



V-Modell XT und Agilität

Dirk M. Sohn
sohn at oio.de

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0

Java und XML

) Akademie)

- **Schulungen, Coaching, Weiterbildungsberatung, Train & Solve-Programme**

) Beratung)

- **Methoden, Standards und Tools** für die Entwicklung von **offenen, unternehmensweiten Systemen**

) Projekte)

- **Schlüsselfertige Realisierung** von Software
- **Unterstützung laufender Projekte**
- **Pilot- und Migrationsprojekte**

Idee des Vortrags



- Die Mutter der Agilität ist das Agile Manifest
- Das V-Modell XT erlaubt eine „Agile Projektdurchführungsstrategie“
- Was ist das eine, und was ist das andere?

3

- **Agilität - Manifest und XP**
- Iterative Inkrementelle Entwicklung
- Wieviel Prozeß und Papier muß denn sein?
- Grundbegriffe V-Modell XT
- Die Agile Projektdurchführungsstrategie

Kurze Orientierung, wie XP zum Agilen Manifest paßt
welche Welt mit „Agilem Vorgehen“ verbinden kann

Einige Ideen, wann das nicht reicht und wie man damit
umgehen kann. **AUßerdem 2. Player beim V-Modell**

Eine **Bedürfnispyramide** für Prozeßbedarf

Wie tickt das V-Modell, Ergebnisse einschätzen können

Ergebnisse - **Einordnung des Agilitätsbegriffs** des V-Modells

3 Minuten um

Agiles Manifest



Manifesto for Agile Software Development

(Beck, Fowler, Cockburn, uvm., 2001)

- **Einzelpersonen und Interaktionen** wichtiger als Prozesse und Werkzeuge
- **Laufende Systeme** wichtiger als umfangreiche Dokumentation
- **Zusammenarbeit mit dem Kunden** wichtiger als Vertragsverhandlungen
- **Fähigkeit auf Änderungen zu reagieren** wichtiger als Verfolgen eines Plans

5

Agile Entwicklung kennt 3 Hebel: Team, Software, Prozesse - weniger auf Praktiken als **kulturelle Werte**

1. Software **ohne Prozess aber nie ohne gutes Team**

Gute Teams **organisieren sich von selbst** Einheitliche wiederholbare Prozesse **paßt nicht zu Projekten**

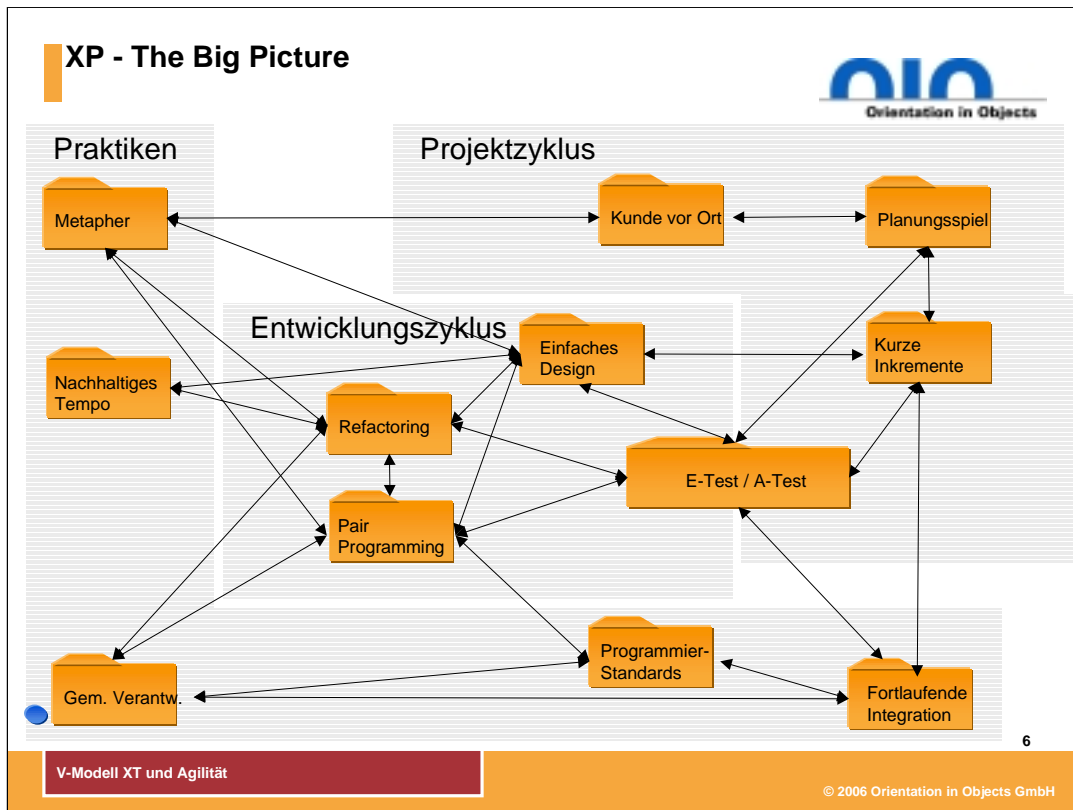
2. **Dokumentation kann schnell fehlinformieren** Wird gemacht wo notwendig

3. **Vollständige Spezifikation ist sehr teuer** und schwierig

Jede Zeile Space Shuttle Code kostet 1 Mannmonat

=> 100.000 BWL Code 1 Milliarde EURO 16 Jahre mit 500 Mann Stattdessen massive Integration des Kunden ins Projekt

4. Der Projekterfolg wird identifiziert mit Übereinstimmung mit Plan **Will man was man plant oder was man braucht**



12 Praktiken basierend auf 5 Grundprinzipien

Feedback, Einfachheit, inkrementelle Entwicklung, Änderungen willkommen, Qualitätsarbeit

Grundprinzipien sind zentral Praktiken sollen angepaßt werden => **kulturelle Werte**

Aufgeteilt in **3 Bereiche der Entwicklung**

Untereinander verzahnt, kaum ohneinander denkbar

Stories als Planspielbasis reichen

9 Minuten um

- Fix the process when it breaks.
- **We don't say if** because we already know you will need to make some changes.
- This doesn't mean the team can do whatever they want. The rules must be followed until the team has changed them.

<http://www.extremeprogramming.org/rules/fixit.html>

7

Wir mußten anpassen:

leider nicht immer Kunde vollständig vor Ort

Kunde möchte am Schluß konsistente Anforderungen

Deshalb => viel von XP, gemixt mit inkrementell

Gliederung



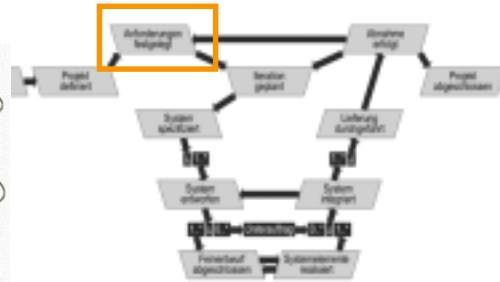
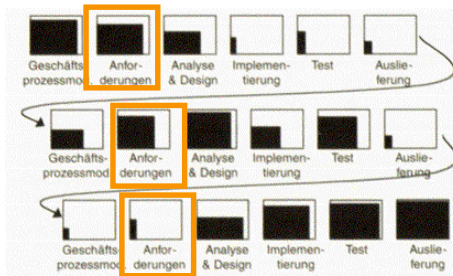
- Agilität - Manifest und XP
- **Iterative Inkrementelle Entwicklung**
- Wieviel Prozeß und Papier muß denn sein?
- Grundbegriffe V-Modell XT
- Die Agile Projektdurchführungsstrategie

8

Iterative und inkrementelle SW-Entwicklung

- Reife nimmt zu von Iteration zu Iteration

Verlagerung des Aufwands der Disziplinen
von Iteration zu Iteration



9

Grundgedanke:

Es wird immer in allen Disziplinen gearbeitet

Schwerpunkte verlagern sich.

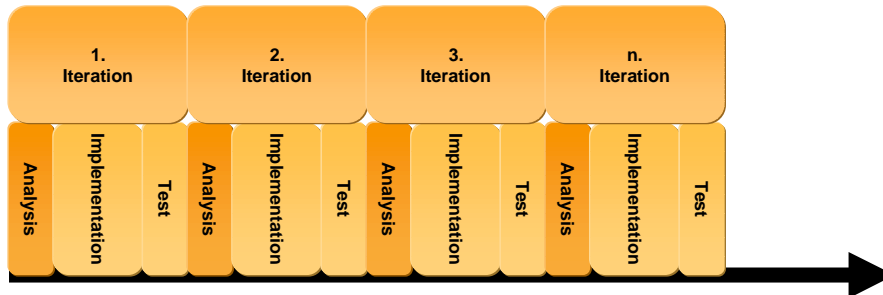
Anforderungen (und anderes) werden Fortgeschrieben

V-Modell sieht das auch so

Starre Iterative Softwareentwicklung

V-Modell XT

Wenn mehrere Iterationen bei der Systementwicklung vorgesehen sind, kann **nach der Abnahme einer Iteration** die detaillierte Planung der nächsten Iteration in Angriff genommen werden.



Where do I put all the analysts in the blue time

10

Sieht man **häufig linear - gerade in planerischer Hinsicht**

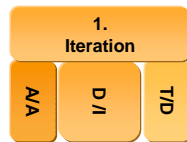
Problem: Entwicklung startet und stoppt ständig

Auch das V-Modell hat diese Sorgen

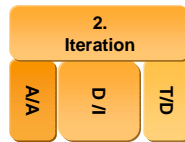
15 Minuten um

Parallele Iterationen - viel Freude mit Konfig-Management

Estimated length of Phases:

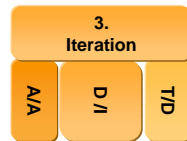


A/A: Analysis and Architecture	1 weeks
D/I: Design and Implementation	4 weeks
T/D: Test and Deployment	1 weeks
float	2 weeks



Release Frequency:

6 weeks / Release
2 weeks buffer time / Release



Konfigurationsmanagement
In dieser Arbeitsweise kann es nützlich sein,
die Anforderungen einem elektronischen
Konfigurationsmanagement zu unterwerfen
=> man braucht mehr als „User Stories“



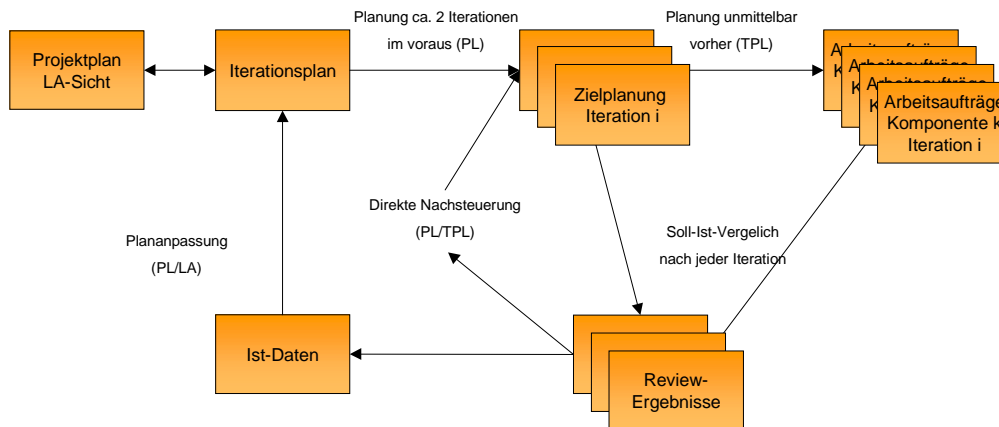
11

Eigentlich **Parallelität** notwendig

Erhöht **Aufwände** für Konfigmanagement

Außerdem zum **Zusammenhalten Makroprozeß** nötig

Makroprozeß benötigt



12

Makro steuert Mikro Arbeitsaufträge wie folgt:

Anforderungen Spezifikation

Rechtzeitiges Fertigstellen einer recht genauen Spezifikation
Innerhalb einer Iteration nur über teures Change Request
Änderungen möglich.

Technik

XP-artig Explorative Technik in freier Reihenfolge mit
spätestens nachträglicher Dokumentation

Gliederung

- Agilität - Manifest und XP
- Iterative Inkrementelle Entwicklung
- **Wieviel Prozeß und Papier muß denn sein?**
- Grundbegriffe V-Modell XT
- Die Agile Projektdurchführungsstrategie

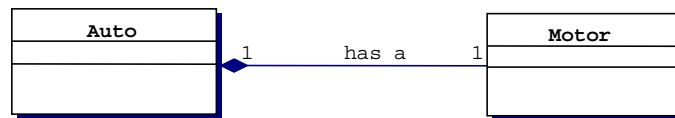
- Der **Formalisierungsgrad** von Requirements

- vorformal / textuell

Welchen würden Sie bevorzugen?

- **Jeder Motor ist nach Produktion Bestandteil eines Autos. Ein Auto hat immer einen Motor**

- semiformal / grafisch



- formal / mathematisch logisch

- $\forall m:\text{Motor} \exists !a:\text{Auto} (\text{teil_von}(m,a))$
- $\forall a:\text{Auto} \exists !m:\text{Motor} (\text{teil_von}(m,a) \wedge \forall m_i:\text{Motor} (\text{teil_von}(m_i,a) \Rightarrow m_i=m))$

14

Am Beispiel von Anforderungen:

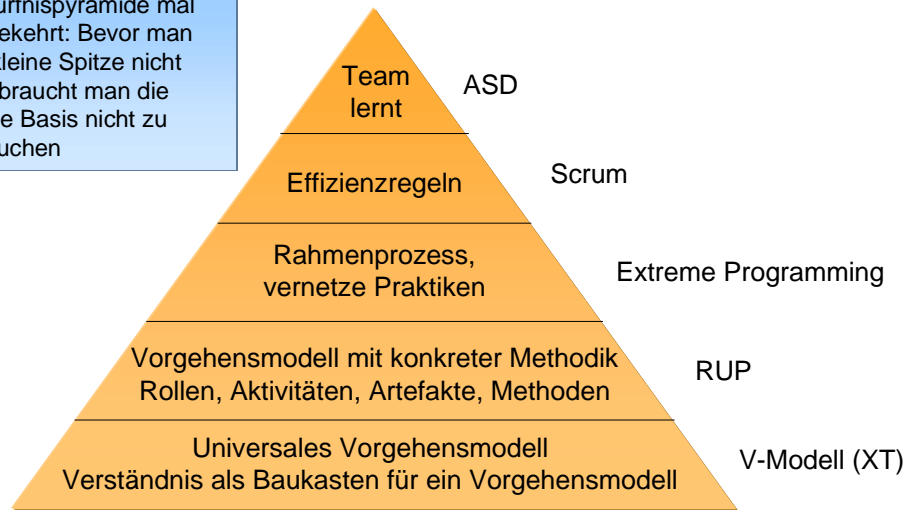
Was man braucht ist **Kontextabhängig**.

Raketenflugbahnen wollen bewiesen sein.

Intranetportale nicht

Prozeßmetzger: Darfs noch ein bißchen mehr sein?

Bedürfnispyramide mal umgekehrt: Bevor man die kleine Spitze nicht hat, braucht man die große Basis nicht zu versuchen



15

Adaptive Software Development - Jim Highsmith

Aus der Biologie - komplexe Adaptive Systeme - Anpassbarkeit

Emergent Behaviour (Fischschwärme, menschliches Bewußtsein)

Einfache Regeln - viele Beziehungen - komplexes Verhalten

Komplexe Regeln - wenig Beziehungen - dämliches Verhalten

Scrum - man hat immer Rückstand

Product- Sprint- Backlog. Hartes Timeboxing in den Sprints

Spring 30 Tage - alle 30 Tage wird mit AG neuer Lauf definiert

Jeden Tag Scrum Meeting 30 Minuten

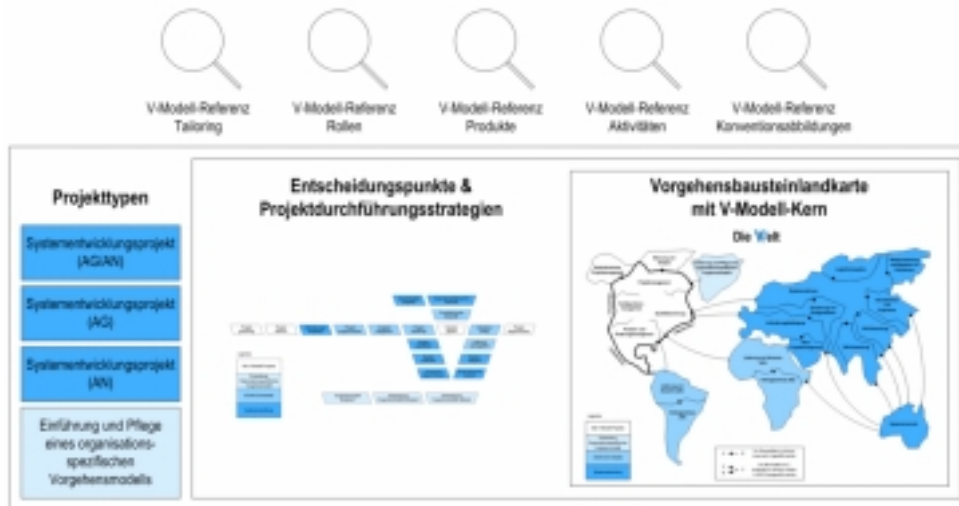
Was habe ich getan, was werde ich tun, was hat mich behindert?

Gliederung

- Agilität - Manifest und XP
- Iterative Inkrementelle Entwicklung
- Wieviel Prozeß und Papier muß denn sein?
- **Grundbegriffe V-Modell XT**
- Die Agile Projektdurchführungsstrategie

21 Minuten um

Gesamtstruktur V-Modell Big Picture



17

Ermittlung von Projekttypen und weiteren Kriterien
=> Input fürs Tailoring

Ermittelt **verbindliche und optionale Vorgehensbausteine**
Diese regeln keinen zeitlichen Ablauf

Außerdem ermittelt Tailoring **Empfehlung für Durchführungsstrategie**. Eine solche könnte das agile Vorgehen sein.

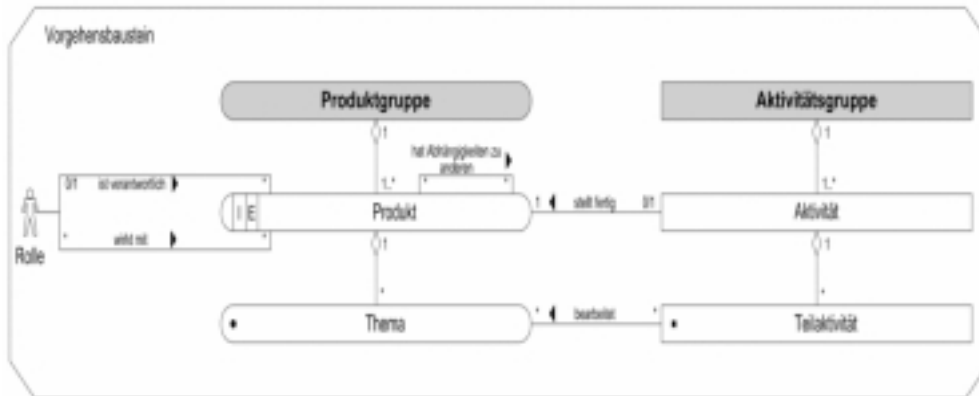


Lustig

Amerika immer verbindlich

Wir schauen gleich ein wenig nach Südostasien rein => Systemerstellung

Inhalt eines Vorgehensbausteins

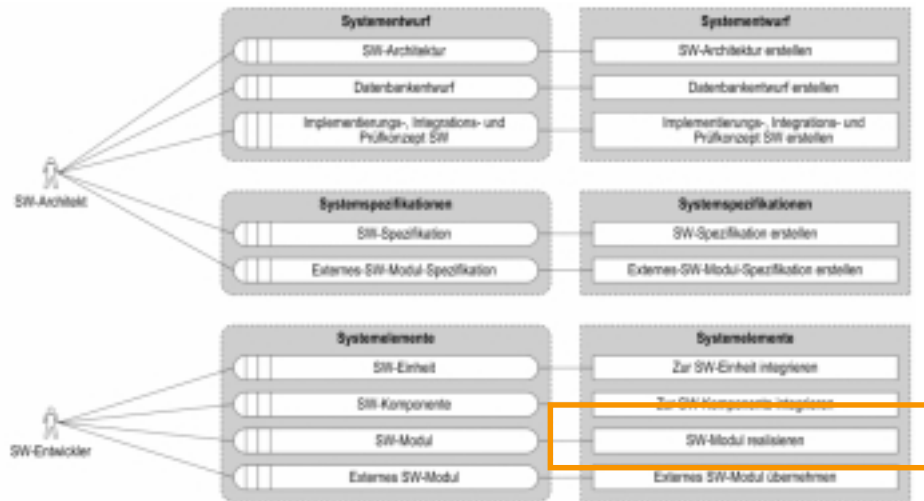


19

Typisches Prozeßvokabular.

Erweitert um Gruppierung nach oben und unten

Beispiel Baustein SW-Entwicklung



20

Südostasien

Der Entwickler erstellt ein Modul (Java Klasse)

27 Minuten um

- **Produkt**
 - SW-Modul
- **Sinn und Zweck**
 - Ein SW-Modul ist entsprechend den Anforderungen seiner **SW-Spezifikation** oder der Spezifikation eines übergeordneten SW-Elements zu implementieren. Das Vorgehen zur Implementierung hat sich an den Vorgaben im **Implementierungs-, Integrations- und Prüfkonzept SW** zu orientieren. Falls in der **Prüfstrategie** gefordert, ist das fertige SW-Modul einer Prüfung durch einen externen Prüfer zu unterziehen.
 - SW-Module sollten nach der Implementierung grundsätzlich einem Entwickler- und Integrationstest unterzogen werden. Als Grundlage kann die **Prüfspezifikation Systemelement** dienen.
 - Die Realisierung von SW-Modulen beinhaltet beispielsweise folgende Aspekte:
 - Programmierung unter Einhaltung der im **Projekthandbuch** festgelegten Standards und Richtlinien,
 - Erstellung von Compile-, Binde-, Lade-, Installations- und Generierprozeduren,
 - Korrekturen bis zur Fehlerfreiheit des Compilierens und Bindens,
 - Gegebenenfalls Programmierung von Test- und Simulationsläufen.

Vorlesen.

Deutlich machen, daß dieser Entwickler in jedem Fall schon **eine Menge Papier BRAUCHT** - und hierzu **nicht mehr gefragt wird** - im GGs. zu XP

30 Minuten um

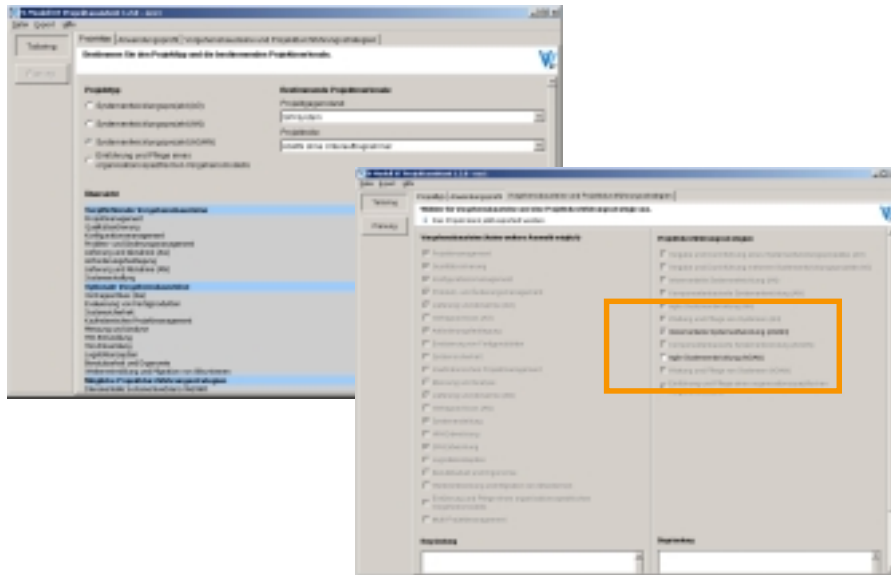
Gliederung



- Agilität - Manifest und XP
- Iterative Inkrementelle Entwicklung
- Wieviel Prozeß und Papier muß denn sein?
- Grundbegriffe V-Modell XT
- **Die Agile Projektdurchführungsstrategie**

22

Demo „Tailoring mit Projektassistent“



23

Zusätzliches Werkzeug - Open Source - mit Prozeß erhältlich - Tailoring

Kleine Demo, wie sich das anfühlt.

**AG/AN anklicken => entspricht vertrauensvollem Arbeiten
Entwicklung
Begründetes Nein bis Security**

=> 2 bleiben übrig.

Man kann fertiges getailortes VM-Generieren lassen.

Kurzdemo der beiden Sites.

Tailoring wählt Durchführungsstrategie

Fazit - Tailoring ist einfach:

Das typische Individualsoftwareprojekt mit enger Zusammenarbeit zwischen AG und AN wählt zwischen Inkrementeller und Agiler Strategie

Projekttyp	Systementwicklungsprojekt (AG)	Systementwicklungsprojekt (AN)	Systementwicklungsprojekt (AG/AN)	Einführung und Pflege eines organisationspezifischen Vorgehensmodells
weitere Projektmerkmale	Projektsziele	Systemlebenszyklusschritt Hohe Realisierungsrisiken Fertigprodukte	Systemlebenszyklusschritt Hohe Realisierungsrisiken Fertigprodukte	Keine weiteren Projektmerkmale zur Bestimmung der Projektdurchführungsstrategie notwendig
Projektdurchführungsstrategie	Vergabe und Durchführung eines Systementwicklungsprojektes (AG)	Inkrementelle Systementwicklung (AN)	Komponentenbasierte Systementwicklung	Einführung und Pflege eines organisationspezifischen Vorgehensmodells
	Vergabe und Durchführung mehrerer Systementwicklungsprojekte (AG)	Agile Systementwicklung	Wartung und Pflege Systemen (AN)	Systeme (AG/AN)



24

Nur kurzes Fazit:

Tailoring ist leicht

2 bleiben übrig

Wieviel Prozeß reduziert der Schneider?

	VM komplett	Inkrementell	Agil
• Rollen	31	19	19
• Produkte	107	58	56
• Aktivitäten	101	58	56
• Entscheidungspunkte	21	15	15

Fazit - Tailoring zeigt keine Unterschiede im "Prozeßgewicht"

Genauere Untersuchung notwendig

25

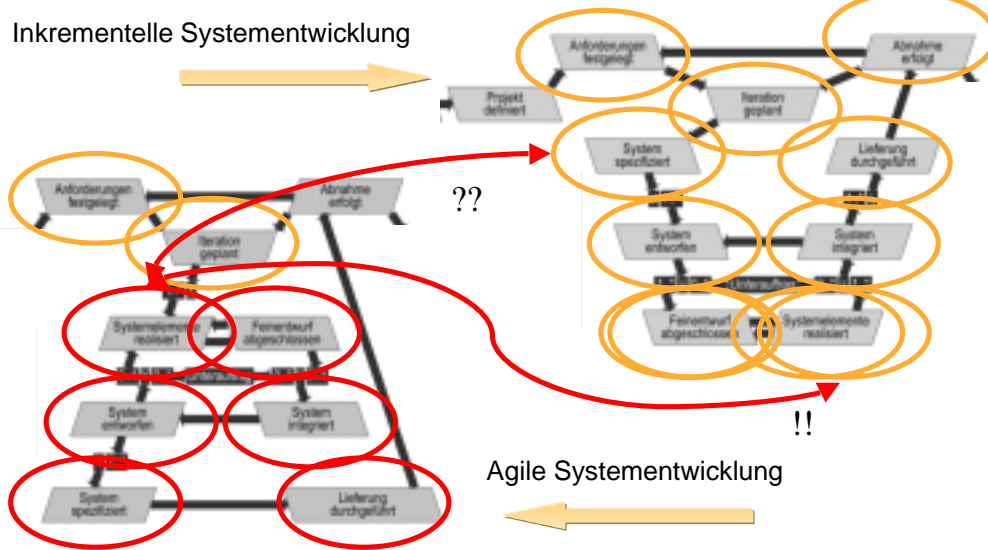
Leider bleibt viel übrig

und es gibt keinen Volumenunterschied zwischen den beiden Strategien.

36 Minuten um

Inkrementelle vs. Agile Systementwicklung

Inkrementelle Systementwicklung



Agile Systementwicklung

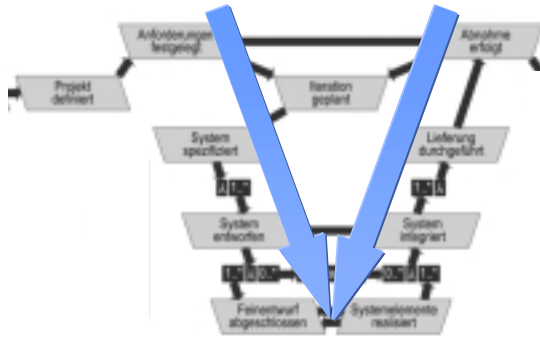
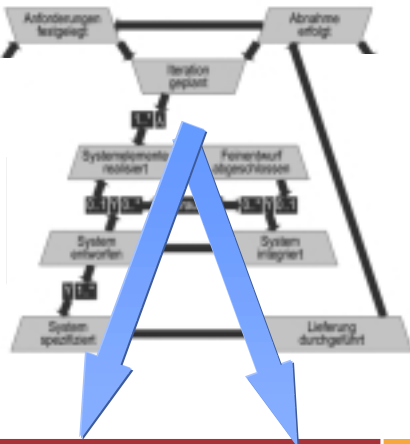
trägt unserer Arbeitsweise Rechnung

mit ausnahme der geplant erst nach Abnahme idee

Inkrementelle vs. Agile Systementwicklung

Inkrementelle Systementwicklung

i.e. „Top down“ Entwicklung



Agile Systementwicklung

i.e. „Bottom up“ Entwicklung

Agiles Manifest



Manifesto for Agile Software Development

(Beck, Fowler, Cockburn, uvm., 2001)

- **Einzelpersonen und Interaktionen** wichtiger als Prozesse und Werkzeuge
 - **Menschen müssen wichtiger sein als 58 Prozeßschritte :-)**
=> V-Modell per se auch Agil nicht „leichtgewichtig“
 - **Prozeß darf noch weiter angepaßt werden. Werkzeugauswahl frei**
- **Laufende Systeme** wichtiger als umfangreiche Dokumentation
 - **Laufende System müssen wichtiger sein als 58 Dokumente :-)**
 - **„Deutlicher Hinweis“ auf den Weg vom Ergebnis zur Forderung**
=> Grundgedanke gut getroffen. Im Detail etwas platt

28

Agile Entwicklung kennt 3 Hebel: Team, Software, Prozesse

Geht weniger auf Praktiken als kulturelle Werte

1.

Software kann ohne Prozess entstehen aber nie ohne gutes Team

Gute Teams organisieren sich von selbst

Einheitliche wiederholbare Prozesse paßt nicht zu Projekten

2.

Dokumentation ist statisch, kann schnell fehlinformieren

Wird gemacht wo notwendig

3.

Vollständige Spezifikation ist sehr teuer und schwierig

Jede Zeile Space Shuttle Code kostet 1 Mannmonat

=> 100.000 BWL Code 1 Milliarde EURO 16 Jahre mit

Agiles Manifest



Manifesto for Agile Software Development

(Beck, Fowler, Cockburn, uvm., 2001)

- **Zusammenarbeit mit dem Kunden** wichtiger als Vertragsverhandlungen
 - **Regelung möglich durch Projekttrolle AG/AN => 1 Projekt**
 - **andererseits erstmalig überhaupt Vertrag im Prozeß**
- **Fähigkeit auf Änderungen zu reagieren** wichtiger als Verfolgen eines Plans
 - **Großer Minimalumfang => SEHR gutes Änderungsmanagement**
 - **„vom kleinen zum Großen“ Gedanke könnte ausgeweitet werden.**



29

Agile Entwicklung kennt 3 Hebel: Team, Software, Prozesse

Geht weniger auf Praktiken als kulturelle Werte

1.

Software kann ohne Prozess entstehen aber nie ohne gutes Team

Gute Teams organisieren sich von selbst

Einheitliche wiederholbare Prozesse paßt nicht zu Projekten

2.

Dokumentation ist statisch, kann schnell fehlinformieren

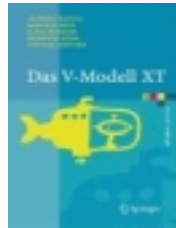
Wird gemacht wo notwendig

3.

Vollständige Spezifikation ist sehr teuer und schwierig

Jede Zeile Space Shuttle Code kostet 1 Mannmonat

=> 100.000 BWL Code 1 Milliarde EURO 16 Jahre mit



- **Das V-Modell XT**
 - **Sprache: Deutsch**
Gebunden - 500 Seiten - Springer
Erscheinungsdatum: 1. Juni 2006
ISBN: 3540302492



- **Das V-Modell XT**
 - **Sprache: Deutsch**
Broschiert - 200 Seiten - Dpunkt
Erscheinungsdatum: 1. Juni 2006
ISBN: 3898643352

Links



- Agiles Manifest
 - <http://agilemanifesto.org/>
- eXtreme Programming
 - <http://www.xprogramming.com>
 - <http://www.extremeprogramming.org/>
- ASD Adaptive Software Development
 - www.adaptivesd.com
- Scrum
 - <http://www.controlchaos.com>
- V-Modell XT
 - <http://www.v-modell-xt.de/>

31



Vielen Dank für Ihre Aufmerksamkeit !

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0



Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0