

VERGLEICH ERFAHRENER UND UNERFAHRENER ENTWICKLERPAARE

Andreas Höfer
Universität Karlsruhe, Fakultät für Informatik

2

Studien zu agilen Methoden

Meta-Studie: Paarprogrammierung vs. Einzelprogrammierung

3

- Dybå et al. 2007
- Fasst 15 Studien zusammen
- Paarprogrammierung mit Vorteilen bei Qualität und Dauer
- Einzelprogrammierung verursacht weniger Aufwand

Einfluss von Erfahrung und Systemkomplexität

4

- Arisholm et al. 2007
- 295 Java-Berater
 - ▣ Anfänger, Fortgeschrittene und Profis
- Zentrale/dezentrale Steuerung
- Kein genereller Vorteil der Paarprogrammierung bzgl. Qualität und Dauer
- Größerer Aufwand als Einzelprogrammierung
- Paarprogrammierung vorteilhaft für Anfänger an Systemen mit dezentraler Steuerung

Testgetriebene Entwicklung (TGE)

5

| Studie | Teilnehmer | Produktivität | Qualität |
|---|------------|------------------|------------------|
| TGE vs. Regressionstest | | | |
| Müller & Hagner 2002 | Studenten | schlechter | kein Unterschied |
| TGE vs. Iterative Test-danach Entwicklung | | | |
| Pančur et al. 2003 | Studenten | kein Unterschied | keine Angabe |
| Geras et al. 2004 | Entwickler | kein Unterschied | keine Angabe |
| Erdogmus et al. 2005 | Studenten | besser | kein Unterschied |
| TGE vs. Konventionelle Entwicklung | | | |
| Edwards 2003 | Studenten | keine Angabe | besser |
| George & Williams 2003 | Entwickler | schlechter | besser |

Auswirkung von Erfahrung auf die TGE

6

- Müller & Höfer 2007: Vergleicht TGE-Prozesse von erfahrenen und unerfahrenen Entwicklern
- Anderes Ziel als bisherige Studien:
 - ▣ Inwieweit folgen die Teilnehmer dem TGE-Prozess?
 - ▣ Wie wirkt sich die unterschiedliche Erfahrung der Teilnehmer auf deren Entwicklungsprozess aus?
- Ergebnis: Prozesse unterschieden sich in fast allen untersuchten Kenngrößen

7

Unsere Studie

Motivation

8

- Sieht die TGE bei erfahrenen Entwicklerpaaren anders aus als bei unerfahrenen Entwicklerpaare?
- Folgeuntersuchung zu Müller & Höfer 2007
- Keine Beurteilung von Effizienz der TGE und Paarprogrammierung

Die Teilnehmer

9

Erfahrene Gruppe

- 14 professionelle Software-Entwickler
- Ø 7,5 Jahre industrielle Programmiererfahrung (7 Jahre mit Java)
- Ø 2,6 Jahre Erfahrung mit TGE
- Ø 5,3 Jahre Erfahrung mit Paarprogrammierung

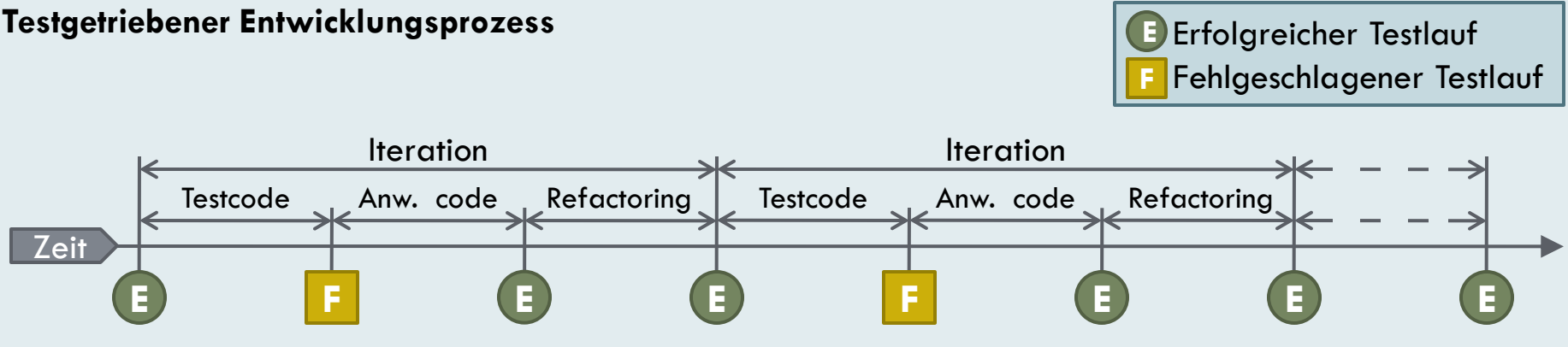
Unerfahrene Gruppe

- 14 Informatikstudenten
- Ø 6,7 Semester
- Ø 4,5 Jahre „Hobby“-Programmiererfahrung (2 Jahre mit Java)
- Teilnehmer des XP-Praktikums
- Ansonsten so gut wie keine Erfahrung mit TGE oder Paarprogrammierung

Schema der TGE

10

Testgetriebener Entwicklungsprozess



Regeln der TGE

11

Regel 1 Eine Änderung im Anwendungscode ist nur in einer Methode erlaubt, die zuvor von einem fehlgeschlagenen Testfall aufgerufen wurde.

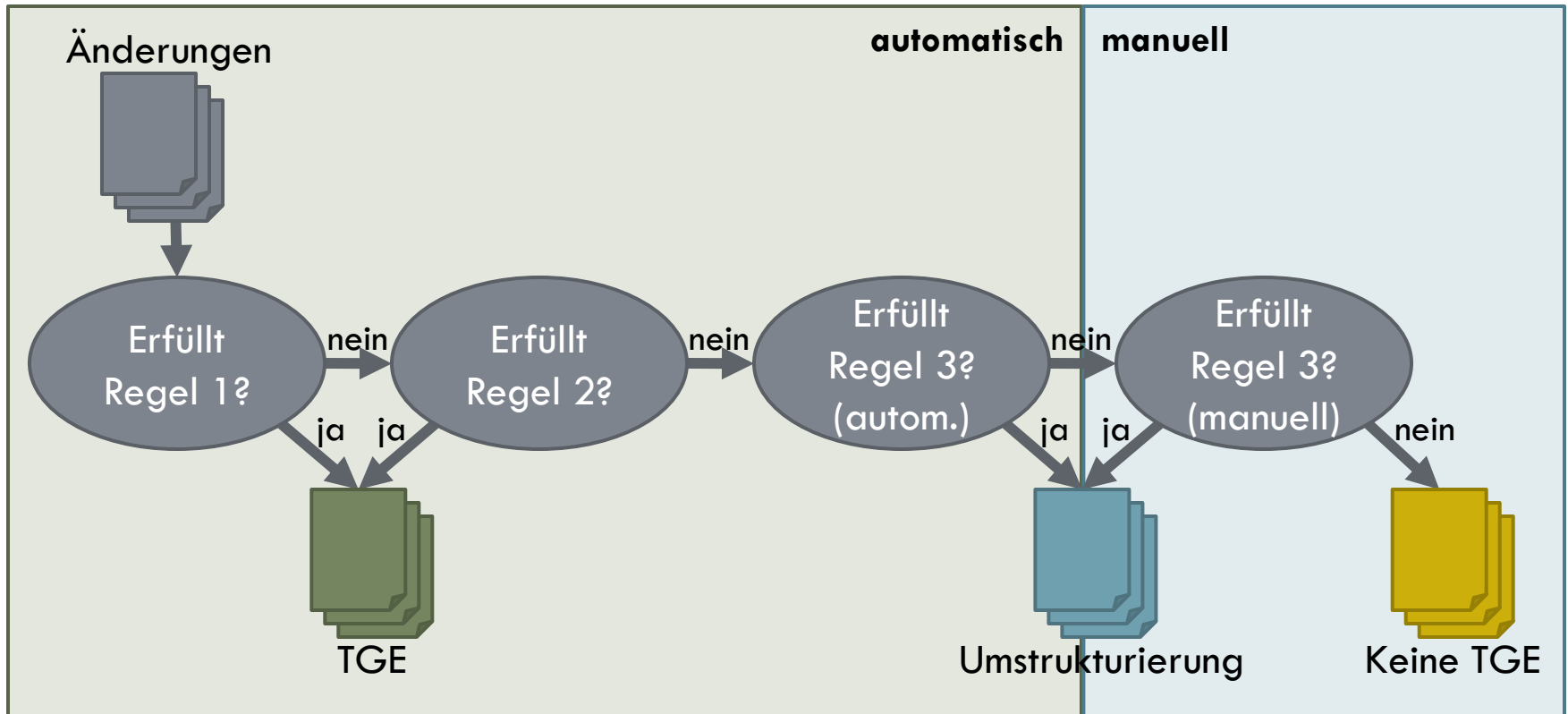
Regel 2 Eine neue Methode muss später von einem fehlschlagenden Testfall aufgerufen werden.

Regel 3 Eine Umstrukturierung ändert die Struktur aber nicht das Verhalten des Programmtextes.

(abgeleitet aus Beck 2000 und Fowler 1999)

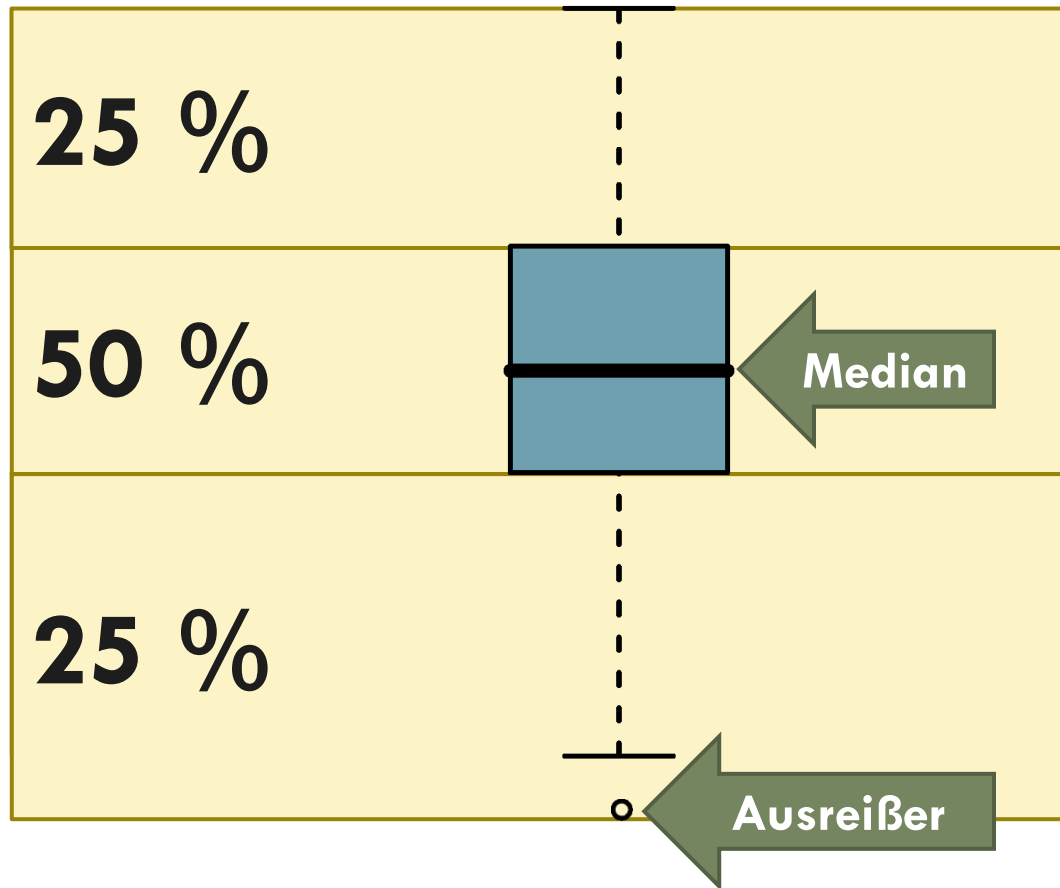
Klassifikation der Änderungen

12



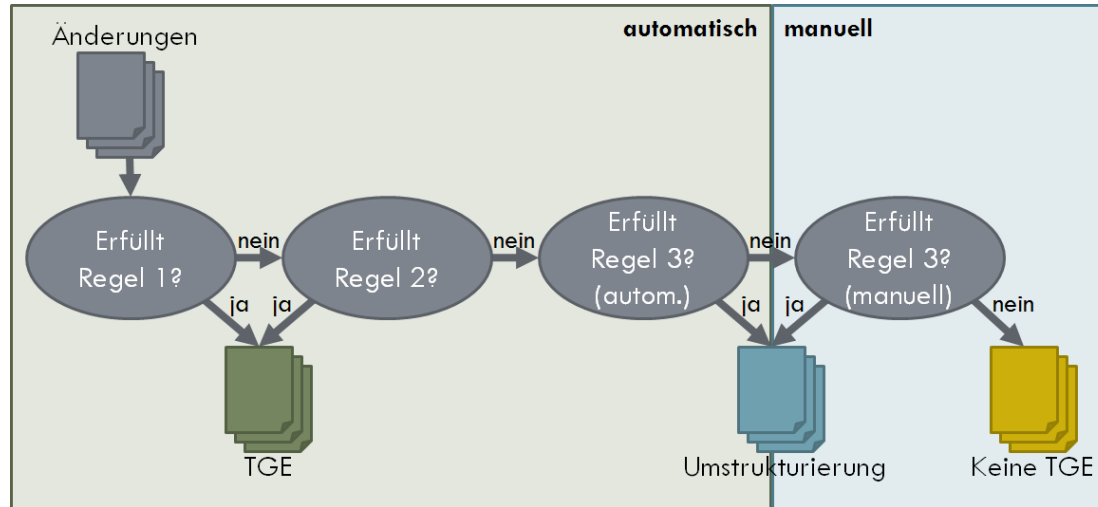
Boxplots

13



TGE-Konformität

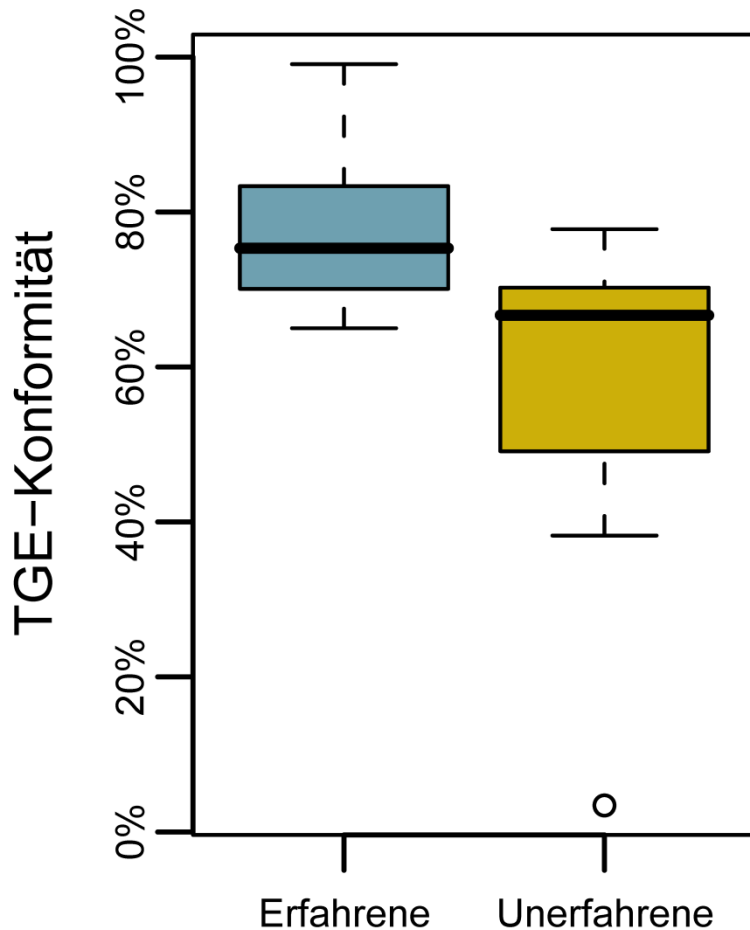
14



$$\text{TGE-Konformität} = \frac{\# \text{ Testg. Änderungen} + \# \text{ Umstrukturierungen}}{\# \text{ Alle Änderungen}}$$

TGE-Konformität

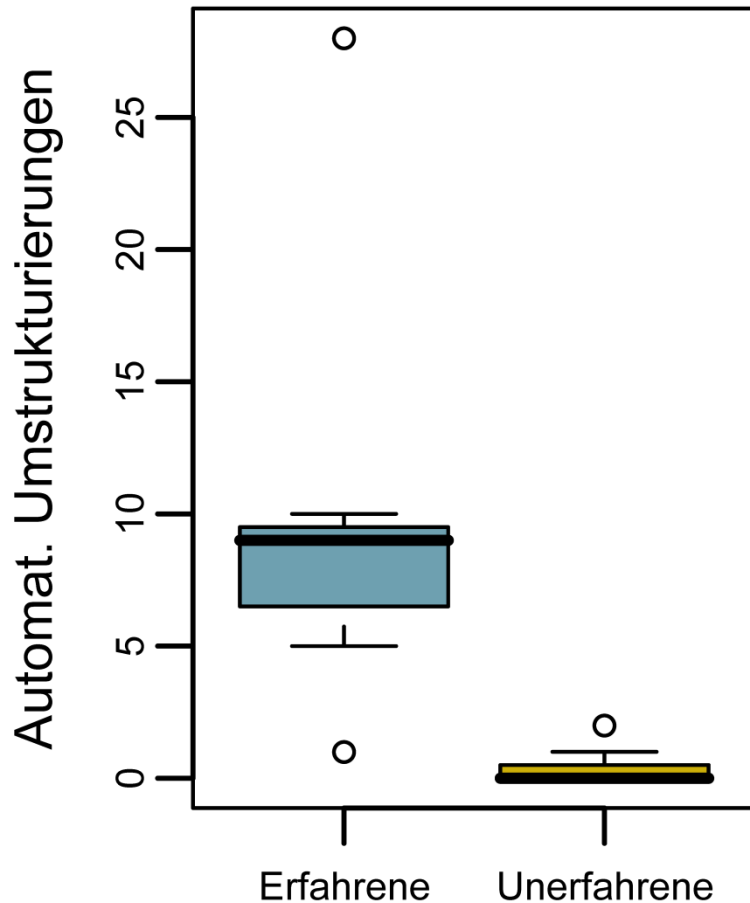
15



- Erfahrene Entwicklerpaare halten sich mehr an TGE-Regeln
- Manche Entwicklerpaare der unerfahrenen Gruppe sind mit erfahrenen Paaren vergleichbar
- Nur ein erfahrenes Entwicklerpaar erreicht mehr als 90%

Umstrukturierungen

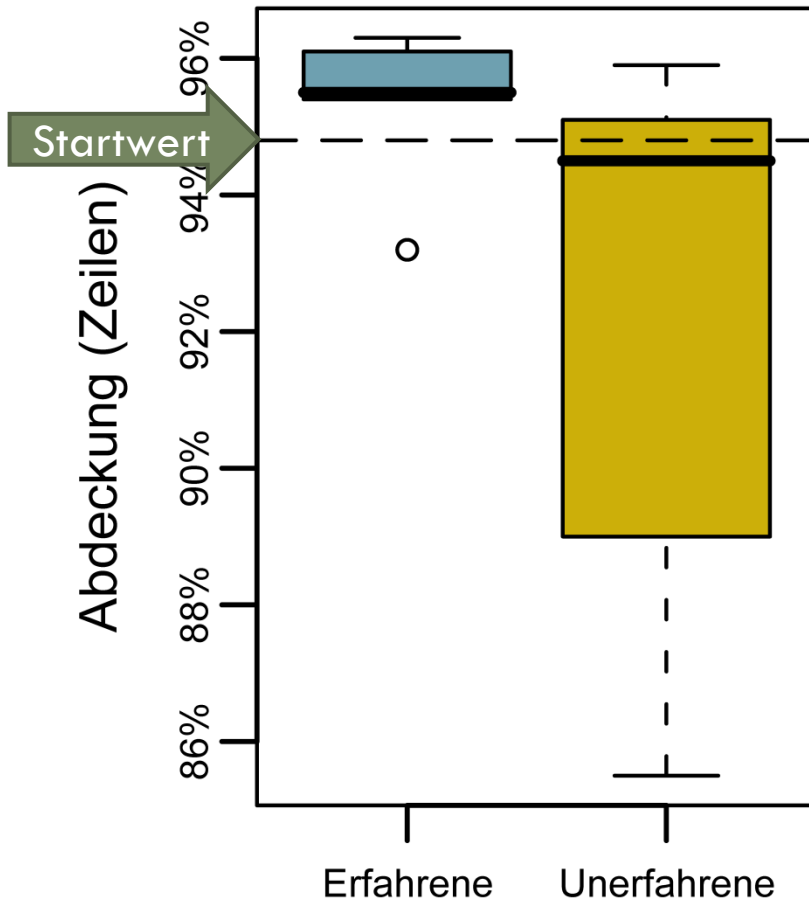
16



- Erfahrene Entwicklerpaare benutzen mehr Umstrukturierungen als unerfahrene Entwicklerpaare
- Gilt sowohl für automatisierte als auch manuelle Umstrukturierungen

Testabdeckung (Zeilen)

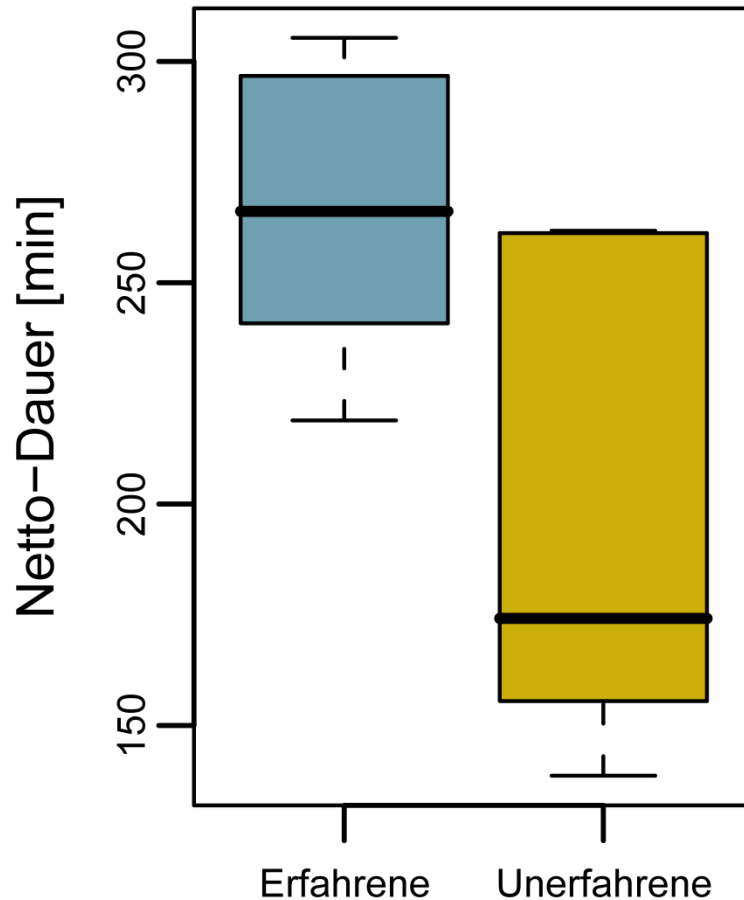
17



- Erfahrene Entwicklerpaare erreichen bessere Testabdeckung
- Große Varianz bei unerfahrenen Entwicklerpaaren
- Qualität der Tests tendenziell besser bei erfahrenen Paaren

Netto-Dauer der Implementierung

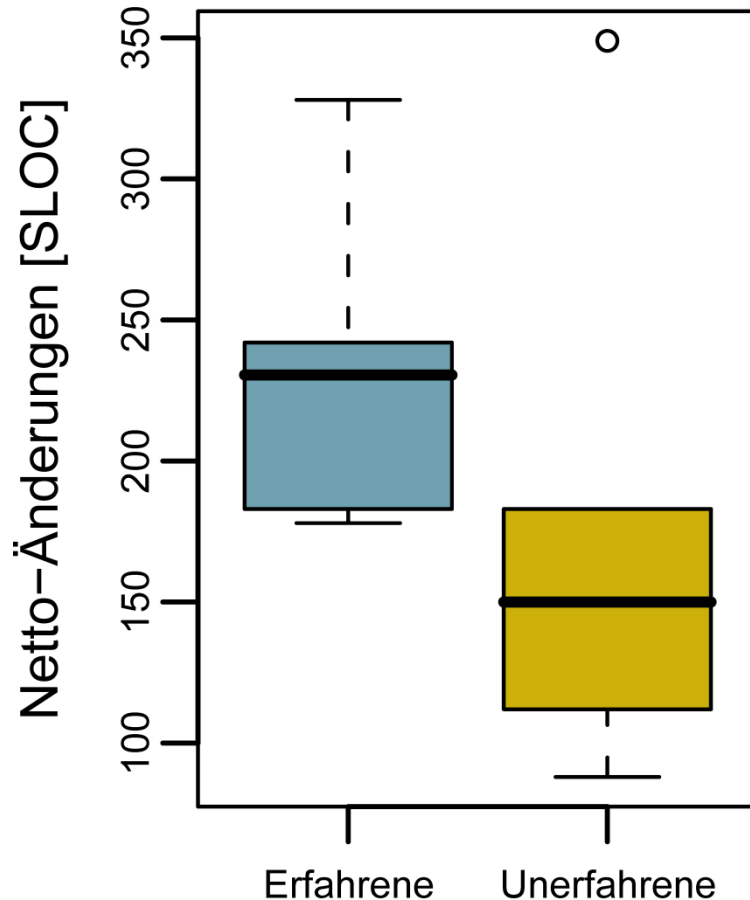
18



- Zeit vom Aushändigen der Aufgabenstellung bis zur Abgabe, ohne Pausen sowie Zeit für Akzeptanztests
- Unerwartet: Unerfahrene Paare benötigen weniger Zeit
- Entgegengesetztes Resultat zu Müller & Höfer 2007

Netto-Änderungen

19



- Vergleich der Ausgangs- und Endversionen der Programme
- Erfahrene Entwicklerpaare nehmen mehr Änderungen vor
- Mögliche Erklärung für größeren Zeitbedarf

Zusammenfassung

20

- Große Unterschiede zwischen erfahrenen und unerfahrenen Entwicklerpaaren
- Ergebnisse von Müller & Höfer 2007 für Einzelentwickler nur teilweise bestätigt:
 - ☺ Erfahrene Paare erreichen höhere TGE-Konformität
 - ☺ Testabdeckung bei erfahrenen Paaren ist höher
 - ☹ Dauer der Implementierung kürzer bei unerfahrenen Entwicklerpaaren
- Agile Techniken bisher unzureichend untersucht


Zukünftige Arbeit

21

- Bisher keine Auswirkung der Erfahrung bei der Paarprogrammierung feststellbar (s. Höfer 2008)
 - ▣ Untersuchung der verbalen Interaktion
- Untersuchung der TGE-Konformität in einem realen Projekt

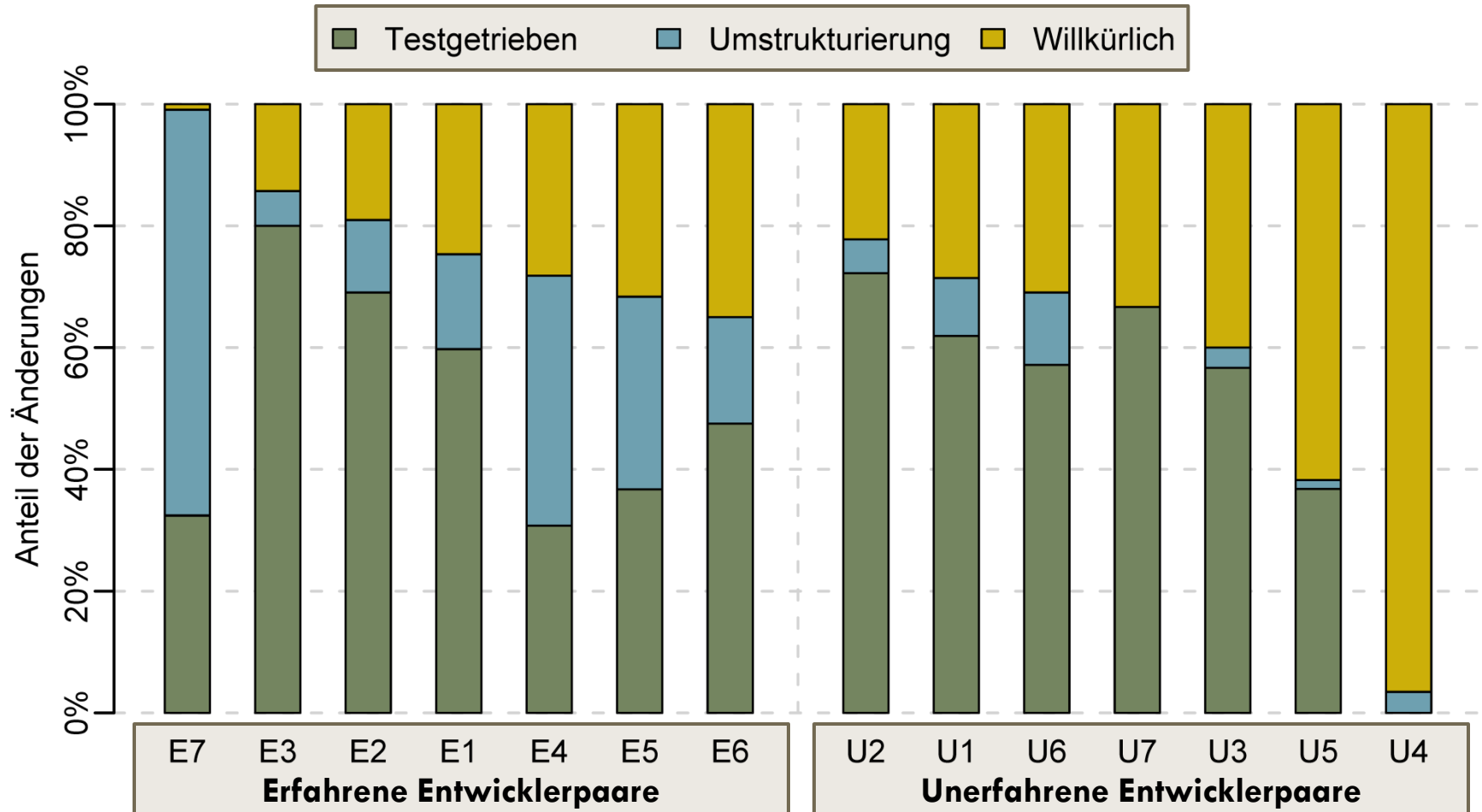
Danksagungen

22

- Marc Philipp für die Implementierung des Auswertungswerkzeugs
- Henning Wolf und Kollegen von *it-agile*  für die finanzielle Unterstützung

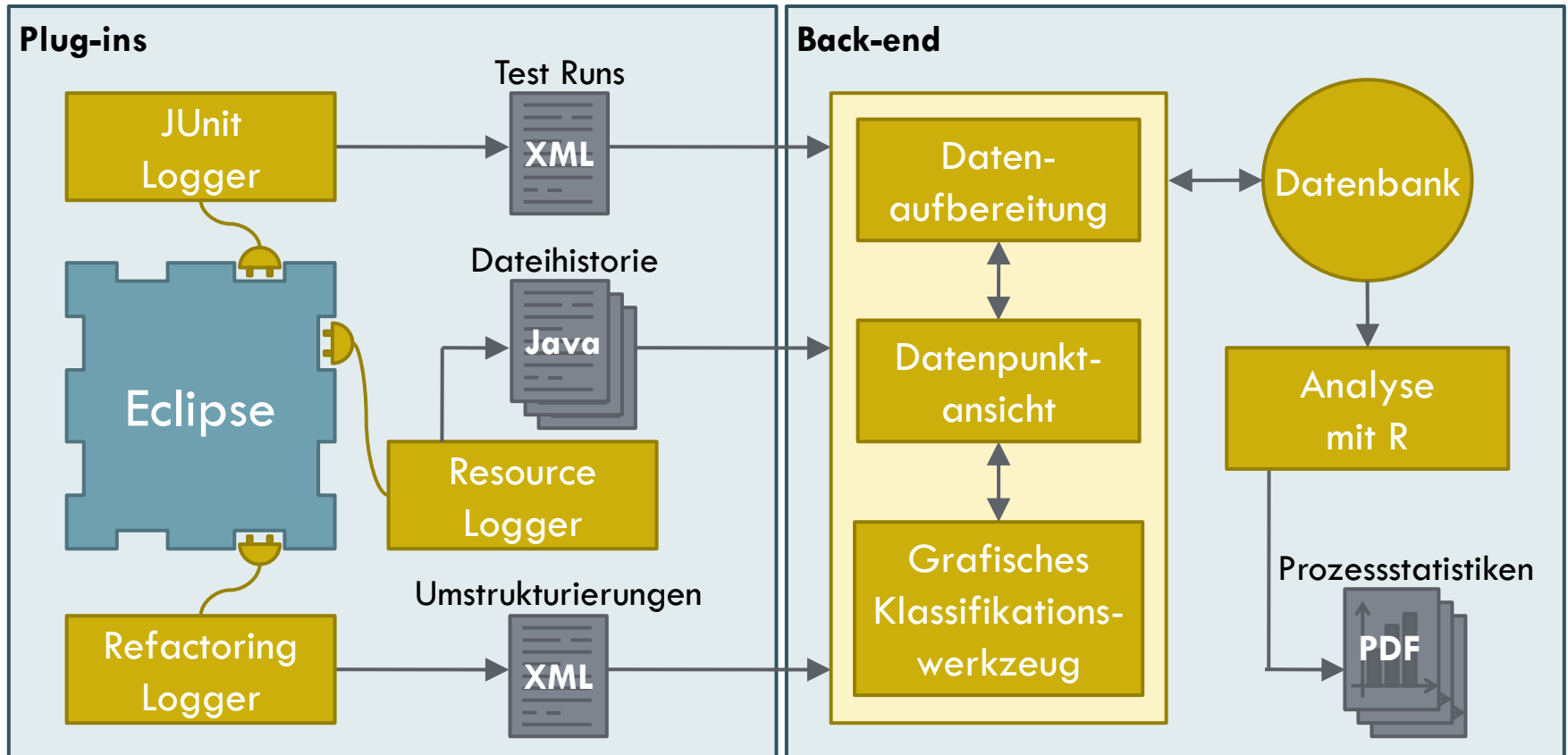
Änderungsverhalten der Paare

23



Das System

24



Referenzen

25

- Arisholm et al. 2007** Arisholm, E.; Gallis, H.; Dybå, T. & Sjøberg, D. I. K.; Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise. *IEEE Transactions on Software Engineering*, 2007, 33(2), S. 65-86
- Beck 2000** Beck, K.; *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 2000
- Dybå et al 2007** Dybå, T.; Arisholm, E.; Sjøberg, D. I.; Hannay, J. E. & Shull, F.; Are Two Heads Better than One? On the Effectiveness of Pair Programming. *IEEE Software*, 2007, 24(6), S. 12-15
- Edwards 2003** Edwards, S. H.; Using Test-Driven Development in the Classroom: Providing Students with Automatic, Concrete Feedback on Performance. *Proceedings of the International Conference on Education and Information Systems: Technologies and Applications (EISTA'03)*, International Institute of Informatics and Systemics, 2003, S. 421-426
- Erdogmus et al. 2005** Erdogmus, H.; Morisio, M.; Torchiano, M.; On the Effectiveness of the Test-First Approach to Programming. *IEEE Transactions on Software Engineering*, 2005, 31(3), S. 226-237
- Fowler 1999** Fowler, M.; *Refactoring – Improving the Design of Existing Code*. Addison-Wesley, 1999
- George & Williams 2003** George, B.; Williams, L.; An Initial Investigation of Test Driven Development in Industry. *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, ACM Press, 2003, S. 1135-1139
- Geras et al. 2004** Geras, A.; Smith, M.; Miller, J. A.; Prototype Empirical Evaluation of Test Driven Development. *METRICS '04: Proceedings of the 10th International Symposium on Software Metrics*, IEEE Computer Society, 2004, S. 405-416

Referenzen

26

- Höfer 2008** Höfer, A.; Exploratory Comparison of Expert and Novice Pair Programmers. *Proceedings of the 3rd IFIP TC2 Central and East European Conference on Software Engineering Techniques, CEE-SET 2008*, 2008, S. 239-252
- Müller & Hagner 2002** Müller, M. M.; Hagner, O.; Experiment about Test-First Programming. *IEE Proceedings – Software*, 2002, 149(5), S. 131-136
- Müller & Höfer 2007** Müller, M. M.; Höfer, A. The Effect of Experience on the Test-Driven Development Process. *Empirical Software Engineering*, 2007, 12(6), S. 593-615
- Pančur et al. 2003** Pančur, M.; Ciglarič, M.; Trampuš, M. & Vidmar, T.; Towards Empirical Evaluation of Test-Driven Development in a University Environment. *EUROCON 2003; Computer as a Tool; The IEEE Region 8*, 2003, Band 2, S. 83-86
- Philipp 2008** Philipp, M.; Comparison of the Test-Driven Development Processes of Novice and Expert Programmer Pairs. Diplomarbeit. Universität Karlsruhe, Fakultät f. Informatik