

UPGRADE is the European Online Magazine for the Information Technology Professional, published bimonthly at <http://www.upgrade-cepis.org/>

Publisher

UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by Novática (<http://www.ati.es/novatica/>) and Informatik/Informatique (<http://www.svifsi.ch/revue/>)

Chief Editors

François Louis Nicolet, Zurich <nicolet@acm.org>
Rafael Fernández Calvo, Madrid <rfoalvo@ati.es>

Editorial Board

Prof. Wolffried Stucky, CEPIS President
Gloria Nistal Rosique and
Rafael Fernández Calvo, ATI
Prof. Carl August Zehnder and François Louis Nicolet, SVI/FSI

English Editors: Mike Andersson, Richard Butchart, David Cash, Arthur Cook, Tracey Darch, Laura Davies, Nick Dunn, Rodney Fennemore, Hilary Green Roger Harris, Michael Hird, Jim Holder, Alasdair MacLeod, Pat Moody, Adam David Moss, Phil Parkin, Brian Robson

Cover page designed by Antonio Crespo Foix, © ATI 2002

Layout: Pascale Schürmann

E-mail addresses for editorial correspondence:
<nicolet@acm.org> and <rfoalvo@ati.es>

E-mail address for advertising correspondence:
<novatica@ati.es>

Copyright

© Novática and Informatik/Informatique. All rights reserved. Abstracting is permitted with credit to the source. For copying, reprint, or republication permission, write to the editors.

The opinions expressed by the authors are their exclusive responsibility.

eXtreme Programming

Guest Editor: Luis Fernández Sanz

Joint issue with NOVÁTICA and INFORMATIK/INFORMATIQUE

2 Presentation: eXtreme Programming

– *Luis Fernández Sanz, Guest Editor*

4 A new method of Software Development: eXtreme Programming

– *César F. Acebal and Juan M. Cueva Lovelle*

What is eXtreme Programming – also known as XP? The aim of this article is to answer that question, and to reveal the nature of this new method of software development to the uninitiated reader. We will try to be sufficiently informative so that you will all come away with some idea of the basic underlying principles, and for anyone who might want to delve deeper into the subject, we will provide suitable references.

9 Programming Extremism – *Michael McCormick*

The author reviews antecedents and experiences of the "agile" methodology of software development called eXtreme Programming, comparing it to other methodologies and pointing to its advantages and disadvantages from a pragmatic standpoint, depending on the kind of project it applies to. He draws the conclusion that it is necessary to stay away from "religious" positions about existing methodologies.

11 The Need for Speed: Automating Acceptance Testing in an eXtreme Programming Environment

– *Lisa Crispin, Tip House and Carol Wade (Contributor)*

How to focus acceptance testing for XP, how to design automated tests that are low-maintenance and self-verifying, how to apply the values of XP to test automation, and ways to gather metrics and provide useful reports.

18 Qualitative Studies of XP in a Medium Sized Business

– *Robert Gittins, Sian Hope and Ifor Williams*

Qualitative Research Methods are used to discover the effects of applying eXtreme Programming in a software development business environment. Problems dominating staff development, productivity and efficiency are parts of a complex human dimension uncovered in this approach. The interpretation and development of XP's "Rules and Practices" are reported, as well as the interlaced communication and human issues affecting the implementation of XP in a medium sized business.

23 XP and Software Engineering: an opinion – *Luis Fernández Sanz*

In this article, the author makes some reflections on certain specific aspects of eXtreme Programming as described in Kent Beck's book "eXtreme Programming explained. Embrace change". The analysis presented here is in relation to principles and techniques of software engineering.

27 XP in Complex Project Settings: Some Extensions

– *Martin Lippert, Stefan Roock, Henning Wolf and Heinz Züllighoven*

XP has one weakness when it comes to complex application domains or difficult situations at the customer's organization: the customer role does not reflect the different interests, skills and forces with which we are confronted in development projects. We propose splitting the customer role into a user and a client role. The user role is concerned with domain knowledge; the client role defines the strategic or business goals of a development project and controls its financial resources. It is the developers' task to integrate users and clients into a project that builds a system according to the users' requirements, while at the same time attain the goals set by the client.

Coming issue:
"Information Retrieval"

Presentation: eXtreme Programming

Luis Fernández Sanz, Guest Editor

This issue is focused on XP (eXtreme programming), one of the recent proposals in the software development field that has achieved a really important media impact among software practitioners. As a new way of improving the agility of software projects, XP relies on several principles (automated testing, pair development, etc.) that shorten the project life cycle between releases. But these principles are also devised to obtain a general improvement of software quality and user satisfaction, avoiding problems due to delays and exceeding budget. All these promises have raised a general interest in this development philosophy. Trying to satisfy the curiosity of our readers, we have decided to publish an interesting set of paper intended to contribute to a deeper understanding of XP, the pros and the cons.

“Extreme programming: a new software development method” was presented in the Sixth Software Quality and Innovation Spanish Conference (2001) organised by the Software Quality Group of ATI. In this paper, a brief description of the main characteristics and principles of XP is included. This contribution helps the reader to know the fundamentals of Extreme Programming.

“Programming extremism”, by M.McCormick, is a keen analysis of the implications of XP in the field of software processes. His observations about the two antagonist communities in software engineering (and the need of a third party of pragmatists) and how the beliefs of each one interfere in their objective perception of which is the best solution for each type of project and organisation are really interesting. The necessary reference to CMM (and other “formal” models of software process improvement and evaluation) is included.

“The Need for Speed: Automating Acceptance Testing in an Extreme Programming Environment” (L.Crispin, T.House and C.Wade) presents details of one of the more robust contributions of XP: automated testing. Instead of the usual unconcerning in testing that can be observed in the traditional software organisations, XP stresses the importance of a proper and efficient testing practice. As well as a brief review and discussion of XP testing principles (in a Q&A format), this paper presents details about a practical experience using JUnit and other “tools”. This paper was presented in the XP 2001 conference (thanks to both the negotiation of Michele Marchesi, co-chairman of the Conference, and the collaboration of the authors).

“Qualitative Studies of XP in a Medium Sized Business” is another paper from XP2001 (thanks again to Marchesi and the

authors). The paper examines the benefits of a flexible management approach to XP methodology. Presentation and discussion of results of an empirical study using qualitative research techniques (questionnaires, direct observation, etc.) are included as part of a good review of situations that emerge when people try to apply XP in a real organisation.

“XP and software engineering” presents my own analysis of XP. From the perspective of somebody who has to approach extreme programming from the “outer world”, this paper is focused on several important improvement proposals. Of course, the doubts of the author about the success of some principles of XP related to classical software engineering practices (e.g. configuration management vs common property of code) are also included.

One of the main advantages of XP is the agility of the proposed software process and its direct application to small projects with a high rate of requirements volatility. But, is XP suitable for larger or more complex projects? To answer this question, M.Lippert, S. Roock, H. Wolf and H. Züllighoven present an extension of the roles of client representatives and the creation of new document types to address the subsequent project situation.

I hope this variety of contributions would satisfy the increasing demand of information about XP of software practitioners (in general sense) and our readers as the persons who we devote our work to.

Luis Fernández Sanz received a degree in informatics engineering from Technical University of Madrid (Spain) in 1989 and a Ph. D. degree in informatics from University of the Basque Country in 1997 (as well as an extraordinary mention for his doctoral thesis). He is currently head of the department of programming and software engineering at Universidad Europea-CEES (Madrid). From 1992, he is the coordinator of the software engineering section of Novática. He is author or coauthor of several books about software engineering and software measurement, as well as different papers in international journals and conferences. He is member of the Software Quality Group of ATI and he has acted as chair of the VI Spanish Conference on Software Quality and Innovation organised by ATI. He is a member of ATI and the Computer Society of the IEEE.
<lufern@dpris.esi.uem.es>

Useful references on eXtreme Programming

Note: See also the references included in the papers published in this issue.

Books

- K. Beck: *Extreme programming. Embrace change*, Addison-Wesley, 2000.
- K. Beck & M. Fowler: *Planning extreme programming*, Addison-Wesley, 2000.
- R. Jeffries, A. Anderson, C. Hendrickson, K. Beck, R. E. Jeffries: *Extreme Programming Installed*, Addison-Wesley, 2000
- G. Succi, M. Marchesi: *Extreme Programming Examined*, Addison-Wesley, 2001.
- W. C. Wake: *Extreme Programming Explored*, Addison-Wesley, 2001.
- R. Hightower & N. Lesiecki: *Java Tools for Extreme Programming: Mastering Open Source Tools Including Ant, JUnit, and Cactus*, John Wiley & Sons, 2001.
- K. Auer, R. Miller: *Extreme Programming Applied: Playing to Win*, Addison-Wesley, 2001.
- M. Fowler, K. Beck, J. Brant, W. Opdyke & D. Orberts: *Refactoring: Improving the Design of Existing Code*, Addison-Wesley, 1999.

Conferences

- Third International Conference on eXtreme Programming and Agile Processes in Software Engineering, May 26-29, 2002, Alghero, Sardinia, Italy. <<http://www.xp2002.org>>.
- XP Agile Universe, August 4-7, 2002, Chicago, Illinois, USA. <http://www.xpuniverse.com>.

Web sites

- Xprogramming, an Extreme Programming Resource. <<http://www.xprogramming.com/>>
- Extreme Programming: A gentle introduction . <<http://www.extremeprogramming.org/>>
- PortlandPatternRepository and WikiWikiWeb: <<http://c2.com/cgi/wiki>>
- XP Developer: <<http://www.xpdeveloper.com/>>
- JUnit and other XUnit testing frameworks. <<http://www.junit.org/>>

- eXtreme Programming. <<http://www.armaties.com/extreme.htm>>
- Pair programming. <<http://pairprogramming.com/>>
- XP123 - Exploring Extreme Programming. <http://xp123.com/>

Some important recent articles

- J. Highsmith, A. Cockburn, *Agile Software Development: The Business of Innovation*, IEEE Computer, September, 2001, pp. 120-122.
- This article include a brief explanation of characteristics and advantages of agile software development approaches (not only XP). Moreover, it include a manifesto where representative persons related to agile development software methods state their compromise with “individuals, working software, customer collaboration and responding to change” as a clear guide for overcoming current problems in software development. The importance of this article, in my opinion, does not reside in its content but in the reaction that it has provoked in the IEEE Computer Magazine Community. For example:
- A letter from S. Rakitin was published in December (Letters, “Manifesto elicits cynicism”, IEEE Computer, December, 2001, pp. 4-7). In it, the author strongly disagreed with the point of view of Highsmith and Cockburn.
 - An article by the well-known B. W. Boehm (B. W. Boehm, “Get Ready for Agile Methods, with care”, January, pp. 64-69) tried to balance the advantages of agile processes and several important preventive measures that should be taken in consideration in order to avoid “overresponding to change”.
 - Two recent letters (Letters, “Professional Approach to Software Development” and “Unavoidable statistics”, IEEE Computer, March, 2002, pp.6-7) in the issue of March 2002 make interesting contributions to the point of view expressed by Rakitin and Boehm.

As anyone can see, XP is a really “hot spot” for the software development community.