Torgeir Dingsøyr (Ed.)

LNCS 3281

Software Process Improvement

11th European Conference, EuroSPI 2004
Trondheim, Norway, November 2004
Proceedings

Springer

Torgeir Dingsøyr (Ed.)

Software Process Improvement

11th European Conference, EuroSPI 2004
Trondheim, Norway, November 10-12, 2004
Proceedings

Springer

Volume Editor

Torgeir Dingsøyr
SINTEF ICT
S P Andersens vei 15, 7465 Trondheim, Norway
E-mail: Torgeir.Dingsoyr@sintef.no

Library of Congress Control Number: 2004113832

CR Subject Classification (1998): D.2, K.6, K.4.2

ISSN 0302-9743
ISBN 3-540-23725-9 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11341086 06/3142 5 4 3 2 1 0

Preface

This was the first year that the European Software Process Improvement Conference – EuroSPI – had a separate research track with its own proceedings. The EuroSPI conference is in its eleventh year, and has become the main meeting place in Europe for the software industry and academia to discuss software process improvement. The conference deals with software process improvement in a broad sense, investigating organizational issues as well as methods and tools for software process improvement.

EuroSPI is an initiative financed by a consortium of Nordic research centers and user networks (SINTEF, DELTA and STTF), ASQF, a German quality assurance association, and ISCN in Ireland, the coordinating network partner.

The research papers describe innovative and significant work in software process improvement, which is relevant to the software industry. The papers are readable for a scientific and industrial audience, and support claims with appropriately described evidence or references to relevant literature.

Thirty-one papers were submitted in this year's research track, and each paper was sent to three or four members of the program committee or additional reviewers. Papers were evaluated according to originality, significance of the contribution, quality of the written and graphical presentation, research method applied, and appropriateness of comparison to relevant research and literature. Almost 100 reviews were received and 18 papers were selected for presentation in the research track, giving a rejection rate of 42%. Many high-quality submissions had to be rejected because of limited space in the conference program.

The selected papers cover a wide area in software process improvement, from improving agile development methods, techniques for software process improvement, and knowledge management in software companies to effort estimation and global software development.

I would like to thank the paper authors for providing papers of high quality, and the program committee and additional reviewers for critiques, praise and advice on how to make the papers even better.

For further information about future EuroSPI conferences, see www.eurospi.net.

August 2004 Torgeir Dingsøyr

EuroSPI 2004 Conference Organization

General Chair

Dr. Richard Messnarz, ISCN, Ireland

Local Chair

Nils Brede Moe, SINTEF ICT, Norway

Scientific Programme Committee Chair

Dr. Torgeir Dingsøyr, SINTEF ICT, Norway

Industrial Programme Committee Chairs

Jørn Johansen, DELTA, Denmark
Mads Christiansen, DELTA, Denmark
Risto Nevalainen, STTF, Finland

Industry Chair

Dr. Bernd Hindel, ASQF, Germany

Exhibition Chair

Robert Treffny, ASQF, Germany

Tutorial Chair

Dr. Richard Messnarz, ISCN, Ireland

Research Track Program Committee

Pekka Abrahamsson, VTT Electronics, Finland
Vincenzo Ambriola, University of Pisa, Italy
Aybuke Aurum, University of New South Wales, Australia
Stefan Biffl, Vienna University of Technology, Austria
Miklos Biro, University of Budapest, Hungary
Christian Bunse, Fraunhofer IESE, Germany
Marcus Ciolkowski, University of Kaiserslautern, Germany
Karl Cox, NICTA, Australia
Kevin C. Desouza, University of Illinois, USA

Taz Daughtrey, James Madison University, USA
Howard Duncan, Dublin City University, Ireland
Tore Dybå, SINTEF, Norway
Jan Pries-Heje, IT University of Copenhagen, Denmark
Natalia Juristo, Polytechnical University of Madrid, Spain
Karl Heinz Kautz, Copenhagen Business School, Denmark
Jyrki Kontio, Helsinki University of Technology, Finland
Dieter Landes, Coburg University of Applied Sciences, Germany
Mikael Lindvall, Fraunhofer Center, USA
Patricia McQuaid, California Polytechnic State University, USA
Jürgen Münch, Fraunhofer IESE, Germany
Matthias Müller, University of Karlsruhe, Germany
Markku Oivo, University of Oulu, Finland
Elixabete Ostolaza, European Software Institute, Spain
Ita Richardson, University of Limerick, Ireland
Gunther Ruhe, University of Calgary, Canada
Per Runeson, Lund Institute of Technology, Sweden
Kurt Schneider, University of Hannover, Germany
Martin Shepperd, Bournemouth University, UK
Magne Jørgensen, Simula Research Laboratory, Norway
Tor Stålhane, Norwegian University of Science and Technology, Norway
Timo Varkoi, Tampere University of Technology, Finland
Claes Wohlin, Blekinge Institute of Technology, Sweden

Additional Reviewers

Ulrike Becker-Kornstaedt, USA
M. Letizia Jaccheri, Norwegian University of Science and Technology, Norway
Daniel Karlström, Lund Institute of Technology, Sweden
Ana M. Moreno, Universidad Politecnica de Madrid, Spain
Knut H. Rolland, Norwegian University of Science and Technology, Norway
Maribel Sanchez-Segura, Universidad Carlos III de Madrid, Spain

Lecture Notes in Computer Science

For information about Vols. 1–3193

please contact your bookseller or Springer

Vol. 3305: P.M.A. Sloot, B. Chopard, A.G. Hoekstra (Eds.), Cellular Automata. XV, 883 pages. 2004.

Vol. 3302: W.-N. Chin (Ed.), Programming Languages and Systems. XIII, 453 pages. 2004.

Vol. 3299: F. Wang (Ed.), Automated Technology for Verification and Analysis. XII, 506 pages. 2004.

Vol. 3293: C.-H. Chi, M. van Steen, C. Wills (Eds.), Web Content Caching and Distribution. IX, 283 pages. 2004.

Vol. 3292: R. Meersman, Z. Tari, A. Corsaro (Eds.), On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops. XXIII, 885 pages. 2004.

Vol. 3291: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE. XXV, 824 pages. 2004.

Vol. 3290: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE. XXV, 823 pages. 2004.

Vol. 3287: A. Sanfeliu, J.F.M. Trinidad, J.A. Carrasco Ochoa (Eds.), Progress in Pattern Recognition, Image Analysis and Applications. XVII, 703 pages. 2004.

Vol. 3286: G. Karsai, E. Visser (Eds.), Generative Programming and Component Engineering. XIII, 491 pages. 2004.

Vol. 3284: A. Karmouch, L. Korba, E.R.M. Madeira (Eds.), Mobility Aware Technologies and Applications. XII, 382 pages. 2004.

Vol. 3281: T. Dingsøyr (Ed.), Software Process Improvement. X, 207 pages. 2004.

Vol. 3280: C. Aykanat, T. Dayar, İ. Körpeoğlu (Eds.), Computer and Information Sciences - ISCIS 2004. XVIII, 1009 pages. 2004.

Vol. 3274: R. Guerraoui (Ed.), Distributed Computing. XIII, 465 pages. 2004.

Vol. 3273: T. Baar, A. Strohmeier, A. Moreira, S.J. Mellor (Eds.), <<UML>> 2004 - The Unified Modelling Language. XIII, 454 pages. 2004.

Vol. 3271: J. Vicente, D. Hutchison (Eds.), Management of Multimedia Networks and Services. XIII, 335 pages. 2004.

Vol. 3270: M. Jeckle, R. Kowalczyk, P. Braun (Eds.), Grid Services Engineering and Management. X, 165 pages. 2004.

Vol. 3269: J. Lopez, S. Qing, E. Okamoto (Eds.), Information and Communications Security. XI, 564 pages. 2004.

Vol. 3266: J. Solé-Pareta, M. Smirnov, P.V. Mieghem, J. Domingo-Pascual, E. Monteiro, P. Reichl, B. Stiller, R.J. Gibbens (Eds.), Quality of Service in the Emerging Networking Panorama. XVI, 390 pages. 2004.

Vol. 3265: R.E. Frederking, K.B. Taylor (Eds.), Machine Translation: From Real Users to Research. XI, 392 pages. 2004. (Subseries LNAI).

Vol. 3264: G. Paliouras, Y. Sakakibara (Eds.), Grammatical Inference: Algorithms and Applications. XI, 291 pages. 2004. (Subseries LNAI).

Vol. 3263: M. Weske, P. Liggesmeyer (Eds.), Object-Oriented and Internet-Based Technologies. XII, 239 pages. 2004.

Vol. 3262: M.M. Freire, P. Chemouil, P. Lorenz, A. Gravey (Eds.), Universal Multiservice Networks. XIII, 556 pages. 2004.

Vol. 3261: T. Yakhno (Ed.), Advances in Information Systems. XIV, 617 pages. 2004.

Vol. 3260: I.G.M.M. Niemegeers, S.H. de Groot (Eds.), Personal Wireless Communications. XIV, 478 pages. 2004.

Vol. 3258: M. Wallace (Ed.), Principles and Practice of Constraint Programming – CP 2004. XVII, 822 pages. 2004.

Vol. 3257: E. Motta, N.R. Shadbolt, A. Stutt, N. Gibbins (Eds.), Engineering Knowledge in the Age of the Semantic Web. XVII, 517 pages. 2004. (Subseries LNAI).

Vol. 3256: H. Ehrig, G. Engels, F. Parisi-Presicce, G. Rozenberg (Eds.), Graph Transformations. XII, 451 pages. 2004.

Vol. 3255: A. Benczúr, J. Demetrovics, G. Gottlob (Eds.), Advances in Databases and Information Systems. XI, 423 pages. 2004.

Vol. 3254: E. Macii, V. Paliouras, O. Koufopavlou (Eds.), Integrated Circuit and System Design. XVI, 910 pages. 2004.

Vol. 3253: Y. Lakhnech, S. Yovine (Eds.), Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems. X, 397 pages. 2004.

Vol. 3252: H. Jin, Y. Pan, N. Xiao, J. Sun (Eds.), Grid and Cooperative Computing - GCC 2004 Workshops. XVIII, 785 pages. 2004.

Vol. 3251: H. Jin, Y. Pan, N. Xiao, J. Sun (Eds.), Grid and Cooperative Computing - GCC 2004. XXII, 1025 pages. 2004.

Vol. 3250: L.-J. (LJ) Zhang, M. Jeckle (Eds.), Web Services. X, 301 pages. 2004.

Vol. 3249: B. Buchberger, J.A. Campbell (Eds.), Artificial Intelligence and Symbolic Computation. X, 285 pages. 2004. (Subseries LNAI).

Vol. 3246: A. Apostolico, M. Melucci (Eds.), String Processing and Information Retrieval. XIV, 332 pages. 2004.

Vol. 3245: E. Suzuki, S. Arikawa (Eds.), Discovery Science. XIV, 430 pages. 2004. (Subseries LNAI).

Vol. 3244: S. Ben-David, J. Case, A. Maruoka (Eds.), Algorithmic Learning Theory. XIV, 505 pages. 2004. (Subseries LNAI).

Vol. 3243: S. Leonardi (Ed.), Algorithms and Models for the Web-Graph. VIII, 189 pages. 2004.

Vol. 3242: X. Yao, E. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J. Rowe, P. Tiňo, A. Kabán, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature - PPSN VIII. XX, 1185 pages. 2004.

Vol. 3241: D. Kranzlmüller, P. Kacsuk, J.J. Dongarra (Eds.), Recent Advances in Parallel Virtual Machine and Message Passing Interface. XIII, 452 pages. 2004.

Vol. 3240: I. Jonassen, J. Kim (Eds.), Algorithms in Bioinformatics. IX, 476 pages. 2004. (Subseries LNBI).

Vol. 3239: G. Nicosia, V. Cutello, P.J. Bentley, J. Timmis (Eds.), Artificial Immune Systems. XII, 444 pages. 2004.

Vol. 3238: S. Biundo, T. Frühwirth, G. Palm (Eds.), KI 2004: Advances in Artificial Intelligence. XI, 467 pages. 2004. (Subseries LNAI).

Vol. 3236: M. Núñez, Z. Maamar, F.L. Pelayo, K. Pousttchi, F. Rubio (Eds.), Applying Formal Methods: Testing, Performance, and M/E-Commerce. XI, 381 pages. 2004.

Vol. 3235: D. de Frutos-Escrig, M. Nunez (Eds.), Formal Techniques for Networked and Distributed Systems – FORTE 2004. X, 377 pages. 2004.

Vol. 3232: R. Heery, L. Lyon (Eds.), Research and Advanced Technology for Digital Libraries. XV, 528 pages. 2004.

Vol. 3231: H.-A. Jacobsen (Ed.), Middleware 2004. XV, 514 pages. 2004.

Vol. 3230: J.L. Vicedo, P. Martínez-Barco, R. Muñoz, M. Saiz Noeda (Eds.), Advances in Natural Language Processing. XII, 488 pages. 2004. (Subseries LNAI).

Vol. 3229: J.J. Alferes, J. Leite (Eds.), Logics in Artificial Intelligence. XIV, 744 pages. 2004. (Subseries LNAI).

Vol. 3226: M. Bouzeghoub, C. Goble, V. Kashyap, S. Spaccapietra (Eds.), Semantics of a Networked World. XIII, 326 pages. 2004.

Vol. 3225: K. Zhang, Y. Zheng (Eds.), Information Security. XII, 442 pages. 2004.

Vol. 3224: E. Jonsson, A. Valdes, M. Almgren (Eds.), Recent Advances in Intrusion Detection. XII, 315 pages. 2004.

Vol. 3223: K. Slind, A. Bunker, G. Gopalakrishnan (Eds.), Theorem Proving in Higher Order Logics. VIII, 337 pages. 2004.

Vol. 3222: H. Jin, G.R. Gao, Z. Xu, H. Chen (Eds.), Network and Parallel Computing. XX, 694 pages. 2004.

Vol. 3221: S. Albers, T. Radzik (Eds.), Algorithms – ESA 2004. XVIII, 836 pages. 2004.

Vol. 3220: J.C. Lester, R.M. Vicari, F. Paraguaçu (Eds.), Intelligent Tutoring Systems. XXI, 920 pages. 2004.

Vol. 3219: M. Heisel, P. Liggesmeyer, S. Wittmann (Eds.), Computer Safety, Reliability, and Security. XI, 339 pages. 2004.

Vol. 3217: C. Barillot, D.R. Haynor, P. Hellier (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004. XXXVIII, 1114 pages. 2004.

Vol. 3216: C. Barillot, D.R. Haynor, P. Hellier (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004. XXXVIII, 930 pages. 2004.

Vol. 3215: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems. LVII, 906 pages. 2004. (Subseries LNAI).

Vol. 3214: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems. LVIII, 1302 pages. 2004. (Subseries LNAI).

Vol. 3213: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems. LVIII, 1280 pages. 2004. (Subseries LNAI).

Vol. 3212: A. Campilho, M. Kamel (Eds.), Image Analysis and Recognition. XXIX, 862 pages. 2004.

Vol. 3211: A. Campilho, M. Kamel (Eds.), Image Analysis and Recognition. XXIX, 880 pages. 2004.

Vol. 3210: J. Marcinkowski, A. Tarlecki (Eds.), Computer Science Logic. XI, 520 pages. 2004.

Vol. 3209: B. Berendt, A. Hotho, D. Mladenic, M. van Someren, M. Spiliopoulou, G. Stumme (Eds.), Web Mining: From Web to Semantic Web. IX, 201 pages. 2004. (Subseries LNAI).

Vol. 3208: H.J. Ohlbach, S. Schaffert (Eds.), Principles and Practice of Semantic Web Reasoning. VII, 165 pages. 2004.

Vol. 3207: L.T. Yang, M. Guo, G.R. Gao, N.K. Jha (Eds.), Embedded and Ubiquitous Computing. XX, 1116 pages. 2004.

Vol. 3206: P. Sojka, I. Kopecek, K. Pala (Eds.), Text, Speech and Dialogue. XIII, 667 pages. 2004. (Subseries LNAI).

Vol. 3205: N. Davies, E. Mynatt, I. Siio (Eds.), UbiComp 2004: Ubiquitous Computing. XVI, 452 pages. 2004.

Vol. 3204: C.A. Peña Reyes, Coevolutionary Fuzzy Modeling. XIII, 129 pages. 2004.

Vol. 3203: J. Becker, M. Platzner, S. Vernalde (Eds.), Field Programmable Logic and Application. XXX, 1198 pages. 2004.

Vol. 3202: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Knowledge Discovery in Databases: PKDD 2004. XIX, 560 pages. 2004. (Subseries LNAI).

Vol. 3201: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Machine Learning: ECML 2004. XVIII, 580 pages. 2004. (Subseries LNAI).

Vol. 3199: H. Schepers (Ed.), Software and Compilers for Embedded Systems. X, 259 pages. 2004.

Vol. 3198: G.-J. de Vreede, L.A. Guerrero, G. Marín Raventós (Eds.), Groupware: Design, Implementation and Use. XI, 378 pages. 2004.

Vol. 3196: C. Stary, C. Stephanidis (Eds.), User-Centered Interaction Paradigms for Universal Access in the Information Society. XII, 488 pages. 2004.

Vol. 3195: C.G. Puntonet, A. Prieto (Eds.), Independent Component Analysis and Blind Signal Separation. XXIII, 1266 pages. 2004.

Vol. 3194: R. Camacho, R. King, A. Srinivasan (Eds.), Inductive Logic Programming. XI, 361 pages. 2004. (Subseries LNAI).

Table of Contents

On-Site Customer in an XP Project: Empirical Results from a Case Study

Juha Koskela and Pekka Abrahamsson

VTT Technical Research Centre of Finland
P.O.Box 1100, FIN-90571 Oulu, Finland
{Juha.Koskela,Pekka.Abrahamsson}@vtt.fi

Abstract. Extreme programming (XP), similarly to other agile software development methods, values close collaboration with customers. One of the XP practices suggests that the customer should be 100% available for the development team. Anecdotal evidence suggests that the XP customer role is costly, difficult and demanding. However, very few empirical studies have been published on the role of customer in an XP project. The results of this controlled case study are in line with the common belief that the on-site customer's role is indeed demanding, requiring a strong ability to resolve issues rapidly. Yet, the study also offers contrasting findings in terms of required actual customer involvement in the development project. This empirical case demonstrates that while the customer was present close to 100% with the development team, only 21% of his work effort was required to assist the team in the development. However, it is also shown that an on-site customer may create a false sense of confidence in the system under development. The implications of these and other findings are discussed.

Keywords: Extreme programming, on-site customer, customer involvement

1 Introduction

Extreme programming (XP), first introduced in [1], is focused on generating early releases of working products and aims to deliver business value from the very beginning of the project. The role of customer is highly valued in XP, and it is considered important for the success of the project [2, 3]. The on-site customer practice of XP suggests that the customer should be 100% available for the development team, so as to be able to provide quick help in, e.g., answering questions and resolving problems.

This paper reports the empirical results from a controlled extreme programming case study, where the customer was present close to 100% of the development time. A team of four developers was acquired to implement a system for managing the research data obtained over years at a Finnish research institute. Both quantitative and qualitative data of the on-site customer role in XP is provided. The quantitative data consists of customer effort usage and effort distribution. The qualitative data comprises development diaries maintained by the developers, a customer diary, postmortem analysis session recordings and developer interviews. It has been argued that the XP customer role is demanding and that it requires lots of involvement [e.g. 4, 5-7]. While this study supports these claims, it also yields contrasting results. It is shown that the on-site customer offers the team a unique situation to consult him

T. Dingsøyr (Ed.): EuroSPI 2004, LNCS 3281, pp. 1–11, 2004.

whenever needed. The development team perceives this as a strong demonstration of commitment to their work. The data also reveals that the on-site customer is in danger to create a false sense of confidence in the remaining of the customer organization. The results of this study offer empirical evidence to support the common belief that the on-site customer role is demanding and requires a strong ability to resolve issues rapidly. However, the empirical case demonstrates that although the customer was close to 100% present with the development team, only 21% of his work effort was required to assist the team in the development.

The paper is organized as follows. The following section introduces extreme programming and the related research. This is followed by a description of research settings, research methods and data collection methods. Section four presents the results and in section five implications of these findings are discussed. Section six concludes the paper.

2 Extreme Programming

This section introduces extreme programming and related research, focusing on the on-site customer.

Agile methods have gained a significant amount of attention in the field of software engineering in the last few years. Currently, extreme programming (XP) is the best-known agile method. XP is primarily designed for object-oriented projects using a maximum of a dozen programmers at one location [3]. This kind of situation is called “agile home ground” by Boehm [8]. The XP process is characterized by short development cycles, incremental planning, continuous feedback, and reliance on communication and evolutionary design [2]. The core of XP is made up of a simple set of common-sense practices. These practices are planning game, small releases, metaphor, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, 40-hour week, on-site customer, coding standards, open workspace and just rules. For more information about XP and an overview of other agile methods readers are referred to [e.g. 9, 10].

From the viewpoint of this study, the most interesting XP practice is the on-site customer. It has been suggested that the customer should be available for the development team throughout the project, to answer questions and resolve problems, for example. In XP, the customer is the person who sits with the project team, generates and prioritizes stories, provides acceptance tests for each release, and makes the final business decisions [11]. Therefore, the on-site customer delivers the requirements and represents all the knowledge that must be available for the development team. Despite this important role, there are only few empirically validated studies focusing on the on-site customer available.

Wallace et al. [12] list three possible customer locations: on-site customer, off-site customer and remote customer. According to XP literature [e.g. 2, 3, 13], the customer optimally works in the same room with the developers. However, this is not always possible; for example, the customer may be too valuable to be on-site [3]. According to Jeffries et al. [3], an XP project may survive even without customer presence, but the project will go faster and smoother if the customer is on-site. If the project team does not include a customer, they have to plan further in advance, which, for its part, adds the level of risk in the project [2].

Yet, it is not only the customer working on-site that makes an XP project successful. According to XP literature [e.g. 2, 11], it is also important to have a customer who plays the role well. According to Beck and Fowler [11, p. 18], a good customer understands the domain, is aware of how software can provide business value in the domain, can make decisions about what is needed at a given moment and what is needed later, and is willing to accept ultimate responsibility for the success or failure of the project. Martin et al. [4] established three research hypotheses covering the characteristics of the customer, the skills of the customer, and the location of the customer. They found the role of XP customer a very demanding one, requiring, e.g., preparation, skills, attention to detail, and the ability to make critical decisions. Martin et al. [4] report that even an ideal preparation for the customer role may not be sufficient to ensure success in the XP customer role.

Farell et al. [5] describe a successful XP implementation from the viewpoint of the customer. According to Farell et al. [5, pp. 4], "it is critical to have a high degree of customer involvement in the process." Griffin [6] has also come to similar conclusions regarding XP implementation, recommending the key customer contact(s) to be placed close to the development team. Nawrocki et al. [14, pp. 294] argue that "a close, personal contact with customers and their instant presence is a must for XP-like processes". Lippert et al. [15] have written a book in which they describe their experiences of XP practices. The authors emphasize the importance of smooth communication between development team and customer. Stephens and Rosenberg [7], for their part, provide a critical viewpoint towards XP in their book. According to Stephens and Rosenberg [7, pp. 133], "the trouble with on-site customer done the XP way is that if the on-site customer is a single person, she becomes a single point of failure in an incredibly difficult, stressful, high-profile position of great responsibility". Table 1 summarizes the most critical arguments of related research.

3 Research Design

This section describes how the research design for the study is laid out.

3.1 Research Setting

A team of four developers was set up to implement an intranet application (called eXpert) for managing the research data obtained over years at a Finnish research institute. The four developers were 5th to 6th year university students, all with 1 to 4 years of industrial experience in software development. The team members were well-versed in the Java programming language and object-oriented analysis and design approaches. Two weeks prior to project launch, the team performed a self-study session by studying two basic books on XP [i.e., 2, 3]. A two-day hands-on training on XP practices, development environment and software configuration management (SCM) was organized to ensure that the team had a basic understanding on XP issues and the specific technical environment used. As development environment, an Eclipse integration framework (http://www.eclipse.org) was used, which is an open source initiative supported by major software engineering tool manufacturers. CVS (Concurrent Versions System) was used as project SCM tool and the JUnit testing framework for unit testing. Both CVS client and JUnit are integrated as a default in the Eclipse

Table 1. Summary of related research

Claim, argument or suggestion	Description	References
A high degree of customer involvement is required	It is critical to have a high degree of customer involvement in the process	[5, 6, 14]
The role of on-site customer is very demanding	XP customer role is highly demanding, requiring, e.g., preparation, skills, attention to detail, and ability to make critical decisions	[4, 7]
The customer should work in the same room with developers	Optimum conditions for an XP project are provided by the customer sharing the same workspace as developers	[2-4]

environment. The application was written in Java and JSP (JavaServer Pages) and it used the MySQL relational database in storing link data. In addition, the Apache Tomcat 4 Servlet/JSP container was used because of its implementation of the JSP 1.2 specifications of Java Software.

The team worked in a co-located development environment. The customer (i.e., the first author) shared the same office space with the development team. The office space and workstations were organized according to the suggestions made in the XP literature to support efficient teamwork.

3.2 Research Method

A detailed description of the general research method, i.e., the controlled case study approach, utilized in this study can be found in [16]. The controlled case study approach strives for replication (experimentation) and in-depth data collection (case study) and it also has the ability to change the process (action research) in a close-to-industry setting in which also business pressure is present [16]. The first author was in the role of on-site customer, participating in planning game, acceptance testing, post-mortem analysis, project meetings and coaching activities. On average, he spent over 80% of his work time in the same room with the developers. The second author was acting in the role of management in the study, mediating the release post-mortem analysis [17] sessions, which were performed after each software release. These post-mortem analysis sessions served as a process change mechanism, where the project team could propose changes to the implementation process.

3.3 Data Collection

Both quantitative and qualitative data were collected. Quantitative data consisted of customer effort usage and effort distribution. Qualitative data included development diaries maintained by the developers, a customer diary, post-mortem analysis session recordings and developer interviews. The developers and the customer were updating their diaries continuously during the project, tracking time and filling in observations. As indicated by XP principles [2], the customer organization placed explicit value on data collection, thus ensuring alignment with agile software development principles (http://www.agilemanifesto.org).

4 Results

This section presents the results of the study, including both quantitative and qualitative data concerning the on-site customer role in XP. Table 2 provides basic information about the size and schedule of the eXpert project. The system development was carried out in six iterations, of which the first three took up two weeks of calendar time, the next two one week, while the sixth iteration was a two-day correction release. The developers were mainly working six hours a day for four days a week. Detailed data of the eXpert project can be found in [18].

Table 2. Background information of the eXpert project

Collected data	R1	R2	R3	R4	R5	R6	Total
Calendar time (weeks)	2	2	2	1	1	0.4	8.4
Total work effort (h)	195	190	192	111	96	36	820
# User stories implemented	5	9	9	4	3	4	34
# Tasks defined	10	30	18	21	19	9	107

4.1 Customer Effort Usage and Distribution

Figure 1 shows the customer presence for each release (i.e. the time the customer was spending in the same room with the developers). As it can be seen, the customer was present at an average of 83%. Figure 1 also reveals that customer presence was at its highest in the first iteration, while decreasing to iteration three, and increasing to the average level in the forth iteration. The change of course can be explained by the change in iteration length from two weeks to one week. When the customer saw that everything was happening at a faster pace, he was trying to be present as much as possible. In the third iteration, the customer presence was at its highest in hours (59 hours), but lowest in percentage (72%). This results from a fragmented presence of the developers during the third iteration. At that time, the developers were working at more differing times compared to the first two iterations, for example.

Figure 2 shows the actual effort the customer spent in performing project activities in each release (i.e. the time the customer spent in XP activities). Despite the high customer presence percentage values, the actual customer involvement during the releases was ranging from 17.4% to 25.0%, with an average of 20.6%. As it can be seen from figure 2, the actual customer involvement was higher in shorter iterations (two week iterations vs. one week iterations). However, a nearly 100% present on-site customer with an actual involvement rate as low as this is a significant result, since the on-site customer is one of the most controversial topics in extreme programming.

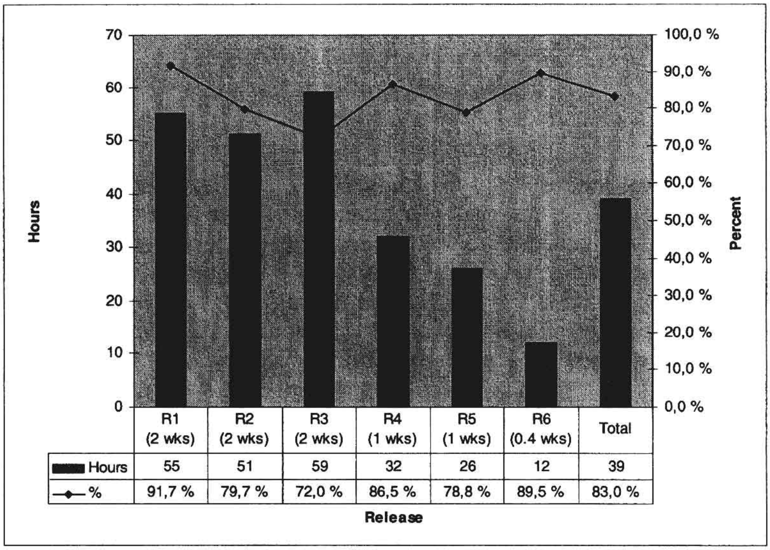

Fig. 1. Customer presence during the project

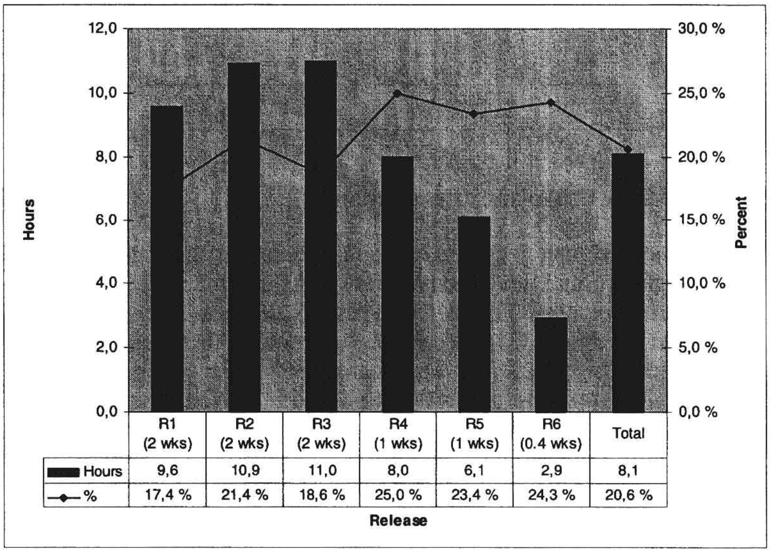

Fig. 2. Actual customer involvement during the project

From the viewpoint of customer effort distribution, participation in planning game and acceptance testing were the two major activities requiring customer's effort (figure 3). Planning game sessions took 42,8% and acceptance testing 29,9% of the total effort. Post mortem sessions [17], which were held at the end of each release cycle, took up 13,4% of the customer effort. The share of project meetings, i.e. planning sessions with the development team during the iterations, and amounted to 8,2% of