



Einführung agiler Verfahren – Ein Erfahrungsbericht aus zehn Jahren Praxis

Jens Coldewey (BDU)
Coldewey Consulting
Toni-Schmid-Str. 10 b
D-81825 München
Germany
Tel: +49-700-COLDEWEY
Tel: +49-700-26533939
Fax: +49-89-74995703
jens.coldewey@coldewey.com
<http://www.coldewey.com>

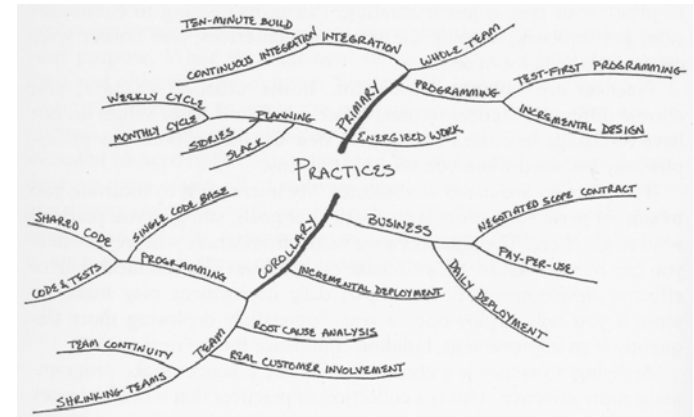
XP User Group Frankfurt,
29. Januar 2008



Agile Entwicklung funktioniert – Aber wie gestaltet man den Übergang?

Das Problem:

- Werte und Prinzipien machen den eigentlichen Unterschied – sind aber zur Einführung wenig hilfreich
- Einzelne Praktiken sind leicht(er) zu erlernen – erleichtern aber auch den Rückfall
- Manche Praktiken sind nur als Bündel sinnvoll



Beispiel I: „Prozessworkshop“ oder „Maria Montessori für Profis“

Ausgangslage

- Umfangreiches Vorhaben im Versicherungsbereich
- Zeitraum: 1998-2006
- Team: 10 Personen, „durchschnittliche“ Anwendungsentwickler
- Wenig Erfahrung mit OO
- ISO 9000 Prozessvorgaben





Beispiel I: Änderungsansatz - Verantwortung für Prozess an das Team

- Iterationen zur fachlichen Klärung und als politische Überlebenshilfe
- „Prozessworkshops“ als treibende Kraft
- Programmiertechnik „nebenbei“
- Testgetriebene Entwicklung als „U-Boot“
- ✓ Team begriff erstmals Prozess als Werkzeug
- ✓ Iterationen ermöglichten Überleben in „feindlicher“ Umwelt
- 💣 Unterstützung des PL ließ nach drei Jahren nach



Beispiel I: Lehren

- ✓ Eigenverantwortung macht die Einführung nachhaltig erfolgreich -> Retrospektiven!
- ✓ Identifikation des Teams ist wichtiger, als genaue Einhaltung der Regeln
- ✓ Umfangreiche Prozessvorgaben sind kein Hinderungsgrund
- ✓ Technische Hürden kann man überwinden – Politik und Management sind entscheidend
- ☞ In fetten Jahren wird die Grundlage des Erfolgs gerne vergessen



Beispiel II: Agile Homöopathie

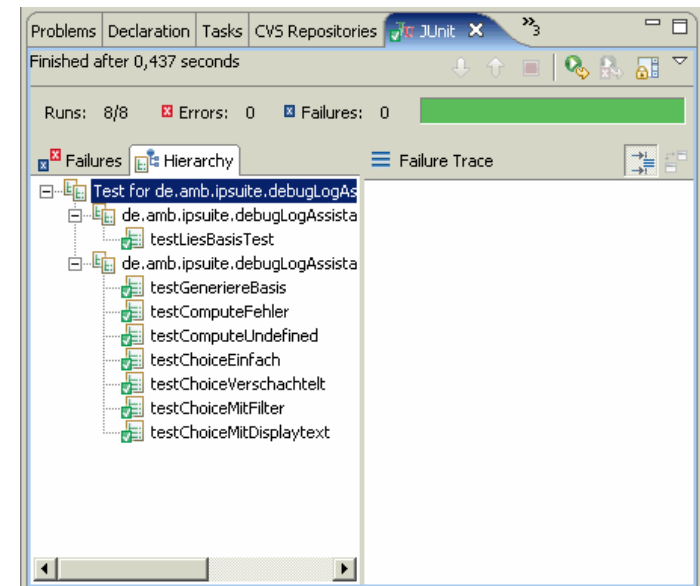
- Mittelständisches Produkthaus
- Zeitraum: Drei Monate im Jahr 2005
- Monatliche Iterationen und agile Planungstechnik als „Motor“ (2 Tage pro Monat)
- Kein Kontakt zu Entwicklern

❗ **Keine nachhaltige
Besserung**



Beispiel III: Testgetriebene Entwicklung

- „Feuerwehrprojekt“ bei Finanzdienstleister
- Zeitraum: 2001-2003
- Testgetriebene Entwicklung als Motor
- Keine agilen Managementtechniken
- ✓ Nachhaltige Besserung für die Mitarbeiter
- 💣 Keine nachhaltige Besserung für die Organisation





Beispiel III: Lehren

- ✓ Testgetriebener Ansatz wirkt auch ohne agiles Umfeld
- ✓ TDD-Infektionen sind nicht heilbar
- 💣 Rein technischer Ansatz bleibt organisatorisch ohne Wirkung

Beispiel IV: Spaghetti a la Chef

Ausgangssituation:

- Mittelständisches Produkthaus (60 MA, 15 Entwickler)
- Große Codebasis in C++, Codequalität „fragwürdig“
- Keine adäquate Organisationsstruktur
- Hierarchische Kultur, geprägt durch Konkurrenz und Resignation





Beispiel IV: Änderung der Organisation

- Einführung dreistufiger Inkremente
- Regelmäßige Retrospektiven
- Übergabe der Planungs- und Prozessverantwortung an das Team
- ✓ **Besserung der Termintreue**
- ✓ **Übernahme der Verantwortung**
- ✓ **Teilweise Besserung der Moral**
- 💣 **Realistischer Blick führte zu weiterer Demotivation**



Beispiel IV: XP Pilotteam zum Aufbau neuer Codebasis

- Eigenes Teilteam mit eigenem Coach
- Ziel: Testgetriebener Aufbau neuer Codebasis (.NET/NHibernate)
- Passives Denken und mangelnde technische Qualifikation
- 💣 Kosten/Nutzen nach drei Monaten nicht mehr vertretbar
- 💣 Teilprojekt eingestellt
- ✓ Akzeptanztests mit FitNesse übernommen



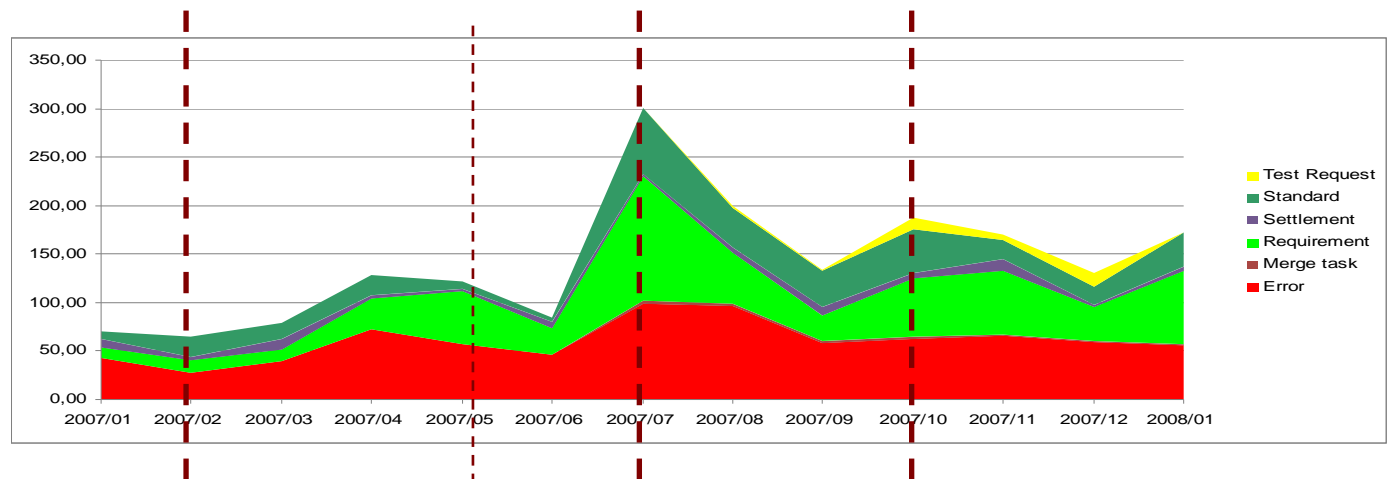
Beispiel IV: Versuche, die alte Codebasis in den Griff zu bekommen

- Erste Versuche, Unit-Tests einzuführen gescheitert
- Build-Automatisierung zäh
- Aufbau von Akzeptanztests mit FitNesse
- Refactoring mit Ziel Entkopplung und ersten Strukturen
- ✓ Problembewusstsein und Codequalität verbessert
- ✓ Erster Entwickler versteht Refactoring
- 💣 **Noch kein Rol für Entwickler**



Beispiel IV: Zwischenbilanz

- ✓ Wertschöpfungsanteil in der Entwicklung steigt seit vier Monaten nachhaltig
- ✓ Kunden spüren deutliche Änderung
- ✓ Langfristig Planung wird realistischer
- ✓ Korrekturmaßnahmen werden zielgerichteter
- 💣 Zwei wichtige Kunden konnten nicht gehalten werden





Beispiel IV: Lehren aus dem Projekt

- Bei großer Codebasis zunächst schnelle Erfolge (und Vertrauen) durch organisatorische Änderungen erarbeiten
- Schnelle technische Erfolge setzen Mindestmaß an Änderungsbereitschaft, Eigenverantwortung und Qualifikation voraus
- Messbare Erfolge auf altem Code brauchen Monate bis Jahre





Meine persönlichen Schlussfolgerungen I

- Retrospektiven sind der Kern agiler Verfahren – der Rest ergibt sich „von selbst“
- Zunächst braucht man mindestens zwei Tage pro Woche Unterstützung
- Es braucht mindestens sechs Monate, bis die Umstellung ausreichend Impuls besitzt
- Erste Erfolge stellen sich nach ein bis zwei Iterationen ein - nach ein bis zwei Jahren „kann“ die Mannschaft nicht mehr anders





Meine persönlichen Schlussfolgerungen II

- Einmal eingeführt sind die Verfahren robust gegen „Abnutzungerscheinungen“
- Ohne agiles Management geht das Team verloren
- Agiles Vorgehen und testgetriebene Entwicklung gehören zusammen
- Altlasten sind vor allem Lasten





Wie es weiter geht...

können Sie in meinem Blog verfolgen:

<http://blog.coldewey.com>

