LNCS 3092

Jutta Eckstein
Hubert Baumeister (Eds.)

Extreme Programming and Agile Processes in Software Engineering

5th International Conference, XP 2004
Garmisch-Partenkirchen, Germany, June 2004
Proceedings

Springer

Jutta Eckstein Hubert Baumeister (Eds.)

Extreme Programming and Agile Processes in Software Engineering

5th International Conference, XP 2004
Garmisch-Partenkirchen, Germany, June 6-10, 2004
Proceedings

Springer

Volume Editors

Jutta Eckstein
Independent Consultant
Thierschstr. 20, 80538 Munich, Germany
E-mail: jutta@jeckstein.com

Hubert Baumeister
Ludwig-Maximilians-Universität München, Institut für Informatik
Oettingenstr. 67, 80538, Munich, Germany
E-mail: baumeist@informatik.uni-muenchen.de

Library of Congress Control Number: 2004106717

CR Subject Classification (1998): D.2, D.1, D.3, K.6.3, K.6, K.4.3, F.3

ISSN 0302-9743
ISBN 3-540-22137-9 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable to prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11011590 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3092

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
New York University, NY, USA

Doug Tygar
University of California, Berkeley, CA, USA

Moshe Y. Vardi
Rice University, Houston, TX, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Preface

Software development is being revolutionized. The heavy-weight processes of the 1980s and 1990s are being replaced by light-weight, so called agile processes. Agile processes move the focus of software development back to what really matters: running software. This is only made possible by accepting that software development is a creative job done by, with, and for individual human beings. For this reason, agile software development encourages interaction, communication, and fun.

This was the focus of the Fifth International Conference on Extreme Programming and Agile Processes in Software Engineering which took place between June 6 and June 10, 2004 at the conference center in Garmisch-Partenkirchen at the foot of the Bavarian Alps near Munich, Germany. In this way the conference provided a unique forum for industry and academic professionals to discuss their needs and ideas for incorporating Extreme Programming and Agile Methodologies into their professional life under consideration of the human factor. We celebrated this year's conference by reflecting on what we had achieved in the last half decade and we also focused on the challenges we will face in the near future.

The focus of the whole conference was on learning and interaction. We brought practitioners together in different sessions to discuss their experiences in solving software problems. XP 2004 facilitated the swapping of ideas in a number of ways, including featured talks by professionals on the cutting edge of extreme programming and agile processes, technical presentations, activity sessions, panels, posters, code camps, workshops, tutorials, and other opportunities to exchange and elaborate on new findings. XP 2004 also featured a PhD symposium for PhD students and a trainer and educator track for everybody with a vested interest in training and education.

This volume is divided into several topical sections. First you will find full papers, then the abstracts of the invited talks, followed by the abstracts of the posters and demonstrations. Then you will find the papers of the trainers' and educators' track, right before the abstracts of the PhD symposium papers. Next, all accepted workshops are presented, followed by the panels and the activities to conclude the book.

The papers went through a rigorous reviewing process. Each paper was reviewed by at least three program committee members and was discussed carefully among the program committee members. Of 69 papers submitted, only 23 were accepted as full papers. These papers were grouped into six categories, focusing on the following themes:

- Acceptance Testing: These papers focus on different ways of dealing with the problem of creating acceptance tests. Most of them suggest a framework that supports the customer as well as the developer to ensure the functionality of the system.

此为试读,需要完整PDF请访问: www.ertongbook.com

- Scalability Issues: This section collects all the papers dealing with agile development in a larger setting. Some of them consider the aspect of dispersed development, others the issues that arise when outsourcing parts of the development, and others discuss the problems and solutions of diverse agile practices, such as continuous integration, in a large team.
- New Insights: These papers present various new ideas in the field of agile development. Some suggest the use of agile development in a different field such as open source development and others explore new techniques such as agile specification driven development.
- Refactoring: This section contains papers discussing problems and solutions within the practice of refactoring. Large refactorings is one topic, whereas refactoring in a legacy system is another one.
- Social Aspects: All papers focusing on social aspects are collected in this section. The characteristics of XP teams and different roles in agile teams are one focus, employee satisfaction is another.
- Practitioner Reports: Several authors report on their experiences with agile development. You will find papers about the influence of user participation in agile development, the loss and gain with adapting the agile process during development, as well as agile project controlling.

Next you will find the abstracts of the invited talks, right before we present the posters and demonstrations. Posters and demonstrations are ideal for presenting preliminary research results, experience reports, late-breaking developments, or for giving an introduction to interesting, innovative work.

Then you will find the papers and posters of the trainers' and educators' track. This track was aimed at industry and academic professionals who are interested in teaching and learning extreme programming and agile processes. Professionals from academia and industry were invited to discuss their needs and ideas for integrating extreme programming and agile processes into training plans and courses.

The next section presents the abstracts of the PhD symposium papers. In the symposium, students presented and discussed their research objectives, methods, and (preliminary) results.

After this, all accepted workshops are presented. The XP 2004 workshop program provided a forum for groups of participants to exchange opinions and to enhance community knowledge about research topics and real-life applications of agile processes. The workshops also provided the opportunity for representatives of the technical community to coordinate efforts and to establish collective plans of action.

In the last section you will find the descriptions of the activities and the panels. Panels and activity-sessions were lively, participatory, educational, and entertaining. They offered an interactive opportunity to share perspectives, debate opinions, and communicate best-practices.

These proceedings contain conference versions of the papers, posters, demonstrations, and panel position statements as well as the papers of the trainer and educator track and the abstracts of the PhD symposium. Besides a collection of ideas and experiences, they represent an aspect of collective learning and understanding within a community of people who came together in Garmisch-Partenkirchen for five days in June 2004.

We would like to thank everybody who contributed to XP 2004; the authors, the workshop and activity leaders, the tutorial speakers, the panelists, those who served on the various committees, our sponsors, those who offered their experience of running previous XP conferences, the staff of the conference center in Garmisch-Partenkirchen and, last but not least, the participants.

April 2004

Jutta Eckstein
Hubert Baumeister

Organization

XP 2004 is organized by Software und Support Verlag in cooperation with the Institut für Informatik of the Ludwig-Maximilians-Unversität München.

Executive Committee

General Chair:	Martin Fowler (ThoughtWorks Inc., USA)
Program Chair:	Jutta Eckstein (Germany)
Academic Chair:	Hubert Baumeister (LMU Munich, Germany)
Tutorial Chair:	Rachel Davies (UK)
Workshop Chair:	Vera Peeters (Belgium)
Panels and Activities:	Steven Fraser and Joshua Kerievsky (USA)
Poster Chair:	Rachel Reinitz (UK)
Educators Track Chairs:	Helen Sharp and Mike Holcombe (UK)
PhD Symposium Chairs:	Barbara Russo (Italy)
Chair of Social Activities:	Nicolai Josuttis (Germany)

Program Committee

Ann Anderson (USA)
Barbara Russo (Italy)
Bernhard Rumpe (Germany)
Charles Poole (USA)
Daniel Karlström (Sweden)
David Hussman (USA)
Diana Larsen (USA)
Dierk König (Switzerland)
Don Wells (USA)
Erich Gamma (Switzerland)
Frank Westphal (Germany)
Giancarlo Succi (Italy)
Helen Sharp (UK)
Hubert Baumeister (Germany)
Jim Highsmith (USA)
Joe Bergin (USA)
John Favaro (Italy)
José H. Canós Cerdá (Spain)
Joseph Pelrine (Switzerland)
Joshua Kerievsky (USA)
Laurie Williams (USA)
Linda Rising (USA)
Marco Abis (Italy)
Martin Fowler (USA)
Martin Lippert (Germany)
Mary Lynn Manns (USA)
Mary Poppendieck (USA)
Michael Hill (USA)
Michele Marchesi (Italy)
Mike Holcombe (UK)
Nicolai Josuttis (Germany)
Paul Grünbacher (Austria)
Rachel Davis (UK)
Rachel Reinitz (USA)
Rick Mugridge (New Zealand)
Ron Jeffries (USA)
Scott W. Ambler (USA)
Sian Hopes (UK)
Steve Freeman (UK)
Steven Fraser (USA)
Till Schümmer (Germany)
Tim Mackinnon (UK)
Vera Peeters (Belgium)
Ward Cunningham (USA)

Lecture Notes in Computer Science

For information about Vols. 1–2976

please contact your bookseller or Springer-Verlag

Vol. 3092: J. Eckstein, H. Baumeister (Eds.), Extreme Programming and Agile Processes in Software Engineering. XVI, 358 pages. 2004.

Vol. 3083: W. Emmerich, A.L. Wolf (Eds.), Component Deployment. X, 249 pages. 2004.

Vol. 3076: D. Buell (Ed.), Algorithmic Number Theory. XI, 451 pages. 2004.

Vol. 3074: B. Kuijpers, P. Revesz (Eds.), Constraint Databases and Applications. XII, 181 pages. 2004.

Vol. 3066: S. Tsumoto, R. S lowiński, J. Komorowski, J.W. Grzymala-Busse (Eds.), Rough Sets and Current Trends in Computing. XX, 853 pages. 2004. (Subseries LNAI).

Vol. 3065: A. Lomuscio, D. Nute (Eds.), Deontic Logic. X, 275 pages. 2004. (Subseries LNAI).

Vol. 3064: D. Bienstock, G. Nemhauser (Eds.), Integer Programming and Combinatorial Optimization. XI, 445 pages. 2004.

Vol. 3063: A. Llamosí, A. Strohmeier (Eds.), Reliable Software Technologies - Ada-Europe 2004. XIII, 333 pages. 2004.

Vol. 3062: J.L. Pfaltz, M. Nagl, B. Böhlen (Eds.), Applications of Graph Transformations with Industrial Relevance. XV, 500 pages. 2004.

Vol. 3060: A.Y. Tawfik, S.D. Goodwin (Eds.), Advances in Artificial Intelligence. XIII, 582 pages. 2004. (Subseries LNAI).

Vol. 3059: C.C. Ribeiro, S.L. Martins (Eds.), Experimental and Efficient Algorithms. X, 586 pages. 2004.

Vol. 3058: N. Sebe, M.S. Lew, T.S. Huang (Eds.), Computer Vision in Human-Computer Interaction. X, 233 pages. 2004.

Vol. 3056: H. Dai, R. Srikant, C. Zhang (Eds.), Advances in Knowledge Discovery and Data Mining. XIX, 713 pages. 2004. (Subseries LNAI).

Vol. 3054: I. Crnkovic, J.A. Stafford, H.W. Schmidt, K. Wallnau (Eds.), Component-Based Software Engineering. XI, 311 pages. 2004.

Vol. 3053: C. Bussler, J. Davies, D. Fensel, R. Studer (Eds.), The Semantic Web: Research and Applications. XIII, 490 pages. 2004.

Vol. 3052: W. Zimmermann, B. Thalheim (Eds.), Abstract State Machines 2004. Advances in Theory and Practice. XII, 235 pages. 2004.

Vol. 3051: R. Berghammer, B. Möller, G. Struth (Eds.), Relational and Kleene-Algebraic Methods in Computer Science. X, 279 pages. 2004.

Vol. 3047: F. Oquendo, B. Warboys, R. Morrison (Eds.), Software Architecture. X, 279 pages. 2004.

Vol. 3046: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications - ICCSA 2004. LIII, 1016 pages. 2004.

Vol. 3045: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1040 pages. 2004.

Vol. 3044: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1140 pages. 2004.

Vol. 3043: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.J.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1180 pages. 2004.

Vol. 3042: N. Mitrou, K. Kontovasilis, G.N. Rouskas, I. Iliadis, L. Merakos (Eds.), NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications. XXXIII, 1519 pages. 2004.

Vol. 3039: M. Bubak, G.D.v. Albada, P.M. Sloot, J. Dongarra (Eds.), Computational Science - 2004. LXVI, 1271 pages. 2004.

Vol. 3038: M. Bubak, G.D.v. Albada, P.M. Sloot, J. Dongarra (Eds.), Computational Science - ICCS 2004. LXVI, 1311 pages. 2004.

Vol. 3037: M. Bubak, G.D.v. Albada, P.M. Sloot, J. Dongarra (Eds.), Computational Science - ICCS 2004. LXVI, 745 pages. 2004.

Vol. 3036: M. Bubak, G.D.v. Albada, P.M. Sloot, J. Dongarra (Eds.), Computational Science - ICCS 2004. LXVI, 713 pages. 2004.

Vol. 3035: M.A. Wimmer (Ed.), Knowledge Management in Electronic Government. XII, 326 pages. 2004. (Subseries LNAI).

Vol. 3034: J. Favela, E. Menasalvas, E. Chávez (Eds.), Advances in Web Intelligence. XIII, 227 pages. 2004. (Subseries LNAI).

Vol. 3033: M. Li, X.-H. Sun, Q. Deng, J. Ni (Eds.), Grid and Cooperative Computing. XXXVIII, 1076 pages. 2004.

Vol. 3032: M. Li, X.-H. Sun, Q. Deng, J. Ni (Eds.), Grid and Cooperative Computing. XXXVII, 1112 pages. 2004.

Vol. 3031: A. Butz, A. Krüger, P. Olivier (Eds.), Smart Graphics. X, 165 pages. 2004.

Vol. 3030: P. Giorgini, B. Henderson-Sellers, M. Winikoff (Eds.), Agent-Oriented Information Systems. XIV, 207 pages. 2004. (Subseries LNAI).

Vol. 3029: B. Orchard, C. Yang, M. Ali (Eds.), Innovations in Applied Artificial Intelligence. XXI, 1272 pages. 2004. (Subseries LNAI).

Vol. 3028: D. Neuenschwander, Probabilistic and Statistical Methods in Cryptology. X, 158 pages. 2004.

Vol. 3027: C. Cachin, J. Camenisch (Eds.), Advances in Cryptology - EUROCRYPT 2004. XI, 628 pages. 2004.

Vol. 3026: C. Ramamoorthy, R. Lee, K.W. Lee (Eds.), Software Engineering Research and Applications. XV, 377 pages. 2004.

Vol. 3025: G.A. Vouros, T. Panayiotopoulos (Eds.), Methods and Applications of Artificial Intelligence. XV, 546 pages. 2004. (Subseries LNAI).

Vol. 3024: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 621 pages. 2004.

Vol. 3023: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 611 pages. 2004.

Vol. 3022: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 621 pages. 2004.

Vol. 3021: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 633 pages. 2004.

Vol. 3019: R. Wyrzykowski, J. Dongarra, M. Paprzycki, J. Wasniewski (Eds.), Parallel Processing and Applied Mathematics. XIX, 1174 pages. 2004.

Vol. 3016: C. Lengauer, D. Batory, C. Consel, M. Odersky (Eds.), Domain-Specific Program Generation. XII, 325 pages. 2004.

Vol. 3015: C. Barakat, I. Pratt (Eds.), Passive and Active Network Measurement. XI, 300 pages. 2004.

Vol. 3014: F. van der Linden (Ed.), Software Product-Family Engineering. IX, 486 pages. 2004.

Vol. 3012: K. Kurumatani, S.-H. Chen, A. Ohuchi (Eds.), Multi-Agnets for Mass User Support. X, 217 pages. 2004. (Subseries LNAI).

Vol. 3011: J.-C. Régin, M. Rueher (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. XI, 415 pages. 2004.

Vol. 3010: K.R. Apt, F. Fages, F. Rossi, P. Szeredi, J. Váncza (Eds.), Recent Advances in Constraints. VIII, 285 pages. 2004. (Subseries LNAI).

Vol. 3009: F. Bomarius, H. Iida (Eds.), Product Focused Software Process Improvement. XIV, 584 pages. 2004.

Vol. 3008: S. Heuel, Uncertain Projective Geometry. XVII, 205 pages. 2004.

Vol. 3007: J.X. Yu, X. Lin, H. Lu, Y. Zhang (Eds.), Advanced Web Technologies and Applications. XXII, 936 pages. 2004.

Vol. 3006: M. Matsui, R. Zuccherato (Eds.), Selected Areas in Cryptography. XI, 361 pages. 2004.

Vol. 3005: G.R. Raidl, S. Cagnoni, J. Branke, D.W. Corne, R. Drechsler, Y. Jin, C.G. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G.D. Smith, G. Squillero (Eds.), Applications of Evolutionary Computing. XVII, 562 pages. 2004.

Vol. 3004: J. Gottlieb, G.R. Raidl (Eds.), Evolutionary Computation in Combinatorial Optimization. X, 241 pages. 2004.

Vol. 3003: M. Keijzer, U.-M. O'Reilly, S.M. Lucas, E. Costa, T. Soule (Eds.), Genetic Programming. XI, 410 pages. 2004.

Vol. 3002: D.L. Hicks (Ed.), Metainformatics. X, 213 pages. 2004.

Vol. 3001: A. Ferscha, F. Mattern (Eds.), Pervasive Computing. XVII, 358 pages. 2004.

Vol. 2999: E.A. Boiten, J. Derrick, G. Smith (Eds.), Integrated Formal Methods. XI, 541 pages. 2004.

Vol. 2998: Y. Kameyama, P.J. Stuckey (Eds.), Functional and Logic Programming. X, 307 pages. 2004.

Vol. 2997: S. McDonald, J. Tait (Eds.), Advances in Information Retrieval. XIII, 427 pages. 2004.

Vol. 2996: V. Diekert, M. Habib (Eds.), STACS 2004. XVI, 658 pages. 2004.

Vol. 2995: C. Jensen, S. Poslad, T. Dimitrakos (Eds.), Trust Management. XIII, 377 pages. 2004.

Vol. 2994: E. Rahm (Ed.), Data Integration in the Life Sciences. X, 221 pages. 2004. (Subseries LNBI).

Vol. 2993: R. Alur, G.J. Pappas (Eds.), Hybrid Systems: Computation and Control. XII, 674 pages. 2004.

Vol. 2992: E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, E. Ferrari (Eds.), Advances in Database Technology - EDBT 2004. XVIII, 877 pages. 2004.

Vol. 2991: R. Alt, A. Frommer, R.B. Kearfott, W. Luther (Eds.), Numerical Software with Result Verification. X, 315 pages. 2004.

Vol. 2990: J. Leite, A. Omicini, L. Sterling, P. Torroni (Eds.), Declarative Agent Languages and Techniques. XII, 281 pages. 2004. (Subseries LNAI).

Vol. 2989: S. Graf, L. Mounier (Eds.), Model Checking Software. X, 309 pages. 2004.

Vol. 2988: K. Jensen, A. Podelski (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. XIV, 608 pages. 2004.

Vol. 2987: I. Walukiewicz (Ed.), Foundations of Software Science and Computation Structures. XIII, 529 pages. 2004.

Vol. 2986: D. Schmidt (Ed.), Programming Languages and Systems. XII, 417 pages. 2004.

Vol. 2985: E. Duesterwald (Ed.), Compiler Construction. X, 313 pages. 2004.

Vol. 2984: M. Wermelinger, T. Margaria-Steffen (Eds.), Fundamental Approaches to Software Engineering. XII, 389 pages. 2004.

Vol. 2983: S. Istrail, M.S. Waterman, A. Clark (Eds.), Computational Methods for SNPs and Haplotype Inference. IX, 153 pages. 2004. (Subseries LNBI).

Vol. 2982: N. Wakamiya, M. Solarski, J. Sterbenz (Eds.), Active Networks. XI, 308 pages. 2004.

Vol. 2981: C. Müller-Schloer, T. Ungerer, B. Bauer (Eds.), Organic and Pervasive Computing – ARCS 2004. XI, 339 pages. 2004.

Vol. 2980: A. Blackwell, K. Marriott, A. Shimojima (Eds.), Diagrammatic Representation and Inference. XV, 448 pages. 2004. (Subseries LNAI).

Vol. 2979: I. Stoica, Stateless Core: A Scalable Approach for Quality of Service in the Internet. XVI, 219 pages. 2004.

Vol. 2978: R. Groz, R.M. Hierons (Eds.), Testing of Communicating Systems. XII, 225 pages. 2004.

Vol. 2977: G. Di Marzo Serugendo, A. Karageorgos, O.F. Rana, F. Zambonelli (Eds.), Engineering Self-Organising Systems. X, 299 pages. 2004. (Subseries LNAI).

Table of Contents

Acceptance Testing

Scalability Issues

New Insights

此为试读，需要完整PDF请访问：www.ertongbook.com

Refactoring

Social Issues

Practitioner Reports

Invited Talks

Posters

Demonstrations

Trainers and Educators Track

Ph.D. Symposium

Workshops

Panels and Activities

The Video Store Revisited Yet Again: Adventures in GUI Acceptance Testing

Johan Andersson and Geoff Bache

Carmen Systems AB, Odinsgatan 9, SE-41103 Göteborg, Sweden
geoff.bache@carmensystems.com

Abstract. Acceptance testing for graphical user interfaces has long been recognised as a hard problem. At the same time, a full suite of acceptance tests written by the Onsite Customer has been a key principle of XP since it began [1]. It seems, however, that practice has lagged behind theory, with many practitioners still reporting weak or no acceptance testing. At XP2003, we presented our successes with text-based acceptance testing of a batch program[2]. In the past year we have extended this approach to apply to a user interface. We have developed an approach based on simulation of user actions via a record/replay layer between the application and the GUI library, generating a high-level script that functions as a use-case scenario, and using our text-based approach for verification of correctness. We believe this is an approach to GUI acceptance testing which is both customer- and developer-friendly.

1 XP Acceptance Testing

We should be clear what we regard as the primary aims of acceptance tests. These are the standards by which we judge acceptance tests and approaches to acceptance testing:

- The tests should model the actions of a user as closely as possible.
- Writing the tests should be quick, painless and require as few programming skills as possible,
- Running the tests should be as smoothe as possible - press a button and watch them go green/red.
- Maintaining the tests should not be too laborious.
- Tests should be as stable under changes as possible. In particular they should be independent of things like font, user interface layout and internal design.
- Tests should document the features of the system in as readable a way as possible.

Let's also be clear at what we are not aiming for. While the following are worthy aims, they are mainly the responsibility of other practices, for example Unit Testing or the various replacements for it that we described in last year's paper [2].

- The tests should not aim to improve or document the design.
- The tests should concentrate on indicating the presence of errors, not primarily help in fixing them.

J. Eckstein and H. Baumeister (Eds.): XP 2004, LNCS 3092, pp. 1–10, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2 Introduction

Our open source acceptance testing tool, TextTest [3], has traditionally been a console application that we have used to test UNIX batch tools. Recently, however, we wrote a GUI for it, and wanted to be able to test this GUI using a variation of the same approach. We have come up with an approach to do this that we found to be highly effective. For the sake of this paper, however, we thought that we would use what we have learned to revisit the classic Video Store problem, as this is likely to be more familiar to readers and avoids the meta-situation of programs testing themselves! The Video Store has been used to illustrate a few aspects of XP already, from refactoring to unit testing.[4]

TextTest is written in Python, and its GUI uses the PyGTK library[5]. The examples are therefore taken from this environment.

3 The Theory: Principles of Our Approach

3.1 Separating Simulation from Verification

Acceptance testing of GUIs has traditionally been regarded as one activity. Perhaps due to our background with applications that do not have an interactive aspect, we have come to regard it as two, largely independent activities: simulating the interactive actions of a user in some persistent way (e.g. a test script) and verifying that the behaviour is correct when performing these actions. For future reference we refer to these as *simulation* and *verification.*

This simplifies matters somewhat because it removes the need for a tool that does both, decoupling the activities. Each tool can then concentrate on being good at one thing only. Armed with a pre-existing verification tool, TextTest[3], (discussed later) which has proved successful in the world of batch applications, the main challenge of testing a GUI is to find an effective approach to simulation.

3.2 An Agile Record/Replay Approach

Record/Replay approaches have a strong theoretical appeal to us. To be able to create tests as a user simply by clicking around the application under test seems to be the easiest imaginable interface. Many tests can be created quickly, it is totally clear to the person creating them what they represent, no (potentially error-prone) code needs to be written per test and the only qualification for writing them is understanding the system under test, which is needed anyway.

Record/Replay tools are nothing new. A wide range of them exist, of which QCReplay[6] is the one we have most experience of. In recent years, a bewildering array of open source varieties for Java have appeared as well[7]. They are generally based on intercepting mouse clicks and keyboard input, recording them in a script, and asserting behaviour by taking screen dumps ("photographing" the screen)

They are not renowned for their popularity in the Agile community, however. They tend to produce long, low-level scripts which are extremely tied to the environment at the time when they were recorded.[8] For example:

此为试读，需要完整PDF请访问：www.ertongbook.com