Abstract

Development of a Web site is not an event. It is a process. Once developed, information in the Web site needs to be maintained. Also the functionality and the amount of information will evolve with time.

Our experience has shown that information maintenance aspects of a Web site have to be incorporated into the design from the very beginning. Based on many years of research, we have now developed a systematic approach to developing large and maintainable Web Based Information Systems. We call this Web Engineering as this approach somewhat mirrors the evolution of Software Engineering.

This paper will describe the approach and illustrate the methodologies that were developed using a large Web site which we produced as a case study.

Introduction

Often in the last few years we have seen Web sites that initially started as a few Web pages, which grew in size and after a while became unmanageable. We have also seen managers wanting to develop Web sites for their organisations, getting frustrated because they could not get the Web site they wanted developed in a timely manner within a reasonable budget. Another common occurrence is that large Web sites become unmaintainable after some time. After a while in a large Web site if the information changes, making appropriate changes to all relevant Web pages becomes very difficult, if not impossible. We have experienced this especially with Web sites developed by Universities. Every six months there are changes in subjects, courses and course coordinators. From time to time staff might leave and new staff will join the University. In one instance the whole Web site had to be removed and redesigned.

These experiences made us explore a systematic way to develop large and maintainable Web based information systems. The first thing we realised was that the development of a Web site is not an event, but a process. We can not develop it and forget about it. We need to build into the design, ways of keeping the information up to date and once developed, ongoing updates become a continuous process.

Firstly, we need a process to follow when developing a large Web based information system. This process will consist of a set of manageable activities that we need to carry out to develop the Web site and then to keep its information current. Also, from time to time we need add new functionality and information resources.

The optimum order in which we need to carry out these activities will vary from Web site to Web site. In the simplest case, we may be able to carry out these activities one after the other. But in reality, we will need to revisit some of the activities previously done. A diagram that shows the path that we have to take through various activities is known as a process model.
A Web site consists of many components such as static HTML pages, dynamic HTML pages, CGI scripts, databases, search engines etc. Another important aspect is to select the appropriate components to create the Web site based on the information maintenance requirement and the complexity of the Web site. These components then need to be arranged into a logical architecture. We refer to this as the product model.

Most of the current difficulties, with respect to development of large Web sites, can be tracked down to a lack of a suitable process models which the development team can follow and the lack of a suitable architecture or product model for the application that needs to be developed. The choice of a product model and a process model depends on many factors. Some of these are, the scale of the application, the nature of the information, frequency at which information needs to be changed or added, experience of the development team and the clients of developing similar systems, computer literacy level of the users, available time to develop the application and the budget. Further, the process model depends on selected product model as well.

In a new and emerging discipline such as development of large Web based information systems, it is common to learn through experience. Development of large-scale software also went through a similar phase in early the 70s. Lack of suitable process models and application architectures gave rise to a software crisis [1]. Most large-scale software that was developed either did not meet the specification, did not work properly, was over budget and could not be delivered within the agreed time frame. To manage the complexity of these software systems people developed new process models, application architectures and development methodologies. This was the birth of Software Engineering, Requirements Engineering, Object Oriented development techniques etc. Now that we are in the process of developing a new discipline “Web Engineering” we have a lot to learn from Software Engineering experience.

There are many similarities between Software Engineering and Web Engineering; but we also have to acknowledge the differences. At present, there is an interesting debate “Can Internet-Based applications be Engineered?” is going on [2]. Users are an integral part of a Web based information system. Thus, when developing any Web based information system it is essential to have appropriate steps built into the development process model that will cater for user related issues. Also the information content and the functions of a Web site tend to evolve while being developed much more than a software application.

When we are dealing with complex systems or processes, it is essential that we break these into manageable modules and organise these modules in a logical manner. This is the guiding principle that we used when developing many large Web sites. More details about various methodologies that were developed can be found in part II of the book, “Hypermedia Engineering” [3].

**Overall Development Process**

When developing a large web site we need to follow a process. A process will enable us to divide a complex development task into manageable coherent phases. This will also enable us to allocate necessary resources to each phase and to measure the progress of the development of the application. Thus, one of the first things that we need to do is to develop a suitable process model for the project.

To develop a process model we need to identify various activities that need to be carried out and organise these into some
logical order. For example, it is not possible to have design before analysis. In order to decide what activities are required, we need to understand the project in a larger context. We call this context analysis. Figure 1 shows the overall development process that we came up with.

![Diagram of overall development process]

Figure 1: Overall development process

Activities such as project management, documentation and quality control and assurance need to take place throughout the project.

There are proven methodologies and techniques to perform some of the above mentioned activities. When analysing the initial failure of some of the projects that we undertook or was aware of, all these pointed to a lack of context analysis and failure to adopt a suitable “Product Model” and a “Process Model”. The rest of the paper will address these issues.

**Context Analysis**

First, we have to understand the overall environment within which the application will exist. This is called the problem domain. For example, if it is an application to deliver education, the problem domain will include issues such as how the education is delivered at present, what new knowledge about delivery of education is available, what are the specific needs to develop a new Web based application, who are the users and how will it be used.

Next, we need to look at possible solutions at a global level and narrow these down. For example, in the above case a decision to develop a Web based system to deliver education will limit the range of possible solutions to the web based systems. The available technology will also constrain the possible range of solutions. For example, due to bandwidth limitations one might decide that no video would be included in the application.

The third step, is to look at constraints that are specific to the development process. Some of these are available budget, available timeframe to develop the application, available expertise etc.

Once we understand the issues that impact the application and the application development process, the next step is to decide various activities that we need to carry out and organise these into a logical structure.

The process we are going to use also depends on the architecture or the information system (product model) that we are going to produce. The product model specifies the various components of the application and how these are connected together. Thus, the process model should have appropriate activities to produce these components.

**Product Models**

There are many ways to develop Web based information systems, ranging from using static HTML pages to dynamic Web pages. Once created, information in a static page remains static. Dynamic Web pages are created by the system using the latest information available when a user requests that page. Also, we can
have intermediate situations where Web pages are updated at regular intervals rather than every time the page is requested, to minimise the time taken to create and send the page. This approach is suitable if information is changing once a month or once every few months. Every time the information changes we can use a batch process to regenerate a new set of static Web pages.

To have a maintainable Web site, we need a way to manage the changes to information. This information can be stored in a database or set of marked up files using a markup language such as SGML (Standard General Markup Language). Depending on the frequency of change, we can either decide to regenerate Web pages from time to time by using some scripts or to have a product model where Web pages are created by the system on demand. Either way, the creation of Web pages needs to be automated.

Information that comes from a Web server to a Web browser will consist of:

1. system information;
2. presentation information;
3. index, menus and link information;
4. content information.

The system information is created by the server. Thus, we need a product model that can generate the other three types of information.

Thus, a typical product model for a maintainable Web site is shown in figure 2.

As part of the overall design and development process, we need to streamline the workflow associated with generating and changing information and link this to the content storage elements used for the Web site.

There are different technologies one can use for creating dynamic Web pages. A common approach is to create dynamic Web pages using a CGI Script as shown in figure 3.

Figure 2: Layers of a product model for a maintainable Web site

Figure 3: Basic physical architecture of a Web server supporting CGI Scripts
In a Web site, it is not only the information that keeps changing. From time to time we will need to enhance the functionality as well as new information sections or modules. Thus, we need a product model than can accommodate these changes. An example of a possible architecture is shown in figure 4.

Thus, a product model of a large complex Web site will consist of many horizontal and vertical layers as shown in figure 5.

<table>
<thead>
<tr>
<th>Presentation layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scripts for generation of indexes, links, menus</td>
</tr>
<tr>
<td>Content storage</td>
</tr>
<tr>
<td>Application 1</td>
</tr>
<tr>
<td>Content storage</td>
</tr>
<tr>
<td>Application 2</td>
</tr>
<tr>
<td>Content storage</td>
</tr>
<tr>
<td>Application 3</td>
</tr>
<tr>
<td>Information management with automated workflow</td>
</tr>
<tr>
<td>Information management with automated workflow</td>
</tr>
<tr>
<td>Information management with automated workflow</td>
</tr>
</tbody>
</table>

Figure 5: A possible logical architecture for a maintainable complex Web site

In a product model like this, it is very easy to change the look and feel of the whole site by making changes to the presentation layer without editing each and every Web page. This is a big advantage for large Web sites.

**Process Model**

A process model essentially considers the development phases or stages of development and how these are integrated into an overall development process. It does not describe the specific techniques used to carry out development activities (these are left to specific methodologies), but rather provides a framework in which we can understand and plan the development process. Indeed this framework is usually instantiated as a project plan early in the development.

We can design different process models to suit different types of development (and development issues), which will in turn be suited to different types of applications. For example, a model incorporating iterative refinement of an initial prototype may be best suited to small-scale trial applications. A competent Web Engineer would be aware of the available models, and would use the most appropriate model for each problem, adapting it in a way which takes into account the limitations and strengths of the model chosen, and the application being developed.

Various activities that we need to carry out when developing large Web sites are generally combined into and called sub-processes or phases. These phases combine to form the overall development process. There are several reasons for considering groups of activities together. First, is the simple issue of cognitive management of the process. By merging activities into phases it allows us to better conceptualise what needs to occur and the relationships between the various development activities.

A second, and in many cases much more important, reason for combining phases is as a project management aid. Effective management of the development process will require that development progress can be evaluated. This in turn requires both intermediate deliverables in the development and suitable development milestones. The completion of development phases (or the movement from one phase into another) act as these milestones - thereby providing valuable project management aids.

The grouping of activities into phases can be undertaken in various ways, often guided by conventional engineering principles. A common approach is to group activities in such a way as to minimise the coupling among sub-processes. One basic grouping is: planning and management, analysis, design, production, delivery, and maintenance.
The order, in which these development phases take place along a time line, the amount of effort required to carry out each activity and types of coupling between phases varies from application to application based on the nature of the application. We need to consider aspects such as the inter-relationships between phases and the extent to which we can iterate the development.

For example, when developing a small application consisting of only a few Web pages, it may be sufficient to follow a reasonably linear development sequence starting with analysis followed by design, production, and delivery (i.e. putting the Web pages into a Web server with appropriate access control) and maintenance. However if we are developing a large complex application in a relatively unknown application domain we may well discover, when in the production phase, that a change to the design can make the production process much more cost effective. In such a situation we would have to re-visit the design phase and check whether these changes could be accommodated. A process model that facilitates iteration would therefore be very beneficial.

In other words, different domains, application scales, and other factors will all influence not only the activities we are carrying out but also the way in which these activities are combined and undertaken.

A very basic process model is the Water Fall model as shown in figure 6. This model in its simplest form will not provide for any iterations. Yet it is a useful model to look at to identify various phases required in a process model.

Often when we are developing large and complex Web sites, we need to go through few iterations. The Spiral Development Process (figure 7) which was originally proposed by Barry Boehm [4] is suitable for developing large Web sites.

**Case Study**

One of the studies we did was to use this approach to develop a Web site for the Graduate School of Engineering at UTS. In this paper only the important steps are discussed. Full details can be found in [5].

According to the client (Head of the Graduate School) the purpose of this Web site was to inform the prospective graduate students both within Australia and oversees, students and staff members within the University and the general community about courses and programs offered by the Graduate School of Engineering.

**Context Analysis**

The problem domain analysis revealed that:

- main source of information is the Postgraduate Studies Handbook;
• information in this book is updated at regular intervals by the secretarial staff, but only printed once a year;
• some information in this book changes every six months if not more often;
• thus the hard copy, which is printed once a year, will be obsolete half way through the year.

Using a Web based solution, we felt that we could address the information maintenance problem. Also, by putting the information on the Web the targeted user group can get access to this information.

This was done as one of the graduate projects. Thus, the development part of the project had to be done within four months. Also, there were no funds available to get a technical person to maintain the Web site when developed. This made it necessary for us to develop the Web site in a way that is possible for the secretarial staff to maintain and update the information.

The information in the graduate school handbook did not have a rigid structure. Thus we decided to store the information for the Web site as SGML marked up documents as opposed to storing this information in a database. As information was not changing on a daily basis, we decided to write some scripts to regenerate the whole Web site from the SGML marked up documents every time the information changed.

**Process Model**

The steps we had to take were based on the Product Model described above. A major constraint we had was that this project had to be finished in four months. This only allowed us to have two iterations; a prototype and the finished product. Thus, it became very important to built in explicit steps to get feedback from the client at regular intervals. Thus, at each phase we adopted a process as shown in figure 8.

**Product Model**

From the context analysis, we found out that it is essential for the secretarial staff to be able to maintain the Web site. Thus, the product model for this Web site had to include the workflow associated with creating and maintain the information.

**Outcomes**

By performing a context analysis and developing a product model and a process model before we got into the details we were able to:
• **meet the client / user requirements:** this was achieved by enabling a high-degree of client-user-participation. All critical activities were built into the process thus avoiding any oversights;

• **complete the project on time:** as we followed a project plan derived from the process model we were able to track the progress and allocate resources when a problem was identified.

**Conclusion**

In this paper, I highlighted the need to have an appropriate product model and a process model when developing large and maintainable Web based information systems.

It is important for us to look at the wider issues related to the information systems before we dive into the details. Thus, the first step in the overall development process needs to be the context analysis. Context analysis consists of three main parts; looking at the need for developing the information system within a wider context, verifying whether the proposed solution addresses the critical issues and identifying the constraints with regard to development of the Web based information system.

Based on the context analysis we then have to develop a product model that will address the critical technical issues. Then we have to decide various activities that are required to create this product and organise these into a process model.

Our experience has shown that by doing this high-level analysis and planning when developing large and maintainable Web based information systems we can meet customer requirements within an agreed time frame and budget.

**Acknowledgments**

The work reported in this paper was done over many years. During this time, I worked with many colleagues and students on various aspects related to development of large and maintainable Web sites. I would like to acknowledge their input in shaping my thinking. The case study used in this paper was based on one of the student’s projects.

I would also like to acknowledge Dr San Murugesan who coined the term "Web Engineering" a few years ago.

**References**


