UNIVERSIDADE DE SÃO PAULO

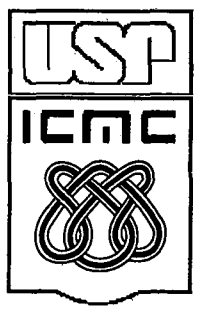
Instituto de Ciências Matemáticas e de Computação

Web Engineering Process – a Case Study from Academic Development

Débora Maria Barroso Paiva
André Pimenta Freire
Renata Pontin de Mattos Fortes

Nº 79

NOTAS



São Carlos - SP

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2577

Web Engineering Process – a Case Study from Academic Development

Débora Maria Barroso Paiva
André Pimenta Freire
Renata Pontin de Mattos Fortes

Nº 79

NOTAS

Série Computação

São Carlos – SP
Ago./2004

Web Engineering Process – a Case Study from Academic Development

Resumo. O desenvolvimento de software realizado por estudantes freqüentemente ocorre em curto período de tempo e difere dos modelos de processo de software convencionais, encontrados tanto na literatura quanto em ambientes empresariais. Este artigo apresenta uma pesquisa sobre o processo envolvido nesse tipo de desenvolvimento, relativo a uma aplicação *web*, realizado por estudantes. O projeto inclui o desenvolvimento das sucessivas versões do software, que visavam atender os requisitos associados com a demanda do meio acadêmico. Apresentamos as atividades técnicas e gerenciais do processo de engenharia de *web*, que tem sido aplicado para manter as evoluções das versões do software e para lidar com o conhecimento adquirido em cada estágio do seu desenvolvimento. Por meio do estudo de caso que tem sido conduzido, as lições aprendidas também são discutidas.

Web Engineering Process – a Case Study from Academic Development

Débora M. B. Paiva, André P. Freire and Renata P. M. Fortes

Universidade de São Paulo
Mathematics and Computing Science Institute
São Carlos – SP, Brazil. P.O. Box: 668 CEP: 13560-970

{debora, apfreire, renata}@icmc.usp.br

Abstract. The development of software by students is usually very fast and presents several distinctions from more traditional software process models found in the literature as well as in enterprise environment. This paper reports on an investigation of the process involved in the development of an academic web application by students. The project involves the development of successive software versions in order to attend requirements associated with the academic demand. We present the technical and management activities of the web engineering process, which have been applied to maintain the evolutions of the software versions and to deal with the knowledge acquired at each stage of development. By means of a case study that has been conducted, lessons have been learned, which are also discussed.

1 Introduction

The process of software development by students presents several distinctions from more traditional software process models found in the literature as well as enterprise environment. The process, that in general involves students performing several roles, is usually very fast, and sometimes its main goal is merely to validate some new approaches. However, the software product frequently has to be kept on evolving by other people, who usually do not count on good documents or other kind of artifacts.

Web Engineering processes have emerged as efficient approaches to develop Web applications in a spiral approach, allowing software prototyping and evolving [5]. The case study presented in this paper reports on the adop-

tion of a Web Engineering process for developing a web application, based on academic resources.

The initial motivation for building the application was that academic context encompasses a series of activities related to teaching and research. Lecturers and academic staff often need to establish dates for exams, to communicate extra classes and to book equipments such as projectors and computers. Other people involved in the academic context, mainly students and secretaries, need to be aware of general schedules as well. For example, secretaries need to know the available schedules of all lecturers in a group to be able to schedule a meeting at an appropriate time. Therefore, the importance and utility of a collaborative calendar is notable [9,15], mainly for supporting the arrangement of academic and educational engagements. *No Risk Planning* is an electronic calendar, available at URL "`http://coweb.icmc.usp.br/norisk/`"[1], developed to facilitate the planning and scheduling of activities carried out in the academic context. It allows arranging engagements for working groups and, alternatively, people can use *No Risk Planning* as their own personal diary.

Many projects have been developed in order to offer a shared calendar or a shared agenda. WebCal [6], Web Organizer [24] and Project Place [21] are systems in which users can add, consult, change or remove items such as room reservations, announcements about meetings or conferences and general appointments. In distance learning, for example, they can be used as a public resource to support selecting important dates: people need to consult a calendar frequently to keep up with academic and educational activities of the distance-learning paradigm.

Although many projects related to the shared calendar have been developed, we emphasize the main motivations for the development of the *No Risk Planning* project: to satisfy educational and research calendar requirements; to offer features to help each member of group find common available time; and to study the web engineering process through the development of an application, training developers in academic environment to deal with the practitioner's web development process. Moreover, the process presented special characteristics since students had to perform several developers' roles and to share their time with academic tasks. As a result, they had to prioritize certain quality assurance tasks and we could observe how the progress of adoption those tasks were made.

This paper describes the process of *No Risk Planning* development by students of four versions of the *No Risk Planning* calendar. The requirements have been established concerning general academic needs. Other activities

[1] In order to provide means to test the *No Risk Planning* calendar, there is an user/password available as anonimo/anonimo

such as planning, analysis, engineering and tests have been accomplished. Lessons learned about the process have been assimilated by the team of students responsible for the project and how the process was applied could be observed as a case study. Particularly, the main quality assurance tasks that have been pointed as essential to evolve the project are software configuration management and requirements engineering.

The paper is organized as follows: Section 2 introduces basic concepts about web engineering. Section 3 presents the web engineering activities related to the development process of the *No Risk Planning* calendar. Section 4 describes the main characteristics and functions of the calendar as well as its four versions. In Section 5 we summarize lessons learned and Section 6 presents our concluding remarks.

2 Web Engineering Overview

In the last years, a number of web applications have exploded as a consequence of its immense potential audience and its apparent facilities for publishing [23].

Discussions about the applicability of software engineering concepts to the development of web applications have been occurred [11]. The dimension of the navigation space, the high amount of information and the hypermedia concepts are important issues that need to be carefully addressed when web applications are designed. Therefore, web engineering is an important discipline because it provides a systematic and disciplined approach for designing, developing, documenting and maintaining web applications, emphasizing specific web-domain characteristics. The goal is that applications result in high quality products, avoiding a tangled structure [5, 23].

Characteristics such as immediacy, security, aesthetics and continuous evolution were identified as fundamental for a successfully web-based development [20]. It is possible to observe that agile methodologies are demanded and that technical and management activities need to be conducted aiming the continuous evolution of the applications.

Proposed web engineering models and development environments take into account hypermedia, information systems and software engineering fields. Essentially, it is necessary to adapt the original concepts and methodologies according to the specificity of web as a novel communication medium.

According to Fraternali and Paolini [7], the state of the practice of web development reveals that most solutions concentrate on implementation, paying little attention to the design process. In general, it is widely disseminated visual HTML editors, HTML-SQL integrators and web application genera-

tors. However, some incipient researches has been developed aiming to establish models and frameworks for web development.

Troyer [23] proposed the WSDM (*Web Site Design Method*) approach that takes as starting points the needs and requirements of the intended audiences of the web site. This approach is called audience-driven. In the WSDM, the first step is to define the *mission statement*, which expresses the purpose and the subjects of the web site and declares the target audience. Based on this mission statement, the different kinds of users are identified and classified (*audience classification* stage). In the next step, *audience class characterization*, the characteristics of the different audience classes are given. In the *conceptual design* stage, information modelling and navigation design are performed. During information modelling, it is modelled the information requirements of the different audience classes. During navigation design it is described the conceptual structure of the web site and modelled how the members from the different audience classes will be able to navigate through the site. During *implementation design* stage, it is designed the page structure and the appearance of the web site. The last stage, *implementation*, is the realization of the web site using the results of the previous stages.

In a similar way, Fraternali and Paolini [7] proposed the Autoweb project, a methodology and a development environment for data-intensive web sites. An important contribution of this project is the HDM-lite, a hypermedia design model tailored to the development of web applications that includes a notation for specifying presentation at a conceptual level. In Autoweb, the first step is the *collection of requirements* and their formalization as a set of conceptual schemas in HDM-lite. The second stage is the *generation of the supporting database*, which consists of two parts: a minidatabase containing a representation of the structure, navigation and presentation and an empty database for storing the application content. The last stage is the *implementation and deployment of the web application.*

Another important web engineering approach was proposed by Pressman [20]. The web engineering framework – WEBE – embodies an effective process model populated by activities and engineering tasks. It begins with a *formulation* stage that identifies the goals and objectives of the web applications. *Planning* stage estimates overall project cost, evaluates risks associated with the development and defines a granulated development schedule for the initial increment of the framework. *Analysis* stage establishes technical requirements for the application and identifies the content items that will be incorporated. The *engineering* stage incorporates content design and production. The purpose of these stages is to design, produce and acquire all text, graphics and other components that integrate the application. At the same time, a set of technical design tasks is conducted. In the *Page Generation* stage, the content defined in the engineering stage is merged with the design of the architecture,

navigation and interface to produce executable web pages. *Testing* stage attempts to uncover errors in the web application. Each increment produced as part of the WEBE process is reviewed during customer evaluation. This is the point at which changes are requested. These changes are integrated into the next path through the incremental process flow.

As stated by Garzotto et al [10], the main goal of such models is to help designers to reason in a structured way on aspects that are specific of hypermedia. Furthermore, they should enhance the communication among the members of a project, such as programmers, system analysts, software managers, users and content-authors.

It is worth noticing that the various process and methodologies present similar approaches for web development. There are only specific details differencing them. In essence there is quite a lot of agreement [12]. So, we decided to guide the *No-Risk Planning* development by Pressman framework, that is sufficiently generic to accommodate the accomplishment of activities suggested by other methodologies, according to the specific necessities of each project. In fact, we carried out some stages in such a way that we could avoid the overload of activities for the developers. We selected what we considered to be the more important practices from the methods, based on our own experience and having in mind that the students should be able to get advantages on their learning during the process period.

3 Activities Related to the *No Risk Planning* Development Process

Many activities and engineering tasks should be executed in order to build a reliable and usable web-based system. This section presents how the main stages of the *No Risk Planning* life cycle have been performed.

3.1 Formulation and Planning

In the *formulation* stage, the group responsible for the project should answer a set of questions. Essentially, it is important to state the motivation for the development of the new system, why the system is necessary and who will use it.

As stated before, the main motivation for the *No Risk Planning* project was related to the relevance and utility of applications that address work in groups in educational and research contexts. It is also important to observe that *No Risk Planning* calendar resulted from the academic needs of our research

group: InCA-SERVE[2] project [19] has as objective offering infrastructure (in terms of software) for building applications of ubiquitous computing in teaching and research. Frequently, lecturers and researchers need to book classrooms that hold all equipments, register appointments and meetings of a working group and schedule lectures and presentations.

Therefore, a system such as the *No Risk Planning* is necessary because it allows arranging appointments and reservations systematically. This is a common stage in academic context and the use of accessible application and database can ease and improve working group. As a result, the main users of this system are students, lecturers, researchers, secretaries and everyone involved with academic issues.

No Risk Planning was to be used by different people involved in distance learning to help scheduling and presenting the engagements related to the courses. In traditional education, academic staff has used it when it is necessary to choose the appropriate time to schedule a meeting.

The important tasks carried out during the planning stage were (a) setting the general activities of the life cycle and (b) defining the chronogram for the cycle in the context of the framework.

3.2 Analysis

According to Pressman [20], the concepts of the traditional software engineering can be applied for the web engineering analysis. In the beginning, the detailed requirements of the *No Risk Planning* project were not well defined. Therefore, a functional prototype was built with the objective of improving the requirements analysis. We took into account the main needs of the potential users to build the prototype. Its main characteristics are described below.

There are two main *No Risk Planning* types of users: administrators and final users. Administrators are responsible for user management. The administrator main screen offers statistical data about *No Risk Planning* use. There is information related to number of institutes, departments, users and groups using the application. Administrator is responsible for (1) managing users areas; (2) creating, changing and removing institutes and departments; (3) defining working groups and (4) setting the system configuration.

The final user of *No Risk Planning* calendar is responsible for managing her/his calendar (adding, changing, searching and removing appointments). The first step is the login and password authentication. After authentication, the user can view her/his appointments as shown in Figure 1. In the applica-

[2] InCA-SERVE is a project funded by an international cooperation program between CNPq/ProTeM-CC in Brazil and NSF/CISE in the U.S. In Brazil, the project is also supported by FAPESP and CAPES

tion, appointments can be classified as *eventual, important, common, leisure or academic*. Each of them is represented with a different color. There are also appointments of the group to which the user belongs. The acronym or the abbreviation of the group name is presented between parentheses. On the top of the screen, the user has some options as *Personal Diary, Search, Your Data, Groups and Help*.

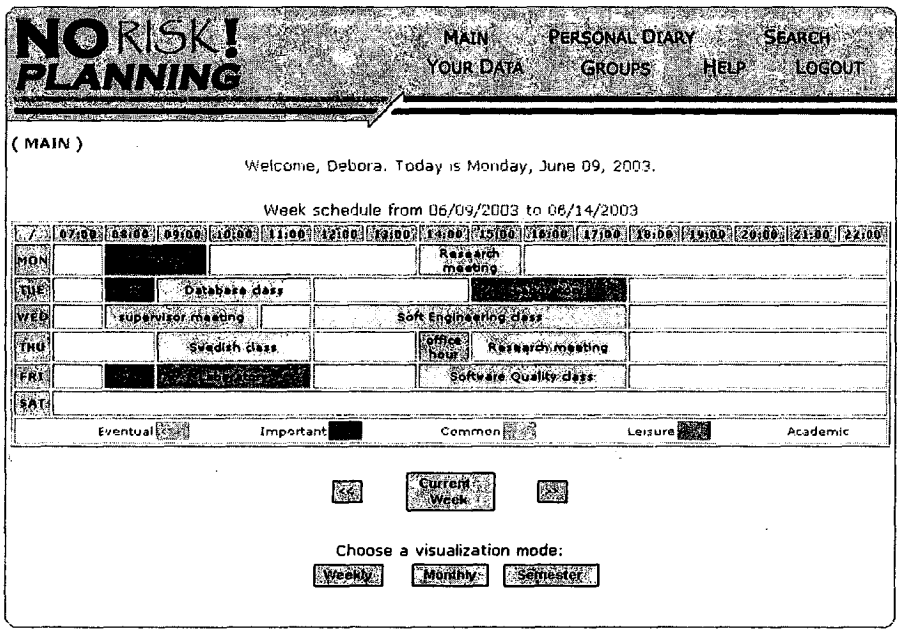


Figure 1. Main screen of *No Risk Planning*

Ten people, two lecturers and eight Computing Science students (potential users) informally evaluated the prototype. They were asked to execute some elementary tasks and express their opinion about implemented functionalities. During the evaluation, many suggestions were gathered that helped to establish other *No Risk Planning* requirements.

While the requirements analysis stage was taking place, the types of analysis recommended by Pressman [20] were carried out in the *No Risk Planning* project.

The *content analysis* stage identified and created the necessary text, graphics, images and icons files to compose the application.

The *interaction* analysis stage was carried out by means of use-cases diagram building. This type of diagram describes the system functionalities observed by external actors. An actor can be a user, a device or other system [9]. In *No Risk Planning*, the actors are people with distinct roles: administrator,

calendar users, group members and group creator. The administrator actor is responsible for the global system management. This includes system configuration and users management. The calendar user actor manages her/his personal diary. The group member actor is the user that is part of some working group. She/He can register appointments for her/his group, interact with other members by means of a chat program and share files and documents as well. The group creator actor is responsible for managing the group, for instance, including and removing group members.

The *Functional* analysis stage involves describing all operations and functions in details. Some UML diagrams were stated: class diagram, activity diagram, collaboration diagram and state diagram [22]. Additionally, operation lists were generated with the objective of to detail each operation of the *No Risk Planning* system, the relationships among operations, the information related to input and output objects and the expected result of each operation.

During the *Configuration* analysis stage, technical specifications for the *No Risk Planning* project have been established: it is a system available on the Internet that can be accessed through the World Wide Web (WWW). We are using PHP (programming language), MySQL (database) and Apache (web server).

The next stage (*Engineering*) was performed according to the initial cycle of the framework proposed by Pressman focusing on the *navigation design* stage. The main activities carried out are presented in next section.

3.3 Engineering

The *analysis* stage stated operations for *No Risk Planning* application generating the operations list. In the *engineering* stage, the simple operations were considered as modules and complex operations were divided into modules. Afterwards, the *navigation design* was accomplished. It was established the links among the modules: for each module was stated which modules it calls and which modules call it. As a result, the *navigational model* was described and was visualized as a whole, giving a general vision of the system.

3.4 Generation and Testing

In this stage, the source code was generated according to the analysis and the engineering specifications. The prototype was partially reused: some functional modules constitute the application resulting of the first increment of the

framework proposed by Pressman. Simple unit and integration tests were accomplished but we had not used specific techniques for this purpose.

3.5 Costumer Evaluation

In this stage, another evaluation was carried out with potential users of the *No Risk Planning* project. It was planned according to ISO/IEC 14598-5 International Standard recommendations [13]. This standard establishes that *Requirements Specification* stage should define the objectives of the evaluation, stating the requirements that will be investigated. *Specification* stage should define the evaluation scope and the measurements that will be carried out. *Planning* stage should document the procedures for executing the evaluation. *Evaluation* stage consists of actions that implement definitions of early stages. *Conclusion* stage is about interpreting evaluation results.

All requirements stated for the *No Risk Planning* project were evaluated. Besides functional requirements, it was included the usability quality requirement focusing on the attributes identified by Olsina et al [17]: global site understandability, online feedback and help features, and interface and aesthetic features.

The *Specification* stage established that evaluation would be accomplished by means of experiments. The *Planning* stage defined the evaluator profile, the evaluation tasks, the questionnaires used for gathering information about the system and other aspects related to the preparation of the evaluation environment and the deployment of the necessary resources (for example, computers and software).

Nine people participated in the *Evaluation* stage. They were allowed to explore the whole system and they were asked to answer a questionnaire. The results demonstrate that users had problems with navigability and usability. In spite of this, positive aspects of the *No Risk Planning* usage were observed from the benefits of having an organized means of calendar information to exchange related academic activities. Hence, we keep on developing and maintaining the *No Risk Planning* software. Other ideas emerged from the *Evaluation* stage and will be used as input for initial stages of the next increment of the framework.

For all stages, the description of what happened in the project was collected by means of the register of project-diaries and TODO lists (to be done, being done and done activities). For this purpose, CoWeb [2] was utilized because it provides the services of collaborative authoring and editing of pages and uploads of any type of file. An example of these artifacts is shown in Figure 2.

4. *No Risk Planning*: Characteristics of the four versions

In the beginning, the requirements of the *No Risk Planning* project were not well defined. Although the undergraduate student responsible for the development of *No Risk Planning* had in mind the main characteristics of a personal calendar, some details about how to make group interactions ubiquitous in a calendar were unknown. Therefore, a functional prototype was built aiming to improve the requirements analysis. So, the **first version** (*NRP 1.1*) was built to be used by two main types of users: administrator and common users.

Figure 3 illustrates the relationship among the entities of the first version of *No Risk Planning*. The entities are: Administrator (who deals with users), Users (who deals with their Agendas), and Groups (of users); the main subject is the Agenda of each user that can be shared with the groups to which users belong.

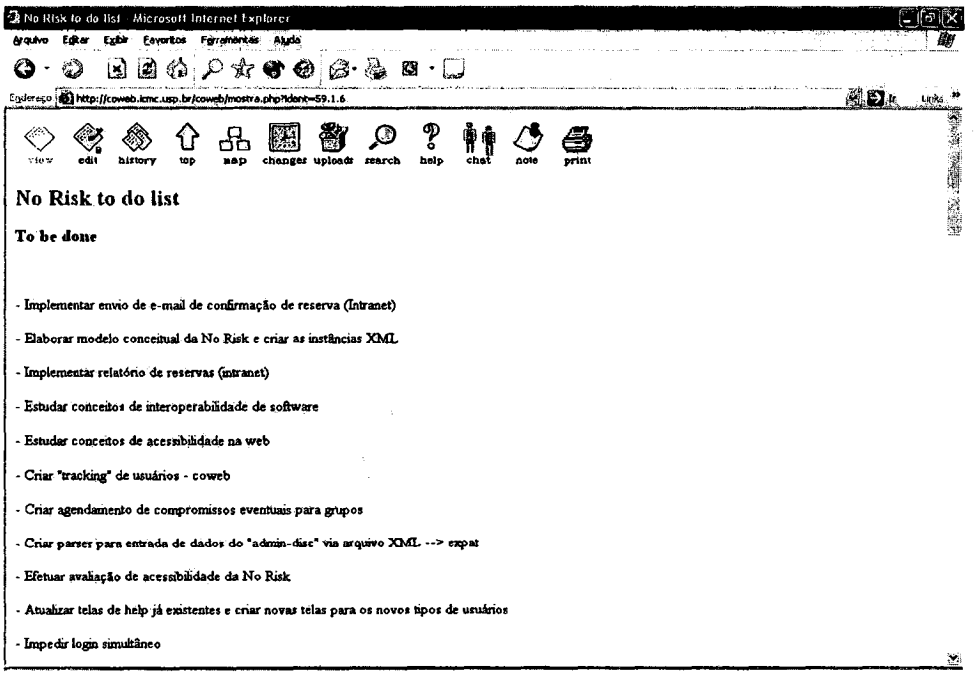


Figure 2 - Example of artifacts resulting from *No Risk Planning* process

The first version of the software product had 6,615 LOC, 71 files including HTML pages, 41 methods implemented in PHP and 14 tables of a Relational Database Management System - RDMS (MySQL). Moreover the code runs on Apache web server. The whole documentation of the product was made by

means of a technical report [16] written by an undergraduate student, including the relational schema of data and a brief description of the methods.

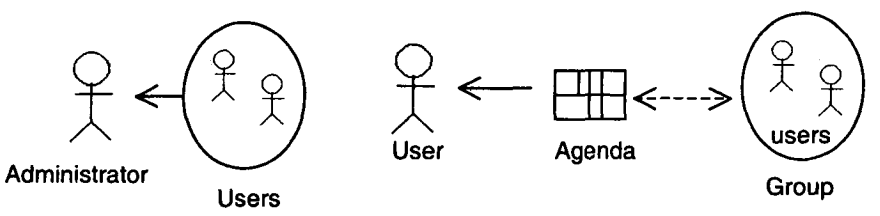


Figure 3: Entities attended by the first version of the *No Risk Planning – (NRP 1.1)*

To develop the first version the student and the advisor carefully chose a well-known set of resources such as programming language, database system and client-server architecture. The familiarity of those mechanisms could let them to advance in other issues related to interacting aspects required by the research under development. It took 3 months of development time and the proper skills on the technical resources used during the initial development made the project management easy.

The **second version** *(NRP 1.2)* had its developer changed. This fact has impacted its software process. Two new students have evolved the *No Risk Planning* code to support the schedule of eventual commitments. Furthermore, because of the intensive use of the tool, users asked for mechanisms to visualize their calendar monthly and by semester. Another small change was implemented to mark and present the timetables hourly, and no more in half hours as it was shown in the first version. The second version of software product had 8,359 LOC (26% increased from previous version), 81 files (14% increased from previous version), 45 (10% increased from previous version) implemented methods in PHP and 15 tables of RDMS (MySQL) (7% increased from previous version). The documentation was made by means of a technical report [22].

It took 8 months of developing time, and included the understanding and documenting activities. The artifacts documented the correspondent abstractions of code. Approximately 80% of the whole time was dedicated to implement the changing code and perform a preliminary test.

The **third version** *(NRP 1.3)* of *No Risk Planning* started aimed at addressing the issues related to a substantial use of the tool. Several lecturers and secretaries identified points of interactions that could be improved, for instance in situations when repetitive steps were demanding unnecessary effort.

The third version was implemented by the same students involved with the second version and it could be observed the high level of productivity occurred. The characteristics of the third version included: possibility of navigation among timetables displayed monthly, possibility of changing scheduled appointments, the data of the timetable can be exported as an image (png file), an appointment can easily be defined to be linked to an URL. The Figure 1 illustrates the main screen of this version. The third version of *No Risk Planning* also enabled the use of the calendar to support courses tasks. For that, a *course* and *courses administrator* user types were created. Figure 4 illustrates the relationships among the entities at the third version.

Another result of this third version of *No Risk Planning* project was its well-succeeded integration with the CoWeb [2]. The integration has been very useful as a support to the learning approach that exists in CoWeb, since it allows people to manage their tasks schedule. In this case, tasks are related to the collaborative editing of lecture web pages. For instance, by using the third version of *No Risk Planning*, a lecturer can arrange and announce the date the students should turn in a homework, and she/he can also state the related topics and important notes in a CoWeb page. Consequently, students can easily access both the established date and notes related to the work.

The third version of software product had 23,247 LOC (178% increased from previous version), 191 files (136% increased from previous version), 144-implemented methods in PHP (220% increased from previous version), and 18 tables of RDMS (MySQL) (20% increased from previous version).

The documentation has been elaborated with less detailed than the previous technical report, since only the essentially modified artifacts have been re-designed. It took (*only*) 3 months of developing time, and included the understanding and documenting activities. The artifacts documented the correspondent abstractions of code. Approximately 80% of the whole time was dedicated to implement the new code and perform a preliminary test. An important result was a paper reporting the process improvements presented in a Brazilian conference [18]. The paper emphasized technical and management activities carried out up to that moment.

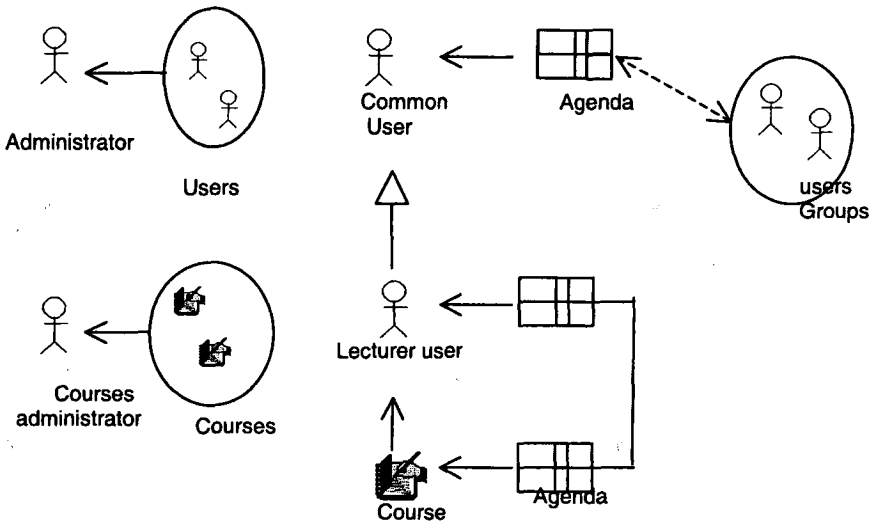


Figure 4: Entities attended by the third version of the *No Risk Planning*

The **fourth version** is a release *(NRP 1.3.1)* that consists of an important change from the third version. It was developed in order to support the management of classrooms. Different from previous versions developments, the fourth one was dedicated to move the characteristics of the owner of an agenda, and it can be seen as a new *branch* of *No Risk Planning* system. In the fourth version, academic users can interact to schedule the time of occupations of classrooms, which have their own agenda.

The fourth version of *No Risk Planning* was integrated to the intranet of the Mathematics and Computing Science Institute (University of Sao Paulo-Brazil)[3]. It took *(only)* 4 months of developing time, and included the requirements understanding and documenting activities. The artifacts documented all the correspondent abstractions of code. Approximately 30% of the whole time was dedicated to implement the new code and perform a preliminary test; the majority of the time was spent for requirements elicitation and interviews with users. Additionally, students had to understand the code of intranet in order to integrate to *No Risk Planning* and to uniform the authorization access according to the existing at intranet; this task took half month.

No Risk Planning for classrooms management is currently responsible for controlling the use of several rooms, such as laboratories, meeting rooms, and classrooms. By using the software, lecturers and employees can schedule rooms' reservations. There are currently 27 rooms (including 5 types of res-

[3] `http://www.icmc.usp.br/intranet/` - 'Consultar Disponibilidade de Sala'

ervations), 96 lecturer users and 39 employees registered in the system, and it has been used since August 2003.

The fourth version of software product has 11,702 LOC, 85 files, 47-implemented methods in PHP, and 7 tables of RDMS (MySQL). Because this version of *No Risk Planning* is a new branch of development, the comparison of its measures with the previous versions does not make sense. The documentation was made by means of a technical report [8].

Potential users have frequently evaluated the application and the data gathered are being used in ongoing works. It is worth noticing that the versions of the *No Risk Planning* calendar have been developed since the middle of 2001. Additionally, they will be important in the next framework cycle of evolutionary model: usability and navigability characteristics are the main focus for improving the application.

5. Lessons Learned

Among the main motivations for the development of the *No Risk Planning* project, we aimed to study the web engineering process through the development of an application, and training developers in academic environment to deal with the practitioner's web development process. Initially, the goals were well established, but we had not idea about how the process would be accepted and applied by students.

The challenges found concentrated in the beginning of each stage of the Web Process adopted by students. But, the actual process presented special characteristics since students had to perform several developers' roles and to share their time with regular academic tasks. Peculiar characteristics of a web development, such as immediacy, aesthetics and continuos evolution were regarded during the development. Final users had emergence of *No Risk Planning* usage, what motivated us to follow a rapid process. Moreover, the requirements evolved quickly because many lecturers and secretaries adopted *No Risk Planning* and suggested modifications. So, we have to maintain all the documents in such a way that the evolution of the project could be ensured.

Students had to prioritize certain quality assurance tasks and we could observe how the progress of adoption those tasks were made. Furthermore, the evolution of *No Risk Planning* versions can be seen as a positive result and its process has been established in a manageable and controllable way. Figure 5 exhibits the evolution of *No Risk Planning* software versions.

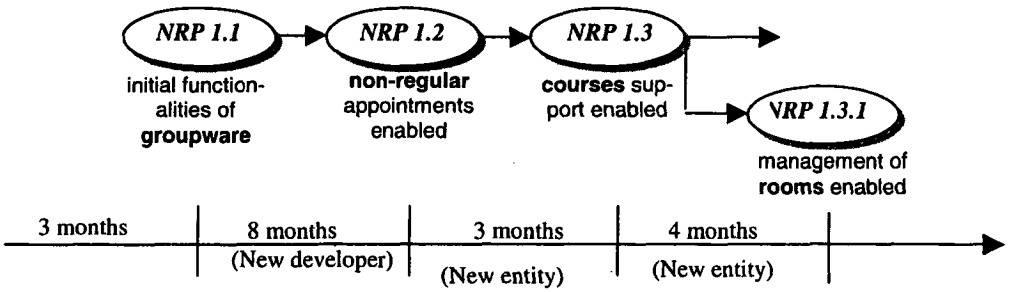


Figure 5: *No Risk Planning* (*NRP*) versions

To summarize the lessons learned, we can point out:

- the students often care of requirements, which have to be well defined and a lot of *design rationale* should be discussed in order they feel *familiar* with the information domain. It was evidenced by the spending time during the development of second version (*NRP1.2*), because the developer had changed, and it has impacted its software process leading to spending more time by documenting the previous version. Additionally, we noticed another kind of familiarity: the choice of programming resources (language, RDMS, and web server) was carefully made and the knowledge of those mechanisms could let them to further in other issues related to interacting aspects required by the research under development.

- the activities such as planning, analysis, engineering and tests have been accomplished without difficulties. The mechanisms to support the communication among developers helped to keep the tasks and respectives discussions registered and updated. Although the beginning of each stage requires students role exchange, and time to students become productive, they were *motivated* to *change* their activity;

- the process have been assimilated by the team of students responsible for the project, and the *main quality assurance tasks* that have been prioritized to evolve the versions are: software configuration management and requirements engineering.

6 Concluding Remarks

The development of a web application by students lead to the case study reported here. In fact, we deeply aimed to apply a web engineering process through the development of an academic actual application, and observe student developers dealing with the practitioner's web development process.

The *No Risk Planning* project started, motivated to be a useful collaborative academic calendar on the web. In the educational context, this application can address the planning of many academic activities. For instance, lecturers can establish dates for exams, communicate extra classes and book equipments.

This project regarded aspects of Web engineering focusing, mainly, on the WEBE framework. Activities such as *Formulation and Planning, Analysis, Engineering, Generation and Testing* and *Customer Evaluation* were accomplished for the first framework increment in detail. Next increments were performed according to the versions were developed, and special characteristics of the process were emphasized. Lessons learned have been assimilated by the team responsible for the project, such as the relevance of familiarity of students on the development tools and on information domain, the stimulation caused by students role exchanging during the process, and the main software quality assurance (SQA) tasks that have been pointed as essential to evolve the project (software configuration management and requirements engineering). Additionally, we saw the evolution of *No Risk Planning* versions as a positive result and its process has been established in a manageable and controllable way.

Nowadays, *No Risk Planning* has 44 common users, 46 lecturers, 81 disciplines and 29 groups registered. The perspective of the increased amount of all users has given us motivation to keep on the research of how the process can attend to their demand, and to see alternative technical solutions (for instance, by means of frameworks or design patterns) to improve its evolution.

The next version of *No Risk Planning* will be implemented using the concept of "Separation of Concerns" (SOC) [1] so that layout, content and logic can be separated during the software process [14]. SOC facilitates the development of service-oriented Web sites by making it more modular. Changes that affect one concern do not affect the other ones, for example [1]: **(a)** changes to the user interface do not affect the logical representation of the data as well as the processes; **(b)** changes to the process logic (e.g., the control flow) can be made independently of the specifications of the layout transformations, and even can be done without affecting existing transformations (and similarly for the data structures, as it is known for XSLT); **(c)** control of the process flow (workflow logic) and generation of user interface layout can be independently distributed as required (client, server) and **(d)** developers can concentrate on the aspects they are interested in (process structure, data structures, layout). When developing the process and data structures, layout aspects can be ignored, and when developing the layout only a partial understanding of the structured specification is required (e.g., only activity types but not their detailed dependencies). Also using more abstract models for specifying the structural models facilitates their understanding.

Students are motivated with the new challenge of SOC technology and we intend to observe how the web engineering process will address it. In the new version of *No Risk Planning*, SOC is going to be achieved by implementing business process using PHP[4], generating the content as an XML document [25] through a template handler component called Smarty[5], and transforming the content information to the respective layout by using XSLT [26] style sheets.

Acknowledgements

We would like to FAPESP and CAPES agencies for funding our researches.

References

[1] Aberer, K; Datta, A; Despotovic, Z. *Separating business process from user interaction in web-based information commerce.* Electronic Commerce Research, Vol. 3, Issue 1-2 (2003) 83-111, ISSN 1389-5753

[2] Arruda Jr., C.R.E.; Pimentel, M.G.C. *Design and Implementation of a Collaborative Editor System.* Eletronic Journal of Scientific Initiation, Brasil, Ano 1 (2001) In Portuguese

[3] Athula, G.; Murugesan, S. *Web Engineering: An Introduction.* IEEE Multimedia, Vol. 18, Issue 1 (2001) 14-18

[4] Benbunan-Fich, R. et al. *A Comparative Content Analysis of Face-to-Face vs. Asynchronous Group Decision Making.* Decision Support Systems, Elsevier, Issue 34 (2002) 457-469

[5] Costagliota, G., Ferrucci, F., Francese, R. *Web Engineering: Models and Methodologies for the Design of Hypermedia Applications.* In Chang, S. K. (ed): Handbook of Software Engineering & Knowledge Engineering, Vol. 2 (2002) 181-199, World Scientific

[6] eXtropia home page. [Online]. http://www.extropia.com/scripts/calendar.html [21/06/2002]

[7] Fraternali, P., Paolini, P. *Model-Driven Development of Web Applications: The Autoweb System.* ACM Transactions on Information Systems, Vol. 18 Issue 4 (2000) 323-382

[8] Freire, A.P., Fortes, R.P.M. *Documentation of Classrooms Management System – Intranet of the Mathematics and Computing Science Institute,* University of Sao Paulo. ICMC-USP, Brasil (2003) 48 pages. (Technical report, number 222). In Portuguese

[9] Furlan, J.D. *Modelling Objects with UML.* Makron Books (1998) In Portuguese

[4] http://www.php.net

[5] http://smarty.php.net

[10] Garzotto, F., Mainetti, L., Paolini P.: Hypermedia Design, Analysis and Evaluation Issues, Communication of the ACM, Vol. 38, Issue 8 (1995)
[11] Glass, R.L. A mugwump's-eye view of Web work, Communications of the ACM, vol. 46, number 8, 2003.
[12] Griffiths, G.; Hebbron, B.D.; Lockyer, M.A.; Oates, B.J. A Simple Method & Tool for Web Engineering. Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, 2002, 755-762.
[13] ISO/IEC 14598-5, *International Standard. Information Technology - Software product evaluation* - Part 5: Process for evaluators (1997)
[14] Kerer, C.; Kirda, E. *Layout, content and logic separation in web engineering.* In: Web Engineering, Software Engineering and Web Application Development.: Springer-Verlag, (2001) 135-147 ISBN 3-540-42130-0
[15] Lee, H. Your Time and my Time: a Temporal Approach to Groupware Calendar Systems. Information & Management, Elsevier, 40 (2003) 159-164
[16] Moura, M. V. No Risk Planning – a web-based system for supporting work in-group, ICMC-USP, Brasil (2001) 53 pages. (Technical report). In Portuguese
[17] Olsina, L., et al. *Specifying Quality Characteristics and Attributes for Web Sites.* First ICSE Workshop on Web Engineering, ACM (1999)
[18] Paiva, D. M. B.; Sanches, R; Fortes, R. P. M. *Software Process Improvement in the Academic Development Context of a Web-Based Agenda.* In: 14o. Internacional Conference on Software Technology. Proceedings. Brasil (2003) 56-68. In Portuguese
[19] Pimentel, M. G. C., Abowd, G. D., Brotherton, J. *Requirements for Capture-based Support to Learning.* Communications of the ACM (2002)
[20] Pressman, R. S. *Software Engineering.* 5th edition, McGraw-Hill (2001)
[21] Project Place home page [Online]. http://projectplace.com/ [21/06/2002]
[22] Ribeiro,T.M.; Fortes, R.P.M.; Freire, A P. – Documentation of No Risk Planning Calendar ICMC-USP, Brasil (2003) 69 pages. (Technical report, number 182). In Portuguese
[23] Troyer, O. *Audience-Driven Web Design.* In: Rossi, M. (ed.): Conceptual Modeling in the Next Millennium, CRC Press, USA (2000)
[24] Web Organizer home page. [Online]. http://weborganizer.sourceforge.net/ [21/06/2002]
[25] World Wide Web Consortium (W3C). [Online]. Extensible Markup Language (XML). http://www.w3c.org/TR/REC-xml [18/02/2004].
[26] World Wide Web Consortium (W3C). [Online]. XSL Transformations (XSLT). http://www.w3c.org/TR/xslt [18/02/2004]

NOTAS DO ICMC

SÉRIE COMPUTAÇÃO

078/2004 SOUZA, R.; SILVA, C.; ARENALES, M.N. – Método do tipo dual simplex para problemas de otimização linear canalizados: teoria

077/2004 SILVA, A M.P.; NUNES, M.G.V. - Using multiword lists for lexically aligning brazilian portuguese and english texts..

076/2004 BÍSCARO, H.H; CASTELO FILHO, A.; NONATO, L.G. – A topological approach to curve reconstruction from scattered points.

075/2004 NONATO, L.G.; CASTELO FILHO, A.; CAMPOS, J.E.P.P.; BÍSCARO, H.; MINGHIM, R. – Topological tetrahedron characterization with applications in volumetric reconstruction.

074/2003 TOMÉ, M.F.; CUMINATO, J.A.; CASTELO, A.; FERREIRA, V.G.; McKEE, S. - Recent Developments in the MAC Technique.

073/2003 SOUZA, F.S.; MANGIAVACCHI, N.; NONATO, L.G.; CASTELO FILHO, A.; TOMÉ, M.F.; FERREIRA, V.G.; CUMINATO, J.A.; McKEE, S. – A front-tracking finite difference method to solve the 3D navier-stokes equations for multi-fluid flows with free surfaces.

072/2003 MANGIAVACCHI, N.; CASTELO FILHO, A.; TOMÉ, M.F.; CUMINATO, J.A.; OLIVEIRA, M.L.B.; McKEE, S. – A numerical technique for incluiding surface tension effects for axisymmetric and planar flows using the gensmac method.

071/2003 TOMÉ, M.F.; GROSSI, L.; CASTELO, A.; CUMINATO, J.A.; MANGIAVACCHI, N.; FERREIRA, V.G.; SOUSA, F.S.; McKEE, S. – A numerical method for solving three-dimensional generalized Newtonian free surface flows.

070/2003 SANTOS, F.L.P.; MANGIAVACCHI, N.; CASTELO, A.; TOMÉ, M.F.; CUMINATO, J.A. – A novel technique for free surface 2D multiphase flows.

069/2003 VIANNA, A.C.G.; ARENALES, M.N.; GRAMANI, M.C.N. – Two-stage and constrained two-dimensional guillotine cutting problems.