

Rezension des Papers
“Recovery, Redemption
and Extreme Programming”
von Peter Schuh
Vorlesung IT-Alignment
Prof. Dr. Heindl

Martin Reiter

HS-Furtwangen

18. Juni 2009

eXtreme Programming - a customer's perspective



Fahrplan

- 1 Hintergründe zum Autor
- 2 XP
- 3 Das Paper
- 4 Fazit

Peter Schuh

- Peter Schuh: Recovery, Redemption, and Extreme Programming. IEEE Software 18(6): 34-41 (2001)
- Pramod Sadalage, Peter Schuh: The Agile Database. XP/Agile Universe 2002: 257-258
- ThoughtWorks
- Zusammenarbeit mit Martin Fowler

- Planungsspiel
- Verfügbarer Kunde (On Site Customer)
- Metapher
- Kurze Releasezyklen
- Test getrieben entwickeln
- Refactoring
- Pair Programming
- Einfaches Design
- Gemeinsamer Codebesitz
- Coderichtlinien im XP
- Kontinuierliche Integration
- 40 Stunden Woche

- Geschichte über ein Projekt
- Projekt steht auf der Kippe
- eXtreme Programming Techniken als Rettungsmaßnahmen eingeführt
- Umfang: 7-8 Seiten

Relevanz

- 40 Verweise laut GoogleScholar
- 18 Verweise laut Uni Trier

Kernaussagen

- XP muss mit bedacht eingeführt werden wenn das Kind schon im Brunnen liegt
- Es ist wichtig welche XP-Praktiken man wann einführt
- Unit-Testing, Refactoring, Simple Design, Continuous-Integration, Coding-Standards als wichtigste hervorgehoben
- XP macht "schlechte" Software wartbar und tragbar
- XP ist nicht das Allheilmittel
- Essentiell ist Kommitment vom Kunden, Management und dem Team
- Man muss das Team bei Laune halten - es kommt auf jeden einzelnen an

Ähnliche Meinungen

- “XP Testing Without XP: Taking Advantage of Agile Testing Practices”
- “The New X-Men”
- “Project Rescue Through Agile”

- “eXtreme Programming is not a silver bullet”
- “Are Reviews an Alternative to Pair Programming?”
- “What Can Go Wrong With XP?”
- “Eight reasons why extreme programming won’t work in your shop”

- Erfahrungen mit XP in Summe eher positiv
- XP offensichtlich in “extremen” Situationen hilfreich
- XP aber stets mit Bedacht einsetzen

Vielen Dank für Ihre Aufmerksamkeit!

Zu Risiken und Nebenwirkungen stellen Sie eXtreme Fragen
oder führen Sie Diskussionen ;-)

Quellen

1. Martha Baer 2003
"The New X-Men"
<http://www.wired.com/wired/archive/11.09/xmen.html?pg=1>
2. Koskela et Al. 2004
"Experiences of Using Extreme Programming to Support a Legacy Information System Migration Project"
<http://www.agile-itea.org/public/papers/koskela-experiences.pdf>
3. Lisa Crispin 2003
"XP Testing Without XP: Taking Advantage of Agile Testing Practices"
<http://www.methodsandtools.com/archive/archive.php?id=2>
4. Jonathan Kohl 2005
"Conventional Software Testing on an Extreme Programming Team"
<http://www.informit.com/articles/article.aspx?p=405513&seqNum=3>
5. Mathew Schwartz 2006
"Requirements Management to the Rescue"
http://career-resources.dice.com/job-technology/requirements_management_to_the_rescue.shtml
6. RobinLab ? ?
"Project Rescue Through Agile"
<http://www.robinlab.com/case-studies/project-rescue-through-agile.html>
7. Brian d Foy 2004
"eXtreme Programming is not a silver bullet"
http://www.oreillynet.com/onlamp/blog/2004/04/extreme_programming_is_not_a_s.html

Quellen

8. Jeff Langr 2008
"What Can Go Wrong With XP"
<http://langrsoft.com/articles/whatCanGoWrong.shtml>
9. Bob Weinstein 2002
"Eight reasons why extreme programming won't work in your shop"
<http://www.builderau.com.au/strategy/projectmanagement/soa/Eight-reasons-why-extreme-programming-won-t-work-in-your-shop/0,339028292,320266423,00.htm>
10. Ralf Buerger ?
"Systematische Softwareentwicklung - Tayloring"
<http://www.ralfbuerger.de/sse/Vorgehensweise/Meine.htm>
11. Kim Man Lui; Chan Keith 2005
"A Road Map for Implementing eXtreme Programming"
12. Matthias M. Müller 2004
"Are Reviews an Alternative to Pair Programming?"
13. Yael Dubinsky, Orit Hazzan and Arie Keren
"Introducing Extreme Programming into a Software Project at the Israeli Air Force"