s expectations...” (our translation)

The fact that the customer has had so much influence on the design process made it difficult to manage requirements. Each sub-group in the project had more or less their own version of the requirements dependent on the feedback, they had got from the customer. Opposing requirements were often discovered and had to be resolved, sometimes by telling the customer that a specific requirement could not be implemented or by changing the original requirements.

The relation between the developing company and the customer is changing. In web-information systems the ‘look and feel’ is even more important. Frequent feedback from the customer has always been essential in information systems design. The concept of a web-site is relatively easy to understand and comment on, especially if we compare it to understanding techniques like data modeling. The nature of the systems is easier to understand for the customers and the technology itself invites to tinkering and quick changes. Requirements management becomes even more complicated.

4.2 New Competencies

The main purpose of the Zyme project was to 'brand' the enzyme company (i.e. the customer) through the web-site. In the Zyme project terminology branding meant that the web-site should promote the enzyme company and signalize the company's values and high quality products. To achieve this and make all the traditional hardware and programming successful too require people with very different competencies in a project.

As described there were very different groups of actors (graphical designers, information architects, programmers hardware architects, domain experts, etc.) in the Zyme project. They had to collaborate closely to develop the application. Compared to traditional software development the target group is divergent and the project team have a less homogeneous background. This fact caused problems with the organization of the project and communication between the different groups involved. We observed several indications of this during our study. For example, the information architects could not understand the use cases and they invented their own syntax for specifying the information architecture, which was difficult for the designers and the developers to grasp. When asked how they informed the other groups about their work all the actors indicated that this was done on oral basis only, and mainly at meetings. They had very little formalized notation or language that were used for sharing information and knowledge.

According to company managers, project managers, and different project members the general understanding in WebSystems is that web-development requires new and different competencies in contrast to the competencies represented in the Company in general. All the different types of actors working in WebSystems indicate this. The new people employed have primarily competencies in areas like visual design, communication and information handling.

It is, however not only a question of involving people with new areas of expertise. Also the ‘old developers’ have to be trained in new skills, tools, and techniques. All the developers and programmers we interviewed saw major differences in the kind of design and development work conducted in WebSystems compared to the rest of the Company where they develop more 'traditional information systems'. Developers moving from 'traditional systems development' have to learn new approaches. They must learn to work in development environments where the software to a large extent is hidden and spread in numerous small components, modules, scripts, tools, etc. The software does not have the same 'visible' overall structure and flow as they have been used to. As one of the developers said:

> “It is damned difficult to get a complete picture of the software when it is distributed in hundreds of small bits and pieces, and no tool provides you with an overview or access to the complete source code” (our translation).

An interesting observation was that a shift in attitude from some of the young and recently educated developers was required. Both the project manager and some of the more experienced developers mentioned that young developers have no training in and understanding of what happens when software development grows and becomes large-scale. The habits from design of small (toy) systems become problematic. One of the software engineers talked about the problems of ensuring consistency and develop a stabile architecture:

> "...we have to do things in a standardized manner, the same every time, and know what each other are doing, etc. etc. The web-world is characterized by young guys, you know, starting here just having finished their education, and wauuww, 'we just have to do this - damn fun yeahh!' Then it's hard because they ask 'why? Why should I do that?' They have never been in a situation where everything goes haywire." (our translation).

4.3 New Roles and Forms of Collaboration

We have encountered some of the challenges the different competencies create. The four groups of people in the Zyme project coordinated their work and collaborated with each other to develop the web-site. The information architects designed the functionality and information architecture and communicated it to the developers. The developers communicated the technical possibilities and limitations back to the information architects. Beside these four main groups of competencies a typical project in WebSystems included two project managers (one internal and one from the customer) and people from the quality assurance department.

Apart from the groups mentioned the Zyme project also had actors from the server supplier and a marketing and advertising company involved. Furthermore there was a close interaction with domain experts from the customer. All these people were organized in small special groups and worked in the same physical place as the people from WebSystems.

Interaction between the groups (especially between the technical and non-technical people) was a critical part of the Zyme project. According to our observations and comparisons with other similar studies this is a general problem in complex web-development. Most of the actors involved in the Zyme project had no or very little experience in collaborating with the actors from the other groups. The interaction was problematic in several ways. A designer stated:

> “..at the end [of the project] they [the developers] just stood and said ‘ohh.. that’s how it should be. I knew what it should look like but I did not know that this one should change’. In some way we needed some material we could use to hand over stuff with. I can’t explain how... we should have had some kind of document we could start our work from, where we could read all these facts about the design and information architecture, for instance structure, menus or something like that” (our translation).

It was problematic for the groups internally at WebSystems to communicate their results, findings and ideas. It was planned that the information architects should describe the overall functionality and navigation by means of use cases. They had, however no experience with using use cases and the result was that the use cases were specified by a project manager and a developer. The use cases were then discussed at meetings between the information architects and the developers. The agreement at these meetings was based upon the verbal presentation only since the information architects still were not able to understand the use case specifications. Despite this problem it was the information architects that specified the information architecture and established a matrix specifying which user groups should have access to which information structures. Both the specifications were made in self-designed structures unfamiliar to the developers.

The division of labor and the responsibilities for each group were difficult to define for the actors involved in the project. The most common comment to this was that 'this is the first time I'm involved in this and I think we must be better in handling this in our next projects.' Some of the people in the project considered it a more permanent problem, not necessarily related to the fact that things were new and more or less unknown to everybody. Normally, it would be expected to be the responsibility of the project manager to ensure clear interfaces between the different groups and activities. However, the Zyme project obviously had so many uncertainties that such a clear division of labor could not be established up front in the project.

The different groups had different ways of working. When the information architects needed an overview of the web-site they invented their own diagrams specifically for this. They also made simple paper-based mock-ups of the user interface. The designers used these mock-ups to get a rough idea of the user interface and designed user interface prototypes in visualization tools (e.g., PhotoShop). The developers used the prototypes, the information architecture diagrams, and the original use cases from the tender material as basis for the implementation of the web-site. There was often lack of consistency because the groups worked with different requirements and understandings of the system. As mentioned the main reason for this was problems with handing over information. A developer stated:

> “It is difficult to figure out how the hand over of information should happen. They had a fine hand over from IA [information architects] to design but it [the information] stopped there and has not proceeded to development. It is probably not enough that the developers talk with the designers, we probably also need to get in contact with IA to get an idea about the functionality besides what they should look like, what are IA’s thoughts, etc. So we need some common hand over. That would have been smart.” (our translation).

The members of the project lacked an understanding of each other's work and of the information the other groups needed to do their work. A common understanding is needed but can be difficult to achieve because of the different background, traditions, etc. Many of the project participants indicated, that a solution might be new development methodologies that addresses the different tasks in the development of web-based information systems. This still needs further investigation. None of the attempts with new methodologies and techniques that were taken in WebSystems appeared successful. All actors agreed that a better understanding between the expertise groups of the other groups' needs is required. Some of the project members stated this could be a matter of more experience, but this might not be sufficient.

5. DISCUSSION — CHALLENGES FOR WEB-BASED INFORMATION SYSTEMS DEVELOPMENT

One of the central characteristics we have observed in web-development is a shift in the basic characteristics of web-applications. Web-applications have been used for publishing information, with limited functionality, and have been peripheral to the core business. From our study and others we have compared to we can observe that *web-applications are becoming critical to the companies' business*. Thus, the demand for a higher quality of web-applications is increasing significantly. This is not reflected in the approaches taken, the methodologies applied, and the organization of the work used in much development of web-applications of today. Web-applications are to some extent still developed as if they were 'toy-systems' primarily used for publishing information. Web-applications will be used for more strategically purposes in the future and therefore be important for business.

The point made here can also be observed from the fact that *much web-application development is driven by technology push* in high speed innovative settings. In parallel with the shifts in importance of the web-applications developed, we expect to see a shift towards more technology pull approaches where needs, requirements, and high quality becomes essential. Hence, *a major challenge for development of web-application is a shift in attitude* and maturity of the work. The approaches taken and the practice applied must be changed. Careful planning of the process and investigations of needs and possibilities must be undertaken. The roles to be involved and their responsibilities and interaction must be outlined much better than today, and the importance of careful considerations of design and architectural choices needs to be understood by the involved actors, i.e., taught in classes and communicated by the key people in the actual development efforts.

We have observed and followed a number of attempts to introduce new methodologies and techniques for coping with some of the problems related to the challenges in web-applications development. The success of these initiatives were very limited. Some of the encountered problems were caused by introducing a methodology, which was perceived as designed primarily for product development. This did not conform to web-application development. A *better understanding of the specific requirements for methodologies* supporting development of web-based applications and information systems is required. The general knowledge of what characterizes web-development is too limited. From our observations we would argue that the methodologies should not be to complex or require a lot of skills in formal modeling and specification. However, the systems are becoming complex, and it is difficult to think of solutions not involving some structured formalization. Iterations and prototyping might not be useful either in order to cope with some of central characteristics of today's web-applications, e.g., systems with known, but highly complex information structure. A more elaborate discussion on relevant approaches along the dimensions of complexity vs. uncertainty and modes of operation vs. means of expression (cf., Mathiassen and Stage, 1992) is required.

In parallel to the challenges concerning methodology and approach is the problem of *how to organize web-application development work*, and how this should be reflected in the methodologies and approaches we choose. The aim is still to handle the process efficiently and effectively. We also have to understand which roles and competencies that are needed in development of web-applications. There is obviously a demand for many different competencies, but which are essential? At WebSystems many different types of expertise were involved. Most of the project and department managers we interviewed had severe doubts on which skills are essential to have in the projects.

An aspect of the organization of the development work is also the role of, and interaction with, representatives of the users or the customer. Since the web has been used mainly for publishing it has been fairly easy to have strong user-involvement in all phases of the projects. However, the nature of the products are changing, and that has to be reflected in order *to identify the most appropriate models for user-involvement*. The customers and users will probably play a much more active role by furnishing the design process with information to both the information content, the actual design, the architecture, etc. due to the fact that the customers have more knowledge and insights about how applications can be build by means of web-based technology. The relation between developers and customers changes (see also DIWA, 2000). This raises customers' expectations and increases the demands on the developers when negotiating with the customers and users. The trend will furthermore increase the needs for methods and tools supporting the communication between the developers and the users and customers.

As mentioned several times the need for different types of expertise will increase. This causes one of the central challenges to web-application and web-information systems development*: How can employees with heterogeneous backgrounds be qualified* to handle technical, management and communication problems? How do we provide these very different groups of actors with a *'common language'* that enables them to not only communicate, but actually to exchange central information about the needs, the design and the implementation of a complex web-based information system? In our studies we observed that the designers and developers had severe problems in understanding each other. They were not able to read each others diagrams and specifications, and at some occasions the responsibility for certain tasks were changed simply because some groups of actors did not master the tool or method intended. Until these problems have been coped with much web-application development will be very inefficient. Of course, better education and teaching might be part of the answer, but we believe *that new concepts and tools are required to really make progress* on this. This requires a deeper and more coherent understanding of the problem and a detailed analysis of requirements for such concepts and tools.

Although the different groups have very different background and traditions for organizing their work they have to collaborate on developing the systems, i.e., they are highly interdependent in their work and they have to coordinate their activities. This has further implications for the approaches and methodologies we provide. Methodologies and techniques have to find a balance, which can conform to both technical and non-technical people and can be understood and used by the different groups. The methodologies must *ensure means for coordinating activities among actors that might not really understand the core problems and challenges in each others activities*. Even more clear cut interfaces between the different activities than what is common today and more explicit support for establishing the required division of labor should be provided. This might be somewhat conflicting with the common approach in much web-development today having a high degree of trial-and-error iterations, very loose specifications, and little or no documentation. Again the right balance has to be found.

Development of web-applications will often be confronted with requirements combining 'traditional requirements for information systems' (e.g., easy to use, task relevant facilities, response time, etc.) and *requirements related to advertising and branding*. The latter is new to many developers within web-development. Design and development must be provided with tools that support a conceptual understanding of the two sets of requirements (that are often in opposition to each other) and how they are interrelated. Such tools should also support actual design that fulfills the requirements. Most of today’s traditional design tools are designed to optimize effectiveness, efficiency, usability, etc. They are not designed for development of applications that are intended to be used for branding and have the users to keep browsing at the web-site as long as possible.

Apart from the changes in the amount of user involvement in the projects our study has also indicated that the division of labor between developers and users shifts in other areas too. Other studies have also indicated that in most web-based information systems the information content and structure is developed and maintained by users, typically local super users or web-administrators (cf., e.g., Bansler et al., 1999). These users are often distributed in time and space. There are therefore no centralized overall instrument for maintaining the overall structure and coherency of the site. The approaches and methodologies for developing and maintaining large web-applications have to take this into account. *Support is needed for building a stable web-site infrastructure that is firm enough to cope with distributed and non-coordinated maintenance, and flexible enough to provide proper support for the individual needs.* Furthermore, the tendency to change the division of labor means that future designers of information systems have to accept that they are not 'in charge' of the information structure and content.

6. CONCLUSION

The development of information systems of all kinds has been heavily influenced by the introduction of the web-technology as a platform for many different types of systems. This implies major changes in the approaches, organization principles, methodologies, tools, etc. we use for systems (web-application) development. In this paper we have aimed at informing this work. We have presented findings from a field study conducted at a fairly large web-development department. Our findings have illustrated that: 1) the web-technology is used for many different kinds of applications, including business critical usage; 2) there are severe problems in just adapting 'traditional' software development approaches in web-application development; 3) many new competencies must be involved, and this involvement complicates collaboration and interaction between the involved actors; and 4) the tools and platforms are shifting extremely fast in the web-development world, causing problems both to development and management.

From this we have discussed a preliminary list of overall challenges for the web-development of today. These challenges include:

- New approaches methodologies supporting the dedicated needs;
- Changes in the organization of the development work, ensuring a better interaction among the different groups of expertise and interaction with customer representatives;
- Experimentation with new roles and expertise to be involved in development of web-based information systems;
- Changes in attitude to rigidness of the processes conducted during development;
- Improved concepts and tools for interaction, collaboration and coordination among very heterogeneous groups of designers and developers; And
- Improved concepts and tools for interaction between developers and users and customers.

The essential challenges for development of web-based information systems of today presented here are derived from a single field study only. Further investigations, field studies, experimentation with approaches, tools, etc. are, of course, required. We do, however believe that our observations are quite general in the sense that they fit with other observation we and others have conducted, cf., the discussion in the introduction and similar studies from the DIWA project (DIWA, 2000). The conclusion is then: A lot of rethinking, conceptualization and development is required in order to provide proper support for web-based information systems development in the future.

ACKNOWLEDGMENTS

This research could not have been conducted without the invaluable help from numerous people at WebSystems and the Company. The data collection and analysis work has been undertaken in close collaboration with Lars Kofod. The work has been funded by The Danish Research Councils.

REFERENCES

Balasubramanian, V., and Alf Bashian (1998). Document Management and Web Technologies: Alice Marries the Mad Hatter. Communications of the ACM, 41 (7), 107-115.

Bansler, Jørgen P., Erling Havn, Jacob Thommesen, Jan Damsgaard, and Rens Sheepers (1999). Corporate Intranet Implementation: Managing Emergent Technologies and Organisational Practices. in J. Pries-Heje at al. (eds.): 7th European Conference on Information Systems, Copenhagen, Copenhagen Business School, , vol. 3, pp. 750-757.

Baron, John P., Michael J. Shaw, and Andrew D. Bailey (2000). Web-based E-catalog Systems in B2B Procurement. Communications of the ACM, 43(5), 93-100.

Braa, Kristin, Carsten Sørensen, and Bo Dahlbom (2000). Changes - From big calculator to global network. in K. Braa, et al. (eds.): Planet Internet, Studentlitteratur, 13-39.

Burdman, Jessica (1999). Collaborative Web Development. Strategies and Best Practices for Web Teams, Addison-Wesley, Reading etc.

Carstensen, Peter H., Carsten Sørensen, and Tuomo Tuikka (1995): Let's talk about bugs!. Scandinavian Journal of Information Systems, 7 (1), 33-53.

DIWA (2000). http://www.diwa.dk.

Grinter, Rebecca E.: (1997). Doing software development: Occasions for automation and formalisation. in J. A. Hughes at al. (eds.): ECSCW '97. Proceedings of the Fifth European Conference on Computer-Supported Cooperative Work, 7-11 September 1997, Lancaster, U.K., Kluwer Academic Publishers, Dordrecht, pp. 173-188.

Isakowitz, Tomas, Michael Bieber, and Fabio Vitali (1998). Web Information Systems. Communications of the ACM, 41 (7), 78-80.

Jacobson, Ivar, Grady Booch, and James Rumbaugh (1999): The Unified Software Development Process, Addison-Wesley, Reading etc.

Kraut, Robert E., and Lynn A. Streeter (1995). Coordination in Software Development. Communications of the ACM, 38 (3), 69-81.

Mathiassen, Lars, and Jan Stage (1992). The principle of limited reduction in software design. Information Technology and People, 6 (2-3), 171-185.

Mowshowitz, Abbe (1997). Virtual Organizations. Communications of the ACM, 40 (9), 30-37.

Orlikowski, Wanda J. (1993). CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development. MIS Quaterly, no. September 1993, 309-340.

Patton, M. Q. (1980). Qualitative Evaluation Methods, Sage Publications, Beverly Hills, CA.

Schmidt, Kjeld, and Peter Carstensen (1990). Arbejdsanalyse. Teori og praksis [Work Analysis. Theory and Practice], Risø National Laboratory.

Turoff, Murray, and Star Roxanne Hiltz (1998). Superconnectivity. Communications of the ACM. 41 (7), 116.

Yin, Robert K. (1989). Case Study Research: Design and Methods, Sage Publications, Beverly Hills.