

Sind wir nicht alle ein bisschen agil?



*Dipl.-Inform. Tammo Freese
xpdays, Karlsruhe, 22. November 2004*

Das Manifest der agilen Softwareentwicklung

- *Ähnliche Werte bei
XP, ASD, Crystal, DSDM, FDD, Scrum, ...*
- *Idee: Gemeinsame Werte formulieren*
- *Treffen im Februar 2001*
- *Ergebnis: Manifesto of Agile Software
Development (<http://agilemanifesto.org>)*

Manifesto (1)



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

Manifesto (2)

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

Manifesto (3)



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

*Kent Beck, Mime Beedle, Arie van Bennekum, Alistair Cockburn,
Ward Cunningham, Martin Fowler, James Grenning,
Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern,
Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber,
Jeff Sutherland, Dave Thomas*

*Copyright 2001, the above authors
this declaration may be freely copied in any form,
but only in its entirety through this notice.*

Was ist für Softwareentwicklung nicht wichtig?

- *Individuen und ihre Interaktionen?*
- *Prozesse und Werkzeuge?*
- *Lauffähige Software?*
- *Umfassende Dokumentation?*
- *Zusammenarbeit mit dem Kunden?*
- *Vertragsverhandlung?*
- *Eingehen auf Änderungen?*
- *Verfolgen eines Plans?*

Wer findet...



- *Prozesse und Werkzeuge wichtiger als Individuen und ihre Interaktionen?*
- *umfassende Dokumentation wichtiger als Lauffähige Software?*
- *Vertragsverhandlung wichtiger als Zusammenarbeit mit dem Kunden?*
- *Verfolgen eines Plans wichtiger als Eingehen auf Änderungen?*

Fragen



- *Ist das Agile Manifest konkret genug zur Definition agiler Softwareentwicklung?*
- *Sind alle agil?*

Beispiele aus der Praxis (1)

Individuen und ihre Interaktionen sind wichtiger als Prozesse und Werkzeuge:

- *“Wie teilt ihr denn das Projekt auf?”*
- *“Schon wieder zu zweit vor einem Computer?”*
- *“Sie können nicht so ohne weiteres einen Raum reservieren”*
- *“Wir können doch nicht einfach den Rechner und die Tische umstellen”*

Beispiele aus der Praxis (2)

Individuen und ihre Interaktionen sind wichtiger als Prozesse und Werkzeuge:

- *“Nur weil Sie meinen, dass wir mit dem Werkzeug produktiver wären, geben wir keine 200 \$ aus.”*
- *“Und wer soll das Projekt durchführen?”*
“Wir denken, vier Entwickler schaffen das in einem Jahr.”

Beispiele aus der Praxis (3)

Lauffähige Software ist wichtiger als umfassende Dokumentation:

- *“Wir müssten erstmal die Anforderungen genau aufschreiben, bevor wir loslegen.”
(in einem zeitkritischen 6-Wochen-Projekt)*
- *“Heißt das jetzt, wir implementieren das Feature ohne das Anforderungsdokument zu aktualisieren? Nur nachdem, was der Kunde gesagt hat?”*

Beispiele aus der Praxis (4)

Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlung:

- *Ein Kunde: “Da müssen wir uns jetzt nur noch beeilen, Sie zu beauftragen.”*
- *“Ist das nun ein Bug oder ein Change Request?”*

Beispiele aus der Praxis (5)

Eingehen auf Änderungen ist wichtiger als Verfolgen eines Plans:

- *“Das haben wir bereits entschieden und jetzt wird das auch so gemacht!”*
- *“Jetzt von X nach Y zu wechseln, ist ein zu großes Risiko!”*
 - $(X, Y) = (\text{Java 1.4}, \text{Java 5})$
 - $(X, Y) = (\text{Access}, \text{Oracle})$
 - $(X, Y) = (\text{Eclipse 2.1}, \text{Eclipse 3.0})$

Fragen und Antworten

- *Ist das Agile Manifest konkret genug zur Definition agiler Softwareentwicklung?*

Ja.

- *Sind alle agil?*

Zumindest ein bisschen.

Was können wir tun, wenn wir ein
bisschen mehr Agilität wagen wollen?

*Um uns zur geliebten Seite der Agilität hin
zu verbessern, müssen wir uns erstmal
mit der ungeliebten Seite beschäftigen.*

Prozesse und Werkzeuge



- *Beschaffungsprozesse überarbeiten, um Teamentscheidungen bei der Auswahl von Soft- und Hardware zu berücksichtigen*
- *Regelungen für Räumlichkeiten anpassen, so dass Teamarbeit besser möglich ist*

Umfassende Dokumentation



- *Ziele schriftlich festlegen*
- *Ergebnisse im Verlauf dokumentieren*
- *Erfahrungsberichte schreiben*
- *Retrospektiven durchführen,
Ergebnisse festhalten
(Buchtipps: N. Kerth:
Project Retrospectives)*

Vertragsverhandlung

- *Mit klassischen Festpreisverträgen ist agile Softwareentwicklung schwierig bis unmöglich*
- *Bei der Vertragsverhandlung möglichst die Rechtsabteilung des Kunden einbinden*
- *Im Vertrag Vereinbarungen treffen, die Änderungen erleichtern*

Verfolgen eines Plans

- *Ziele exakt definieren, Erfolg/Misserfolg muss entscheidbar sein*
- *Plan zum Erreichen der Ziele erstellen*
- *Dabei Änderungsmuster berücksichtigen
(Buchtipps: L. Rising und M. L. Manns:
Fearless Change:
Patterns for Introducing New Ideas)*

Zusammenfassung

- *Das Agile Manifest wirkt schwammig, ist es aber nicht.*
- *Bei genauem Hinschauen bleibt oft nur wenig Agilität übrig.*
- *Um mehr Agilität zu erreichen, müssen wir wagen, eine zeitlang weniger agil auszusehen.*

Fragen

?