

A Theoretical Agile Process Framework for Web Applications Development in Small Software Firms

Haroon Altarawneh
Al-balqa Applied University
haroon@teacher.com

Asim El Shiekh
Arab Academy for banking and financial sciences
a.elshiekh@aabfs.org

Abstract

As web based-applications become increasingly important to all aspects of life, there is a need to encourage practitioners to adopt best practices so as to improve the quality of the processes in use, and therefore achieve targets related to time, budget and quality. The web development industry worldwide is dominated by a myriad of small firms. This presents a challenge in terms of determining the current practices of industry participants, and in devising improvement initiatives which are feasible for small firms. The objectives of this research is to propose a software process model framework that assists the small firms in designing web based-applications so as not to run over budget and time. To achieve these objectives, an extensive literature survey of web development methodologies and software process improvement in small firms were conducted to identify web engineering practice in web development. The literature survey clarified that, there were several development methodologies of web-based application. These studies have not taken into consideration software process improvements. A software process framework for internet development in small firms is proposed in this research. This framework contains several steps to follow; the first step is to begin with small web project. The second steps is adopt the modified XP process model which is also a suggested process model for internet development in small firms, and then apply XXPM as a quality model for software process improvement. If the proposed software process framework is to be successful then it must be laid on web engineering practices.

1. Introduction

Web-based applications are becoming so popular in our daily life in the sense that it would not go a single day without we use them. These applications range from simple to sophisticated ones, where millions of

dollars in revenue are generated. Developing, testing and quality assuring these applications become a challenging task [1]. Although the development of web-based applications made many improvements, there is still a lack of an established software engineering methodology for constructing web-based applications [2]. A number of software process frameworks for web application development have been introduced to solve various problems associated with the development of web applications, but they all need a lot of resources like number of developers available, the quality of tools and equipments, and Quality Assurance (QA) team [3], which are limited in Small-size software companies.

The software process is becoming a major concern in most software development organizations as one of ways to assure the software quality while developing software system, with the software process [4], there are still questions on whether we perform the process in a right way and how we evaluate the level of conformance of the process. There are two battles over process that every small software company must win to be successful. The first is the battle to convince the company to adopt reasonable development processes. Discussion of what makes up a good process may be an interesting meditation, but is entirely moot until the company commits to a policy of process improvement. The second battle is never over. It is to change existing processes to match changing circumstance [5].

Small-size software companies, in general and internet-based in particular, have very stiff and stringent conditions. They have limited resources in the number of developers available, the quality of tools and equipments [7].

2. Literature Review

The literature review is divided into three sections, Web-based application development, Software process improvement, and Web-based application development methodologies.

2.1 Web-based Application Development

Web development methodologies history identifies and explores eras of development and speculates on their future. Today's post methodology era involves methodologies that can be viewed by developers as outdated and inappropriate for rapid development, web applications, and other current requirements [8]. "There are very few standard methods for the web developers to use. Hence, there is a strong need to understand and undertake web engineering" [9]. Ad-hoc development of WBA has brought disasters to many organizations. A survey on web based project development by the Cutter Consortium [10] highlighted problems for Web-based projects:

- * Delivered systems didn't meet business needs 84 percent of the time.
- * Schedule delays plagued the projects 79 percent of the time.
- * Projects exceeded the budget 63 percent of the time.
- * Delivered systems didn't have the required functionality 53 percent of the time.
- * Deliverables were of poor quality 52 percent of time.

It is indeed important, to examine web engineering to determine what is already accomplished, what is not yet done, and how to fill the gap. For a long time, IS researchers examined issues related to systems development (including transaction systems, decision support systems) and built a cumulative body of research. It is, indeed, an opportunity for researchers interested in Web engineering to capitalize on prior research to examine Web Engineering critical issues [11].

2.2 Software Process Improvement

Hans and others [12] found that the growth of the software industry has produced many small companies that do not do contract software, but rather compete in other areas. This gives rise to at least four significant development issues that have not been adequately addressed in software engineering literature: company size, development mode, development size, and development speed. They discuss these issues and then discuss some of the shortcomings of current software engineering thinking for small companies. Definitions of "small" businesses vary by industry and by government agency from 100 to 500 employees or more. They define companies of 50 or fewer employees as small.

First step toward process improvement is identifying the strengths and weaknesses of an organization's software processes to determine effective improvement

actions. An assessment can help an organization examine its processes against a reference model to determine the processes' capability or the organization's maturity, to meet quality, cost, and schedule goals. Many small organizations are unaware of existing software process assessment models and standards. A study on 15 small software companies showed that, none of the companies were using a process model in a "text-book" fashion[13], choosing instead either to drop elements of their chosen model or, develop something proprietary instead.

2.3 Web-based Application Development Methodologies

Most of web application development methodologies used these days are extensions of standard software engineering methodologies. The usual iterated waterfall model is too rigid an approach to developing web applications [14]. The waterfall model process was perfect for developing a file maintenance program for mainframes, but far too restrictive a process for building a Web application. Web application development needs to be an iterative process and most agree that a spiral approach is best. But, the exact steps at each cycle of the spiral are debated, as is the metric to be used to determine the completion of a cycle. Web application development is definitely component-oriented and most believe that the appropriate process should be object-oriented.

An agile approach [15] for web application development have been proposed that applies the concept of agile modeling, adopts a standard software architecture and is heavily based on frameworks, speeding up system analysis, design and implementation.

3. Literature Analysis

In the last couple of years, one of the major trends for software development organizations was the move towards web application systems. Their processes usually are immature and ad-hoc. Often this is coupled with a less than positive attitude towards software engineering practices and, especially, software process improvement initiatives and software metrics collection. In particular, code metrics (such as lines of code, code complexity etc.) and process improvements standards (such as the Capability Maturity Model) are often viewed as obsolete and assessing processes and metrics.

The literature shows that, the majority of web development methodologies do not consider the size of the development firm, and most of the development

methodologies require a lot of resources (like skilled process engineers, staff ,etc) which are not available in small firms. The literature also shows that, a number of methodologies have emerged in the last few years to develop web based applications. We want a method that is simpler than those currently available and thus easier to use in small firms.

It is indeed important, to examine web engineering and software process improvement to determine what is already accomplished, what is not yet done, and how to fill the gap. Figure (1) shows the gap.

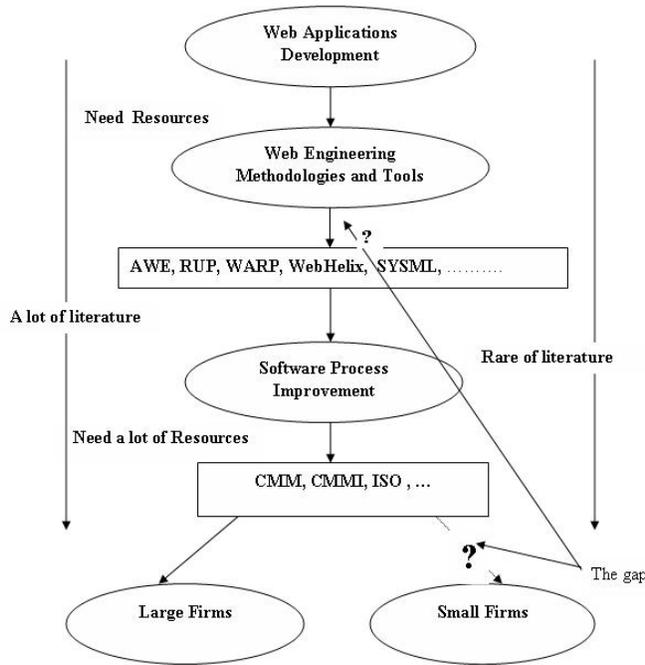


Figure 1.4: Literature Related to Web Development and Software Process Improvement.

4. Agile Development Methods

The field of software development is not shy of introducing new methodologies. Indeed, in the last 25 years, a large number of different approaches to software development have been introduced, of which only few have survived to be used today.

The term agile has recently been used to categorize a number of lightweight approaches to building software. These include: Extreme Programming (XP), Adaptive Software Development and Dynamic Systems Development Methodology (DSDM) [16]. AWE is an iterative and incremental process, researchers believe this will allow for: early and

continuous delivery of valuable software; the ability to harness changing requirements, even late in development; and the delivery of working software frequently. The AWE Process supports multidisciplinary development treating business experts, domain experts, and creative designers as developers along side software engineers [17]. Beck [16] believes that the developers and organizations involved in web engineering projects are the primary factor in the success or failure of web application development. Given the diversity of disciplines required to develop Web-based applications. Fowler [17] believes that people are the most important factor in project success is the fundamental reason why we have not tried to develop a monumental process to tackle the problems associated with web application development.

4.1 Extreme Programming

Extreme Programming (XP) has evolved from the problems caused by the long development cycles of traditional development models. It first started as “simply an opportunity to get the job done” [18], with practices that had been found effective in software development processes during the preceding decades. After a number of successful trials in practice, the XP methodology was "theorized" on the key principles and practices used. Even though the individual practices of XP are not new as such, in XP they have been collected and lined up to function with each other in a novel way thus forming a new methodology for software development.

4.2 Why XP

Different software development methodologies and quality assurance methods are used in order to attain high quality, reliable, and bug free software. Extreme Programming (XP) is a software development methodology that integrates many of the known ideas (that we all were familiar with) in order to achieve such software systems. Specifically, XP emphasizes code-unit testing (preferably before its writing), and thorough testing of software functionality. The contribution of XP to software development is expressed, among other ways, in the quality improvement of both the entire process of software development and of the software quality itself. Currently, XP is used mainly in small-medium size software projects [19].

High software quality is one of the main principles that guide any software development methodology. This is also correct with respect to XP. For this aim,

XP defines two levels of automatic testing [20]. The first is unit testing, which must be coded by the programmers, preferably right before coding a given feature. In contrast to the traditional software development methods, where the quality control entity checks the software quality and provides feedback to the programmers [21].

We have used the XP model for particular purposes that enhance the quality of the development process. Thus, the reasons that stand behind using such a model are:

- 1- several researchers have advised using such a model.
- 2- XP is an easy model for learning.
- 3- XP is one of the most convenient models for web applications development among agile methods.
- 4- XP could be easily adapted with requirements changing.
- 5- XP achieves software process improvement better than agile methods; it conforms to level two in CMMI quality model.
- 6- The literature showed that extreme programming is the lightweight process model that can help small firms in the implementation of software process improvement.

4.3 Limitations of XP

Agile processes are intended to support early and quick production of working code. This is accomplished by structuring the development process into iterations, where an iteration focuses on delivering working code and other artifacts that provide value to the customer and, secondarily, to the project[22]. Agile process proponents and critics often emphasize the code focus of these processes.

Proponents often argue that code is the only deliverable that matters, and marginalize the role of analysis and design models and documentation in software creation and evolution. Agile process critics point out that the emphasis on code can lead to corporate memory loss because there is little emphasis on producing good documentation and models to support software creation and evolution of web systems.

5. Proposed Framework

Our proposition is to modify the XP life cycle and to add a lightweight project management activity at the beginning of a project. That phase does not have to be very long. A few weeks would be enough. Project management in XP is limited, the planning perspective is very short: not longer than two month (one release). Some people consider this a disadvantage. Time and

money could be saved if the stakeholders (the customer representatives and the developers) spend, at the beginning of a project, a week or two discussing high-level goals, constraints, scope, risks and feasibility.

Pressman [23] warns, "There is a tendency to rush to a solution, even before the problem is understood. This often leads to elegant software that solves the wrong problem".

Project management is a necessity for big projects that involve large teams perhaps working separately on different parts of the project. Project management can help even on the smallest project [24]. Figure 2 shows the framework):

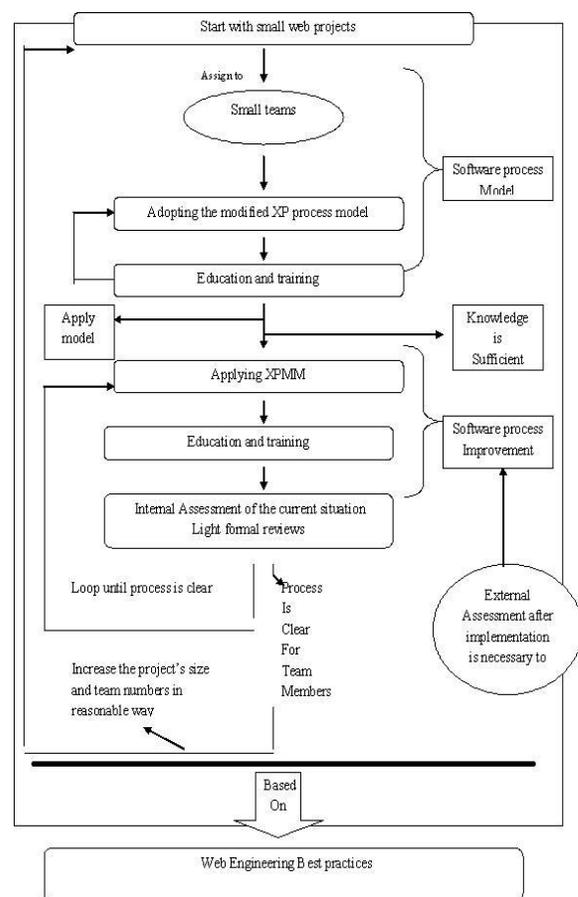


Figure 2: Proposed Framework

6. The Framework processes

Figure (2) shows the main steps in the framework. The framework consists of the following steps:

- 1- Start with small web projects.
- 2- Adopt the modified XP process model.
- 3- Apply XPMM.
- 4- Education and training.

5- Internal assessment and light formal reviews
6- External assessment
And all of the over processes must be implemented with web engineering best practices.

6.1 Start with Small Web Projects

The surveys and literature stated that most of web development projects in small firms are small or medium size projects (1-2 months or 3-6 months). In small web projects it is highly desirable to plan all phases at the start of the project. This means that the phase sections of the web project management plan, web configuration management plan and web verification and validation plans are combined. The plans may be generated when writing the proposal.

Typically, qualified developers are involved in large or medium-size projects whereas small web projects are carried out by under-qualified or inexperienced web developers. The reactionary development scenario and the lack of clear guidelines to face the process, push developers to follow an ad-hoc development process.

The heavyweight software methodologies are limited to support such scenario. This is because they involve several stages and roles that require an important amount of communication and coordination in order to get a final product.

6.2 Adopt the Modified XP Process Model

In the modified XP we have proposed the XP lightweight project management phase, and it consists of the following phases: Requirement management, risk management and, quality assurance management.

The aim of the XP lightweight project management phase is to align between the methodology and tools of project management and XP methodology to achieve software process improvement with little efforts, resources, and cost. It is common to talk about XP method for modern project management processes in the context of a set of lightweight activities used to manage the development or acquisition of web applications. These activities include requirements, design, coding, and testing processes based on a minimal set of activities needed to reach the end goal.

Although XP development method addresses the management aspects of software projects – people, processes, and technology – they are primarily focused on coding, testing, and software artifact delivery.

If lightweight project management method were properly applied, in the proper domain, to the proper set of problems, with properly trained participants to this framework, software process improvement would

be achieved. The deployment of Lightweight project management methodology in an existing organization faces several obstacles:

1- The legacy project management processes must be displaced in some way to make room for the new process.

2- The gaps that existed in the legacy process must be filled with the new process while maintaining the integrity provided by the legacy process.

6.2.1 Requirement Management

The requirement management process focuses on the multiple customer representatives. In our modified XP process, model the requirement management process work as follow:

1- Setup plans for the project releases: in the original XP, the planning perspective is very short. Time and money could be saved if the stakeholders (the customer representatives and the developers) spend, at the beginning of a project, a week or two discussing high-level goals, constraints, scope, risks and feasibility. During that phase the XP team could do the following things:

- Collect the use scenarios from multiple perspectives: Scenarios concerning the current system are collected from the customer's representatives and they describe problems and motivations for building a new system. This is a simple form of feasibility study. This is also a good starting point for collecting domain knowledge and understanding the problem. Scenarios concerning a new system can be written by the customer representatives and/or by the XP team. They show the vision how the system could work
- Assess system feasibility: One of the questions that should be answered is if it is the right decision to develop the system according to the XP methodology.
- Identify system stakeholders.
- Roughly describe the system's operating environment.
- Look for main domain constraints.
- Prototype poorly understood requirements. At this stage the prototype can be simple and should mainly focus on human-machine communication.
- Organize the requirements into multiple layers
- Plan for conflicts and conflict resolution: Conflicts are inevitable if a system is to serve many people with different expectations and fears.

- Make the collected requirements available to the team and to the customer.
- Carefully choose attributes for your requirements. The set of attributes should be as small as possible. Then it is much easier to find information and maintain the attribute values.

2- Manage changes to the requirements as they evolve during the project.

3- Maintain bi-directional traceability among the requirements and the project plans and work products.

6.2.2 Risk Analysis

XP practices address risks common to web development projects at early stage of the project lifecycle. By decomposing the web project into multiple small releases where each release of the web application contains only a subset of the required functionality. These Small Releases are incremental production versions of the project's expected final deliverable providing limited subsets of functionality to the system's users. Each release of additional functionality offers stakeholders an opportunity to use the evolving capabilities of the software and provide high quality feedback, thereby improving the quality of progressive elaboration. Extreme Programming acts on the observation that smaller projects have a higher success rate. It decomposes all software projects into multiple small releases where each release of the software contains only a subset of the required functionality.

6.3.3 Quality Assurance Management

The purpose of web applications quality assurance is to provide management with appropriate visibility into the process being used by the web project and of the products being built. Web applications quality assurance involves reviewing and auditing the web applications products and activities to verify that they comply with the applicable procedures and standards and providing the software project and other appropriate managers with the results of these reviews and audits.

The web applications quality assurance group works with the web project during its early stages to establish plans, standards, and procedures that will add value to the web project and satisfy the constraints of the project and the organization's policies.

6.4 The XPMM

XPMM [25] consists of 4 levels and its structure resembles CMM (I). The highest maturity level is connected with the team performance (no overtime, all code must pass all unit tests, the customer must be satisfied). The model should help to distinguish between "real" XP projects and pseudo-XP projects, which have only one common characteristic with XP, namely lack of written documentation.

6.5 Education and Training

Good web engineering practice requires expertise in a complex set of activities that involve the intellectual skills of planning, designing, evaluating, and revising. A web engineering process must take into account the different types of developer required to build a successful solution. In order to this framework to be successful, all of the people involved in software development (developers, managers,) must have a good knowledge in web engineering development and they must be trained. The best way to learn modified XP is in an experiential-learning training course. Your entire development team (including the testers, the customer, and the manager) should attend a one-week immersion course on XP.

6.6 Internal Assessment and Light Formal Reviews

An internal assessment is necessary to assure that project team learns how to implement the modified XP model and XPMM. Light formal reviews could be done as follow:

- 1- The web project manager reviews the process progress.
- 2- The web project manager reviews the process with the teams' leaders.
- 3- The team members review their work with each others.

6.7 External Assessment

External assessment is necessary to assure the quality, and this can be done by another quality assurance company. Benefits of External assessment:

- 1-Early warning-system for problems
- 2-Measure of product quality
- 3-Indicator of where to direct improvement efforts
- 4-Monitor of changes in technology and web practices.

6.8 Web Engineering Practices

Many practitioners in the field of Web engineering and software engineering have commented on the lack of suitable software engineering processes that can be used to build Web applications. We investigated the way industrial Web engineering is being carried out by making a survey consists of questions relating to the development process being used to develop Web applications and web engineering practices. If a Web engineering process is to be successful then it must address the following:

1. Short development life-cycle times.
2. Delivery of bespoke solutions.
3. Multidisciplinary development teams.
4. Small development teams working in parallel on similar tasks.
5. Analysis and Evaluation.
6. Requirements and Testing.
7. Maintenance.

7. Framework Limitations

- 1- This framework needs knowledge in XP.
- 2- Adoption of this framework by small firms needs along time (change process required time)
- 3- This framework is a theoretical framework.

8. Future directions in research

This work will open many possibilities for further research in the field of web engineering and process models for web-based applications development.

This research could be extended to investigate the impact of culture on adoption of this framework and software process improvement models. Future research is needed to explore the problems and limitations related to web development in SMEs (Small and Medium Enterprises).

9. Conclusion

Modified XP codifies a set of practices that many web developers are willing to adopt in both action and spirit. Many of these practices are grounded in fundamental project management theory. When software development teams embrace the practices of Extreme Programming an opportunity is created for a broad set of project management practices to become meaningful and accessible to the developers, while at the same time making clear, unambiguous information available to the project managers. To ease the project-

management burden, modified XP includes practices aimed at reducing management overhead, while keeping the customer's interest at close range. The literature and surveys in this research revealed many problems in web based application development in small firms. The ability of this framework to solve the problems could be inherent in the following points (See Table 1):

Table 1: Problems solved by the framework

Problem	Solution
Ad hoc Development and few standards to use.	Proposed framework offers to small firms a standard approach to web development
Software process improvement in small firms	Applying XPMM with low cost, less resources and little efforts.
Project management issues	Applying the modified XP model
Development practices(Organizational Issues, Standards and Procedures, Metrics, Control of the development process	Applying web engineering practices and XPMM within the new framework
The majority of web development methodologies not consider the size of the development firm, and most of the development methodologies required a lot of resources (like skilled process engineers, staff... etc) which are not available in small firms.	Proposed framework offers to small firms a standard approach to web development
Basic knowledge of web applications development	Proposed framework is easy to learn
In adopting a software process model, many small software companies are ignoring standard process models and models for process improvement	The framework is a standard model
Project management issues	The framework adds a lightweight project management activities
A common weakness of all of the SPI methods identified in the literature is that they do not identify specific best practices within the software domain	The model is based on web engineering practices, which offers a specific best practices within the software domain

10. References

- [1] Abdesselam Redouane ,” Guidelines for Improving the Development of Web-Based Applications” *Proceedings of the Fourth International Workshop on Web Site Evolution (WSE'02)* 0-7695-1804-4/02 2002 IEEE
- [2] Said Hadjerrouit ,“Web-based Application Development:A Software Engineering Approach “*ACM SIGCSE Bulletin June 2001* Vol 33. No. 2 p 31-34
- [3] Ahmed E. Hassan and Richard C. Holt “Migrating Web Frameworks Using Water Transformations” ,*Proceedings of the 27th Annual International Computer Software and Applications Conference (COMPSAC'03)* 0730-3157/03 © 2003 IEEE
- [4] So-Young Kim and Ho-Jin Choi, “An Evaluation of Process Performance for a Small-Team Project -A Case Study”, *Proceedings of the Fourth Annual ACIS*

International Conference on Computer and Information Science (ICIS'05)
0-7695-2296-3/05 © 2005 IEEE

[5] Robert P. Ward, Mohamed E. Fayad, and Mauri Laitinen
“Software Process Improvement in the Small”
“, *COMMUNICATIONS OF THE ACM April 2001*/Vol. 44,
No. 4 pp 105-107

[6] Abdesselam Redouane, “Towards a New Method For The
Development Of Web-Based Applications”, *Proceedings of
the Third IEEE International Conference on Cognitive
Informatics (ICCI'04)* 0-7695-2190-8/04 © 2004 IEEE

[7] Kathleen Coleman Dangle, Patricia Larsen, and Michele
Shaw, “Software Process Improvement in Small
Organizations: A Case Study”, *IEEE software
November/December 2005 (Vol. 22, No. 6)* pp. 68-7513.

[8] David E. Avison, Guy Fitzgerald, “Where Now for
Development methodologies”, *COMMUNICATIONS OF
THE ACM January 2003*/Vol. 46, No. 1 pp 79-82

[9] Yogesh Deshpande, Martin Gaedke "Web Engineering:
Developing Successful Web Applications In A Systematic
Way", *14th International World Wide Web Conference, 10-
14 May, 2005, Chiba, Jap*

[10] Cutter Consortium *Research Briefs*. 7 Nov. 2000.;
<http://www.cutter.com>

[11] Pressman, R.S. *Software Engineering: A Practitioner's
Perspective*, 5th edition, McGraw-Hill, New York, 2000.

[12] Hans-w, Gellersen and Martin Gaedke, “
OBJECTORIENTED WEB APPLICATION
DEVELOPMENT”, JANUARY - FEBRUARY 1999
<http://computer.org/internet/1089-7801/99> ©1999 IEEE
INTERNET COMPUTING

[13] Christiane Gresse von Wangenheim, Alessandra
Anacleto, Clênio F. Salviano, “Helping Small Companies
Assess Software Processes”, January / February 2006
IEEE SOFTWARE, pp 91-98

[14] Andrew McDonald and Ray Welland, “Evaluation of
Commercial Web Engineering Processes”, 2002 available at :
<http://www.dcs.gla.ac.uk/> accessed 20/5/2006

[15] Vitor Estevao Silva Souza and Ricardo de Almeida
Falbo, “An Agile Approach for Web Systems Engineering”,
*WebMedia'05, December 5-7, 2005, Poços de Caldas,
Minas Gerais, Brazil*. Copyright 2004 ACM 1-58113-000-
0/00/0004

[16] Beck K. et al., 'Manifesto for Agile Software
Development', *The Agile Alliance*, February 2001,
<http://www.agilealliance.org/>

[17] Fowler M. & Highsmith J., 'The Agile Manifesto',
Software Development Magazine, August 2001,
<http://www.sdmagazine.com/documents/s=844/sdm0108a/0108a.htm>

[18] Beck, K. *Extreme Programming Explained*, Addison-
Wesley, 1999.

[19] C. Keith Ray, “adopting xp The path to—and pitfalls
of—implementing Extreme Programming”, *STQE*, pp : 34-
40, JULY/AUGUST 2002 www.stqemagazine.com

[20] Limitations of Agile Software Processes
<http://www.agilealliance.org/system/article/file/1096/file.pdf>

[21] I. Sommerville, and P. Sawyer, *Requirements
Engineering: A Good Practice Guide*, John Wiley & Sons,
Chichester, 1997

[22] NAWROCKI Jerzy R. and WALTER Bartosz, “Toward
Maturity Model for eXtreme Programming”, *27th Euromicro
Conference 2001: A Net Odyssey (euromicro'01)* p. 0233

[23] Pressman, R S '*Software Engineering: a Practitioner's
Approach* (4th ed.)', McGraw-Hill Publishing Company 1997.

[24] Crispin, L. (2001). Is Quality Negotiable?, *XP Universe
Conference*.

[25] NAWROCKI Jerzy R. and WALTER Bartosz, “Toward
Maturity Model for eXtreme Programming”, *27th Euromicro
Conference 2001: A Net Odyssey (euromicro'01)* p. 0233