Towards Reconciling Quality and Agility in Web Application Development

Américo Sampaio[1] , Alexandre Vasconcelos[1] and Pedro R. Falcone Sampaio[2]

[1] Centro de informática, Universidade Federal de Pernambuco
Recife, Pernambuco – Brazil
{atfs, amlv}@cin.ufpe.br
[2] Computation Department, University of Manchester Institute of Science and Technology
Manchester, United Kingdom
p.sampaio@co.umist.ac.uk

Abstract. Delivering high quality web applications complying with severe project delivery time constraints is still an elusive goal for a software process. In many software projects, development teams often resort to "short cuts" in the software development process avoiding recommended software process disciplines to speed up delivery. The usual side-effects of this ad-hoc approach are low software quality and high maintenance costs. In this paper we empirically address the tension between quality and agility (speed) in web application development describing a set of software disciplines that were added to Extreme Programming to improve web software quality without sacrificing development agility.

1 Introduction

Delivering high quality web applications complying with severe time constraints is a highly challenging dilemma within the software engineering community [20]. In many software projects, development teams often resort to "short cuts" to accelerate the development process, adopting ad-hoc approaches to build web applications [3,7]. In these situations the success of the project relies heavily on the skills and knowledge of the people in the software team, increasing the risk of potential negative side effects on usability, maintainability and robustness of the application. Therefore, it is highly important to have a more rigorous and organized approach to build high quality web applications while retaining agile properties in the development effort.

A possible approach towards reconciling quality and speed can follow two converging directions: (1) start with a prescriptive software process such as Rational Unified Process [21] and evaluate how the gradual relaxation of disciplines under similar project development conditions affects quality and speed; or alternatively, (2) start with an agile software process and evaluate how the gradual incorporation of disciplines under similar project development conditions affects quality/speed.

In this paper we adopt the latter approach by departing from an agile software process [1] and gradually incorporating quality related software disciplines. A discipline represents a set or related process elements (artifacts, activities, roles) grouped by a common theme or objective [4,21]. Some examples of disciplines are: requirements definition and revision sessions, design inspections and release planning. The term "discipline" was chosen for two reasons: (i) other similar terms such as *stage* or *phase* are more appropriate to sequential software process models (waterfall) and are not well suited for incremental and iterative models. (ii) "discipline" is a term that increased in popularity due to its usage in mainstream processes such as RUP [21].

The agile way of building software is a viable alternative to traditional methods that has already shown encouraging results over the last years [1,8]. The resulting process XWebProcess [2,17] is an agile method for building web applications that seeks to retain the agile properties of Extreme Programming (XP) while enforcing quality improving disciplines.

During the process of extending agile methods with quality related disciplines to develop XWebProcess, two key elements are being experimentally investigated:

- Effectiveness of disciplines: the prescribed disciplines added to XP actually improve the quality of the delivered artifacts in factors such as better UI design, usability, and structure;
- Efficiency of the disciplines: the prescribed disciplines added to XP do not sacrifice agility.

The agility of XWebProcess has been compared to XP via an experimental setting which shows its superiority in supporting web development dimensions such as requirements gathering, user navigation design, and software testing, helping to deliver superior quality artifacts while retaining the agile properties of Extreme Programming.

The remainder of this paper is organized as follows. Section 2 discusses the tension between quality and agility in web application development. Section 3 presents an overview of XWebProcess describing how the process deals with web development quality attributes. Section 4 describes portions of the experiment and survey conducted to analyze XWebProcess empirically aiming to verify the total effort spent in the process disciplines and also to verify process and product quality factors. Finally, section 5 presents a summary and future work.

2 Tension between quality and agility

Striking the right balance between speed and quality is a challenging undertake due to the inverse forces affecting the two attributes [18,19]. When asked to speed up the development, developers tend to focus on the accomplishment of the core functional requirements abandoning or reducing the amount of effort spent in tasks such as documentation, inspections, unit testing, and documentation, which will ultimately impact the maintainability and longevity of the delivered software artifact.

To achieve the speed-up necessary for faster cycle times, a common project management approach is to perform deviations from a defined process model [20]. While the process acceleration issue has received considerable attention in the literature [3,7,14,15], the reconciliation of quality and speed is still a challenge

[19,20]. To perform process deviations and assess the impact on quality, two possibly converging experimental directions can be enacted: depart from a prescriptive process (e.g., RUP) and evaluate how the gradual relaxation of disciplines under similar project conditions affects quality/speed or alternatively, depart from an agile software process and evaluate how the gradual incorporation of disciplines under similar project development conditions affects quality/speed.

The choice of departure point and level of relaxation in the process within the experimental spectrum will often depend on the problem domain and the business strategy relevance of the system:

- *Domain issues*: for safety critical projects, for instance, there is a limited leeway for relaxing process disciplines given the regulations usually linked to projects (e.g., breadth and depth of testing, documentation scope, formal proofs attached to critical code elements). A natural starting point would be to depart from a prescriptive process and opportunistically relax some disciplines provided that regulations and key safety issues are met.
- *Strategic relevance*: for company mission critical projects where timeliness is a key element of the strategy (e.g., B2C site development), a natural starting point would be an agile process and successively experiment with added disciplines that have a noticeable effect on quality (e.g., breadth and depth of inspections)

In our work we restrict the scope to address business strategy critical web applications that do not encompass safety critical aspects. In particular, the short time frames to develop and deliver strategic web software only helps to aggravate the tension between quality and speed. Within this framework, the following trade-offs provide additional challenges:

- *Agility vs Product-related quality factors*: As a web application often reaches a wide variety of users/organizations, it is highly critical to verify user requirement fulfillment criteria considering the different user perspectives. This can be a challenging task for web applications as users can have different kinds of needs, intents, and IT expertise. As well as checking that the system does what is required by users it is also imperative to asses its performance and robustness in supporting concurrency, heavy network loads and different kinds of clients (hardware, OS, browsers). Moreover, from the user´s perspective, the web application must present interesting usability features providing not only an attractive appearance but also an easy-to-use application facilitating navigation through its content.
- *Agility vs Process-related quality factors:* A web development process has to balance the need for speed without sacrificing the development of artifacts that impact on quality issues such as maintainability and longevity of the application. It is necessary to produce a flexible and structured architecture, have precise descriptions of requirements and a sound testing strategy to cater for non-functional requirements. The software process itself should also be defined in a way that it is easy to understand, execute and tailor, facilitating improvements and adaptations necessary to comply with organizational or project needs.

Considering the possible approaches and trade-offs described above, XWebProcess is aimed at providing an agile process based on Extreme Programming that will deliver a reasonable balance between agility and product/process quality attributes. The process is focused at building web applications in a fast and organized way providing disciplines that improve quality of the produced artifacts - this is detailed in the next section. Moreover, the definition of the process itself seeks to provide simple abstractions that make the process easy to understand, to apply and also to adapt to further needs.

An overview of the quality model used to assess XWebProcess is described in Table 1. The product quality factors were chosen based on the description found in ISO 9126 standard [23] and the process quality factors used are described by Sommerville in [13]. These definitions will be used in Section 4.2 where the core results from the survey are presented.

Table 1. Quality Model Used

Quality Model			
Quality Factor	**Characteristic**	**Definition**	**Measure**
Product Related	Usability	Easiness to be understood and used by the users (user-friendliness) of the product	H M L*
	Reliability	The product presents few faults and has appropriate mechanisms to recover from failures	H M L
	Maintainability	Easiness to evolve and correct defects on the product	H M L
Process Related	Easiness to evolve	Easiness to evolve the process to comply with new organization or project requirements	H M L
	Easiness to understand	Easiness to understand the definition and structure of the software process elements	H M L
	Visibility	Easiness to follow the process activities and know what to do and when to do, and also what are the inputs and outputs for each activity	H M L

*** H, M, L stands for High, Medium and Low**

3 XWebProcess = XP + Quality Disciplines

In this section, XWebProcess is described in two views using the software process modeling language SPEM [4] to facilitate process construction [5]. The use of SPEM enables an abstract description of the software process core elements (artifacts, roles, activities, etc.) and the description of how they relate to each other, facilitating the use of the process by the development teams. The SPEM notation also helps to illustrate the adaptations and tailorings performed over XP to target web application development. SPEM was chosen due to its OMG standard status for software process modeling and due to the extensive endorsement provided by software companies such as IBM, Rational Software and Unisys.

The first view of the process model uses UML´s activity diagram with the discipline stereotype defined in SPEM as shown in Figure 1. This model helps to understand how the process behaves through time (dynamic view). The disciplines highlighted are the ones adapted or inserted in the original XP modeling, shown in [2].

XWebProcess starts with an exploratory discipline encompassing experiments with technologies, architectures and system prototypes aiming at verifying viability of the project and defining initial requirements.

Afterwards, clients and programmers write story cards representing requirements of the next release. Programmers estimate the effort of implementing each story based on their past experience. The decision of what stories are implemented first is left to clients, who define priorities for the stories and select the ones with higher priority to be implemented first. The number of stories implemented in the release depends on the speed of the team and on the difficulties estimated for each story. Therefore, programmers and clients need to agree and plan the release together.

Inside a release, many iterations occur. In each iteration, stories are implemented and tested. Every iteration contains a set of disciplines, which are frequently enacted, including: plan iteration, design, writing of unit tests, coding, testing and integration. Moreover, during an iteration the requirements can change and previous estimates can be reassessed to reflect new customer's needs. Functional test cases as well as web navigation and presentation design are done in parallel with the previous set of activities in order to generate important artifacts (functional test cases and web components) used later.

At the end of an iteration, it is important to verify if the functionalities implemented conform to what was specified previously. So, while functional testing is done to assure that the system does what is intended to, web testing is performed to verify if the system works appropriately when considering non-functional issues.

If the iteration corresponds to the last iteration of the release, the current version of the system enters into production. This can be done in the client's company or in another place that simulates the real production environment. After the first release of the system is delivered the web support activities start. The process finishes when all stories are implemented and delivered to the client.

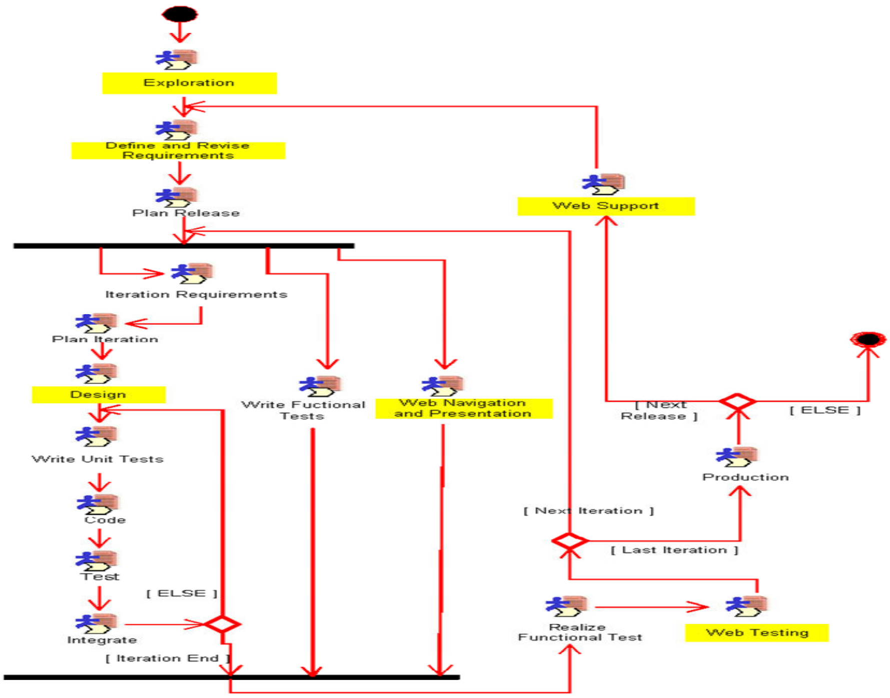

Fig. 1. Dynamic modeling of XWebProcess

It is important to mention that XWebProcess is a general web development process that does not depend on any specific technology, method, tool or technique. For instance, the process can be supported either with .NET or J2EE platforms. It also works with OOHDM [9], OOWS [10] or W2000 [22] that are design methods for modelling web applications. What is important to consider is if the specific method, tool or technology will have a negative impact on the agility of the process.

The highlighted disciplines in Figure 1 relate to core issues that a web development process should tackle. While some of them were adapted from the previous modeling of XP, not shown in this paper but shown in [2], others were inserted in XWebProcess. Table 2 explains the core disciplines and why they were added or modified in XP.

Table 2. Inserted and adapted disciplines

DISCIPLINE	EXTENSION	DESCRIPTION
Exploration	Modified	Modified to include prototype sessions. During initial explorations is important to investigate if the system is viable or not. In web applications it is important to conduct prototype sessions with clients and business managers to help define initial requirements, scope and business

		goals of the system.
Define and Revise Requirements	Modified	Modified to include an architecture design activity. In web development the construction of a sound and flexible architecture is vital for the system's success, because web applications are constantly being integrated with other systems and incorporating new technologies.
Design	Modified	Modified to include the design data layer activity. It is not clear in the definition of XP where this important activity is addressed therefore we decide to include it here.
Web Navigation and Presentation	Inserted	Introduced due to the importance of navigation and presentation in web applications. Web user interfaces can be complex including graphics, sound, animations, etc. It is also common to have different navigation paths that can be followed by users.
Web Testing	Inserted	Introduced to contemplate extensive testing. In web application, testing some requirements, especially non-functional, is fundamental. It is essential to verify issues like performance, network load, number of users, etc.
Web Support	Inserted	This discipline focus is to deal with the organization of the hardware and software components that form the website. In web applications there are many distinct components (hypertext, figures, code, database, etc.) that can be distributed along the network. Therefore, it is important to have a good organization of those components in order to make corrections and updates easy.

Figure 2 also shows how the web navigation and presentation discipline is detailed using a static view to illustrate the structure and relationship of its elements. The discipline includes three roles: programmer, architect and web designer. The elements navigation design and web user interface design are SPEM stereotypes called "work definition" representing a set of activities. The web design techniques and guidelines map to the "guidance" stereotype in SPEM and are used to provide assistance in the software process. The elements presented in the right side of Figure 2 represent two kinds of artifacts called "work product" in SPEM. The one that is represented by a notepad relates to a document while the other (triangle, circle and square) refers to a model.

The web designer, a new role added to XP, realizes navigation design activities assisted by the programmer and architect roles. To produce the navigation design artifact, some techniques like OOHDM [9], OOWS [10] and W2000 [22] can be used to provide guidance on how to perform these tasks in a structured way.

Other important set of activities relate to the design of the web user interface. The web designer is also responsible for this and creates the elements of the website content, such as hypertext, sound, animation, etc. Some guidelines for doing attractive web designs can be followed to improve the appearance and organization of the web content. In the case of web applications an attractive design can make the difference of whether the website will reach success alongside visitors.

All disciplines shown in Figure 1 were modeled using class diagrams with SPEM stereotypes to describe how the elements relate to each other, simplifying the understanding of what artifacts are produced or consumed by each activity and what role to perform or assist in an activity. Due to lack of space, only one discipline is presented in Figure 2. The complete process is available in [2].

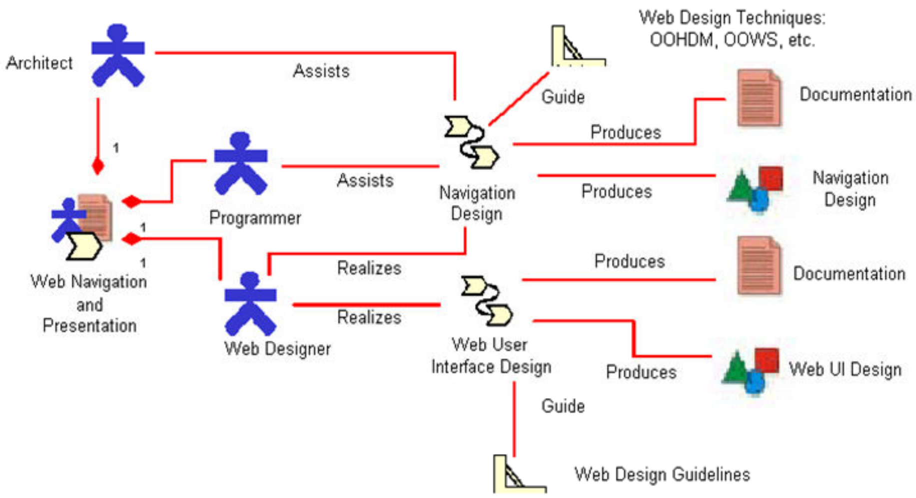

Fig. 2. Web Navigation and Presentation

4 Experimental Results

XWebProcess and Extreme Programming were evaluated in the context of an empirical study assessing the effort[1] spent in both processes to develop the same web project. The experiment was designed to demonstrate that XWebProcess retains the agile properties of XP despite the extensions performed on XP to enhance web application development quality. The experiment was planned according to guidelines specified in [11,12]. Data about the effort spent in each discipline and artifact were collected for both processes and compared. Furthermore, a survey was realized to obtain qualitative feedback about the capabilities, usefulness and agility of both

[1] Effort was measured by each group of students in person-hour while the project was implemented following the process disciplines. The goal was to collect data about the total effort spent in person-hour for each process.

processes in helping to produce high quality web applications. For more details about the experiment and the survey see [2,24]

4.1 Quantitative Analysis

The project upon which the study was done consisted of the implementation of a mid size web application. The same application was implemented by eight teams of five programmers each. All team members involved in the experiment were senior year students at UFPE (a top 3 software engineering research institution in Latin America) with at least three years of exposure to programming. The project was also made a formal coursework for the software engineering's course grade. The description of the experiment is summarized below.

Object of Study: XP and XWebProcess.

Purpose: Evaluate the use of both processes in web application development.

Quality Focus: Effort spent.

Perspective: Researchers.

Context: The experiment was performed by forty undergraduate students as subjects, divided in eight teams of five programmers. Four teams used XWebProcess and the other four used XP to develop the same web system. All teams have a previous training in XP and JUnit and the XP teams attended classes of XWebProcess only after finishing the experiment.

The t-test was used to perform the statistical analysis as defined in [12]. The analytical goal relates to verifying the null hypothesis (NH) stated in equation (1). The NH hypothesis assumes that the effort spent to develop a web system using both processes is the same. This can be verified by calculating the t statistic as described in [12]. The Alternative hypothesis (AH) is assumed to be true if NH is refuted and in this case indicates that the efforts are not the same.

NH: $\mu_x = \mu_y$, i.e. the expected mean values of the effort spent are the same.

AH: $\mu_x \neq \mu_y$: Reject NH if $|t| > t_{\alpha/2,f}$ where f is the degree of freedom and α is the significance level. $t_{\alpha/2,f}$ is obtained from a statistical table. **(1)**

The x value represent the effort spent in each of the four XP teams (T1,T2, T3, T4) and the y value represent the effort spent in XWebProcess teams (T5, T6, T7, T8). After calculating the t value we want to confirm that the alternative hypothesis does not hold. This is summarized in equation (1).

We performed the analysis using one of the Excel´s data analysis tools for the t-test. The main results obtained are shown in Table 3. All the results obtained are described in [2].

The results show that $t = -0.8954$. The value $t_{\alpha/2,f}$ can be seen in statistical tables [12] and in this case represents the critical value for the two-tailed t-test ($t_{\alpha/2,f} = 2.446$ for $\alpha = 5\%$ and $f = 6$). Therefore $|t| < t_{\alpha/2,f}$ and the null hypothesis (NH) is true. This result means that the effort spent in both processes was the same proving what we assumed before.

Table 3. Data Analysis

GENERAL RESULTS (Effort in person-hour)		
	XP Teams	**XWebProcess Teams**
Mean	104.175	113.9125
Variance	169.8617	303.1819
Observations	4	4
T-TEST RESULTS		
t stat	-0.89542	
t Critical two-tail ($t_{\alpha/2,f}$)	2.446914	
Conclusion	$\|t\| < t_{\alpha/2,f,}$ therefore NH is true	

It is important to observe that in absolute values the effort spent by XWebProcess teams were higher than in XP teams. This can be seen by the mean values in Table 3 (104.1 in XP and 113.9 in XWebProcess). The higher number of process elements such as disciplines, activities and artifacts in XWebProcess demands more effort, what explains these results. However, the statistical analysis performed by the t-test shows that this effort is not significant and both processes can be considered equal in terms of effort. The next section discusses the advantages the added elements bring in terms of quality.

4.2 Qualitative analysis

Students involved in the experiment were asked to answer a survey to assess quality factors about XWebProcess and XP advantages and disadvantages regarding web development. The aim was to compare XWebProcess with XP in order to check if the disciplines added to XP provided effective benefits for web development. In addition, the students were also asked to comment about XWebProcess description with SPEM. The key qualitative results surfacing from these questions are presented below. For a detailed reference of the results see [2,24].

- All subjects involved in the experiment rated XWebProcess more suitable for web development than XP. In particular, web user interface and navigation design disciplines promoted the development of a better website in terms of organization and appearance (***Usability product quality factor***). Moreover the use of prototyping sessions benefited requirements gathering and improved communication with clients;
- All subjects reported that the web testing discipline of XWebProcess has a positive impact on enforcing time, load and security constraints even agreeing that this could not be seen in the project realized due to its simplicity. They mentioned that in large web projects testing for high concurrency, heavy network loads and different types of clients (browsers, OS and hardware) is crucial (***Reliability product quality factor***);
- All subjects using XWebProcess reported that despite having to cope with more activities and artifacts than XP there was a pay off in the quality of the outcome mainly the web application interface, navigability and also testing;

- 95% of the subjects using XP agreed that it needs to be adapted and tailored to web development as done in XWebProcess;
- All subjects reported that the SPEM model helped in understanding the process structure (***Easiness to understand and visibility*** process quality factors). The students said that the description using activity diagrams enables a clear view of the process progress simplifying the task of knowing what and when to do an activity and also what are the inputs and outputs for each activity;
- Most students (90%) believed that the process model also facilitates the task of adapting the process to further organizational or project requirements (***Easiness to evolve*** process quality factor), and some of them even suggested some modifications related to supporting disciplines related to project management and configuration management.

The results described above provide evidence that the disciplines added to XWebProcess were effective in facilitating the delivery of a better quality solution to web application development. The benefits are not only related to product quality factors such as usability and reliability but also related to process quality factors such as easiness to understand, enact and evolve the process. Complete details about the measures of each quality factor can be found in [2,24] and were not included here due to lack of space. Some threats to the validity of the experiment were:

- Some students had no previous experience with the web technology chosen to implement the project (JSP) but efforts were made by us to choose the most homogenous teams as possible;
- The students had no previous practical experience with neither XP nor XWebProcess. To minimize this problem all students learned about their corresponding process before starting the experiment by attending classes. The students that stayed in XP groups only attended XWebProcess classes after finishing the experiment, to avoid being influenced, and before answering the survey.

5 Summary and Further Work

In this paper we have described the initial step of an ongoing research effort towards reconciling quality and speed in web application development. The approach adopted defines an experimental spectrum that departs from an agile software process (XP) and gradually incorporates quality related software disciplines to improve software quality without sacrificing agility.

Although the experiments and surveys conducted gave important feedback on the impact of the disciplines added to XP towards improving quality in web application development, there are still challenging issues that need to be further explored in the quest for finding the optimal balance between speed and quality. Some of the noteworthy issues are:

- What other disciplines can be added to improve web application quality without sacrificing agility, i.e., how far in the direction of prescriptive methods can we move;
- Given the current set of disciplines, how do speed and quality indicators perform, when confronted with mid and large scale web application projects (e.g., more than 50000 lines of code);
- What disciplines can have a cross domain impact, generalizing to a best practice status;

We are also planning experiments that will confront XWebProcess and RUP for the same project. This will also give important insights on the quality/speed trade-off at the prescriptive end of the spectrum.

Acknowledgements

Américo thanks the Brazilian National Research Council (CNPq) for supporting this research.
Pedro thanks the Starting Lecturer Scheme from UMIST for partially supporting this research.

References

[1] Beck, K. Extreme Programming Explained. Addison-Wesley, 1999.
[2] Sampaio, A. XWebProcess: Um processo ágil para o desenvolvimento de aplicações Web. Msc by Research Dissertation, 2004, Universidade Federal de Pernambuco.
[3] Pressman, R.S. What a Tangled Web We Have. IEEE Software, January – February 2000, pp 18-21.
[4] Object Management Group. Software Process Engineering Metamodel Specification. Version 1.0, November 2002.
[5]Heinecke, H., Noack, C., and Schweizer, D. Constructing Agile Software Processes. In Proceedings of Extreme Programming Conference (XP2002), Alghero, Sardinia, Italy 2002.
[6] Object-oriented Process, Environment and Notation (OPEN) Web site. Available at: http://www.open.org.au/
[7] McDonald, A., and Welland, R. Agile Web Engineering (AWE) Process. Department of Computing Science Technical Report TR-2001-98, University of Glasgow, 2001.
[8] Rumpe, B, and Schröder, A. Quantitative Survey on Extreme Programming Projects. In Third International Conference on Extreme Programming and Flexible Processes in Software Engineering, XP2002, May 26-30, Alghero, Italy, pp. 95-100, 2002.
[9] Schwabe, D. and Rossi, G. The Object-Oriented Hypermedia Design Model. In Communications of the ACM, volume 38(8), pages 45-46, 1995.
[10] Pastor, O., Fons, J., Abrahão, S., and Rámon, S. Object Oriented Conceptual Models for WEB Applications. In 4th Iberoamerican Workshop on Requirements Engineering and Software Environments (lIDEAS´2001), San Jose, Costa Rica, 2001.
[11] Pfleeger, S. L. Design and Analysis in Software Engineering – Part1: The Language of Case Studies and Formal Experiments. Software Engineering Notes, vol 19, no 1, October 1994, pages 16-20.

[12] Wohlin, C., et al. Experimentation in Software Engineering: An Introduction. Kluwer Academic Puslishers, 2000.
[13] Sommerville, I., Software Engineering, 6th Edition, Addison Wesley, 2003
[14] Henderson-Sellers, B., Lowe, D., and Haire, B., OPEN Process Support for Web Development, Annals of software Engineering, vol. 13, pp 163-202, 2002.
[15] Lowe, D., and Eklund, J., Client Needs and the Design Process in Web Projects, Journal of Web Engineering, vol. 1, p. 23-36, 2002, Rinton Press.
[16] Wallace, D., Raggett, I., and Aufgang, J., Extreme Programming for Web Projects (XP Series), Addison-Wesley, 2002.
[17] Sampaio, A., Vasconcelos, A., and Sampaio, Pedro R. Falcone, XWebProcess: Agile Software Development for Web Applications, Accepted Poster Presentation to appear in International Conference on Web Engineering – ICWE04, Munich Germany, 2004.
[18] The Challenges of Complex IT Projects, Report presented by The Royal Academy of Engineering and The British Computer Society in April 2004.
[19] Proceedings of the 7Th European Conference on Software Quality, Helsinki, Finland, 9-12 June 2002.
[20] Rombach, Dieter. The Dilemma of Software Development, Quality Connection, May 2002.
[21] Kruchten, Philippe. The Rational Unified Process: An Introduction, Addison-Wesley, 2003.
[22] Baresi, L., Garzotto, F., and Paolini, P., Extending UML for Modeling Web Applications, 34th Hawaii International Conference on System Sciences, USA, 2001.
[23] ISO 9126 standard, available at http://www.iso.org/iso/en/ISOOnline.openerpage
[24] Sampaio, Américo, The Official XWebProcess Website, Available at: http://www.cin.ufpe.br/~atfs/xwebprocess