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

XPDays 2007,
Karlsruhe, 22. November 2007

Der Start agiler Verfahren aus sehr persönlicher Sicht…

From: ACockburn@aol.com

Date: Wed, 20 Feb 2001 20:53:10 EDT

Subject: Re: Column on Lightweight Processes

To: jens.coldewey@coldewey.com

Jens,

That’s great news. Concerning the name of the column, hurry if not run to http://agilemanifesto.org – we found the word “agile” is much more appropriate.

Alistair

 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Vier Jahre später die “Firmung”

Remember Standish, the company that publishes those "Chaos" reports that tend to show that the software crisis is for real? Well, they're at it again.

The Standish 2004 Chaos study shows a decline in project success rates and (of course) an increase in failures. From 2002 to 2004, we have:

- 15% failure rate increasing to 18%
- 34% success rate declining to 29%
- cost overruns increasing from 43% to 56%
- schedule overruns increasing from 82% to 84%
- features/functions requirements met, declining from 67% to 64%

Why do things seem to be getting worse? Standish says it's because those 2004 projects were bigger and more expensive. They also cited lack of user involvement and project executive support. The study went on to report that the top success factors for IT projects are

- user involvement
- executive support
- clear business objective
- experienced project manager
- minimal scope and requirements
- standard tools and infrastructure
- skilled resources in place
- a formal methodology and financial management

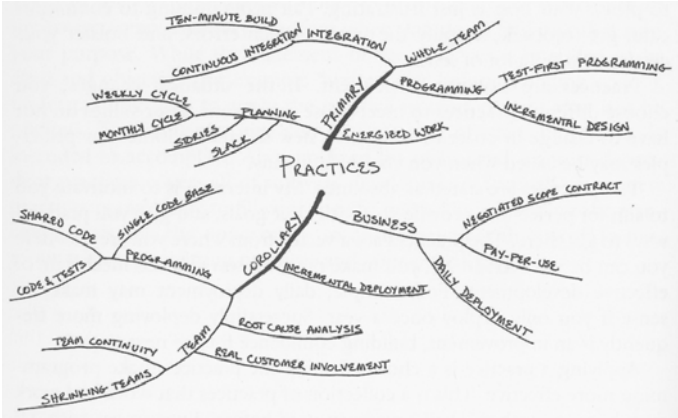
This analysis of the 2004 Chaos findings is based on data from the publication e-Wise Solutions, Dec. 21, 2004.

 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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



 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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

Folie 5; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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

 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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

 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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

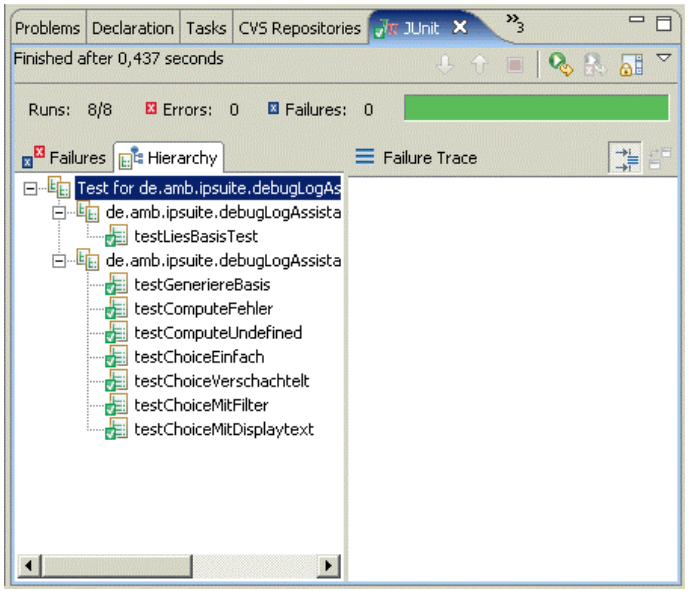
 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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



 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Beispiel III: Lehren

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

 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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

 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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

 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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

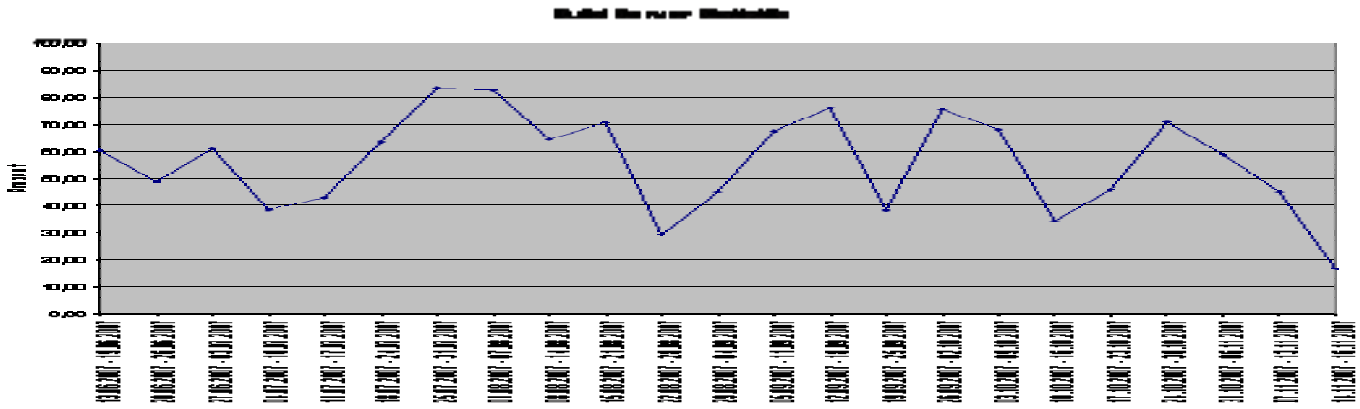
 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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

✓ Problembewusstsein und Codequalität verbessert

💣 Noch kein RoI



Folie 14; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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

 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

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

 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Meine persönlichen Schlussfolgerungen II

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

 © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved