



Sozial und Agil – Ist das die Zukunft der Softwareentwicklung?

Sebastian Graf, PROMATIS software GmbH
Ettlingen, 9. Juni 2015

- ◆ Über den Referenten / Einleitung
- ◆ Grundlegendes zum Thema agile Softwareentwicklung
- ◆ Praxiserfahrungen
- ◆ Studien zum Thema agile Softwareentwicklung
- ◆ Agile Stolpersteine
- ◆ Diskussion, Q & A

- ◆ 1987 – 1993: Studium Informatik an der Universität Karlsruhe (TH) (heute KIT)
- ◆ 1994 – 1995: Wissenschaftlicher Mitarbeiter Fraunhofer Institut (IITB), Karlsruhe
- ◆ 1995 – 1999: PROMATIS AG, Karlsbad, Deutschland
- ◆ 1999 – 2002: PROMATIS Corp., San Ramon, CA, USA
- ◆ 2003 – heute: PROMATIS software GmbH, Ettlingen, Deutschland
 - Division Manager *Project Standards and Methods*
- ◆ Leiter der Special Interest Group Methodik und Architektur der DOAG

Sozial und Agil


Einstieg



Playstation

Armata







Notes® 9

Social Edition

L-GHUS-968PRV

Licensed Materials - Property of IBM Corp. © Copyright 1985, 2013 IBM Corporation and by other(s). IBM, das IBM Logo, Lotus und Notes sind Marken von IBM Corporation, die in zahlreichen Rechtssystemen weltweit registriert sind. Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken von Oracle und/oder deren verbundene Unternehmen. Built on Eclipse ist eine Marke der Eclipse Foundation, Inc. Dieses Programm unterliegt den Bedingungen der Lizenzvereinbarung, die mit dem Programm ausgeliefert wird. Diese Lizenzvereinbarung kann in einem Programmverzeichnisordner oder in einer Bibliothek gespeichert sein, der/die als "License" oder "Non-IBM-License" angegeben ist, sofern zutreffend. Die Lizenzvereinbarung kann auch gedruckt vorliegen. Lesen Sie diese Lizenzvereinbarung bitte sorgfältig durch, bevor Sie das Programm verwenden. Durch die Nutzung des Programms erkennen Sie die Bedingungen dieser Vereinbarung an.

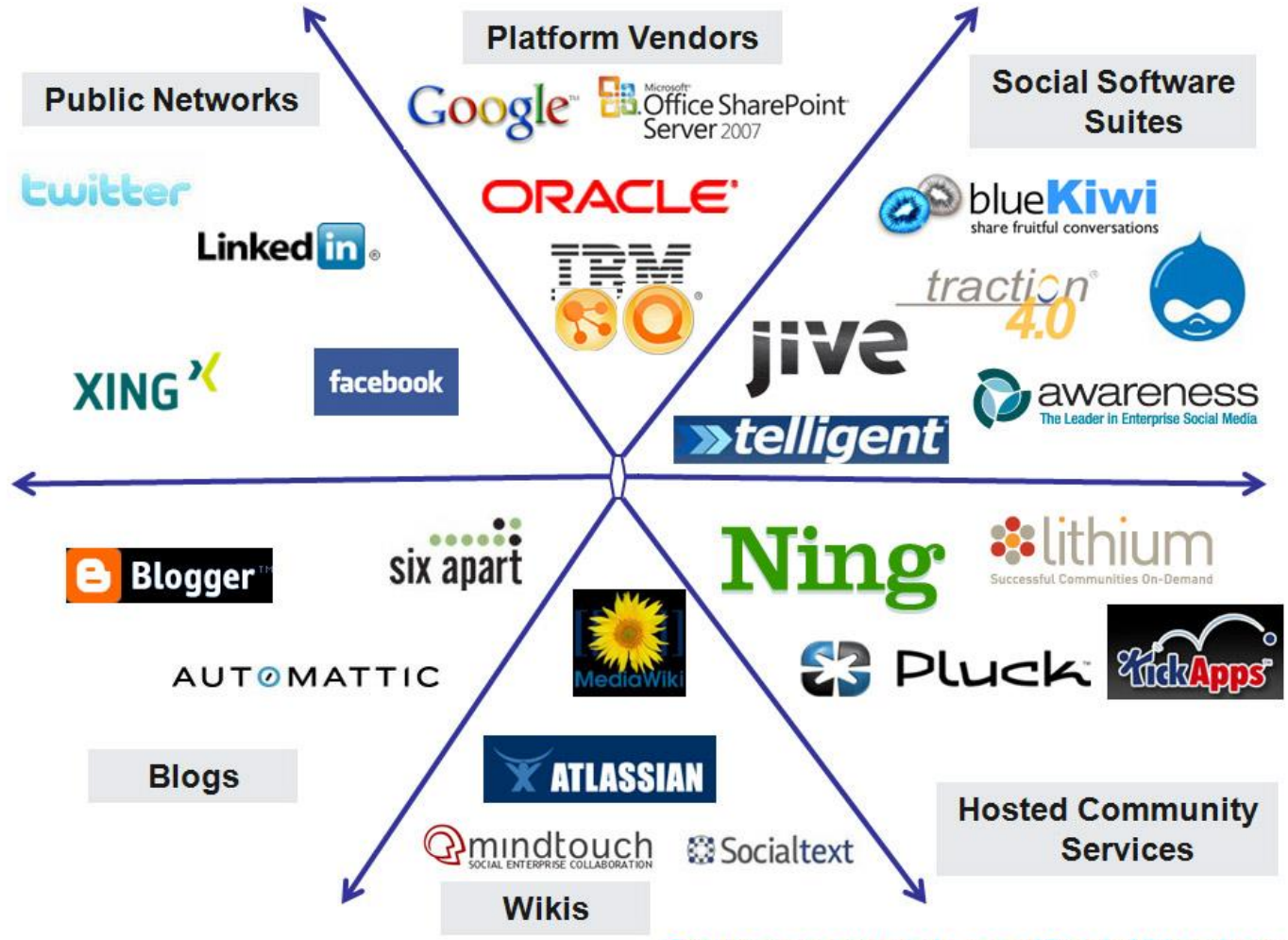
Das Kennwort wird abgefragt



BUILT ON
eclipse

Sozial und Agil

Einstieg



Source: cmswatch.com/Social/Vendors

◆ Ich bin ...

- ... kein Scrum Master
- ... kein Extreme Programmer
- ... kein Kanban Post-It Fetischist
- ... kein Verfechter agiler Softwareentwicklung
- ... kein Gegner agiler Softwareentwicklung

- ... lediglich ein naiver Beobachter diverser agiler Projekte!

- ◆ Wir erschließen bessere Wege Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:
 - **Individuen und Interaktionen** mehr als Prozesse und Werkzeuge
 - **Funktionierende Software** mehr als umfassende Dokumentation
 - **Zusammenarbeit mit dem Kunden** mehr als Vertragsverhandlung
 - **Reagieren auf Veränderung** mehr als das Befolgen eines Plans
- ◆ Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.

Quelle: <http://agilemanifesto.org/iso/de/>

Sozial und Agil

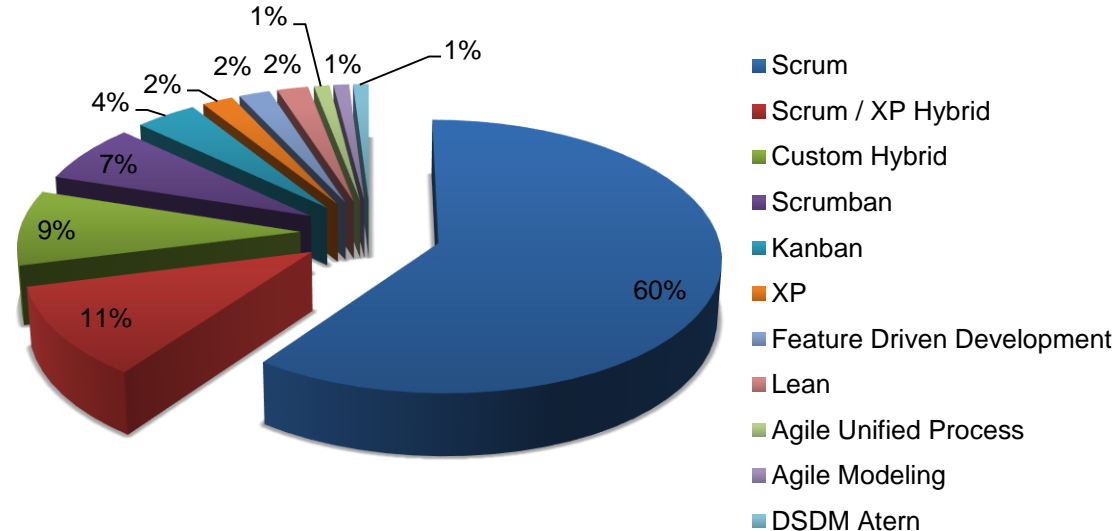
Grundlegendes – Die zwölf Prinzipien

1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
2. Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projekts täglich zusammenarbeiten.
5. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
6. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.

7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell.
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.

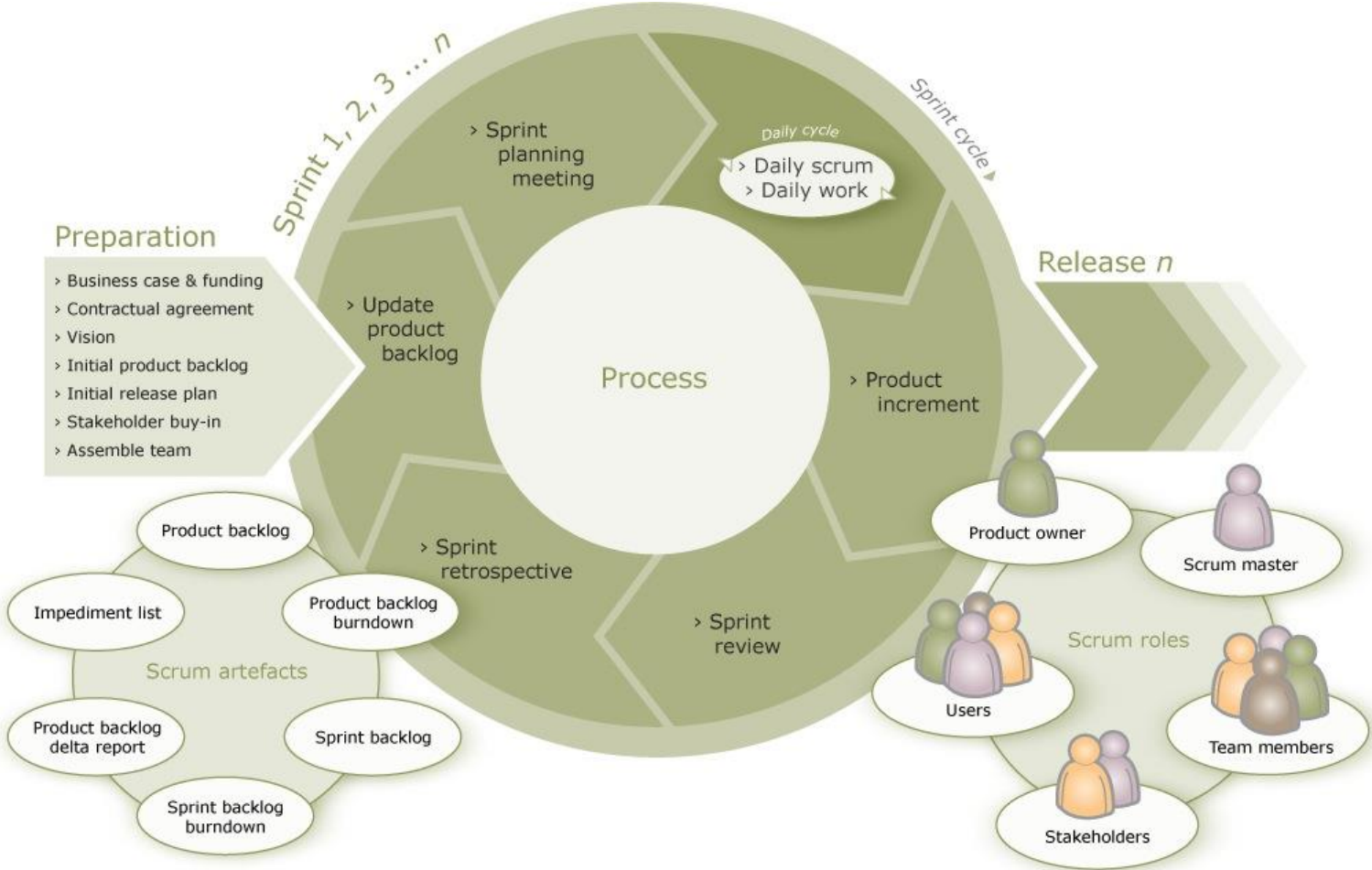
- ◆ Scrum
- ◆ Scrum / XP Hybrid
- ◆ Custom Hybrid
- ◆ Scrumban
- ◆ Kanban
- ◆ XP
- ◆ Feature Driven Development
- ◆ Lean
- ◆ Agile Unified Process
- ◆ Agile Modeling
- ◆ DSDM Atern

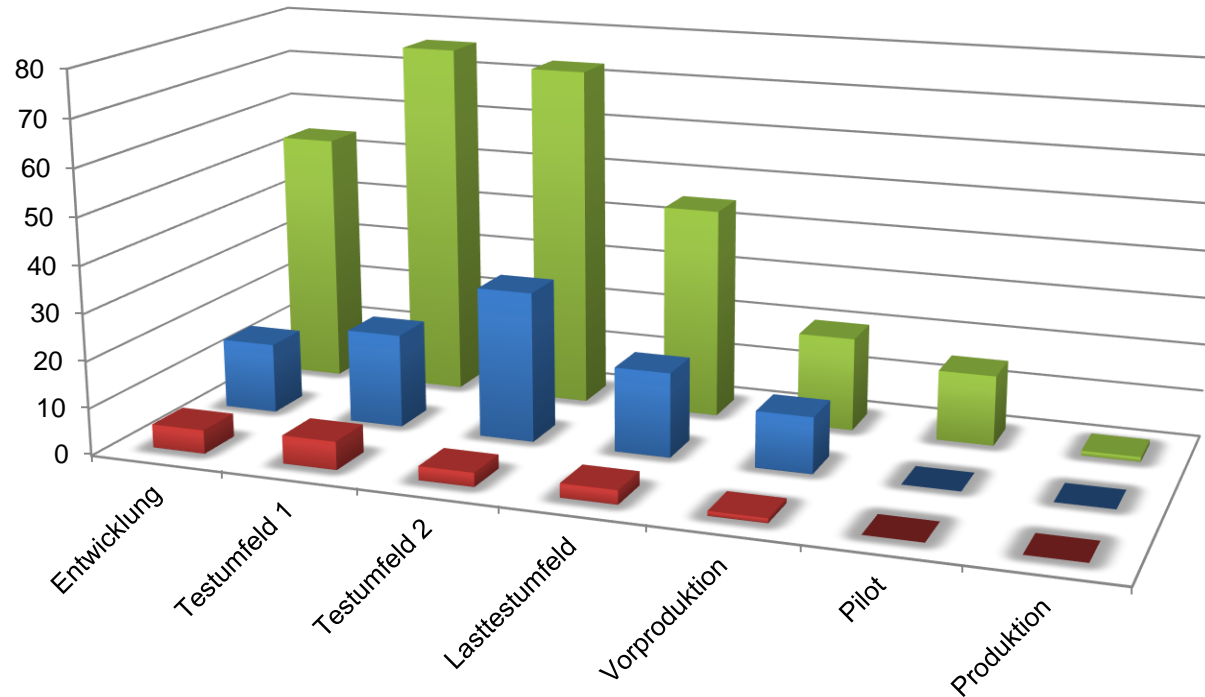
Verbreitung agiler Methoden



Sozial und Agil

Grundlegendes – Die Scrum Methode

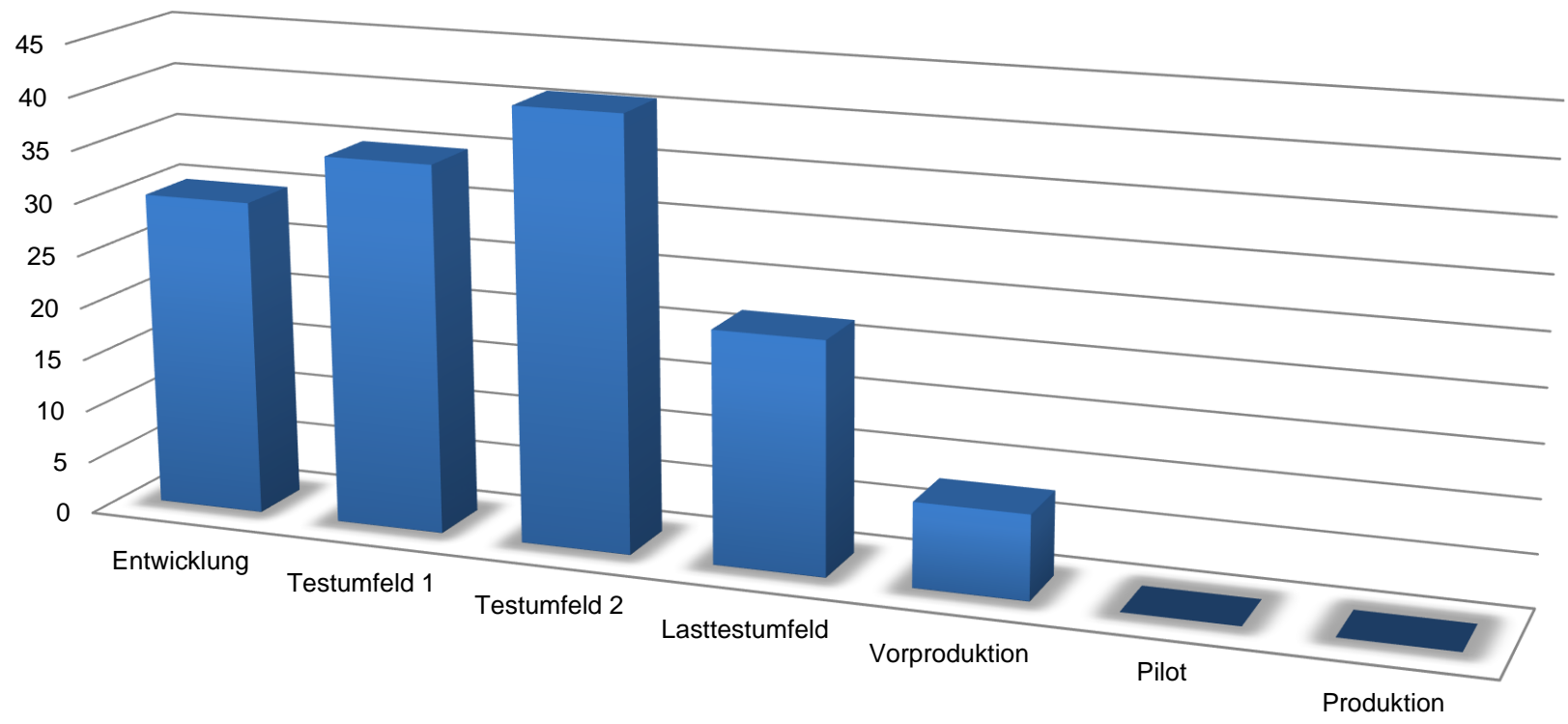




- Schwere Änderungen (DROP mit Strukturänderung)
- Mittlere Änderungen (DROP ohne Strukturänderung)
- Einfache Änderungen (ALTER)

Sozial und Agil

Praxiserfahrung – Eigene Beobachtungen



■ Änderungen an der gleichen Tabelle an unterschiedlichen Tagen

- ◆ Es gibt unzählige pseudowissenschaftliche Studien, die den Erfolg agiler Methoden „belegen“.
- ◆ Es gibt einige, wenige wissenschaftlich fundierte Studien:
 - *...found no change in overall productivity when comparing results from before and after the introduction of an agile process...*
 - *...the XP teams being, on average 337% more productive than the V-model teams. However, this productivity gain was due to the XP team delivering 3.5 times more lines of code without delivering more functionality...*
 - *...a further limitation that was reported...was the lack of attention to design and architectural issues...*
 - *...team members are less interchangeable in agile teams, which has consequences for how projects are managed...*

Quelle: T. Dyba, T. Dingsoyr, *Empirical studies of agile software development: A systematic review*, Inform. Softw. Technol. (2008), doi:10.1016/j.infsof.2008.01.006

- ◆ Umstellung auf agiles Verfahren wird unterschätzt
- ◆ Vorbereitung eines Projekts wird unterschätzt
- ◆ Planungs- und Abgrenzungsaufwand wird unterschätzt
- ◆ Agiles Manifest wird nicht richtig verstanden
- ◆ Missstände im Projekt werden als Eigenarten der agilen Methode verstanden

- ◆ Vorteile agilen Vorgehens bei der Code-Entwicklung wird auf alle anderen Artefakte übertragen
- ◆ Agilität für Datenmodelle / DB Artefakte
 - DB Modell ist das Fundament der Anwendung
 - Querbeziehungen einer punktuellen Änderung sind fatal
 - DB Änderungen sind selten kapselbar
- ◆ Agilität für Prozesse
 - Analyse eignet sich gut für agile Vorgehensweise
 - Von der Implementierung eines Prozesses mit agilen Methoden sei abgeraten
 - Know-how über Gesamtprozess ist erforderlich

- ◆ Binden Sie das Management als Stakeholder für ein agiles Projekt auf jeden Fall von Anfang an mit ein! Agile Methoden dürfen nicht geduldet, sondern müssen erwünscht sein.
- ◆ Begreifen Sie Agilität nicht fälschlicherweise als Revolution von unten!

- ◆ Agile Methoden sind kein Ersatz für herkömmliche und planungsorientierte Ansätze, sondern als Ergänzung bzw. Alternative zu verstehen.
- ◆ Wähle weise!

...Insbesondere wird dann der Einsatz agiler Verfahren problematisch, wenn ein Projekt klar (vorher) definierte Anforderungen erfüllen muss und engen Zeit- oder Budgetvorgaben unterliegt. Hier bieten die klassischen, ingenieurmäßigen Vorgehensmodelle mit klar definierten Phasen große Vorteile. Agile Verfahren eignen sich hingegen gut bei weichen und wenig ausformulierten Anforderungen bzw. einem hohen Maß an externen Störfaktoren/Marktveränderungen...

Quelle: http://de.wikipedia.org/wiki/Agile_Softwareentwicklung

- ◆ Entwickler fühlen sich in agilen Projekten wohler
- ◆ Kein Platz für Kommunikationsmuffel und Knowledge Hider
- ◆ Zusammenarbeit zwischen Entwickler und Fachbereich
- ◆ Agilität fordert Selbstorganisation
- ◆ Agilität erzwingt kontinuierlich nachprüfbare Ergebnisse in kurzen Abständen

- ◆ Agile Methoden geraten dann in Verruf, wenn Missstände im Projekt mit der agilen Methode begründet werden:
 - Die Dokumentation ist qualitativ schlecht und nicht nutzbar.
 - Am Datenmodell der Anwendung werden ständig Änderungen an der gleichen Stelle vorgenommen (... Wir nähern uns der Lösung durch exploratives Agieren...).
 - Die Vorabplanung war so ungenau und hat Querbeziehungen nicht berücksichtigt, was zur häufigen Revision von Sprint Ergebnissen führt.
 - Prozesse werden vorab nicht verstanden, weil sich das ja „im Projekt ergibt“.
 - Sprints wurden vor dem Projektstart falsch geschnitten.
 - Die beteiligten Personen sind überfordert bzw. nicht akkurat für agile Methoden ausgebildet.



Sebastian Graf

Dipl.-Inform.

Division Manager Project Methods & Standards

PROMATIS software GmbH

Pforzheimer Str. 160

76275 Ettlingen

Tel. +49 7243 2179 0

Fax +49 7243 2179 99

E-Mail: sebastian.graf@promatis.de

Web: www.promatis.de