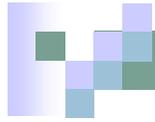


Adaptive Software Development

von

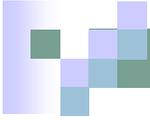
James Highsmith III.

Christoph Eberle

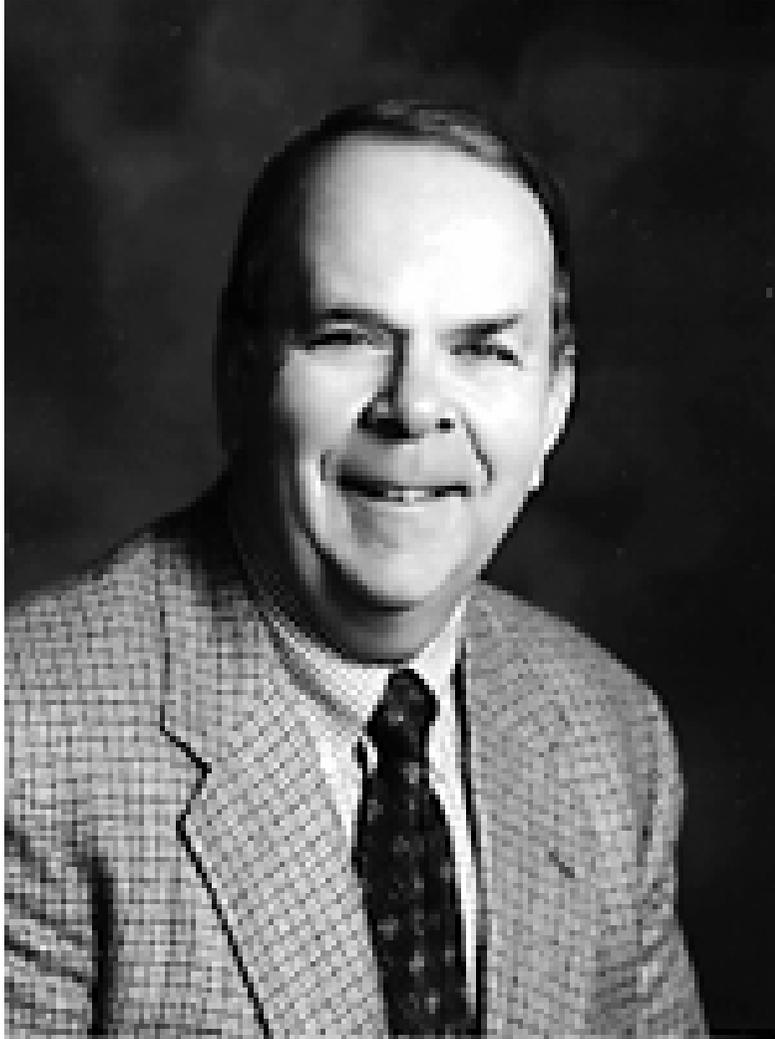


Übersicht

- Jim Highsmith
- Entstehung
- Abgrenzung zu anderen agilen Methoden
- Adaptive Software Development
- Beurteilung



James Highsmith III.

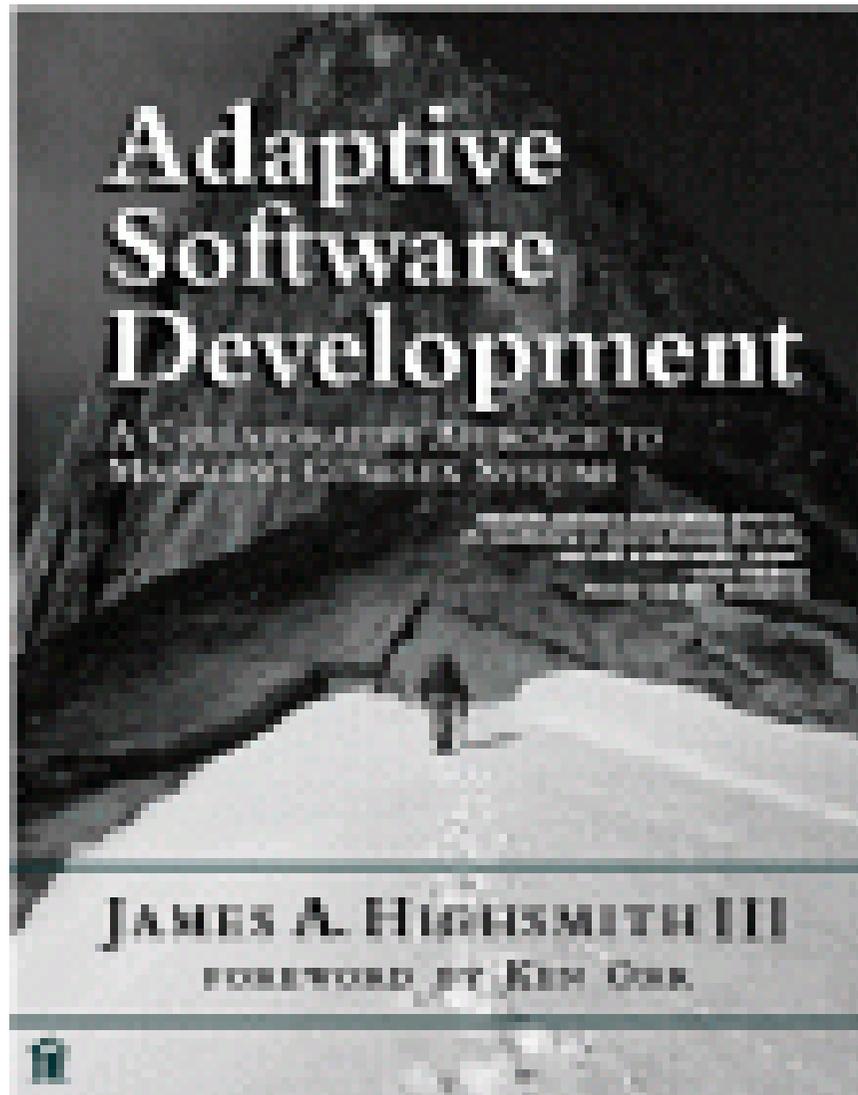


- President of Information Architects
- Director of e-Project Management Practice for the Cutter Consortium
- Editor of e-business Application Development
- RADical Software Development
- B.S. in Electrical Engineering
- M.S. in Management



Inhalt

- 320 Seiten lang alter Wein in neuen Schläuchen
- Man nehme:
BWL Grundstudium + Bauknecht's IM +
ein wenig Trivialpsychologie
= Adaptive Software Development



- Ein Softwareprojekt ist wie eine Steilwand, sie ist nur mit viel Mut und Ausdauer zu erklimmen.
blablablablabla...



Was ist Adaptive Software Development?

- One of a growing number of alternatives to traditional, process-centric software management methods with a focus on people, results, minimal methods and maximum collaboration. It is geared to the **high speed and high change** of today's e-business projects.

Highsmith



Geschichte des Softwareengineering

- 70er Jahre: Softwareentwicklung als geheimnisvolle Kunst weniger Gurus
ABER: frühe Fehler sind teuer!
- 80er Jahre: Totale Planung (Boem, Function Points, Cocomo, Wasserfall)
ABER: nicht alles voraussehbar



Geschichte des Softwareengineering

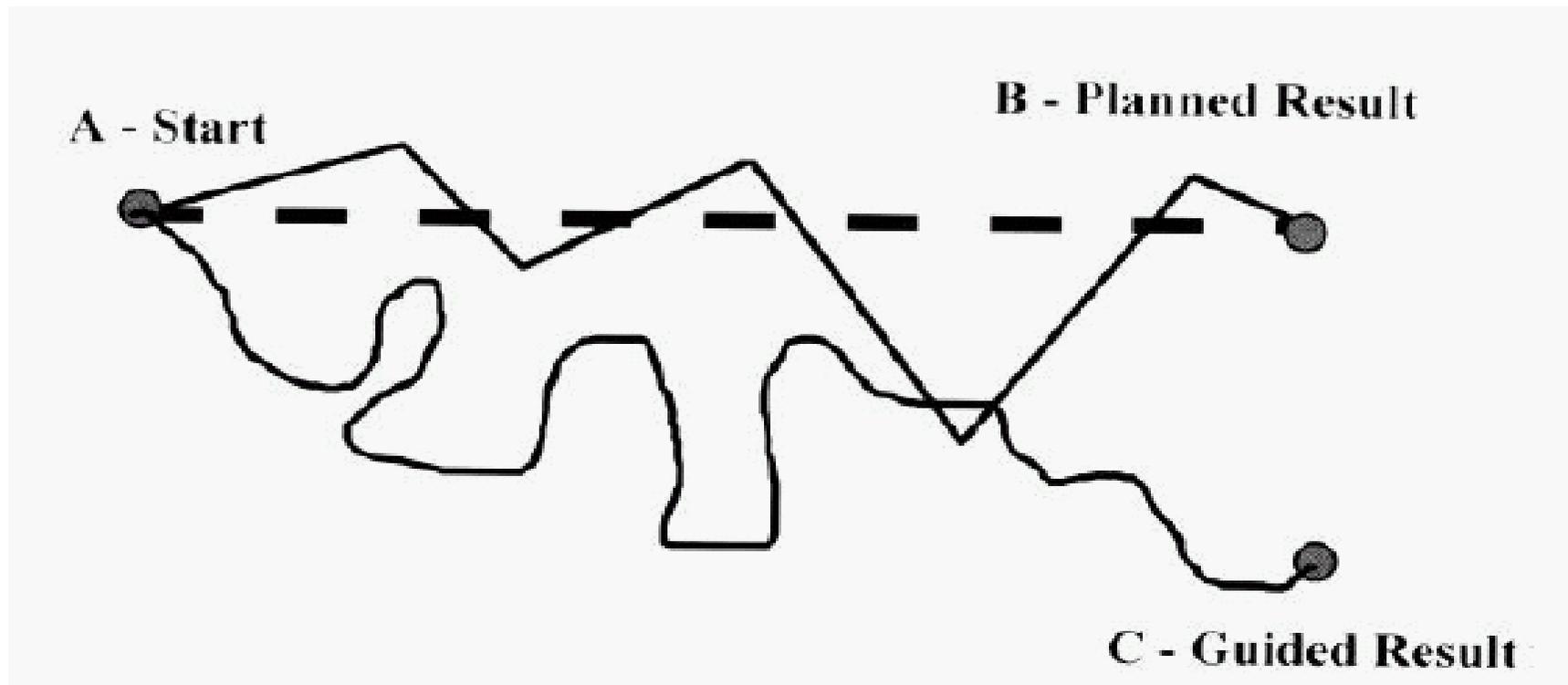
- späte 80er: Iterative Phasen (die selbst keinem Wandel mehr unterliegen)
ABER: Alles ändert sich immer schneller
- Heute: Internet, E-Commerce, Just-in-Time, virtuelle Firmen

Wirtschaftlich korrekte Software



Coldeway Consulting 2001

Was ist das Ziel?



James A. Highsmith III: *Adaptive Software Development - A Collaborative Approach to Managing Complex Systems*, Seite 43



Geschichte des Softwareengineering

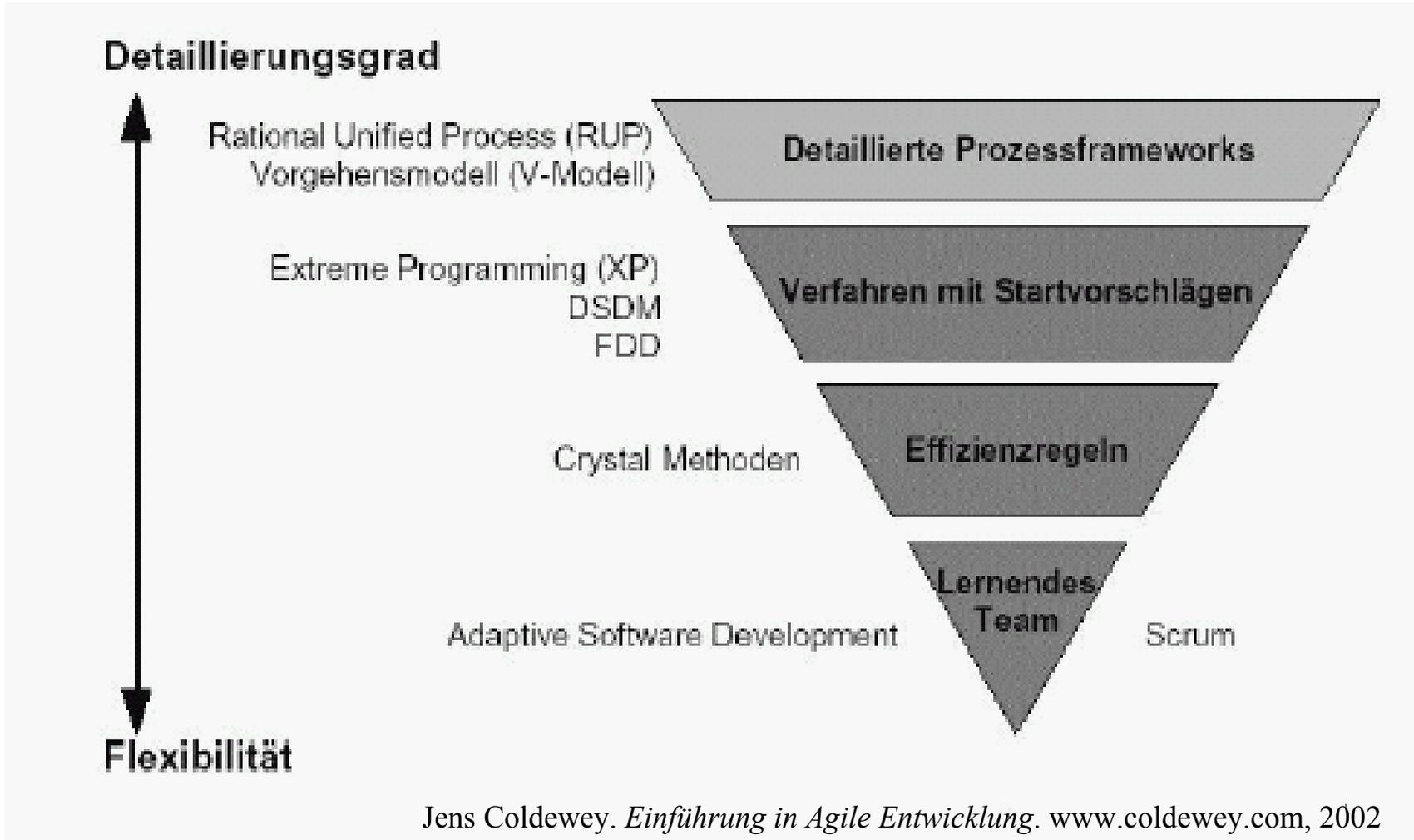
- 90er Jahre:
starke Verbreitung der Personal Computer
- Vollkommen neue Phänomene:
Sekretärin programmiert VBA-Excel-
Makro, welches später zur
Intranetworkflow-Lösung ausgebaut wird

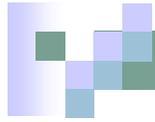


Geschichte des Softwareengineering

- Gesucht:
etwas Besseres als Adhoc-Vorgehen!
- Die Lösung:
Agile Methoden

Einordnung





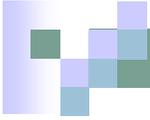
Grundaussagen

- Bürokratieabbau → schlanke Methoden
- Menschen wieder im Zentrum
- Änderungen gehören zum Alltag



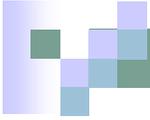
Was Adaptive Software Development *nicht* ist:

- ASD produziert *nicht* kurzfristige Wegwerfsoftware
- ASD bedeutet *nicht* Anarchie in der Unternehmung
- ASD verzichtet *nicht* auf Planung
- ASD verzichtet *nicht* auf Entwicklungswerkzeuge



Komplexe Adaptive Systeme

- bisher Mensch als Zahnrad im Getriebe der Unternehmung
- survival of the fittest in einer Welt von „decreasing returns“
- NEU „arrival of the fittest“ in einer Welt von „increasing returns“ (2000)



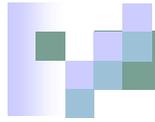
Komplexe Adaptive Systeme

- Anleihe bei autonomen Agenten Theorie
- z.B. Vogelschwarm
- wichtigstes Ziel: „EMERGENZ“



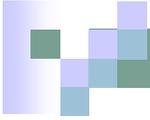
Emergenz

- Ein guter Projektleader leitet sein Team am Rande des Chaos zum Ziel
- $1+1=3$
- Alles fließt...



Adaptive Development Framework

- Adaptive Conceptual Model
- Adaptive Development Model
- Adaptive (Leadership-Collaboration) Management Model

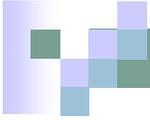


Adaptive Conceptual Model

- Projekt Mission:
 - gemeinsame Passion
 - Patterns statt Regeln
 - Trade Offs → Prioritäten!!!

Highsmith:
„pure rubbish!“

~~gutes Projekt:
-Budget
-Zeit
-Umfang
-Qualität~~



Die 3 kritischen Fragen

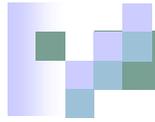
- *Um was geht es bei diesem Projekt?*
- *Warum sollen wir das Projekt durchführen?*
- *Wie sollen wir das Projekt durchführen?*

(Project Vision, Data Sheet, Specification Outline)



Software-Qualität

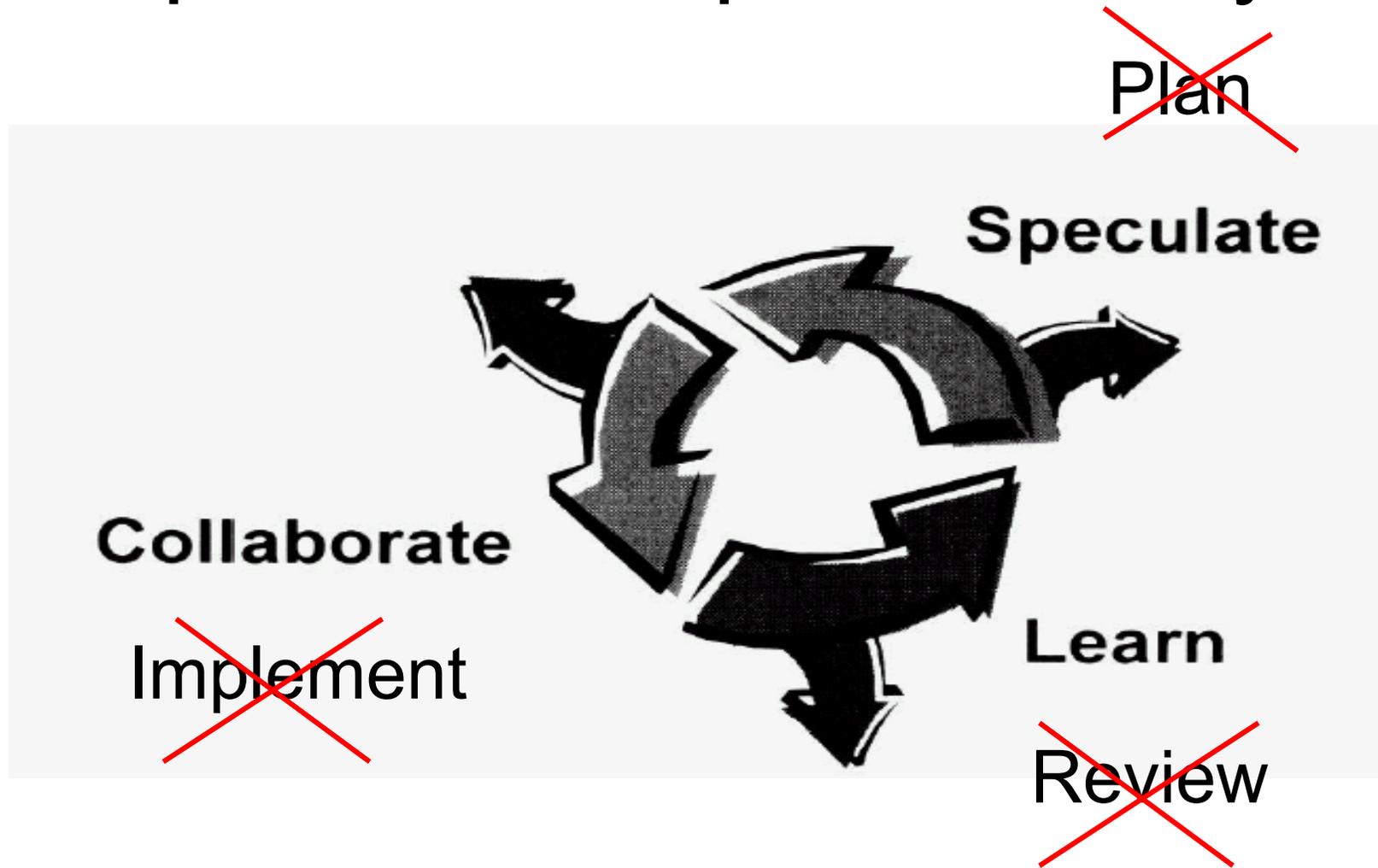
- Akademiker: „bitte sauber!“
- Firmenboss: „schnell und günstig!“
- Highsmith:
„gerade gut genug, etwa so wie Microsoft“

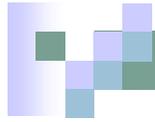


Adaptive Development Lifecycle

	low Speed	high Speed
low Change	Waterfall	RAD
high Change	Spiral + “evolutionary“	Adaptive

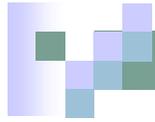
Adaptive Development Lifecycle





Hilfe, Chaos!!!

- Highsmith: „ein gutes Team weiss, wann ein wenig Strenge angebracht ist.“

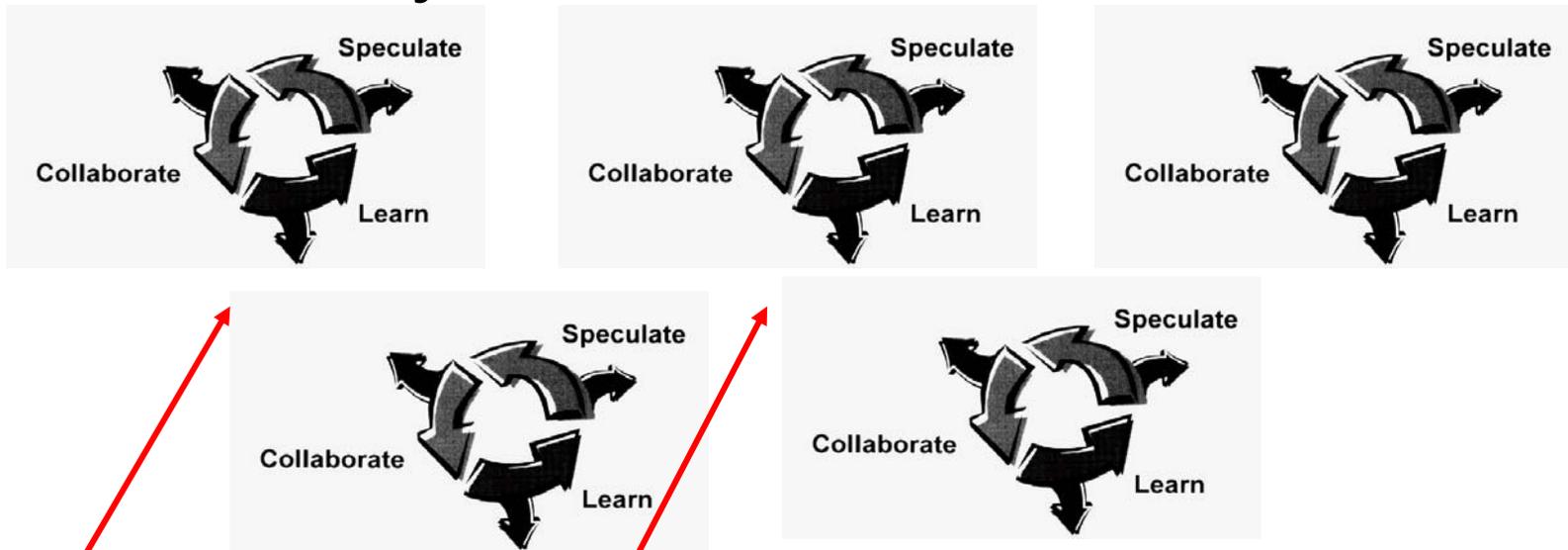


Adaptive Development Framework

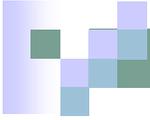
- Adaptive Conceptual Model
- **Adaptive Development Model**
- Adaptive (Leadership-Collaboration) Management Model

Adaptive Development Model

- mehrere Zyklen:

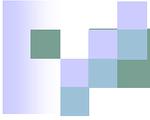


Achtung: Ueberschneidungen



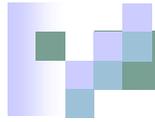
Inhalt eines Zyklus

- Jeder Zyklus liefert ein brauchbares Stück Software!
- nicht taskorientiert, sondern komponentenorientiert
- Abgrenzung zum Wasserfall: „nicht „flow“, sondern „cyclen“ ist wichtig



Time Boxing

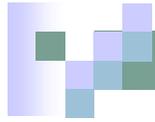
- absolut harte Zeitrahmen
- bei Nichterreichen
→ schmerzliche Trade-Offs
(aber gut für Wissensaustausch der Mitarbeiter)
- Ausnahmen nur mit Zustimmung aller Beteiligten



Konkretes Projekt

■ Schritt 1

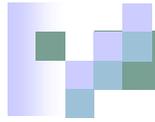
- Machbarkeitsstudie
- Basisarchitektur
- Umfang gemessen in Functionpoints
- Exekutivsponsor



Konkretes Projekt

- Schritt 2
 - Projektdauer
 - Personalressourcen

- Schritt 3
 - Anzahl Zyklen und Time Boxing



Konkretes Projekt

- Schritt 4

Welche Komponenten werden von
welchen Zyklen produziert



Zyklus 1

- totale Verunsicherung und Ärger
- Ziel: Möglichst schnell ein erstes Mal zum Kunden

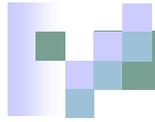
- Resultat: Menus + Anfragemasken,
Middleware + Technologie,
Namensstandards + Designgrundsätze



Zyklus 2

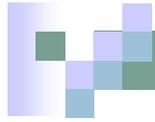
- Neugewonnene Anforderungen umsetzen
- komplexe Algorithmen ausprogrammieren

- Ziel: Kunde soll ein erstes Mal selbst am Produkt rumspielen können



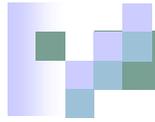
Zyklus 3

- Verfeinerung
- akzeptabler Fehlerlevel
- Technische Seite komplett



Zyklus 4

- Anwenderdokumentation
- Installationsroutinen
- Trainingsprogramme



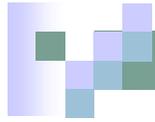
Adaptive Development Framework

- Adaptive Conceptual Model
- Adaptive Development Model
- **Adaptive (Leadership-Collaboration) Management Model**



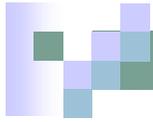
Highsmith's Umfrage

- Was schätzen Sie am wenigsten im Projektalltag?
- Resultat:
Meetings, Präsentationen,
Dokumentation und Zusammenarbeit mit
anderen Gruppen

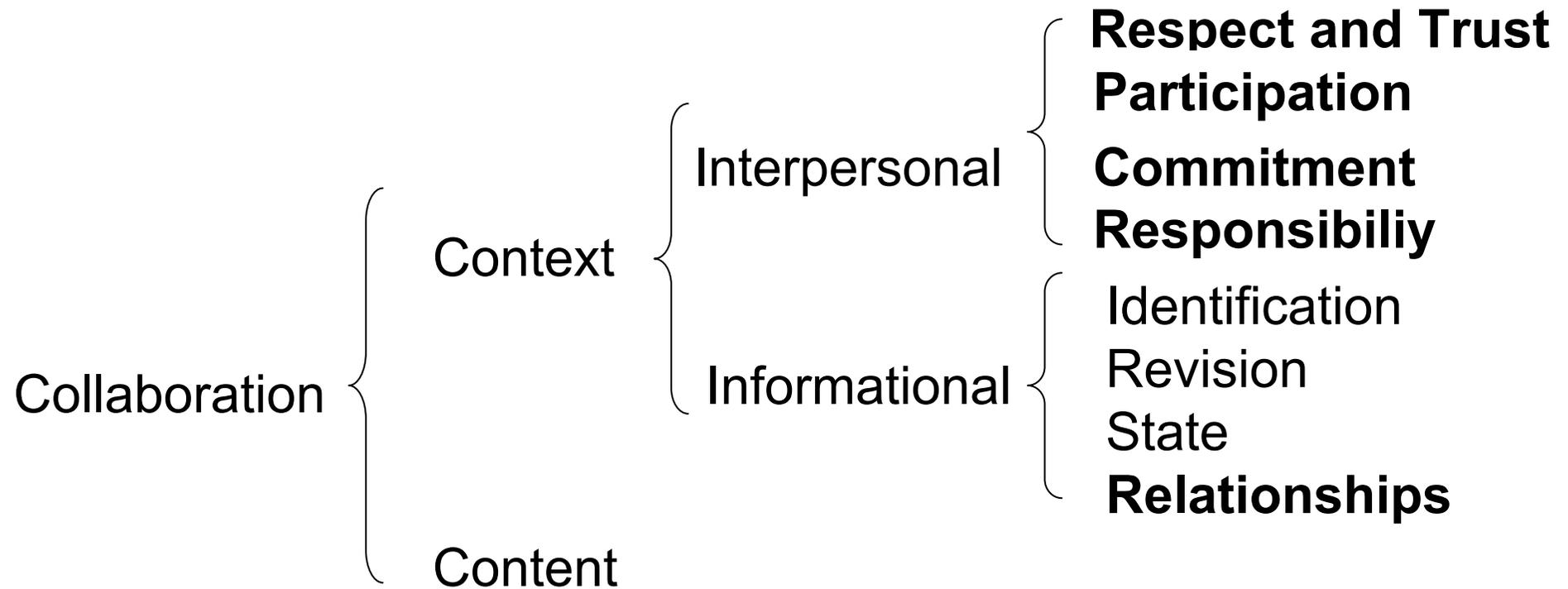


Teamwork

- DAS Buzzword der letzten Jahre
- Leadership statt Befehl
- Zusammenarbeit statt Kontrolle



Collaboration





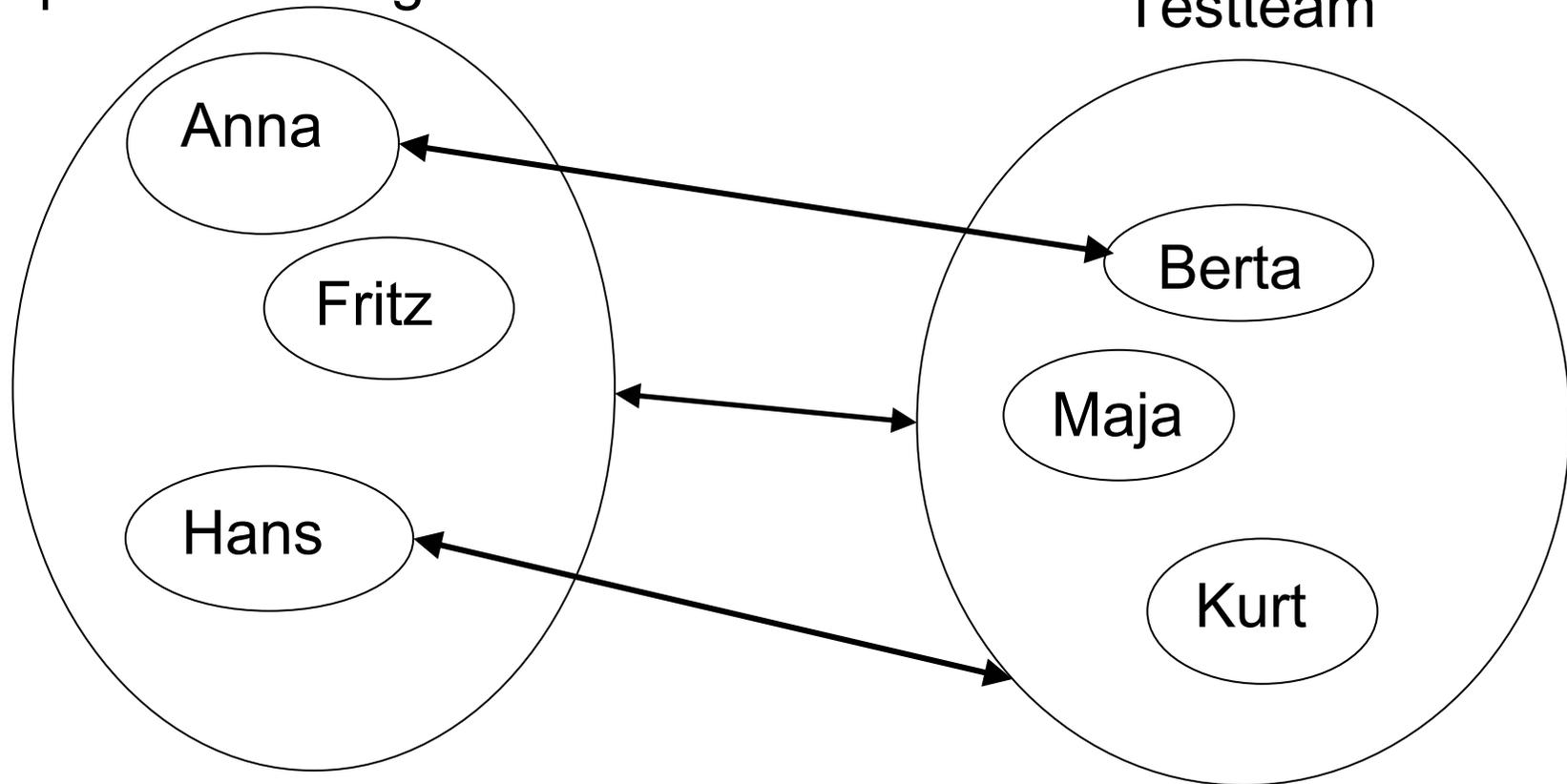
„rich context“ für gute Beziehungen

- E-mail
- Chat
- gemeinsames Zeichnen auf einem virtuellen Whiteboard
- Videokonferenz

1:1-Beziehungen zwischen Teammitgliedern

Implementierungsteam

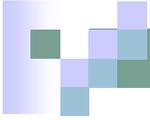
Testteam





Der „Facilitator“

- In Meetings nimmt sich der Leader zurück und überlässt dem Facilitator die Führung
- Facilitator sind Kommunikationsprofis
- Einzige Aufgabe:
Planen und Durchführen von Meetings



Contra Highsmith:

- Grosse Gefahr, dass einfach aus Bequemlichkeit auf unangenehme Aufgaben verzichtet wird
- Typisches Beratergeschwafel ohne ein konkretes Beispiel zu geben



Pro Highsmith:

- Denken statt Prozesssturheit
- Mensch im Mittelpunkt
- hübsche (ungewollte)
Zusammenfassung aller Hypes der
letzten Jahre ;-)