Empirische Bewertung agiler Softwareentwicklung – Controlled Case Study

Ulrich Stärk

Institut für Informatik

FU Berlin

01.02.2007

Gliederung

- Einführung in empirische Methoden
- Agile Softwareentwicklung
- Controlled Case Study
- Erkenntnisse aus eXpert
- Controlled Case Study und CMMI

Empirische Methoden

- Empirie bedeutet Erfahrung
  - gewonnen durch Beobachtung, Versuche, Schlussfolgerung, usw.
  - nicht durch theoretische Überlegungen
- Beobachtung
  - Fallstudie (case study)
  - Befragung (survey)
- Versuche
  - Kontrolliertes Experiment
  - Quasi-Experiment
  - Benchmarking
- Schlussfolgerung
  - Software-Archäologie
  - Literaturstudie

Kontrolliertes Experiment

- Fokus auf einer oder mehreren Variablen, die gemessen werden sollen
  - Beispiel
- Ausschluss möglichst aller anderer Einflüsse
  - Kontrolle sowohl über das Messen als auch die Ausführung
- Durchführung in Labor- oder Universitätsumgebung
- hat hohe Glaubwürdigkeit
  - wird wahrscheinlich mit gleichem Versuchsaufbau gleiches oder ähnliches Ergebnis liefern
- aber unter Umständen niedrige Relevanz
  - Generalisierung schwierig
  - zu unrealistisch
- hohe Kosten

Case Study

- Beobachtung eines Vorgangs während er passiert
  - Beobachtung eines realen Projektes mit echtem Ergebnis
  - keine Kontrolle über die Ausführung (wohl aber über das Messen)
  - tiefgehendes Verständnis
- Durchführung am Ort des Geschehens unter echten Bedingungen

- hohe Glaubwürdigkeit
- geringe Relevanz aufgrund schwieriger Generalisierung
  - Einzigartigkeit der untersuchten Variablen
  - keine Kontrolle über unabhängige Variablen
- geringere Kosten als Kontrolliertes Experiment

Agile Softwareentwicklung

- Begriff Agile Methoden hervorgegangen aus Treffen von Experten die verschiedene „leichtgewichtige“ Softwareprozesse erfunden oder vorangetrieben haben
  - SCRUM, XP, Crystal, Adaptive Software Development, ...

- heraus kam das Agile Manifesto

Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Agile Softwareentwicklung

- Fokus also auf
  - funktionierender Software
  - Kommunikation
  - wechselnden Anforderungen
  - Zusammenarbeit mit dem Kunden

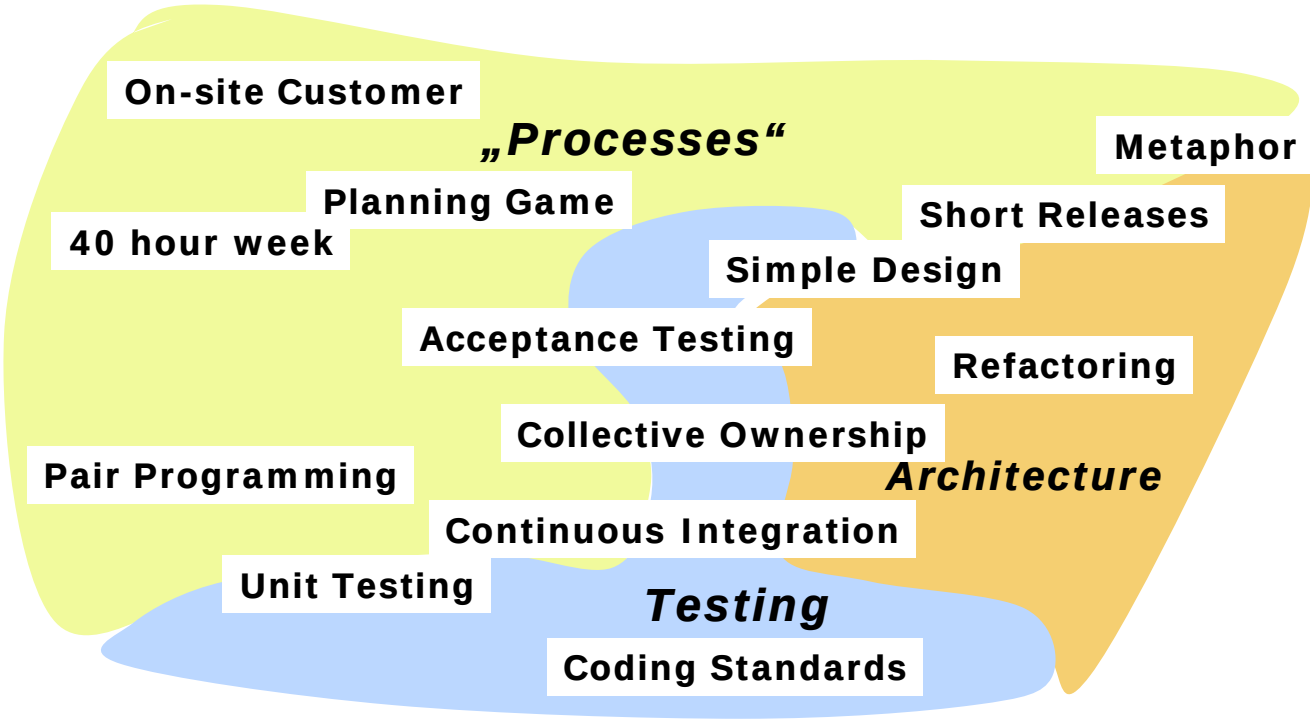
- Zusätzlich verschiedene Prinzipien
  - Kundenzufriedenheit durch „wertvolle“ Software
  - kurze Entwicklungszyklen mit häufigen Releases
  - Reflektion über die Effektivität des Teams
  - ...

Agile Methode: XP

- Menge von Regeln und Praktiken um fünf Ideen
  - Kommunikation
    - mangelnde Kommunikation Ursache für viele Fehler
    - Sorge für frühzeitige, häufige Kommunikation
  - Einfachheit
    - Baue die einfachste Lösung, die deine heutigen Anforderungen erfüllt
  - Feedback
    - es ist hilfreich, immer sofort oder sehr zeitnah Feedback zu Handlungen oder Plänen zu bekommen
  - Mut
    - Schaffe eine Umwelt, in der es möglich ist mutig zu sein
  - Respekt
    - zwischen Entwicklern
    - zwischen Entwicklern und Kunden

Agile Methode: XP

- Praktiken



- Graphik von K. Schneider aus Lutz Prechelt: Spezielle Themen der Softwaretechnik – Agile Methods: XP

Controlled Case Study

- um Agile Softwareentwicklung zu untersuchen ist ein Ansatz erforderlich, der die iterative und inkrementelle Natur der Softwareentwicklung in sehr kurzen Entwicklungszyklen berücksichtigt (P. Abrahamsson, O. Salo)
- Ansatz vereint das beste aus den empirischen Methoden
  - Glaubwürdigkeit durch Wiederholbarkeit (kontrolliertes Experiment)
  - tiefgehendes Verständnis (Case Study)
- und bietet zusätzlich die Möglichkeit sich an neue Anforderungen anzupassen bzw. sich zu verbessern
  - Action Research
  - Agiles Prinzip
- Controlled Case Study

Controlled Case Study

Freie Universität Berlin

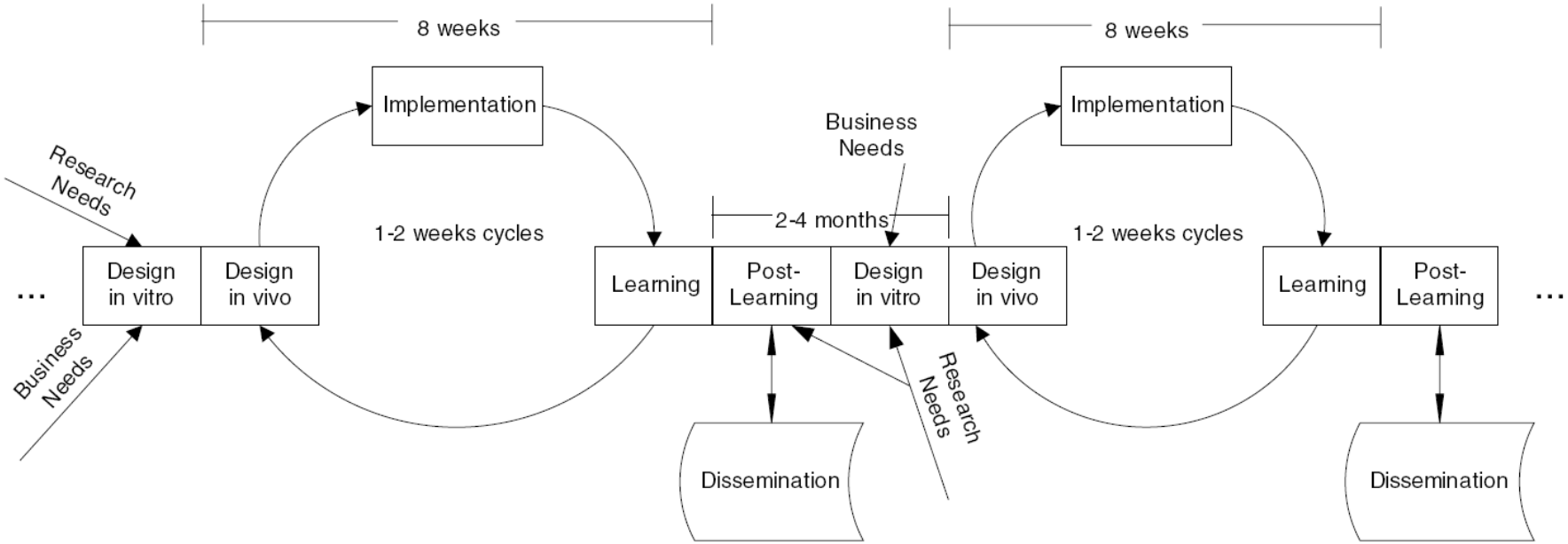
- erzeugt Qualitative und Quantitative Daten
  - über ein bestimmtes Forschungsziel
    - z.B. die Benutzung von XP
- liefert ein fertiges Softwareprodukt

- Quantitativ
  - z.B. Zeit, Größe, Defekte

- Qualitativ
  - Forschungstagebücher
  - Postmortem Sitzungen
  - Abschlussinterviews

Controlled Case Study

- Überblick



- angewandt auf eXpert-Projekt
  - Tool zur Datenverwaltung von Forschungsdaten

Design-Phase – Design in Vitro

- Dauer: ein bis wenige Monate höchstens
- Identifizierung von Forschungs- und **Geschäfts**anforderungen
  - Liste potentieller Themen
- Background Research
  - Themen mit größter Auswirkung
- Zielsetzung
- Background Research
  - derzeitiger Wissensstand über Ziel
- Festlegung des Kontexts
  - Forschungsumgebung
- Festlegung der Probanden
  - theoretische, nicht zufällige Stichprobe
  - also Wahl von bestimmten Probanden, nicht zufällig

Design in Vitro im eXpert-Projekt

Freie Universität Berlin

- Literaturdurchsicht von agilen/XP Methoden
  - förderte mögliche Forschungsthemen zu Tage
  - (Identifizierung von Forschungsanforderungen)
- Kartierung der bisherigen empirischen Bewertung von agilen/XP Prozessen
  - (Background Research)
- Ziel: „Evaluation des XP Softwareentwicklungsprozesses und Schaffung einer Grundlage für zukünftige Wiederholungen und stärker fokussierte Forschungsanstrengungen“
- Festlegung des Projektumfelds
- Wahl der Probanden
  - convenience sampling – Auswahl der am erfahrensten Studenten
  - Annahme: Forschung mit erfahrenen Studenten ist vergleichbar mit Forschung mit Teilnehmern aus der Industrie (Höst, Regnell, Wohlin, 2000, „Using Students as Subjects …“)

Design-Phase – Design in Vivo

- Festlegung der zu untersuchenden Variablen
  - ohne vorherige Theorie oder Hypothese: vorläufige Variablen
  - in weiteren Iterationen dann aktualisiert
- Planung der Datensammlung
  - Qualitative und Quantitative Daten
- Festlegung der Messtechnik
  - Datensammlungstools
  - Dokumentationsvorlagen
  - Schulungsmaterialien
  - Standards
  - Softwareentwicklungstools

Design in Vivo im eXpert-Projekt

Freie Universität Berlin

- Festlegung der Variablen / Planung der Datensammlung
  - Qualitative Daten
    - Gruppeninterviews
    - postmortem Sitzungen
    - Entwicklertagebücher
  - Quantitative Daten
    - Zeitaufwand in Minuten für XP-Praktiken und für Aufgaben
    - Größe im Hinblick auf Codezeilen
    - Defekte (systematisch kategorisiert und aufgezeichnet)
- Festlegung der Messtechnik
  - Datensammlungstools
  - Dokumentationsvorlagen
  - Standards
  - Schulungsunterlagen
  - Richtlinien
  - physische Einrichtung (Raum, Rechner, etc.)
  - technische Umgebung (Sprache, IDE, etc.)

Implementierungsphase

- sehr intensiv – nur 8 Wochen
- Vorbereitung der Studie
  - Schulung
  - Abschluss der Einrichtung der Meßinfrastruktur
    - Installation der Datensammlungstools
    - Entwurf der Vorlagen
- Durchführung der Studie
  - Sammlung der Forschungsdaten
- **Erfüllung der Geschäftsanforderungen**
  - funktionierendes Softwareprodukt
- Validierung der Daten
- Analyse und Visualisierung der Daten
  - Überlappung von Analyse und Erhebung

Lernphase – während einer Iteration

Freie Universität Berlin

- postmortem reviews / Analyse
  - Vorschläge zur Prozessverbesserung
  - Prozessverbesserung

- Interpretation der analysierten und visualisierten Daten aus der vorangegangenen Iteration
  - Ideen für die folgende In Vivo Design Phase
  - bspw. Vorschläge für weitere zu untersuchende Variablen
  - oder Verbesserung der Datensammlung

Lernphase – nach Projektende

- Gruppeninterview
  - Erkenntnisse und Erfahrungen der Entwickler
- Analyse und Visualisierung aller gesammelten Daten
- Schlussfolgerungen ziehen falls möglich
- Ergebnisse weitergeben
  - Generalisierung und Schlussfolgerungen
    - Weitergabe an Wissenschaftsgemeinschaft
    - Anwendung in der Industrie
- Identifizierung zukünftiger Forschungsansätze
  - Wiederholungen
  - neue Projekte für Controlled Case Study

- Verbesserung des Controlled Case Study Prozesses

Erkenntnisse über CCS aus eXpert

Freie Universität Berlin

- Design Phase
  - detaillierte Instruktionen wie coding standards hätten in die In Vitro Design Phase wandern sollen
  - die In Vivo Design Phase hat sich als wertvolles Werkzeug zur Verbesserung der Datensammlung herausgestellt
- Implementierungsphase
  - die Entwickler fanden es ermutigend, dass die gesammelten Daten täglich von mehreren Forschungsteilnehmern begutachtet wurden
  - die Einbindung der Entwickler als Mitforscher mit eigenen Forschungsinteressen wurde als Schlüssel zum Erfolg der Studie erachtet
  - es sollte festgehalten werden, dass der Sinn und Zweck des Projektes aber immer funktionierende Software ist
    - Daten(-sammlung) sollten dem Team nützen und es nicht an der Arbeit hindern

Erkenntnisse über CCS aus eXpert

Freie Universität Berlin

- Lernphase
  - Lernphase nach Projektende war zu kurz angesetzt
    - Gefahr der Vermischung mit Ergebnissen der Folgestudie
  - Verbesserung des Softwareprozesses konnte leider nicht stattfinden
    - Fokus lag auf postmortem reviews
    - keine harten Zahlen, nur Eindrücke der Teilnehmer
  - insgesamt aber hilfreich
    - Verbesserung des Entwicklungsprozesses
    - Verbesserung des Forschungsprozesses

Fazit der Autoren

- Controlled Case Study erhöht die Kontrolle – insbesondere über das Messen und die Ausführung – in realitätsnahen Projekten
- dennoch bleibt Problem der Generalisierung
- traditionelle Experimente und Case Studies zur Untermauerung weiterhin erforderlich
- jedoch kann Controlled Case Study durch das Erlangen von tiefgehendem Verständnis Methoden, Prozesse oder Techniken identifizieren, die in Experimenten näher betrachtet werden könnten
  - also exploratives Mittel

Ergebnisse aus eXpert

- 4 studentische Entwickler
  - alle ohne Erfahrung in XP
- sollten System zur Verwaltung von Forschungsdaten implementieren
- mit über 300 Benutzern
- innerhalb von 8 Wochen
  - wobei der Zeitrahmen fest stand lediglich über die Funktionen konnte verhandelt werden
- unter Anwendung von XP
  - On-Site-Customer wurde als sehr hilfreich erachtet
  - Akzeptanztests waren teilweise frustrierend, da von extra Test-Team des Auftraggeber durchgeführt, dessen Motivation stark nachließ

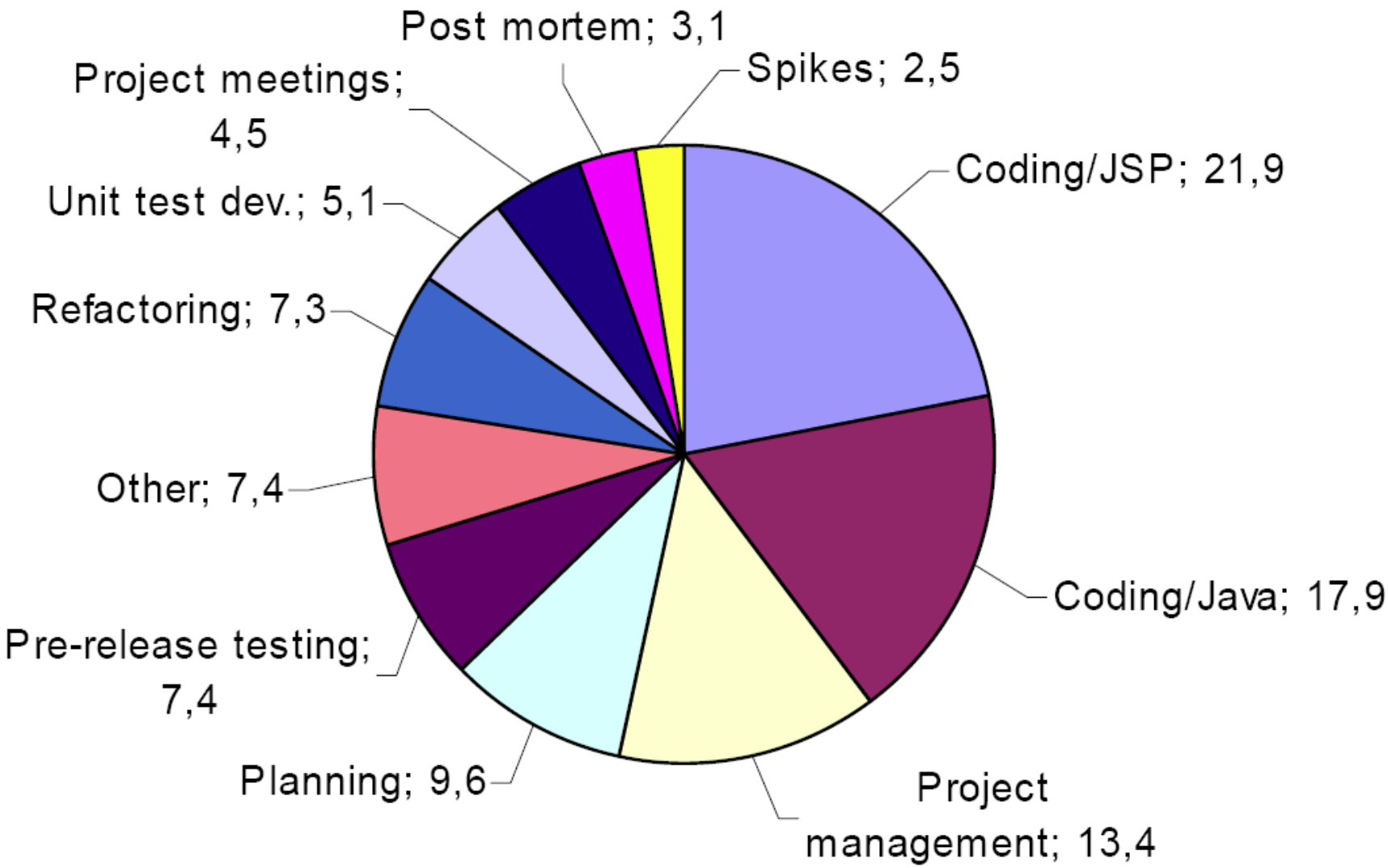
Ergebnisse aus eXpert

Freie Universität Berlin

Id	Collected data	Release 1	Release 2	Release 3	Release 4	Release 5	Correction release	Total
1	Calendar time (weeks)	2	2	2	1	1	0.4	8.4
2	Total work effort (h)	195	190	192	111	96	36	820
3	Task allocated actual hours	136 (70%)	95 (50%)	118 (61%)	51 (46%)	42 (44%)	27 (75%)	469 (57%)
4	# LOCs implemented in a release	1821	2386	1962	460	842	227	7698
5	Team productivity (loc/hour)	13.39	25.12	16.63	9.02	20.05	8.4	16.90
6	Code integrations (integrations/day)	8.1	10.1	7.9	10.5	8.2	8.5	8.9
7	Avg. time between integrations (minutes)	26	21	40	31	27	30	29
8	Avg. number of files per integration	1.7	2.4	3.1	2.6	3.0	3.0	2.6
9	# User stories implemented	5	9	9	4	3	4	34
10	# User stories postponed for next release	0	1	0	1	2	0	4
11	User story effort (actual, median, h)	10.1	8.3	7.6	5.9	5.2	2.8	6.8
12	User story effort (actual, max, h)	63.1	26.9	41.7	21.8	15.9	7.6	63.1
13	# Tasks defined	10	30	18	21	19	9	107
14	Task effort (actual, median, h)	11.7	2.9	5.9	1.7	2.6	0.7	2.7
15	Task effort (actual, max, h)	32.3	8.8	14.0	8.8	5.3	3.4	32.3
16	# post-release defects	4	5	4	4	11	-	28
17	Post-release defects/KLoc	2.19	2.10	2.04	8.70	13.06	-	1.43 (3.75)
18	# post-release enhancement suggestions made by testers	17	13	5	3	0	-	38
19	Pair programming (%)	81.7	76.3	73.0	78.8	54.2	90.4	75.9
20	Required customer involvement (%)	17.4	21.4	18.6	25.0	23.4	24.3	20.6
21	Rework costs (%)	-	8.7*	11.8	11.6	2.6	61.5	9.8

*includes also enhancements

Ergebnisse aus eXpert



Freie Universität Berlin

CMMI process areas in XP

Mark Paulk: *"Extreme Program'g from a CMM perspective"*, IEEE Software, Nov. 2001

- Level 2: Managed
  - + Requirements Mgmt
  - + Project Planning
  - + Project Monitoring&Control
  - - Supplier Agreement Mgmt
  - (Measurement and Analysis)
  - o (Process and) Product Quality Assurance
  - + Configuration Management
- Level 3: Defined
  - (Req's. Development)
  - + Technical Solution
  - (Product Integration)
  - + Verification
  - (Validation)
  - o Organizational Process Focus
  - o Organ'l Process Definition
  - o Organizational Training
  - + Integrated Project Mgmt.
  - (Risk Management)
  - (Decision Analysis and Resolution)
- Level 4: Quantitatively Manag'd
  - - Organizational Process Performance
  - - Quantitative Project Mgmt
- Level 5: Optimizing
  - - Organizational Innovation and Deployment
  - o Causal Analysis and Resolution

+ usually available
o avail. in reduced form
- usually mostly absent

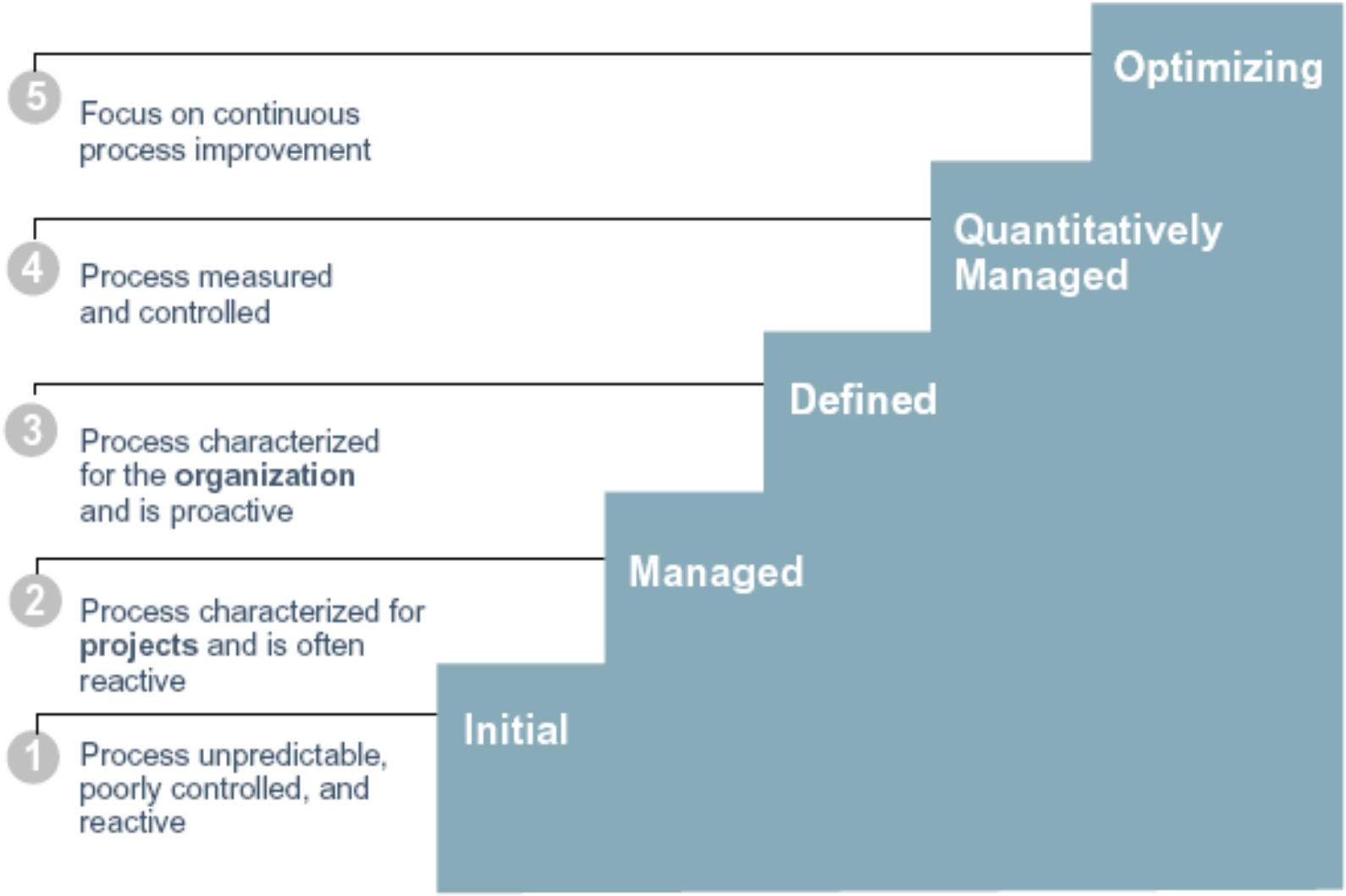
39 / 43

CMMI

- Capability Maturity Model Integration (-SW)
- Modell, das den Reifegrad einer Organisation beschreibt, bezogen auf ihre Prozesse

- Probleme mit Prozessen können in Prozessbereiche eingeteilt und ihre Vermeidung mit Zielen beschrieben werden
- zum Zwecke der Vermeidung gibt es bestimmte Praktiken
  - Beschreibung, WAS zu tun ist, nicht WIE
- zur Erreichung eines Reifegrads müssen die Ziele erfüllt werden, nicht aber unbedingt die Praktiken angewandt werden

CMMI Level



Freie Universität Berlin

Measurement and Analysis, MA

Freie Universität Berlin

- SG 1 Align Measurement and Analysis with Objectives
  - SP 1.1 Establish Measurement Objectives based on Needs
    - Very important step!
  - SP 1.2 Specify Measures
  - SP 1.3 Specify Data Collection and Storage Procedures
  - SP 1.4 Specify Analysis and Reporting Procedures
- SG 2 Provide Measurement Results
  - SP 2.1 Collect Measurement Data
  - SP 2.2 Analyze and Interpret Measurement Data
  - SP 2.3 Store Data and Results
  - SP 2.4 Communicate Results to Stakeholders

- MA is hardly useful on Level 2
  but is an important foundation for Level 3

Lutz Prechelt, prechelt@inf.fu-berlin.de

Prozessbereich MA

- SG 1 Align Measurement and Analysis with Objectives
  - SP 1.1 Establish Measurement Objectives based on Needs
  - SP 1.2 Specify Measures
  - SP 1.3 Specify Data Collection and Storage Procedures
  - SP 1.4 Specify Analysis and Reporting Procedures
    - Genau, was in der Design-Phase getan wird
- SG 2 Provide Measurement Results
  - SP 2.1 Collect Measurement Data
  - SP 2.2 Analyze and Interpret Measurement Data
  - SP 2.3 Store Data and Results
    - SP 2.1 – 2.3 wird genau so in der Implementierungsphase durchgeführt
  - SP 2.4 Communicate Results to Stakeholders
    - Lernphase
- Ziele aus MA werden durch Design-Phase, Implementierungsphase und Lernphase erfüllt

Freie Universität Berlin

Organizational Process Performance, OPP

Freie Universität Berlin

- SG 1 Establish Performance Baselines and Models
  - SP 1.1 Select Processes to Include in Performance Analysis
    - "What is important for us?"
  - SP 1.2 Establish Process Performance Measures
    - "Which measures tell us how good we are?"
  - SP 1.3 Establish Quality and Process-Performance Objectives
    - "How good do we need to be?"
  - SP 1.4 Establish Process Performance Baselines
    - "How good are we typically in process X?"
  - SP 1.5 Establish Process Performance Models
    - "How does process performance change when other observable factors change?"
    - Examples: System dynamics models, Reliability growth models, Complexity models

Lutz Prechelt, prechelt@inf.fu-berlin.de

Prozessbereich OPP

- SG 1 Establish Performance Baselines and Models
  - SP 1.1 Select Processes to Include in Performance Analysis
    - "What is important for us?"
  - SP 1.2 Establish Process Performance Measures
    - "Which measures tell us how good we are?"
  - SP 1.3 Establish Quality and Process-Performance Objectives
    - "How good do we need to be?"
  - SP 1.4 Establish Process Performance Baselines
    - "How good are we typically in process X?"
  - SP 1.5 Establish Process Performance Models
    - "How does process performance change when other observable factors change?"
    - Examples: System dynamics models, Reliability growth models, Complexity models
- Genau das geschieht in der Design in Vitro Phase wenn ein Ziel der Studie Prozessperformance wäre
  - Design in Vitro ist Mittel zur Erreichung des Ziels von OPP

Freie Universität Berlin

Organizational Innovation and Deployment, OID

Freie Universität Berlin

- SG 1 Select Improvements
  - SP 1.1 Collect and Analyze Improvement Proposals
  - SP 1.2 Identify and Analyze Innovations
  - SP 1.3 Pilot Improvements
  - SP 1.4 Select Improvements for Deployment
- SG 2 Deploy Improvements
  - SP 2.1 Plan the Deployment
  - SP 2.2 Manage the Deployment
  - SP 2.3 Measure Improvement Effects

- In contrast to Organizational Process Focus OPF, OID is based on quantitative management

Lutz Prechelt, prechelt@inf.fu-berlin.de

Prozessbereich OID

- SG 1 Select Improvements
  - SP 1.1 Collect and Analyze Improvement Proposals
  - SP 1.2 Identify and Analyze Innovations
  - SP 1.3 Pilot Improvements
  - SP 1.4 Select Improvements for Deployment
- Lernphase
- SG 2 Deploy Improvements
  - SP 2.1 Plan the Deployment
  - SP 2.2 Manage the Deployment
  - SP 2.3 Measure Improvement Effects
- Übergang Lernphase -> Design in Vivo Phase

Fazit

- Betrachtet man CCS als Mittel für SPI können damit Ziele in CMMI erreicht werden, die mit XP alleine nicht erreicht werden können

- nimmt man an, dass durch reifere Prozesse auch qualitativ hochwertigere Software geschaffen werden kann, könnte CCS zu qualitativ hochwertigerer Software beitragen

Referenzen

- P. Abrahamsson, O. Salo: *„Empirical Evaluation of Agile Software Development: A Controlled Case Study Approach“*, 5th International Conference on Product Focused Software Process Improvement, Japan, 2004
- P. Abrahamsson, J. Koskela: *„Extreme Programming: A Survey of Empirical Data from a Controlled Case Study“*, IEEE International Symposium on Empirical Software Engineering (ISESE 2004)
- Lutz Prechelt: Vorlesungsunterlagen zu „Spezielle Themen der Softwaretechnik“; Kapitel „CMMI“ und „Agile Methoden: XP“
- http://en.wikipedia.org/

Vielen Dank!