

# TUM

INSTITUT FÜR INFORMATIK

Proceedings zur  
Software & Systems Engineering Essentials 2010

Marc Sihling, Andreas Rausch, Jan Friedrich,  
Marco Kuhrmann



TUM-I1009

Mai 10

TECHNISCHE UNIVERSITÄT MÜNCHEN



TUM-INFO-05-I1009-0/1.-FI

Alle Rechte vorbehalten

Nachdruck auch auszugsweise verboten

©2010

Druck: Institut für Informatik der  
Technischen Universität München

Veranstalter

4SOFT

TU Clausthal

Bundesverwaltungsamt  
- Bundesstelle für  
Informationstechnik -

# SEE 2010

Köln 03./04. Mai

Sponsoren

Microsoft®

microTOOL  
making IT better

init[  
Digitale Kommunikation

Hi solutions

IBM®

Capgemini sd&m

berner & mattner

optimizing your development

ITIL

Agile

CMMI

Testing

Standards

Architektur

V-Modell XT

## Die Konferenz

für Vorgehensmodelle, Prozesse & Techniken  
des Software & Systems Engineering

Fachvorträge & Erfahrungsaustausch

Schwerpunkt: **Erfolgreich mit Standards**

[www.see-conf.de](http://www.see-conf.de)



# Software & Systems Engineering Essentials

## 1 Einführung

Der Nutzen einer einheitlichen Abwicklung von IT-Projekten ist zumindest theoretisch unstrittig: Standardisierte Abläufe und Methoden machen Projekte vergleichbar, erleichtern den Wechsel von Mitarbeitern zwischen Projekten, vereinfachen das Projektmanagement und führen letztendlich zu einer höheren Produktqualität. Häufig müssen Prozessverantwortliche jedoch leidvoll feststellen, dass sich die Umsetzung von Standards schwierig gestaltet und Vorgaben nicht eingehalten und nicht gelebt werden. Hinzu kommt, dass Auftraggeber oft weitere Standards und Richtlinien diktieren, welche die eigenen Vorgaben ergänzen oder ihnen möglicherweise sogar widersprechen. Bei IT-Projekten für die öffentliche Hand sind beispielsweise die Standards V-Modell XT und SAGA verpflichtend, ergänzt um eine Vielzahl weiterer Richtlinien und Vorgaben, etwa zur barrierefreien Gestaltung von Bedienoberflächen.

Die SEE 2010 greift mit Ihrem Schwerpunktthema die Gestaltung und Einführung von Standards sowie den Nachweis der Standardkonformität auf. Sie beantwortet damit Fragen wie:

- Welche Standards sollte ein Auftraggeber verlangen, ohne einen Auftragnehmer unnötig einzuschränken?
- Wie kann der Auftragnehmer diese Standards mit seinen eigenen Vorgaben in Einklang bringen?
- Wie kann er die Einhaltung nachweisen?
- Welches sind die Erfolgsfaktoren für Standardisierung in Unternehmen?
- Wie führt man Standards so ein, dass Projektteams sie akzeptieren?

## 2 Überblick

Die Konferenz fand am 03. und 04. Mai 2010 in Köln statt. Den Besuchern wurden zwei Keynotes, drei Tutorials und 42 Vorträge, zwei Tool Shootouts sowie eine Ausstellung von Werkzeugherstellern und Beratungsfirmen geboten. Als Keynoter sprachen Prof. Dr. Manfred Broy, Technische Universität München, sowie Dr. Stefan Weisgerber vom DIN (Deutsches Institut für Normung e.V.). Die Eröffnung der Konferenz erfolgte durch den Präsidenten des Bundesverwaltungsamtes, Christoph Venenkotte. Die 42 Vorträge sind in folgende Themengebiete gegliedert:

- Vertragsgestaltung
- Vorgehensmodelle und Methoden
- Projekt- und Anforderungsmanagement
- Systems Engineering und Embedded Engineering
- Management und Modellierung von Anforderungen

- Testen
- Entwicklung und Einführung von Vorgehensmodellen
- V-Modell XT und Projektmanagement
- Standardisierung mit XÖV
- Einführung und Nutzung von ITIL und CMMI
- Agilität und Softwareprozesse
- Architekturmanagement und der Architekturstandard SAGA

### **3 Zusammenfassung**

Fast ein Jahr ist seit der letzten SEE vergangen und auch dieses war geprägt von den Auswirkungen der Finanzkrise und den unterschiedlichen Maßnahmen, um die Wirtschaft wieder flott zu machen. Im Rahmen des IT Investitionsprogramms wurden in 2009 rund 250 Millionen Euro in die Wirtschaft investiert. Mit diesen Mitteln sollte unter anderem „schnell und zielgerichtet“ die Informations- und Kommunikationstechnik der Bundesverwaltung modernisiert werden. Doch wie gelingt es den Verantwortlichen, 350 Maßnahmen zügig und erfolgreich abzuwickeln?

Als ein Erfolgsfaktor wird die Standardisierung von Prozessen, Methoden, Daten und Werkzeugen gesehen. Auf diese Weise gelingt die Verbesserung von Effizienz und Qualität in IT-Projekten. Die SEE 2010 greift mit dem Motto „Erfolgreich mit Standards“ diesen Zusammenhang auf. Erstmals wird die Konferenz dieses Jahr auch von der Bundesstelle für Informationstechnik (BIT) des Bundesverwaltungsamts (BVA) veranstaltet.

Das Interesse an der SEE ist ungebrochen. Eine Rekordzahl von 79 Beiträgen ging durch das Review des Programmkomitees. Ein besonderer Dank gilt deshalb allen Autoren, die Ihre Vorträge für die SEE2010 eingereicht haben. Auch die konstante Anzahl von Ausstellern und der Zuwachs an Sponsoren bestätigen, dass die Bedeutung des Informations- und Erfahrungsaustauschs in diesem Bereich weiter zunimmt.

Wir bedanken uns bei allen Beteiligten und Teilnehmern für das Interesse an der SEE 2010 und für die Unterstützung.

# Inhaltsverzeichnis

<b>I. Tag 1</b>	<b>1</b>
<b>1. Vertragsgestaltung</b>	<b>3</b>
1.1. Festpreisprojekt – Behörde – V-Modell: Agilität im Grenzbereich . . . . .	5
1.2. Case Study Agiler Festpreis . . . . .	13
1.3. Vorgehensmodelle in der öffentlichen Beschaffung – zwei Welten finden zusammen . . . . .	23
<b>2. Vorgehensmodelle und Methoden</b>	<b>35</b>
2.1. Enriching RUP with key success factors for large-scale custom software development projects . . . . .	37
2.2. Software EKG . . . . .	57
2.3. Einsatz von Software-Engineering-Methoden im DLR . . . . .	71
<b>3. Projekt- und Anforderungsmanagement mit Scrum</b>	<b>73</b>
3.1. Agile-Scrum Planning . . . . .	75
3.2. Scrum mit individuellen Erweiterungen – ein Erfahrungsbericht . . . . .	85
3.3. Integration von Usability Engineering in den agilen Entwicklungsprozess nach Scrum . . . . .	101
<b>4. Systems Engineering und Embedded Engineering</b>	<b>137</b>
4.1. Domänenübergreifende Abhängigkeiten . . . . .	139
4.2. Erfahrungen von durchgängiger Entwicklung mittels SysML . . . . .	155
4.3. Integration von Software-Werkzeugen für die Entwicklung von Automatisierungssystemen mit dem Engineering Service Bus . . . . .	165
4.4. CESAR: Kosteneffizienz bei der Entwicklung sicherheitsrelevanter Echtzeitsysteme . . . . .	177
<b>5. Management und Modellierung von Anforderungen</b>	<b>187</b>
5.1. Agiles Anforderungsmanagement . . . . .	189
5.2. Integration von textuellen Anforderungen und Modellen . . . . .	205
5.3. Geschäftsregelmanagement und Fachmodellierung . . . . .	210
5.4. Nichts ist beständiger als der Wandel . . . . .	239
<b>6. Testen</b>	<b>257</b>
6.1. Abbildung von Standard-Testprozessen in modernen Testmanagementwerkzeugen . . . . .	259
6.2. Erfolgsfaktoren von Testprozessbewertungsmodellen . . . . .	277
6.3. Erweiterung des TPI-Modells zur Reifegradbewertung unter Berücksichtigung von projektspezifischen Anforderungen . . . . .	299
6.4. Beide Zweige des V aus einer Hand . . . . .	315
<b>II. Tag 2</b>	<b>319</b>
<b>7. Entwicklung und Einführung von Vorgehensmodellen</b>	<b>321</b>
7.1. Gibt es ein Leben nach dem V-Modell? . . . . .	323
7.2. Entwicklung und Einführung eines standardisierten Softwareentwicklungsprozesses . . . . .	347
7.3. Die Entwicklung eines Vorgehensmodells für E-Government-Projekte . . . . .	379
<b>8. Standardisierung mit XÖV</b>	<b>403</b>
8.1. Standardisierung auf allen Ebenen: XÖV und V-Modell XT in XPolizei-Vorhaben . . . . .	405
8.2. Ein Musterkonzept zur Wartung und Pflege von XÖV-Standards . . . . .	419
8.3. Interoperabilität durch Standardisierung – Zertifizierte XÖV-Konformität für die Verwaltung . . . . .	439

8.4. Vom Meldewesen zu XÖV – Erfolgsfaktoren der Standardisierung im eGovernment . . . . .	455
8.5. Semantische Repositories in Europa . . . . .	469
8.6. Ein Standard für Registeranwendungen: die Register Factory . . . . .	491
<b>9. Einführung und Nutzung von ITIL und CMMI</b>	<b>509</b>
9.1. ITIL bei der Optimierung von Serviceprozessen . . . . .	511
9.2. ITIL richtig gemacht . . . . .	527
9.3. Einführung von CMMI unter agilen Gesichtspunkten . . . . .	539
<b>10. V-Modell XT und Projektmanagement</b>	<b>553</b>
10.1. Projektmanagement im V-Modell XT . . . . .	555
10.2. Tailoring PRINCE2:2009 . . . . .	569
10.3. Steuerung von IT-Projekten durch ein Projekt Management Office . . . . .	585
<b>11. Agilität und Softwareprozesse</b>	<b>601</b>
11.1. Agil trotz Standards . . . . .	603
11.2. Lean MDD – Minimale Modellierung und maximale Generierung . . . . .	605
11.3. Zwischen Wasserfall und Agilität . . . . .	617
<b>12. Architekturmanagement und der Architekturstandard SAGA</b>	<b>629</b>
12.1. Automatisierte Architektur-Audits als Mittel zur Vermeidung von Architekturerosion . . . . .	631
12.2. SAGA-Konformität: Verbindlichkeit von Standards . . . . .	655
12.3. SAGA-konforme Webapplikationen – Herausforderung und Perspektiven . . . . .	669

**I.**  
**Tag 1**





# 1. Vertragsgestaltung

## Sessionüberblick

---

1.1. Festpreisprojekt – Behörde – V-Modell: Agilität im Grenzbereich . . . . .	5
1.2. Case Study Agiler Festpreis . . . . .	13
1.3. Vorgehensmodelle in der öffentlichen Beschaffung – zwei Welten finden zusammen .	23

---



## Festpreisprojekt - Behörde - V-Modell: Agilität im Grenzbereich

Markus Reinhold<sup>1</sup>, Stefan Rook<sup>2</sup>

<sup>1</sup>CoCOO, Hermann-Hesse-Str. 27  
D-85551 Kirchheim  
reinhold@cocoo.de

<sup>2</sup>it-agile GmbH, Paul-Stritter-Weg 5  
D-22297 Hamburg  
stefan.rook@it-agile.de

**Abstract:** Seit einigen Jahren erleben wir sog. „agile“ Software-Entwicklungsansätze wie z.B. *Scrum*, *eXtreme Programming* (XP) oder *Feature Driven Development* (FDD). Diese Entwicklungsansätze erheben den Anspruch, qualitativ hochwertige Software auf eine schnellere und ballastfreiere Art und Weise erstellen zu können, als traditionelle (sog. reichhaltige) Entwicklungsansätze dies vermögen.

Der Beitrag untersucht die Möglichkeiten der Integration des „reichhaltigen“ V-Modell® XT und agiler Ansätze und analysiert, ob sich unter Berücksichtigung dieser Erkenntnisse aus dem „reichhaltigen“ Baukastensystem V-Modell® XT ein agiler (leichtgewichtiger) Entwicklungsansatz formen lässt oder ob dies inhärent unmöglich ist?

### 1 Über die Vortragenden

**Markus Reinhold** studierte Informatik an der Fachhochschule Nürnberg, wie auch an der Universität Erlangen. Er befasst sich seit 1988 mit dem Thema Software-/System-Engineering und Toolunterstützung. Er war an der Weiterentwicklung des V-Modell'97 beteiligt, speziell im Bereich Integration der UML (Unified Modeling Language), wie auch an der Erstellung des V-Modell® XT (u.a. Abbildung V-Modell® XT vs. RUP).

Seit 1998 hat er in mehr als sieben Buchbeiträgen die Themen Entwicklungsprozesse, Werkzeuge, Objekttechnologie – UML verarbeitet (z.B. „OO-CASE Tools und Methoden“, „Erfolgreich mit Objektorientierung: Vorgehensmodelle und Managementpraktiken für die objektorientierte Softwareentwicklung“, „Agility kompakt“). Als Querdenker ist er als regelmäßiger Sprecher auf verschiedenen internationalen Konferenzen anzutreffen.

Seit 1999 unterstützt er als Inhaber und Principal Consultant von CoCOO Firmen u.a. bei der Einführung von angepassten Entwicklungsprozessen auf Basis V-Modell®, RUP® und/oder CMMI sowie der Einführung und Nutzung von natürlichsprachlichem

Anforderungsmanagement wie auch modellgestützten Spezifikationstechniken (UML, ER, SA, SD).

**Stefan Roock** studierte Informatik an der Universität Hamburg. Er arbeitet seit 1999 in agilen Projekten mit verschiedenen agilen Ansätzen wie Scrum, eXtreme Programming und Feature Driven Development.

Er hat neben zahlreichen Artikeln und Konferenzbeiträgen zum Thema auch Bücher über eXtreme Programming und Refactoring geschrieben.

Stefan Roock arbeitet bei der akquinet it-agile GmbH, die sich auf Durchführung und Coaching agiler Projekte spezialisiert hat. Er arbeitet sowohl als Trainer und Coach für Entwicklungsteams seiner Kunden als auch in agilen Projekten, die akquinet it-agile für Kunden durchführt. In diesem Zusammenhang hat er auch agile Festpreisprojekte durchgeführt, für die der Auftraggeber ein Vorgehen nach V-Modell XT vorgeschrieben hat.

## 2 Motivation

Heute stehen häufig Fragen zur Effizienzsteigerung beim Erstellen von Software ganz vorne auf der Agenda von betroffenen Unternehmen. Viele suchen nach ganzheitlichen und transparenten Entwicklungsansätzen die sich an die jeweiligen organisatorischen und projektspezifischen Bedürfnisse leicht anpassen lassen. Das Definieren und Aufbauen einer vollständig eigenständigen Entwicklungsphilosophie ist jedoch sehr kostenintensiv und erweckt bei den Betroffenen häufig den Eindruck, das Rad neu zu erfinden. Hier lohnt also ein Blick auf bestehende Ansätze.

In vielen Organisationen ist das V-Modell bereits etabliert weil es einen ganzheitlichen und transparenten Ansatz verfolgt oder es ist aus formalen Gründen (z.B. Vergabeverfahren) gesetzt. Die agilen Ansätze wirken auf der anderen Seite häufig attraktiv, weil sie eine schnellere und kostengünstigere Systementwicklung versprechen und viele Success-Stories vorweisen können.

Damit sehen sich viele Organisationen mit der Frage konfrontiert, ob es eine Möglichkeit gibt, beide Welten gewinnbringend zu integrieren. Auch wenn die Potenziale agiler Methoden hoch sind, lassen sich sofort schwerwiegende Probleme identifizieren, wenn ein radikaler Umstieg versucht wird. Ein Teil der Organisationen ist an das V-Modell gebunden und darf dieses gar nicht so ohne weiteres aufgeben und zu einem anderen Verfahren wechseln. Andere Organisationen dürften wechseln, gingen damit aber auch ein hohes Risiko ein. Schließlich sind die Denkwelten zwischen V-Modell und agilen Ansätzen doch sehr unterschiedlich. Und nicht zuletzt befriedigt das V-Modell heute Bedürfnisse in Organisationen, die agile Ansätze „out of the box“ so nicht abdecken können (z.B. Nachvollziehbarkeit, Wiederholbarkeit etc.).

Damit liegen ausreichend Gründe vor, sich mit der Frage zu beschäftigen, ob und wie V-Modell® und Agilität miteinander verbunden werden können und welche Konsequenzen daraus entstehen.

### 3 Ansatz

Der vorliegende Beitrag fokussiert bzgl. der Integration von V-Modell® und agilen Ansätzen auf zwei Fragen:

1. Nach welchem Vertragsmodell wird die Entwicklung durchgeführt?
2. Welche Produkte werden wann in welcher Form geliefert?

Grundsätzlich sollte jedoch berücksichtigt werden, dass auf abstrakterer (z.B. kultureller) Ebene häufig ein gewisser grundlegender Unterschied existiert, der jedoch eher durch die Organisation, die den jeweiligen Entwicklungsansatz konkretisiert und lebt, festgelegt wird.

#### 3.1 Vertragsmodell

Eine ganzheitliche (Vorab-)Planbarkeit von IT-Projekten setzt voraus, dass die zu entwickelnde Funktionalität / Qualität rechtzeitig festgelegt wird und auch die architekturelle Umsetzung klar ist. Das V-Modell versucht diesem Bedürfnis Rechnung zu tragen.

Die agilen Ansätze gehen im Gegensatz davon aus, dass so früh im Projekt noch nicht das Wissen vorliegt, um die Anforderungen optimal zu gestalten und zu beschreiben. Durch die frühzeitige Festlegung von Anforderungen besteht tendenziell das Risiko, dass wichtige Anforderungen auf Basis von Erkenntnisgewinn im Projektfortschritt vergessen und nicht beauftragt werden oder das inzwischen überholte oder letztlich nicht Mehrwert generierende Anforderungen umgesetzt werden. Folglich setzen die agilen Ansätze darauf, immer nur die nächsten wichtigen Anforderungen zu definieren (*Just-In-Time*). Das hat natürlich zur Konsequenz, dass sich Architektur und Entwurf des Systems inkrementell entwickelt. Das stellt besondere Herausforderungen an die Architektur und den Prozess zur Gestaltung der Architektur. Wir haben in Projekten die Erfahrung gemacht, dass den Entwicklern zu Projektbeginn das *Big Picture* für das System bekannt sein muss. So können Sie einen passenden Architekturstil wählen, der die passende Flexibilität für die Weiterentwicklung des Systems bietet. Die Kenntnis aller Anforderungen auf einer feingranularen Ebene ist hingegen nicht notwendig.

Für den Prozess der Architekturgestaltung gilt, dass in agilen Projekten dem automatisierten Testen und dem Refactoring (Umgestaltung existierender Codes) eine besondere Rolle zukommt. Code wird ständig in seine jeweils optimale Form (verständlich, redundanzfrei, entkoppelt) überführt und die Änderungen werden durch automatisierte Unit-tests abgesichert.

Auch wenn der agile Ansatz hier zu einer effizienteren Nutzung des entstehenden Erkenntnisgewinns im Projektfortschritt beiträgt, leistet er keinen aktiven Beitrag zur Befriedigung bestimmter Bedürfnisse der beauftragenden Organisation, z.B. nach früherer Sicherheit über das benötigte Budget oder einer Vergleichbarkeit von Angeboten in Ausschreibungsverfahren.

Trotzdem bedeuten Festpreisverträge für agile Ansätze keineswegs den Todesstoß. Schließlich gehen die agilen Ansätze auch mit dem Anspruch in den Kampf, beauftragte Funktionen *effizient* umsetzen zu können. Tatsächlich lassen sich sehr wohl Festpreisprojekte agil durchführen. Der Auftraggeber kann dann „lediglich“ einige Vorteile agiler Softwareentwicklung nur eingeschränkt nutzen: schnellere Time-to-Market, Vermeidung von Funktionen mit schlechtem ROI. Trotz dieser Einschränkungen können agile Verfahren ein hohes Maß an Flexibilität im Projekt erreichen. Im agilen Festpreisprojekt kann der Auftraggeber während des Projektes unmittelbar und ohne administrativen Overhead die Priorisierung von Anforderungen verändern und so Einfluss auf die Reihenfolge bei der Entwicklung nehmen. Er kann so frühzeitig testbare und ggf. sogar schon einsetzbare Systemversionen erhalten. Und er kann Anforderungen noch im Projekt tauschen, wenn er feststellt, dass wichtige Anforderungen übersehen wurden. Dann wird eine gleichzeitige – weniger wichtige – Anforderung aus der Vereinbarung gestrichen und durch die wichtigere Anforderung ersetzt. Da diese Situationen als natürlich und alltäglich angesehen werden, ist der hierfür notwendige administrative Aufwand auf ein Minimum beschränkt. Agiles Vorgehen führt damit zu *weniger administrativ aufwändig zu handhabenden* Change-Requests als klassisches Vorgehen, bei dem häufig der Eindruck entsteht, das durch eine hohe administrative Hürde bei den Change-Requests eine Art Abwehrmechanismus definiert wird.

Wenn der Auftraggeber die Priorisierung der Anforderungen kontinuierlich durchgeführt hat, bleiben zum Projektende hin nur die relativ unwichtigen Anforderungen übrig. Jetzt kann der Auftraggeber mit dem Anbieter darüber verhandeln, diese gar nicht mehr entwickeln zu lassen und stattdessen eine Art Stornogebühr zu bezahlen (z.B. 20% des Funktionswertes).

### 3.2 Produkte

Eine große Gefahr bei klassischen (nicht agilen) Entwicklungsprojekten besteht darin, dass der (frühe) Projektfortschritt an dem Fertigstellungsgrad von Dokumenten festgemacht wird. Diese Vorgehensweise ist jedoch keine inhärente Vorgabe des V-Modells sondern eine Frage der Projektdurchführungsstrategie.

Wieder verfolgen die agilen Methoden bewusst und explizit den Just-In-Time-Ansatz: Die Dokumente/Produkte werden schrittweise erstellt und zwar immer genau soweit, wie es im Moment sinnvoll ist. Neben diesem bewusst und explizit definierten zeitlichen Vorgehen, präferieren agile Ansätze außerdem nichtklassische Formen der Dokumentation.

So erstellen agile Projekte die Spezifikationen nicht klassisch als vom System getrennte Dokumente, sondern in ausführbarer Form (als Akzeptanztest, Unittest oder Verhaltensspezifikation). Diese ausführbaren Spezifikationen werden dabei jeweils vor der eigentlichen Implementation erstellt (*test driven development* (TDD) bzw. *behaviour driven development* (BDD)). Durch die Automatisierung ist sichergestellt, dass das System tatsächlich der dokumentierten Spezifikation entspricht, auch wenn seit der ursprünglichen Programmierung viele Änderungen am System durchgeführt wurden. Durch die vorherige Erstellung der Spezifikation ist weiterhin sichergestellt, dass das System ge-

nau das tut, was die Spezifikation verlangt und keine unspezifizierte/ungetestete Funktionalität existiert. Nicht jede Anforderung lässt sich in eine ausführbare Form überführen (z.B. Usability). Der Großteil der typischerweise formulierten Anforderungen lässt sich jedoch relativ einfach in eine ausführbare Spezifikation überführen.

Für die Integration agiler Ansätze in V-Modell-Projekte schlagen wir vor, eine geänderte Perspektive auf die Dokumente/Produkte einzunehmen und wo möglich ausführbare Spezifikationen zu verwenden. Durch passende Konventionen bei der Erstellung dieser Spezifikationen können klassische Spezifikationsdokumente bei Bedarf aus den ausführbaren Spezifikationen generiert werden. Neben der reinen Spezifikation haben die ausführbaren Spezifikationen weitere Vorteile als Nebenprodukte wie z.B. *Traceability* zwischen Anforderungen und Code. Da die Spezifikationen ausführbar sind, bedeutet das, dass alle beteiligten Artefakte (Anforderungsbeschreibung, zugehörige Testfälle und schließlich die Realisierung in Programmcode) formal miteinander verbunden sein müssen. Mit aktuellen Entwicklungsumgebungen lassen sich diese Verbindungen leicht nachverfolgen. Damit kann jederzeit leicht festgestellt werden, welche Anforderung zu welchem Code geführt hat.

## 4 Bewertung

### 4.1 Abgrenzung zur inkrementellen Projektdurchführungsstrategie (PDS)

Die vorgeschlagene Kombination aus V-Modell und agilen Ansätzen, weist strukturelle Ähnlichkeiten mit der inkrementellen Projektdurchführungsstrategie (PDS) des V-Modells auf. Tatsächlich basiert das vorgeschlagene Verfahren auf der inkrementellen PDS wie sie in der V-Modell® Version 1.3 definiert ist.

Allerdings geht der vorgeschlagene Ansatz über die inkrementelle PDS hinaus. Während die inkrementelle PDS die Spezifikationen jeweils je Inkrement komplett vor der Programmierung erstellt, wird im vorgeschlagenen Ansatz jeweils der Ausschnitt der Spezifikation ergänzt, der für die gerade anliegende Programmieraufgabe notwendig ist (Just in Time-Ansatz).. Die folgende Tabelle stellt die unterschiedliche Vorgehensweisen bei inkrementeller PDS und agiler Vorgehensweise gegenüber (die beschriebene Vorgehensweise bei der inkrementellen PDS wird so nicht vom V-Modell XT gefordert, faktisch in den meisten V-Modell-Projekten aber so praktiziert).

Schritt	Inkrementelle PDS	Agiles Vorgehen
1	Plane Iteration	Plane Iteration
2	Spezifiziere Inkrement	Für die wichtigste Anforderung des Inkrements:  • Spezifiziere Anforderung (Akzeptanztests)
3	Entwerfe Inkrement	
4	Erstelle Feinspezifikation für Inkrement	



5	Realisiere Inkrement	<ul style="list-style-type: none"> <li>• Entwerfe Anforderung (Quick Design Session am Flipchart)</li> <li>• Erstelle Feinspezifikation für Umsetzung der Anforderung (Unit-tests)</li> <li>• Realisiere Anforderung</li> <li>• Integriere Anforderung</li> </ul> Danach das ganze für die zweitwichtigste Anforderung usw.
6	Integriere Inkrement	
7	Liefere Inkrement	Liefere Inkrement

Außerdem ist der vorgeschlagene Ansatz spezifischer als die inkrementelle PDS. Das V-Modell® XT lässt bewusst offen, wie die Spezifikationen konkret aussehen. Dieser Freiheitsgrad legt vielen Anwendern ein klassisches Verständnis als Dokument nahe. Der im vorliegenden Beitrag vorgeschlagene Ansatz basiert jedoch explizit auf der Idee, wenig(er) klassische Dokumente zu schreiben, sondern stattdessen auf automatisch ausführbare Spezifikationen zu setzen. Außerdem schlagen wir vor, die Produkte jeweils zeitlich eng verzahnt parallel zu erstellen und nicht sequenziell nacheinander.

#### 4.2 Fazit

Unter bestimmten Rahmenbedingungen bietet eine konsequente Integration des ganzheitlichen und transparenten Managementrahmens des V-Modell XT mit den erfolgreichen und bewährten Ansätzen der agilen Entwicklung unter Berücksichtigung und Nutzung der interdisziplinären Zusammenhänge neue Chancen zur erfolgreicherer und effektiveren Abwicklung von Softwareprojekten mit hoher Qualität. Ziel jedes erfolgreichen Projektes ist es, mit vorgegebenen Ressourcen zügig den maximalen Mehrwert für den Kunden zu realisieren, ohne dabei die Transparenz des Vorgehens oder die Qualität der entstehenden Software zu beeinträchtigen. Die Umsetzung dieses Ziels in der Praxis ist jedoch nicht trivial.

Die genannten Ansätze und Techniken zur Integration agiler Ansätze mit dem V-Modell haben wir in mehrjährigen Projekten erfolgreich durchgeführt. Dabei hat sich gezeigt, dass die vorgestellten Ansätze funktionieren und tatsächlich Vorteile generieren. Leichtgewichtige Entwicklung ist auch mit dem V-Modell möglich und dafür können nach wie vor Festpreisprojekte eingesetzt werden. Sie bieten dem Auftraggeber aber deutlich mehr Flexibilität und erhöht damit für ihn die Effektivität der Projekte.

Neben der rein technischen Betrachtung spielt jedoch die Ausprägung des Wertesystems eines Entwicklungsansatzes eine entscheidende Rolle. Durch die Generizität des V-Modells wird dieser Aspekt, im Gegensatz zu den agilen Ansätzen, nicht explizit ausgeprägt. Das Wertesystem eines konkreten V-Modell-Projektes wird also durch das Unternehmen bestimmt und nicht das Vorgehensmodell. Daher kann es in V-Modell-Projekten auch zu einem expliziten Widerspruch zum agilen Wertesystem [AM] kom-

men. Dies liegt jedoch immer in der Verantwortung der jeweiligen Organisation bzw. der verantwortlichen Menschen in dieser Organisation.

### **Literaturverzeichnis**

- [MR04] Reinhold, Markus: Der agile RUP® in „Agility kompakt“, Spektrum Akademischer Verlag, Heidelberg – Berlin, 2004 (Hruschka, Peter et al, Hrsg.)
- [WRL05] Wolf, Henning; Roock, Stefan; Lippert, Martin: eXtreme Programming – Eine Einführung mit Empfehlungen und Erfahrungen aus der Praxis, dpunkt Verlag, 2005
- [AM] <http://agilemanifesto.org>

Copyright © 2009 by CoCOO und akquinet it-agile GmbH. Publikation urheberrechtlich geschützt. Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung oder Teilen daraus, vorbehalten. Kein Teil der Publikation darf in irgendeiner Form, egal welches Verfahren, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Dies gilt auch für Zwecke der Unterrichtsgestaltung. Eine schriftliche Genehmigung ist einzuholen. Die Rechte Dritter bleiben unberührt.

## 1.1 Festpreisprojekt – Behörde – V-Modell: Agilität im Grenzbereich

## Case Study Agiler Festpreis

Michael Malenke

Bredex GmbH  
Mauernstr. 33  
38108 Braunschweig  
michael.malenke@bredex.de

**Abstract.** An einem abgeschlossenen Projekt wird der Agile Festpreis vorgestellt und eine Bewertungen sowohl aus Kunden- wie aus Dienstleister-Sicht vorgenommen. Aus diesen werden Entscheidungshilfen für die Eignung des Agilen Festpreises für eigene Projekte sowie konkrete Verfahrenshinweise abgeleitet.

### 1 Definition

Der Agile Festpreis wird formal definiert und in seiner konkreten Ausprägung in dem Beispielprojekt dargestellt. Dabei wird auf vertragliche Aspekte eingegangen und der Widerspruch aus Agilität und festem Umfang diskutiert.

### 2 Projektstruktur

Im agilen Projektmanagement ist die Time-Box (Sprint) das zentrale Strukturierungselement. Ergänzt durch das Timebox-Meeting und das Timebox-Release definiert es den Arbeitszyklus des Projektes. Der Umgang damit im aktuellen Projekt wird vorgestellt und an einer Beispiel-Timebox veranschaulicht.

### 3 Aufgabenteilung

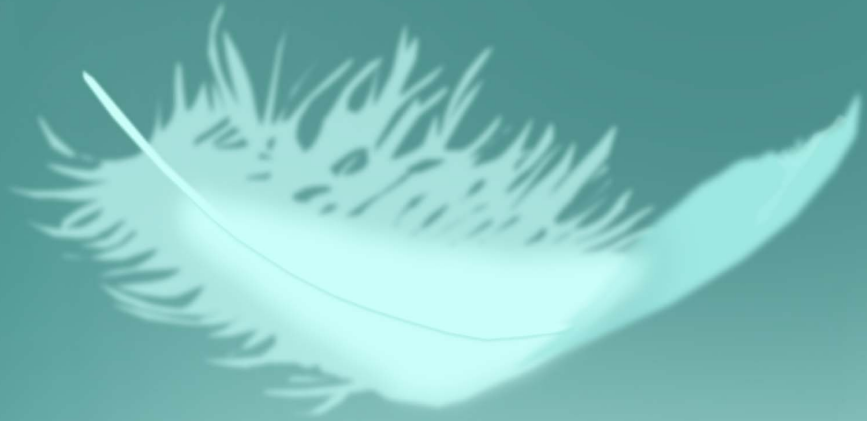
Die Notwendigkeit der aktiven Mitarbeit des Kunden zum Gelingen des Projekts ist ein wichtiger Bestandteil des agilen Vorgehens und führt dazu, dass Kunde und Auftragnehmer sich auf Augenhöhe gegenüberstehen. Die besonderen Anforderungen an den Dienstleister aber auch – und vielleicht unerwartet – insbesondere an den Kunden werden erarbeitet und mit klassischen Verfahrensweisen verglichen. Die Effektivität dieser Aufgabenteilung und die Konsequenz diese über die Projektlaufzeit hinweg auf hohem Niveau zu halten, entscheidet über den Projekterfolg. Ist der Kunde auf diese Art von Mitarbeit nicht vorbereitet, wird er seine Rolle nicht ausfüllen und letztlich mit Projektergebnis, das er mit zu verantworten hat, nicht zufrieden sein.

#### 4 Erfahrungen und Bewertung

Im Verlauf des Projektes und der abschließenden gemeinsamen Analyse zwischen Kunden und Auftraggeber haben beide Seiten nicht nur unterschiedliche Wahrnehmungen gezeigt und Bewertungen gezogen, sondern gemeinsam Punkte extrahiert, die in Form eines Lessons-Learned vorgestellt werden. Spezielle Themen sind:

- Maßnahmen zur effektiven Zusammenarbeit
- Anwendungstest der Timebox-Releases (Kunde)
  - o Der Nutzen einer gemeinsamen Ticket-Datenbank und der Umgang damit
- Umgang mit Anforderungsänderungen
  - o Geänderte Aufwände gegenüber dem Festpreis
  - o Definition von Abnahmekriterien

Die „neue Freiheit“ des Kunden im agilen Vorgehen von Timebox zu Timebox über das weitere Vorgehen zu entscheiden und aktiv mitzuarbeiten, hat sich dabei nicht nur als „lustvoll“ sondern auch „lastvoll“ herausgestellt. Es werden die Bewertungen aus Sicht des Dienstleisters und des Kunden vorgestellt.



# Case Study Agiler Festpreis

*Lustvolles und Lastvolles*

© 2008 Bredex GmbH Version 2.3

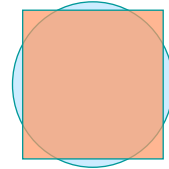


## Übersicht

- ▶ **Der rechtliche Rahmen**
- ▶ **Das Projekt**
- ▶ **Arbeitsweise**
- ▶ **Aufgabenteilung**
- ▶ **Erfahrungen und Bewertung**
- ▶ **Checkliste**



## Der rechtliche Rahmen



### ► Agiler Festpreis

Auf Basis eines definierten Umfangs wird ein Festpreis angeboten.

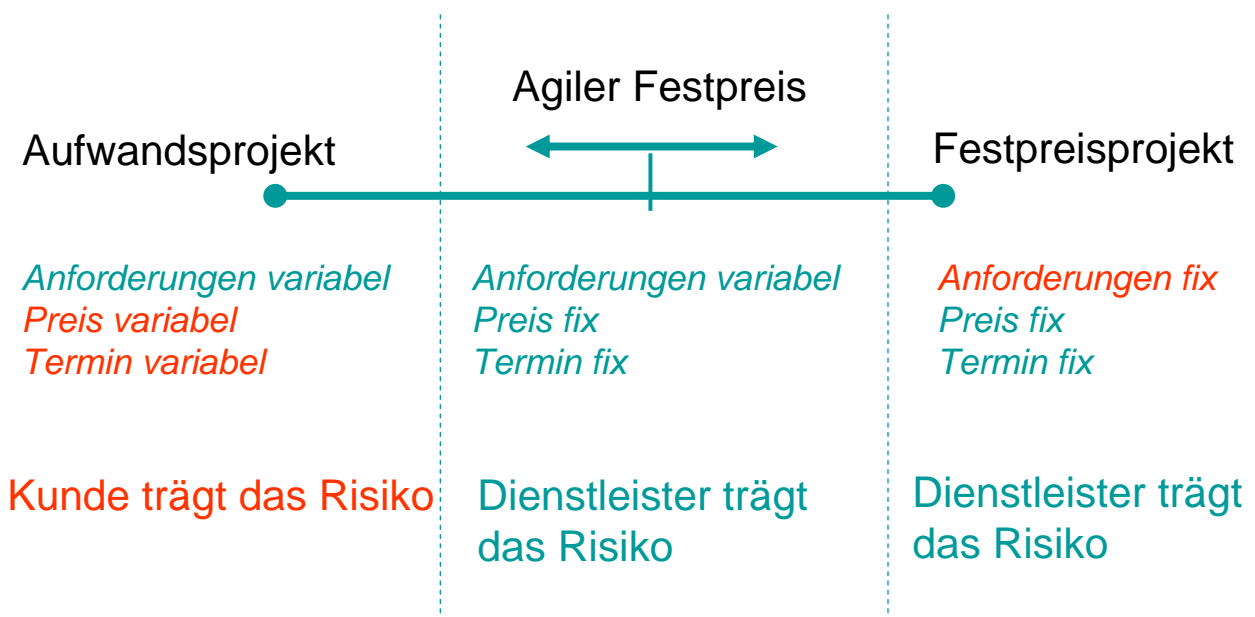
Die Umsetzung erfolgt auf Basis von Timeboxen fester Länge (hier: 6 Wochen)

Zu Beginn jeder Timebox kann der Kunde Änderungen am definierten Umfang vornehmen, die sich in Summe im Aufwand ausgleichen müssen.

Vertrag nach Abschluss der Timebox kündbar.



## Ein Kunden-Traum wird wahr?





## Das Projekt

- ▶ **Erstellung einer Anwendung zur Ausführung und Verwaltung von Berechnungsläufen**

- ▶ **Technologien**

Eclipse RCP basierter Rich-Client (Java 1.6) für Linux und Windows.

Hibernate (ORM)

Oracle Datenbank 10g

C-Bibliothek für Fortran (implementiert JSON-Dateischnittstelle)

CaseStudy Agiler Festpreis

09.04.2010

5



- ▶ **Build/Test**

Automatisierte funktionale Tests (Guidancer)

Automatisierte Builds (Hudson)

- ▶ **Projektlaufzeit: Februar-Dezember 2009**

CaseStudy Agiler Festpreis

09.04.2010

6





## Arbeitsweise

- ▶ **Aufteilung der Projektlaufzeit in Timeboxen von 6 Wochen**
- ▶ **Timebox-Meeting zu Beginn jeder Timebox**
  - Vorstellung der Ergebnisse der letzten Timebox
  - Das Timebox-Release der Software wird dem Kunden übergeben zum Test.
  - Vorstellung der umzusetzenden Aufgaben der aktuellen Timebox.



## Timebox

- ▶ **Kunde:**
  - Testen des Timebox-Release
  - Fehlererfassung per Trac
  - Erarbeiten der Anforderungen der nächsten Timebox
- ▶ **Entwicklungsteam**
  - Umsetzung der Anforderungen der Timebox
  - Unterstützung des Kunden bei der Definition der Anforderungen der nächsten Timebox
- ▶ **wöchentlicher Jour fixe**



## Agilität

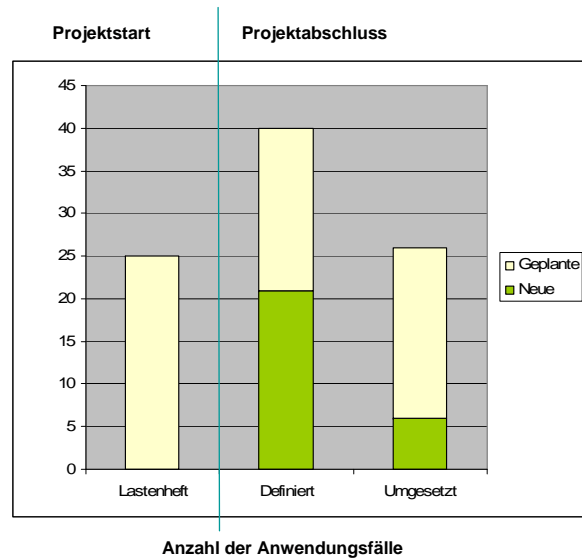
### ► Lastenheft

25 Anwendungsfälle

### ► Nach Abschluss der Projektes

Definiert: 40 Anwendungsfälle

Umgesetzt: 26  
Anwendungsfälle



CaseStudy Agiler Festpreis

09.04.2010

9



## Prozess der Anforderungsanpassung

### ► Der Aufwand für die Änderung ist zu ermitteln

Implementierungsaufwand

Dokumentationsaufwand

Testaufwand

### ► Mehraufwände durch Wegfall/Reduzierung von Anforderungen kompensieren

Kunde muss Anforderungen priorisieren

Prozess der Anforderungsanpassung erzeugt selber Aufwände



CaseStudy Agiler Festpreis

09.04.2010

10



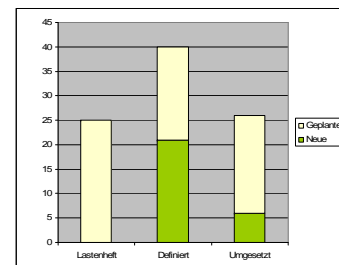
## Bewertung aus Kundensicht

- ▶ **Agiler Festpreis: richtige Entscheidung!**
- ▶ **Die intensive Arbeit an den Anpassungen der Anforderungen hat das Ergebnis besser gemacht.**

- ▶ **Aber:**

Der Aufwand beim Kunden war größer als erwartet.

Die erste Euphorie über die Möglichkeiten des agilen Vorgehens wurde durch eine realistischere Einschätzung der Grenzen ersetzt.



## Bewertung aus Dienstleistersicht

- ▶ **Intensive Zusammenarbeit mit dem Kunden**
- ▶ **Frühzeitiges Erkennen „falscher“ Anforderungen sowie neuer Anforderungen**
- ▶ **Regelmäßige Priorisierung**
- ▶ **Kunde „arbeitet“ mit Timebox-Releases**
- ▶ **Gefahr: Euphorie beim Kunden über die Möglichkeiten des agilen Vorgehens.**



## Bewertung aus Dienstleistersicht

- ▶ **Belastbare Aufwandsabschätzungen für geänderte Anforderungen nötig**
- ▶ **Aber:**  
hoher Aufwand für Aufwandsschätzung und deren Dokumentation
- ▶ **Kunden im Erlernen agiler Prozesse unterstützen**

CaseStudy Agiler Festpreis

09.04.2010

13



## Alter Wein in neuen Schläuchen?

- ▶ **Ja:**  
denn auch bei einem Festpreisprojekt können Anforderungsänderungen berücksichtigt werden.
- ▶ **Nein:**  
der Änderungsprozess beim agilen Festpreis ist nicht die Ausnahme sondern die Regel.  
Der Kunde wird zum Partner in der Entwicklung: Sein Einsatz entscheidet mit über den Erfolg des Projektes.

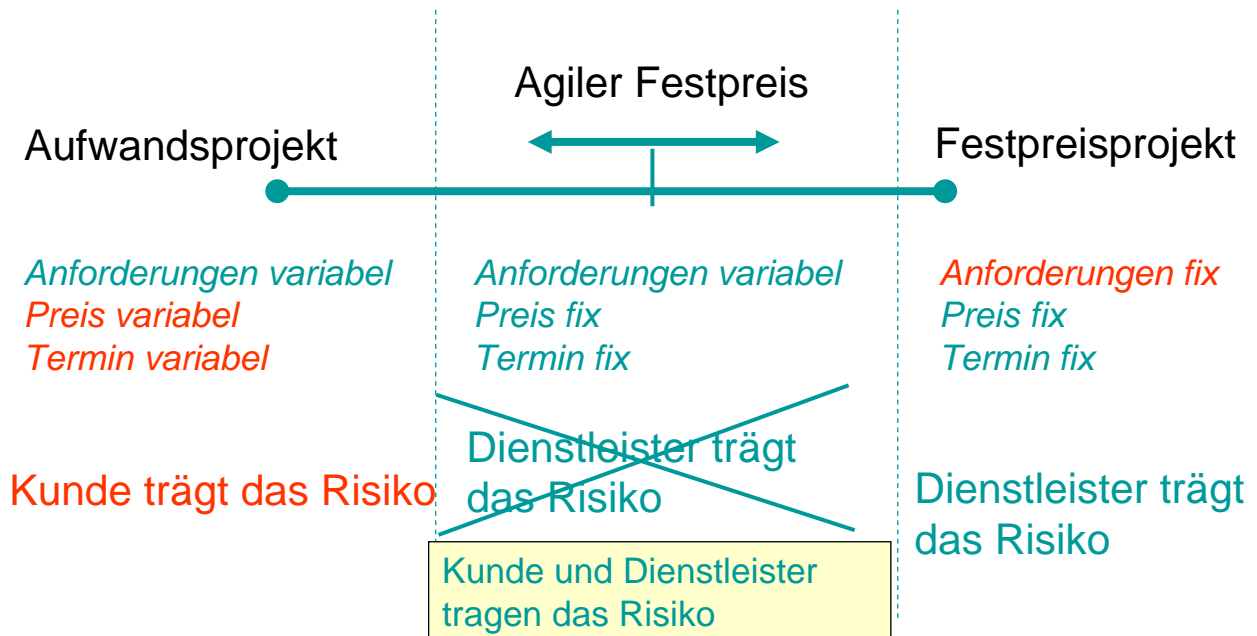
CaseStudy Agiler Festpreis

09.04.2010

14



## Ein Kunden-Traum wird wahr?



CaseStudy Agiler Festpreis

09.04.2010

15



## Checkliste

- ▶ **Aufklärung des Kunden im Vorfeld über seine Aufgaben**
  - Verfügbarkeit der KeyUser gewährleisten
  - Entschlussfähigkeit herstellen: Entscheider benennen, Prozess definieren.
  - Abnahmekriterien benennen (Test der Timebox-Releases als Teilabnahmen)
  - Priorisierung der Anwendungsfälle
  - Kosten von Anforderungsänderungen diskutieren.
- ▶ **Anforderungsänderungen dokumentieren**
- ▶ **Zugriff auf gemeinsames Bugtracking-Tool einrichten**
- ▶ **Qualität sichern: automat. Build, automat. funktionale Tests**
- ▶ **Gemeinsame Arbeitskultur schaffen**

CaseStudy Agiler Festpreis

09.04.2010

16

## Vorgehensmodelle in der öffentlichen Beschaffung – zwei Welten finden zusammen

Dirk Schreiber<sup>1</sup>, Marc Sihling<sup>2</sup>

<sup>1</sup>Beschaffungsamt des BMI, St. Augustiner Straße 86  
53255 Bonn  
Dirk.Schreiber@bescha.bund.de

<sup>2</sup>4Soft GmbH, Mittererstraße 3  
80336 München  
sihling@4soft.de

**Abstract:** Moderne Vorgehensmodelle, wie etwa das V-Modell XT, gehen über die Beschreibung der reinen Entwicklungsleistung hinaus und definieren präzise das Zusammenspiel zwischen Auftraggeber und Auftragnehmer. Hier treffen sie – zumindest im Umfeld der öffentlichen Vergabe – auf etablierte Standards wie etwa die Verdingungsordnung für Leistungen (VOL) und EVB-IT. Der Vortrag beleuchtet die Projektdurchführung einmal von Seiten der Entwicklungsprozesse und einmal aus Sicht der Ausschreibung und zeigt, ob und wie beide Seiten zusammenfinden.

### 1 Motivation

Auftraggeber der öffentlichen Hand verpflichten sich und ihre Auftragnehmer regelmäßig zur Anwendung des Standards V-Modell XT. Dieses Vorgehensmodell regelt nicht nur die Abläufe der Systementwicklung, sondern steuert auch das Zusammenspiel zwischen den beteiligten Parteien. Damit eint es zwei Welten, die zusammen gehören, aber nicht immer zusammen passen:

In der Welt der Vorgehensmodelle für die Systementwicklung werden „best practices“ kommuniziert, um den Erfolg von IT Projekten zu steigern. Das V-Modell XT propagiert etwa die iterative, inkrementelle Herangehensweise für die Minderung von Projektrisiken. Aber auch Elemente einer agilen Projektdurchführung sind dem Standard bekannt und helfen, Risiken einer unpräzisen Leistungsbeschreibung durch die frühzeitige Erarbeitung von Prototypen zu begegnen. Doch, wie können mit einem Prototypen die Anforderungen geschärft werden, wenn das Projekt doch ausgeschrieben wurde?

In der Welt der Projektanbahnung, der Ausschreibung und des Vertragsschlusses steht einem präzisen Lastenheft eine passende Lieferung entgegen. Auch hier sind Iterationen möglich, werden aber mangels Zeit oder Budget selten konsequent genutzt. Lieferungen werden per Abnahme bestätigt, womit nicht nur die Rechnungsstellung, sondern auch rechtliche Konsequenzen einher gehen. Ist es nun möglich, dem Auftragnehmer Zwi-

schenergebnisse (etwa das Pflichtenheft) abzunehmen? Wann beginnen Gewährleistungsfristen zu laufen?

Aus Sicht der Ausschreibungsstandards sind Vorgehensmodelle Qualitätsmodelle, die dem Auftragnehmer „aufgebürdet“ werden und die daher nur für diesen interessant sind. Aus Sicht der Vorgehensmodelle sind VOL und EVB-IT notwendige Übel, um Rechtssicherheit zu erlangen und den Vorgaben der öffentlichen Hand zu entsprechen.

## 2 Der Vortrag

Im Vortrag argumentieren die Autoren, dass ein besseres Ineinandergreifen der beiden Welten zu Klarheit bei den Beteiligten und mittelfristig zur Verbesserung des Projekterfolgs führt. Dies kann und sollte unterstützt werden durch eine Reihe konkreter Anwendungshilfen, wie etwa Checklisten und Produktvorlagen.

Anhand von Beispielen aus eigener Erfahrung und durchgeführten Vergaben im Beschaffungsumfeld wird beispielsweise aufgezeigt, wie Ausschreibungen in Situationen gestaltet wurden, in denen das Ergebnis der Entwicklung zum Zeitpunkt der Ausschreibung nicht abschließend beschrieben werden konnte.

Als Resumé ist festzuhalten, dass

1. der Auftraggeber die Ziele des Projektes bei der Planung dezidiert und möglichst detailliert festlegen muss,
2. der Auftraggeber hinsichtlich der Rollenverteilungen, die er in der Projektdurchführung festlegt und hinsichtlich der Fragen, wie die Betreuung und Weiterentwicklung des entwickelten Systems ausgestaltet werden soll, vor Veröffentlichung der Ausschreibung alle wichtigen Festlegungen getroffen hat,
3. die Abbildungslücke zwischen den Hilfsmitteln der VOL /A , den EVB-IT-Vertragsmustern, der UfAB und den existierenden Vorgehensmodellen sehr groß ist und keine Hilfsmittel/Anleitungen/Muster zur Verfügung stehen, um Ausschreibungen zu gestalten, in denen ein iterativ-inkrementelles VM angewendet werden soll,
4. bei den öffentlichen Auftraggebern (soweit dem Referenten bekannt) bisher nur wenige praktische Erfahrungen mit Softwareprojekten auf Basis eines iterativ-inkrementellen Vorgehensmodells bestehen



# Vorgehensmodelle in der öffentlichen Beschaffung - zwei Welten finden zusammen

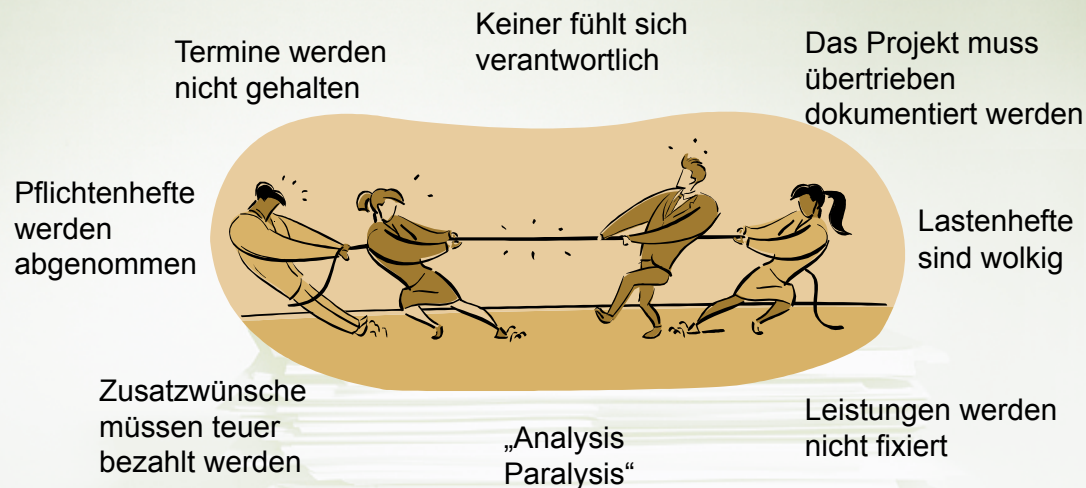
Dirk Schreiber  
Beschaffungsamt des BMI  
Marc Sihling  
4Soft GmbH  
SEE 2010 - 3. bis 5. Mai 2010

## Agenda

- Motivation
- Der Standard V-Modell XT
- Der Standard UfAB
- Praxisbeispiele
- Zusammenführung und Zusammenfassung



## Durchführung von IT-Projekten



### Situationsanalyse von Schwartz 1968 zur Softwaretechnik

- Keiner hat ein Gesamtverständnis der Problemstellung
- Die Entwickler kommunizieren schlecht miteinander
- Innerhalb einer Gruppe von Entwicklern gibt es erhebliche Unterschiede in ihren Fähigkeiten
- Gute Entwickler sind kreativ.  
Sie sind aber auch in Situationen kreativ, die gar keine Kreativität erfordern.
- Nicht-Entwickler verstehen Entwickler nicht. Bekommen diese mehr Verantwortung, hält niemand sie davon ab, immer mehr Entwickler anzustellen.
- Es ist schwierig von Entwicklern den Projektfortschritt zu erfahren. Zudem sind die meisten Manager sehr leichtgläubig. Die Methoden zur Vorhersage des Projektfortschritts sind unglaublich schlecht.
- Die Tester eines Systems haben keine Methoden, die gewährleisten, dass das System umfassend getestet wird.
- Kein Mensch kennt das Gesamtsystem oder seinen augenblicklichen Zustand.

Er bemerkt außerdem: „Ich muss zugeben, ich habe mich dauernd gefragt, ob diese Systeme so viele Leute (Entwickler) benötigen, weil sie so groß sind, oder ob diese Systeme, so groß sind, weil so viele Leute an Ihnen arbeiten...“.

## Das V-Modell XT

- Das V-Modell XT ist ein ergebnisorientiertes Vorgehensmodell für Entwicklungsprojekte
- Standard
  - V-Modell seit 1992
  - V-Modell XT seit 2005
  - V-Modell XT Bund seit 2010
- Ziele
  - Steigerung des Projekterfolgs
  - Hohe Anwendbarkeit und Erlernbarkeit
  - Ergebnisorientierte Arbeitsweise
  - Nutzung des „gesunden Menschenverstands“



## Besonderheiten des V-Modell XT

- Angebotserstellung und Ausschreibung erfolgen innerhalb des Projekts
- Das Lastenheft ist der zentrale Hebel für den Projekterfolg; Anforderungen werden bewertet
- Minderung der Projektrisiken durch eine iterative, inkrementelle Vorgehensweise
- Keine Abnahme von Zwischenergebnissen (etwa Projektdokumentation, Pflichtenheft, etc.)



## Das V-Modell XT in der Praxis

- Anwendung des Standards erfordert umfassendes (auch technisches) Know-how beim Auftraggeber - dies ist nicht immer gegeben
- Die Idee, einzelne Iterationen auszuschreiben und zu beauftragen, kommt selten zum Einsatz
- Lieferungen ohne Abnahme würden dem Projekt zu (noch) mehr Flexibilität verhelfen
- Eine methodische Unterstützung im Bereich Ausschreibung ist selten gegeben
- Bei der Wahl des „besten“ Angebots verlässt sich das V-Modell XT auf externe Standards (bildet diese aber nur unzulänglich ab - siehe etwa Verhandlungsverfahren)



## Vergabeverfahren UfAB V

- UfAB
  - Unterlage für Ausschreibung und Bewertung von IT-Leistungen
- Ziele
  - Etablierung einheitlicher, praktikabler Bewertungsmethoden
  - Grundlage für die Erarbeitung eindeutiger und vollständiger Verdingungsunterlagen
  - Auswertung von Angeboten transparent und nachvollziehbar
- Gesetzliche Grundlage
  - Gesetz gegen Wettbewerbsbeschränkungen (GWB)
  - Vergabeordnung (VgV)



### Beispiele aus der Praxis: Projektrahmenvertrag

- Typische Inhalte:** Migrations- und Modernisierungsaufgaben
- Budget:** Zwischen ca. 500 T € und 10 Mio €  
Kontingent an Leistungen (in PT oder Volumen) ist unverbindlich
- Leistungen:** Dienstleistungen und Werkleistungen
- Anforderungsziele:** Sind in der Verdingungsunterlage beschrieben
- Vergabeart:** Nichtoffenes Verfahren mit vorgeschaltetem TN
- Zielerreichung:** durch Einzelabrufe
- **Start:** 1. Teilziel ist in den Verdingungsunterlagen spezifiziert
  - **Durchführung:**
    - bei Dienstleistungen: Abrechnung nach Aufwand
    - bei Werkleistungen: Festlegen von Teilzielen/Teilanforderungen  
Festpreis, Transparenz durch bepreiste Rollen  
Erfahrungen aus vergleichbaren Teilzielen

### Beispiele aus der Praxis: Projektrahmenvertrag

#### **Variante:**

- Erste Teilanforderung: Entwicklung Protoyp
- Entscheidung beim AG über weiteres Vorgehen
- Mögliche Alternativen:
  - ⊕ Wettbewerb (Ausnahme)
    - ⊕ Projektrahmen ausschöpfen (s.o.)
    - ⊕ Projekt stoppen (z.B. Buy statt Make)

## Zusammenspiel Auftraggeber - externe QS/Bauaufsicht - Lieferant



### Beispiele aus der Praxis: Entwicklung in Stufen

**Anzahl Stufen:** 2 - 3

**Typische Inhalte:** große Projekte mit neuen Anforderungen

**Budget:** Zwischen ca. 5 Mio € und 10 Mio €

**Anforderungseinheiten:** Sind in der Verdingungsunterlage grob beschrieben

**Vergabeart:** Nichtoffene Verfahren mit vorgeschaltetem TN

**Vorgehen 1. Stufe:** Umsetzung / Beauftragung in Einzelaufträgen

**Aufteilung in 2 Fachlose:**

1. Los: Vorgaben fachlich/inhaltlich und technische Architektur  
Unterstützung bei QS der Realisierung
2. Los: Realisierung

**Mögliche Probleme:**

- 2 Vergabeverfahren (Aufwand und zeitliche Taktung)
- Wissenstransfer problematisch

## Beispiele aus der Praxis: Entwicklung in Stufen – 2. Teil

### Vorgehen 2. Stufe

**Vergabeart:** Nichtoffenes Verfahren mit vorgeschaltetem TN

#### **Umsetzung:**

- Beauftragung in Einzelaufträgen
- Aufteilung in 2 Mengenlose und Rahmenvereinbarung mit Mehreren:
- Losinhalte:
  - ⊗ Vorgaben inhaltlich und technische Architektur
  - ⊗ Unterstützung bei QS der Realisierung
  - ⊗ Realisierung

- **Vorteil 2. Stufe:** Geringere Abhängigkeit von einem Lieferanten
- **Mögliches Problem:** Gewährleistungsansprüche zuordnen und durchsetzen

## Beispiele aus der Praxis: Entwicklung in Stufen

**Anzahl Stufen:** 2

**Inhalt:** Migration, Sanierung und Zukunftsfähigkeit

**Gesamtbudget:** Zwischen ca. 5 Mio € und 10 Mio €

**Anforderungseinheiten:** Sind in der Verdingungsunterlage grob beschrieben

**Vergabeart Stufe 1:** Nichtoffenes Verfahren mit vorgeschaltetem TN

**Preismodell Stufe 1:** Anforderungseinheitenpreis

#### **Aufgaben 1. Stufe:**

- ⊗ Betriebsfähigkeit sicher stellen (Festpreis)
- ⊗ Erkennen weiterer Schwachstellen
- ⊗ Entwicklung Pilot für künftige Anwendungen (Styleguide und Architekturvorgaben)
- ⊗ Implementierung XP-Arbeitsweise im Entwicklungsteam des Auftraggebers

## Beispiele aus der Praxis: Entwicklung in Stufen - 2. Stufe

**Vergabeart Stufe 2:** Verhandlungsverfahren mit vorgeschaltetem TN

**Preismodell Stufe 2:** Anforderungseinheitenpreis für Sanierung  
Agile Phasenfestpreise für Neuentwicklung

### **Aufgaben 2. Stufe:**

- ④ Sanierung aufgrund der gewonnenen Erkenntnisse
- ④ Neue Anwendungen auf Basis der Ergebnisse aus Stufe 1 (Styleguide, Architekturvorgaben, XP-Arbeitsweise)

## Resumée:

- Die VOL/A bietet eine Reihe von Möglichkeiten auch für agile Vorgehensweisen
- Der (öffentliche) Auftraggeber muss in der Ausschreibung:
  - Die Ziele des Projektes dezidiert und detailliert festlegen
  - Die Rollenverteilungen für die Projektdurchführung festlegen
  - Festlegungen treffen bzgl. der Wartung und Weiterentwicklung des Systems
- Es gibt wenig Hilfe / Empfehlungen zur Ausgestaltung (z.B. in der bestehenden UfAB) insbesondere bei iterativ-inkrementellen VM
- Bei öffentlichen Auftraggebern bestehen bisher nur wenige Erfahrungen mit Softwareprojekten auf Basis iterativ-inkrementellen VM
- Die Einbindung einer externen, unabhängigen „Bauaufsicht“ ist bei komplexen Projekten in jedem Fall zu empfehlen



## Konventionsabbildung UfAB im V-Modell XT Bund



## Zusammenfassung

- Mit dem V-Modell XT Bund finden Vorgehensmodell und Vergabeverfahren endlich (auf dem Papier) zusammen
- In der Praxis führt ein „gutes“ Lastenheft zu einer effizienten, erfolgreichen Ausschreibung
- Auch mit Beschaffungsreferaten / Beschaffungsamt ist Erfahrung im Projektteam notwendig. Anwendungshilfen, wie etwa Checklisten, versprechen Unterstützung im Projekt
- Iterative, inkrementelle Abläufe sind in den Standards abgebildet und werden zu Unrecht nur selten angewendet
- Eine gründliche Vorbereitung ist in jedem Fall Erfolgsfaktor für ein Projekt



# Fragen?



Weiterführende Informationen auf <http://www.cio.bund.de/>



# 2. Vorgehensmodelle und Methoden

## Sessionüberblick

---

2.1. Enriching RUP with key success factors for large-scale custom software development projects . . . . .	37
2.2. Software EKG . . . . .	57
2.3. Einsatz von Software-Engineering-Methoden im DLR . . . . .	71

---



## **Enriching RUP with key success factors for large-scale custom software development projects**

Marianne Heinemann, Bettina Duwe, Gregor Engels

Capgemini sd&m Research  
Carl-Wery-Str. 42  
81739 München  
Marianne.Heinemann@Capgemini-sdm.com  
Bettina.Duwe@Capgemini-sdm.com  
engels@uni-paderborn.de

### **1 Motivation**

Capgemini sd&m has a long-standing experience in executing custom software development (CSD) projects with a special focus on large-scale and rightshore projects, where resources are distributed over different locations in Germany, Poland (nearshoring) or India (farshoring).

For many years, that experience was brought together in a proprietary life cycle model for CSD projects. However, this proprietary life cycle model was not based on an existing industrial standard and therefore was difficult to communicate to clients as well as to employees with other cultural background.

Besides the proprietary life cycle model, Capgemini sd&m has developed its software engineering methodology named QUASAR that focuses on the special requirements for Enterprise Architecture and Custom Software Development projects. With having enhanced QUASAR at fast pace in the last years, it became important to not only update the life cycle model, but to additionally integrate the software engineering methodology.

With the continuously increasing trend to software industrialization and thus the growing demand for standardization, the goal was to build on a standard life cycle model and integrate the key factors of our successful software engineering methodology QUASAR.

### **2 Approach**

To unify the understanding of our software engineering processes and work products, we defined an ontology of relevant software engineering notions as the basis for all further actions.

The RUP disciplines have been validated, and we found a need to extend and detail them in order to reflect the special requirements of custom software development.

Next step was to validate the work products. A fine-grained artifact model has been developed that elaborates the artifact hierarchy based on a categorization for different artifact types. The result is a comprehensive catalogue of artifacts for all software engineering disciplines. We identified the critical artifacts and defined relevant in- and output artifacts for each discipline.

In parallel to the development of the artifact model, we commenced the life cycle model evolution. We identified the key success factors of our traditional approach and substantiated RUP to explicitly reflect them.

We elaborated our life cycle model based on the three dimensions results, time and tasks, the latter being reflected by the RUP concepts of phases, disciplines and activities. The results dimension was one elementary pillar within our traditional model: We work with a concept of stages that coarsely define, to which degree the system has been built. The explicit definition of *development stages* allows parallelized work and thus a quicker project execution.

The approach to build a development stage is iterative, where cycling through the development activities occurs by component and includes testing and software integration. The resulting step-by-step integration of software from early on is seen as a further success factor within our traditional life cycle model and as such has been transferred to the new model.

The bridging element between the life cycle model and the software engineering methodology is a new kind of milestone that is not foreseen by RUP: the discipline milestone. These milestones define the to-be status of critical artifacts at special points in time for each discipline and based on our development stages.

The resulting life cycle model is called “Quasar project incremental” and is meant to be our standard life cycle model for large-scale and rightshore projects.

### 3 Evaluation

“Quasar project incremental” combines the standard RUP approach with our traditional key success factors for custom software development. Furthermore, it integrates project management with our software engineering methodology hence providing a comprehensive model of interaction of those two levels within one project that practically helps to communicate.

In the meantime, “Quasar project incremental” has been deployed in several custom software development projects. Concrete samples and a critical evaluation will be given in the presentation.



## Enriching RUP with key success factors for large-scale custom software development projects

Marianne Heinemann, Bettina Duwe, Gregor Engels  
SEE 2010, Cologne, May 2010




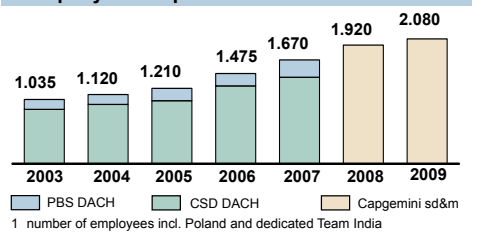
TOGETHER. FREE YOUR ENERGIES.

---

### AGENDA

- **Motivation**
- RUP Extension
- Summary and Evaluation

## Capgemini sd&m stands for powerful, comprehensive process and software solutions that bring competitive advantages to our customers

<p><b>Customers</b></p> <p>Our customers are both well-known companies in all industries and organizations whose success depends on sophisticated process and software solutions</p> <p>They benefit from increased competitiveness thanks to</p> <ul style="list-style-type: none"> <li>• Differentiation of business critical solutions</li> <li>• Improved efficiency of existing solutions</li> </ul>	<p><b>Offices</b></p> 
<p><b>Service Offerings</b></p> <ul style="list-style-type: none"> <li>• Process oriented technology consulting</li> <li>• Custom solution development</li> <li>• Implementation and roll-out of standard software</li> <li>• System integration</li> <li>• Application Management</li> <li>• Business Information Management</li> </ul>	<p><b>Company development<sup>1</sup></b></p> 
<p><b>Competences</b></p> <ul style="list-style-type: none"> <li>• Implementation oriented consulting</li> <li>• Management of complex IT projects</li> <li>• Software engineering             <ul style="list-style-type: none"> <li>– Design of sophisticated IT architectures</li> <li>– Implementation of Package-based solutions (i.e. SAP)</li> <li>– Rightshore®</li> <li>– Research &amp; innovation</li> </ul> </li> <li>• Close working relationships with customers</li> </ul> <p><small>Rightshore ® is a trademark belonging to Capgemini</small></p>	



## Industrialized software development requires standardization

**Challenges of industrialized software development require**

**precision in communication and processes.**

- ❖ World-wide distributed development
- ❖ Repeatability, predictability and efficiency of development processes
- ❖ Predictable high quality of all development artifacts

Life cycle model should be:

- ❖ Standards-driven
- ❖ Precise
- ❖ Work product focused
- ❖ High process efficiency
- ❖ Tool-supported
- ❖ Ready-to-use



## We transferred high levels of precision, product focus and efficiency from our own mature and well accepted assets into the RUP standard

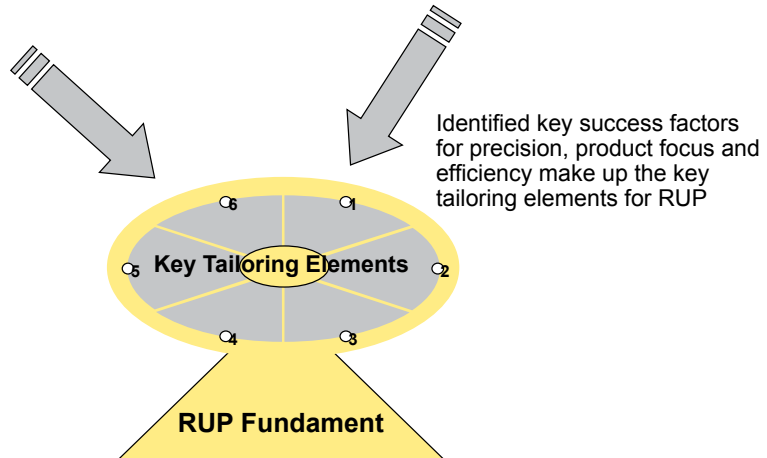
### Solution approach

**Capgemini sd&m's software engineering methodology QUASAR**

- Enhanced from a design-centric framework towards a fully-fledged software development method
- Offering concrete methodical guidelines and tools for the entire software development process
- Including analytical techniques for ensuring and controlling quality
- Easily accessible and immediately usable
- Integrates our rightshore® experience

**Capgemini sd&m's proprietary life cycle model "Stage model"**

- Optimized for large-scale projects
- Process efficiency



Rightshore® is a trademark belonging to Capgemini



© 2010 Capgemini sd&m - All rights reserved 5

## AGENDA

- Motivation
- **RUP Extension**
- Summary and Evaluation

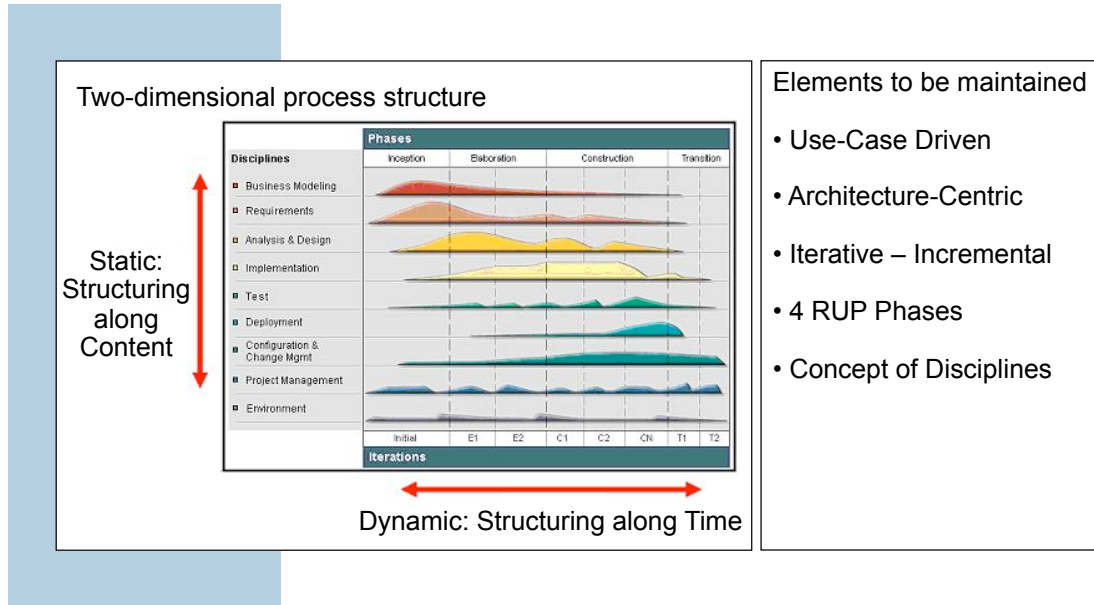


© 2010 Capgemini sd&m - All rights reserved 6



## The Rational Unified Process is a process framework that can be adapted and extended with our identified key success factors

### The Unified Process as Basis to build upon – Elements to be maintained



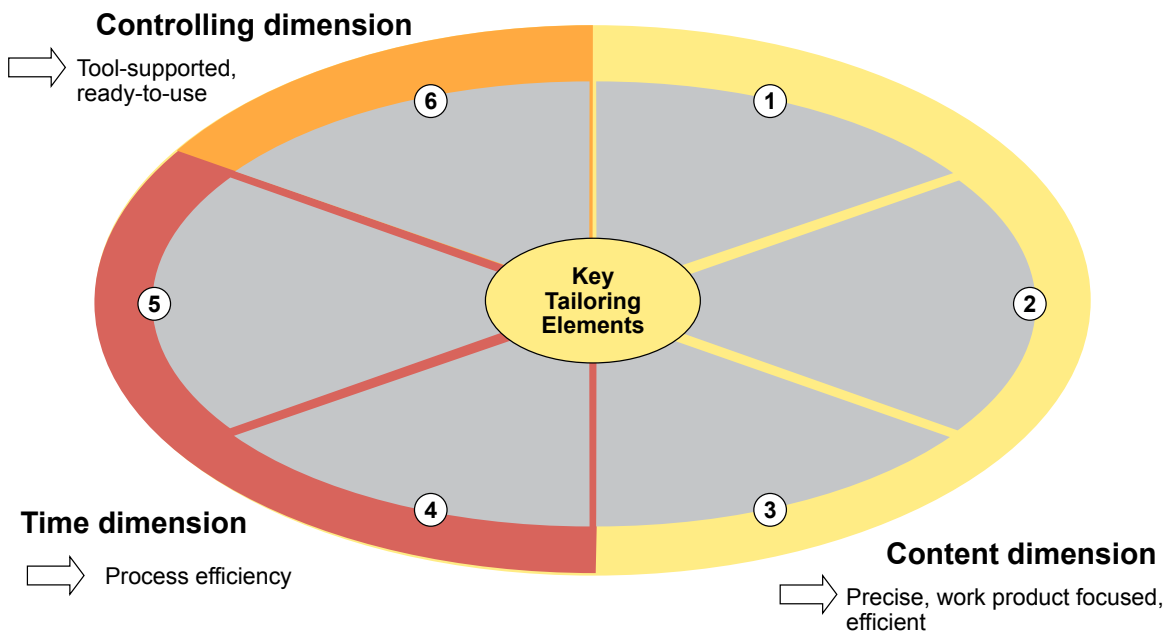
Reference: P. Kruchten, The Rational Unified Process – an Introduction, Addison-Wesley, 2004



© 2010 Capgemini sd&m - All rights reserved 7

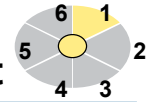
## With the RUP extensions, we tailor the content and time dimensions of the process and bring them together in the controlling dimension

### Elements from QUASAR and Stage Model for Integration into RUP

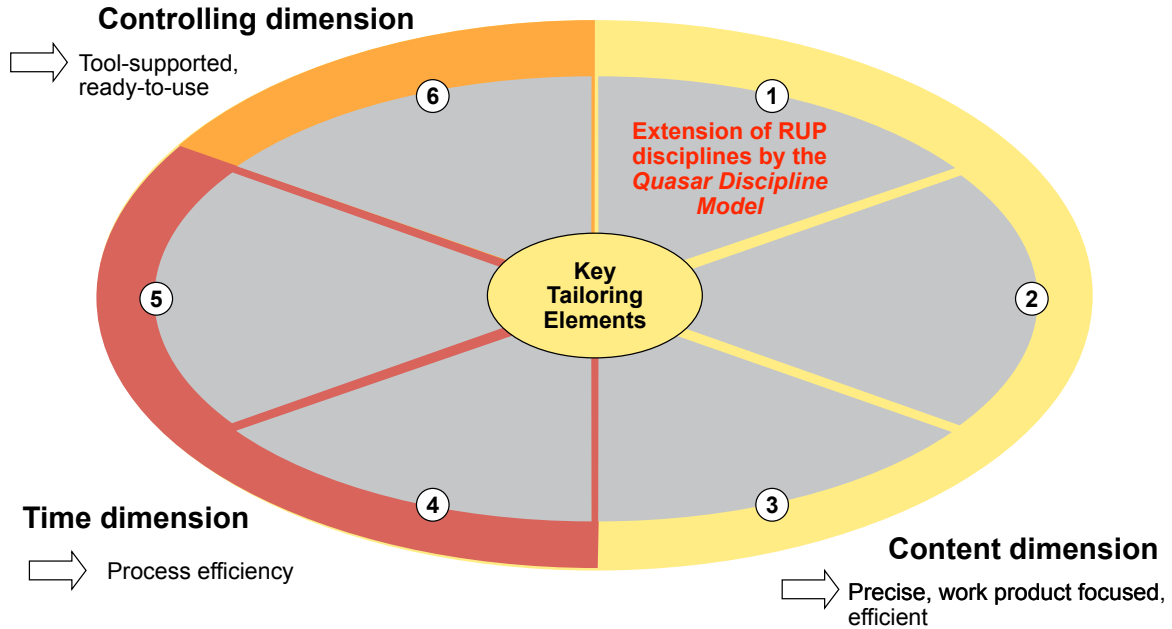


© 2010 Capgemini sd&m - All rights reserved 8

With the *Quasar Disciplines Model*, disciplines reach a detailed & precise level with a special focus on custom software development

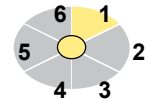


Elements from QUASAR and Stage Model for Integration into RUP

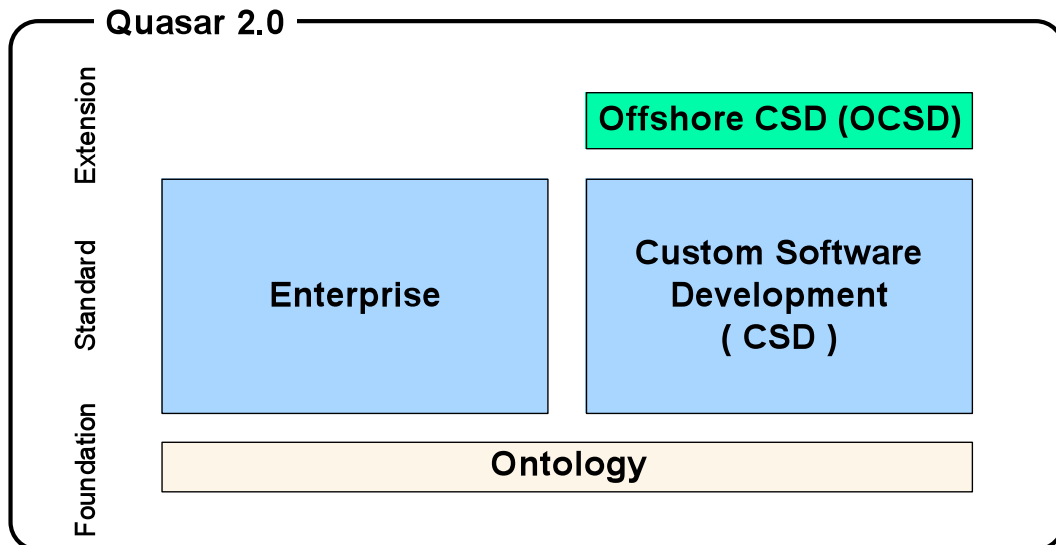


© 2010 Capgemini sd&m - All rights reserved 9

Quasar 2.0 consists of Domains Quasar Enterprise and Quasar CSD, extended with Offshore CSD methodology

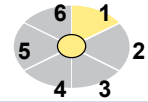


The House of Quasar

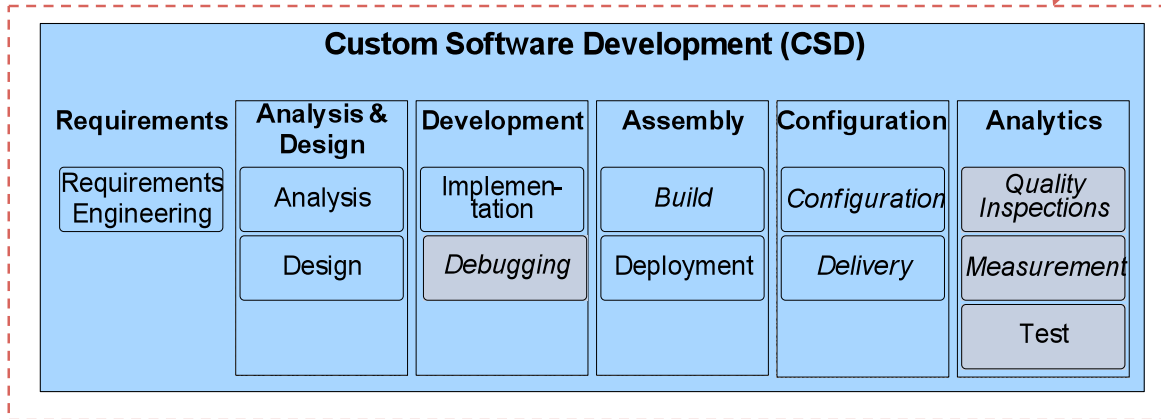
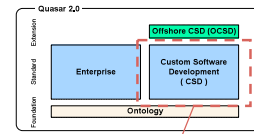


© 2010 Capgemini sd&m - All rights reserved 10

## The Quasar Discipline Model adds new disciplines to the known RUP disciplines, due to the special requirements for CSD



### The Quasar Discipline Model

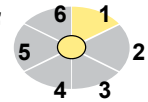


Legend: Domain Constructive Discipline Analytical Discipline

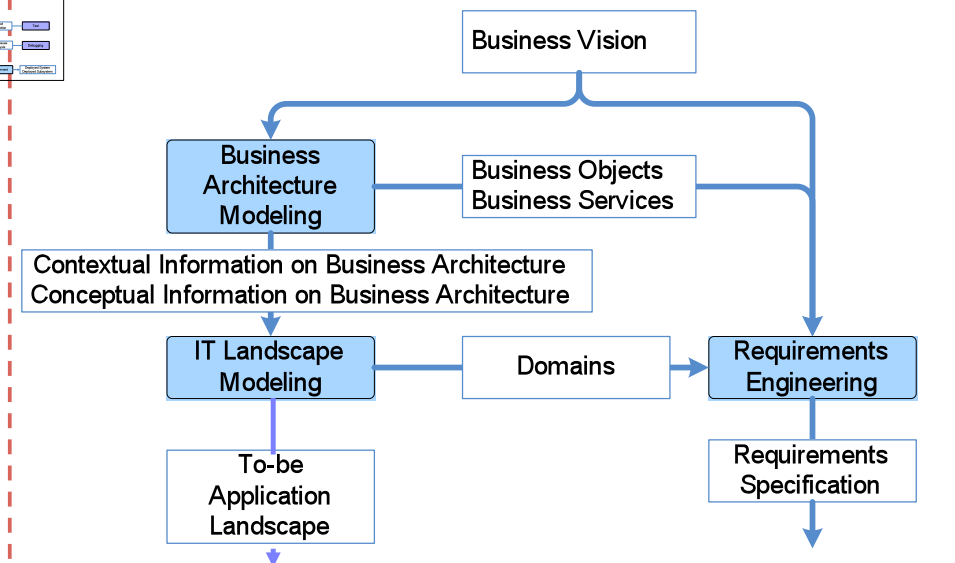
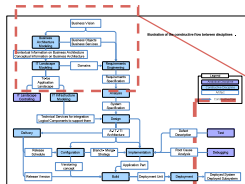
*Italic* = Quasar discipline extending RUP



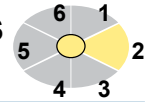
## The Quasar Discipline Model leads the way through the content of software engineering and architecture development projects



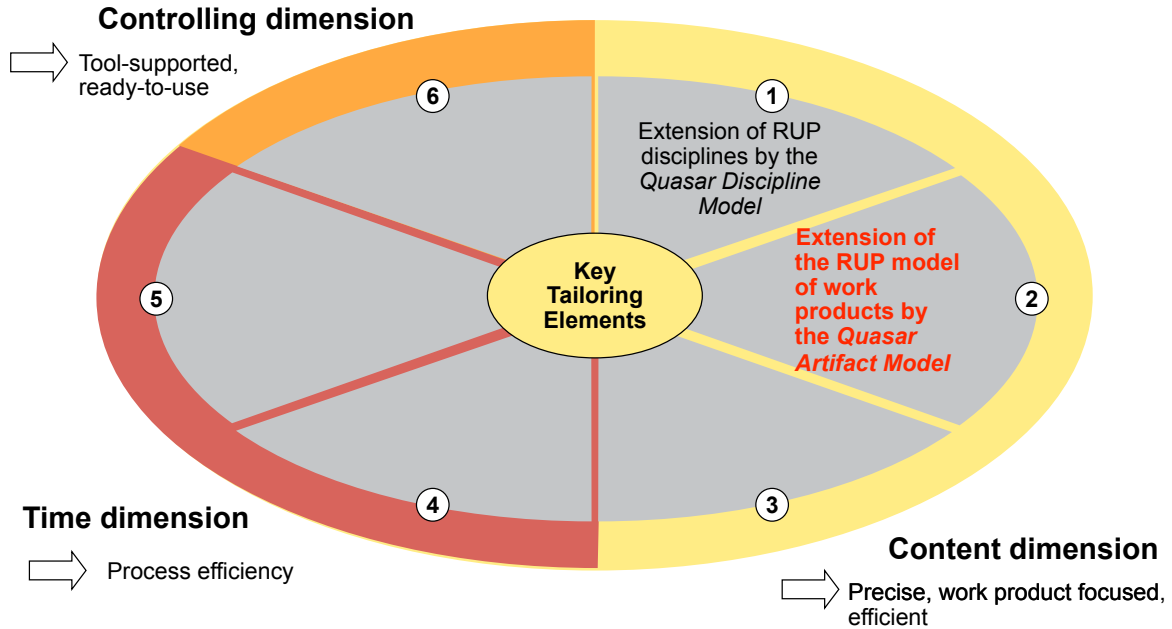
The constructive flow illustrates the transformation of content between the disciplines based on exemplary relevant in- and output artifacts. Further flows are the documentation and the quality flow.



## The Quasar Artifact Model details the RUP model of work products and forms the basis for a product-based controlling

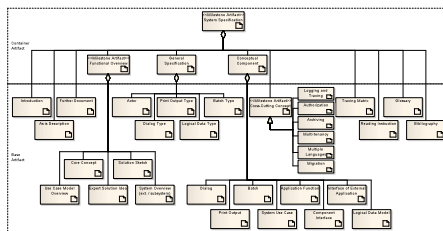
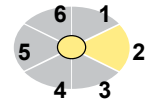


### Elements from QUASAR and Stage Model for Integration into RUP



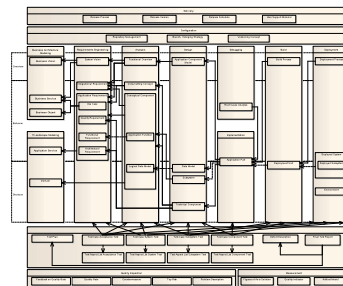
© 2010 Capgemini sd&m - All rights reserved 13

## The Quasar Artifact Model provides an artifacts catalogue and a wallpaper of interdisciplinary dependencies



**The artifact structure diagrams present the hierarchy of artifacts and their associations for each discipline.**

**This results in a complete catalogue of artifacts. Projects applying and tailoring this catalogue will gain transparency about their set of work products.**



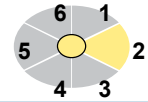
**The interdisciplinary dependencies show which artifacts are used and refined by which disciplines, focused on the most relevant artifacts.**

**Responsibilities and ownership for the artifact development and the interfaces between disciplines become transparent.**

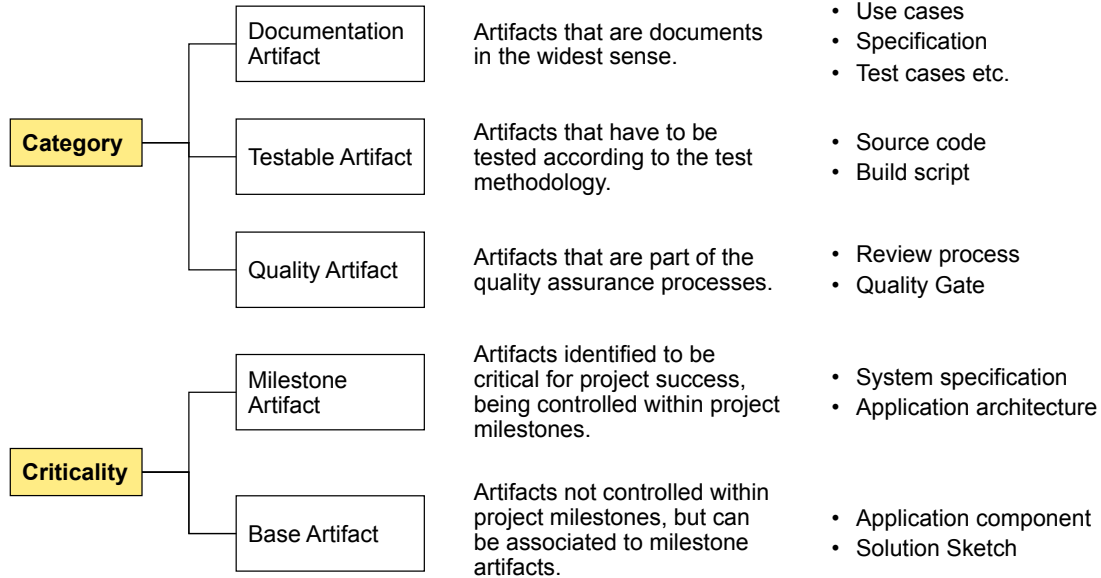


© 2010 Capgemini sd&m - All rights reserved 14

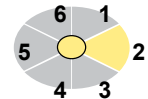
## While the artifact categories preset the respective status model, their criticality determines the level of control



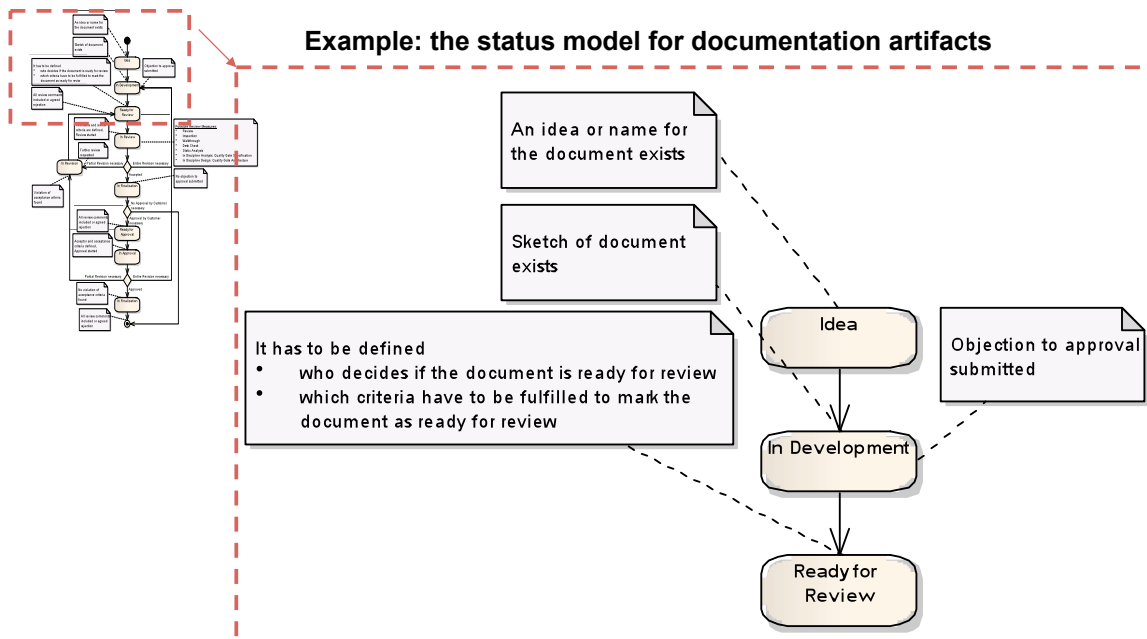
### Quasar Artifact Model: Artifact categories and levels of criticality



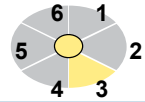
## The Quasar Artifact Model provides a uniform status model as a basis to assess and control the status



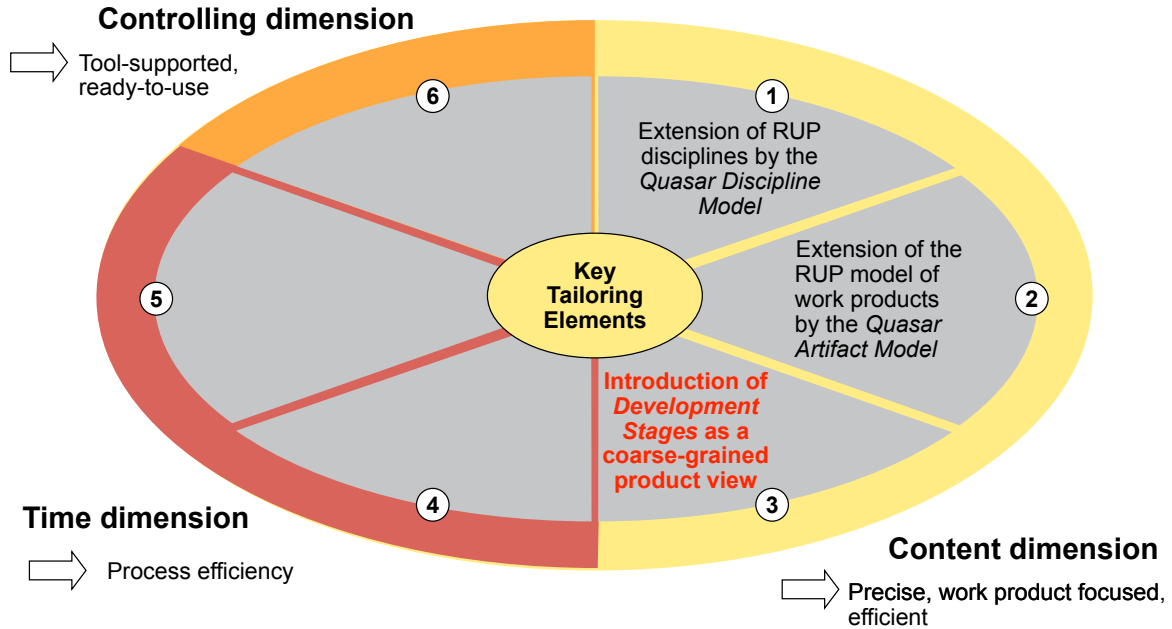
### A different status model applies for documentation artifacts and testable artifacts



## The concept of *Development Stages* strengthens the product view and provides additional process efficiency

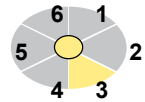


Elements from QUASAR and Stage Model for Integration into RUP

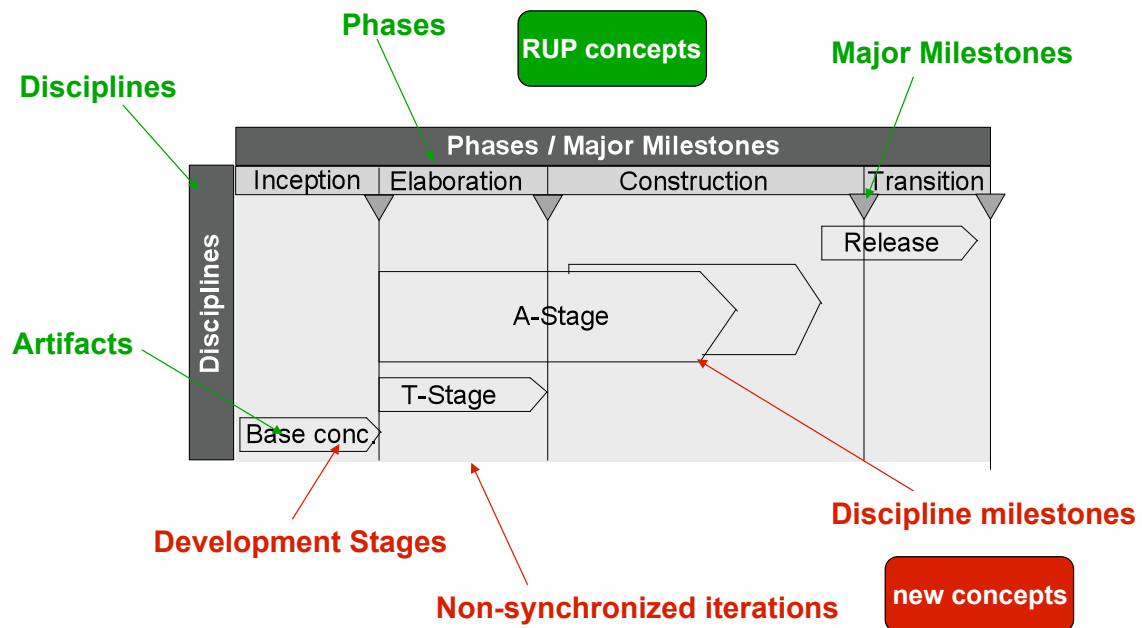


© 2010 Capgemini sd&m - All rights reserved 17

## Capgemini sd&m's classical life cycle model integrates new concepts in the well-known RUP concepts

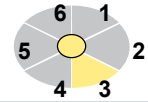


Bird's eye view on Capgemini sd&m's classical life cycle model

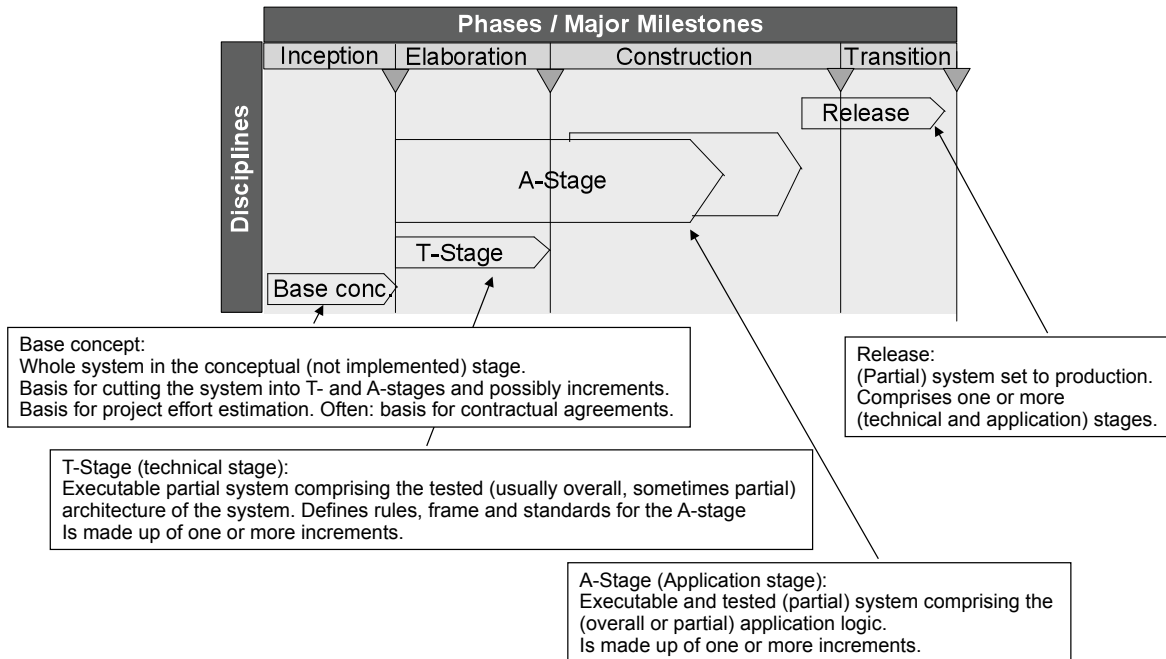


© 2010 Capgemini sd&m - All rights reserved 18

## Development stages represent the product view of the project in the large

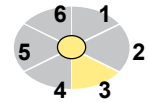


### Definition of Development Stages



© 2010 Capgemini sd&m - All rights reserved 19

## The concept of *development stages* is based on over 25 years of CG sd&m experience in software development



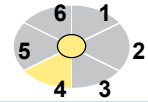
### Advantages of the concept “development stages”

- The development stages support the clear separation of T-stage versus A-stages, T-software versus A-software etc. As such, they allow:
  - a different dynamism of technical basis and business
  - specialization of skills
  - reuse of the technical basis
  - functional cut of A-stages
- Certain types of projects require a project model with large increments. Large increments are better reflected with development stages than with an original RUP model. Especially the integration of larger stages and testing of the whole system is outlined more explicitly in the release stage than in the RUP construction phase.
- The cutting of the project team into sub-projects de facto often occurs by development stage, reducing dependencies between the teams and thereby making project teams more efficient and effective. The development stage often forms the basis for team identification.
- Creation of different development stages often is parallelized, allowing a quicker project execution.

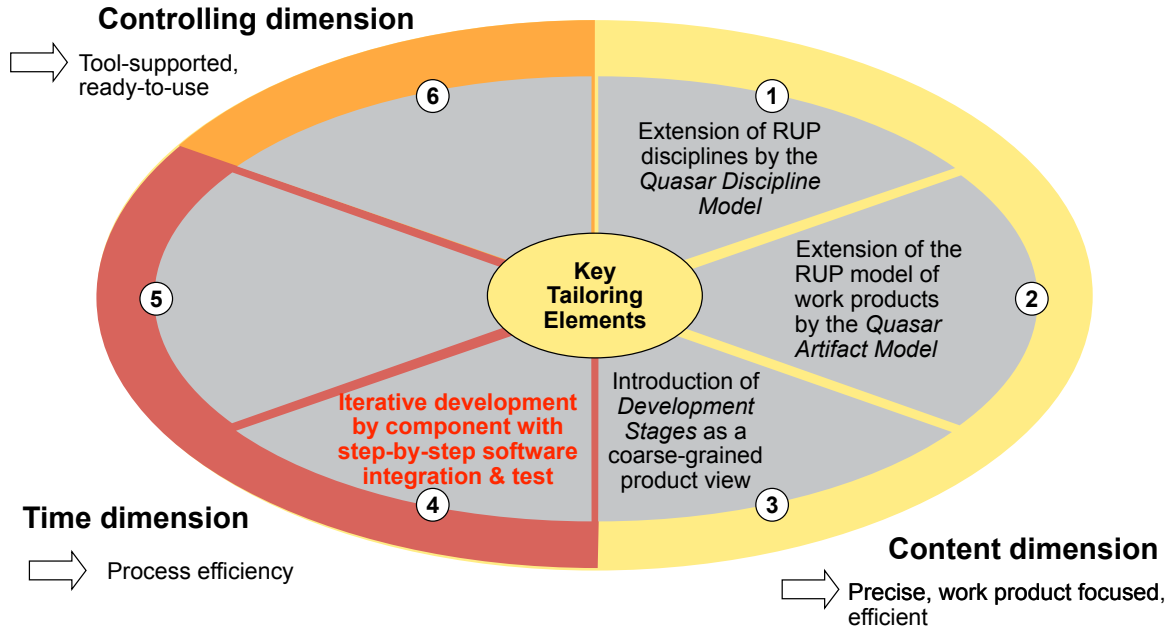


© 2010 Capgemini sd&m - All rights reserved 20

## The iterative development by component with a step-by-step integration and test serves for high quality and efficiency

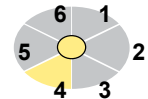


Elements from QUASAR and Stage Model for Integration into RUP

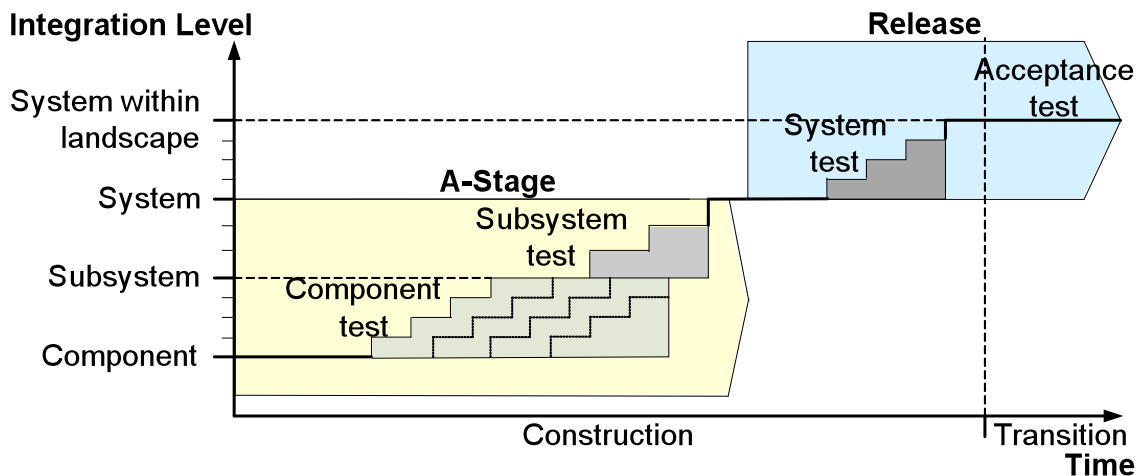


© 2010 Capgemini sd&m - All rights reserved 21

## Integration of the software occurs as early as a component / subsystem has been implemented and internally tested



Step-by-step software integration and test



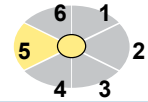
- Component-wise design, implementation, internal and integrative component test
- Subsystem-wise internal and integrative subsystem-test
- Areas of overlapping integration levels



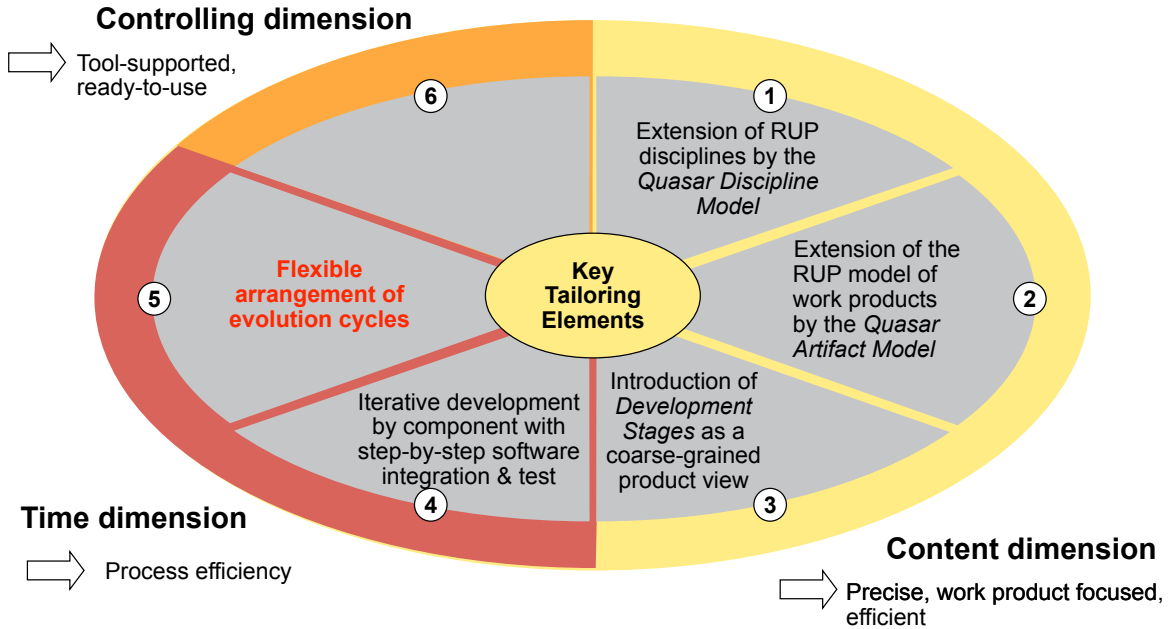
© 2010 Capgemini sd&m - All rights reserved 22



## Flexible arrangement of evolution cycles allows to optimize for the project needs

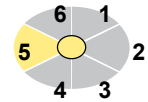


### Elements from QUASAR and Stage Model for Integration into RUP

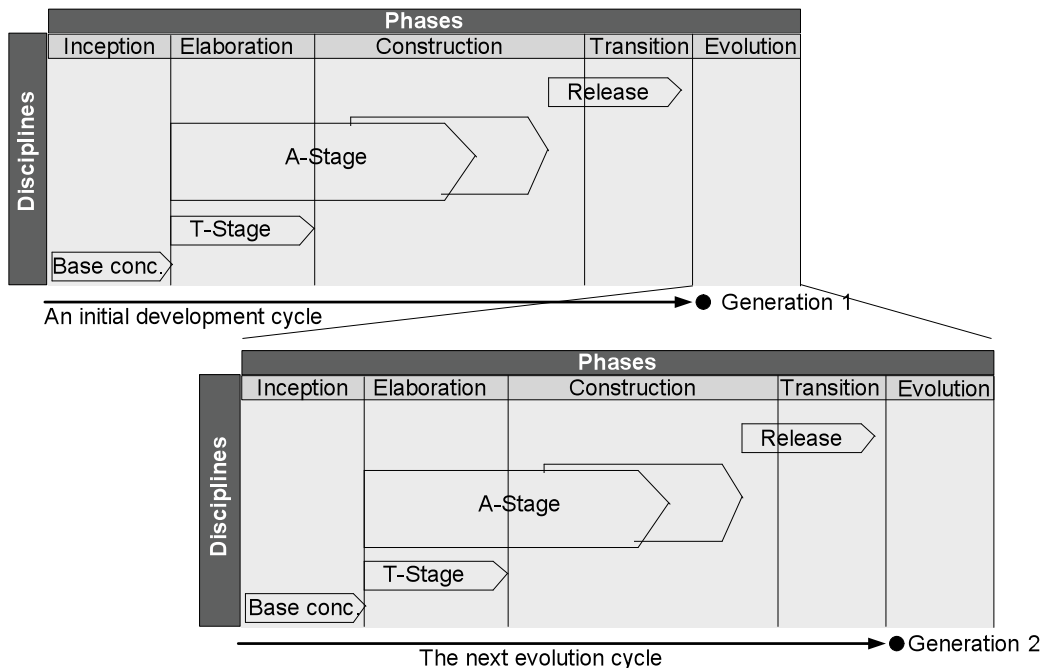


© 2010 Capgemini sd&m - All rights reserved 23

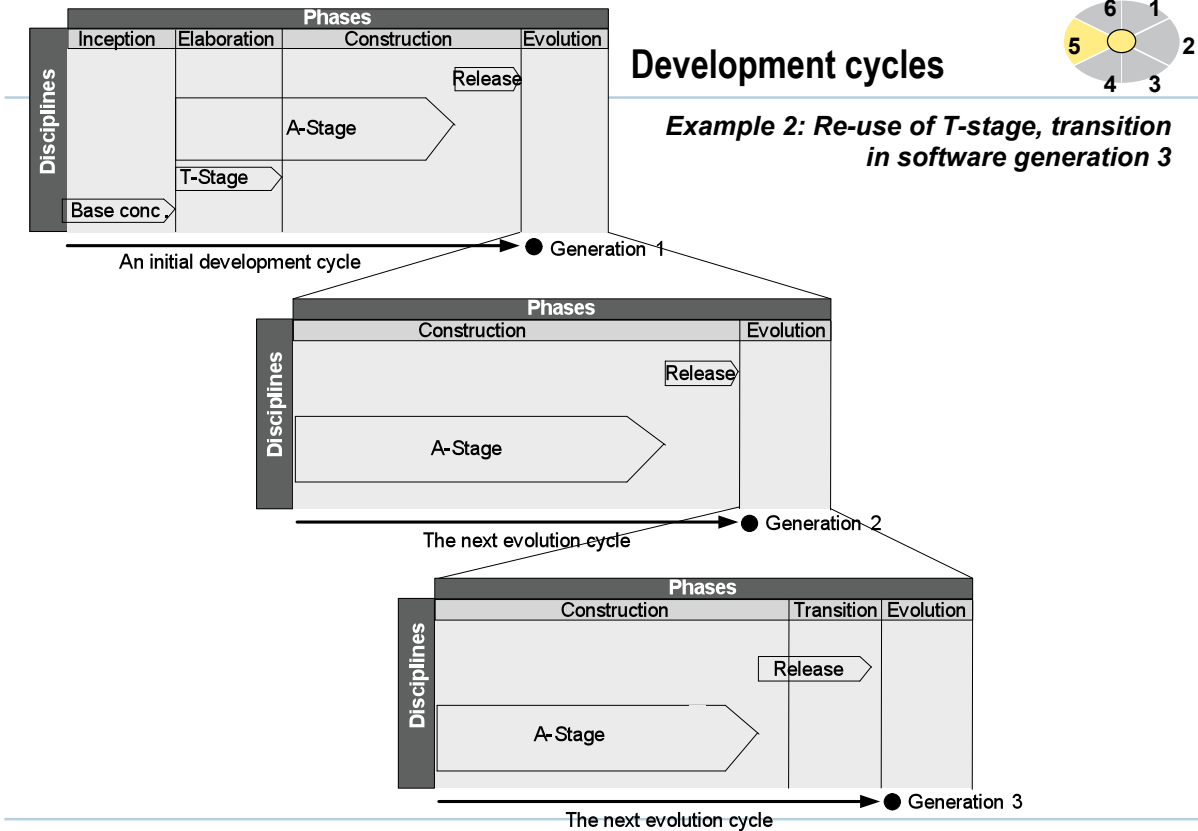
## Generations of software can be produced and delivered via several development cycles



### Example 1: Significant product or architectural redefinition

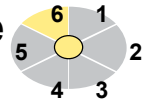


© 2010 Capgemini sd&m - All rights reserved 24

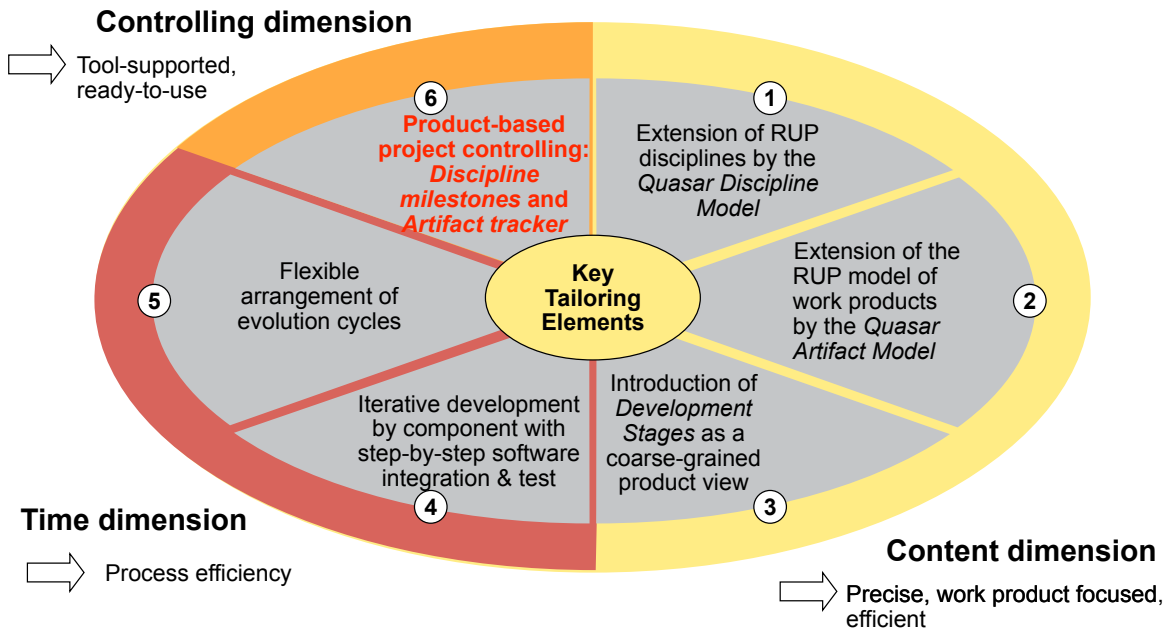


© 2010 Capgemini sd&m - All rights reserved 25

**Discipline milestones and Artifact tracker provide the ready-to-use and tool-supported means for a product-based project controlling**

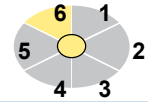


*Elements from QUASAR and Stage Model for Integration into RUP*

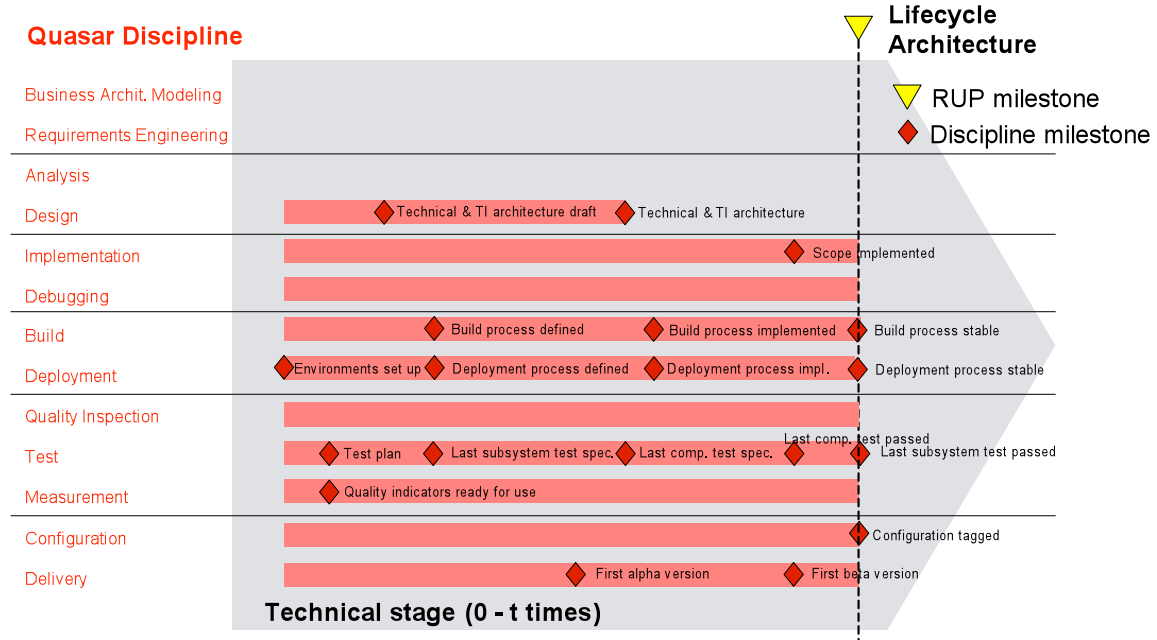


© 2010 Capgemini sd&m - All rights reserved 26

## Discipline milestones form points of hand-shake between the disciplines

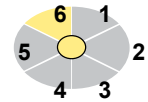


### Zooming into one development stage – example T-stage (technical stage)

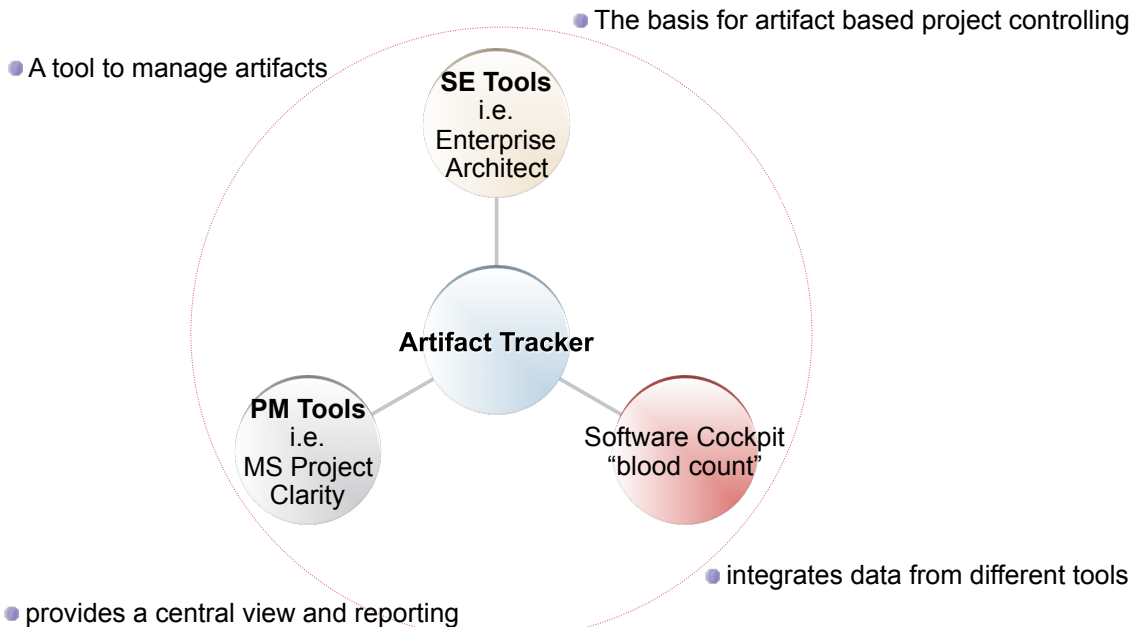


© 2010 Capgemini sd&m - All rights reserved 27

## The Artifact Tracker is the tool for a product-based controlling of the project

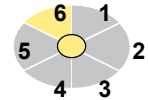


### What is the Artifact Tracker

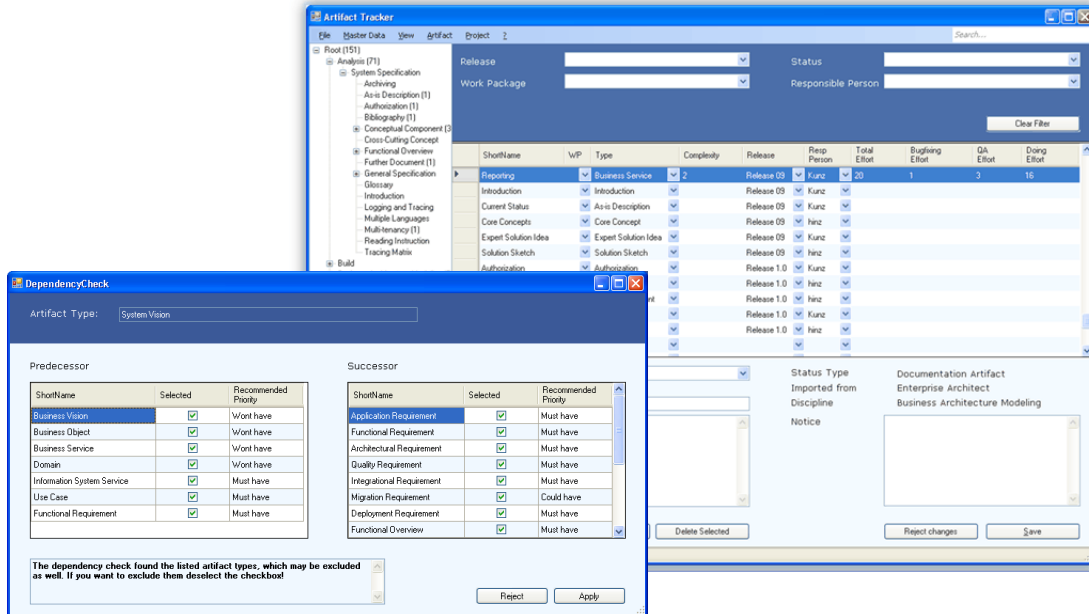


© 2010 Capgemini sd&m - All rights reserved 28

The **Artifact Tracker** sits upon the **Artifact Model**. It provides a dependency check and can be tailored to the project's needs



**Artifact Tracker: GUI main view and dependency check**



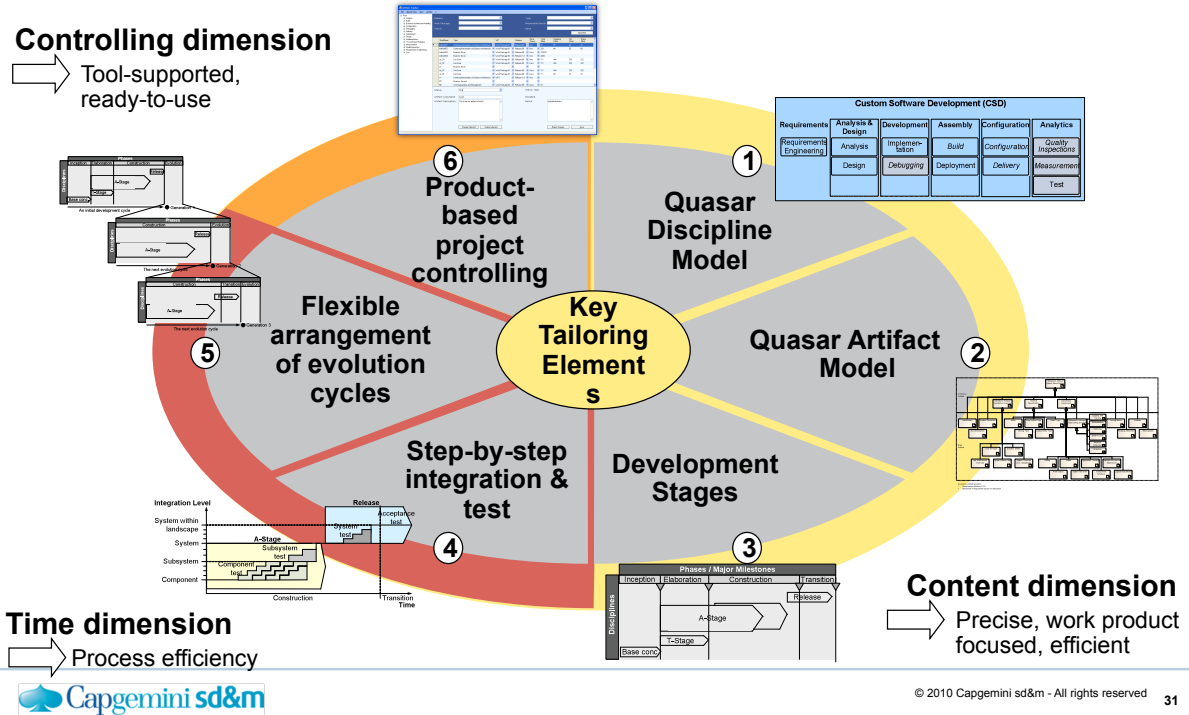
Taken from: thesis of Christopher Schlüter and Daniel Mollnau

## AGENDA

- Motivation
- RUP Extension
- **Summary and Evaluation**

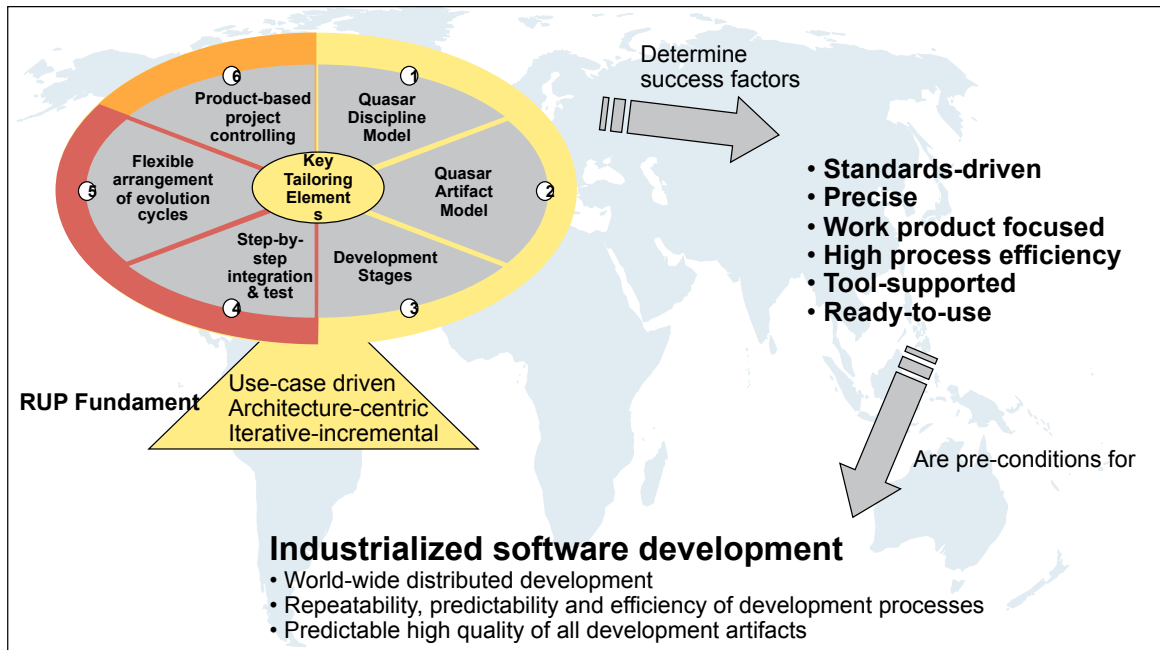
## Capgemini sd&m's classical life cycle model is based on RUP and tailored for precision and a strong product focus together with high efficiency

### RUP extending elements in our classical life cycle model



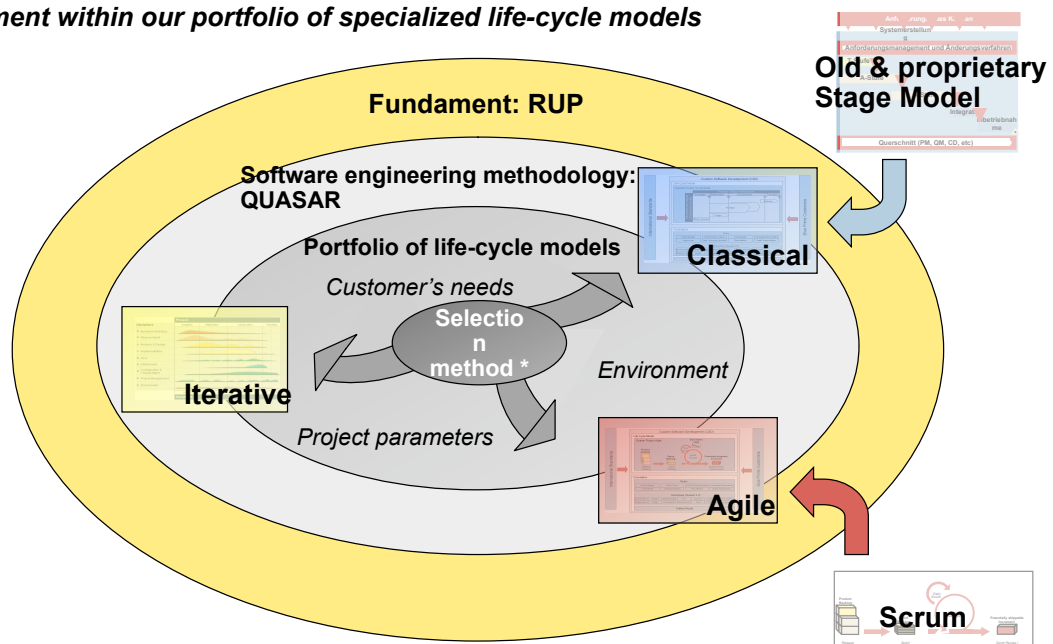
## Capgemini sd&m's classical life cycle model is based on RUP and tailored for the needs of an industrialized software development

### RUP extending elements and their advantage for industrialized software development



## Capgemini sd&m's classical life cycle model is specialized on the needs of large and distributed projects

Placement within our portfolio of specialized life-cycle models



Published in:

M. Heinemann and G. Engels, Auswahl projektspezifischer Vorgehensstrategien, 17. Workshop der Fachgruppe Vorgehensmodelle der GI, 12.-13.4.2010



© 2010 Capgemini sd&m - All rights reserved 33

## Research and teaching depend on the exchange of ideas, a concept which is supported by explicit rules:

The contents of this presentation (among other things texts, graphic arts, photos, logos, etc.) as well as the presentation itself are protected by copyright. Capgemini sd&m holds all rights unless noted otherwise. Capgemini sd&m expressly permits making small parts of the presentation accessible to the public for the purposes of noncommercial teaching and research. Any other use is subject to expressive, written approval of Capgemini sd&m.

### Disclaimer

Even though this presentation, including the cohesive results, has been provided to the best of our knowledge and as subject to diligent review, Capgemini and the authors disclaim all liability for its use.

### Please direct your inquiries to:

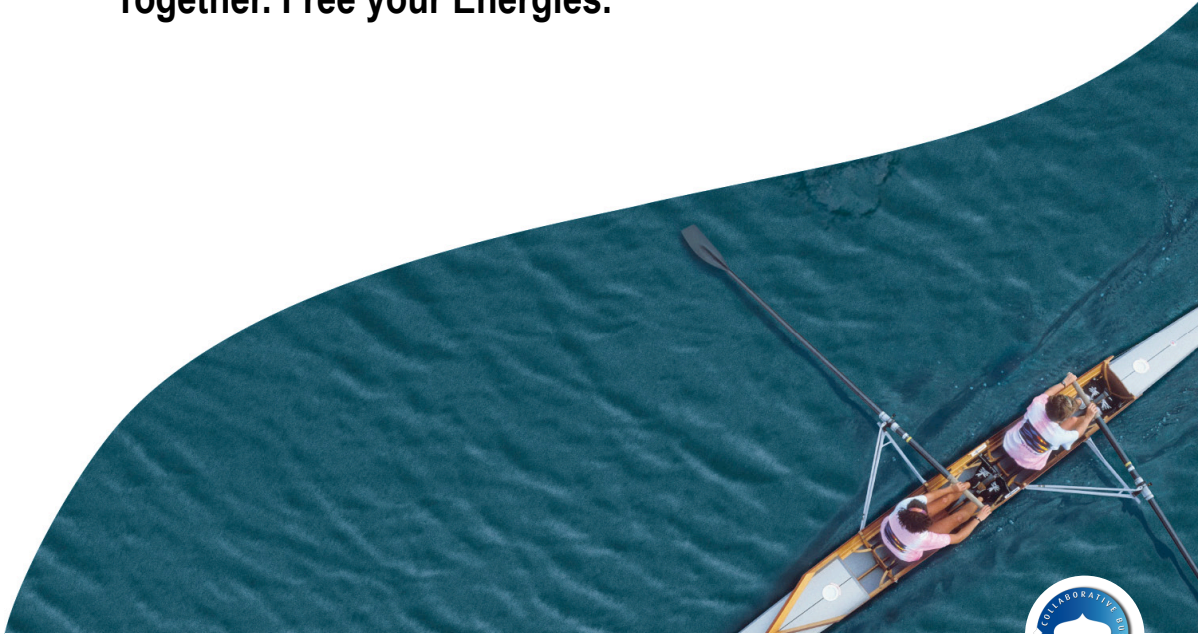
Capgemini sd&m AG  
 Dr. Marianne Heinemann  
 Wanheimer Straße 68  
 40468 Düsseldorf  
 Phone: 0211 56623-255  
[marianne.heinemann@capgemini-sdm.com](mailto:marianne.heinemann@capgemini-sdm.com)



© 2010 Capgemini sd&m - All rights reserved 34



**Together. Free your Energies.**



TOGETHER. FREE YOUR ENERGIES.



## Software EKG

Dipl.-Inf. (FH) Johannes Weigend, Prof. Dr. Johannes Siedersleben

QAWARE, Aschauer Str. 32  
D-81549 München  
johannes.weigend@qaware.de  
johannes.siedersleben@qaware.de

**Abstract:** Das Papier beschreibt die dynamische Vermessung von Systemen und Systemlandschaften mit dem Ziel Anomalien im Laufzeitverhalten zu erkennen. Wir stellen einen werkzeugunterstützten Ansatz zur systematischen Vermessung und Analyse vor, der Transparenz in das Laufzeitverhalten von Systemlandschaften bringt.

### 1 Wie untersucht man kranke Systeme?

Manche Systeme sind krank: Sie werden langsam, verhalten sich eigenartig, produzieren Fehler und reagieren am Ende gar nicht mehr. Andere verhalten sich erratisch, laufen tagelang problemlos, um dann plötzlich ohne Warnung zu abstürzen. Woran kann das liegen? Zuwenig Speicher, zu viele Threads, zu wenig Bandbreite, Programmierfehler, inkompatible Versionen von Softwareprodukten, Fehler in einem Produkt?

Wie der Arzt mit Stethoskop, Ultraschall und EKG in das Innere des Patienten schaut, so stehen auch dem Software-Ingenieur eigene Messgeräten zur Verfügung:

1. Über das Betriebssystem sieht er CPU-Auslastung, Prozesse, den belegten Speicherplatz und die Last auf dem Netz.
2. Über die Plattensysteme sieht er Speicherzugriffe, Wartezeiten und die Verwendung von Caches.
3. Über die Datenbank sieht er die stattgefundenen SQL-Abfragen, die Ausführungsgeschwindigkeit, möglicherweise auch Deadlocks.
4. Über JMX sieht er, was der Garbage-Collector tut, wie viele Objekte erzeugt und freigegeben werden, und er sieht den Speicherverbrauch der virtuellen Maschine.
5. Der Profiler zeigt an, welcher Code wie oft durchlaufen wird, und mit dem Debugger kann er den Programmablauf schrittweise verfolgen.

Diese Liste ist notwendigerweise unvollständig, denn für jedes Betriebssystem, jede Programmiersprache gibt es weitere Messtechniken. Informationen sind jedenfalls genug da, aber trotzdem erweist sich die Diagnose kranker Systeme immer wieder als die Suche nach der Stecknadel im Heuhaufen, vor allem dann, wenn es um Systeme geht,



die auf mehrere Rechner verteilt sind, oft mit verschiedenen Betriebssystemen, Datenbanken und Programmiersprachen, das Ganze verbunden mit einem Web-Client.

Der Grund für diese Misere sind zunächst einmal die handelnden Personen: Da sind die verschiedenen Experten für Betriebssystem, Datenbank oder Web-Anbindung, die sich in der Regel in ihrem Fachgebiet sehr gut auskennen, aber von den anderen Bereichen – vor allem von der Anwendung selbst – zu wenig wissen. Dann gibt es die Projektmitarbeiter, vom Programmierer bis zum Chefarchitekten, die ihre Anwendung kennen, aber wenig wissen über die Internas der technischen Infrastruktur.

Diese verschiedenen, im Extremfall disjunkten Kompetenzen sind regelmäßig Anlass für ein Kommunikationsproblem, das dadurch verstärkt wird, dass alle Beteiligten Fehler im jeweils eigenen Zuständigkeitsbereich ausschließen: Der Software-Architekt erlaubt nicht, dass man seinen Entwurf in Frage stellt, der Programmierer weist den Vorwurf zurück, er habe seine Threads falsch synchronisiert, und der Microsoft-Berater hält es für abwegig, eine Microsoft-Komponente zu verdächtigen.

Aber disjunkte Kompetenzen und psychologische Empfindlichkeiten sind keine ausreichende Erklärung für die oft beobachtete Hilflosigkeit bei der Analyse kranker Systeme. Der wichtigste Grund lautet nämlich: Wir arbeiten unprofessionell. Die genannten Messverfahren werden meistens ad hoc eingesetzt, man misst über zufällige Zeiträume und in der Regel viel zu kurz, manchmal in der Produktion, manchmal in der Testumgebung, die Messprotokolle liegen in unterschiedlichen Formaten in diversen Verzeichnissen herum und werden mit trickreichen grep-Aufrufen durchforstet. Weil es für die verschiedenen Messungen keine gemeinsame Zeitachse gibt, lässt sich kaum feststellen, wie die Benutzer mit ihrem Verhalten das System beeinflussen. Im Übrigen skaliert der manuelle Ansatz nicht: Zehn Rechnern mit zehn Prozessen und zehn Messwerte pro Prozess ergeben 1000 Messwerte. Kein Mensch kann so etwas manuell verfolgen geschweige denn auswerten.

Das ist die Situation in vielen Projekten. Aber die korrekte, schnelle Diagnose sollte das Ergebnis einer systematischen Analyse sein, und nicht der Geistesblitz eines Genies, das zufällig in der Nähe ist. Das war jedenfalls unser Ziel. Heute, nach mehreren erfolgreichen Projekten, besitzen wir ein Vorgehen und einen Werkzeugkasten für die systematische Analyse kranker System. Das Prinzip lässt sich in sechs Punkten zusammenfassen:

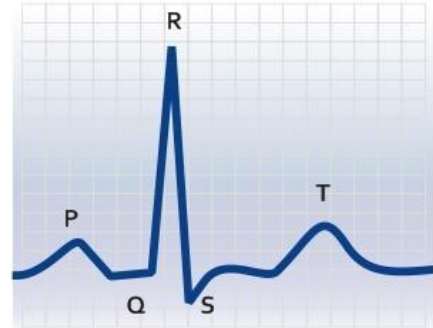
1. Wir beobachten das System über definierte, oft sehr lange Zeiträume. Wir verwenden jede Art von Messung, auf jeder Ebene (Betriebssystem, Datenbank, Application Server, o.a.), auf verschiedenen Rechnern.
2. Wir verwenden fertige Schnittstellen zu den gängigen Messwerkzeugen; neue oder exotische Umgebungen lassen sich mit geringem Aufwand anbinden. So wäre die Messung eines Cobol/CICS-Systems kein Problem. Wir legen alle Messwerte in einer einzigen zentralen Datenbank ab, die sehr groß werden kann, und deren Rückgrat die Zeitachse ist. Die beteiligten Rechner können irgendwo auf der Welt stehen. Diese Datenbank gestattet zahlreiche Auswertungen und Visualisierungen.

3. Wir messen nicht nur das Verhalten des Systems, sondern auch das der Benutzer über dieselben Zeiträume. Wir speichern alle Protokolle in derselben Datenbank mit derselben Zeitachse.
4. Dadurch sind wir in der Lage, Systemanomalien (z.B. eine drastisch erhöhte CPU-Last) durch vertikale Schnitte über alle Messungen und Protokolle hinweg zu analysieren. Wir sehen genau, was auf den verschiedenen Ebenen in den verschiedenen Rechnern in einem bestimmten Zeitpunkt passiert ist – und das ist genau die Information, die man für die Diagnose braucht. Wir kennen inzwischen eine Reihe typischer Krankheitsbilder, genauso wie die Ärzte bei den EKGs.
5. Wir erkennen synchrone Änderungen und vermutete Abhängigkeiten (Messwert x erhöht sich immer dann, wenn Messwert y sinkt) mit Hilfe eines einfachen, selbst geschriebenen Tools.
6. Erst jetzt setzen wir – wenn nötig – Profiler und Debugger ein, und zwar nicht aufs Geratewohl, sondern genau am Krankheitsherd.

Das Papier erklärt unseren Ansatz im Detail und macht ihn für jedermann nachvollziehbar. Die zentrale Datenbank mit den vielfältigen Auswertungen ist eine gemeinsame Informationsbasis für die unterschiedlichen Experten, die man natürlich immer noch braucht. Aber jetzt können sie sich endlich unterhalten, und gegen psychologische Empfindlichkeiten helfen gute Argumente, die sich mit unserer Datenbank untermauern lassen.

## Software-EKG Systematische Untersuchung kranker Systeme

SEE Köln, 3. Mai 2010  
Prof. Dr. Johannes Siedersleben,  
Dipl.-Inf. (FH) Johannes Weigend



11.05.2010

### QAware-Manifest zum Software Engineering

1. Services statt Silos
2. Komponenten statt Monolithen
3. Standardsoftware als Partner statt Konkurrent
4. Stufen statt Mammutprojekte
5. Virtualisierung statt Hardware-Zoo
6. Software-Controlling statt Beten und Hoffen
7. **Systematische Software-Diagnose statt planloser Suche**
8. Professionelle Programmierung statt Spaghetti-Code
9. Technikverliebtheit statt Kundenorientierung
10. Augenmaß statt Utopie

Mehr dazu auf [www.qaware.de](http://www.qaware.de)

## Agenda

---

1. Wo ist das Problem?
2. Was ist der Software-Elektrokardiograph (SW-EKG)?
3. Beispiele für Datenquellen: Nagios, JMX, Windows Performance Counter
4. Architektur des SW-EKG
5. Wie organisiert man eine Rettungsaktion?
6. Typische Analyse-Ergebnisse
7. CASD (Computer Aided Software Diagnosis)
8. Beispiele am laufenden System
9. SW-EKG von innen

3

## Wo ist das Problem?

---

### Typische Situation

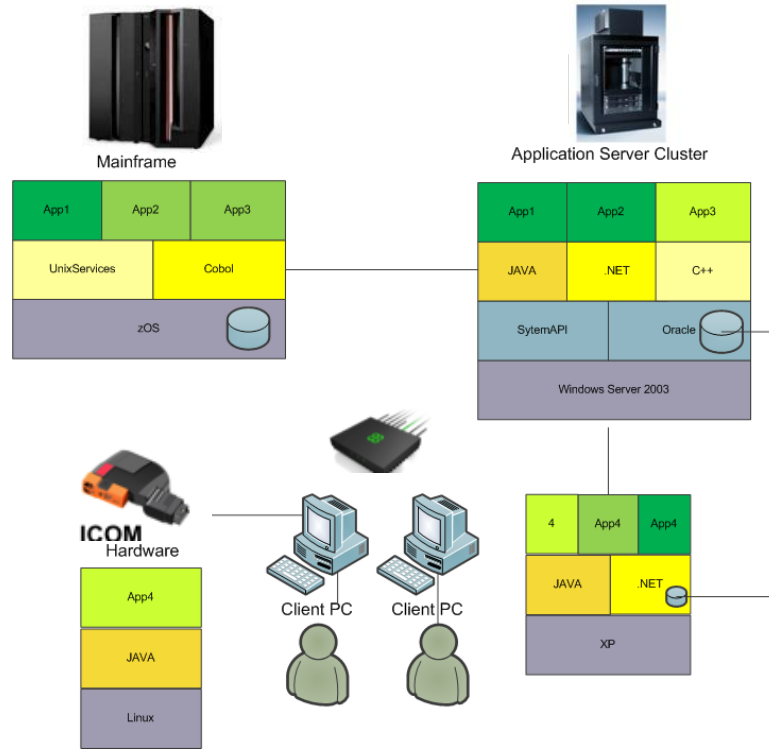
- Unklare Symptome: gelegentliche Abstürze, System wird quälend langsam, Neustart hilft nur vorübergehend.
- Heterogene Systeme: Oft verschiedene Programmiersprachen und Betriebssysteme im Verbund.
- Software verschiedener Hersteller und Lieferanten verteilt auf mehrere Rechner, Prozesse, Threads.

### Offene Fragen

- Debugging, Profiling: ja, aber wo?
- Monitoring: Viele Datenquellen (z.B. Taskmanager, JMX, Windows Performance Counter, Nagios), aber wie führt man die Messwerte zusammen?
- Mengenproblem: 10 Rechner mit je 10 Prozessen mit je 10 Threads, 10 Stunden lang 10 Messungen pro Minute → 6 Mio Messwerte
- Speicherung, Dokumentation, Nachvollziehbarkeit?
- Wer analysiert? Probleme: Disjunkte Kompetenzen, verletzte Eitelkeiten, Rechthaberei, zu viel Druck im Kessel.

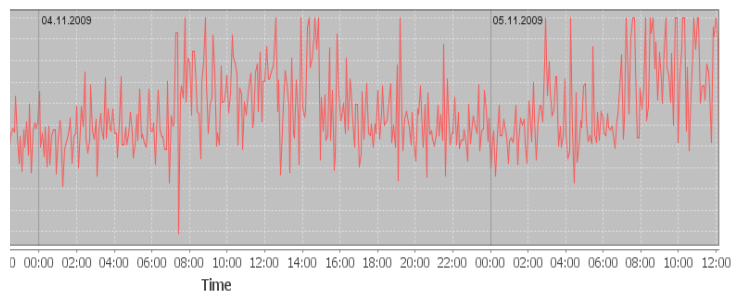
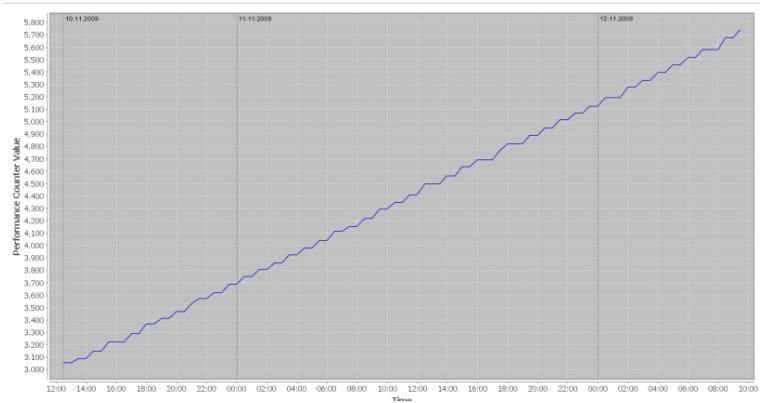
4

## Ein komplexes System



5

## Eine gefundene Stecknadel



6

## Der Software-Elektrokardiograph (SW-EKG)

---

1. Systematische Beobachtung verschiedener Kennzahlen und des Benutzerverhaltens über kurze und lange Zeiträume.
2. Speicherung aller Messdaten in einer Datenbank; komfortable Funktionen zum Verdichten und Visualisieren der gemessenen Kennzahlen.
3. Schnittstellen zu gängigen Messwerkzeugen (JMX, Performance Counter, Nagios), Offenheit gegenüber künftigen Tools.
4. Erkennung von Anomalien durch vertikale Schnitte, Unterstützung beim Finden typischer Korrelationen durch einfaches Tool.

SW-EKG = Vorgehen + Tool + technische Kompetenz

SW-Engineering = Handwerk (80%) + Kreativität (20%)

7

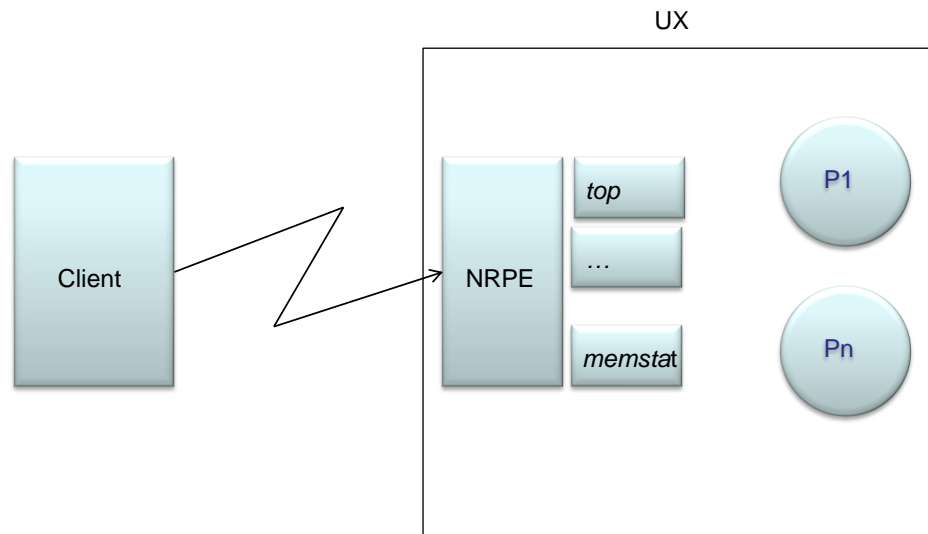
## Einordnung und Abgrenzung

---

- Einsatzszenarien:
  - Rettungsaktion: System erweist sich kurz vor, bei oder nach der Einführung als krank → Druck im Kessel.
  - Optimierung: Durchsatzverbesserung bei unveränderter Hardware.
  - Prophylaxe: SW-Messung begleitend zum Entwicklungsprozess, bevor das Kind in den Brunnen fällt.
  - Qualitätssicherung bei Integration und Abnahme.
- Keine Annahmen über SW-Architektur, Plattform, Vorgehensmodell im Projekt: Wir analysieren (fast) alles.
- *Statische Analyse* (Sotograph, SonarJ, Structure 101): Interessant, aber ganz andere Baustelle.
- Vorhandene Tools (wie Taskmanager, Perfmon) nur geeignet für kurze Ad-Hoc-Messungen bei überschaubar vielen Messwerten.

8

## Nagios



NRPE = Nagios Remote Plugin Executor  
 top, memstat = Beispiele für UX-Kommandos  
 P1, ..., Pn = zu beobachtende Prozesse

9

## JMX – Java Management Extension

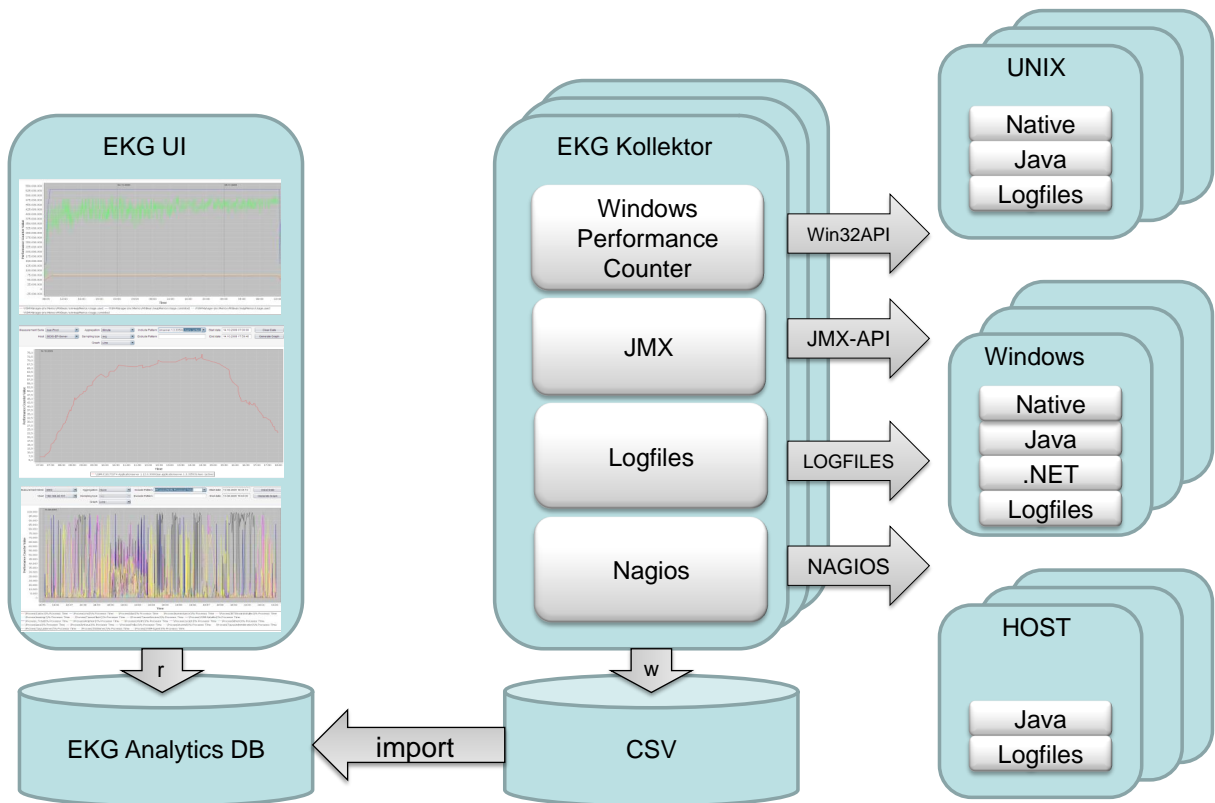
```
// 1. Connect to JMX server
String jmxConnectionUrl =
    "service:jmx:rmi:///jndi/rmi://<<host>>:<<port>>/jmxrmi;
JMXConnector connector =
    JMXConnectorFactory.connect(new JMXServiceURL(jmxConnectionUrl));
MBeanServerConnection connection = connector.getMBeanServerConnection();

// 2. Access JMX beans...
MemoryMXBean mem =
    ManagementFactory.newPlatformMXBeanProxy(connection,
    ManagementFactory.MEMORY_MXBEAN_NAME, MemoryMXBean.class);
// for other beans identical

// 3. Call bean methods multiple times
mem.getHeapMemoryUsage().getUsed();
```

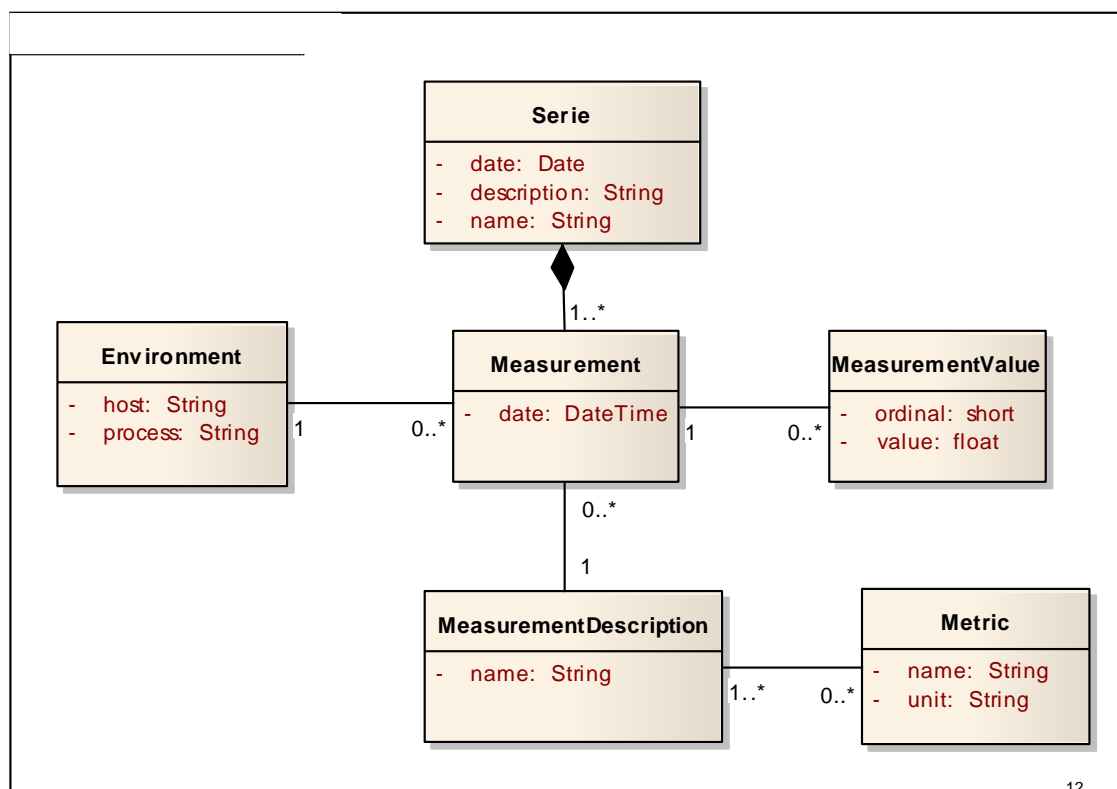
10

## Architektur des Software-EKG



11

## Das SW-EKG-Datenmodell



12



## Typische Analyse-Ergebnisse (nicht überraschend)

- **Leaks** (Speicher, Threads, Handles): Man braucht ein Konzept.
- **CPU-Verbrauch**: Naive Programmierung, exzessive Serialisierung, Garbage Collection in Kombination mit Speicherloch.
- **Locks** (Short locks, Deadlocks): Naive Programmierung; auch hier braucht man ein Konzept.
- **I/O-Last**: Logging (wird serialisiert), Lesen/Schreiben großer Objektklumpen, naiver Datenbank-Entwurf (Normalisierung), naiver Datenbankzugriff.
- **Wie beugt man vor?**
  - Bewusstsein, Aufklärung
  - Gezielte QS gegen Flüchtigkeitsfehler, fehlende oder mangelhafte Konzepte, mangelhafte Architektur
  - Software-Controlling (systematischer Einsatz von EKG und anderen Prüftools in der Entwicklung → eigener Vortrag)
  - Ausbildung !

13

## Thread-Programmierung ist schwer: Double-Checked Locking Just Doesn't Work

```

class Singleton {

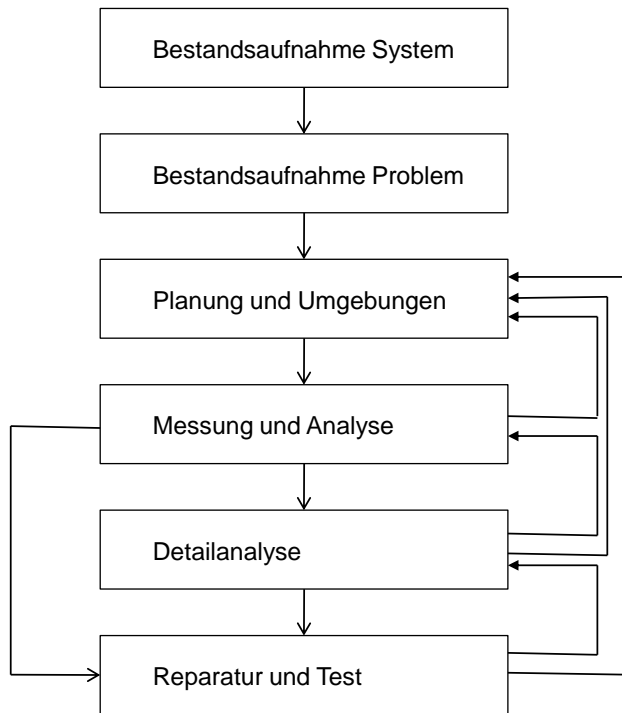
    private Singleton instance = null;

    public static Singleton getInstance() {
        if (instance == null) { // das geht schief!
            synchronized(this) {
                if (instance == null) {
                    instance = new Singleton();
                    // initialization
                    // ...
                }
                return instance;
            }
        }
    }
    // more methods and members
}

```

14

## Wie geht man bei einer Rettungsaktion vor?

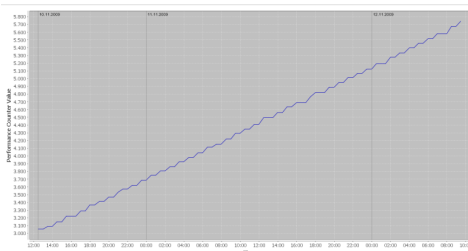


### Erfolgsfaktoren:

- Rückendeckung durch höheres Management
- Quick Wins, vor allem zu Beginn
- Kooperative Ansprechpartner für alle relevanten Themen
- Kompetenter Chefdetektiv
- Brauchbare Messumgebungen  
→ Virtualisierung

15

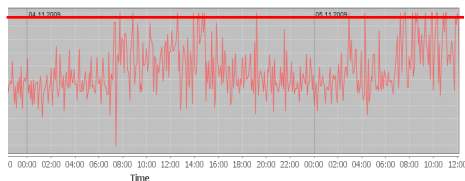
## CASD (Computer Aided Software Diagnosis) /1



### Leak-Erkennung (allgemein)

#### → Sättigungsanalyse

Median der Werte ( $W$ ) in konstant breiten ( $d$ ) Sampling-Bereichen bilden. Leak, wenn der Wert in  $m$  aufeinander folgenden Bereichen steigt.



### Leak-Erkennung (Memory, indirekt über CPU Verbrauch)

#### → Grenzwert-Analyse

Der CPU-Verbrauch durch den Garbage Collector überschreitet mit einer Frequenz  $f$  den Grenzwert von  $I$  (ca. 95%). „CPU-Kammerflimmern“-Muster.

### • Lock-Erkennung

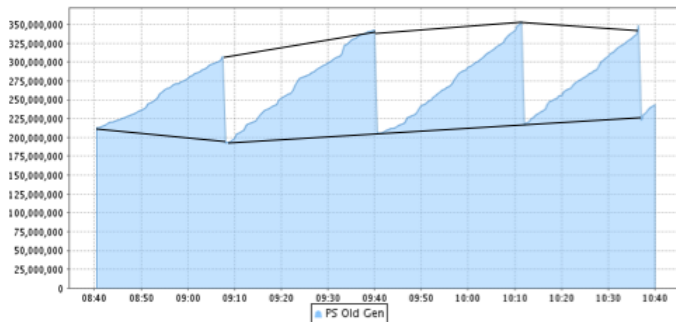
→ Grenzwert-Analyse: CPU-Verbrauch  $\cong 0$  UND Netzwerk-I/O  $\cong 0$  UND HD-I/O  $\cong 0$

### • Hohe I/O Last

→ Grenzwert-Analyse: CPU-Wait-Time > 95% UND (Netzwerk-I/O ist signifikant hoch ODER HD-I/O ist signifikant hoch)

16

## CASD (Computer Aided Software Diagnosis) /2



### Leak-Erkennung (Memory, direkt) → Sättigungsanalyse

Wir ermitteln Maximal- und Minimalwerte der Sägezähne. Leak, wenn der Maximal- oder Minimalwert über n Sägezahn-Phasen hinweg steigt.

### Unser Ziel:

- Automatische Erkennung bekannter Anomalien,
- Rahmen, mit dem man neue Anomalien einfach beschreibt und integriert.

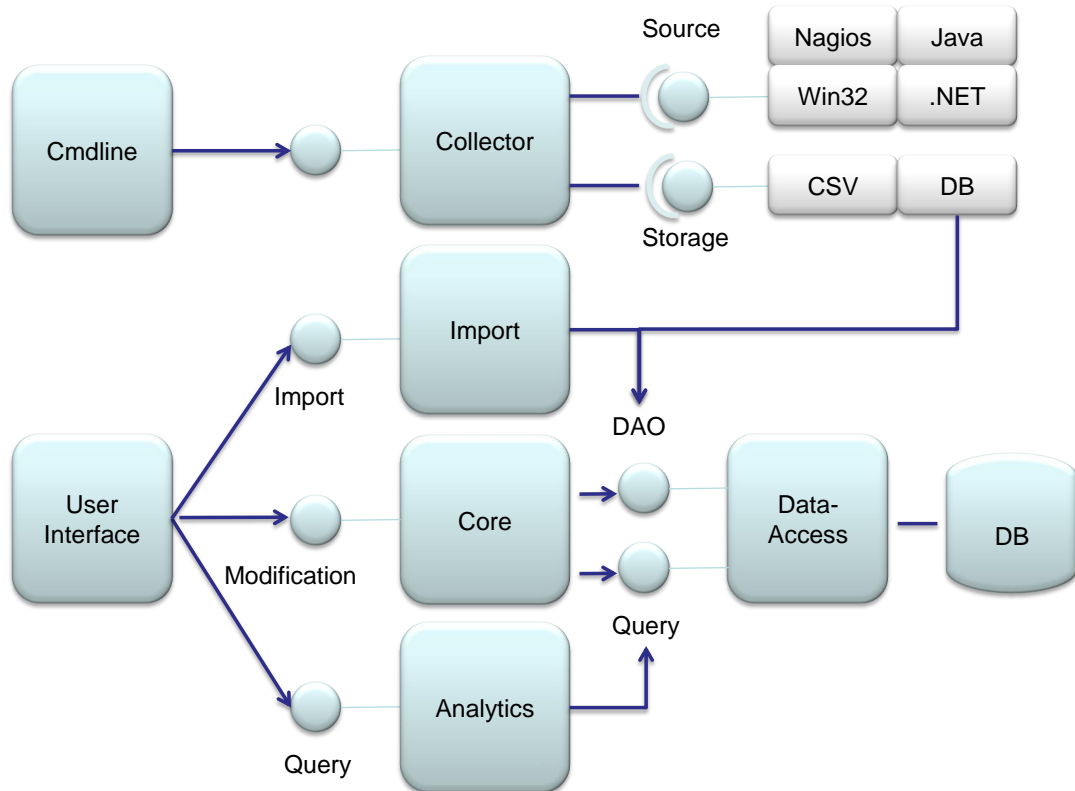
17

## Live Demo

- CPU Hotspots
- Thread-Leaks
- Speicher / GC-Korrelation
- Langzeitmessungen

18

## SW-Komponenten und Schnittstellen machen das EKG flexibel erweiterbar



## Performance Counter Interface

```

package de.qaware.ekg.collector.source;

/**
 * Access performance counter.
 *
 * @author Johannes Weigend, Josef Adersberger, Florian Lorenz
 */
public interface Counter {

    /**
     * Request a single counter value for a given counter.
     *
     * @param counter the counter.
     * @return the counter.
     */
    public double getCounter(String counter);
}

```

## Windows Performance Counter – JNI Bridge

```

JNIEXPORT jdouble JNICALL Java_de_qaware_ekg_collector_source_wincount_
WindowsPerformanceCounter_getCounter
(JNIEnv * env, jobject obj, jstring counter)
{
    PDH_FMT_COUNTERVALUE pValue = {0};
    PDH_STATUS            pdhStatus = {0};
    Counter counterInfo = {0};
    const TCHAR * szCounterPath = toLPCWSTR(env, counter);

    // Open a query object.
    pdhStatus = ::PdhOpenQuery (NULL, 0, &(counterInfo.hQuery));
    // ... /// check status

    // Add a counter providing the data.
    pdhStatus = ::PdhAddCounter (counterInfo.hQuery,
                                szCounterPath,
                                0,
                                &(counterInfo.hCounter));
    // ... /// check status

    pdhStatus = ::PdhCollectQueryData (counterInfo.hQuery);
    // ... /// check status

    DWORD dwType;
    pdhStatus = ::PdhGetFormattedCounterValue (
                                                counterInfo.hCounter, PDH_FMT_DOUBLE, &dwType, &pValue);
    // ... /// check status
    if (szCounterPath != NULL) {
        delete [] szCounterPath;
    }
    return pValue.doubleValue;
}

```

21

## Qualität und Agilität im Software Engineering



QAware GmbH  
 Aschauer Str. 32  
 81549 München  
 Tel 089 6809210-0  
[www.qaware.de](http://www.qaware.de)

## Einsatz von Software-Engineering-Methoden im DLR

Tobias Schlauch, Andreas Schreiber

Deutsches Zentrum für Luft- und Raumfahrt e.V.,  
Simulations- und Softwaretechnik

Tobias.Schlauch@dlr.de, Andreas.Schreiber@dlr.de

**Abstract:** Im Vortrag sollen die DLR-Aktivitäten im Bereich des Software-Engineerings vorgestellt werden. Insbesondere die grundlegende Vorgehensweise sowie die etablierten Services und Werkzeuge sollen anhand von plakativen Beispielen dargestellt werden.

### 1 Motivation

Das Deutsche Zentrum für Luft- und Raumfahrt e.V. (DLR) ist das Forschungszentrum der Bundesrepublik Deutschland für Luft- und Raumfahrt. Zur Erfüllung des Forschungsauftrages wird in verschiedenen Bereichen des DLR Software entwickelt. Analysen dazu haben ergeben, dass von insgesamt 6500 Mitarbeitern zwischen 2000 und 3000 Mitarbeiter in allen 13 DLR-Standorten in Vollzeit oder Teilzeit an der Entwicklung von Software für komplexe Anwendungen aus allen Bereichen der Luft- und Raumfahrt beteiligt sind. Umgerechnet auf Vollzeitstellen sind dies mehr als 1000 Mitarbeiter. Die in diesem Zusammenhang entwickelte Software reicht von eingebetteter Software für Flugzeuge, Satelliten oder Raumfahrzeuge, über Unterstützungssoftware beispielsweise zum Datenmanagement oder zur Prozessunterstützung, bis hin zu Software mit hohen Effizienzanforderungen wie Strömungssimulationswerkzeuge.

Der Umfang dieser Software-Entwicklungen reicht typischerweise vom kleinen Ein-Mann-Projekt bis hin zu kooperativen Projekten mit bis zu 50 beteiligten Wissenschaftlern. Viele der Wissenschaftler, die z.T. sehr umfangreiche, komplexe Softwarepakete entwickeln, besitzen jedoch nur geringe Kenntnisse im Bereich des Software-Engineerings. Hinzu kommt die im wissenschaftlichen Umfeld übliche Fluktuation der Mitarbeiter, wodurch z.T. das Wissen über einmal entwickelte Systeme mit den Mitarbeitern das Unternehmen verlässt. Ein weiteres oft zu beobachtendes Phänomen ist, die Tatsache, dass häufig Systeme klein anfangen und durch Bewährung bis zu komplexen beispielsweise Strömungslösern anwachsen. Durch den Verzicht auf grundlegende Software-Engineering-Aktivitäten schon zu Beginn, entstehen im späteren Verlauf der Weiterentwicklung oft Reibungsverluste.

Zum Schutz der Investitionen des DLR ist aus diesen Gründen das Software-Engineering-Netzwerk etabliert worden.

## 2 Das Software-Engineering-Netzwerk

Das Software-Engineering-Netzwerk besitzt die Aufgabe den DLR-weiten Austausch im Bereich des Software-Engineerings zu fördern sowie die Koordination der Nutzung bzw. Einführung gängiger Methoden, Prozesse und Werkzeuge zu übernehmen. Diese Zielstellung soll zum Einen durch die Etablierung von direkten Ansprechpartnern in den jeweiligen Instituten erreicht werden. Zum Anderen werden ergänzend zentral kostenfreie Werkzeuge bzw. Services zur Unterstützung der DLR-Software-Entwickler angeboten. Zu diesen Services gehören die beiden Webanwendungen Software-Katalog und Software-Standards sowie die DLR-Werkzeugsuite.

Beide Webanwendungen sind für die DLR-Mitarbeiter über die Informationsseite des Netzwerks im Unternehmens-Intranet erreichbar. Der Software-Katalog bietet die Möglichkeit Software-Entwicklungen nach einem umfangreichen Klassifikationsschema zu katalogisieren. Dadurch kann der Überblick über die verschiedenen Projekte eines Instituts gewahrt werden und im übergreifenden Kontext des DLR soll somit der Erfahrungsaustausch bis hin zur eventuellen Wiederverwendung von erfolgreich erprobten Konzepten gefördert werden. Die Anwendung Software-Standards ist darauf ausgerichtet den Projektverantwortlicher bei der Auswahl geeigneter Software-Engineering-Maßnahmen für den gesamten Lebenszyklus der zu entwickelnden Software zu unterstützen. Diese Maßnahmen werden anhand der jeweiligen Projektspezifika auf das Projekt zugeschnitten und basieren auf den für das DLR festgelegten Mindestanforderungen für den Bereich des Software-Investitionsschutzes.

Da die Kosten zur Etablierung und für den Betrieb essentieller Software-Entwicklungswerkzeuge wie z.B. von Versionsverwaltungssystemen für die einzelnen Institute relativ hoch ausfallen, ist die Idee einer zentralen Sammlung von Softwarewerkzeugen - die DLR-Werkzeugsuite - entstanden und umgesetzt worden. In der momentanen Ausbaustufe umfasst die Werkzeugsuite die Open-Source-Werkzeuge Subversion und Mantis, wodurch der für Software-Entwicklungen essentielle Bereich des Konfigurationsmanagements unterstützt wird. Beide Services sind in die bestehende IT-Infrastruktur integriert und werden durch den DLR-IT-Provider betrieben. Ergänzend dazu sind durch das Netzwerk Schulungen zur methodischen Vorgehensweise bei der Nutzung dieser Werkzeuge ausgearbeitet und durchgeführt worden. Der Trend der Nutzung dieses Angebot ist als positiv zu bewerten und scheint insbesondere durch die gezielte Information über die Schulung weiter anzuhalten.

## 3 Ausblick

Insbesondere im Bereich der DLR-Werkzeugsuite sollen die Aktivitäten weiter ausgebaut werden. Dazu zählt die Evaluierung und Bereitstellung weiterer Werkzeuge wie beispielsweise für den Bereich des Anforderungsmanagements. Zudem steht auch die Vernetzung beider Werkzeuge sowie die Integration in die Projektseiten des Unternehmens-Intranets im Vordergrund.

# 3. Projekt- und Anforderungsmanagement mit Scrum

Sessionüberblick

---

3.1. Agile-Scrum Planning . . . . .	75
3.2. Scrum mit individuellen Erweiterungen – ein Erfahrungsbericht . . . . .	85
3.3. Integration von Usability Engineering in den agilen Entwicklungsprozess nach Scrum	101

---





## Agile Scrum Planning

Ross English

Can Do GmbH, Kapuzinerstr 9  
80337 München  
r.english@candoprojects.com

**Abstract.** The purpose of this presentation is to describe the differences between the traditional “Waterfall” and Agile-Scrum methodologies with regard to the development of new software products. The examples are based on over 15 years of “real-world” software development experience by the author, as well as, an overview of what worked and what didn’t work.

### 1 Waterfall Cycle Methodology

The so called Waterfall methodology has been advanced and used by most organizations since the mid 80’s. It is rich in a priori analysis and documentation. It usually entails very distinct levels of separation of responsibilities between the analysts, programmers, testers and customer/end-users. This is not necessarily always the case, especially with very small organizations; however, the processes; customer requirements’, analysis, programming, testing and release usually follow a very sequential pattern.

The main purpose of the waterfall method is to gather as much knowledge upfront about “what” the product is to be and which risks might constrain its over-all development. One major criticism of these processes is that they usually consume a relative large amount of time with regards to the total projected duration of the product development plan. Usually these processes neither increase total product knowledge nor identify enough substantial risks proportional to the amount of time invested. The major problems that often surface only during implementation of the actual programming phase are impossible to document in the analysis phase. Another more worry some problem is that the “what”, the product to be built, is not the same “what” that the end users really want, or as it is sometimes referred to as the “Big Bang” at the end of the production cycle. After performing all the detailed analysis of user requirements and satisfying all identified risks, the one risk that was not removed was the risk of user or even market (in the case of a new software product), rejection. Perhaps so much time had elapsed since the initial requirements were taken that they were no longer valid when the product was released. This so called “Big Bang” at the end of the production cycle is a very real concern, which cannot be easily mitigated because of the inherent nature of the waterfall methodology. There have been many suggestions on how to improve the process, however, the core problems with the waterfall methodology, decentralized processes, lack of effective communication, etc., are difficult to over-come in larger organizations.

It must be noted that many criticisms of this methodology focus on the organizational intra-departmental rigidity of the implementation. With a small team, comprised of analysts, designers and programmers who all have adequate communication between the members and a common product overview it is possible to successfully develop a software product following the waterfall methodology.

## 2 Agile-Scrum Methodologies

Although the use of Agile methodologies for software development has been advocated for almost 10 years the basic lack of understanding and skepticism of the tenets of Agile by many programmers and managers has hampered their implementation in larger organizations. The goal of Agile is to break down the organizational impediments that constrain self-managing teams. This is anathema to the “command & control” mentality in many large organizations. This is one reason why organizations with a dedicated, competent IT department find it impossible to develop software that is required within the organization. Instead they are forced to rely on smaller, “leaner” third party companies for their software needs.

In the waterfall method, knowledge about the product and the accounting of the risks are determined (where possible) upfront at the beginning of the product development. Agile confronts the problem of product knowledge acquisition and risk avoidance by limiting the development to more or less strict time iterations or time boxes, where the goal at the end of each time box is a workable piece of software. In this way product knowledge and associated risks are dealt with over a smaller scope of the total product. This has the benefit of avoiding unnecessary analysis, which may not have been required and focusing directly on the immediate knowledge and risks associated with the development at hand.

The new comer in the Agile arsenal of productivity tools is Scrum. Scrum encourages the creation of self-managing teams, whose members should be co-located. Open verbal communication between all team members and disciplines that are involved in the product development is a major tenant of Scrum.

One of the principles of Agile-Scrum is the emphasis on the “How” and not the “What” in product development or in other words the planning process and not the plan. Another major tenant of Agile-Scrum is how it deals with the inevitable change in product requirements. Scrum embraces change as a natural part of the planning process. This is in complete contrast to the traditional waterfall methodology, which abhors change after the initial analysis of requirements.



## Agile-Scrum vs. Waterfall-Cycle Methodology



Speech by Ross English

© Can Do GmbH, Munich

### Agenda



Waterfall Cycle

Agile - Scrum

Terminology

Risk Management Overview

Advantages & Disadvantages

© Can Do GmbH, Munich

## What is the „Waterfall“ methodology?



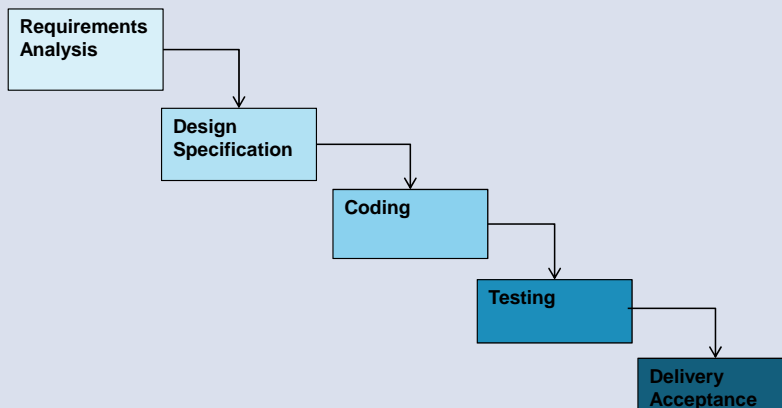
- Does the Waterfall methodology really exist?
- Barry Boehm – „Spiral Model of Software Development“
  - Probably makes first reference early 90's
  - Early advocate of Agile
- Waterfall Model – is a term representing any software development process, which:
  - is Sequential
  - Requires substantial up-front formal requirements analysis
  - Flows unabated through the phases:
    - Requirements Analysis
    - Design
    - Coding
    - Testing
    - Delivery
  - Progress is seen as flowing steadily downwards (like a waterfall)

© Can Do GmbH, Munich

## Waterfall Development Cycle

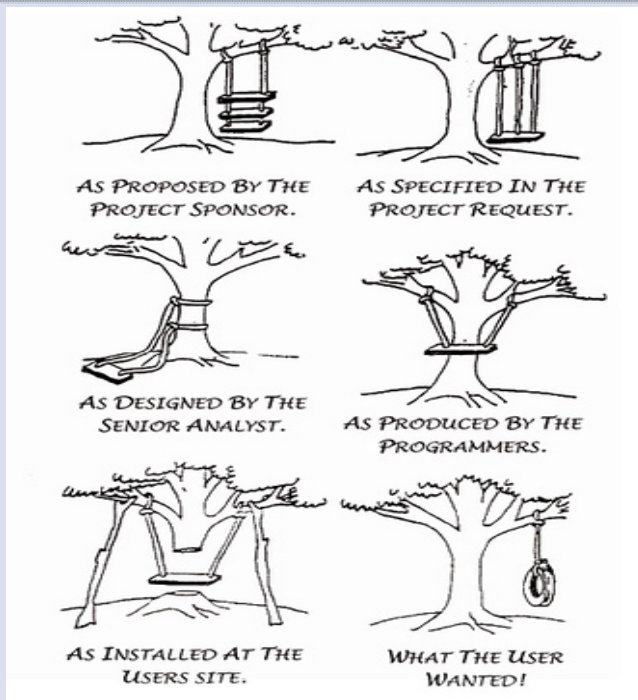


- Up-front analysis and design
- Attempts to define the whole project from the beginning
- Structured approach
- Emphasis on documentation
- Decentralization of software development



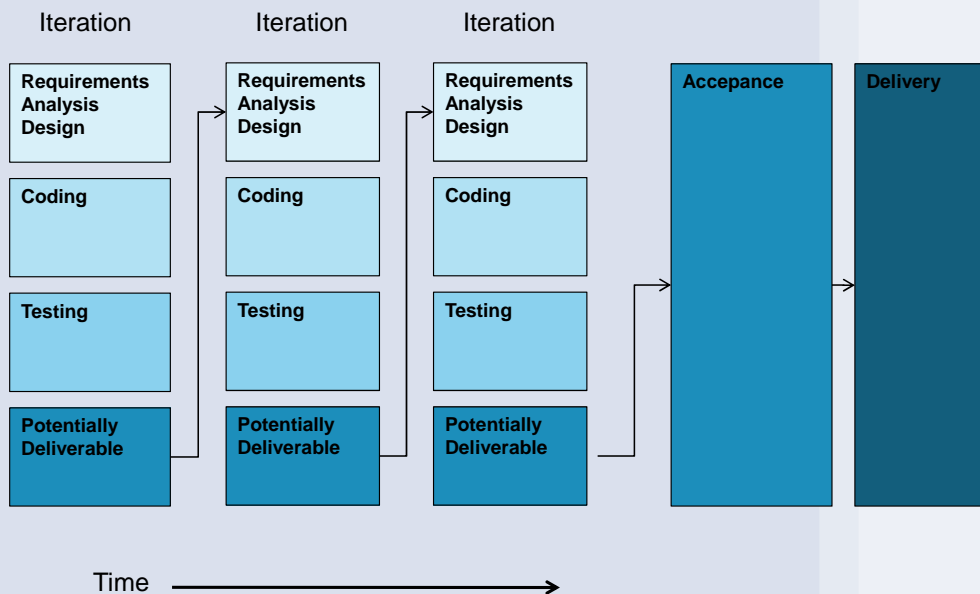
© Can Do GmbH, Munich

## What the User *Really* wanted



© Can Do GmbH, Munich

## Iterative Approach

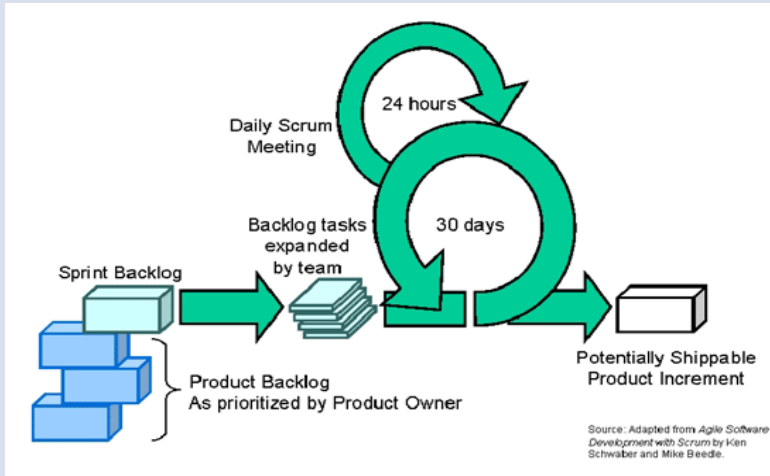


© Can Do GmbH, Munich

## Scrum process view



- Self-organizing, motivated teams
- Customer interaction and involvement
- Enhanced customer satisfaction
- Working software is the principal measure of progress
- Frequent deliveries



© Can Do GmbH, Munich

## Project Management Terminology



Traditional	Scrum
Project	Product
Phase	Sprint
Task	Sprint Feature
Milestone	Sprint review
Resource	Team Member
Requirements	Backlog, User Stories
Time Estimates (ideal Man hrs, days, weeks, months)	Story Points

## Risk Management - Waterfall



cando

### Traditional „Waterfall“

- Always tries to remove uncertainty about „What“ is going to be built
- Then tackles the uncertainty of „How“ it will be built
- Product first fully defined and analyzed before the „means“ (How) are considered

© Can Do GmbH, Munich

## Risk Management - Agile



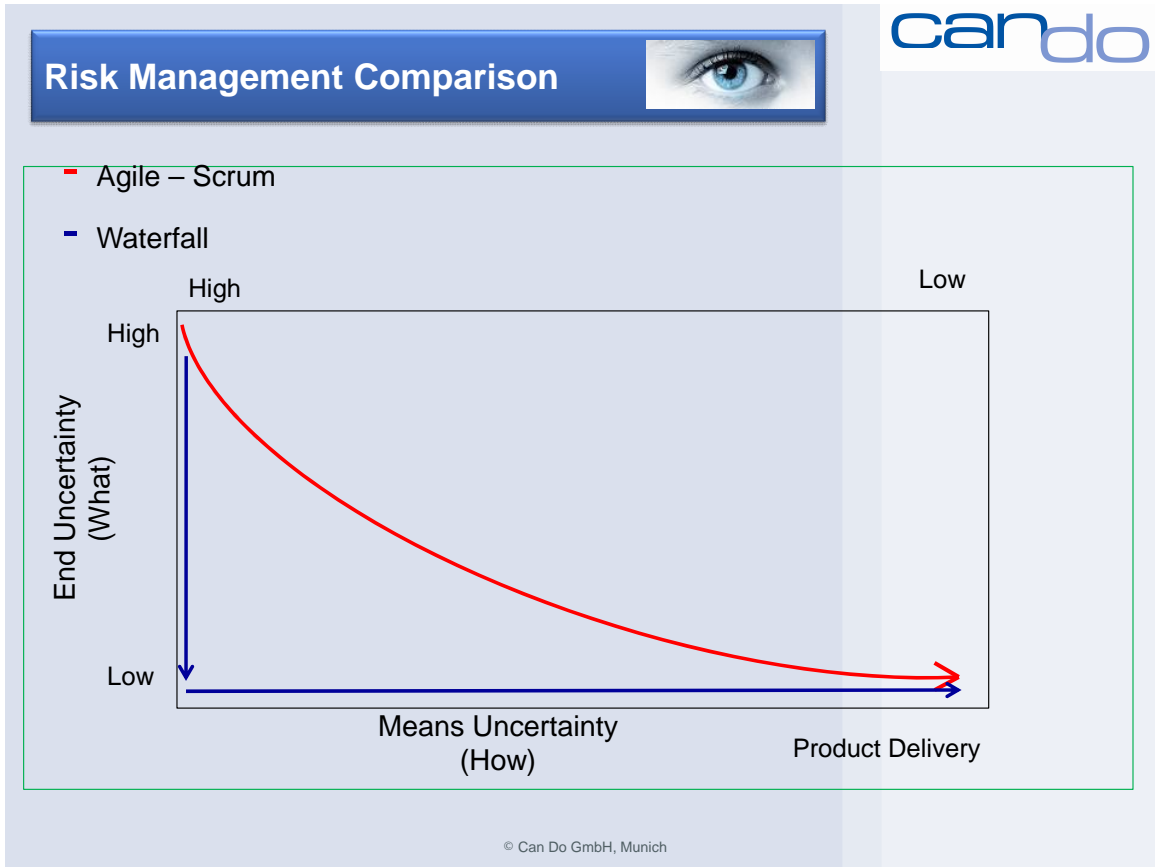
cando

### Agile - Scrum


- It is not possible to eliminate all uncertainty about what a product will be at the beginning
- Present the customer with smaller workable parts of the product
- Re-evaluate, plans adjusted by customer feedback
- Reduces the risk of building the wrong product

© Can Do GmbH, Munich





## Advantages / Disadvantages



	Traditional (waterfall) Management	Agile - Scrum Methodology
<b>Prioritization</b>	<ul style="list-style-type: none"> <li>✓ Support for long-term plans</li> </ul>	<ul style="list-style-type: none"> <li>✓ Few delays on important features</li> <li>✓ Prioritization of features</li> </ul>
<b>Planning</b>	<ul style="list-style-type: none"> <li>✓ Tries to define the overall project from the beginning</li> <li>✗ Might be headed toward wrong direction</li> </ul>	<ul style="list-style-type: none"> <li>✓ Micro Planning</li> <li>✓ Avoids Requirement cramming</li> </ul>
<b>Communication</b>	<ul style="list-style-type: none"> <li>✗ Communication between customer and team members not encouraged</li> </ul>	<ul style="list-style-type: none"> <li>✓ Good internal communication</li> <li>✓ Customer Involvement</li> </ul>
<b>Documentation</b>	<ul style="list-style-type: none"> <li>✓ Highly emphasized</li> <li>✓ Required at every phase</li> </ul>	<ul style="list-style-type: none"> <li>✗ Knowledge is stored mentally</li> <li>✗ Code is the documentation</li> </ul>
<b>Risk Management</b>	<ul style="list-style-type: none"> <li>✗ Might not find bugs until much later phase</li> </ul>	<ul style="list-style-type: none"> <li>✓ Continuous testing and review</li> <li>✓ Issues found more rapidly</li> </ul>

© Can Do GmbH, Munich

## References



cando

- "Agile Manifesto." 19 Apr 2007, 10:26 UTC.  
<http://www.agilemanifesto.com>
- „Agile Estimating and Planning“ Mike Cohn
- „Scrum and XP from the Trenches“, Henrik Kniberg
- „The New Methodology“, Martin Fowler
- Waterfall model, From Wikipedia,  
[http://en.wikipedia.org/wiki/Waterfall\\_model](http://en.wikipedia.org/wiki/Waterfall_model)

© Can Do GmbH, Munich

## Contact



cando

Can Do GmbH

Tel: +49 89 512 65 100

Fax: +49 89 512 65 500

E-Mail: [cando@candoprojects.com](mailto:cando@candoprojects.com)

Web: [www.candoprojects.de](http://www.candoprojects.de)

Ross English

[r.english@candoprojects.com](mailto:r.english@candoprojects.com)

© Can Do GmbH, Munich

### 3.1 Agile-Scrum Planning

## Extended Abstract

# Scrum mit individuellen Erweiterungen: Erfahrungen bei Projektron

Roland Petrasch<sup>1</sup>, Maik Dorl<sup>2</sup>, Florian Fieber<sup>3</sup>

<sup>1</sup> FB VI Informatik und Medien  
Beuth Hochschule für Technik Berlin  
Luxemburger Str. 10  
13353 Berlin  
petrasch@beuth-hochschule.de

<sup>2</sup> Projektron GmbH  
Charlottenstr.  
Berlin  
maik.dorl@projektron.de

<sup>3</sup> qme Software GmbH  
Gustav-Meyer-Allee 25  
13355 Berlin  
florian.fieber@qme-software.de

### Abstract:

Als weit verbreitetes agiles Verfahren findet Scrum [Sc04] breite Anwendung in der Praxis. Allerdings ist die Nutzung „out-of-the-Box“ oftmals nicht möglich, da individuelle Anforderungen an das Vorgehen existieren. Besonders bei der Produktentwicklung trifft dies zu, wo Produkt- und Projektmanagement ineinander greifen. Dieser Vortrag zeigt anhand einiger ausgewählter Aspekte, wie Scrum beim Software-Hersteller Projektron eingeführt und erweitert wurde. Im Folgenden seien die an der Scrum-Einführung beteiligten Partner kurz vorgestellt:

- Projektron GmbH: Sie ist eine Berliner Projektfirma, die seit 2001 das Produkt Projektron BCS entwickelt und bisher europaweit an über 300 Kunden vertrieben hat. U.a. durch die Expansion entstand der Wunsch, Scrum in der Software-Entwicklung einzuführen, um das eigene agile Vorgehen zu professionalisieren und skalierbarer zu gestalten. Der Geschäftsführer Maik Dorl begleitete den gesamten Prozess der Scrum-Einführung.
- Beuth Hochschule für Technik Berlin: An der BHT in Berlin-Mitte studieren über 9600 Studenten in 72 Studiengängen. Praxisorientierte Lehre und angewandte Forschung stehen im Vordergrund. Prof. Dr. Roland Petrasch (Fachbereich VI - Informatik und Medien) beschäftigt sich u.a. mit agilem Projektma-

nagement und Software-Engineering. Die Kooperation mit der Projektron GmbH führte in den letzten Jahren zu zahlreichen gemeinsamen Aktivitäten, z.B. Betreuung von Studenten, Workshops und Schulungsmaßnahmen.

- qme Software GmbH: Das unabhängige Berliner IT-Beratungsunternehmen unterstützt die Kunden vor Ort in den Bereichen Projekt- und Qualitätsmanagement, Software-Engineering, Anforderungsmanagement, Usability-Engineering und Software-Test. Bewährte Techniken und innovative Ansätze kommen dabei so zum Einsatz, dass optimale und individuell zugeschnittene Konzepte und Lösungen für die Kunden entstehen.

Die schrittweise Einführung von Scrum bei der Projektron GmbH begann 2008 mit Untersuchungen und Evaluierungen agiler Verfahren. In diesem Rahmen wurde u.a. analysiert, welche Anforderungen das Projektmanagement-System Projektron BCS erfüllen muss, um Scrum zu unterstützen. Dies führte zur Entwicklung entsprechender Funktionen, z.B. für die Verwaltung von Epics, User Stories, Requirements und Test Cases. Auch die Sprint-Rollen und das Reporting, z.B. Burn Down Chart, wurden umgesetzt, so dass 2009 schließlich die erste Version zur Verfügung stand. Parallel dazu begann das Entwicklungsteam bei Projektron nach Scrum zu arbeiten.

Neben den bewährten Techniken von Scrum wie z.B. Release- und Sprint-Planung, Daily Scrum, Scrum Review und Retrospektive, werden mittlerweile mit weniger bekannten oder abgewandelten Methoden kombiniert. So dient beispielsweise der „Estimation Monday“ dazu, neue Anforderungen bzw. den Aufwand für Fehlerkorrekturen zu schätzen, die dann teilweise noch während des laufenden Sprints umgesetzt werden. Ein Puffer von 15%, der bei der Sprint-Planung dafür vorgesehen ist, macht diesen Mechanismus möglich. Die Release-Planung lässt dadurch stets aktualisieren und die Aufwandsschätzung für die Product Backlog Items überprüfen, korrigieren und ergänzen. Wie der Name vermuten lässt, findet dieses Schätz-Meeting immer Montags statt, weil u.a. die neuen oder geänderte Anforderungen in Form von Tickets der Vorwoche ausgewertet und in der Schätzung berücksichtigt werden (Stichwort Ticket Tracking).

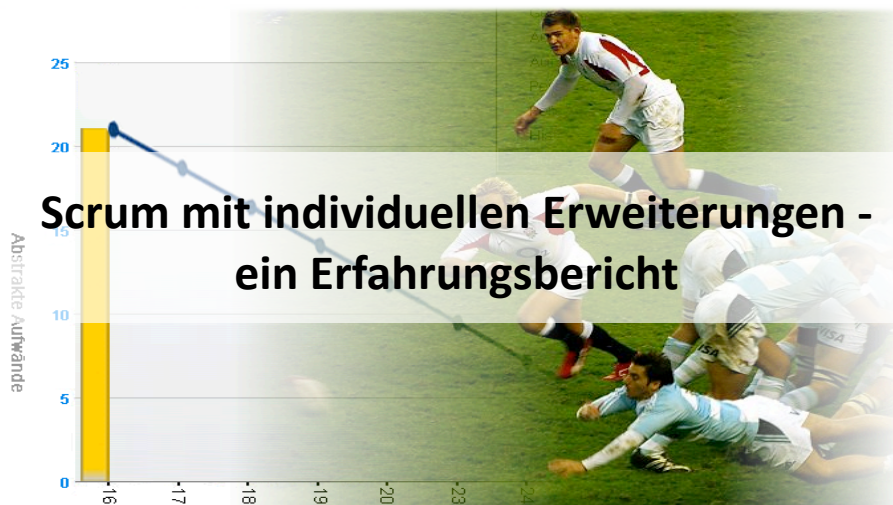
So kann das Produktmanagement schnell und realistisch auf Anforderungen der Kunden reagieren und mit der eigenen Produktstrategie abgleichen. In diesem Zusammenhang ist der Chief Product Owner (CPO) quasi der Produktmanager mit dem Gesamtüberblick, der eng mit den verschiedenen Domain Product Owners (DPO) zusammenarbeitet. Eine zweistufige Sprint-Planung hilft bei der Vorbereitung: Die erste Sprint-Planung umfasst u.a. die Auswahl und Priorisierung der Anforderungen durch den CPO (Schätzungen der Aufwände liegen bereits durch den Estimation Monday vor). Wenn dann das Sprint-Ziel definiert ist, wird in einer 2. Planungssitzung der Sprint-Backlog erstellt. Die DPO stehen dabei für Fragen zur Verfügung. Auch die Aufwände sind in diesem Schritt an die Gegebenheiten neu abzuschätzen, z.B. beim Einsatz neuer Mitarbeiter.

Eine weitere Besonderheit ist der Support-Buffer für die Mitarbeiter: Damit auch das Scrum-Team die Nähe zum Kunden behält und damit ein tiefes Verständnis für die Anwendungsdomänen bekommen bzw. aufrecht erhalten kann, leisten Entwicklungsmitarbeiter an einem bestimmten Tag oder bei besonderen Anlässen direkte Unterstützung

und kommunizieren mit den Kunden. Der Begriff Support ist dabei nicht negativ, sondern positiv besetzt, da der Nutzen für das Team direkt sichtbar wird. Die Schätzungen und die Umsetzung der Anforderungen ist viel einfacher möglich, wenn alle Beteiligten ein Verständnis für die Sachverhalte haben – in diesem Fall für das Projektmanagement. Ein Support-Puffer wird also zusätzlich bei der Planung und der Ermittlung der Velocity eingeplant.

Die zunehmende Bedeutung agiler Prozesse für Software-Entwicklungsvorhaben macht sich auch bei dem erhöhten Bedarf nach einer adäquaten Tool-Unterstützung bemerkbar. Gerade bei umfangreichen Vorhaben ist neben der Qualifikation und dem richtigen Erfahrungsmix im Team ein passender Werkzeugeinsatz von entscheidender Bedeutung. Ab einer bestimmten Menge an Projekten, Anforderungen und Mitarbeitern wird ein Projektmanager bzw. ein Scrum-Master den Fertigstellungsgrad als Kostenmanagementplan oder ein Burn-Down-Chart (Restaufwand) kaum manuell erstellen können oder wollen. Dabei muss sich ein Projektmanagement-Tool quasi „minimal-invasiv“ in die durch die Beteiligten geprägten agilen Projekte und Methoden einbringen. Ein simples Überstülpen eines klassischen Projektmanagementansatzes über ein agiles Vorhaben wäre wenig zielführend. In diesem Fall wurde das selbst entwickelte Tool so weiterentwickelt, dass es sich für den Einsatz in Scrum-Projekten eignet, was für die Akzeptanz des Teams von entscheidender Bedeutung war.

Das Scrum-Einführungsprojekt bei Projektron wurde wissenschaftlich von Prof. Petrasch von der Beuth Hochschule und praktisch vom Beratungsunternehmen qme Software GmbH begleitet.



Roland Petrasch  
Beuth Hochschule für Technik  
prof.beuth-hochschule.de/petrasch



Florian Fieber  
qme Software GmbH  
www.qme-software.de



Maik Dorl  
Projektron GmbH  
www.projektron.de



## Agenda

- Produkt- und Projektmanagement bei Projektron
- Einführung von Scrum & individuelle Erweiterung
- Scrum4PD → Diskussion



PM & RM with Scrum:  
The Magic of Velocity

## Projektpartner

- Projektron GmbH: Produkthersteller
  - Webbasierte Projektmanagement-Software *Projektron BCS*:  
Angebotserstellung, Ressourcenmanagement, Zeiterfassung etc.
  - Seit 2001, Zentrale in Berlin, 8 Geschäftsstellen, > 60 Mitarbeiter
  - Über 300 Kunden in Europa, z. B. EADS, HanseMerkur, Charité
- Beuth Hochschule für Technik Berlin
  - > 9600 Studenten, 72 Studiengänge, Bachelor/Master
  - FB VI: Informatik und Medien, Studiengänge u. a. MI, DM, TI/ES
  - Praxisnahe Lehre und Forschung, z. B. ExzellenzTandem, FA
- qme Software GmbH: Beratungsunternehmen
  - Beratung und Training: Projekt- und Qualitätsmanagement, Software-Engineering, Usability Engineering, Anforderungsmanagement
  - Angewandte Forschung & Entwicklung, z. B. Model Based User Interface Design (MBUI)
  - Kunden, z. B. European Energy Exchange, T-Mobile, Projektron

## Produkt- und Projektmanagement bei Projektron

- Projektron als Produktfirma mit ergänzenden Dienstleistungen
- Entwicklung auf Basis von Kundenanforderungen und Vorgaben des Produktmanagements (GF)
- Anforderungsmanagement mit dem integrierten Ticket-System
- Iteratives Vorgehen: „Sammeln“ von Anforderungen,  
→ Aufsetzen von „Entwicklungsprojekten“
- Release-Zyklen typischerweise 3-6 Monate
- Projektmanagement bisher
  - vorwiegend Kunden-/Ticket-getrieben, kurze Iterationen
  - Projektleitung: GF + Entwicklungsleiter
  - wenig formal, ansatzweise agil (aber kein Scrum)



## Produkt- und Projektmanagement bei Projektron

- Produkt „Projektron BCS“
  - wenig Varianten/Branches, Konfiguration durch Kunden
  - stabile Software-Architektur, hohes Qualitätsniveau der Basis-SW
  - bewährte Entwicklungsinfrastruktur, teilweise automatische QS, Continuous Integration, Deployment, Betrieb etc.
- Agiles Vorgehen: Rahmenbedingungen bei Projektron
  - Team: 10 Entwickler, starke Identifizierung mit dem Produkt
  - Hohe Kundenbindung: Schulung, Support, Weiterentwicklung
  - Management unterstützt Innovationen und Veränderungen aktiv
  - Kompetenz: Entwickler kennen und akzeptieren Agilität

5

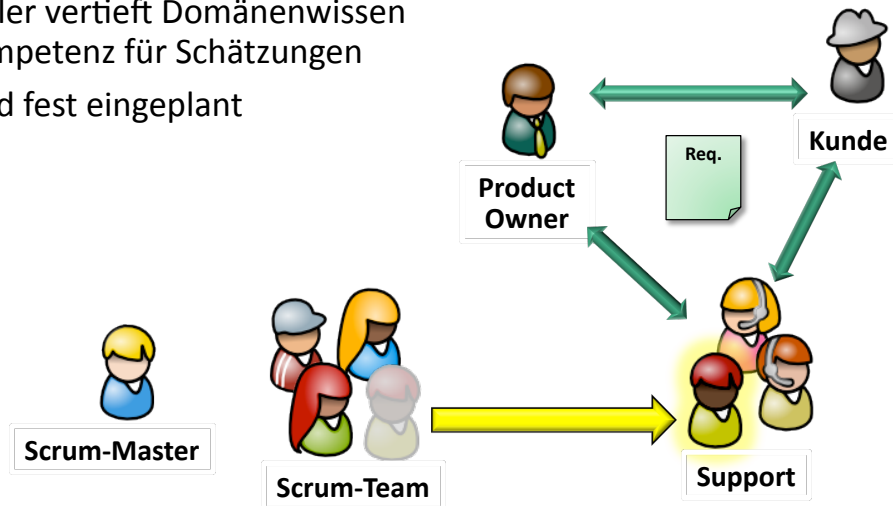
## Einführung von Scrum

- Vorbereitung und schrittweises Vorgehen
- Diplomarbeit (BHT-Student) in 2008, Thema: Tool-gestütztes Scrum mit Projektron BCS → neue Anforderungen
- Workshops, Weiterbildung der Mitarbeiter
- Entwicklung des Scrum-Moduls für Projektron BCS
- Umstellung auf Scrum in 2009
- Erfahrungen → Erweiterungen / Anpassungen:
  - Scrum Master, Scrum-Team (9 Entwickler): Support-Tag
  - Chief Product Owner (CPO) + Domain Product Owner (DPO)
  - Requirements: 15 % Puffer + Estimation Monday
  - Release-Zyklus mit 2 Sprint-Planungen

6

## Scrum-Erweiterungen: Support-Tag

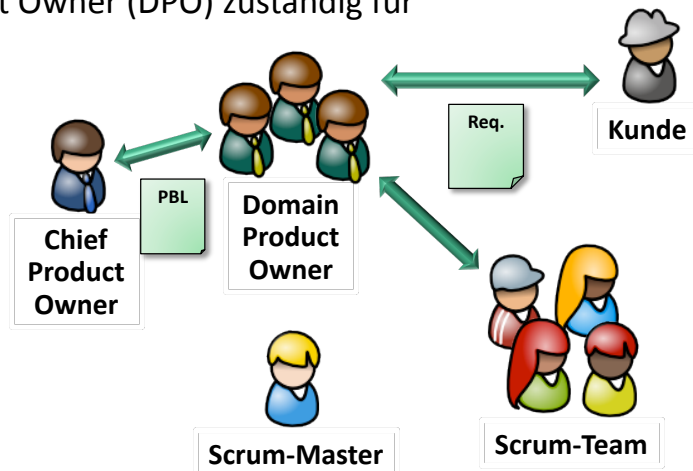
- Scrum-Team: Jeder Entwickler arbeitet im Support (Dauer: 1 Tag)
- Entwickler vertieft Domänenwissen und Kompetenz für Schätzungen
- Aufwand fest eingeplant



7

## Scrum-Erweiterungen: Chief Product Owner

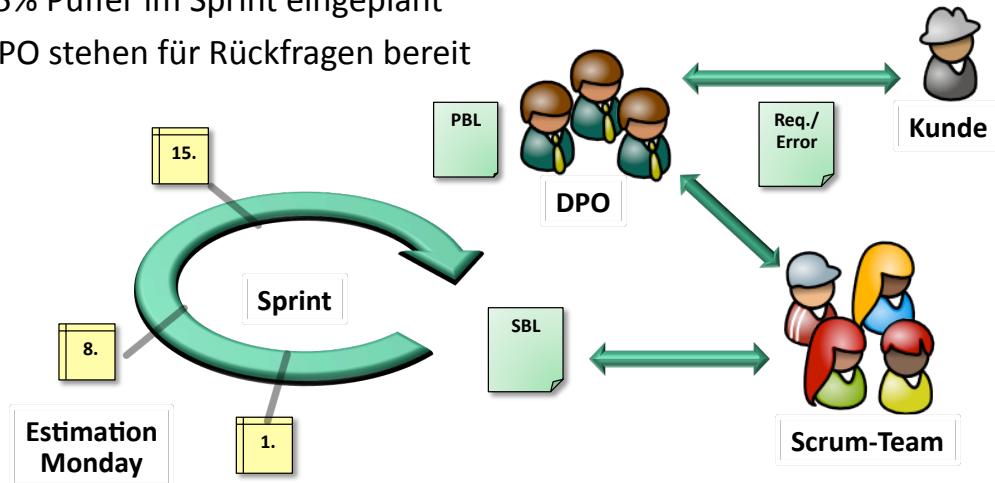
- Chief Product Owner (CPO): Geschäftsführung hat Gesamtüberblick und priorisiert die Anforderungen
- ca. 10 Domain Product Owner (DPO) zuständig für bestimmte Features



8

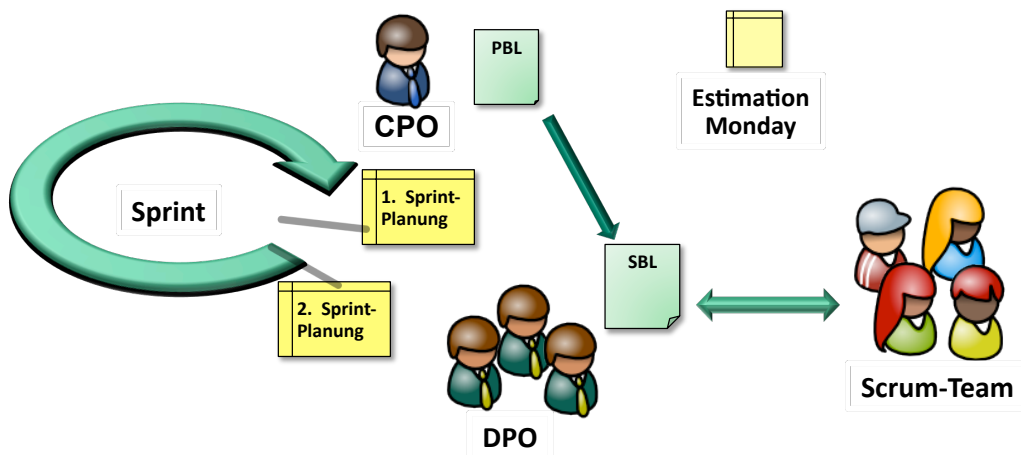
## Scrum-Erweiterungen: Estimation Monday

- Montag: Schätzung neuer Anforderungen auf der Basis von Tickets, Fehlermeldungen und Vorgaben des CPO
- 15% Puffer im Sprint eingeplant
- DPO stehen für Rückfragen bereit



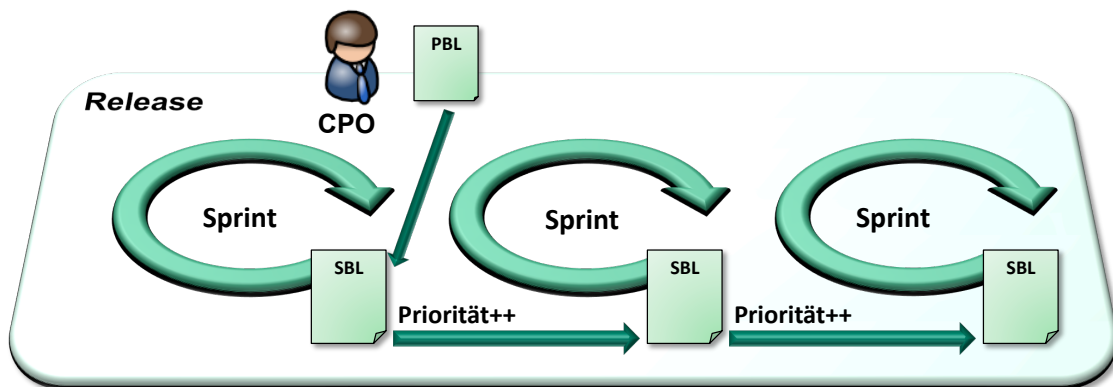
## Scrum-Erweiterungen: 2 Sprint-Planungen

- 1. Sprint-Planung: Auswahl der User Storys durch CPO (PBL), Priorisierung, Schätzung (Estimation Monday)
- 2. Sprint-Planung: Detailplanung, erneute Schätzung, DPO stehen zur Verfügung



## Scrum-Erweiterungen: Release-Zyklus

- Release: 3-5 Sprints (ca. 3-6 Monate)
- Neben Kundenanforderungen auch Vorgaben des CPO
- Deckelung des Aufwandes zu Beginn, zunehmende Priorität



11

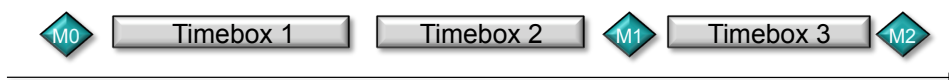
## Zur Erinnerung: Timebox und Meilenstein

- Prozessqualität bei agilen Projekten
  - Iteration (Sprint) mit Timebox und Meilenstein: Planung



Max. Prozessqualität: Ergebnisse (Artefakte) liegen pünktlich vor

- Iteration (Sprint) mit Timebox und Meilenstein: Diskrepanz



Prozessqualitätsmangel: Ergebnisse (Artefakte) liegen zu spät vor  
(fehlende Features werden in die nächste Timebox übernommen)

### 3.2 Scrum mit individuellen Erweiterungen – ein Erfahrungsbericht

Sie sind **Roland Petrasch** | Abmelden

Mein BCS | Zeiterfassung | **Projekte** | Intern | Extern | Administration | Links

Überblick | Kommunikation | Termine

**Nachrichtenüberblick** Standard | Anpassen ...

**Nächste Termine**

Datum	Startzeit	Dauer	Betreff	
Heute	09:00h	00:15h	Daily Scrum	(OK)
Morgen	09:00h	00:15h	Daily Scrum	(OK)

**Neueste Notizen**

Vom	Betreff
Mi 07.10.09	Anforderungsworkshop: Checkliste erstellen

**Aktuelle Wiedervorlagen**

Zum	Betreff	Bezug	
Mi 07.10.09	Abstimmung Produktmanagement Mautsystem	Petrasch, Roland	(OK)

**Aufgabenliste**

Projekt	Aufgabe	Prio	Aufgabenende
ScrumSuperIM_Petrasch	Sprint 2 (26 SP)	3 (Niedrig)	Do 26.11.09

**Tickets**

Ticket-Id	Betreff	Status	Bearbeiter	Prio int	Eingang
#28	Bilder werden immer über die Seiten	6-Aufgenommen	Programmierer8, Mark	Dringend	So 13.09.09

Sie sind **Roland Petrasch** | Abmelden

Mein BCS | Zeiterfassung | **Projekte** | Intern | Extern | Administration | Links

Projekt

Überblick | Suche | Planung | Durchführung | Erfassung | Auswertungen | Berichte | Rechnungen | Werkzeuge

**Projektbaum** Voreinstellung | Anpassen ...

Name	Status	Prio	Start
Projekte			
+ Angebote	Angeboten	3 (Niedrig)	Mo 14.09.09
+ Archiv	Geschlossen		
+ Vorlagen			Mi 27.01.10
+ Intern	Offen		
+ V-Modell	Geplant	1 (Hoch)	Mo 12.02.07
+ AutomotiveSPICE (ISO/IEC 15504)	Geplant	3 (Niedrig)	Di 15.09.09
+ Website-Projekte	Offen		
+ English Projects	Offen		
+ Kundenprojekte	Offen		
+ SuperIM12	Geplant	3 (Niedrig)	Sa 26.09.09

### 3. Projekt- und Anforderungsmanagement mit Scrum

projektron BCS 6 Sie sind **Roland Petrasch** | Abmelden

← → Lesezeichen Hilfe Konfiguration

**Mein BCS** | **Zeiterfassung** | **Projekte** | **Intern** | **Extern** | **Administration** | **Links**

Adressen Einfügen

Überblick Suche Akquisen Angebote Rundschreiben

Adressbaum Personensuche

Adressbaum Voreinstellung Anpassen ...

Name	Land	PLZ	Stadt	Telefon	E-Mail
Adressen					
+ Dienstleister					
- Interessenten					
+ Emimesa GmbH & Co.KG	Deutschland	70161	Stuttgart	+49 711 24587870	info@emimesa.local
+ Heimann Onlineshop GbR	Deutschland	10629	Berlin	+49 30 2623857	info@heimann.local
- Kunden					
+ Deutsches Institut für Wirtschaftsforschung	Deutschland	14191	Berlin	+49 30 7777777	info@diw.local
+ Kyrabit GmbH	Deutschland	12105	Berlin	+49 30 240020	info@kyrabit.local
+ Lykoranet AG	Deutschland	81545	München	+49 89 55123005	info@lykoranet.local

Druckansicht | CSV/Ansicht

projektron BCS 6 Sie sind **Roland Petrasch** | Abmelden

← → Lesezeichen Hilfe Konfiguration

**Mein BCS** | **Zeiterfassung** | **Projekte** | **Intern** | **Extern** | **Administration** | **Links**

Adressen Einfügen

Überblick Suche Verwaltung Verträge

Firmenbaum Voreinstellung

Name	Telefon	E-Mail
Adressen		
+ Computerbuch Gr	030 5555555	computerbuch@computerbuch.local
+ Administration (	030 5555555	administration@computerbuch.local
+ Buchhaltung (F	030 5555540	buchhaltung@computerbuch.local
+ Design (ZE 1)	030 5555555	design@computerbuch.local
+ Entwicklung (M	030 5555555	entwicklung@computerbuch.local
- Entwickler		entwicklung@computerbuch.local
- Entwickler1, Stefan	030 5555552	entwicklung@computerbuch.local
- Entwickler2, Eugen	0330 555777	entwicklung@computerbuch.local
- Entwickler3, Stefan	030 5555555	entwicklung@computerbuch.local
- Entwickler4, Torsten	030 5555555	entwicklung@computerbuch.local
- Entwickler5, Hendrik	030 5555555	entwicklung@computerbuch.local
- Entwickler6, Klaus	030 5555553	entwicluna@computerbuch.local

### 3.2 Scrum mit individuellen Erweiterungen – ein Erfahrungsbericht

The screenshot shows the 'projektron BCS 6' interface. The user is logged in as 'Roland Petrasch'. The main menu includes 'Mein BCS', 'Zeiterfassung', 'Projekte', 'Intern', 'Extern', 'Administration', and 'Links'. The current view is 'Projekte > English Projects > ScrumSuperIM\_Petrasch'. The 'Planung' menu is open, showing options like 'Product backlog', 'Strukturplan', 'Gantt-Diagramm', 'Meilensteine', 'Basispläne', 'Vorgänger/Nachfolger', 'Risiken', 'Vorlagen', 'Kunden', 'Angebote', 'Auftrag', 'Auftragsplan', 'Funktionsätze', 'Projektantrag', and 'Angebotsdaten'. The 'Projektüberblick' section shows project details: Start: Mo, Ende: Fr, Dauer: 45, Status: Off. The 'Kunde' section lists 'Deutsches Institut für Wirtschaftsforschung' with address 'Königin-Luise-Strasse 5, 14191 Berlin, Deutschland'. The 'Projektleiter Kde' section is empty. The 'Meilensteine' section shows 'Evaluation finished' on 'Fr 16.10.09' with a priority of '3 (Niedrig)'. The 'Bemerkung' section contains the text 'Project for the messenger system'.

The screenshot shows the 'Product backlog' view in the 'projektron BCS 6' interface. The table lists tasks with columns for 'Pos', 'Titel', 'Status', 'Prio', 'Aufwand [SP]', and 'Release'. A dropdown menu is open for the 'Status' of the task 'show result list', showing options: 'In Entstehung', 'Zu Prüfen', 'Akzeptiert', 'Abgelehnt', 'Zurückgestellt', 'Eingeplant', 'In Arbeit', and 'Umgesetzt'. The 'show result list' task is highlighted in yellow and has a status of 'In Entstehung' and an effort of 2 SP. Other tasks include 'SuperIM PBL' (Akzeptiert), 'Connect to contact' (Akzeptiert), 'search a contract' (Akzeptiert, 1 SP), 'enter search criteria' (In Entstehung, 1 SP), 'execute search function' (In Entstehung, 1 SP), 'select contact(s)' (In Entstehung, 1 SP), 'search no results', and 'Akzeptanztest 2'.

Pos	Titel	Status	Prio	Aufwand [SP]	Release
	SuperIM PBL	Akzeptiert			
	Connect to contact	Akzeptiert	Normal		
	search a contract	Akzeptiert	Niedrig	1	[Meilensteinname]
	enter search criteria	In Entstehung		1	
	execute search function	In Entstehung		1	
	show result list	In Entstehung		2	
	select contact(s)	In Entstehung		1	
	search no results				
	Akzeptanztest 2				



### 3. Projekt- und Anforderungsmanagement mit Scrum

projektron BCS6 Sie sind **Roland Petrasch** | Abmelden

Lesezeichen | Hilfe | Konfiguration

Mein BCS | Zeiterfassung | **Projekte** | Intern | Extern | Administration | Links

Projekte > English Projects > **ScrumSuperIM\_Petrasch** (Projekt, Offen) Einfügen | Aktionen

Eigenschaften | Planung | Durchführung | Erfassung | Auswertungen | Berichte | Rechnungen | Wer

Unterprojekt  
Arbeitspaket  
Meilenstein

**Product backlog** Voreinstellung

Titel

Schlagerwörter:

Beziehungen zu anderen Anforderungen

Voraussetzungen:

Quellen Tickets:

Bearbeiter:

Speichern | Übernehmen | Abbrechen

**Aufwandsplan** Voreinstellung | Anpassen ...

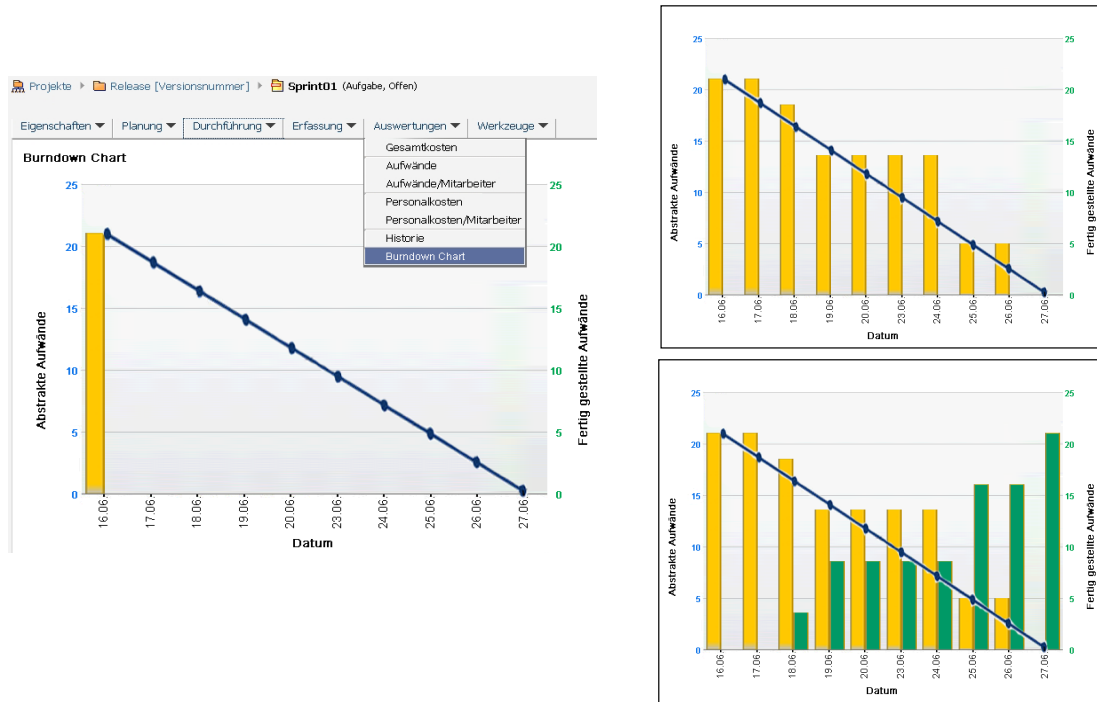
Übernehmen | Abbrechen | < Zurück | Weiter > | Fertig

Aufgabe	Stundensätze		Bottom-Up Plan		Wert
	StdSatz ext	StdSatz int	Plan		
Release [Versionsnummer]			30	t 0 h 0 m	9.600,00
Sprint01			30	t 0 h 0 m	9.600,00
Entwickler, Emil	0,00 €	40,00 €	7	t 4 h 0 m	2.400,00
Entwickler, Erik	0,00 €	40,00 €	7	t 4 h 0 m	2.400,00
Designer, Dennis	0,00 €	40,00 €	7	t 4 h 0 m	2.400,00
Architekt, Alexander	0,00 €	40,00 €	7	t 4 h 0 m	2.400,00
Sprint02			0	t 0 h 0 m	0,00
Sprint03			0	t 0 h 0 m	0,00
Product Backlog			0	t 0 h 0 m	0,00

**Strukturplan** Voreinstellung | Anpassen ...

Pos	ZNr	Name	Abstr. A	Status	Prio
1		Release [Versionsnummer]		Offen	1 (Hoch)
2		Sprint01		Offen	1 (Hoch)
3		Backlog Item01		Spezifiziert	1 (Hoch)
4		User story01	3	Neu	1 (Hoch)
5		Akzeptanztest01		Erstellt	2 (Normal)
6		User story02	5	Neu	1 (Hoch)
7		Backlog Item02		Spezifiziert	1 (Hoch)
8		User story03	5	Neu	1 (Hoch)
9		Akzeptanztest02		Erstellt	2 (Normal)
10		Akzeptanztest03		Erstellt	2 (Normal)
11		User story04	8	Neu	1 (Hoch)
12		Sprint02		Offen	2 (Normal)
13		Epic01	5	Neu	2 (Normal)
14		Epic02		Neu	3 (Niedrig)
15		Sprint03		Offen	3 (Niedrig)
16		Product Backlog		Offen	1 (Hoch)





## Fazit / Diskussion

- Vorbereitung, Akzeptanz, Management-Unterstützung
- Keine Dogmatisierung, individuelle Anpassungen zulassen
- KISS, weniger ist (manchmal) mehr
- Erfahrungsberichte: Rahmenbedingungen beachten, Vorsicht bei „absoluten Wahrheiten“ über Scrum
- Tool-Unterstützung: Integrierte RE-Funktionen, Einbindung der Kunden durch Ticket-System, Kommunikation



Danke für Ihre Aufmerksamkeit



### *3.2 Scrum mit individuellen Erweiterungen – ein Erfahrungsbericht*

## Integration von Usability Engineering in den agilen Entwicklungsprozess nach Scrum – ein Erfahrungsbericht

Ursula Meseberg

microTOOL GmbH  
Voltastraße 5  
13355 Berlin  
ursula.meseberg@microtool.de

**Abstract:** Usability Engineering (UE) hat sich zu einer eigenständigen Disziplin entwickelt. Die Softwareentwicklung profitiert bis heute wenig davon: Die Schnittstelle zwischen UE und Softwareentwicklung besteht bei den meisten Software produzierenden Unternehmen immer noch aus dem „Nachcodieren“ der UE-Ergebnisse, die in Form von Rastergrafiken und mehr oder weniger „klickbaren“ Prototypen übergeben werden. Microsoft verspricht mit Windows Presentation Foundation (WPF) und der Oberflächensprache XAML die Überwindung dieses Medienbruchs. Warum dann nicht gleich UE in den Softwareentwicklungsprozess integrieren? Der UE-Prozess, wie ihn z. B. die DIN EN ISO 13407 beschreibt, ist iterativ, anforderungsgetrieben und feedback-orientiert. Er erfüllt damit auf den ersten Blick die wichtigsten Voraussetzungen für eine Integration in den agilen Softwareentwicklungsprozess. Aber passen Geschwindigkeit, Iterationszyklen, Methoden, Rollen und Menschen wirklich zusammen? Es werden die Erfahrungen aufgezeigt, die microTOOL in einem vom BMBF geförderten Scrum-Projekt mit der Integration von UE gesammelt hat.

### 1 Der Begriff der Usability

Im Software Engineering (SE) ist *Usability* nur eines von mehreren Qualitätsmerkmalen – neben Functionality, Reliability, Performance und Supportability –, die Software aufweisen sollte. Obwohl allgemein akzeptiert ist, dass sich Qualitätsmerkmale – und damit auch Usability – nicht in Software „hineintesten“ lassen, sind konstruktive Vorgehensweisen, um Usability zu schaffen, im SE kaum verbreitet.

Außerhalb des SE hat Usability ein weit höheres Gewicht gewonnen. In den vergangenen 25 Jahren hat sich um den Begriff Usability eine ganze Disziplin mit eigenen Methoden, Techniken, Prozessen und Normen entwickelt: das *Usability Engineering* (UE).

Hier wird der Usability-Begriff nach ISO 9241-11 zugrunde gelegt. Danach ist Usability (oder Gebrauchstauglichkeit) „das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.“ Die *Effektivität* bezeichnet dabei die Genauigkeit und Vollständigkeit, mit der der Benutzer sein Ziel erreicht. Mit *Effizienz* ist der vom Benutzer erbrachte Aufwand im Verhältnis zur Genauigkeit und Vollständigkeit gemeint, mit der er sein Ziel erreicht. Das Kriterium der *Zufriedenheit* versucht, die subjektive Einstellung der Benutzer zum Produkt greifbar zu machen.

Für Softwarehersteller, die ein Produkt viele Jahre lang erfolgreich vermarkten und kontinuierlich weiterentwickeln, wird durch die wachsende Funktionalität die Effizienz ihrer Produkte nach und nach zum Problem. Je mehr Funktionen in eine bestehende Benutzeroberfläche integriert werden, um so schwieriger wird es für den Benutzer, eine gewünschte Funktion zu finden. Eine Zeit lang kann dieses Problem durch Individualisieren, das heißt durch benutzerspezifisches Konfigurieren der Oberfläche beherrscht werden. Das Beispiel von Microsoft Word 2003 in Abbildung 1a zeigt anschaulich, dass ein Redesign aber irgendwann unausweichlich wird.

### 3.3 Integration von Usability Engineering in den agilen Entwicklungsprozess nach Scrum

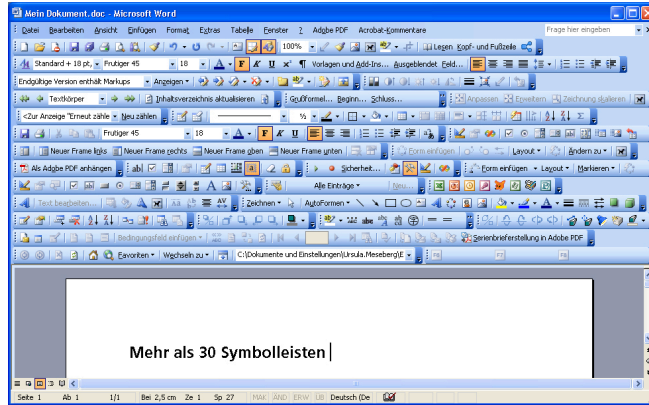


Abbildung 1a: Microsoft Word 2003: Öffnet man alle Symbolleisten, zeigt sich der Umfang der Funktionalität – und das Problem, seltener gebrauchte Funktionen schnell zu finden (Effizienz)

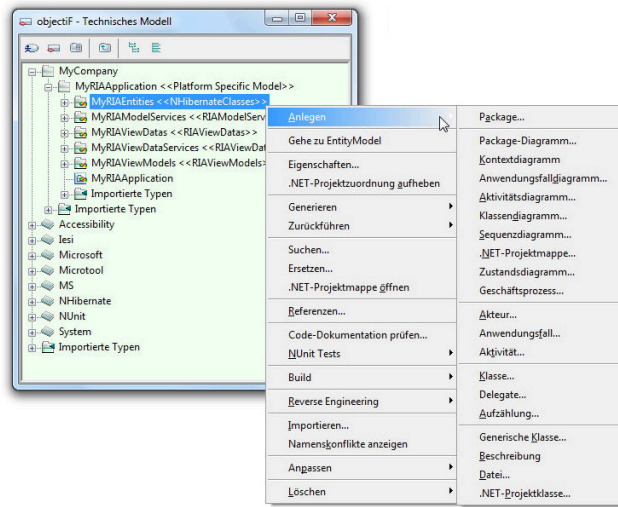


Abbildung 1b: microTOOL objectiF – eine andere Ausprägung des gleichen Effizienzproblems: Lange und geschachtelte Kontextmenüs machen es gelegentlichen Benutzern nicht leicht, gewünschte Funktionen zu finden.

microTOOL stand 2008 vor einer ähnlichen Herausforderung.

## 2 Das SPSF-Projekt

### 2.1 Der Projektkontext

Die nachfolgend beschriebenen Erfahrungen stammen aus einem Projekt, das microTOOL im Herbst 2008 gestartet hat und das Ende 2010 abgeschlossen wird. Gegenstand ist die Entwicklung einer Software-Produktlinie für modellgetriebene Software-Fabriken (SPSF). Das SPSF-Projekt wird vom Bundesministerium für Bildung und Forschung im Rahmen des Förderprogramms *KMU-innovativ: IKT* unterstützt.

Das Projekt hat folgenden Hintergrund: microTOOL bietet seit Jahren ein Werkzeug für die modellgetriebene Softwareentwicklung auf der Basis der UML an. Die Erfahrung hat gezeigt, dass Anwender an ein solches Werkzeug sehr individuelle Anforderungen stellen. microTOOL will durch die Entwicklung einer Software-Produktlinie im Rahmen des SPSF-Projekts in die Lage kommen, seinen Kunden zeitnah individuell ausgeprägte und dennoch kostengünstige Entwicklungswerkzeuge (sprich: Software-Fabriken) zur Verfügung zu stellen. Die Benutzeroberfläche des vorhandenen Werkzeugs (siehe Abbildung 1b) wird in der Produktlinie mit dem Ziel höhere Effizienz grundlegend grafisch und technisch überarbeitet.

## 2.2 Die Technologie und ihre Konsequenzen für den Workflow im Projekt

Die Produktlinie wird unter .NET entwickelt. Sie erhält die Softwarearchitektur einer Rich Internet Application mit WPF-Clients.

WPF (Windows Presentation Foundation) von Microsoft und die XML-basierten Oberflächensprache XAML, die zur Realisierung der Benutzeroberfläche eingesetzt werden, ermöglichen einen medienbruchfreien Workflow zwischen Usability Engineer und Softwareentwickler. Abbildung 2 zeigt den „neuen“ Workflow im Vergleich zur heute vielfach noch üblichen Form der Zusammenarbeit, die auf dem Nachcodieren von Oberflächenentwürfen basiert, die den Softwareentwicklern als Rastergrafiken übergeben werden.

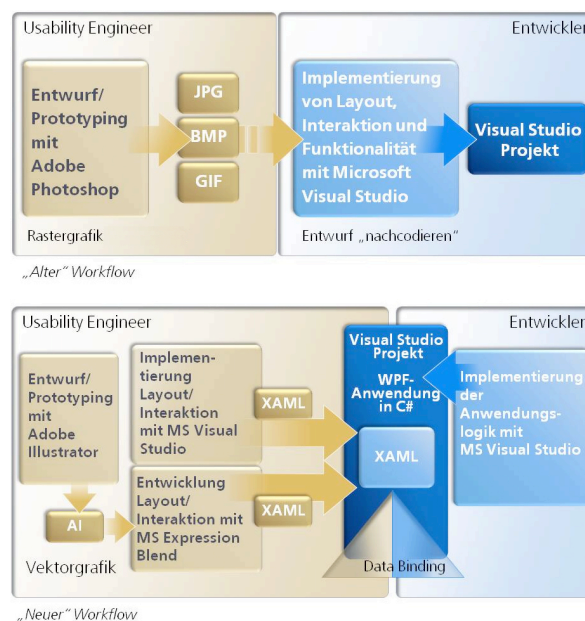


Abbildung 2: „Alter“ und „neuer“ Workflow zwischen Usability Engineer und Softwareentwickler

Im Gegensatz zum „alten“ Workflow erlaubt dieser „neue“ Workflow dem Usability Engineer ein hochgradig iteratives Arbeiten: Er entwirft die Oberfläche entweder als Vektorgrafik und setzt sie dann – selbst! – mit Microsoft Expression Blend in XAML um. Oder er beschreibt sie mit Microsoft Visual Studio direkt in XAML. In beiden Fällen wird der XAML-Code in *demselben* Visual Studio Projekt abgelegt, das auch den C#-Code der Anwendungslogik aufnimmt, den die Entwickler erstellen. Layout und Anwendungslogik, Oberflächenelemente und die Daten, die sie präsentieren sollen, werden über die Technik des *Data Binding* miteinander verknüpft. Dadurch gewinnt der Usability Engineer den Spielraum, das Layout auch dann noch zu ändern, wenn die Anwendungslogik bereits implementiert ist.

## 2.3 Die Anforderung an den Prozess

microTOOL arbeitet seit 2002 agil und setzt seit 2007 in seinen Projekten Scrum ein.

Angestoßen durch den „neuen“ Workflow zwischen Usability Engineer und Softwareentwickler entstand die Idee, die Usability Engineers nicht länger als Zulieferer eines Projekts zu verstehen, sondern direkt im Scrum-Team mitarbeiten zu lassen und das UE in den Scrum-Ablauf zu integrieren. Die Aufgaben zur Gestaltung der Benutzeroberfläche sollten zusammen mit der Entwicklung der Anwendungslogik über dasselbe Product Backlog auf der Basis einheitlicher Ziele vom Product Owner gesteuert werden.

### 3 Scrum im SPSF-Projekt aus Entwicklersicht

microTOOL praktiziert Scrum im SPSF-Projekt in folgender Weise: Das Projektteam arbeitet in zweiwöchigen Sprints. So wie es Scrum vorsieht, sucht das Team bei der Sprint-Planung gemeinsam mit dem Product Owner aus der Liste der Anforderungen, dem *Product Backlog*, die hochpriorisierten Anforderungen heraus, die im Sprint bearbeitet werden sollen. Zuvor geht es aber darum, die Anforderungen – das Team spricht von *Stories* – zu verstehen und zu schätzen. Um zeitnah zur Sprint-Planung zu schätzen, hat sich das Team dafür entschieden, keine separaten Schätzklausuren durchzuführen, sondern in die Sprint-Planung mit einer Schätzzrunde einzusteigen.

Das Product Backlog enthält drei Arten von Stories für die Softwareentwickler:

- Stories der Art **Denke**: Ergebnis sind Referenzimplementierungen, also lauffähiger Code, der einen Lösungsweg für ein Problem aufzeigt.
- Stories der Art **Implementiere Funktion**: Ergebnis sind getestete, lauffähige Funktionen, die im Kontext des Gesamtsystems versioniert sind.
- Stories der Art **Entwickle Tests**: Ergebnis sind NUnit-Tests für Komponenten, die in die Build-Jobs integriert werden und im Daily Build mit ablaufen.

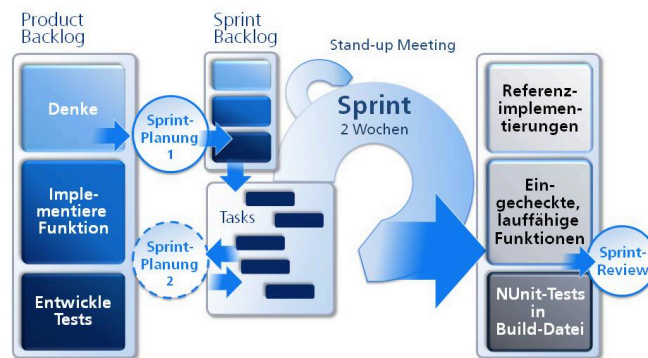


Abbildung 3: Scrum aus Entwicklersicht – wie es im SPSF-Projekt praktiziert wird

Im Anschluss an die Sprint-Planung zerlegt das Team die ausgewählten Stories in elementare Tasks. Es überprüft dabei die Schätzung. Passen die ausgewählten Stories nicht in den Sprint, treffen sich Team und Product Owner am selben Tag zu einer zweiten Sprint-Planungsrunde.

Im Verlauf des Sprints ist der Product Owner für das Team fast ständig verfügbar, diskutiert Lösungsansätze und prüft Ergebnisse, sodass das eigentliche Sprint-Review nur noch formalen Charakter hat.

## 4 Ein Prozessmodell für UE

### 4.1 Der ISO-Prozess

Im UE gibt es ebenso wenig *den* allgemein praktizierten Prozess wie im SE. microTOOL ist bei seinem Vorhaben von dem in der ISO 13407 beschriebenen Prozessmodell zur *benutzerorientierten Gestaltung interaktiver Systeme* (User-Centered Design, UCD) ausgegangen, das in Abbildung 4 skizziert ist. Die vier Prozessschritte haben folgenden Inhalt:

*Benutzer und Nutzungskontext analysieren*: Aufgaben, Arbeitsumgebung und Arbeitsprozesse werden analysiert und dokumentiert. Es werden Benutzer- und Aufgabenprofile erstellt. Dabei werden grundlegende Nutzungsanforderungen erkannt.

*Anforderungen spezifizieren:* Aus den Analyseergebnissen werden konkrete Anforderungen abgeleitet und beschrieben.

*Design-Lösungen entwickeln:* Zur Umsetzung der Anforderungen entstehen im Allgemeinen zunächst sogenannte Low-Fidelity (Lo-Fi) Prototypen, d. h. Skizzen der Benutzeroberfläche auf Papier. Später werden dann High-Fidelity (Hi-Fi) Prototypen wie klickbare Strukturmodelle ohne oder auch mit Layout erstellt. Und schließlich werden lauffähige Lösungen für die Benutzerinteraktion entwickelt.

*Lösungen gegen Anforderungen evaluieren:* Die Design-Lösungen werden auf die Erfüllung der Anforderungen geprüft. Dies kann z. B. in Form von Experten-Reviews, Befragungen oder Usability-Tests geschehen. Abweichungen werden bewertet und sind Ausgangspunkt der nächsten Iteration.

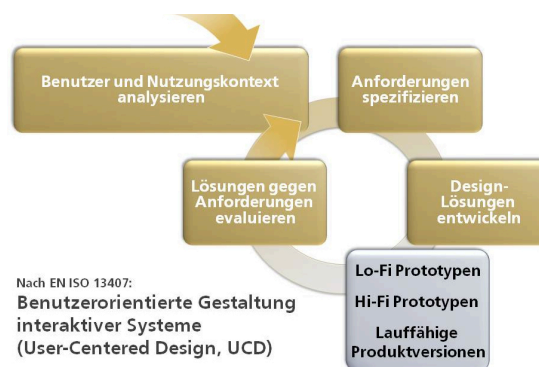


Abbildung 4: Der Usability Engineering Prozess

Dieses UE-Prozessmodell soll – so ist es konzipiert – bestehende Prozessmodelle des SE ergänzen.

#### 4.2 Kompatibilität zur agilen Softwareentwicklung

Die Frage, ob das UE-Prozessmodell mit Scrum verträglich ist, wird man auf den ersten Blick mit Ja beantworten, denn beide sind iterativ, durch Benutzeranforderungen gesteuert und feedback-orientiert. Bei detaillierter Betrachtung ergeben sich jedoch drei Probleme:

*Das Up-front Problem:* Der ISO-Prozess wird in der Praxis häufig analyselastig umgesetzt, d. h., Benutzer- und Nutzungskontextanalyse nehmen zu Projektbeginn großen Raum ein. Vielfach wird vorgeschlagen (siehe u. a. [CL02]), vorab sowohl ein visuelles Schema als auch eine grundsätzliche Interaktionsarchitektur zu entwerfen, um ein einheitliches Look & Feel und eine konsistente Benutzerführung sicherzustellen. Up-front Aktivitäten in diesem Umfang sind bei agilem Vorgehen nicht akzeptabel. *Aber wie kommt man ohne aufwendige Up-front Aktivitäten zu einem homogenen Design der Benutzerschnittstelle und konsistenten Benutzerinteraktionen?*

*Das Synchronisationsproblem:* Zyklen können beim UE von sehr unterschiedlicher Länge sein. So lässt sich ein Papierprototyp sehr schnell erstellen, mit Experten evaluieren und verbessern. Die Entwicklung einer lauffähigen Lösung und der Usability-Test mit Anwendern nimmt sehr viel mehr Zeit in Anspruch. *Wie taktet man diese unterschiedlich langen Zyklen in die gleich langen Sprints eines Scrum-Projekts ein?*

*Das Problem der Nutzerbeteiligung:* UE erfordert Endnutzerbeteiligung. Um anforderungsgerechte Systeme zu entwickeln, wird beim Einsatz von Scrum – wie bei anderen agilen Vorgehensweisen – der Kunde in die Entwicklung einbezogen. Dabei wird oft übersehen, dass mit der Rolle des Kunden der Auftraggeber, nicht aber der Endnutzer gemeint ist. *Wie erreicht man, dass nicht nur anforderungsgerechte, sondern auch endnutzergerechte Systeme entstehen?*



## 5 Die Integration von UE in Scrum

### 5.1 Parallel Tracking – eine Option

Erfahrungen mit UE in agilen Projekten nach Scrum sind bisher nur in geringem Umfang dokumentiert. Zu den dokumentierten Ansätzen gehört das *Parallel Tracking* (siehe [Mi05] und [Sy07]). Es verspricht eine Lösung für das Synchronisationsproblem und das Up-front Problem.

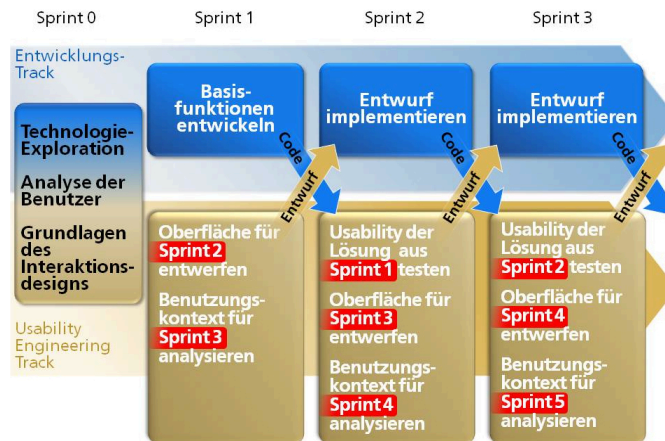


Abbildung 5: Der Ansatz des Parallel Tracking

Beim Parallel Tracking werden, wie Abbildung 5 zeigt, zwei parallele Entwicklungsstränge (Tracks) im Projekt eingerichtet: je einer für die Softwareentwicklung und das UE. Das Team wird also in zwei Teilteams aufgeteilt. Beide Teams starten mit den allernötigsten Up-front Aktivitäten. Dazu gehören für das UE-Team eine erste Nutzeranalyse und ein grobes Interaktionskonzept. Im Sprint 1 erstellen die Softwareentwickler technische Basisfunktionen ohne Benutzeroberfläche. Das UE-Team entwirft parallel dazu die Oberfläche für die Funktionen, die im nächsten Sprint realisiert werden sollen. Außerdem analysiert es den Benutzungskontext für die Funktionen, die im übernächsten Sprint entstehen sollen. Am Sprint-Ende übergibt das UE-Team die Entwürfe für die Benutzeroberfläche für den nächsten Sprint an das Entwicklerteam. Im Gegenzug erhält es die im Sprint implementierten Funktionen, um die Usability zu testen. Dieses Vorgehen setzt sich dann sprintweise fort.

microTOOL hat sich aus folgenden Gründen gegen das Parallel Tracking entschieden:

- Bei diesem Vorgehen muss über drei Sprints geplant werden. Aus den praktizierten 2-wöchigen Sprints entstehen damit implizit 6-wöchige Sprints – eine Einschränkung der Agilität, die für das SPSF-Projekt nicht akzeptabel war.
- Hinter der Übergabe eines Oberflächenentwurfs am Sprintende steht der „alte“, unproduktive Workflow (vgl. Abbildung 2) zwischen UE-Team und Softwareentwicklern. Damit widerspricht das Parallel Tracking dem angestrebten Ziel der gemeinsamen Erarbeitung von Lösungen auf der Basis des „neuen“ iterativen Workflows.
- Erfahrungen anderer Unternehmen (vgl. [SS09]) zeigen, dass die Synchronität nur schwer aufrechtzuerhalten ist. So wird u. a. von Wartezeiten des UE-Teams zum Projektende hin berichtet.

### 5.2 Scrum im SPSF-Projekt aus UE-Sicht

Das SPSF-Projekt begann mit einem Sprint 0, den die Softwareentwickler zur Exploration der Basistechnik nutzten. Parallel dazu nahm der Usability Engineer seine Arbeit auf. Für ihn standen im Sprint 0 zwei Aufgaben an:

- Benutzer- und Nutzungskontextanalyse mit sogenannten Discount-Methoden<sup>1</sup>, d. h. schnellen und kostengünstigen Methoden des UE. Dazu zählt die Entwicklung von *Assumption-Based Personas* (vgl. [Wi10] und [CRC07]).
- Anlegen eines Style Guide im Firmen-Wiki, der kontinuierlich fortgeschrieben wird.



Abbildung 6: Interaktive Entwicklung von *Assumption-based Personas* im Sprint 0

Nach dem Abschluss von Sprint 0 wurde die Arbeit in folgender Weise organisiert: Als „Vollmitglied“ des Scrum-Teams nimmt der Usability Engineer an allen Meetings zur Sprint-Planung und zum Sprint-Review sowie am Daily Scrum teil.

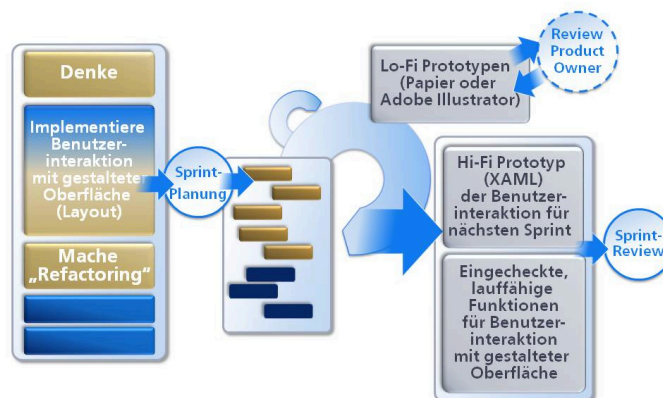


Abbildung 7: Scrum aus Sicht des Usability Engineers – wie es im SPSF-Projekt praktiziert wird

Wie für die Softwareentwickler gibt es auch für den Usability Engineer im Product Backlog drei Arten von Stories (vgl. Abbildung 7):

- Stories der Art **Denke über ein Problem der Benutzerinteraktion** nach. Sie sind eine Anleihe vom Parallel Tracking und dienen der Vorbereitung einer Lösung, die im darauffolgenden Sprint (also nicht, wie bei Parallel Tracking, erst im übernächsten Sprint) realisiert wird. Zunächst entstehen Lo-Fi Prototypen, Skizzen und Abläufe auf Papier, gegebenenfalls auch Grafiken mit Adobe Illustrator. Sie werden vom Product Owner bereits im laufenden Sprint geprüft. Diese Lo-Fi Prototypen sind Zwischenergebnisse auf dem Weg zu Hi-Fi Prototypen der Benutzerinteraktion, die in XAML implementiert werden.
- Stories der Art **Implementiere Benutzerinteraktion** einschließlich Layout der Benutzerschnittstelle. Sie werden meist gemeinsam mit Entwicklern bearbeitet. Ergebnis ist eine im Kontext des Gesamtsystems versionierte, lauffähige Funktion mit Benutzerinteraktion und vollständig gestalteter Oberfläche.

<sup>1</sup> Der Begriff wurde ca. 1989 geprägt und geht auf Jakob Nielsen zurück, einem der renommiertesten Experten im Bereich der Web Usability.

### 3.3 Integration von Usability Engineering in den agilen Entwicklungsprozess nach Scrum

- Stories der Art **Mache Refactoring**. Sie dienen der Vereinheitlichung des Layouts der Benutzerschnittstelle und der Benutzerinteraktion. Sie sollen die fehlende Phase der Up-front Analyse und des Up-front Designs ausgleichen.

Wie die Aufgaben von Usability Engineer und Softwareentwickler innerhalb eines Sprints konkret ineinandergreifen, zeigt das nachfolgende Fallbeispiel aus dem SPSF-Projekt.

## 6 Ein Fallbeispiel aus dem SPSF-Projekt

Die Zusammenarbeit von Softwareentwicklern und Usability Engineer im Scrum-Ablauf wird wesentlich von der Architektur der zu entwickelnden Anwendung beeinflusst. Deshalb hier zunächst ein kurzer Blick darauf: Es handelt sich um eine Rich Internet Anwendung mit Client- und Server-Komponenten (vgl. Abbildung 8). Auf dem Server befinden sich eine Datenbank – beschrieben durch ein Entity Modell – und Service-Operationen, die dazu dienen, Dateninhalte aus der Datenbank zusammenzustellen und als sogenannte *View Models* auf einen Client zu übertragen bzw. – umgekehrt – die auf dem Client geänderten Daten eines View Model zu speichern.

Die Daten eines View Model besitzen auf dem Client ein Layout, in dem sie angezeigt und vom Benutzer bearbeitet werden – wir sprechen von einem *View*. Zu einem View können Steuerelemente gehören. Der Benutzer kann auf der Benutzeroberfläche Ereignisse auslösen und mit Kommandos Service-Operationen aufrufen, die die Daten von View Models bereitstellen oder speichern.

Views (Layout) und View Models (Dateninhalte) werden über die bereits erwähnte Technik des *Data Binding* miteinander verknüpft.

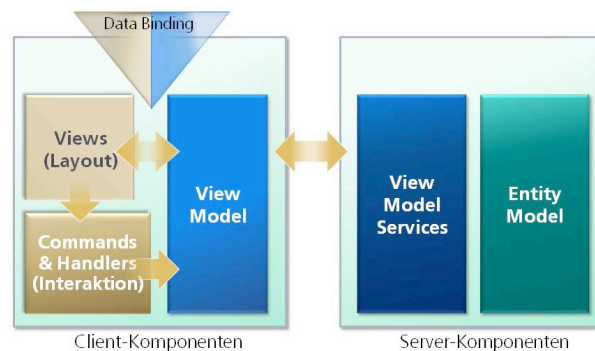


Abbildung 8: Rich Internet Application für .NET – die Architektur

Aufgabe des Usability Engineer ist es, einen View (auf der Basis eines im vorherigen Sprint entworfenen Prototyps) zu entwickeln. Der Softwareentwickler erstellt das zugehörige View Model.

Daraus ergeben sich die in Abbildung 9 dargestellten Tasks für Usability Engineer und Softwareentwickler:

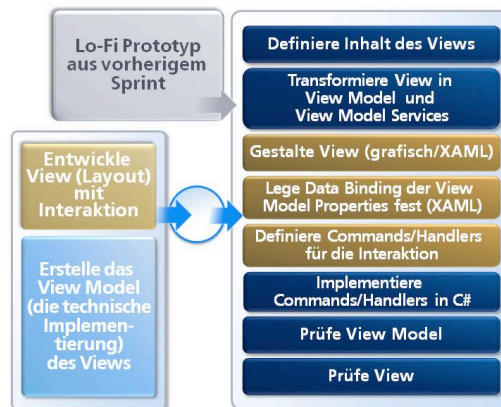


Abbildung 9: Stories (links) und die daraus abgeleiteten Tasks (rechts) für Entwickler (blau) und Usability Engineer (braun)

Was hinter diesen Tasks steht und welchen Umfang sie typischerweise haben, soll am Beispiel der Entwicklung einer Funktion zur *Diagrammvorschau* verdeutlicht werden.

Abbildung 10 zeigt anhand des Original Lo-Fi Prototyps aus dem Projekt, wie die Diagrammvorschau funktionieren soll: Klickt der Benutzer auf einen Eintrag in einer Elementliste, dann sollen die zu dem Listeneintrag gehörenden Diagramme – reduziert auf ihre Geometrie und Farbigkeit – in einem Vorschaubereich angezeigt werden. Bei Doppelklick auf ein Vorschauelement soll das zugehörige Diagramm geöffnet werden.

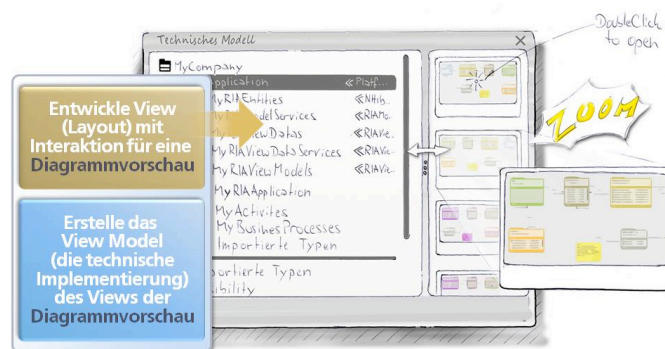


Abbildung 10: Stories zur Entwicklung eines Features zur Diagrammvorschau auf der Basis eines zuvor entworfenen Lo-Fi Prototyps – exemplarisch

Die ersten beiden Tasks *Definiere den Inhalt des Views für die Diagrammvorschau* und *Transformiere den View in ein View Model und View Model Services* sind vom Softwareentwickler durchzuführen. Die dazu eingesetzte, hoch automatisierte Technik der *modellgetriebenen Entwicklung (Model-Driven Development, kurz MDD)* reduziert den benötigten Aufwand auf maximal einen Tag. Der genauere Blick auf beide Tasks zeigt, wie dies möglich wird:

Den Inhalt des Views zu definieren bedeutet, die Daten aus dem Entity Modell zusammenstellen, die vom Server benötigt werden, um eine Diagrammvorschau auf der Benutzeroberfläche des Client anzuzeigen. Der View wird einfach – d. h. mithilfe einer Tool-Funktion – „zusammengeklickt“ (siehe Abbildung 11, links). Die zweite Task *Transformiere den View in ein View Model und View Model Services* beinhaltet die eigentliche Implementierung des View Model und der zugehörigen View Model Services in C#. Sie ist durch den Einsatz von *Modelltransformationen* vollständig automatisiert und erfordert vom Softwareentwickler lediglich das Auslösen einer Kontextmenüfunktion seines Tools (siehe Abbildung 11, rechts).



### 3.3 Integration von Usability Engineering in den agilen Entwicklungsprozess nach Scrum

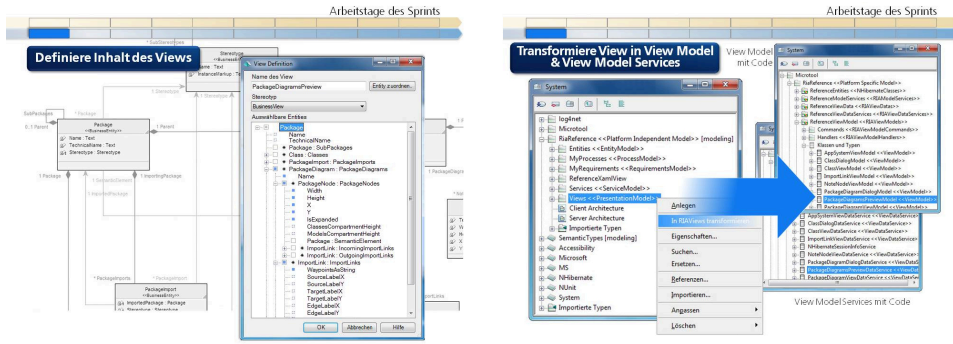


Abbildung 11: Die ersten beiden Entwickler-Tasks – verteilt auf die Arbeitstage des Sprints (siehe Zeitstrahl am oberen Rand der Grafik)

Danach ist der Usability Engineer mit der grafischen Gestaltung des Views dran: Hierzu muss er ein passendes Oberflächenelement auswählen und die Oberfläche im Detail in XAML programmieren. Da das Layout der Diagramme in der Vorschau nicht statisch ist, kann er es nicht mit Microsoft Expression Blend grafisch entwerfen und automatisch in XAML umsetzen lassen. Er muss es selbst mit Microsoft Visual Studio implementieren. Dauer: ca. 3 Tage (Abbildung 12, links). Dann muss der Usability Engineer für das Data Binding sorgen, indem er im XAML-Code den Oberflächenelementen die Datenelemente des View Model zuweist, die darin jeweils angezeigt werden sollen. Dauer: ca. 1 Tag (Abbildung 12, rechts).

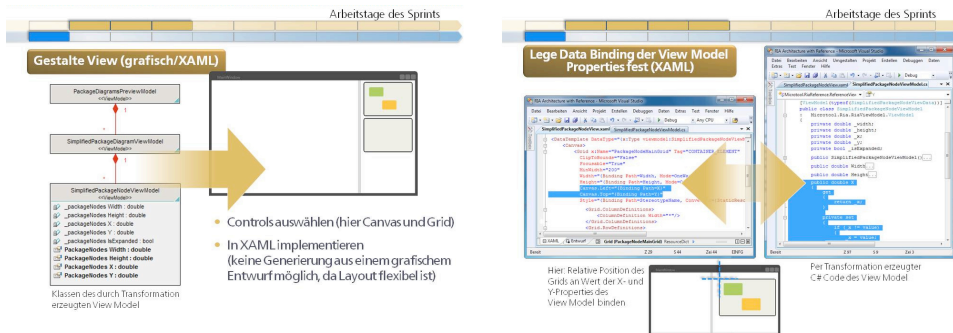


Abbildung 12: Folge-Tasks für den Usability Engineer (Dauer wiederum am Zeitstrahl ablesbar)  
 Per Doppelklick soll ein Diagramm aus der Vorschau heraus geöffnet werden können. Hierzu muss ein Event Handler definiert und eine Zuordnung in XAML implementiert werden. Diese Festlegung der Benutzerinteraktion ist Aufgabe des Usability Engineer. Dauer: ca. 1 Tag. Die Implementierung des Event Handler übernimmt der Softwareentwickler. Dauer: ca. 2 Tage (siehe Abbildung 13).

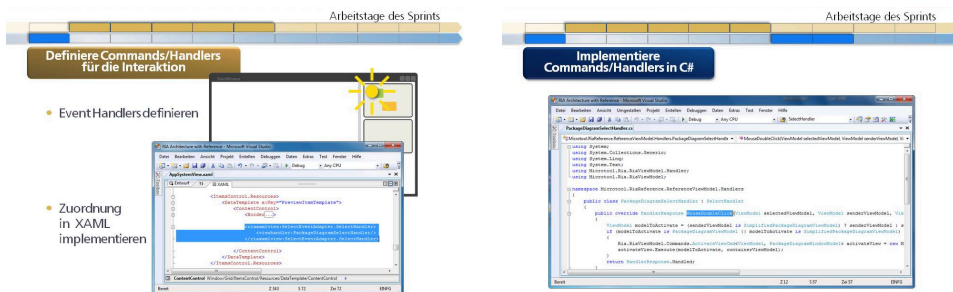


Abbildung 13: Die abschließenden Tasks von Usability Engineer (links) und Entwickler (rechts)  
 Bis auf den Test ist die Entwicklung der Diagrammvorschau damit abgeschlossen. Abbildung 14 veranschaulicht das Ergebnis.

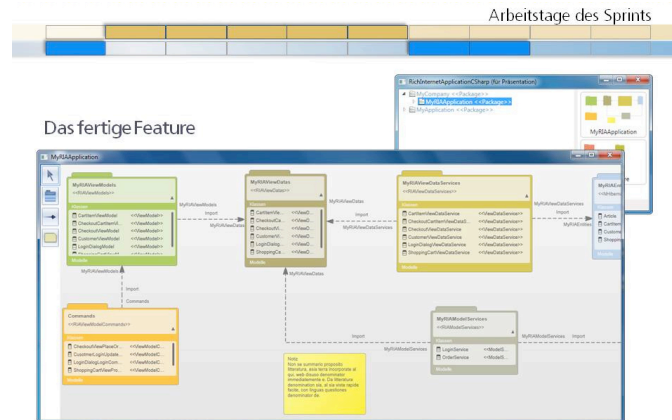


Abbildung 14: Das fertige Feature

## 7 Fazit

Die hier vorgestellte Integration des UE in Scrum hat sich in der Praxis bewährt. Die in Abschnitt 4.2 dargestellten Probleme wurden weitgehend gelöst:

*Das Up-front Problem:* Durch den Einsatz von Discount-Methoden für die Benutzeranalyse im Sprint 0 und das kontinuierliche Refactoring der Benutzeroberfläche in jedem Sprint ist für die Entwicklung einer konsistenten Benutzeroberfläche gesorgt. Voraussetzung ist allerdings – auch das hat die Praxis gezeigt –, dass Product Owner und Usability Engineer eine gemeinsame Vision von Layout und Interaktionsarchitektur der Anwendung entwickeln.

*Das Synchronisationsproblem:* Drei Dinge tragen dazu bei, dass dieses Problem nicht mehr auftritt: die Zusammenarbeit nach dem „neuen“, iterativen Workflow, die modellgetriebene Entwicklung, durch die die Geschwindigkeiten von Softwareentwicklern und Usability Engineer angeglichen werden, und Anleihen beim Parallel Tracking. Letzteres bedeutet speziell: Product Owner und Usability Engineer müssen den Entwicklern konzeptionell immer einen Schritt voraus sein.

*Das Problem der Nutzerbeteiligung:* Die Anforderungen an die Benutzeroberfläche wurden anfangs allein vom Product Owner gestellt. Der Usability Engineer war Anwalt der Endnutzer und Berater des Product Owner. Es wurde inzwischen damit begonnen, die Zusammenarbeit des Usability Engineer mit Endnutzern beim Kunden zu organisieren und Endnutzer u. a. bei Usability-Tests einzubeziehen. Die Verbesserung der Endnutzerbeteiligung ist die zentrale Herausforderung, vor der microTOOL bei der Weiterentwicklung seines Entwicklungsprozesses steht.

## Literaturverzeichnis

- [Co02] Constantine, L. L.: Process Agility and Software Usability: Toward Lightweight Usage-Centered Design, Information Age, August 2002.
- [CRC07] Cooper, A., Reimann, R., Cronin, D.: About Face 3. The Essentials of Interaction Design, Wiley Publishing, Inc., 2007.
- [Mi05] Miller, Lynn: Case Study of Customer Input For a Successful Product, Proceedings of the Agile Development Conference 2005, S. 225-234, IEEE Computer Society
- [SSt09] Schauderna, F., Stucki, J.: Praxisbericht aus drei Perspektiven: Usability Engineering, Design und Entwicklung im agilen User Centered Design, Usability Professionals 2009, S. 274-278.
- [Sy07] Sy, D.: Adapting Usability Investigations for Agile User-Centered Design, Journal of Usability Studies, Band 2, 3. Ausgabe, S. 112-132.
- [Wi10] Wilson, Chauncey (Ed.): User Experience Re-Mastered, Your Guide to Getting the Right Design, Morgan Kaufmann Publishers, 2010.

Ursula Meseberg  
microTOOL GmbH · Berlin

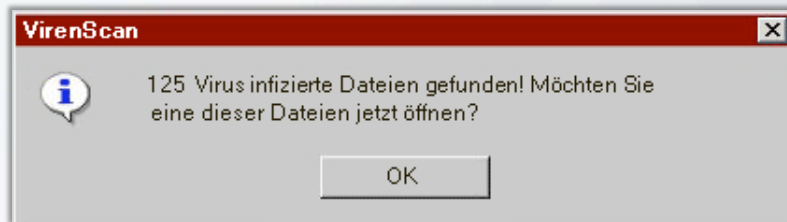
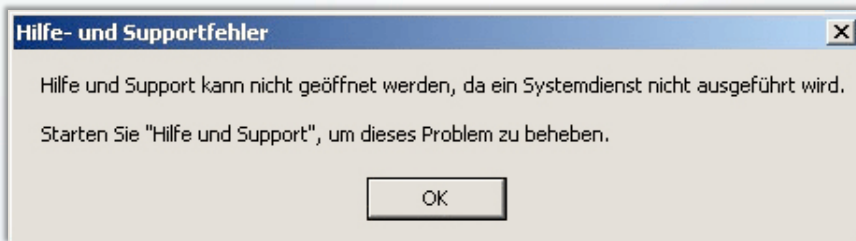
SEE<sup>2010</sup>  
Köln 03./04. Mai

# Integration von Usability Engineering in den agilen Entwicklungsprozess nach Scrum

Ein Erfahrungsbericht

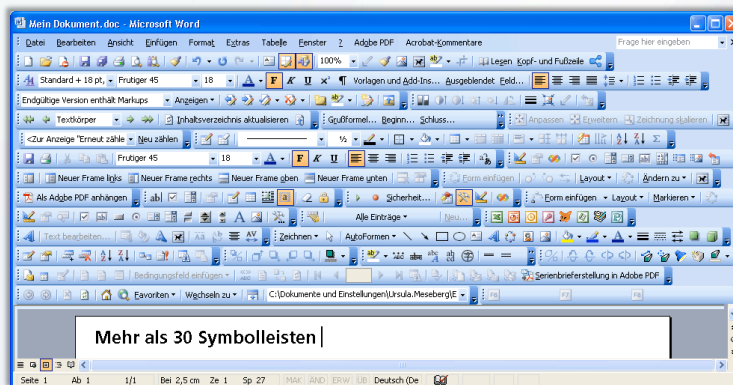
© 2010 microTOOL GmbH, Berlin. Alle Rechte  
vorbehalten.

# Usability Um was geht es?

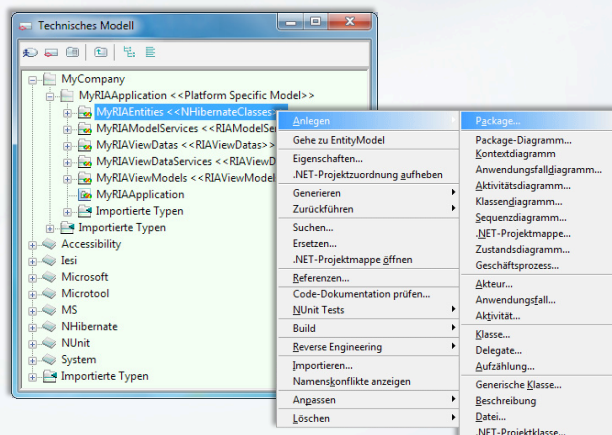


# Darum?

3



# Darum!



4





Bessere **Usability** durch einen besseren **Entwicklungsprozess**

Integration von **Usability Engineering** in die **Softwareentwicklung** nach **Scrum**

Unsere Ziele

6

The slide features a dark blue background with white and yellow text. The main heading is 'Bessere Usability durch einen besseren Entwicklungsprozess', where 'Usability' is in yellow and 'Entwicklungsprozess' is in blue. Below this is the subtitle 'Integration von Usability Engineering in die Softwareentwicklung nach Scrum', with 'Usability Engineering' in yellow and 'Softwareentwicklung' in blue. At the bottom, the text 'Unsere Ziele' is written in white. The number '6' is in the bottom right corner.



7

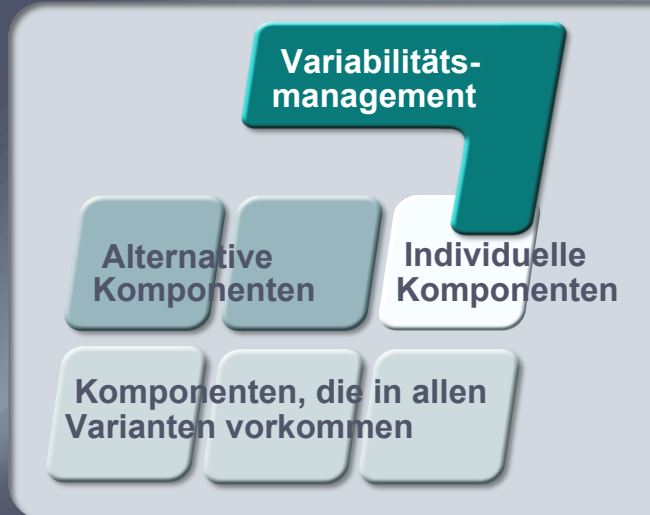
8

- Das Projekt
- Scrum aus Entwicklersicht
- Der Usability Engineering Prozess
- Integrationsansatz: Parallel Tracking
- Integration: Unser Weg
- Scrum aus Sicht des Usability Engineer
- Ein Beispiel aus dem Projekt
- Lessons Learned

## Agenda

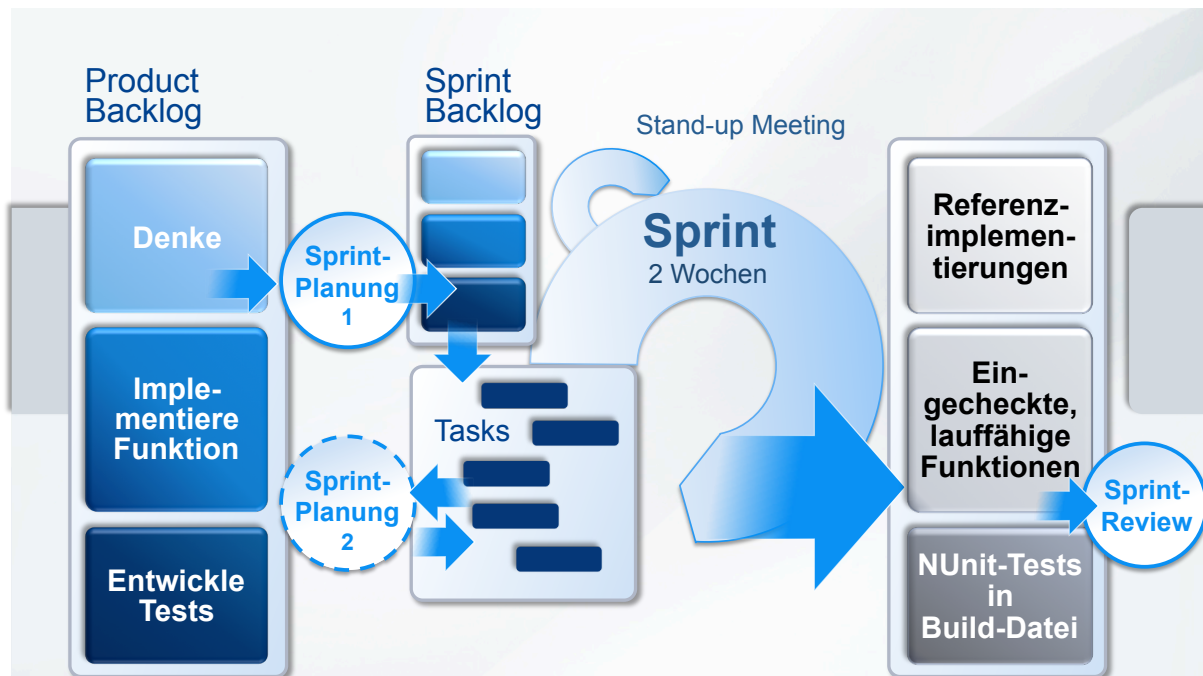
## Entwicklung einer Software-Produktlinie für modellgetriebene Software-Fabriken

- Gefördert vom BMBF im Rahmen von **KMU-innovativ: IKT**
- Technologie: Microsoft .NET, WPF, C#



## Das Projekt SPSF

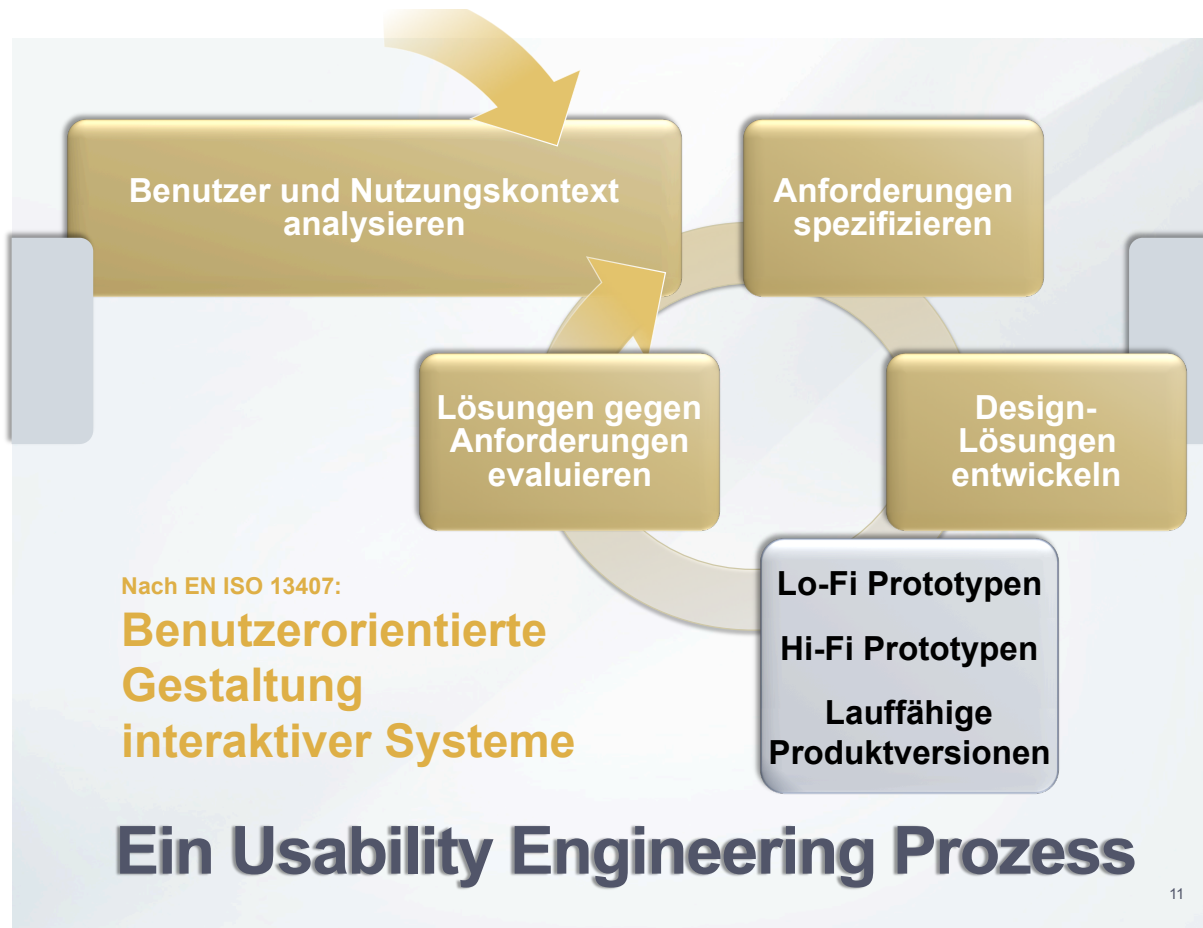
9



Scrum aus Entwicklersicht – wie wir es nutzen

## Der Entwicklungsprozess

10



Passen **Scrum** und der **Usability Engineering Prozess** zusammen?

**Beide sind**

**anforderungsgetrieben**

**iterativ**

**feedback-orientiert**

13

**Aber...**

**Usability Engineering**  
setzt auf ...

**Up-front-Analyse** von Nutzerwünschen  
und Nutzerverhalten

**Up-front-Design** von visuellen Schemata und  
Interaktionsarchitektur für eine einheitliche und  
konsistente Benutzeroberfläche

14



**Up-front-Aktivitäten** sind ein No-go  
der agilen Entwicklung:

Wie kommt man ohne sie zu

**homogenem Interface Design** und  
**konsistenter Benutzerinteraktion** ?

## Das Up-front-Problem

15

Aber...

## Usability Engineering hat ...

**sehr kurze Zyklen:** Anforderungen  
definieren – Lo-Fi Prototyp erstellen – Usability  
am Prototyp evaluieren

**auch sehr lange Zyklen** z.B. durch  
Usability-Tests mit Endnutzern

16

Wie **taktet** man diese **Zyklen**  
in gleich lange Sprints ein?

**Das Synchronisations-  
problem**

17

**Aber...**

**Usability Engineering** erfordert  
Endnutzerbeteiligung

In Scrum werden die Anforderungen des  
Kunden/Auftraggebers umgesetzt

Der Kunde ist nicht der **Endnutzer**

18

Wie erreicht man, dass nicht nur

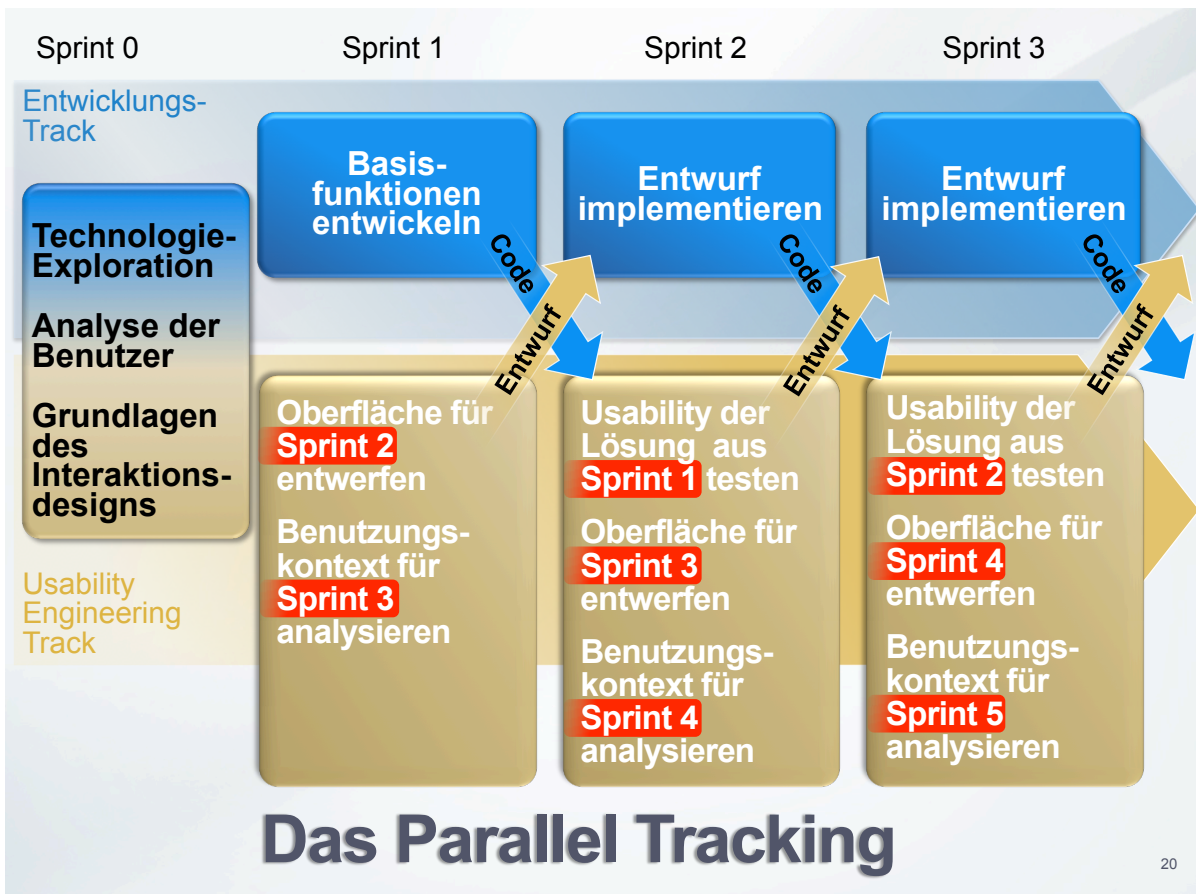
**anforderungsgerechte,**

sondern auch **endnutzergerechte**

**Systeme** entstehen?

## Das Problem der Nutzerbeteiligung

19



20



## Agilität eingeschränkt

Planung über drei Sprints

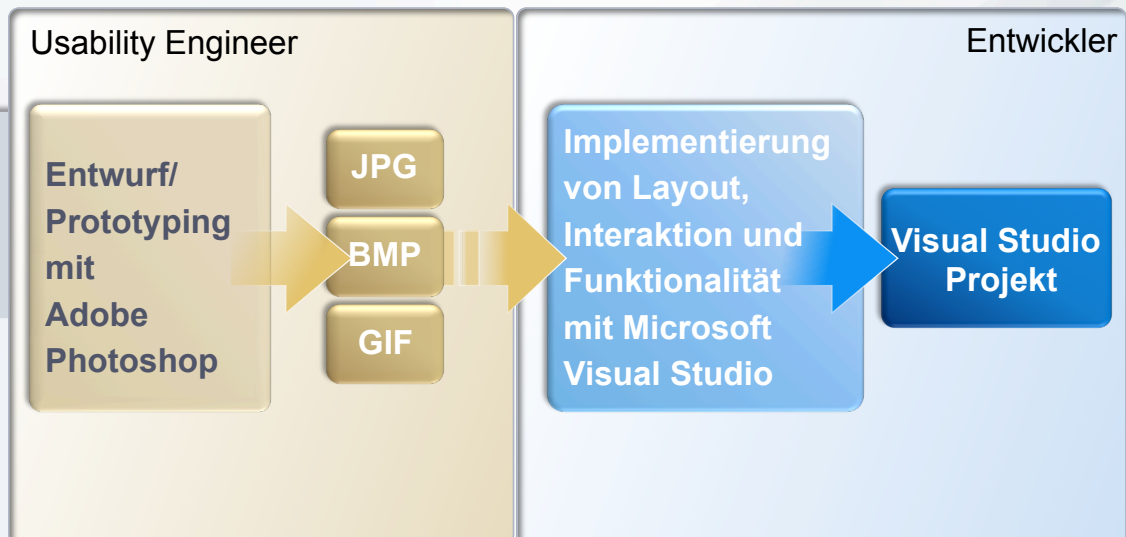
## Synchronität nicht stabil

## Zusammenarbeit unproduktiv

Klassischer Workflow Usability Engineer → Entwickler

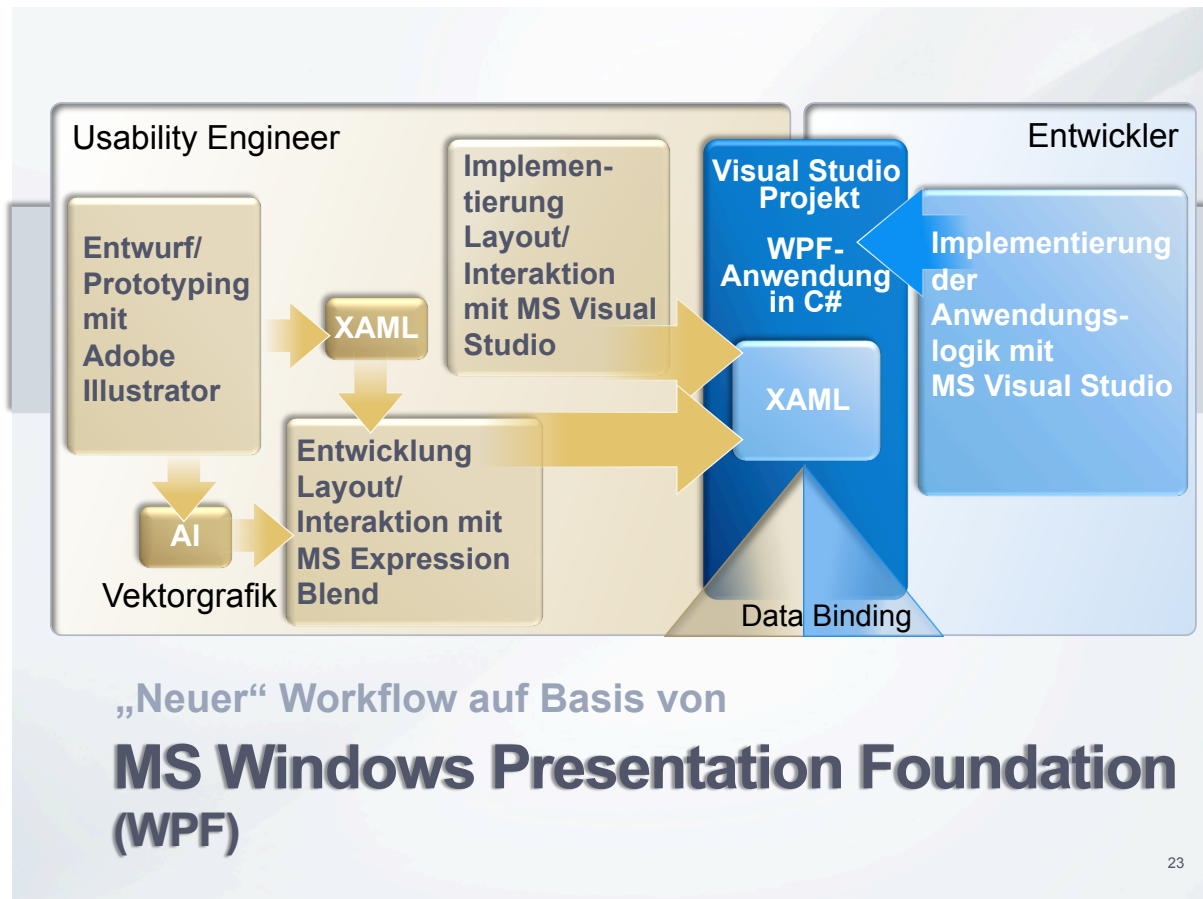
**Parallel Tracking**

21



„Klassischer“ Workflow mit  
**nicht direkt verwertbarer Rastergrafik**

22





- Benutzer- und Nutzungskontextanalyse mit Discount-Methoden wie Assumption-Based Personas
- Anlegen eines Style Guides im Firmen-Wiki
- Konzentration auf Interface Design – Interaction Design nachgelagert

## Guter Start mit Sprint 0

25

# Aber dann ...

- Aufteilung des Usability Engineering auf zwei Rollen unproduktiv
- Layout vor Interaktion: Falsche Reihenfolge
- Interface Designer technisch überfordert

## ... nur Bilder statt Lösungen

26



Das Team:

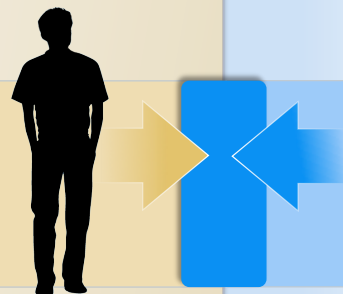
**„Aber wir brauchen doch jemanden fürs User Interface!“**





29

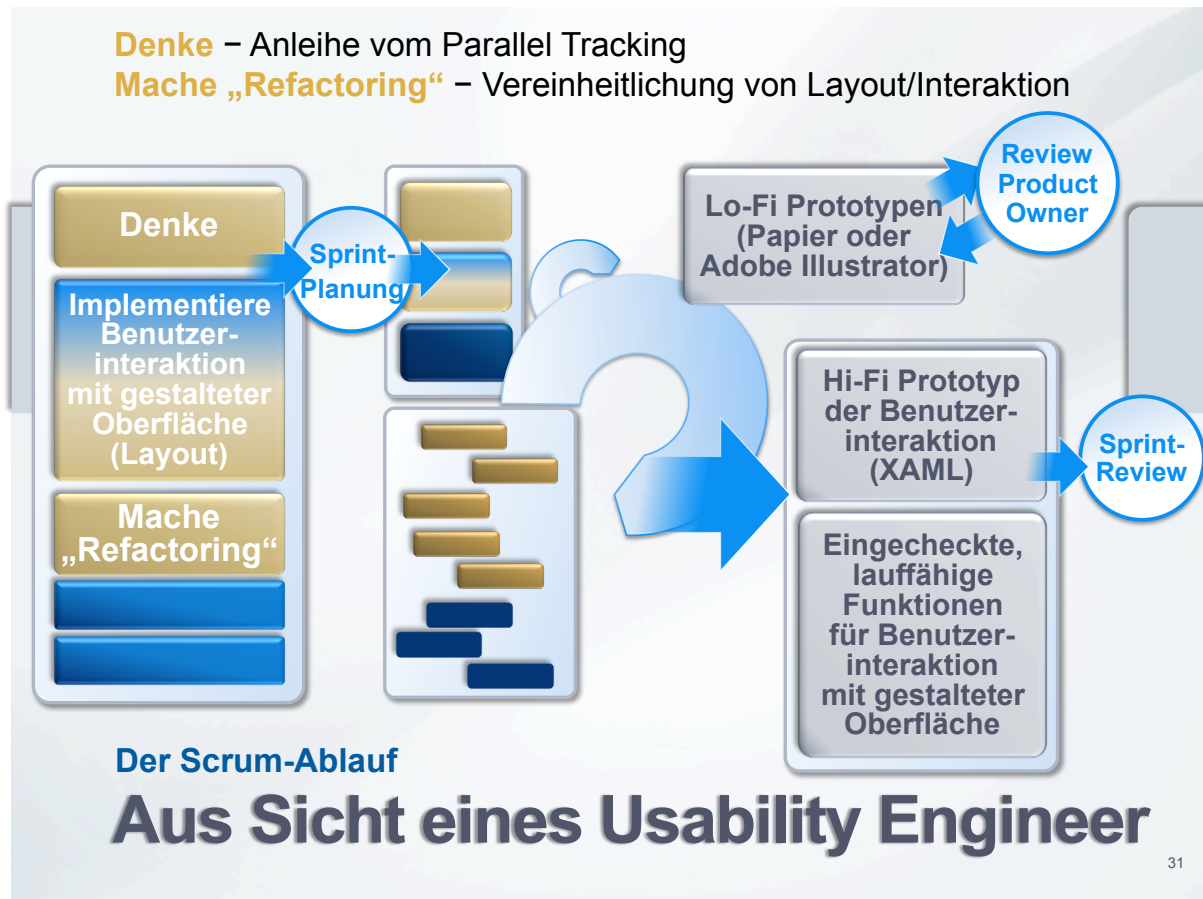
## Im zweiten Anlauf

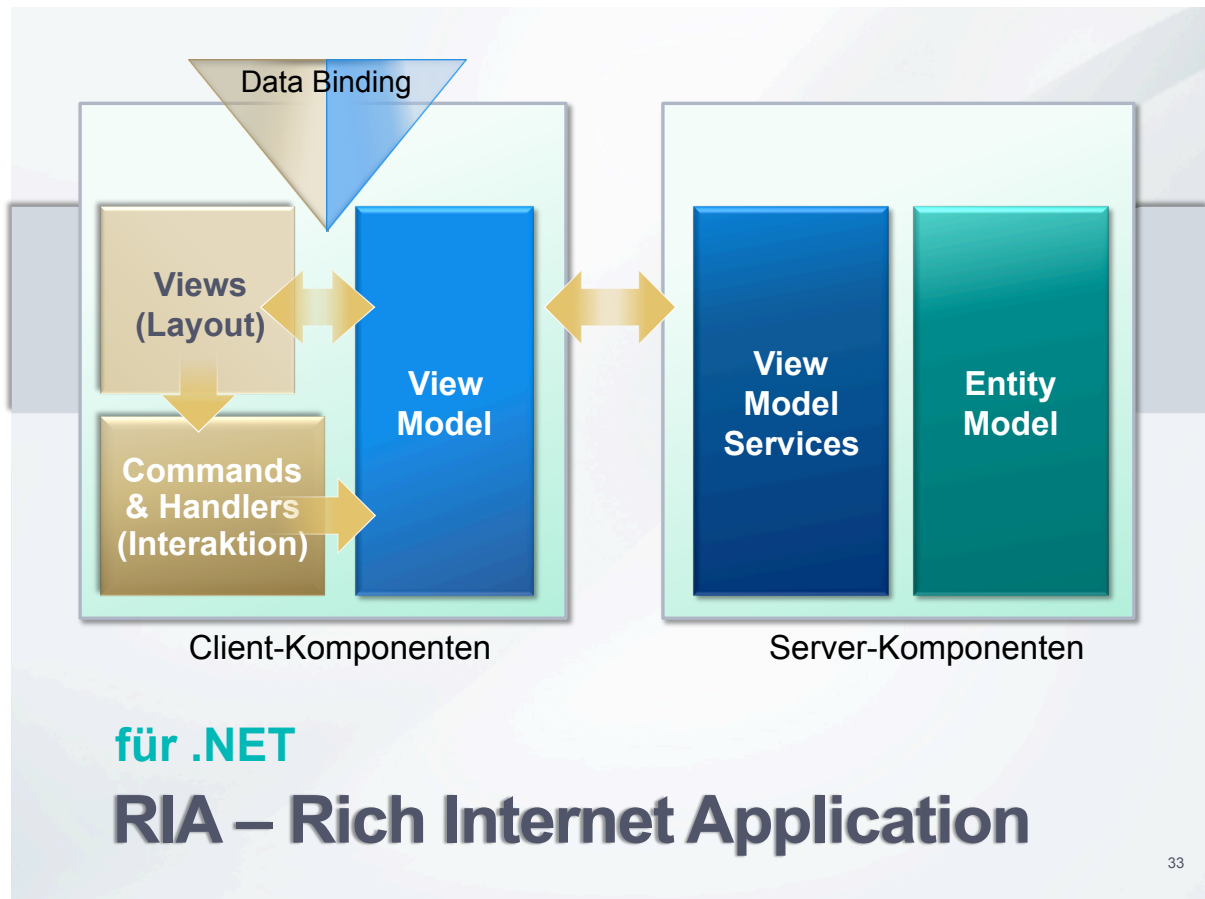


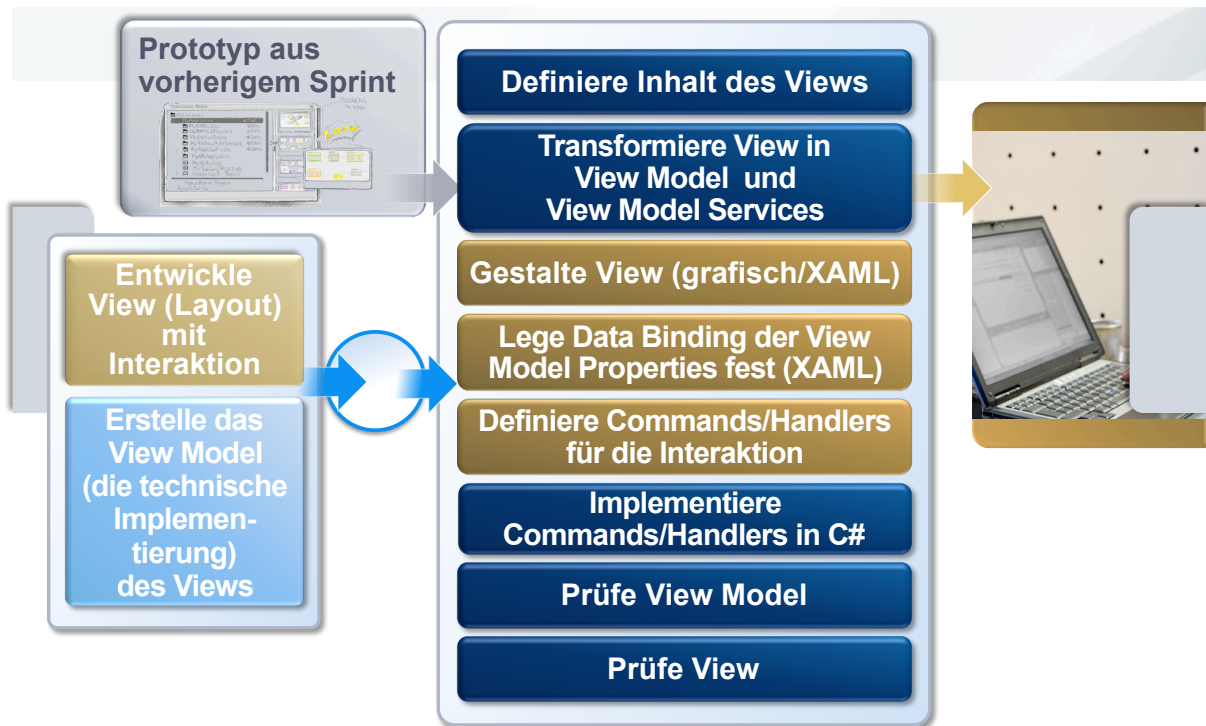
- Festhalten am „neuen“ Workflow
- Usability Engineer – besetzt mit hoher fachlicher und technischer Kompetenz
- Usability Engineer bleibt „Vollmitglied“ des Scrum-Teams
- Erst Interaction Design – dann Interface Design

# Unser Weg

30



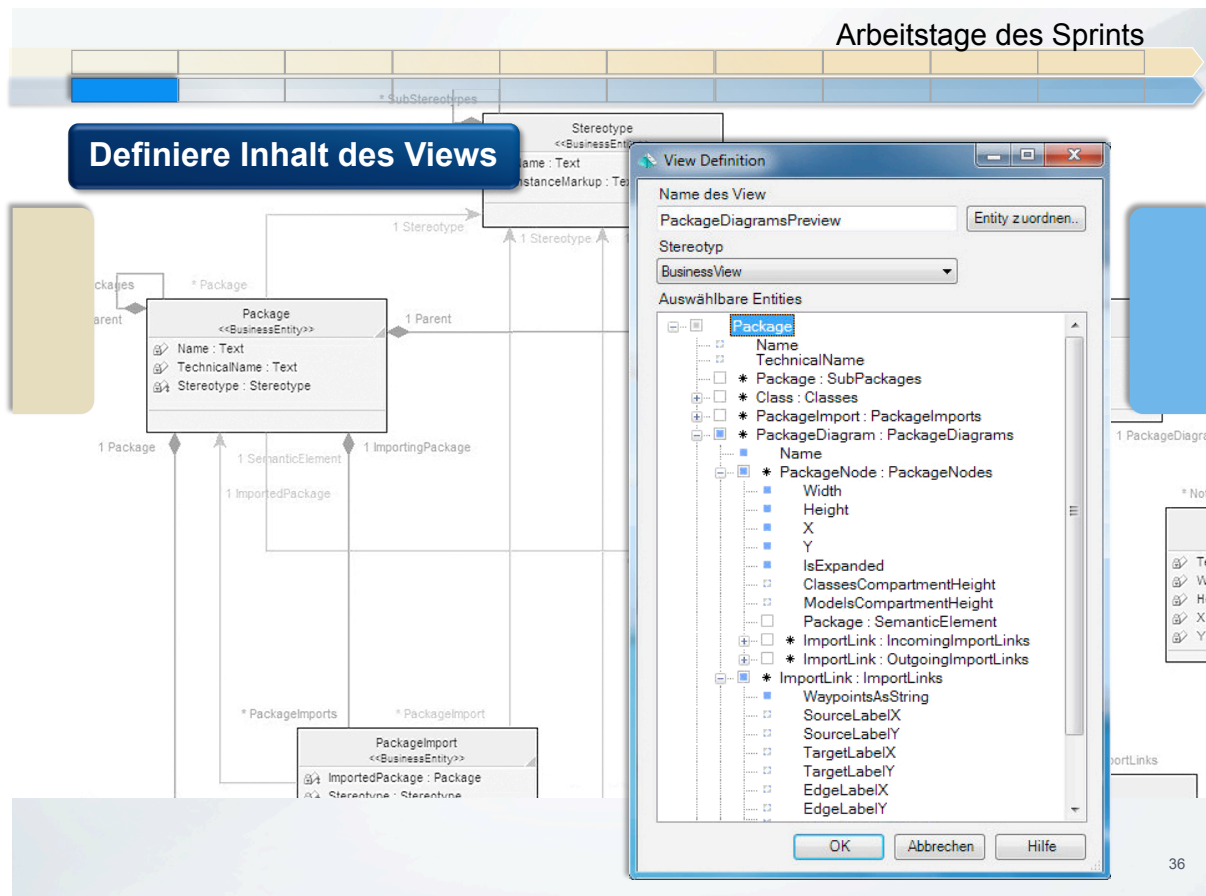




## Tasks

für Usability Engineer und Entwickler

35



36



Arbeitstage des Sprints

**Transformiere View in View Model & View Model Services**

View Model mit Code

**Automatisiert: Model-driven Development**

View Model Services mit Code

37

Arbeitstage des Sprints

**Gestalte View (grafisch/XAML)**

```

classDiagram
    class PackageDiagramsPreviewModel["PackageDiagramsPreviewModel <<ViewModel>>"]
    class SimplifiedPackageDiagramViewModel["SimplifiedPackageDiagramViewModel <<ViewModel>>"]
    class SimplifiedPackageNodeViewModel["SimplifiedPackageNodeViewModel <<ViewModel>>"]
    PackageDiagramsPreviewModel "1" -- "*" SimplifiedPackageDiagramViewModel
    SimplifiedPackageDiagramViewModel "1" -- "*" SimplifiedPackageNodeViewModel
            
```

PackageDiagramsPreviewModel <<ViewModel>>  
 SimplifiedPackageDiagramViewModel <<ViewModel>>  
 SimplifiedPackageNodeViewModel <<ViewModel>>  
 \_packageNodes Width : double  
 \_packageNodes Height : double  
 \_packageNodes X : double  
 \_packageNodes Y : double  
 \_packageNodes IsExpanded : bool  
 PackageNodes Width : double  
 PackageNodes Height : double  
 PackageNodes X : double  
 PackageNodes Y : double

Klassen des durch Transformation erzeugten View Model

- Controls auswählen (hier Canvas und Grid)
- In XAML implementieren (keine Generierung aus einem grafischem Entwurf möglich, da Layout flexibel ist)

38

Arbeitstage des Sprints

Lege Data Binding der View Model Properties fest (XAML)

```

<DataTemplate DataType="{x:Type viewModel:SimplifiedPackageNodeViewModel}">
<Canvas>
<Grid x:Name="PackageNodeMainGrid" Tag="CONTAINER_ELEMENT"
ClipToBounds="False"
Focusable="True"
MinWidth="200"
Width="{Binding Path=Width, Mode=OneWay}"
Height="{Binding Path=Height, Mode=OneWay}"
Canvas.Left="{Binding Path=X}"
Canvas.Top="{Binding Path=Y}"
Style="{Binding Path=StereotypeName, Converter={StaticResource...}}"/>
<Grid.ColumnDefinitions>
<ColumnDefinition Width="*" />
</Grid.ColumnDefinitions>
<Grid.RowDefinitions>
</Grid.RowDefinitions>

```

```

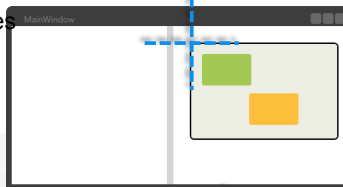
[ViewModel(typeof(SimplifiedPackageNodeViewDataModel))]
public class SimplifiedPackageNodeViewModel
: Microtool.Ria.RiaViewModel.ViewModel
{
private double _width;
private double _height;
private double _x;
private double _y;
private bool _isExpanded;

public SimplifiedPackageNodeViewModel()
{
}

public double Width
{
get
{
return _width;
}
private set
{
if (_width != value)
{
_x = value;
}
}
}
}

```

Hier: Relative Position des Grids an Wert der X- und Y-Properties des View Model binden



Per Transformation erzeugter C# Code des View Model

Arbeitstage des Sprints

Definiere Commands/Handlers für die Interaktion

- Event Handlers definieren



- Zuordnung in XAML implementieren

```

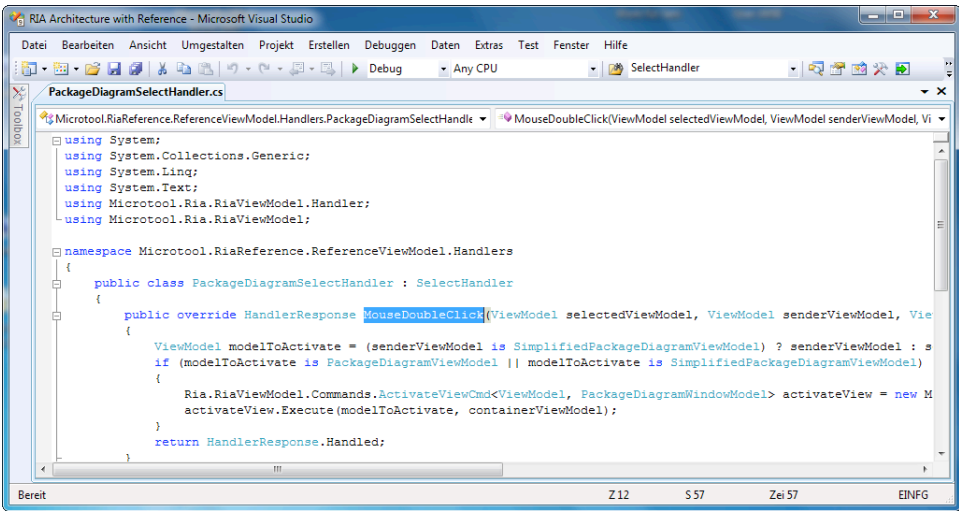
<ItemsControl.Resources>
<DataTemplate x:Key="PreviewItemTemplate">
<ContentControl>
<Border...>
<riaxamlview:SelectEventAdapter.SelectHandler>
<viewhandler:PackageDiagramSelectHandler/>
</riaxamlview:SelectEventAdapter.SelectHandler>
</ContentControl>
</DataTemplate>
</ItemsControl.Resources>

```

### 3.3 Integration von Usability Engineering in den agilen Entwicklungsprozess nach Scrum

Arbeitstage des Sprints

**Implementiere  
Commands/Handlers in C#**



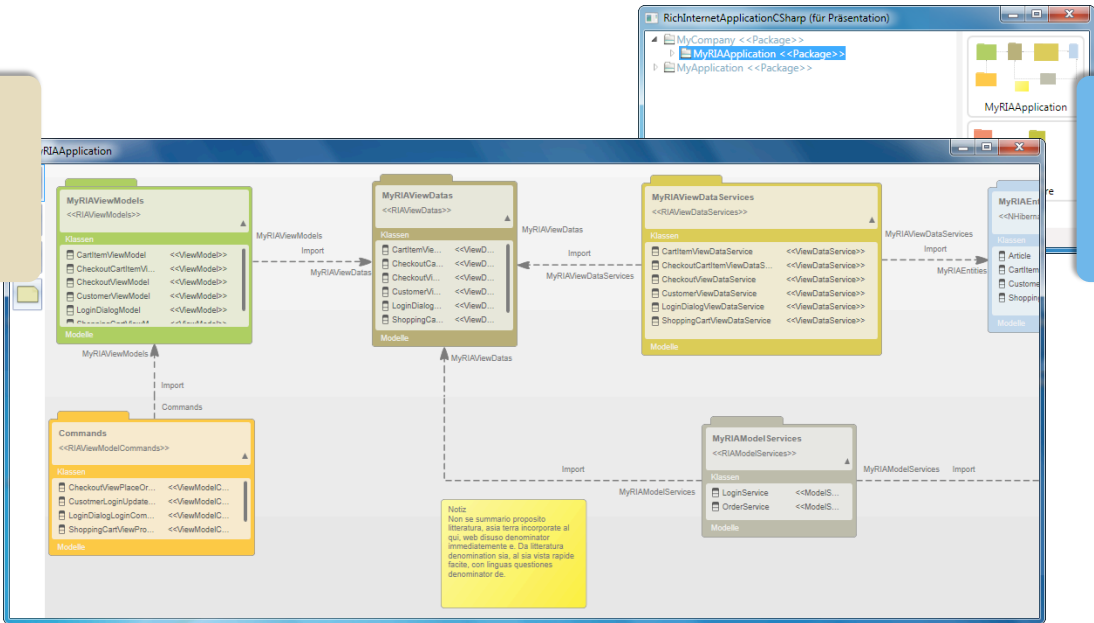
```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Microsoft.Ria.RiaViewModel.Handler;
using Microsoft.Ria.RiaViewModel;

namespace Microsoft.RiaReference.ReferenceViewModel.Handlers
{
    public class PackageDiagramSelectHandler : SelectHandler
    {
        public override HandlerResponse MouseDoubleClick(ViewModel selectedViewModel, ViewModel senderViewModel, View
        {
            ViewModel modelToActivate = (senderViewModel is SimplifiedPackageDiagramViewModel) ? senderViewModel : s
            if (modelToActivate is PackageDiagramViewModel || modelToActivate is SimplifiedPackageDiagramViewModel)
            {
                Ria.RiaViewModel.Commands.ActivateViewCmd<ViewModel, PackageDiagramWindowModel> activateView = new M
                activateView.Execute(modelToActivate, containerViewModel);
            }
            return HandlerResponse.Handled;
        }
    }
}
    
```

41

Arbeitstage des Sprints



**Das fertige Feature**

42

# Lessons Learned

43

**Up-front-Problem** gelöst durch:

**Sprint 0**

**Discount-Methoden** des Usability Engineering

Kontinuierliches **Refactoring** der Usability

Product Owner und Usability Engineer müssen einheitliche Vision von Layout und Interaktionsarchitektur entwickeln

44

**Synchronisationsproblem** gelöst durch:

den „neuen“ **Workflow**

**Anleihen** beim Parallel Tracking

die **hohe Geschwindigkeit** der modellgetriebenen Softwareentwicklung

Product Owner und Usability Engineer müssen Entwicklern immer einen Schritt voraus sein

45

**Problem der Nutzerbeteiligung** noch nicht vollständig gelöst

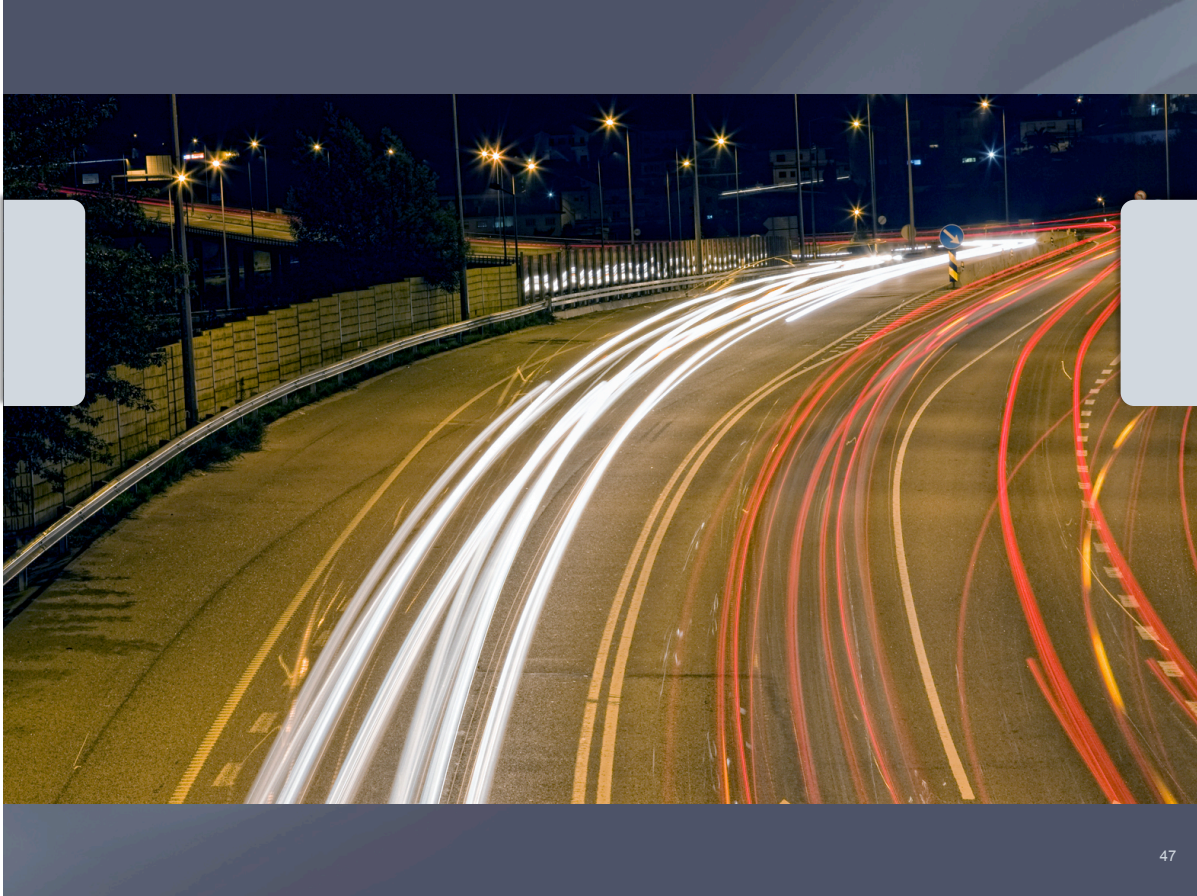
**Bisher:** Usability Engineer definiert, was der Endnutzer braucht

**Auf den Weg gebracht:** Einbeziehung von Endnutzern bei Kunden in Benutzungsanalyse und Usability Tests

Usability Engineer muss Anwalt der Endnutzer und Berater des Product Owners sein

46





47

**Vielen Dank**

[ursula.meseberg@microTOOL.de](mailto:ursula.meseberg@microTOOL.de)

48

### *3.3 Integration von Usability Engineering in den agilen Entwicklungsprozess nach Scrum*

## 4. Systems Engineering und Embedded Engineering

### Sessionüberblick

---

4.1. Domänenübergreifende Abhängigkeiten . . . . .	139
4.2. Erfahrungen von durchgängiger Entwicklung mittels SysML . . . . .	155
4.3. Integration von Software-Werkzeugen für die Entwicklung von Automatisierungssystemen mit dem Engineering Service Bus . . . . .	165
4.4. CESAR: Kosteneffizienz bei der Entwicklung sicherheitsrelevanter Echtzeitsysteme . .	177

---





## Domänenübergreifende Abhängigkeiten – Mehrwert durch Modellverknüpfung in der Systementwicklung

Rainer Stark<sup>1,2</sup>, Grischa Beier<sup>1</sup>, Asmus Figge<sup>2</sup>, Thomas Wöhler<sup>1</sup>

<sup>1</sup> Fraunhofer Institut für Produktionsanlagen und Konstruktionstechnik

<sup>2</sup> Technische Universität Berlin, Fachgebiet Industrielle Informationstechnik  
Pascalstraße 8-9, 10587 Berlin  
Rainer.Stark@ipk.fraunhofer.de

**Abstract:** Im Rahmen des BMBF-geförderten Projekts ISyProM wird ein Modellierungswerkzeug zur domänenübergreifenden Definition von Verknüpfungen zwischen Teilmodellen in der Produktentwicklung entwickelt. Beinhaltete Funktionalitäten unterstützen dabei die Änderungsverfolgung, ein proaktives Qualitätsmanagement und bedarfsorientierte Informationsreduktion.

### 1 Motivation

Die stetig steigende Komplexität von mechatronischen Produkten stellt für viele Firmen eine große Herausforderung dar. Zur Beherrschung der Komplexität ist es notwendig, bisher implizit bekannte Abhängigkeiten der Teilsysteme abzubilden. Zur Unterstützung des Entwicklers beim Entwurf komplexer Systeme, müssen Funktionalitäten für die Analyse, die Kontrolle und die entwicklungsbegleitende Definition der Abhängigkeiten einzelner Teilmodelle des Systems bereitgestellt werden. Ziel des Projekts ISyProM ist es, einen modellbasierten Prozess- und Systementwurf zu unterstützen. Der Schwerpunkt des vorgestellten Ansatzes liegt in der Integration und Verknüpfung der Teilmodelle, welche Anwendung im Entwicklungsprozess finden. Methodische Vorarbeiten wurden im BMBF-Leitprojekt „integrierte Virtuelle Produktentstehung“ (iViP), dem BMBF-Projekt MIKADO sowie durch Maurer geleistet [Ba02], [Fi08], [Ma07].

### 2 Modellierung domänenübergreifender Verknüpfungen

Der generelle Ansatz des hier beschriebenen Verknüpfungsmodellierers adressiert diese Herausforderungen. Er bietet Methoden und Funktionalitäten, mit denen es ermöglicht wird, domänenübergreifende Verknüpfungen zwischen den Elementen unterschiedlicher systembeschreibender Modelle zu erstellen (Abbildung 1). Um die notwendige Aktualität der Modelle zu garantieren, werden diese bei jedem Aufruf des Modellierers neu geladen und nur Verknüpfungsinformationen gespeichert. Da die Herausforderungen im Aufwand zur Erfassung sowie der Sicherstellung der Qualität der Verknüpfungsinformationen bestehen, werden zwei grundlegende Arten der Erstellung implementiert, die einen ersten Schritt in Richtung einer besseren Integration in den Gesamtprozess darstellen.

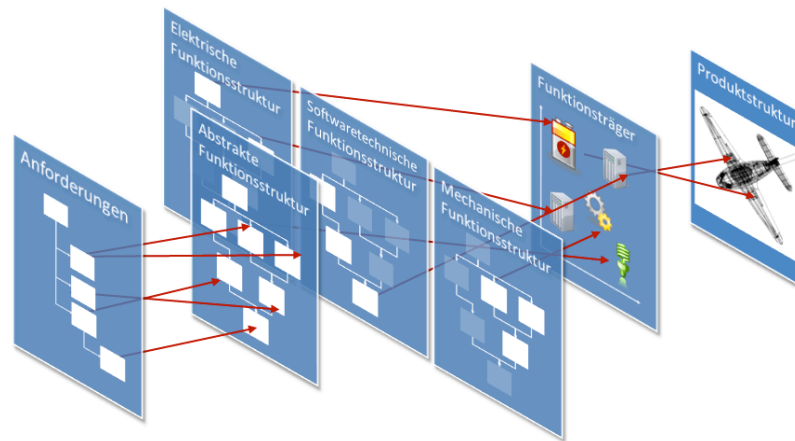


Abbildung 1: Verknüpfung der systembeschreibenden Teilmodelle

**Automatische Erstellung:** Der Verknüpfungsmodellierer stellt Elemente der Teilmodelle für die Erstellung neuer Modelle in der dafür genutzten Software zur Verfügung. Die Verknüpfungen zwischen neuem Modell und Teilmodell werden dabei automatisch erstellt. **Manuelle Erstellung:** Die manuelle Definition von Verknüpfungen wird mit einem Bewertungssystem kombiniert, mit dem eine Gewichtung vorgenommen wird. Somit kann eine Aussage über die Korrektheit einer Verknüpfung getroffen werden.

Durch diese Form der entwicklungsbegleitenden Verknüpfung der Teilmodelle gelingt es, einen durchgängigen Zusammenhang bspw. zwischen Anforderungen und Komponentenstruktur herzustellen. Speziell bei Änderungen birgt dieser große Vorteile, da domänenübergreifend die Auswirkungen auf direkt und indirekt verknüpfte Elemente analysiert werden. Auf Basis der Verknüpfungsinformationen können vom Nutzer einfache Constraints und Regeln definiert werden, um ein **proaktives Qualitätsmanagement** zu unterstützen. Ebenfalls soll bei der Erstellung neuer und der Überprüfung bestehender Verknüpfungen Werkzeugunterstützung angeboten werden. Zur Effizienzsteigerung in der Systementwicklung werden Methoden entwickelt, um eine **bedarfsorientierte Informationsreduktion** zu realisieren, die dem Entwickler lediglich Informationen bereitstellt, die für seinen Arbeitskontext Relevanz besitzen.

### 3 Literaturverzeichnis

- [Ba02] Baumann, R.; Kaufmann, U.; Leemhuis, H.; Kühn, T.; Ragan, Z.; Swoboda, F.: Funktionsorientiertes Entwerfen. In (Krause, F.-L.; Tang, T.; Ahle, U. Hrsg.): Integrierte Virtuelle Produktentstehung – Abschlussbericht, Stuttgart. Fraunhofer IRB-Verlag, 2002; S. 92-101.
- [Fi08] Figge, A.; Hajtmar, M.; Hegewald, T.; Rothenburg, U.; Swoboda, F.: Integrierte funktionale Validierung mechatronischer Produkte mit FMU. In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb, Jg. 103, H. 10, 2008; S. 726-731.
- [Ma07] Maurer, M.: Structural Awareness in Complex Product Design. In Produktentwicklung München Band 56, München: Dr. Hut Verlag, 2007.



## Domänenübergreifende Abhängigkeiten

Mehrwert durch Modellverknüpfung in der Systementwicklung



3. Mai 2010 Grischa Beier, Asmus Figge



## Übersicht



- **ISYPROM – Integrierte System- und Prozessmodellierung**

Herausforderungen

Modellübergreifende Verknüpfung produktbeschreibender Daten

Funktionalitäten des Verknüpfungsmodellierers



2

## ISYPROM - Integrierte System- und Prozessmodellierung

<b>Endanwender</b>			
<b>Software/ Consulting</b>			
<b>Forschung</b>			

<p><b>GEFÖRDERT VOM</b></p>  <p>Bundesministerium für Bildung und Forschung</p> <p>Förderkennzeichen 02PC105x</p>	<p><b>BETREUT VOM</b></p>  <p>PTKA Projektträger Karlsruhe im Karlsruher Institut für Technologie</p>
--	--

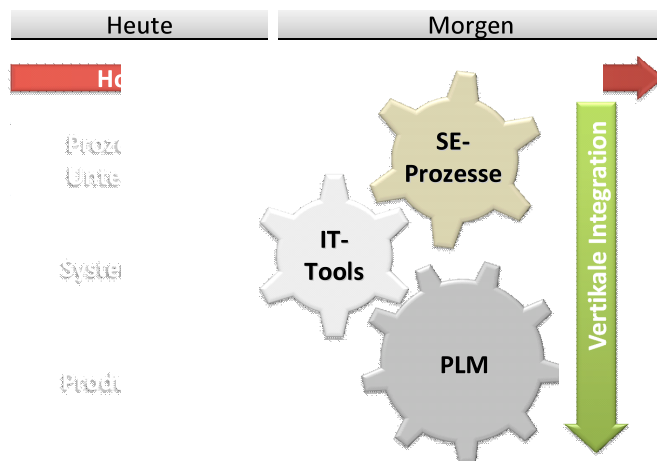
**Kennzahlen**

- Projektzeitraum: 01.11.2008 – 31.04.2011 (2,5 Jahre)
- Budget: 4,6 Mio € (474 PM)
- Fördersumme: 2,5 Mio €



3

## ISYPROM – Zielvorstellung und Arbeitspakete



Arbeitspakete:

Kontextuelle System- und Prozessmodellierung

PLM-Integration

Optimiertes Innovationsmanagement



## Übersicht

---



ISYPROM – Integrierte System- und Prozessmodellierung

- **Herausforderungen**

Modellübergreifende Verknüpfung produktbeschreibender Daten

Funktionalitäten des Verknüpfungsmodellierers

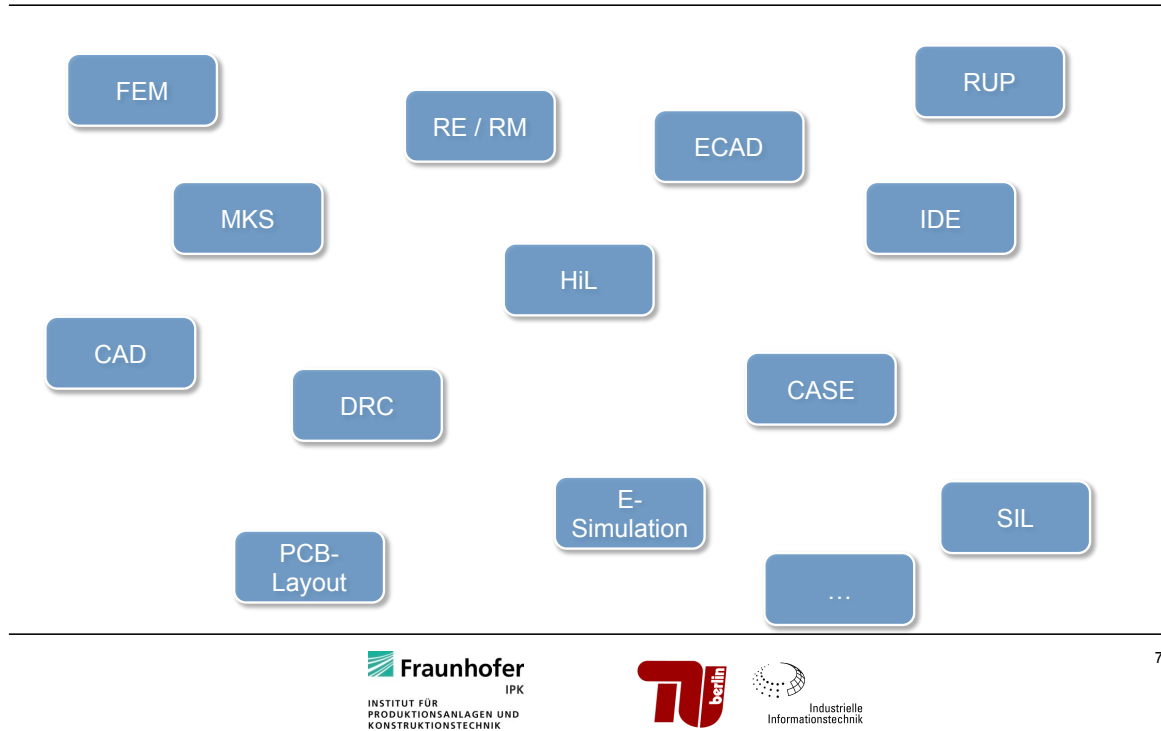
## Komplexe Systeme

---



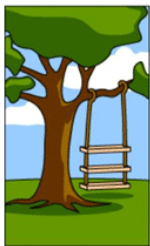
[www.jally.de/pages/wallpaper/wallpaper\\_kunst.php](http://www.jally.de/pages/wallpaper/wallpaper_kunst.php)  
[http://de.academic.ru/pictures/dewiki/83/S-Bahn\\_Berlin\\_Baureihe\\_481.jpg](http://de.academic.ru/pictures/dewiki/83/S-Bahn_Berlin_Baureihe_481.jpg)

## Spezialisierte Werkzeuge

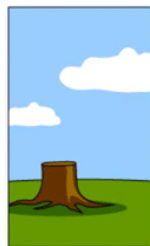


## Auswirkungen von Änderungen

Definition von Anforderungen



Änderung der Randbedingungen



Änderungen wurden nicht kommuniziert



[http://www.uidesign.at/wp-content/uploads/2007/05/treeswing\\_enlarged.gif](http://www.uidesign.at/wp-content/uploads/2007/05/treeswing_enlarged.gif)



## Datenaustausch

---



## Übersicht

---



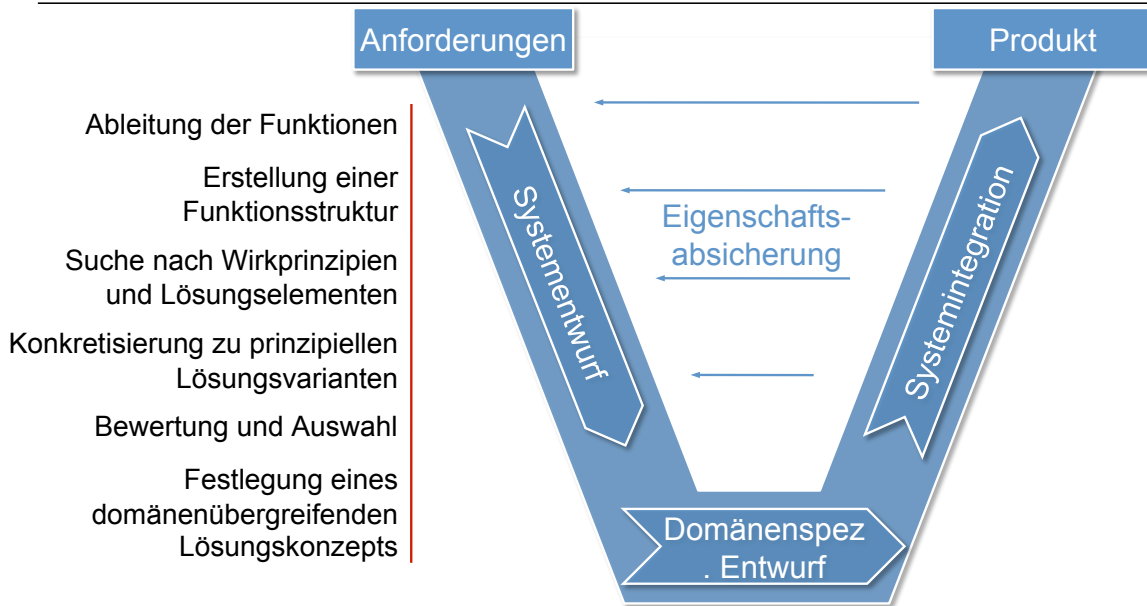
ISYPROM – Integrierte System- und Prozessmodellierung

Herausforderungen

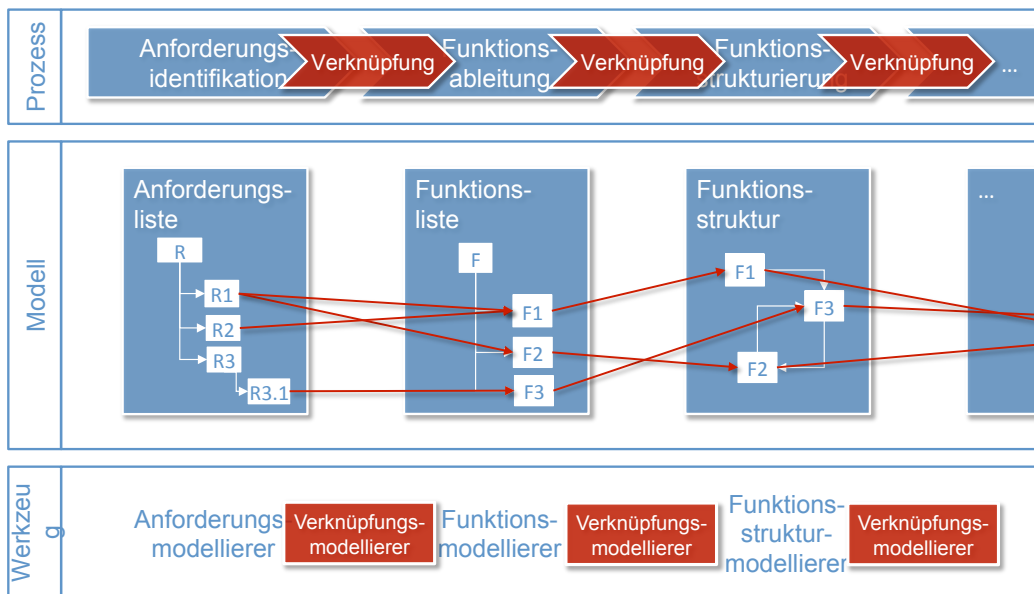
- **Modellübergreifende Verknüpfung produktbeschreibender Daten**

Funktionalitäten des Verknüpfungsmodellierers

## Der Entwicklungsprozess nach VDI 2206



## Konzept der modellübergreifenden Verknüpfungen



## Übersicht



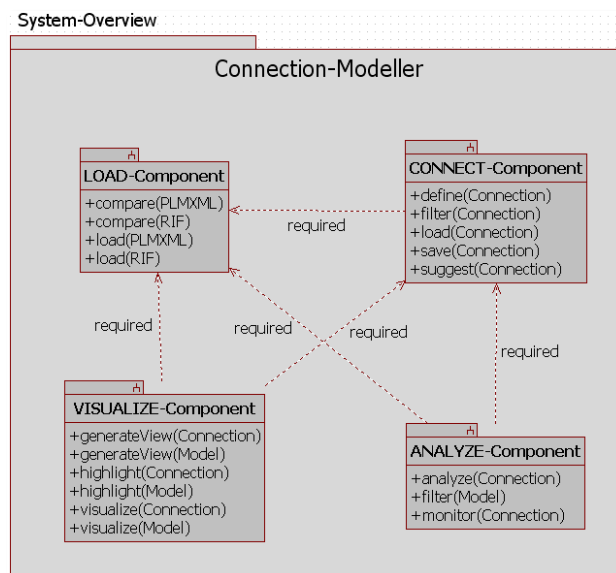
ISYPROM – Integrierte System- und Prozessmodellierung

Herausforderungen

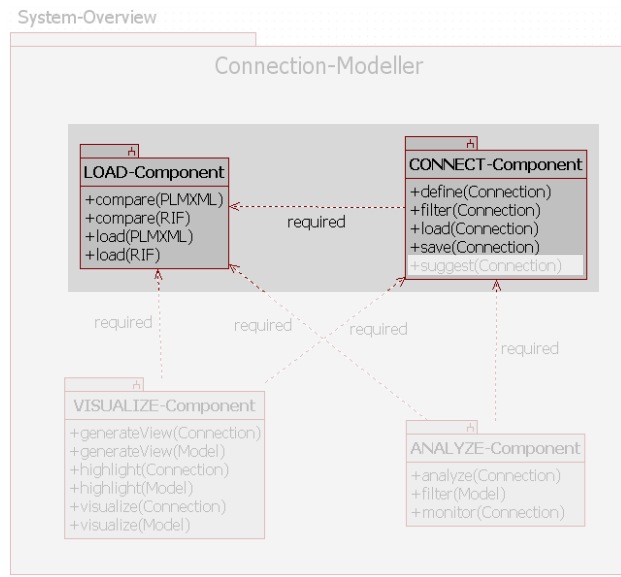
Modellübergreifende Verknüpfung produktbeschreibender Daten

- **Funktionalitäten des Verknüpfungsmodellierers**

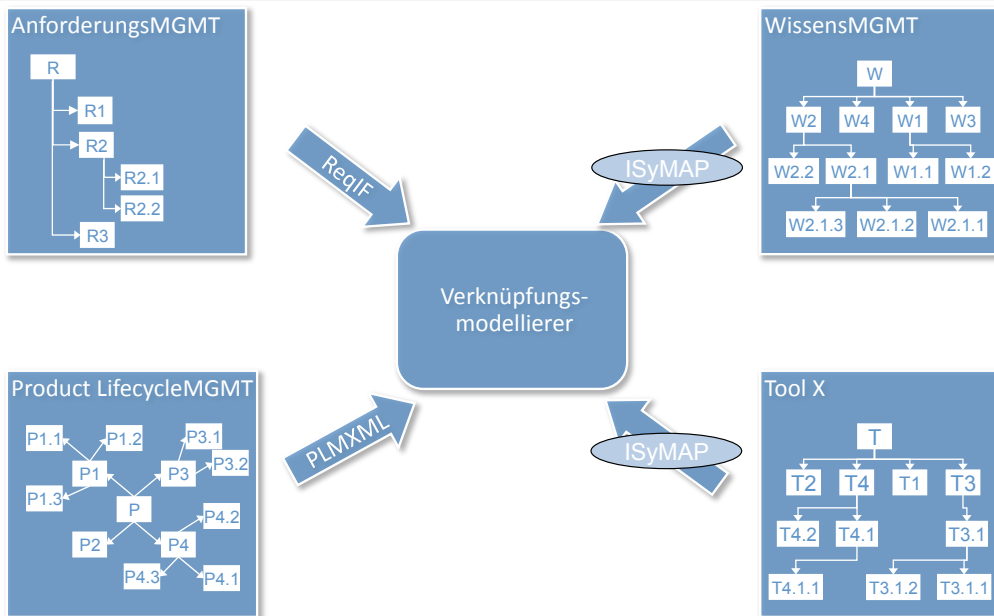
## Komponenten des Verknüpfungsmodellierers



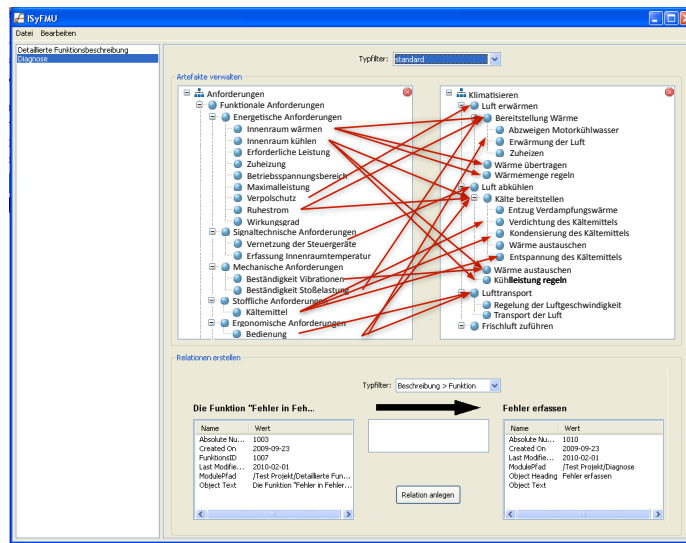
## Komponenten des Verknüpfungsmodellierers



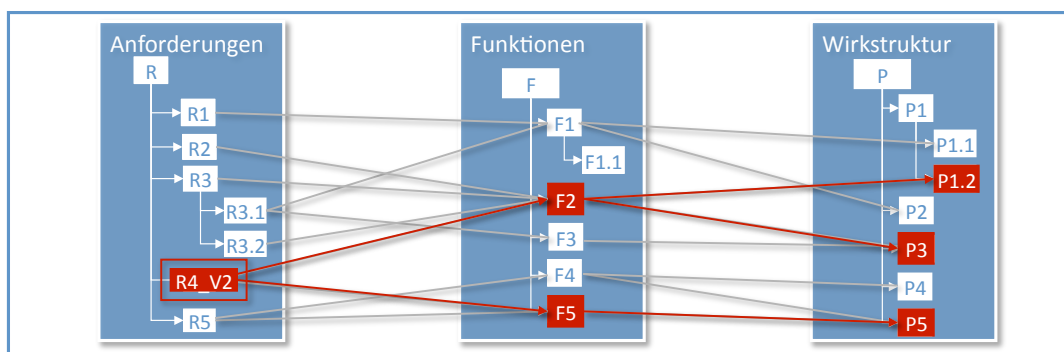
## Weiterverwendung der bestehenden Tool-Landschaft



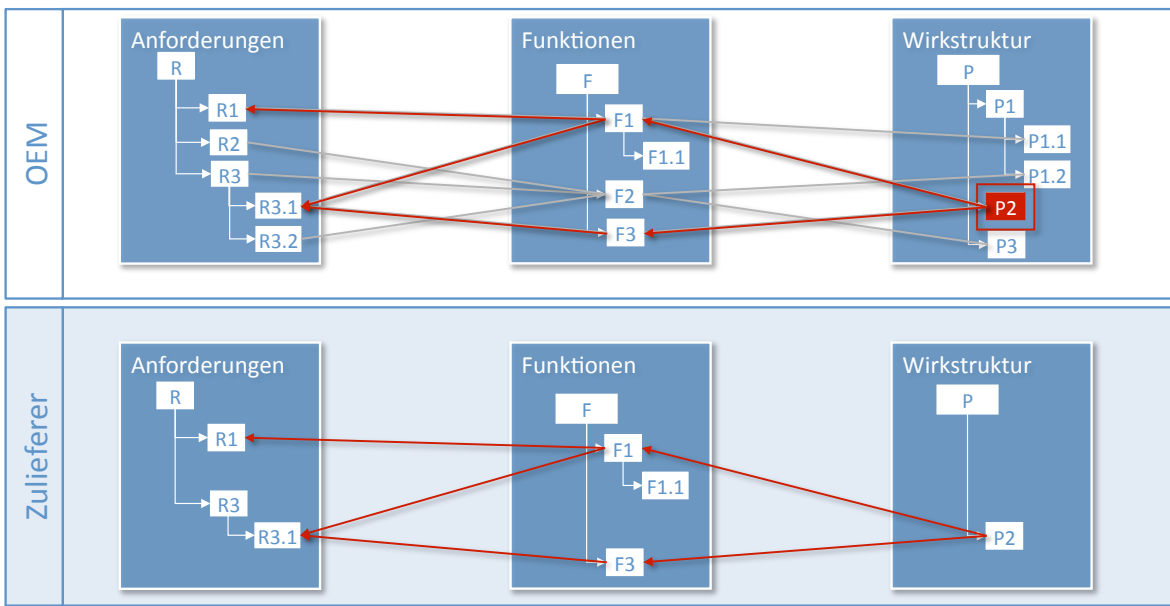
## Modellübergreifende Verknüpfung



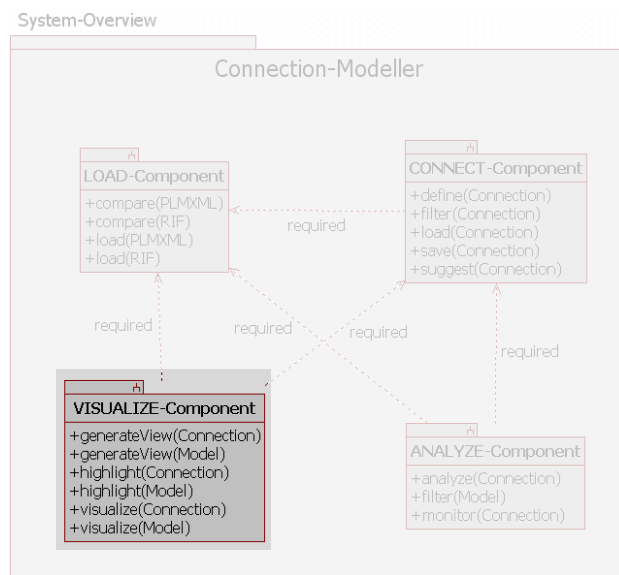
## Verbesserte Konsistenz der Partialmodelle



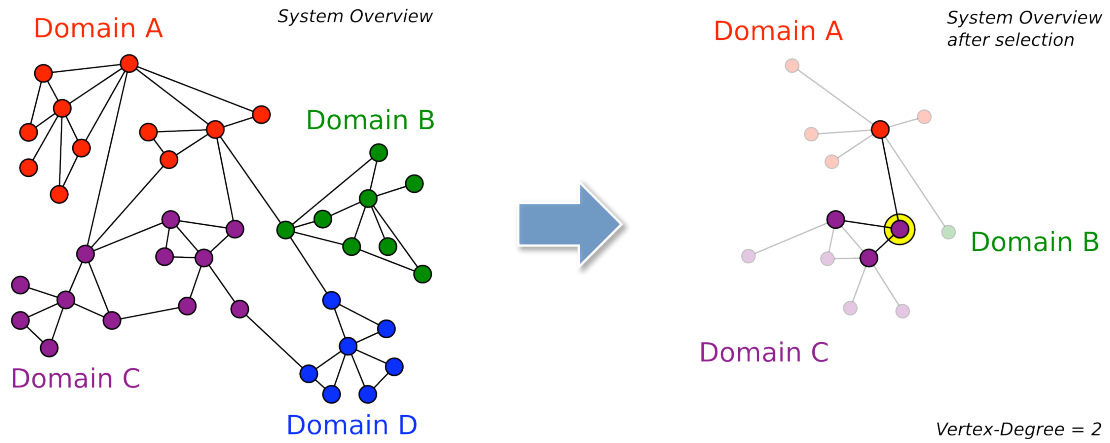
### OEM-Zulieferer-Kommunikation



### Komponenten des Verknüpfungsmodellierers

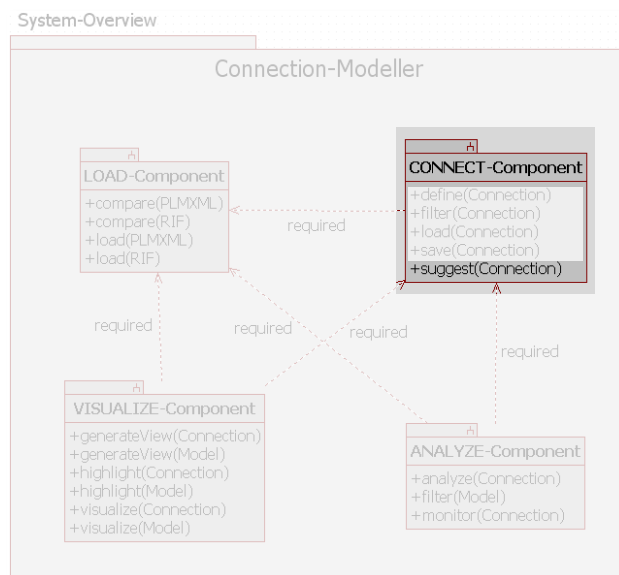


## Verbesserte Übersichtlichkeit durch Sichtenbildung



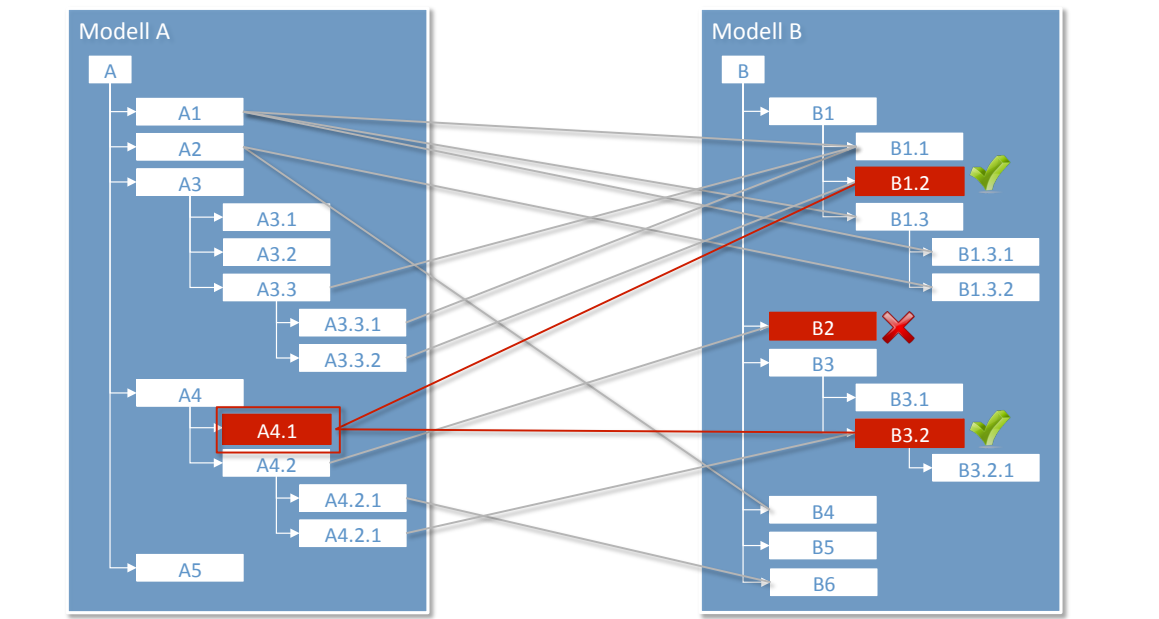
Sichtenbildung erfolgt auf Basis der Verknüpfungsinformationen

## Komponenten des Verknüpfungsmodellierers

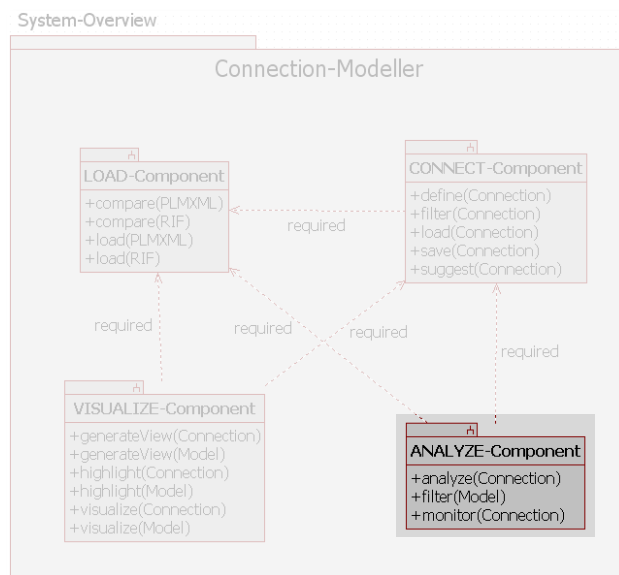




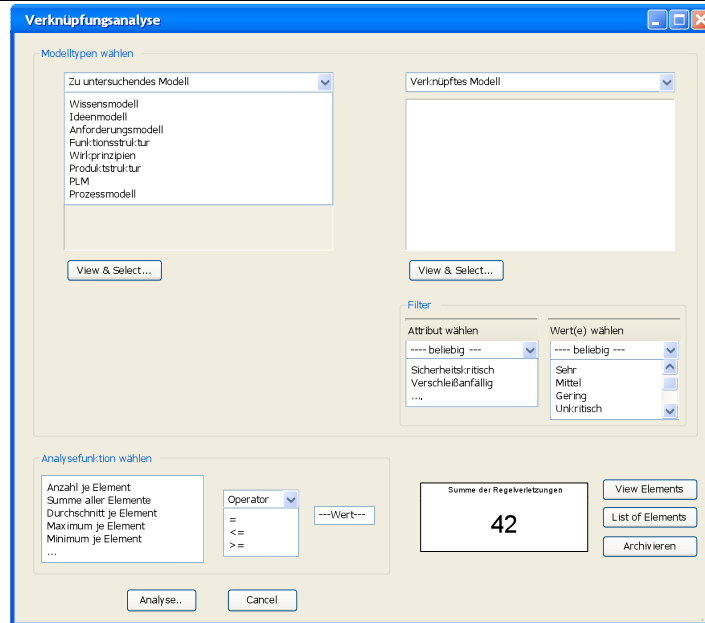
## Unterstützte Erstellung



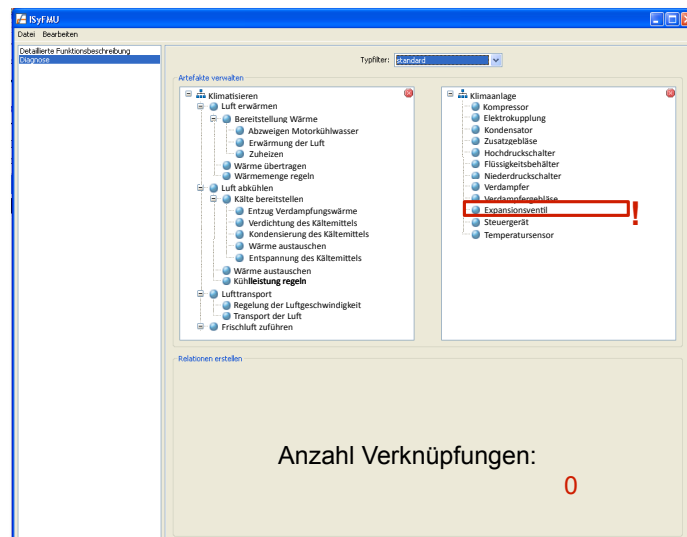
## Komponenten des Verknüpfungsmodellierers



## Unterstützte Analyse



## Vollständigkeitsanalyse



Defense Acquisition  
University Press (2001):  
Systems Engineering  
Fundamentals. Defense  
Acquisition University  
Press. Fort Belvoir,  
Virginia.

“Each part must meet at least one functional requirement...”

## Produktionstechnisches Zentrum Berlin

---

Vielen Dank für Ihre  
Aufmerksamkeit

---



## Durchgängige Entwicklung mit dem SysML Standard

Sven-Olaf Schulze<sup>1</sup>, Rodrigo Batista<sup>2</sup>, André Brückmann<sup>3</sup>

Berner & Mattner Systemtechnik GmbH  
Hein-Saß-Weg 38  
21129 Hamburg  
sven-olaf.schulze@berner-mattner.com  
rodrigo.batista@berner-mattner.com  
andre.brueckmann@berner-mattner.com

**Abstract:** Anhand eines Beispiels aus der Luftfahrt sollen die Erfahrungen unter Nutzung des SysML Standards durch ein Entwicklungswerkzeug, das mit einem Anforderungsmanagementwerkzeugs verknüpft ist, aufgezeigt werden. Es werden am Anfang die Projektanforderungen und Aufgaben des Projektes dargestellt, die mittels einer durchgängigen Werkzeugkette und geeigneten Methoden und Prozessen an einem physikalischen System umgesetzt und gelöst wurden. Durch eine verknüpfte und frühe Nutzung der fachübergreifenden Anforderungen, der Architektur und einer parallelen Modellbildung werden die Konsistenzprüfung, die Rückverfolgbarkeit, die Vollständigkeit und damit die Qualität des Entwicklungsprozesses verbessert. Einhergehend wird durch den modellbasierten Ansatz und der Durchgängigkeit der Informationen die Suche nach der bestmöglichen globalen Lösung unterstützt und weitere Effekte lassen nutzen.

### 1 Motivation

Bei der Entwicklung von Systemen nehmen branchenunabhängig die Umweltaforderungen zu und die Suche nach Ersatz für die auslaufenden Energieressourcen nimmt für Neuentwicklungen und deren Lösungen einen höheren Stellenwert ein. Dieses ist auch in dem betrachteten Projekt der Fall. Einhergehend müssen neue und innovative Technologien das Geschäftsmodell nachweisen und die Wirtschaftlichkeit aufzeigen, um realisiert zu werden. Viele dieser Technologien können diesen Nachweis nur durch eine ganzheitliche und fachübergreifende Betrachtung erbringen. Ein Tausch von Alt- gegen Neu-System innerhalb der Systemgrenzen ist nicht ohne weiteres möglich und führt dadurch zu neuen Grenzen, Ansätzen und Auslegungen. In diesem praktischen Beispiel ist es das Ziel, das Geschäftsmodell unter Berücksichtigung aller verteilten Disziplinen und einer globalen Auslegung von den Anforderungen bis zum Test aufzuzeigen und mit Hilfe von „Vollständigkeit durch Konstruktion“ zu ermöglichen. In dem Entwicklungsprozess sollen rückverfolgbar verschiedene Architekturen betrachtet und auf Basis von technischen und wirtschaftlichen Faktoren bewertet, verbessert und optimiert werden. Dieses immer unter Einhaltung des vorgegebenen Budgets und den Meilensteinen eines internen Kunden. Das Team soll eine Brennstoffzelle als Stromerzeuger in ein Flugzeug integrieren und den formalen Nachweis der Technologiereife erbringen.

## 2 Informationen zu SysML und dem Standard

Die Begriffe „Modelle“ und „systematisches Vorgehen“ bei der Entwicklung und dem Betrieb von Systemen sind branchenübergreifend präsent. Ingenieure arbeiten täglich mit Werkzeugen, um die Lösung für ihre Kunden sichtbar und greifbar zu machen. Bei der Entwicklung von Software werden auch Begriffe wie z.B. Blockdiagramme verwendet, die die Visualisierung von strukturiertem Vorgehen beschreiben und Zusammenhänge und Abhängigkeiten aufzeigen. Aus diesem Ansatz für die Softwareentwicklung hat sich eine neue Sprache entwickelt, die es ermöglicht auch einen physikalischen modellbasierten Ansatz zu beschreiben. Die Möglichkeiten dieser genormte Notation Systems Modeling Language [SysML] erlauben dabei sowohl die Beschreibung von physikalischen Systemen (Hardware) als auch von Software. Daten- und Materialflüsse lassen sich damit abbilden und interdisziplinär nutzen. Durch Verknüpfung der Modelle mit den Anforderungen lässt sich eine durchgängige Werkzeugkette herstellen und für die Designphase nutzen.

## 3 Das Projekt und seine Umsetzung

Das Projektteam besteht aus 5 verteilten Disziplinen an 3 verschiedenen Standorten, die unter Nutzung von IBM DOORS [REQ] die Anforderungen verwalten und die Entwicklung der Modelle in IBM Rhapsody [RHY] mittels SysML abbilden. Durch Nutzung dieser und weiterer in den Fachteams bereits genutzten Werkzeuge soll mittels modellbasierten Ansatzes das Gesamtsystem nachgewiesen, realisiert und optimiert werden.

Auf Grund der neuen Technologie ist ein 1:1 Austausch mit dem bestehenden System nicht möglich, denn die Systemgrenzen müssen aus verschiedenen Gründen angepasst werden. Es gibt keine Erfahrungen auf dem Gebiet der Brennstoffzelle im Flugzeugbau, die Leistung der am marktbehaftlichen Brennstoffzellen ist nicht optimal und weitere Aspekte müssen in die Betrachtung und damit in das System aufgenommen werden. Außerdem sind die Aspekte Sicherheit und Logistik an den Flughäfen mit zu berücksichtigen. Weiterhin sind die klassischen Designtreiber wie Gewicht, Zuverlässigkeit und Wartbarkeit im Flugzeugbau nachzuweisen. Nur durch die integrierte Betrachtung der möglichen Missionen von Flugzeugen und der Nutzung der Abfallprodukte einer Brennstoffzelle lässt sich ganzheitlich das System optimieren. Diese Abfallprodukte sind z.B. das erzeugte Wasser als auch der abgereicherte Sauerstoff, der als Inertgas im Flugzeug verwendet werden kann, und damit als positive Effekte in die Betrachtung einfließen können.

Die bestmögliche Architektur des physikalischen Systems, der Brennstoffzelle, wurde aus den Anforderungen und der Funktionsanalyse entwickelt und mit den Modellen verifiziert. Die Modelle aus dem genutztem Werkzeug sind mit den einzelnen Anforderungen verlinkt. Änderungen der Anforderungen oder der Architektur können damit direkt erkannt und deren Einfluss ganzheitlich und global analysiert werden. Diese Vorgehensweise hat zur Folge, dass in einem frühen Stadium die lokalen Annahmen der einzelnen Fachabteilungen sofort überprüft werden können und nicht erst bei der Integration der gefertigten Teilkomponenten verbliebene Inkonsistenzen aufgedeckt werden.

Durch die Verwendung von Rhapsody als Modellierungswerkzeug ist eine dynamische Simulation aus dem Modell heraus möglich und wird auch genutzt. Insgesamt gibt es 200 variable Eingangsparameter in den verschiedenen fachlichen Sichten, die betrachtet und verbessert werden können. Damit werden mehr als 1000 Parameter im Modell gesteuert, die in einem entsprechenden Bericht für jede Sicht (Disziplin) angepasst und ausgegeben werden können. Durch die Nutzung der verschiedenen SysML Diagramme lassen sich damit Masseströme, Informationen, Regelungen und elektrische Abhängigkeiten im Modell abbilden. Ein weiterer positiver Aspekt ist es, dass durch die Verknüpfung der Anforderungen mit den Modellen und den darin enthaltenen Modellbeschreibungen sich zusätzlich auch rückverfolgbar und kontrolliert automatisch die Spezifikationen und die Modellbeschreibungen der verschiedenen Varianten und Architekturen erstellen lassen. Die globale Optimierung und Auslegung erfolgt ganzheitlich mit Hilfe der ausführbaren Modelle in den Fachabteilungen. Damit wird sichergestellt, dass alle an einer Lösung arbeiten und die Komplexität beherrschbar wird.

### 4 Zusammenfassung und Ausblick

Durch die konsequente Nutzung eines modellbasierten Werkzeugs und der SysML lassen sich auch die Anforderungen mit den Modellen verknüpfen und damit verifizieren. Für die Analyse der unterschiedlichsten Aufgabenstellungen können schnell die Zusammenhänge zwischen Architektur und Systemverhalten für die gewählte Lösung aufgezeigt werden. Nutzt man auch die Code Generierung der Werkzeuge, so eröffnet sich die Gelegenheit in einer frühen Phase die Spezifikation durch Simulation zu testen. Durch die Verknüpfung von Anforderungen und Modellen inklusive der Modellbeschreibungen, lassen sich hieraus rückverfolgbar und aktuell einige Dokumente automatisch generieren und für das Projekt nutzen. Das Modell und die SysML dienen als Integrator für eine fachübergreifende und durchgängige Werkzeugkette von den Anforderungen bis zum Test. Die Testanbindung wurde nachgewiesen und wird in 2010 in einem Forschungsprojekt im Detail realisiert.

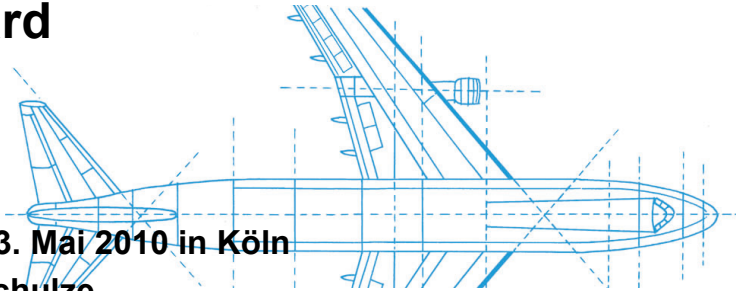
### Literaturverzeichnis

- [SysML] OMG System Modeling Language™ Version 1.1; Internet <http://omgsysml.org>
- [REQ] IBM Rational DOORS Software; Internet <http://www-142.ibm.com/software/products/de/de/ratidoor>
- [RHY] IBM Rational Rhapsody Designer for Systems Engineers; <http://www-142.ibm.com/software/products/de/de/ratirhapdesiforsystemgi>



# Durchgängige Entwicklung mit dem SysML Standard

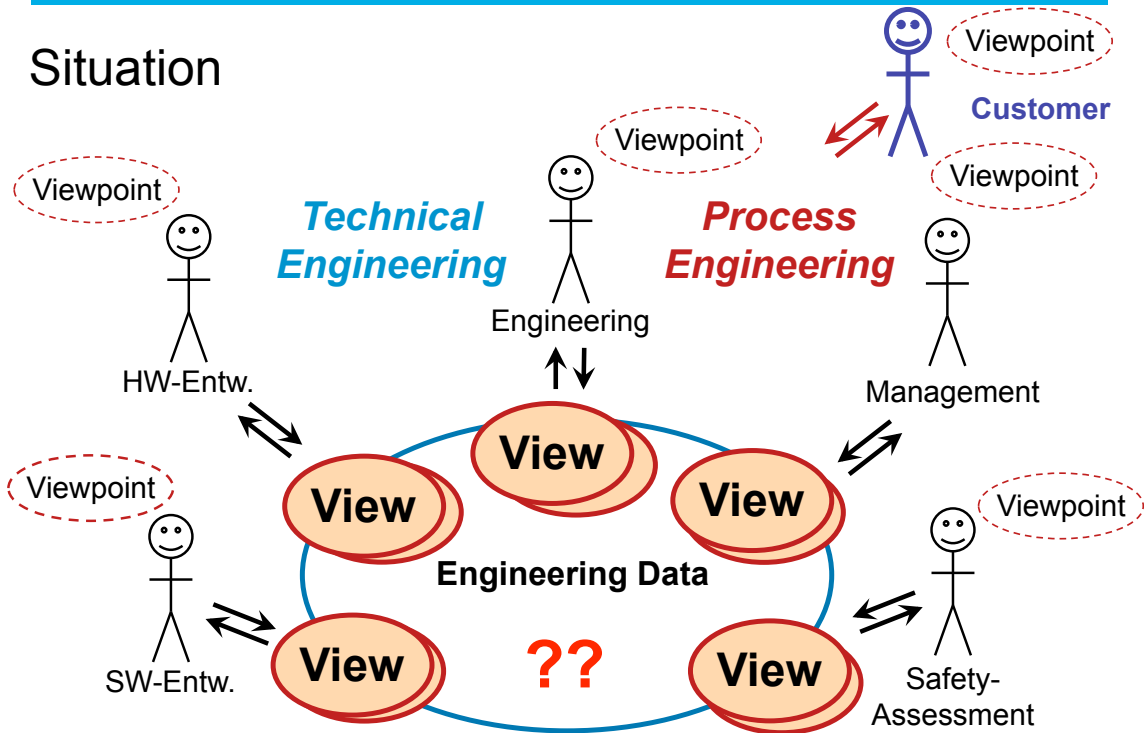
SEE2010 Konferenz, 3. Mai 2010 in Köln  
 Dipl.-Ing. Sven-Olaf Schulze  
 Teamleiter Aeronautics



1



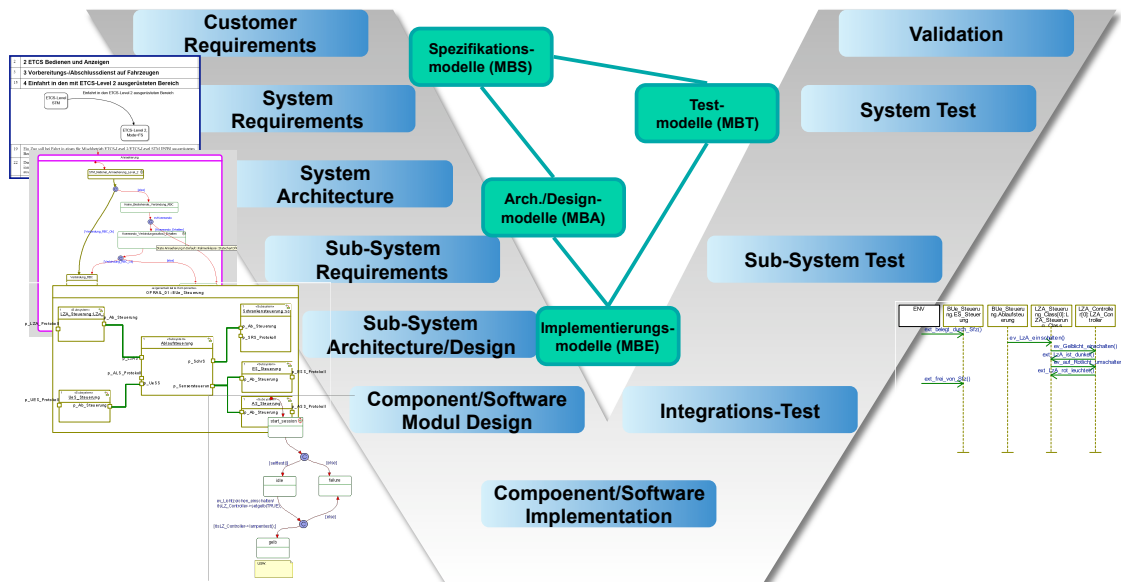
## Situation



2



## Nutzung der Durchgängigkeit



3



## Spezifikationsmodelle

- ▶ System wird in Aufbau und Verhalten grafisch abgebildet
  - UML für reine Softwaresysteme
  - SysML für reine Hardware oder gemischte Systeme
- ▶ Umgebung wird ebenfalls modelliert
  - Verhalten der Umgebung in Bezug aufs System
  - Schnittstellen mit dem System
- ▶ Das Modell wird im Laufe der Entwicklung immer weiter verfeinert
  - Elemente entstehen
  - Schnittstellen werden genauer spezifiziert
- ▶ Spezifikationsdokumente
  - werden aus dem Modell generiert
  - oder sie sind mit dem Modell verbunden/verlinkt

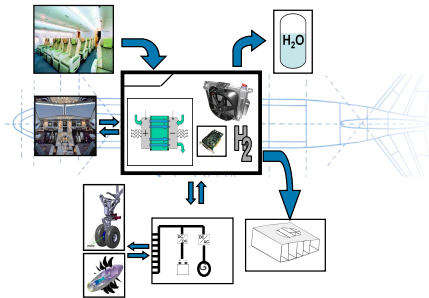
4





## Projektbeispiel Luftfahrt: Gründe für SysML

- Einführung eines modellbasierten Ansatzes
  - „Vollständigkeit durch Konstruktion“
  - Ableiten von Anforderungen aus dem Modell
- Modellierung des Brennstoffzellensystems und der relevanten Peripherie
- Nutzung des Modells für Simulation zur Unterstützung der „Trades“ in Designphase
- Nutzung des Modells zur Emulation mit Anbindung an den Teststand
- Unterstützung des formalen Nachweises der Technologiereife / Business Case



5



## Ziel des modellbasierten Ansatzes mit SysML

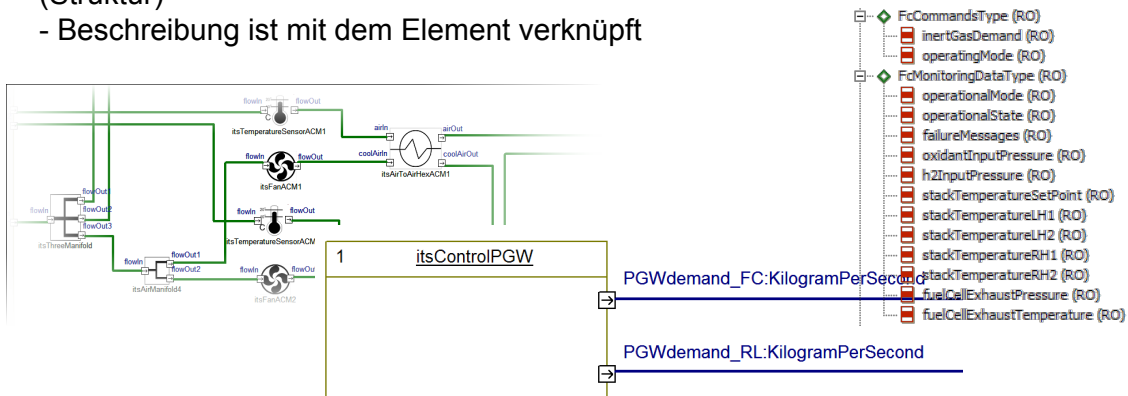
- Fachübergreifende Zusammenarbeit in der frühen Phase zur schnelleren Identifikation von
  - Inkonsistenzen
  - Widersprechenden Anforderungen
  - Lücken in der Analyse
  - Optimierte Schnittstelle (fachlich und technisch)
- Handhabung und Visualisierung von komplexen Zusammenhängen
  - Unterstützung der Entwicklung
- Eine kontrollierte Datenbasis für den Entwicklungsprozess
  - Automatisierte Lastenheftgenerierung mit Rückverfolgbarkeit
- Wiederverwendung von Testfällen (SiL, MiL, HiL)
- Dynamische Simulation des Gesamtsystems
- Die Modelle als Integrator für eine durchgängige Toolkette von den Anforderungen bis zum Test nutzen

6



## Modellierung mit SysML

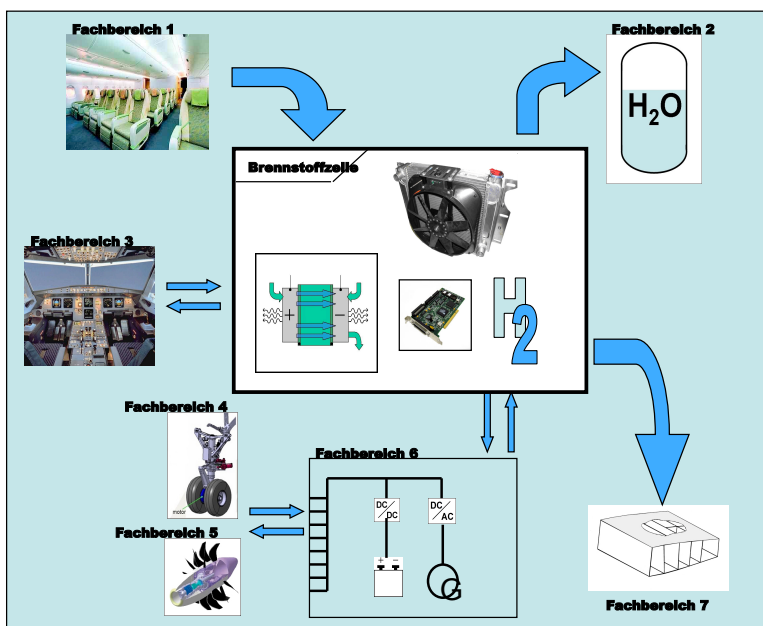
- SysML ist ein Standard, der inzwischen weltweit genutzt wird
- Nutzung von modularen Strukturen (parametrisierte Modelle)
- Verbindung zwischen Aktivitäten (Funktionen) und Blöcken (Lösungen) erlaubt ein kontrolliertes Änderungsmanagement
- Schnittstellen sind spezifiziert, z.B. Datentyp (SI-Einheiten), Datensprache (Struktur)
- Beschreibung ist mit dem Element verknüpft



7



## Interaktion der eingebundenen Modelle (techn.-Sicht)

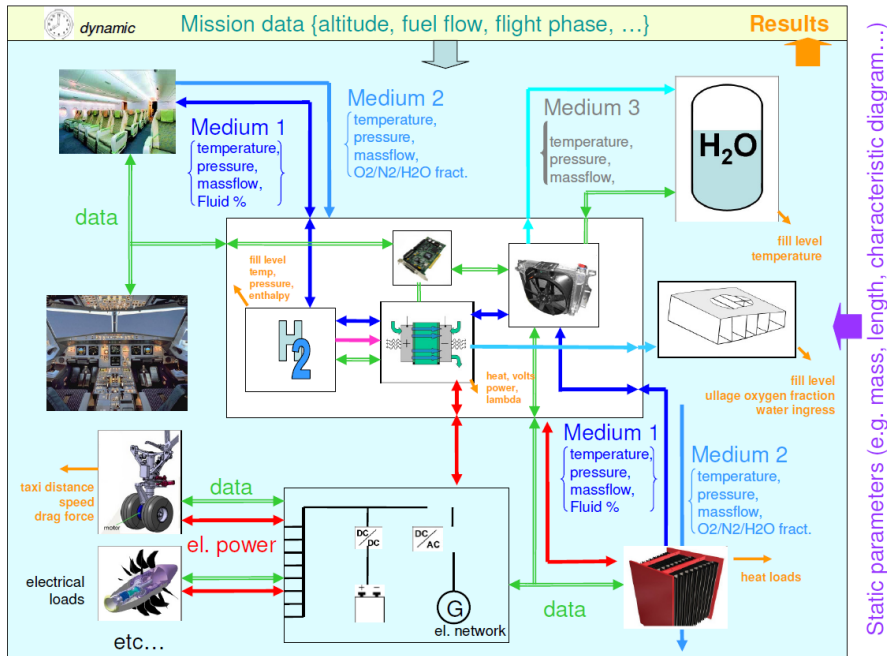


- Ziel:
  - Ersatz des Stromgenerators durch eine Brennstoffzelle
- Lokale versus globale Optimierung
- Berücksichtigt:
  - Masseströme
  - Elektrik
  - Schaltungen
- Auslegungsparameter
  - Gewicht
  - Kosten
  - ....
- Konsistenzprüfung zwischen Anforderungen, Funktion und Schnittstellen
- Qualität der Anforderungen erhöht

8



## Die Parameter-Sicht (ca. 1000 im dynam. Modell)



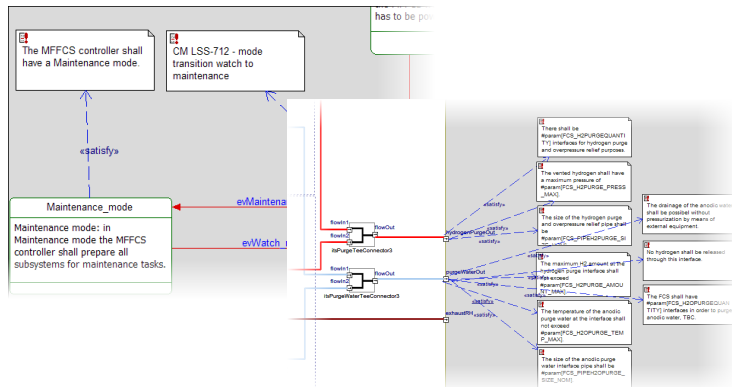
9



## Schnittstelle Rhapsody zu DOORS

Verknüpfung von Anforderungen und Diagrammen:

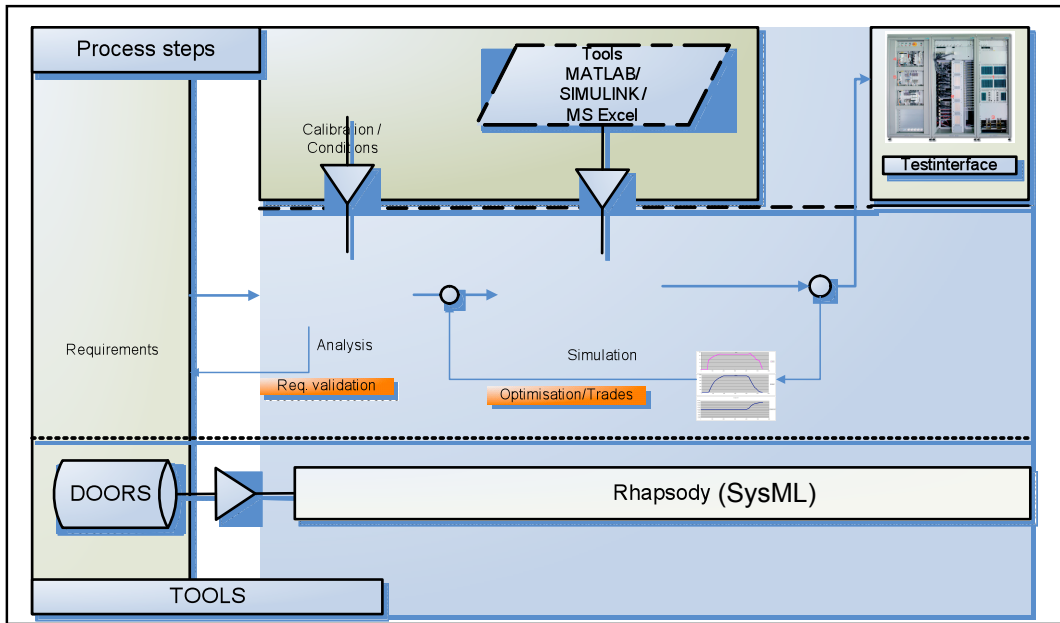
- Functional Requirement      <-> <satisfy>      Activity
- Interface Requirement      <-> <satisfy>      Port
- States/Modes      <-> <satisfy>      State chart (State or Transition)
- Component/System Req.      <-> <satisfy>      Block



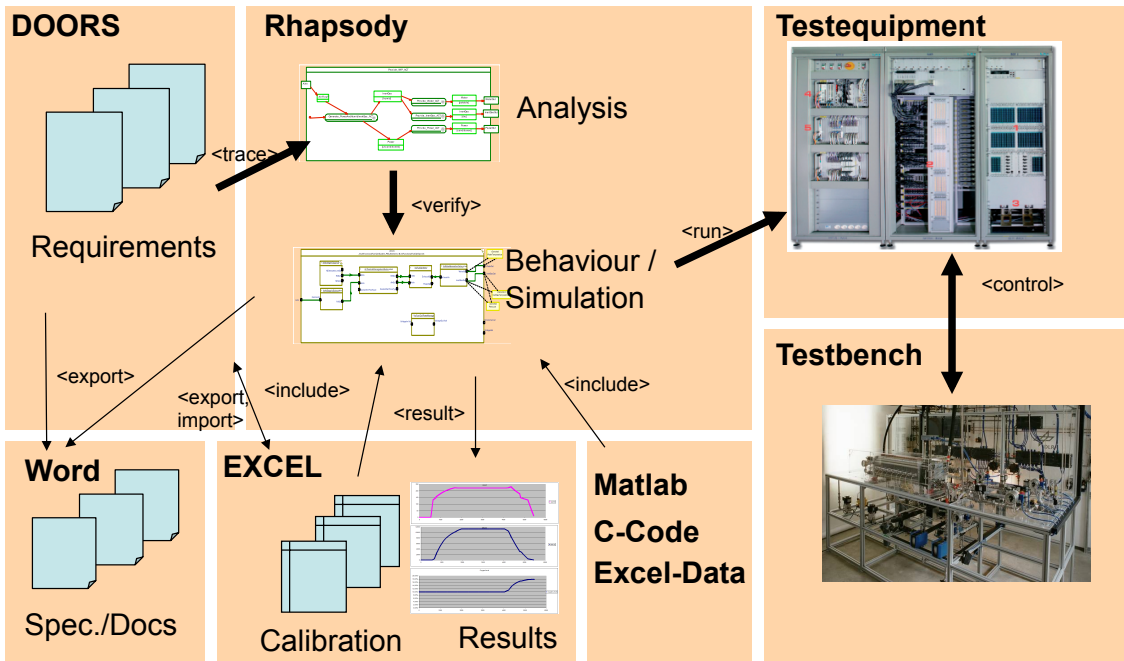
10



## Exemplarischer Prozess



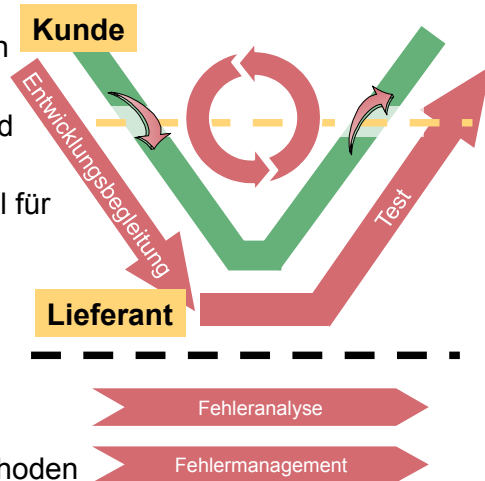
## Software Beziehungen





## Durchgängiger modellbasierter Prozess - Nutzen

- ▶ Zuverlässige Absicherung der Daten und Informationen über die Kunden / Lieferanten Schnittstelle hinaus
- ▶ Kostensenkung durch frühe Aufdeckung und Behebung von Fehlern
- ▶ Modellbasierte Methoden sind ein Schlüssel für Modularisierung von Systemen (Schnittstellentests an den Systemgrenzen)
- ▶ Durchgängige Verwendung der Testfälle; bessere Wiederverwendbarkeit und Anpassbarkeit gegenüber skriptbasierten Testverfahren
- ▶ Modellbasierte Spezifikations- und Testmethoden mit UML/SysML liefern hierzu einen entscheidenden Beitrag



13



## Kontaktinformationen



**berner & mattner**  
optimizing your development

Berner & Mattner Systemtechnik GmbH

Dipl.-Ing.  
**Sven-Olaf Schulze**  
Teamleiter  
Aeronautics

Hein-Saß-Weg 38  
D-21129 Hamburg  
Tel. +49 172 8104006

svn-olaf.schulze@berner-mattner.com  
www.berner-mattner.com

14

## Integration von Software-Werkzeugen für die Entwicklung von Automatisierungssystemen mit dem „Engineering Service Bus“

Dietmar Winkler, Stefan Biffli

Christian Doppler Labor „Software Engineering Integration  
für flexible Automatisierungssysteme“

Technische Universität Wien, Institut für Softwaretechnik und Interaktive Systeme  
Favoritenstrasse 9/188, A-1040 Wien.  
{Dietmar.Winkler, Stefan.Biffli}@tuwien.ac.at

**Abstract.** Automatisierungssysteme, wie Fertigungseinrichtungen, Industrieanlagen oder Kraftwerke, beinhalten komplexe Software-Komponenten zur Steuerung verteilter Anwendungen (*software-intensive systems*). Die zunehmende Komplexität und Flexibilität in der modernen Systemherstellung erfordern einerseits eine gute Einbindung von spezifischen Werkzeugen und andererseits die systematische Integration dieser Werkzeuge in den Entwicklungsprozess zur effizienten und effektiven Zusammenarbeit unterschiedlicher und heterogener Fachbereiche, wie etwa Elektrik, Mechanik und Software.

Etablierte Methoden und Werkzeuge mit individuell hohem Nutzen sind primär auf den jeweiligen Fachbereich zugeschnitten. Die Zusammenarbeit zwischen unterschiedlichen Fachbereichen wird allerdings durch inkompatible Methoden und Werkzeuge, Medienbrüche und verschiedene Projektsichten der beteiligten Rollen erschwert. In der industriellen Praxis sind daher vielfach individuell zugeschnittene und werkzeugabhängige Integrationsansätze (meist *Point-to-Point* Schnittstellen) zu finden. Nachteile dieser individuellen Integration sind aber meist eine eingeschränkte Flexibilität gegenüber Werkzeugänderungen, hoher Mehraufwand bei der Integration neuer Werkzeuge und eine eingeschränkte Wiederverwendbarkeit dieser punktuellen Integrationskonzepte. Eine integrierte Betrachtung des gesamten Systementwicklungsprozesses mit abgestimmter Werkzeugunterstützung kann nicht nur eine effiziente und effektive Kollaboration unterschiedlicher Fachbereiche ermöglichen, sondern auch integrierte Maßnahmen der Qualitätssicherung sowie ein effizientes Änderungsmanagement beinhalten und unterstützen.

Zentrale Herausforderungen bei einer umfassenden Integration von heterogenen Methoden und Werkzeugen sind (a) die *technische Integration* der Werkzeuge in einen definierten Entwicklungsprozess, d.h. wie können unterschiedliche Werkzeuge effizient in den Entwicklungsprozess eingebunden werden, (b) die *semantische Integration*, d.h. wie erfolgt ein effizienter Datenaustausch zwischen den unterschiedlichen Fachbereichen und (c) integrierte und übergreifende *Qualitätssicherung* sowie effizientes *Änderungsmanagement*.



### 4.3 Integration von Software-Werkzeugen für die Entwicklung von Automatisierungssystemen mit dem Engineering Service Bus

Abbildung 1 illustriert die zentralen Herausforderungen einer integrierten Systementwicklungsumgebung (Engineering Service Bus) und eine Laufzeitankopplung (Control Service Bus) in der Automatisierungstechnik. Der *Engineering Service Bus* (EngSB) ermöglicht dabei (1) eine effektive Zusammenarbeit der beteiligten Rollen innerhalb des Entwicklungsprozesses, (2) eine Ankopplung der individuellen Methoden und Werkzeuge an eine zentrale Engineering-Plattform und (3) einen effektiven Datenaustausch zwischen Methoden und Werkzeugen über diese Engineering-Plattform. In weiterer Folge können über einen *Control Service Bus* (4) geeignete Rückkopplungen von der Laufzeitumgebung (Betriebsphase) in den Entwicklungsprozess sowohl Wartung als auch Weiterentwicklung unterstützen, sowie (5) integrierte Analysekomponenten Daten aus der Laufzeitumgebung für Prozessbeobachtung und Simulation nutzen. Der Schwerpunkt dieses Vortrags liegt auf dem Engineering Service Bus.

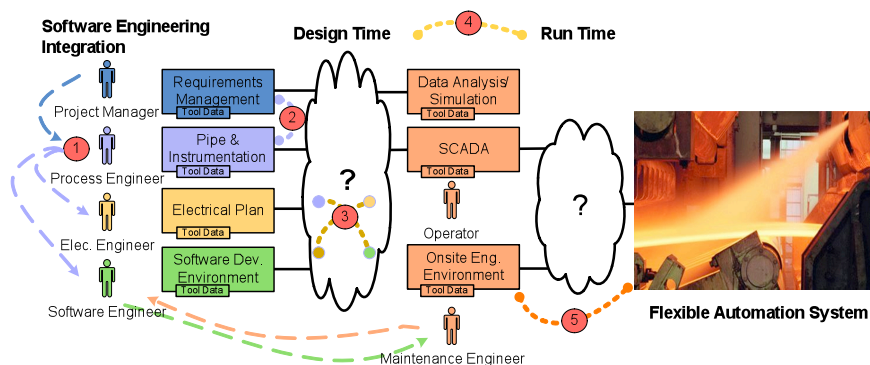


Abbildung 1: Herausforderungen an eine integrierte Werkzeugunterstützung.

Das Konzept des Engineering Service Bus kann somit nicht nur eine effiziente Entwicklung und Wartung von Automatisierungssystemen unterstützen, sondern auch die Basis für die Standardisierung bilden.

Die Präsentation richtet sich an (a) Projekt- und Qualitätsmanager, (b) Systemintegratoren in Organisationen mit einer heterogenen Werkzeuglandschaft und (c) an Werkzeughersteller im Hinblick auf eine mögliche Anbindung an den EngSB in einem heterogenen Systementwicklungsumfeld. Die konkreten Ziele der Präsentation beinhalten:

- Motivation und Vorstellung des EngSB-Konzeptes im Kontext von komplexen und heterogenen Automatisierungssystemen sowie erste Ansätze zur technischen Realisierung.
- Exemplarische Prototypen von konkreten Anwendungsmöglichkeiten im Rahmen der Qualitätssicherung (End-to-End Test) und des Änderungsmanagements.

Das vorgestellte Konzept des EngSB stellt die Grundlage für die Umsetzung im Rahmen eines Christian Doppler Forschungslabors dar. Zielsetzung des Labors ist die Unterstützung komplexer Systementwicklungsprojekte in einer heterogenen Werkzeuglandschaft in der Automatisierungstechnik zur Verbesserung der Produkt- und Prozessqualität. Wesentliches Ergebnis ist eine Open Source Version des „Engineering Service Bus“, die für Wissenschaftler und Praktiker zur Verfügung gestellt werden soll.

## Literaturverzeichnis

- [BS09] Biffi S. and Schatten A. (2009) “A Platform for Service-Oriented Integration of Software Engineering Environments”; invited keynote paper; Proc. of the 8th International Conference on Software Methodologies, Tools and Techniques; Prague, September 2009.
- [BS+09] Biffi S., Schatten A., Zoitl A. (2009) “Integration of Heterogeneous Engineering Environments for the Automation Systems Life Cycle”; Proc. IEEE Industrial Informatics (IndIn) Conf., 2009.
- [LH+09] Lüder A., Hundt L., Biffi S. (2009) “On the Suitability of Modeling Approaches For Engineering Dis-tributed Control Systems”; Proc. IEEE Industrial Informatics (IndIn) Conf., 2009 (best paper award).
- [WB+09] Winkler D., Biffi S., Östreicher T. (2009) “Test-Driven Automation – Adopting Test-First Development to Improve Automation Systems Engineering Processes”, Proc. of European Conf. on Software Process Improvement (EuroSPI), Madrid, Spain.
- [WH+09] Winkler D., Hametner R., Biffi S. (2009) “Automation Component Aspects for Efficient Unit Testing“, Proc. of 14th IEEE Int. Conf. on Emerging Technologies and Factory Automation, Mallorca, Spain.



## Engineering Environment Integration Across Disciplines with the Engineering Service Bus

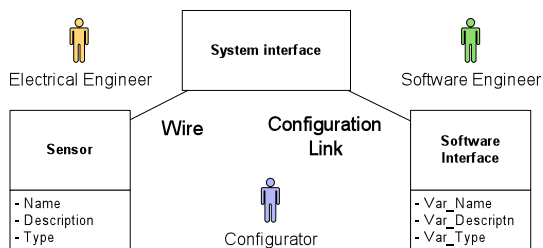
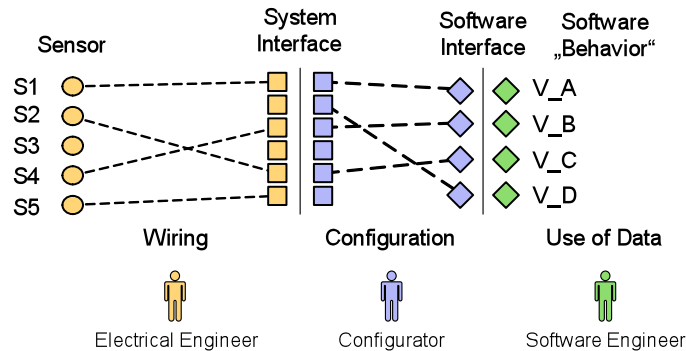
**Dietmar Winkler** **Stefan Biffli**  
 Christian Doppler Laboratory SE-Flex-AS  
 Institute of Software Technology and Interactive Systems (ISIS)  
 Vienna University of Technology  
<http://cdl.ifs.tuwien.ac.at>



## End-to-End Test Across Engineering Models



Use of common concepts in models across engineering disciplines



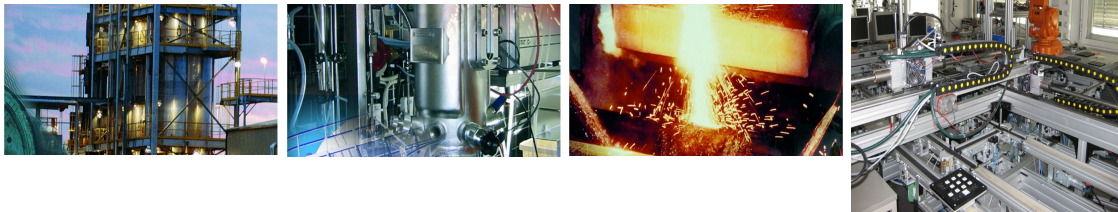
### End-to-End Analysis

- List of sensor name/description/type with Variable name/description/type
- Warnings for incomplete chains between variables and sensors

## Motivation and Overview



- Software-intensive systems.
- Several disciplines cooperate in an industry environment.
- Engineering models (e.g., mechanics and electronics) contain requirements and design constraints for software engineers.
- Existing models and tools focus on supporting engineers in specific disciplines.
- Human experts bridge technical and semantic gaps between models and tools of different engineering disciplines.

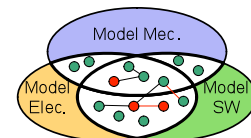
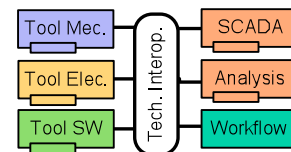


3

## Scope of Research



- **What hinders effective collaboration across disciplines?**
  - Domain- and vendor-specific solutions (e.g., point-to-point integration).
  - Heterogeneous models in various disciplines.
  - Different stakeholders and different “languages”.
  - Limited connection between development and operation.
- **Concept evaluation based on real-world use cases and prototypes**
  - Technical Integration of Tools.
  - Semantic Integration of Data.
  - Quality Assurance across Engineering Disciplines.
- Christian Doppler Laboratory started on January 2010.



4

# Software Engineering Integration for Flexible Automation Systems

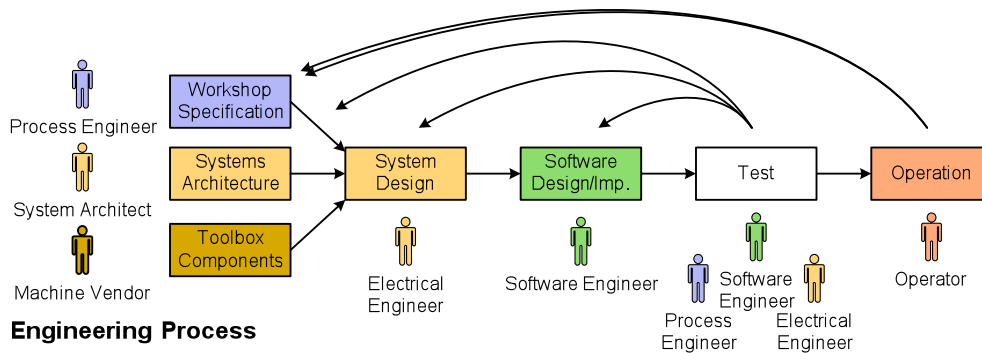


## Basic research challenges

- Early defect detection across engineering discipline and tool boundaries.
- Engineering process analysis using design- and run-time data sources.

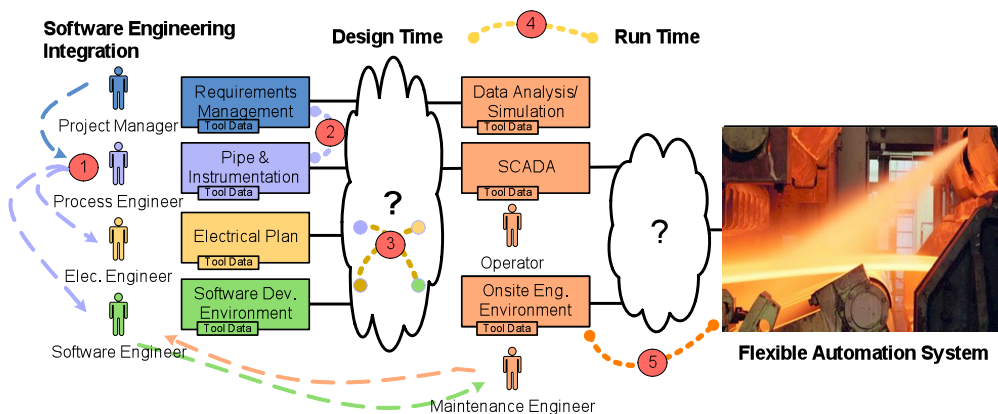
## Research applications in the industry partners' domains

- Platform to build integrated tools for automation systems development & QA.
- SCADA systems with data analysis for monitoring automation systems.



5

# Challenges and Requirements



## Challenges from weak integration of software tools for engineering

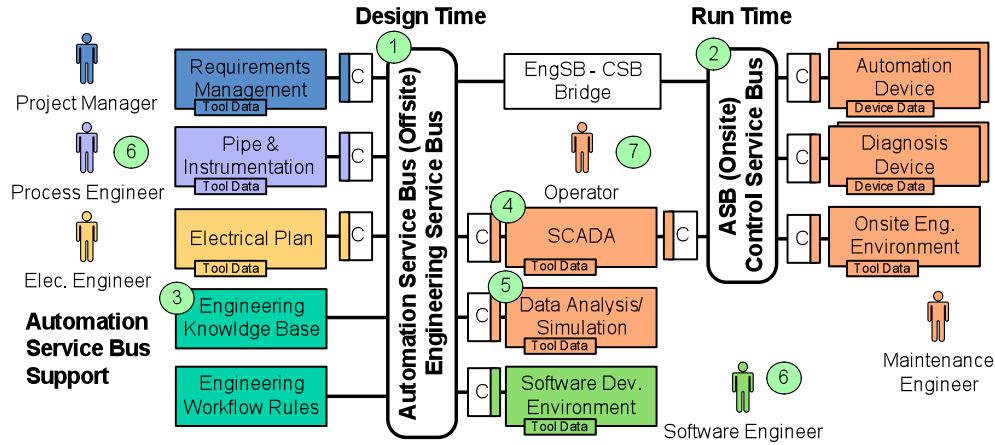
1. Engineering process on event level is hard to track and analyze.
2. Integration of software tools is often vendor-specific and/or fragile.
3. Sharing of data models across software tools is inefficient and risky.
4. Run-time defect detection cannot easily access design knowledge.
5. Integration of run-time environments is hard to observe for analysis.

6



## Automation Service Bus

Goal: Approaches for the integration of software tools in automation engineering.



- Technical Integration: Engineering Service Bus (1), Control Service Bus (2).
- Semantic Integration: Engineering Knowledge Base (3).
- Flexible integration of SCADA (4) with data analysis/simulation (5).
- Defect detection approaches for design time (6) and run time (7).

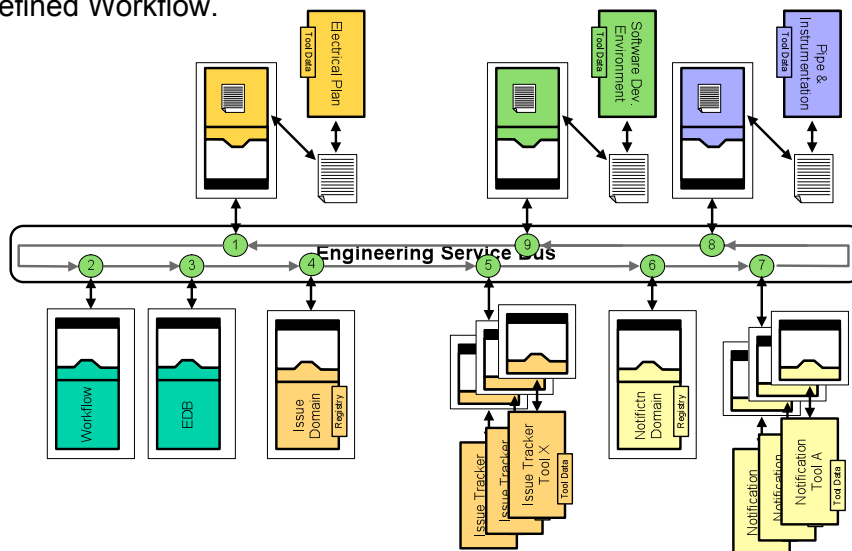
7

## Example: Technical Systems Integration and Interoperability



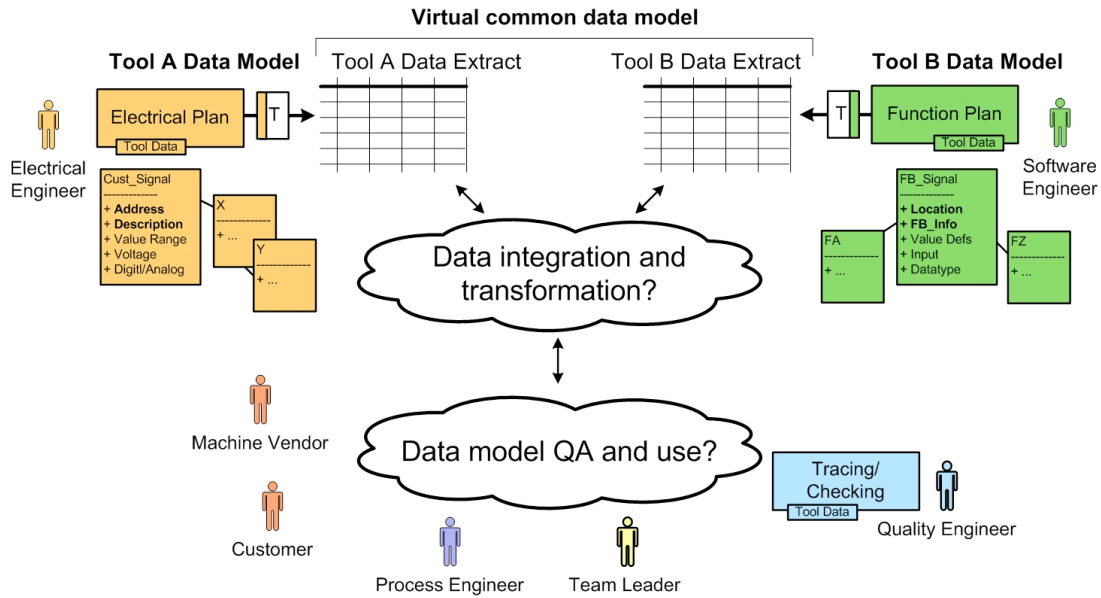
Approach for integrating available automation engineering tools

- 3 heterogeneous engineering tools.
- Defined Workflow.



8

## Collaboration Across Disciplines: Semantics?

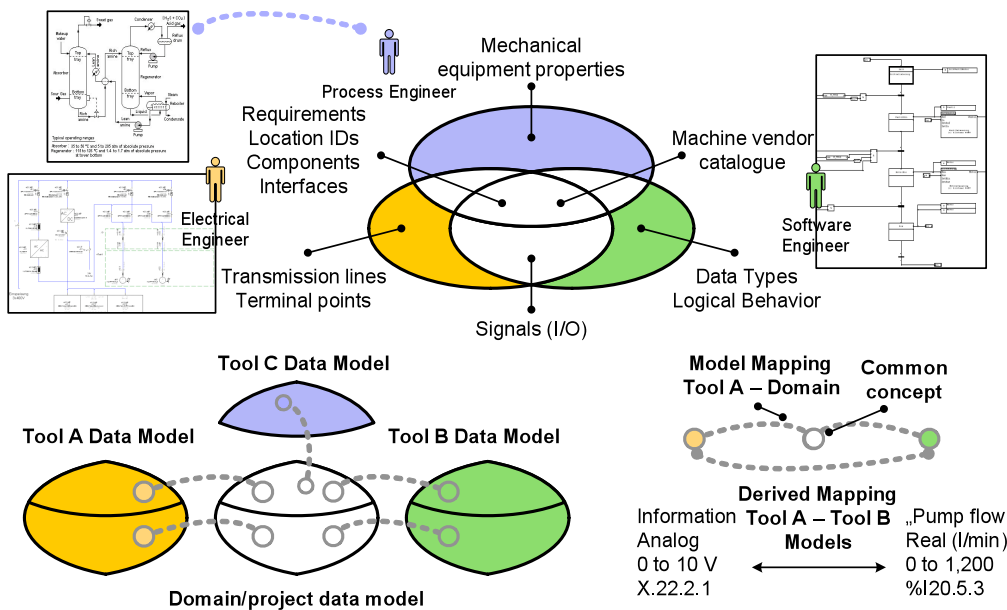


9

## Semantic Integration of Engineering Knowledge



Identification of common concepts across engineering disciplines.

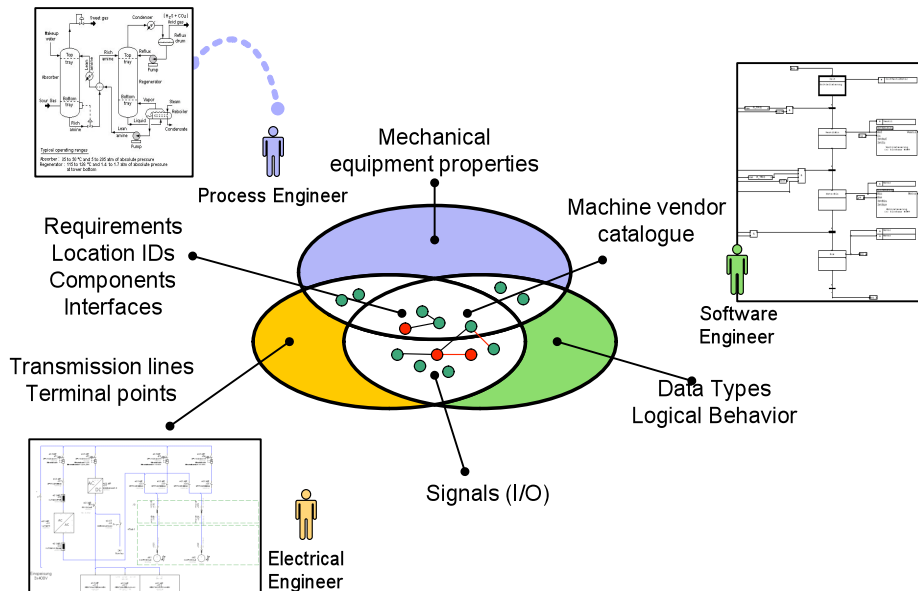


10

# Defect Detection Across Tool Boundaries and Disciplines



Use of common concepts in models across engineering disciplines

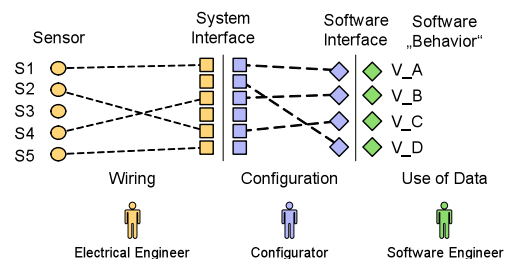
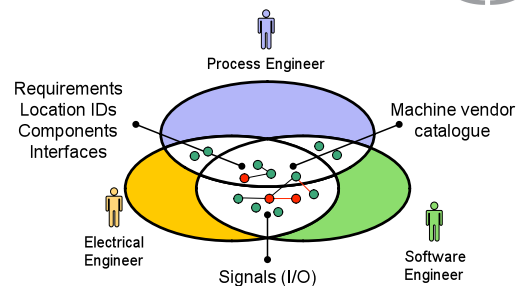


11

# End-to-End Quality Assurance



- Challenge: Defect Detection across engineering disciplines
- **Identification of various defect types:**
  - Missing, wrong, inconsistent model elements or relationships.
  - Conflicts from changes to overlapping model elements.
  - Run-time violation of model constraints.
- **Quality Assurance approaches**
  - Review of overlapping model parts, e.g., with inspections.
  - Automated check of model assertions (syntactic and semantic).
  - Change conflict detection and resolution.



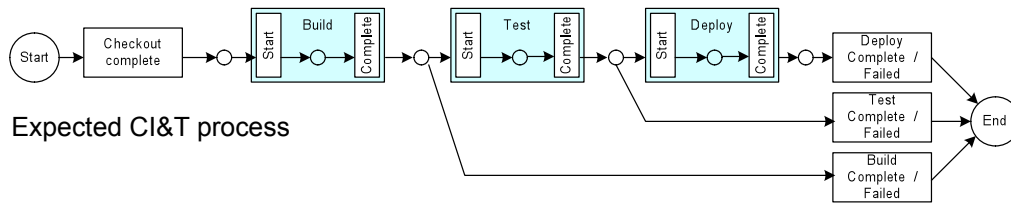
12



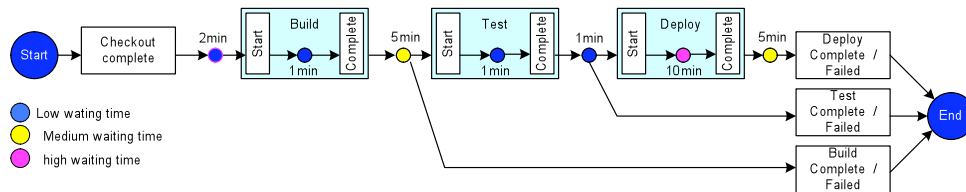
## Engineering Process Analysis (CI&T)



- **Process automation, analysis and assessment** based on (EngSB) event logs
  - Visualization of the expected engineering process.
  - Comparison of expected with traces of actual engineering processes.
  - Analysis of actual engineering process variants (frequency of paths taken).
  - Measurement of engineering process duration, waiting and execution times.
- Example: Continuous Integration and Test (CI&T).



Process analysis based on sample engineering logs..



13

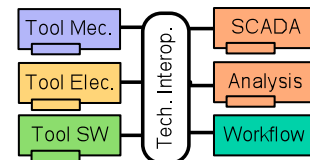
## Summary



- Multi-disciplinary engineering projects are prone to risks from defects and delays due to technical gaps between tools and semantic gaps between data models.
- Technical and semantic integration provide the foundation for engineering process automation and quality management to lower these project risks.

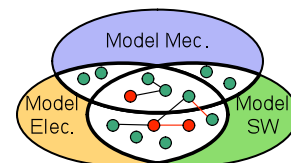
- **The Engineering Service Bus (EngSB) environment provides:**

- Technical Integration: Workflow-Rules and Events.
- Semantic Integration: Data Models across disciplines.
- Defect Detection & Engineering Process Automation: Engineering rules and process analysis.



- **Industry Use Cases**

- End-to-End Quality Assurance.
- Difference analysis between signal versions.
- Defect detection in data models across tools and engineering disciplines.
- Engineering Process Automation, Analysis and Improvement.



14

Thank you ...



## Engineering Environment Integration Across Disciplines with the Engineering Service Bus

**Dietmar Winkler**  
**Stefan Biffli**

Christian Doppler Laboratory "Software Engineering Integration  
for Flexible Automation Systems"  
<http://cdl.ifs.tuwien.ac.at>

Institute of Software Technology and Interactive Systems (ISIS)  
Vienna University of Technology

{Dietmar.Winkler, Stefan.Biffli}@tuwien.ac.at



#### *4.3 Integration von Software-Werkzeugen für die Entwicklung von Automatisierungssystemen mit dem Engineering Service Bus*

## **CESAR: Kosteneffizienz bei der Entwicklung sicherheitsrelevanter Echtzeitsysteme**

Dr. Uwe Kühne, Silke Gerlach

OPES 45 (Real-Time Application Software)  
EADS Deutschland GmbH  
Claude Dornier Str.  
88090 Immenstaad  
uwe.kuehne@eads.com  
silke.gerlach@eads.com

Co-Autoren:  
DI Gerhard Griessnig, Mag (FH) Ingrid Kundner

AVL List GmbH  
Hans-List-Platz 1  
A-8020 Graz  
gerhard.griessnig@avl.com  
ingrid.kundner@avl.com

**Abstract:** Sicherheitskritische Echtzeitsysteme werden immer komplexer und variantenreicher. Hinzu kommen hohe Anforderungen durch die Kritikalität und aufgrund der verlangten Performance, sowie des starken Kostendrucks des Marktes. Bisher wurden solche Systeme im EU-Kontext relativ unabhängig bzgl. Vorgehen, Methoden und Tooling entwickelt. CESAR ist ein EU-Projekt, welches zur wesentlichen Aufgabe hat, eine europaweit oder sogar weltweit anerkannte Plattform von Standards, Methoden und Toolumgebungen zur Entwicklung sicherheitskritischer Echtzeitsysteme bereitzustellen. Die Basis dafür bilden etablierte Vorgehen, Best Practices, Standards, Methoden und Tools sowie neue Herangehensweisen aus den Bereichen der Bahnautomatisierungsindustrie für den Nah- und Fernverkehr, Luft -und Raumfahrt sowie der Automobilindustrie. Der Vortrag stellt die wesentlichen Inhalte des EU-Projektes sowie die Ergebnisse der Beiträge der EADS-DE (Defence Electronics) im Bereich Requirements-Engineering aus der ersten Projektphase vor.

## 1 Motivation

Harte Echtzeitanforderungen, immer komplexere Funktionalitäten (Schnittstellenkomplexität, komplexe Algorithmen, steigende Anzahl von technischen Anforderungen), hohe Sicherheitsanforderungen sind die Herausforderungen an sicherheitsrelevante Echtzeitsysteme der heutigen Zeit. Zivile und militärische Qualifizierungen und Zertifizierungen müssen besonders in der Luft- und Raumfahrttechnik berücksichtigt werden. Demgegenüber steht die Forderung nach Verringerung der Entwicklungskosten und der Entwicklungszeit solcher Systeme. Auch der Avionikbereich der EADS-DE (Defence Electronics) steht, wie alle anderen Firmen, vor diesen Herausforderungen. Diese können nur beantwortet werden, wenn sich die Entwicklungsbereiche kontinuierlich technologisch weiterentwickeln insbesondere bzgl. Standardisierung der Prozesse, Methoden und Tools.

## 2 Ansatz

Zur Steigerung der Kosteneffizienz bei der Entwicklung von sicherheitsrelevanten Echtzeitsystemen hat sich im europäischen Rahmen das CESAR-Konsortium gebildet. Im Rahmen eines ARTEMIS-Projektes werden 55 europäische Partner in einer ersten Phase eine Plattform mit entsprechenden Prozessen und Methoden für die Entwicklung von sicherheitsrelevanten Systemen zur Verfügung stellen. In einer zweiten und dritten Phase des Projektes wird diese Plattform technologisch weiterentwickelt und neue Ansätze integriert. Somit vereint diese Reference Technology Platform (RTP) verschiedene Aspekte (Tools, Prozesse, Komponenten, Entwicklungsumgebungen, Best Practices aus der Industrie und andere Ergebnisse von Forschungsprojekten) und stellt damit eine integrierte Umgebung für die Entwicklung von sicherheitskritischen Echtzeitsystemen bereit. Neben der RTP sind wesentliche Innovationen im Requirements-Engineering und der Komponenten-basierten Entwicklung zu erwarten.

Die Präsentation gibt im ersten Teil einen Überblick über das Projekt CESAR, im Allgemeinen, dessen Herausforderungen, Zielsetzung und erste Ergebnisse. In einem zweiten Teil der Präsentation werden im Detail die Ergebnisse der ersten Projektphase im Zusammenhang mit dem vom EADS-DE erstellten Avionik Show Case vorgestellt und bewertet.

### Keywords:

Embedded systems, Avionic systems, Safety-relevant software, Development process



**CESAR - Cost-efficient methods and processes for safety relevant embedded systems**

## A CESAR Objective Overview

SEE – 2010, 03.- 04. May 2010

Silke Gerlach, EADS Deutschland GmbH



### Agenda



- Project Overview
- Motivation and Rationale
- Project Organisation
- Reference Technology Platform
- EADS-DE Contributions
- Outlook & Conclusion

N°	SN	Legal Name	N°	SN	Legal Name
1	AVL	AVL List GmbH	29	Infineon	Infineon Technologie
2	A-D	ADIR, Deutschland	30	Infineon	Infineon Technologie
3	A-F	AIRBUS France SAS	31	INRIA	National de Recherche Scientifique
4	ABB NO	ABB AS	32	ISI	Industrial Institute of Technology
5	ABB SE	ABB AB	33	KTH	Kungliga Tekniska Hogskolan
6	AbsInt	AbsInt, Ansgarwacht	34	NTNU	Norwegian University of Science and Technology
7	ACCIONA	ACCIONA Infraestructuras	35	NTUA	National Technical University of Athens
8	ASF	Ansaldo Sistemi	36	OFFIS	OFFIS e.V.
9	ASTRIUM Satellites	ASTRIUM SAS	37	ONERA	Office National de Recherches Aeronautiques et Spatiales
10	AUK	AIRBUS UK Limited	38	OSC-ES	OSC - Embedded Oxford
11	AUTH	Autoteia, University of Coimbra	39	Oxford	Oxford University
12	CEA	Commissariat a l'Energie Atomique	40		left project
13	CNRS	Centre National de la Recherche Scientifique	41	SAGEM	SAGEM
14	CRF	Centro de Investigación y Desarrollo en Automoción de la Universidad de Zaragoza	42	SIA	AleniaSIA Spa
15	CSW	Critical Software S.A	43	SIEMENS	Siemens Siemens AG
16	DA	Daniell Automation	44	SINTEF	Sintef Sintef
17	DELPHI	Delphi	45	TC&E	Quintec Associates
18	DLR	Deutsches Zentrum für Luft- und Raumfahrt	46	THALES	Thales Thales Communications
19	DS	Dassault Systemes	47	THALES	Thales Thales Avionics S.A.
20	EADS-DE	EADS Deutschland	48	THALES	Thales S.A.
21	EADS-IT	EADS Italia	49	UNIBO	Alma Mater Studiorum - University of Bologna
22	ED	European Datacenter	50	UNIMAN	The University of Manchester
23	ESI	ESI-Tecnalia	51	UNITS	University of Twente
24	EST	ESTEREL Technologie	52	VIF	The Virtual Vehicle
25	Fraunhofer	Fraunhofer Gesellschaft	53	VOLVO	Volvo Technology
26	FSC	Fernal Software	54	HS	Hispac-Suiza
27	Geensys	Geensys	55	MB	Messier-Bugatti
28	HAI	Hellenic Aerospace	56	TM	Turbomeca

Stand: August 2009

Silke Gerlach, EADS-DE  
SEE-2010; 10-05-03

### An ARTEMIS-JU\* project

#### Consortium

- with 55 Partners (10 European countries)

#### Project Performing:

- Duration: 3 years
- Start: 01.03.2009

#### Manpower:

- Effort: 5124 MM

#### Project Figures:

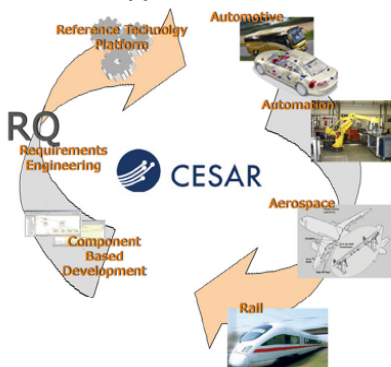
- Total Budget: 58.535.000 €
- Total Funding: 28.317.000 €

#### Contact:

- <http://www.cesarproject.eu>

\* Advanced Research & Technology for EMbedded Intelligence and Systems – Joint Undertaking; see <https://www.artemis-ju.eu>

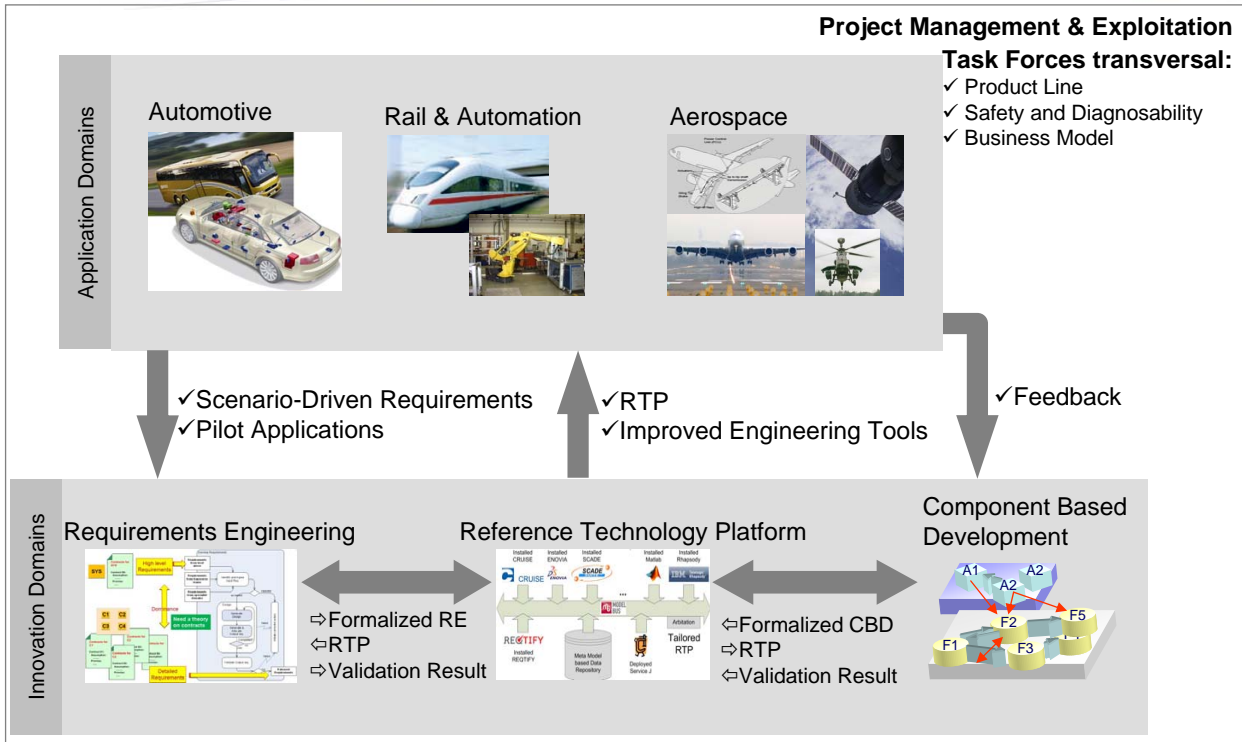
### Multidomain Approach



- Strengthens the European Industry for development of safety-critical embedded systems
- Automotive, Aerospace, Rail and Automation: ultra-reliable embedded systems for increased mobility and ensuring safety
- Boost cost efficiency of embedded systems development and safety processes by an order of magnitude.
- Creating the European cross-sectoral standard reference technology platform (RTP) providing meta-models, methods, and tools for safety-critical hard-real-time system development.
- Compliant to relevant industry standards such as AUTOSAR and safety related standards e.g. ISO CD 26262 (automotive), IEC 61508 (automation), DO178B (aerospace) and EN50128 (rail).
- Project results will be provided to the whole embedded community\*

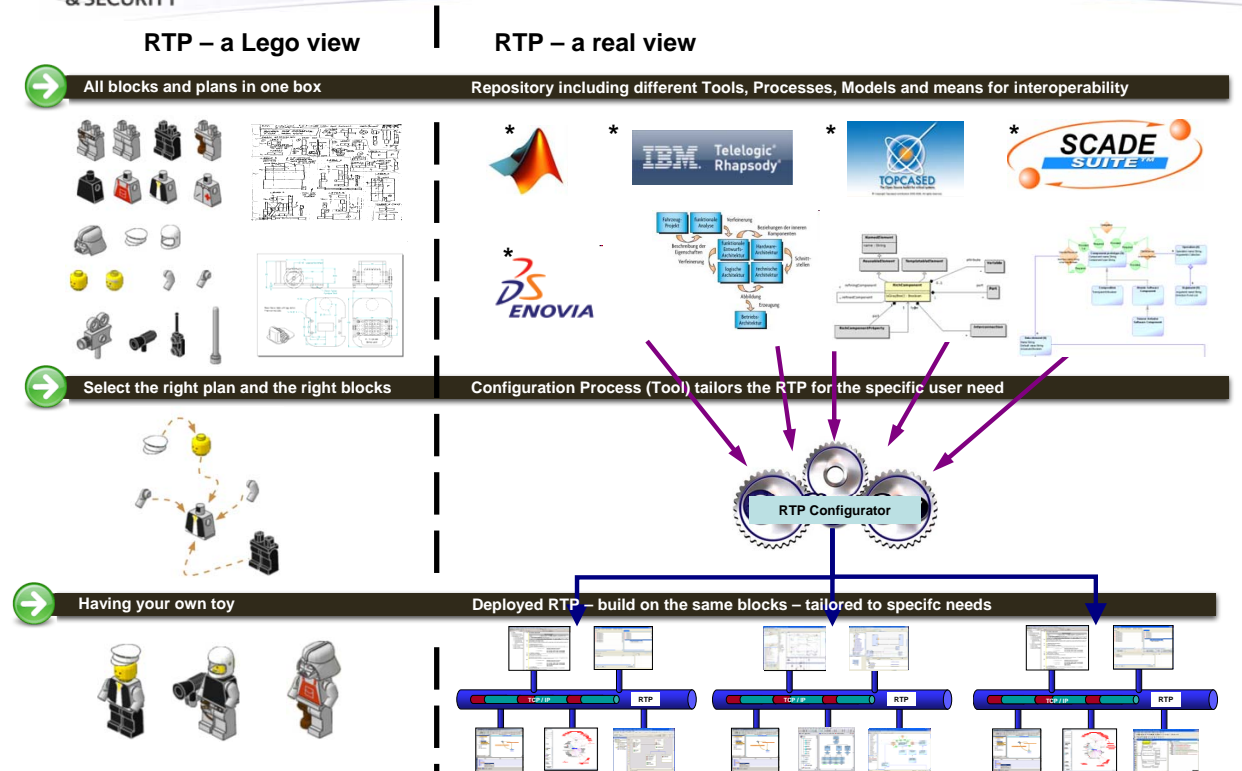
Silke Gerlach, EADS-DE  
SEE-2010; 10-05-03

\* terms & conditions to be defined



Silke Gerlach, EADS-DE  
SEE-2010; 10-05-03

5



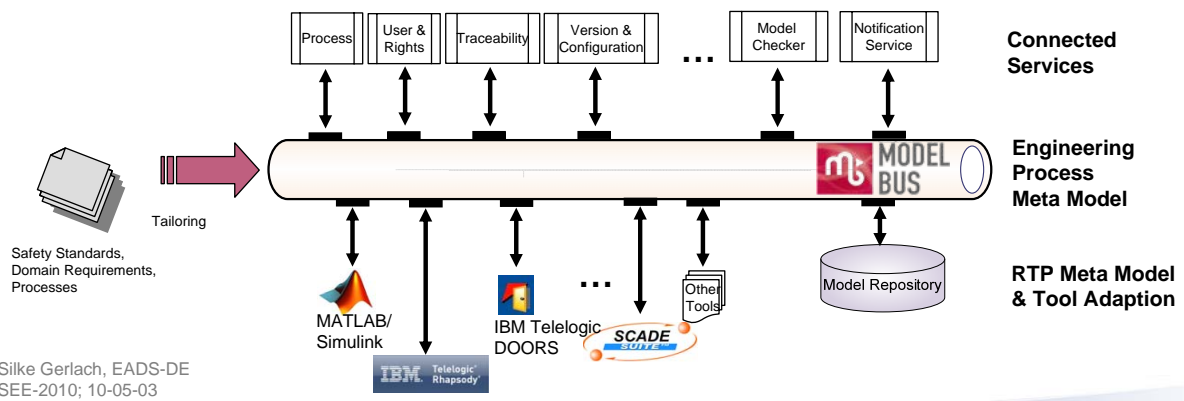
Silke Gerlach, EADS-DE  
SEE-2010; 10-05-03

\* Not confirmed, examples only.

6

- Multi-user & integrated development environment for the development of embedded systems
- Providing tools, means for interoperability and methods for the development of embedded systems
- Including probably more tools, standards and methods than one user/company will need (cross domain approach / tool options)

The RTP has to be tailored (via a configuration tool) in order to be deployed



7

### Main EADS-DE Contribution

- Requirements Engineering work packages with
  - Requirements language definition
  - Definition of improved RE methods
  - Implementation of improved RE methods
  - RE guidelines
  - Support of evaluation phase
- Aerospace Domain work packages with
  - Providing pilot application and domain requirements for RTP
  - Definition of Avionic show case and support the show case implementation
  - Validation of improved methods and tools of RTP (constructive feedback)

8



## EADS-DE: Practical Implementation

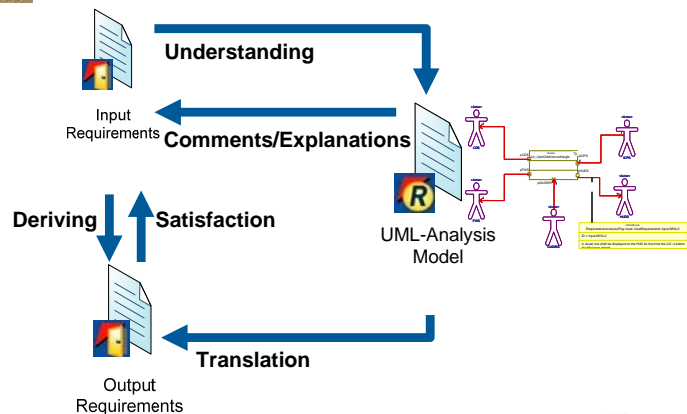
- Threshold Warning as Pilot Application

## EADS-DE: RE Methods & Tools\*

- SysML / UML 2.0
- IBM Rhapsody
- IBM DOORS

\* An extract related to CESAR RTP V0.5

Silke Gerlach, EADS-DE  
SEE-2010; 10-05-03



9

## Functional Analysis by using an executable “Black Box model”

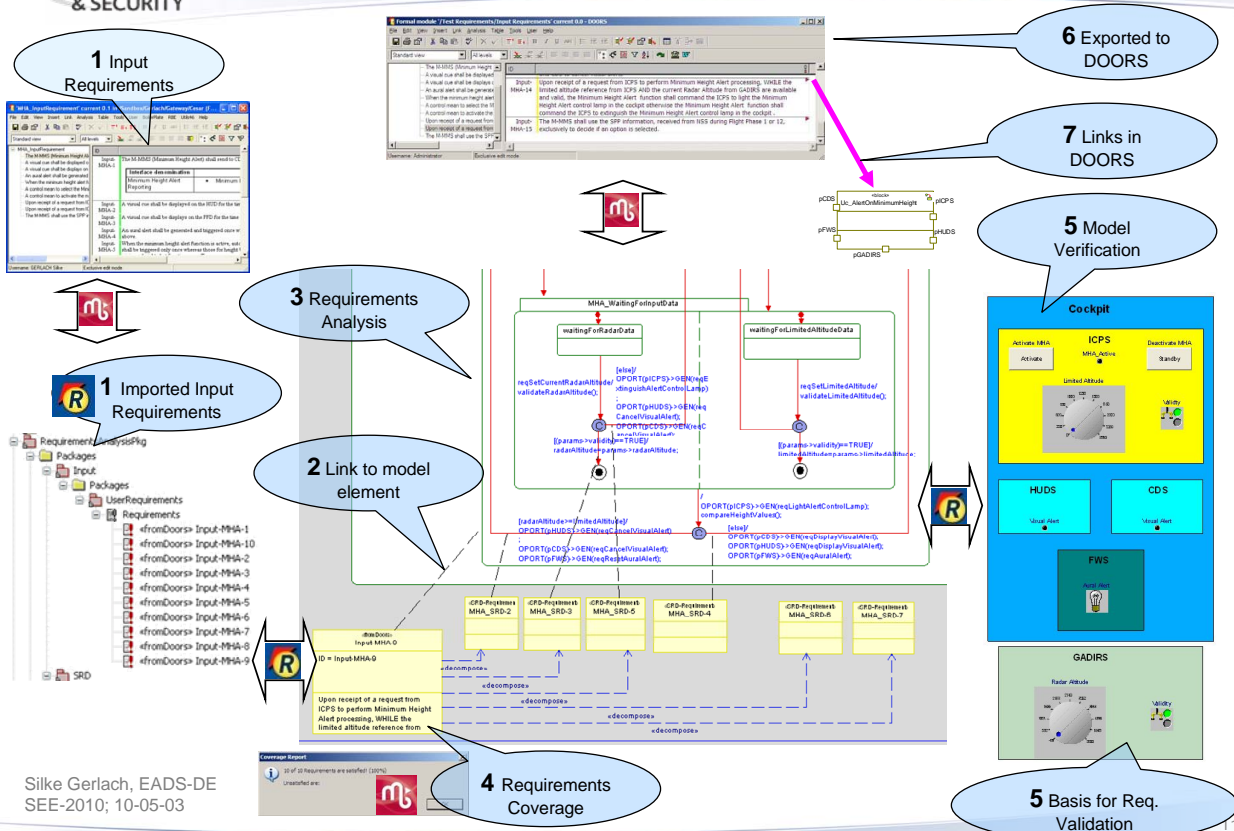
Storyboard:

1. Import user requirements from DOORS to Rhapsody
2. Linking the imported requirements to corresponding model elements
3. Getting on with requirements analysis (refine, derive, split, validate, etc.)
4. Providing an automatic requirements coverage
5. Verification of the Rhapsody model by model executing
6. Providing an automatic export of the Rhapsody model to DOORS
7. Automatic link creation in DOORS

Silke Gerlach, EADS-DE  
SEE-2010; 10-05-03

10





Silke Gerlach, EADS-DE  
SEE-2010; 10-05-03

### Functional Analysis by using an executable “Black Box model”

#### Affected domains

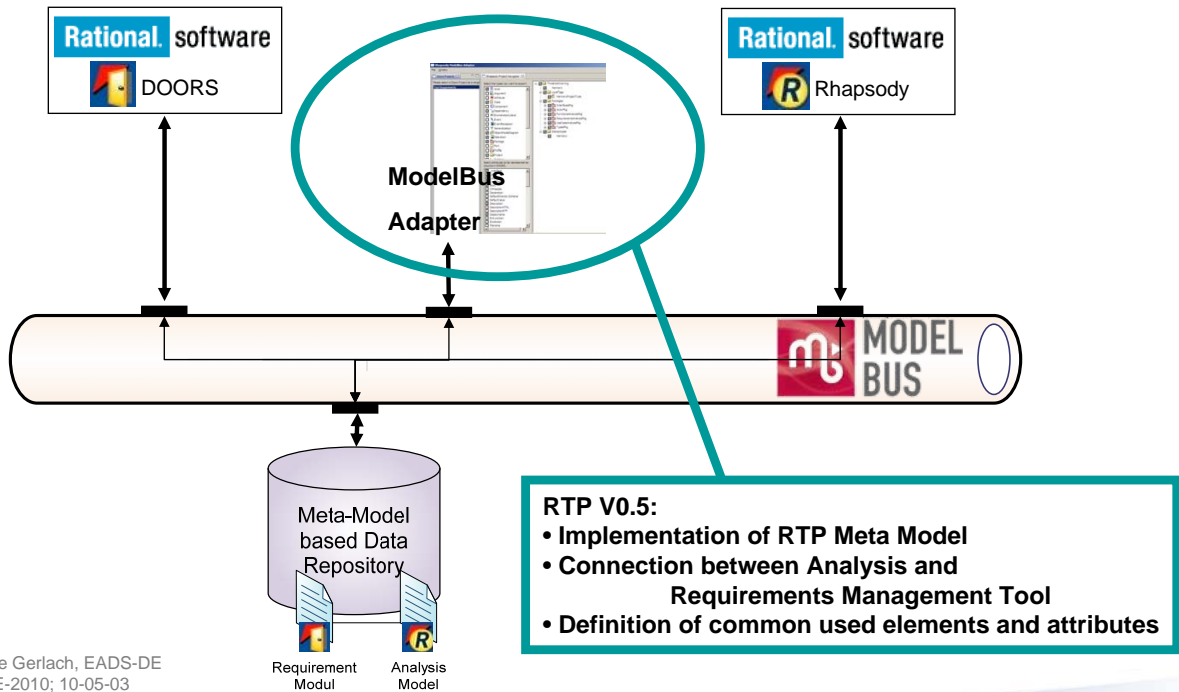
- Requirements Analysis / Functional Analysis
- System- & Software-Modelling

#### Involved assets

- SysML / UML 2.0 (meta model)
- Repository (Basic Service)
- Model Transformation Service (Basic Service)
- Model bus (Tool Integration Infrastructure)
- Rhapsody (Tool; Owner: IBM)
- DOORS (Tool; Owner: IBM)

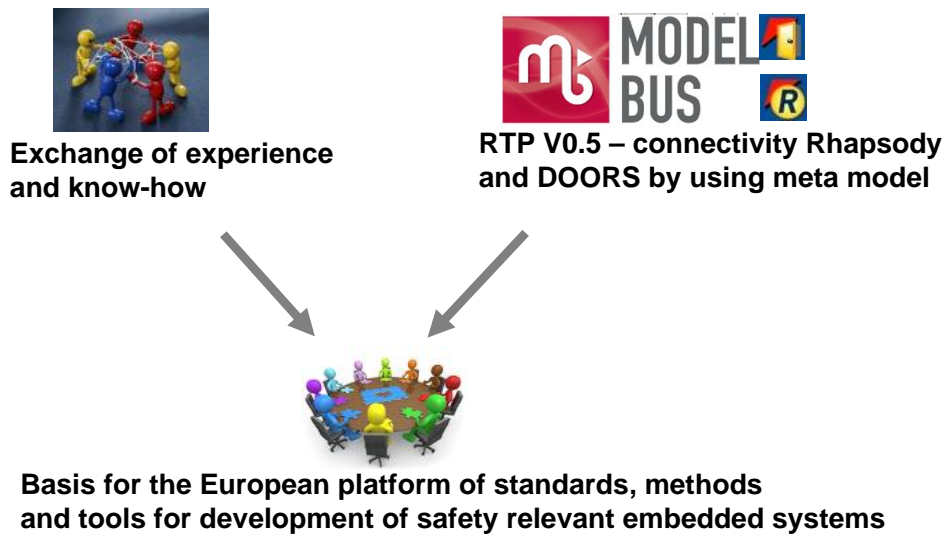
Silke Gerlach, EADS-DE  
SEE-2010; 10-05-03

**RTP realisation issues related to Avionic show case**



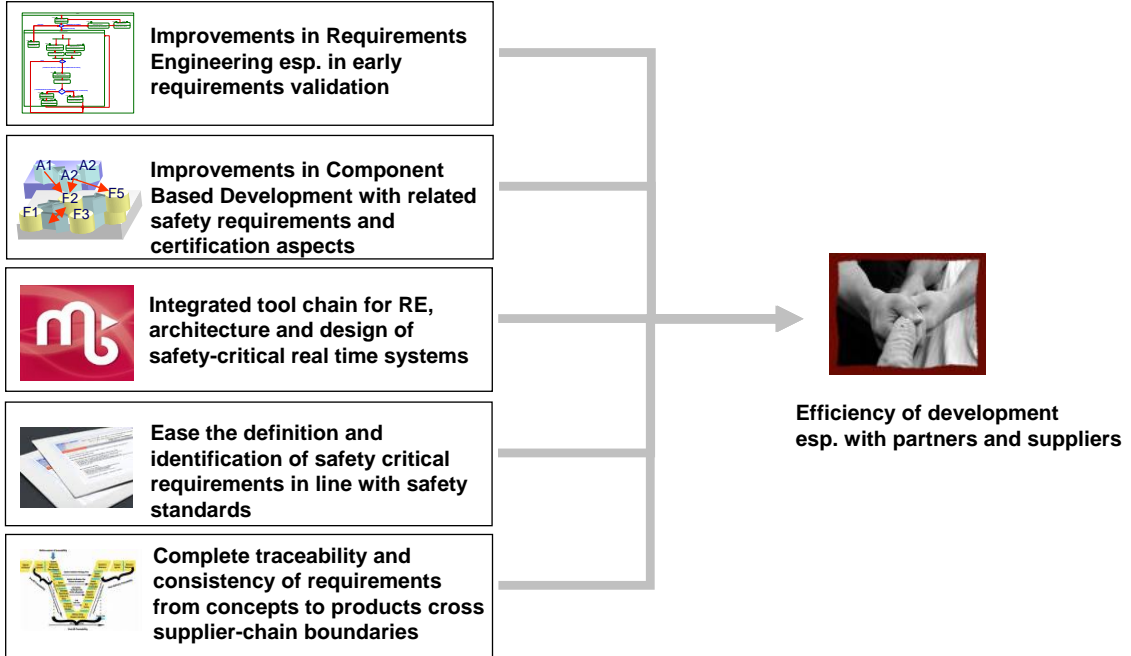
13

**CESAR benefit for EADS-DE in 1<sup>st</sup> project phase:**



14

### EADS-DE expectations from next project phases:



Silke Gerlach, EADS-DE  
SEE-2010; 10-05-03

15

## Thank you for your Attention!

Discussion, any Questions, Feedback, ...



Silke Gerlach  
EADS-DE  
OPES45,  
Real-Time Application Software  
Email: [silke.gerlach@eads.com](mailto:silke.gerlach@eads.com)

CESAR: [www.cesarproject.eu](http://www.cesarproject.eu)  
Project Management and Coordinator:  
I. Kundner or G. Griessnig  
AVL List GmbH  
Hans-List-Platz 1  
8020 Graz  
Austria  
[info@cesarproject.eu](mailto:info@cesarproject.eu)

Silke Gerlach, EADS-DE  
SEE-2010; 10-05-03

16

## 5. Management und Modellierung von Anforderungen

### Sessionüberblick

---

5.1. Agiles Anforderungsmanagement . . . . .	189
5.2. Integration von textuellen Anforderungen und Modellen . . . . .	205
5.3. Geschäftsregelmanagement und Fachmodellierung . . . . .	210
5.4. Nichts ist beständiger als der Wandel . . . . .	239

---



## Agiles Anforderungsmanagement - Weiterentwicklung von Standards oder Neuerfindung des Rades?

Bernd Oestereich

Geschäftsführung  
oose Innovative Informatik GmbH  
Strassenbahnring 7  
20251 Hamburg  
bernd.oestereich@oose.de

**Abstract.** Die agile Softwareentwicklungsmethode Scrum macht streng genommen keine Aussagen, mit welchen Techniken Anforderungen erhoben und vermittelt werden sollen. Ein sehr gängiges Mittel sind aber Benutzergeschichten. Dieser Beitrag setzt sich damit auseinander, was Benutzergeschichten eigentlich sind, vor allem wie sie in die Reihe bestehender Anforderungstechniken einzuordnen sind. Die Disziplin des Requirements-Engineering (RE) hat in den letzten 10 – 20 Jahren eine ganze Reihe anspruchsvoller Herausforderungen bearbeitet und zu mittlerweile vielen bewährten Techniken und Best Practices geführt. Dennoch hat es den Anschein als würde Scrum mit Benutzergeschichten darauf nicht zugreifen, sondern losgelöst davon eigene Ansätze entwickeln.

Der Beitrag erläutert, warum das so ist, zeigt die Verwandtschaft und Unterschiede agiler und klassischer Anforderungstechniken, wie diese gegenseitig adaptiert werden können und wie sie sich gegenseitig befruchten können.

### 1 Motivation

Requirements Engineering (RE) ist eine reife Disziplin und hat in den letzten 1 – 2 Jahrzehnten eine Reihe erfolgreicher und bewährter Standardtechniken hervorgebracht. Nicht zuletzt deswegen gibt es mittlerweile auch anerkannte Personenzertifikate [CPRE] mit denen der Kompetenznachweis für diese Disziplin erbracht werden kann.

Der negative Ruf wasserfallgetriebenen Vorgehens in IT-Projekten hat auch erheblich auf die Disziplin RE abgefärbt, widmet das Wasserfallmodell dem Thema doch mehr oder weniger eine ganze Phase. Auch wenn agile Verfahren längst den Mainstream erreicht und das Gegenteil bewiesen haben, so werden Pflichtenhefte, Ausschreibungs- und Festpreissituationen doch immer noch als Indikatoren für die Notwendigkeit zu schwergewichtigem Anforderungsmanagement verkannt.

Requirements Engineering an sich kostet nur. Jedes Requirements Engineering muss daher einen Nutzen bringen, der seine Kosten deutlich übersteigt. Diese Erwartungen an

das Requirements Engineering sind soweit normal, auch im herkömmlichen Requirements Engineering weitgehend Konsens und haben mit agilem Requirements Engineering noch nicht viel zu tun. Wenngleich das agile Manifest bereits ähnliche Forderungen stellt.

Andererseits ist Requirements Engineering die Achillesferse so populärer agiler Verfahren wie Scrum. Scrum definiert mit dem Product Owner eine Rolle, die einen starken Bezug zum Requirements Engineering hat, fordert eine Geschäftwertpriorisierung und legt ein paar Verantwortlichkeiten im RE-Kontext fest. Darüber hinaus macht Scrum jedoch keine Aussagen oder Vorschriften, in welcher Form Anforderungen zu erfassen, zu vermitteln und zu verwalten sind. Dennoch ist beispielsweise die Arbeit mit so genannten Benutzergeschichten weit verbreitet.

Die Rolle des Product-Owners ist nach Meinung des Autors sehr anspruchsvoll, möglicherweise gar überfrachtet und die Praxis, Publikationen und Internetforen zu Scrum bzw. Benutzergeschichten nehmen auffällig wenig bis kein Bezug auf die Errungenschaften der Disziplin RE. Das hängt sicherlich mit dem zuvor genannten unagilen Ruf des Requirements Engineerings zusammen und ist insoweit verständlich. Andererseits bekommen RE-Profis schnell den Eindruck, hier würde das Rad des Requirements Engineerings gerade neu erfunden werden.

Meiner Meinung nach sind die agilen Vorbehalte gegenüber der Disziplin Requirements Engineering nicht gerechtfertigt. Requirements Engineering als Standard und Disziplin bietet viele Techniken und Best Practices, die sich auch in agilen Vorgehensmodellen mit hohem Nutzen adaptieren lassen. Andererseits liefern auch Scrum und Benutzergeschichten Anregungen fürs nicht-agile Requirements Engineering.

## 2 Allgemeine Herausforderungen

Welchen Herausforderungen muss sich das agile RE typischerweise stellen?

- Anforderungs-, Planungs-, Realisierungs-, Liefer- und Abnahmeeinheiten erfordern teilweise spezifische Beschreibungsmerkmale und sollten daher konzeptionell unterschieden werden können.
- Identität von Anforderungen (z.B. für eindeutige Referenzen)
- Wie können unterschiedliche Meinungen verschiedener Stakeholder zu einer Anforderung erkannt und geklärt werden?
- Abhängigkeiten zwischen Anforderungen sollen vermieden werden. Aber wie genau können diese Abhängigkeiten erkennen, vermeiden und ggf. trotzdem handhaben?
- Eine Geschäftwertpriorisierung alleine ist oftmals nicht ausreichend, es sind verschiedene sich ergänzende wie widersprechende und konzeptionell heterogene Aspekte bedarfsweise zu berücksichtigen:

- Geschäftsnutzen
  - Kosten-Nutzen-Verhältnis oder return on Investment (ROI),
  - Stabilität,
  - Verbindlichkeit,
  - Risikograd,
  - Dringlichkeit/Zeitfenster,
  - Abhängigkeiten
- Wie können Dubletten, d.h. inhaltliche Überlappungen, auch solche aufgrund sprachlicher und terminologischer Varianz, erkannt und gehandhabt werden?
  - Wie werden inhaltliche und formale Widersprüche erkannt und behandelt?
  - Welche Abstraktions- und Aggregationsmöglichkeiten und –ebenen für Anforderungen gibt es? Wie sind sie zu erkennen, zu benennen, zu reproduzieren und sicher zu handhaben?
  - Wie können inhaltliche, formale und sprachliche Lücken, Tilgungen und zu grobe Anforderungen sowohl temporär ermöglicht, ebenso aber erkannt und behoben werden?
  - Inwieweit müssen und wie können Anforderungsartefakte personenunabhängig reproduziert werden?
  - Verständlichkeit und Übersichtlichkeit
  - Welche Skalierbarkeit, Effizienz, methodische Kosten und Einfachheit erlauben oder fordern welche Anforderungstechniken und –konzepte?
  - Was ist die INVEST-Kriterien (Independent, Negotiable, Valuable, Estimable, Small, Testable) und warum reichen sie nicht (immer) aus?
  - Wie können Anforderungsartefakte mengenmäßige und organisatorisch effizient gehandhabt und bewältigt werden?

## Literaturverzeichnis

- [CPRE] International Requirements Engineering Boards (IREB e.V.): *Certified Professional for Requirements Engineering*. <http://certified-re.de/>
- [RP07] Roman Pichler, *Scrum*, dpunkt-Verlag, 2007.



# Agiles Requirements Engineering

oose.  
Innovative Informatik



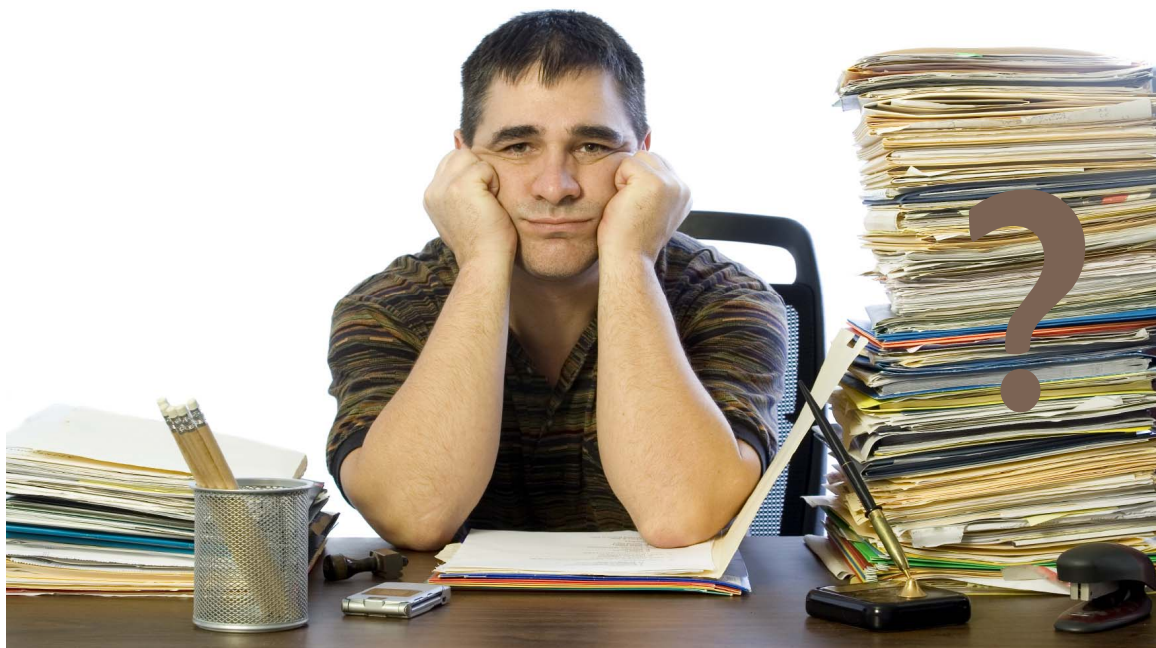
- Agenda
1. Einleitung
  2. Product Backlog



## Was ist agiles RE?

oose.  
Innovative Informatik

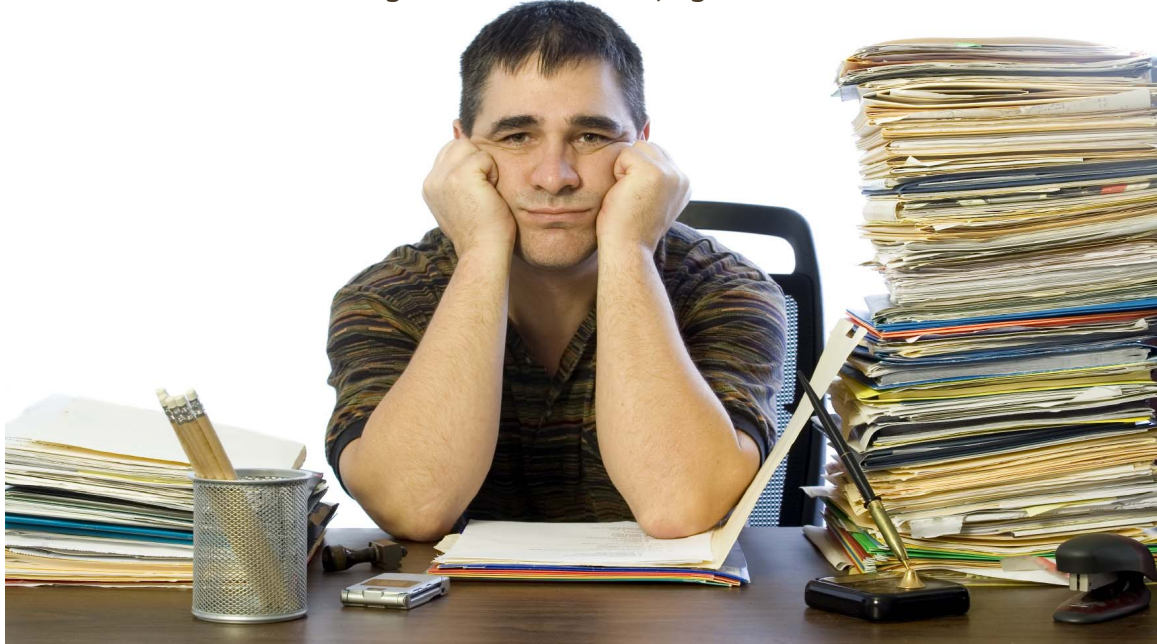
➔ Gilt ebenso für PM und Architektur.  
RE an sich kostet nur und kann nur durch einen höheren Nutzen gerechtfertigt werden.



## Vom RE zu agilen RE

Klassisches RE ist auf Perfektion konditioniert:  
Anforderungen möglichst weitgehend verstehen und präzise beschreiben.

Nichts steht beim Übergang zum agilen RE mehr im Weg als RE-Pefektionismus.  
Insofern haben es erstklassige RE-ler nicht leicht, agil zu werden!



## Vom einfachsten zum agilen RE

Agiles Verfahren wie Scrum und XP basieren  
de facto auf einfachsten RE-Mitteln.

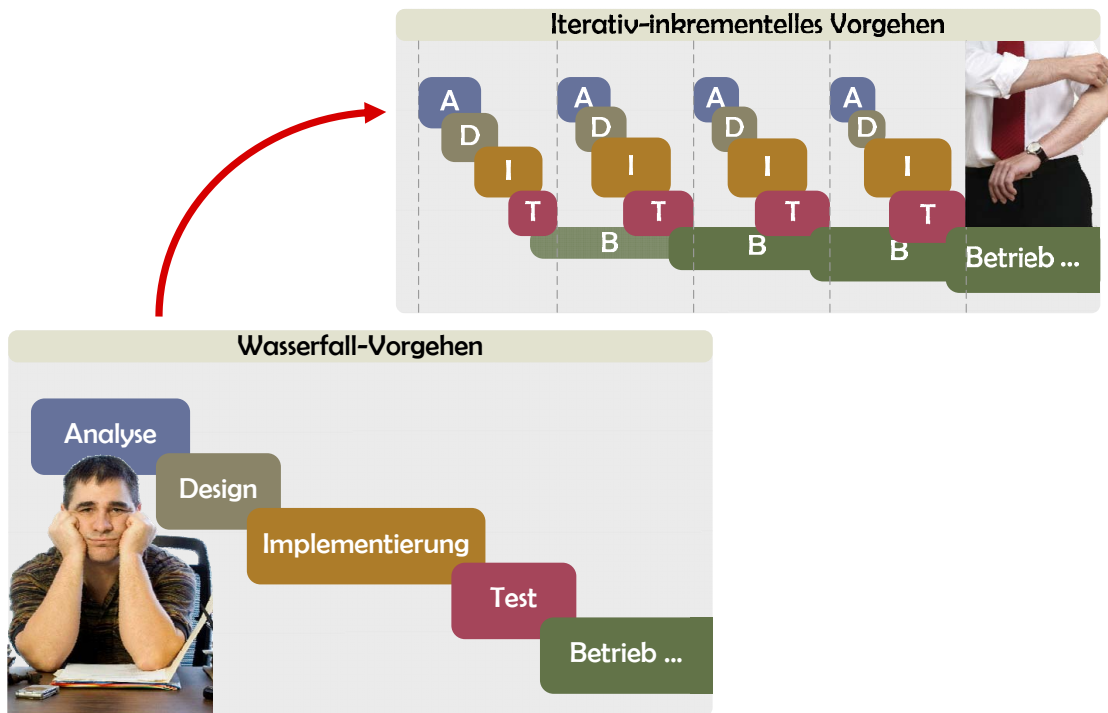
Nichts steht aber einem effizienten agilen RE  
mehr im Weg, als die Nichtbeherrschung  
fortgeschrittener RE-Techniken.

Insofern haben Scrum- und XP-ler es nicht  
leicht, erstklassig zu werden.





# Erster Schritt zum agilen RE: Iterativ-inkrementelle Entwicklung



# Unvollkommenheit aktiv steuern



Klassischen RE-ler sollten die gezielte Unvollkommenheit als notwendigen Optimierungsschritt akzeptieren lernen.  
Erfahrenen Agilisten sollten anspruchsvollere RE-Techniken beherrschen lernen.

# Benutzergeschichte vs. Anwendungsfall OOSE. Innovative Informatik

## Benutzergeschichte

<i>Als Callcenter-Agent möchte ich der Bestellung einen Artikel hinzufügen.</i>



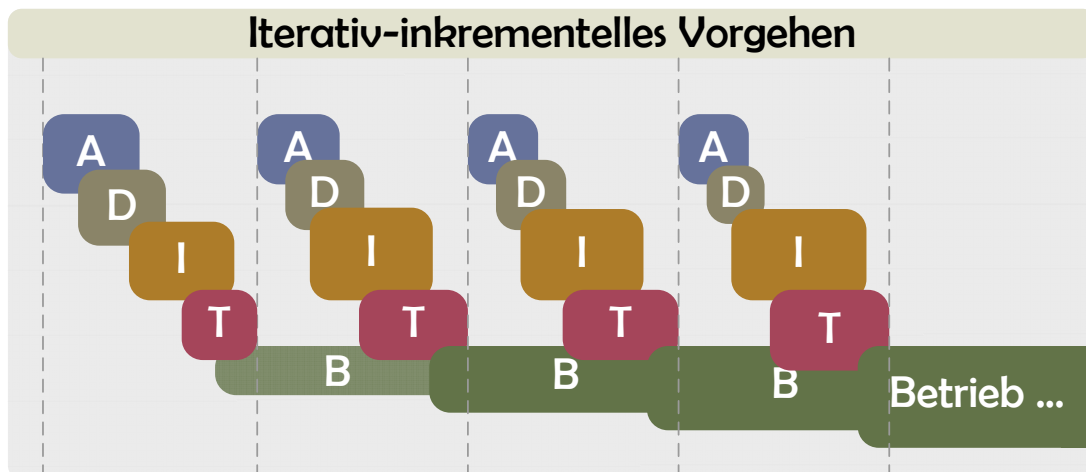
## Anwendungsfall



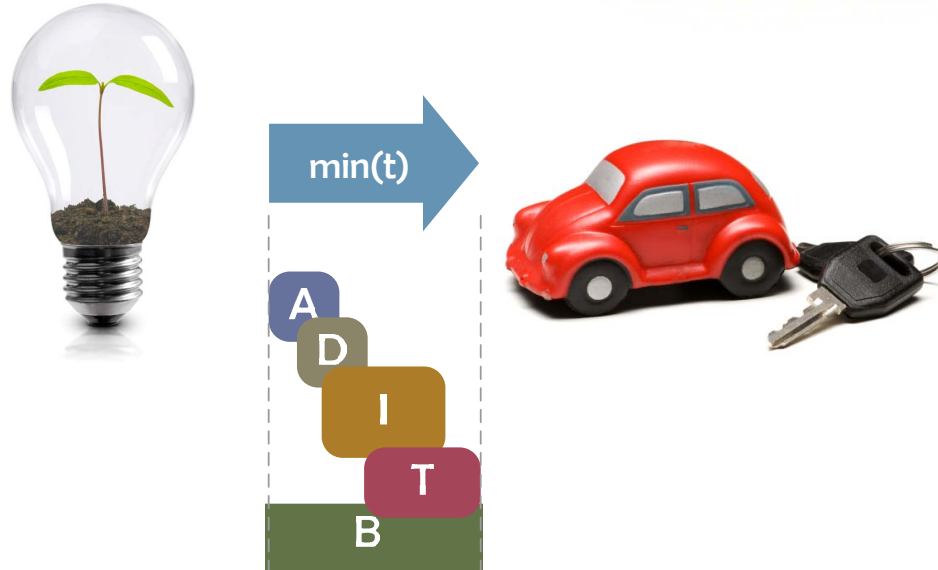
Titel	<i>Artikel zufügen</i>
Beschreibung	<i>Als Callcenter-Agent möchte ich einer Bestellung einen weiteren Artikel zufügen [so dass die Bestellung um den Artikel erweitert wurde].</i>
Akteur	<i>Callcenter-Agent</i>
Auslöser	<i>Kunde äußert Bestellwunsch</i>
Ergebnis	<i>Bestellung um Artikel erweitert</i>

## Iterativ-inkrementelles RE: Anforderungen zeitlich verteilen

**OOSE.**  
Innovative Informatik



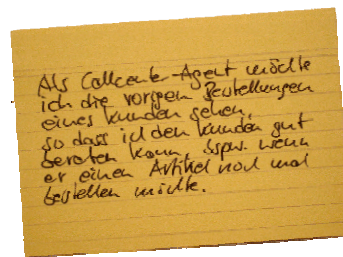
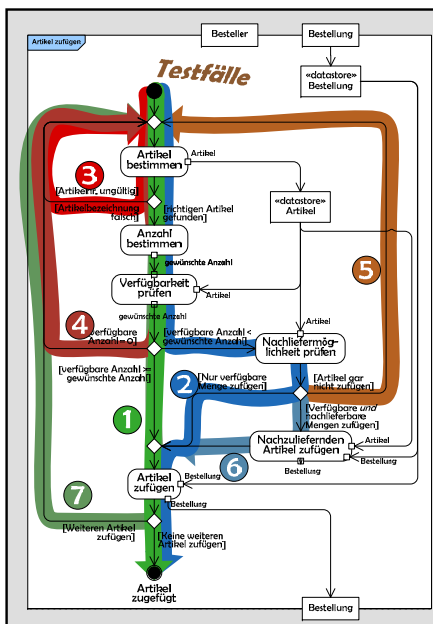
# Strategie: Zeit vom Beachten einer Anforderung bis zur Akzeptanz ihrer Umsetzung minimieren



## Anforderungsspezifische Verständnistiefe

Mal so...

... und mal so





## Wann ist genug?



- Wie vertraut ist der Product Owner / die Fachabteilung / der Anforderungsanalytiker / das Entwicklungsteam mit der fachlichen Domäne?
- Wie homogen werden Geschäftsprozesse / Anwendungsfälle von verschiedenen Fachabteilungsmitarbeitern / Benutzern bearbeitet?
- Wie viele fachliche Varianten und Ausnahmen sind zu berücksichtigen bzw. werden erwartet?
- Wie neu / verändert ist der Ablauf?
- Wie viele verschiedene Personen sind als Anforderungsgeber oder -beeinflusser zu berücksichtigen?
- Wie viele Erkenntnisse über noch zu berücksichtigende Details / Varianten / Anforderungen, die in einfacher Weise mit RE vor dem Sprint erkannt werden könnten, entgehen uns?
- Wie hoch ist das Risiko, durch zu wenig RE wichtige Abhängigkeiten oder Details zu vergessen, die später Mehrkosten oder Verzögerungen verursachen?
- Wie gravierend sind die Folgen, wenn wichtige fachliche Varianten oder Details vergessen werden?
- Sind die Teams sehr / oft voneinander abhängig?

Benutzergeschichte (Epos, Thema), Anwendungsfall (Systemanwendungsfall, Geschäftsanwendungsfall), Geschäftsprozess (Kern-Geschäftsprozess, unterstützender Geschäftsprozess), Story Map, Produktkarton, Feature (Produktfeature, Releasefeature, Iterationsfeature), Systemkontext-Modell, Stakeholder-Diagramm, Stakeholder-Profil, Linguistische Techniken (W-Fragen, Universalquantorenanalyse etc.), Natürlichsprachliche Phrasen/Syntaktische Muster, Schablonen, Formale Sprachen und Diagramme (UML, DSL, BPMN, ERM, EPK), Geschäftsmotivationsmodell (Ziel, Strategisches Ziel, Operatives Ziel, Strategie, Vision, Mission), Glossar, Begriffsmodell, Akteurmodell, Abhängigkeitsgraph, Ursache-Wirkungs-Graph, Entscheidungstabelle, Entscheidungsbaum, Ablaufdiagramm, Interview-Techniken, Workshop-Techniken, Kommunikationstechniken, Verhandlungstechniken, Abzanzkriterium, Priori-

# ARE: Wie viel RE ist genug?



- Unstrukt. Wünsche, Absichten und Ideen
- Übersichten (Story Map, Systemkontext)
- Strukturen (Phrasen, Benutzer geschichten)



- Linguistik
- Strukturen (Phrasen, Anwendungsfälle)
- Formalien (IDs, Referenzen)

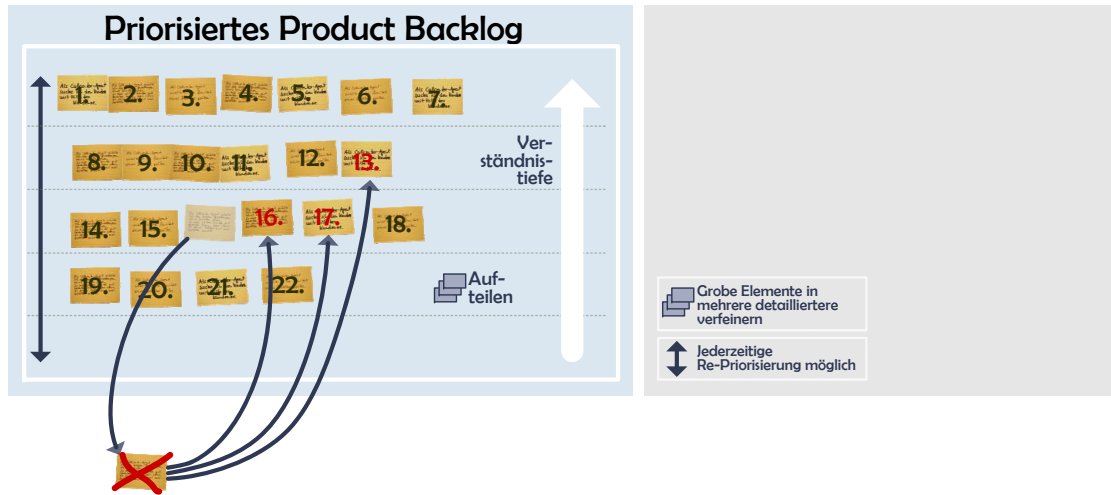


- Linguistik
- Strukturen (Phrasen, Anwendungsfälle)
- Formale Sprachen (UML, DSLs, BPMN, ERM, EPK etc.)

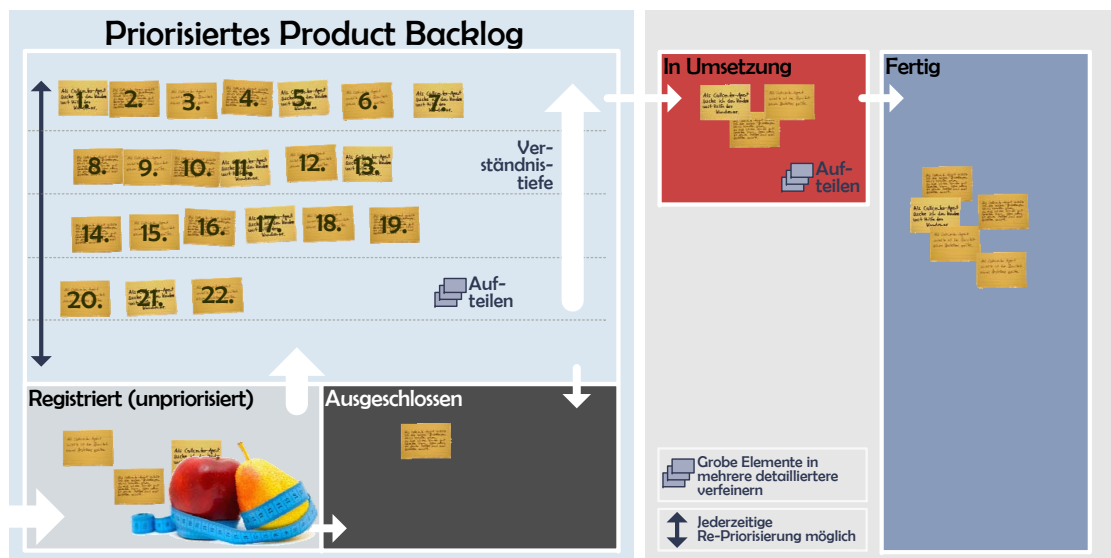


## 2. Product Backlog

# Das Product Backlog und die Eigenschaften seiner Elemente (1)

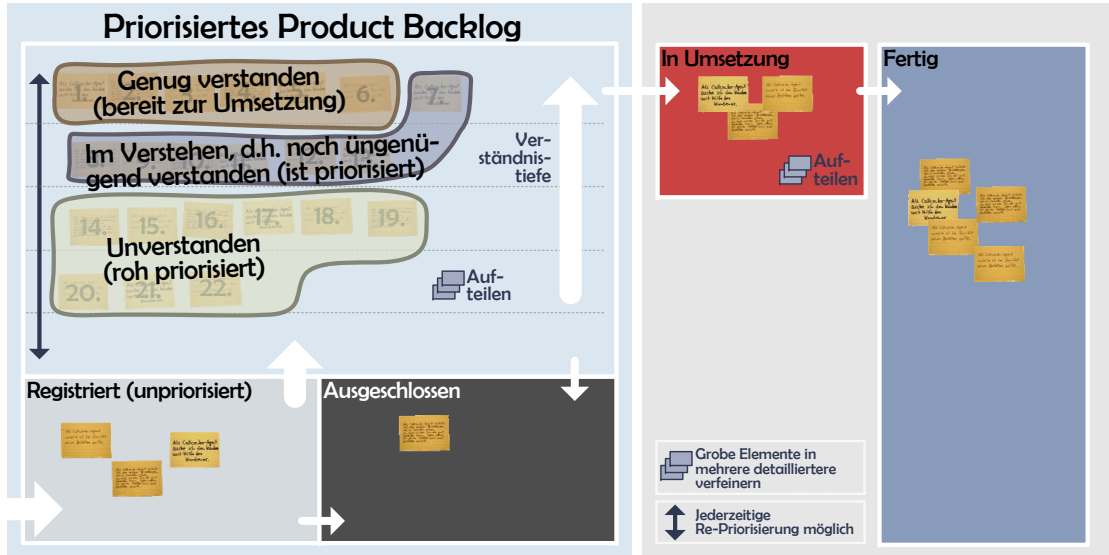


# Das Product Backlog und die Eigenschaften seiner Elemente (2)

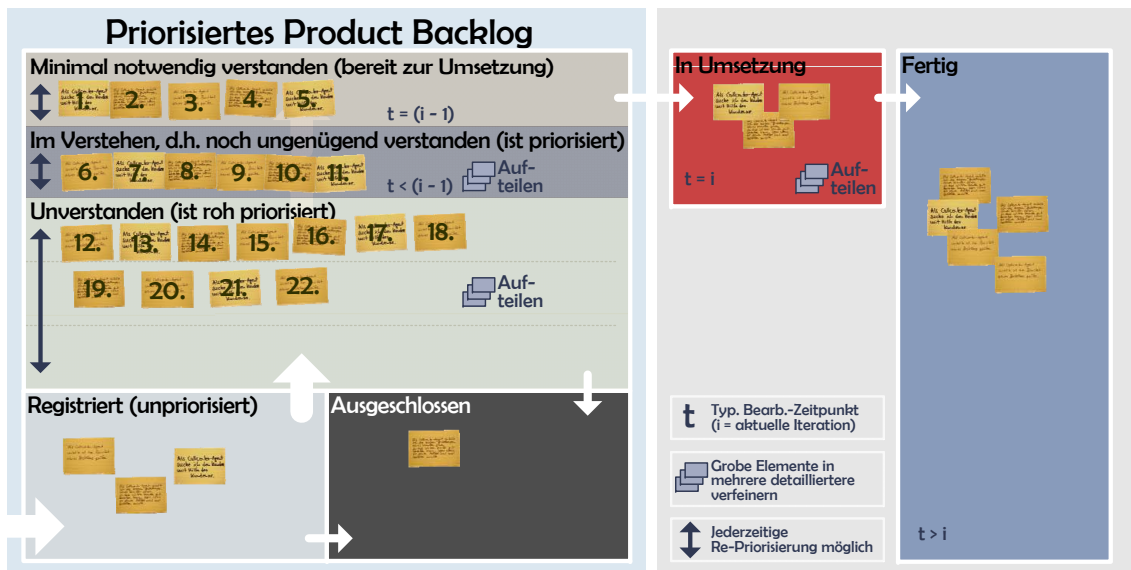




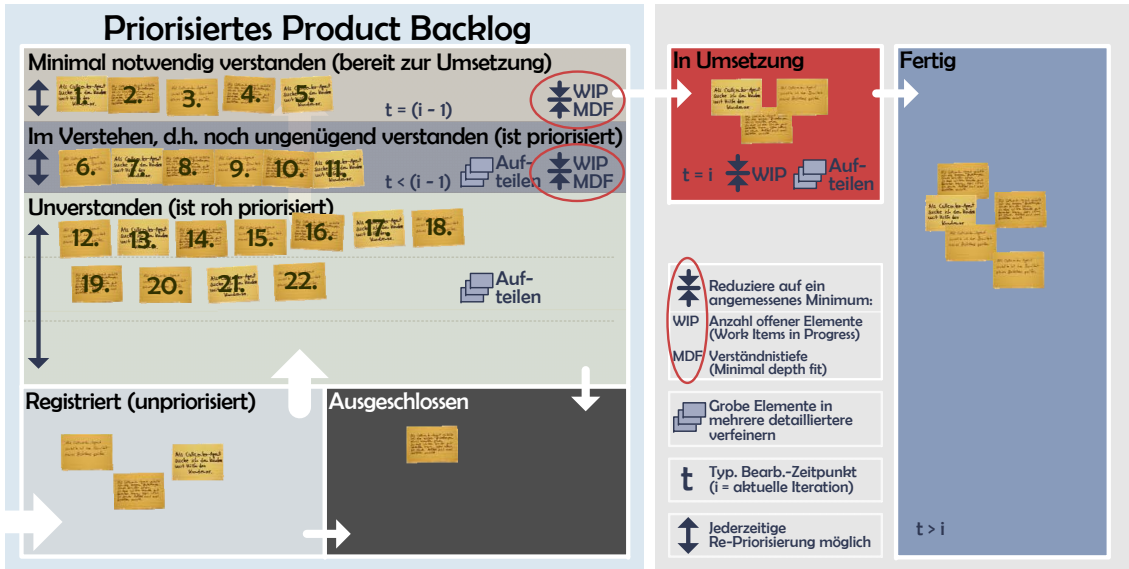
# Das Product Backlog und die Eigenschaften seiner Elemente (3)



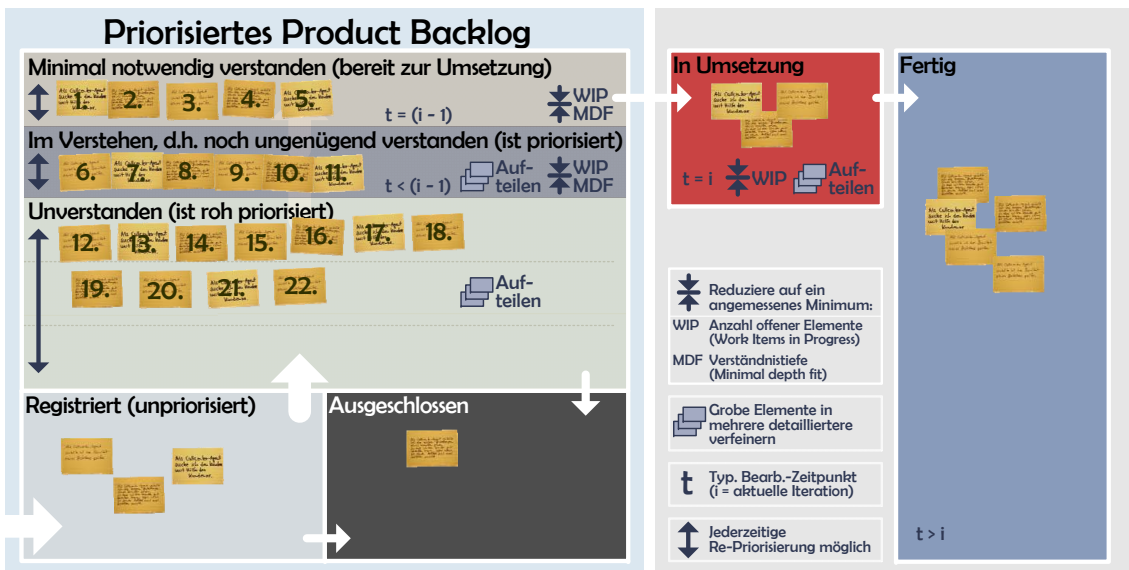
# Das Product Backlog und die Eigenschaften seiner Elemente (4)



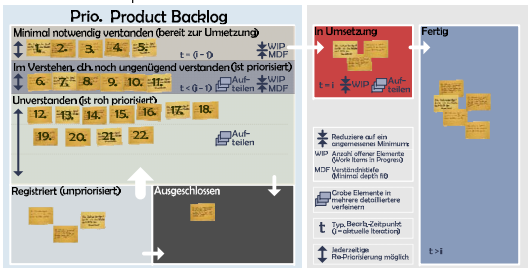
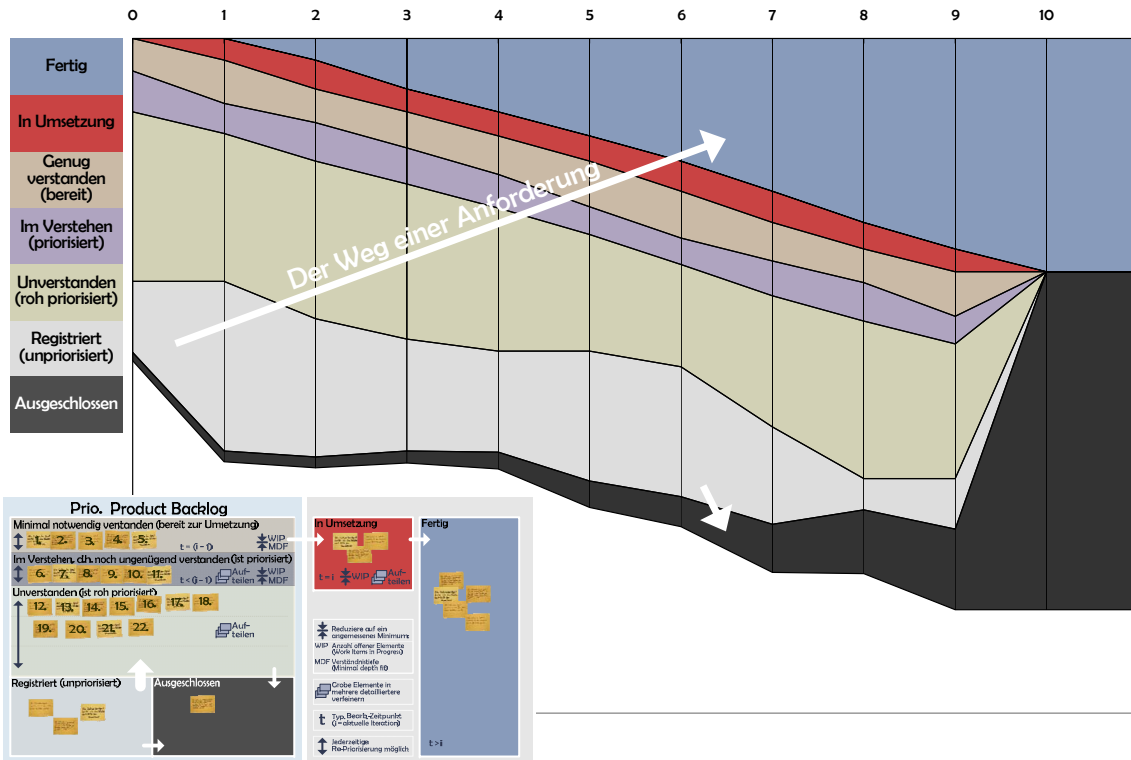
# Das Product Backlog und die Eigenschaften seiner Elemente



# Das Product Backlog und die Eigenschaften seiner Elemente



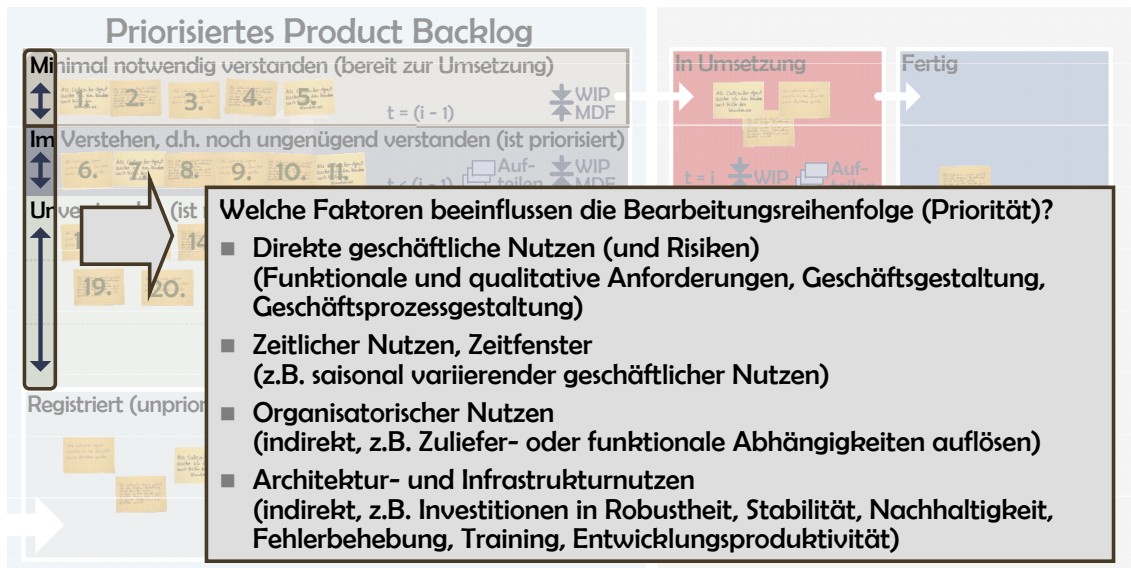
# Entwicklung des Product Backlog



## Wann ist eine Anforderung genug verstanden?



# Wann ist eine Anforderung genug verstanden?

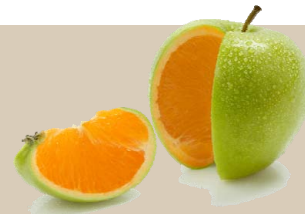


## Kernaussage

Die erfahrenen RE-ler sollten die gezielte Unvollkommenheit als notwendigen Optimierungsschritt akzeptieren lernen.



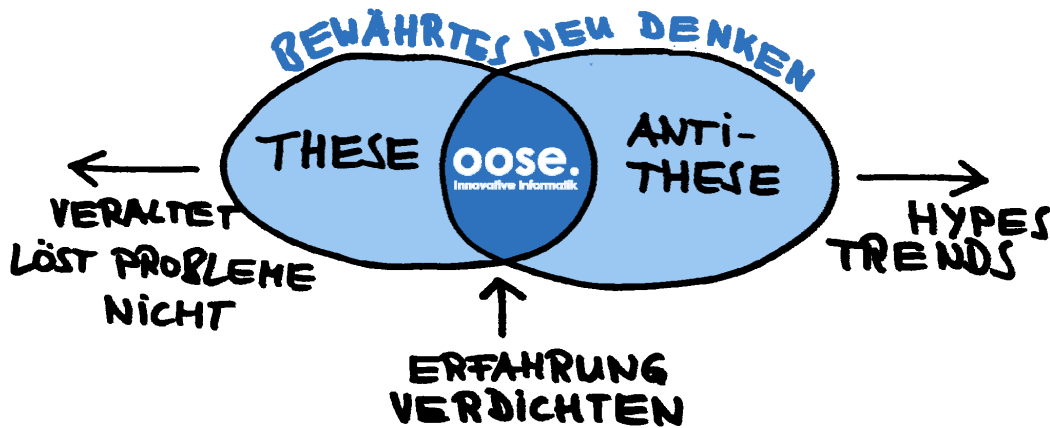
Die erfahrenen Agilisten sollten anspruchsvollere RE-Techniken beherrschen lernen.



Vielen Dank!

oose.  
Innovative Informatik

Jetzt ist Zeit für Fragen.  
Und gerne auch später am Stand von oose!



Oder Sie nutzen unsere Coachings- und Trainings-Dienstleistungen

Unsere Kunden

oose.  
Innovative Informatik



## Integration von textuellen Anforderungen und Modellen – Das Beste aus zwei Welten

Alexander Bösl, Klaus Weber

MID, Enterprise Software Solutions GmbH  
Eibacher Hauptstraße 141  
90451 Nürnberg  
a.boesl@mid.de

**Abstract.** Langjährige Praxiserfahrungen im Projektalltag der modellbasierten Systemanalyse zeigen immer wieder auf das gleiche und wohlbekannte Problem: Viele IT-Projekte beginnen mit der textuellen Aufnahme von Anforderungen, bevor sie mit Hilfe von Modellen analysiert und aufbereitet werden. Dabei wird schnell klar, dass die bisherigen Lösungsansätze die Kluft zwischen textuellen Anforderungen einerseits und Modellen andererseits nicht hinreichend überwinden. Unter Benutzung eines innovativen Konzepts zur Integration von textuellen Anforderungen und Modellen stellen wir mit Hilfe unseres Werkzeugs *Innovator for Requirements* eine dauerhafte Nachverfolgbarkeit von Anforderungen über den gesamten Lebenszyklus dar.

Die Verknüpfung von Anforderungen mit Modellen ermöglicht dabei ein probates Änderungsmanagement und stellt dabei die notwendige Durchgängigkeit in der Applikationsentwicklung sicher. Mit *Innovator for Requirements* kann der Benutzer textuelle Anforderungen in Word definieren, sie im *Innovator-Server* verwalten, Traceability von Anforderungen zu Modellen in verschiedenen Formen visualisieren und so die Auswirkung von Änderungen ganzheitlich analysieren.

### 1 Zusammenfassung

Viele Anwender nutzen die Modellierung zur Analyse, zum Design und zur Codegenerierung von IT-Systemen. Oft sind Modelle dabei der Ort, an dem zum ersten Mal alle Anforderungen und Überlegungen an einer zentralen Stelle zusammengeführt werden. Diese Vorgehensweise kann mit dem Bau eines Geschäftshauses verglichen werden: Alle späteren Nutzer liefern ihre teilweise sehr unterschiedlichen Anforderungen an den Bauträger und den Architekten in Form von Dokumenten, aber auch informell in Gesprächen, per Telefon oder E-Mail. Nehmen wir zum Beispiel die Anforderungen, dass der Treppenaufgang offen zum Eingangsfoyer sein soll, gleichzeitig das Foyer selbst durch die Grundfläche des Gebäudes und die Position der Tiefgarageneinfahrt begrenzt ist. Die Tiefgarage wiederum soll möglichst direkt zur Straße führen und gleichzeitig kurze Wege zum Treppenhaus bieten. All diese Anforderungen treffen vollständig zum ersten Mal im Entwurf des Architekten aufeinander, der als Modell allen Beteiligten



## 5.2 Integration von textuellen Anforderungen und Modellen

vorgelegt wird. Wir würden nie den Bau des Gebäudes in Auftrag geben, nur in der Hoffnung, dass all diese Anforderungen basierend auf unseren Dokumenten und E-Mails richtig berücksichtigt wurden. Wir sehen uns das Modell des Gebäudes an, meist in Form von Grundrissen und Aufrisszeichnungen und beurteilen, ob die Gesamtlösung die Anforderungen in adäquater Form realisiert. Vielleicht nimmt man die Verletzung der einen oder anderen Anforderung für eine gute Implementierung eines anderen in Kauf.

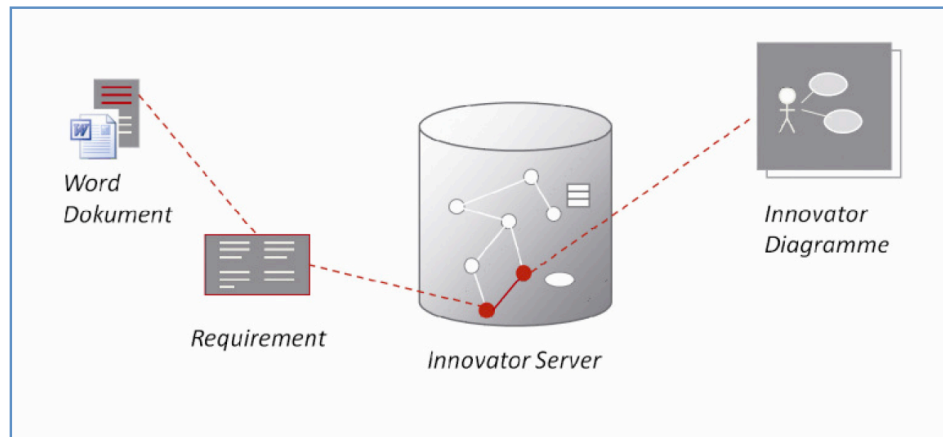


Abbildung 1 Innovator for Requirements verbindet textuelle Anforderungen aus Microsoft Word mit Innovator-Modellen

Bei der modellbasierten Analyse eines IT-Systems passiert genau das Gleiche. Nur sind die Beurteilung der Lösung und die Art und Weise, wie und welche Anforderung wo in die Lösung eingeht, abstrakter und damit erheblich schwieriger. Bei genauer Betrachtung gibt es zwei wesentliche Probleme:

- Fachbereiche sind oft nicht in der Lage ihre Problemstellungen zu modellieren. Viele Informationen aus den Fachbereichen werden deshalb in Form von Dokumenten, die sehr oft informell sind, wie beispielsweise E-Mails, festgehalten. Dadurch entsteht ein Workflow, bei dem der Fachbereichsexperte den Business Analysten oder auch Anforderungs- bzw. Systemanalysten mit immer neuen Dokumenten "füttert", wobei sich die Dokumente selbst im Laufe des Projekts oft ändern bzw. diese variieren.
- Der Fachbereich lernt erst während des Projekts den eigenen Bedarf richtig einzuschätzen. Nehmen wir zum Beispiel einen Stakeholder Review eines Sprints bei agilen Vorgehensweisen. Der Fachbereich besichtigt den aktuellen Stand der Software und stellt fest, dass die implementierten Anforderungen doch noch nicht den eigentlichen Anforderungen entsprechen. In diesem Fall müssen die Requirements abgeändert werden - und jetzt spielt die "Impact Analyse" eine entscheidende Rolle, welche geänderten Requirements zu welchem "Preis" noch umgesetzt werden können. Hier setzt Innovator for Requirements mit der Lösung an, die Modelle mit Requirements zu verknüpfen und zusammen zu verwalten.



Zum einen können Anforderungen in Microsoft Word mit Hilfe eines Innovator-Add-Ins für Word aus dem Fließtext heraus definiert und mit dem Innovator-Server synchronisiert werden (Abbildung 2). Dies funktioniert in beide Richtungen, d. h. wenn der Anforderungsmodellierer Requirements kommentiert, ändert oder neue hinzufügt, dann sind diese Modifikationen für den Autor eines Dokuments im Fachbereich transparent und er kann sie sogar automatisiert nachziehen.

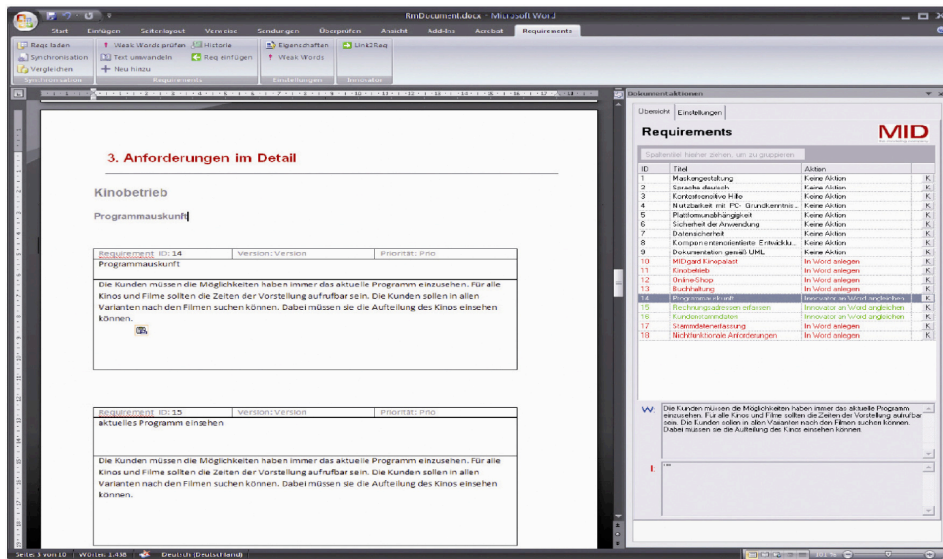


Abbildung 2 Microsoft Word mit Innovator Add-In. Links das Requirements-Dokument, rechts die Requirements aus dem Innovator Server

Für das zweite Problem bietet *Innovator for Requirements* die Möglichkeit, Anforderungen an Modelle, wie z.B. Anwendungsfallmodelle, zu knüpfen und damit genau aufzuzeigen, wo und wie eine Anforderung in das IT-System eingehen wird (siehe Abbildung 3).

## 5.2 Integration von textuellen Anforderungen und Modellen

Abbildung 3 Traceability Matrix in Innovator zeigt die Abhängigkeiten zwischen Anforderungen und Modellelementen

Textuelle Requirements nutzen dabei die Infrastruktur, die Innovator bereits für textuelle SysML-Requirements bereitstellt.

Abbildung 4 zeigt beispielsweise die Verknüpfung von Requirements mit UML-Elementen direkt per Call-out-Notiz im Diagramm.

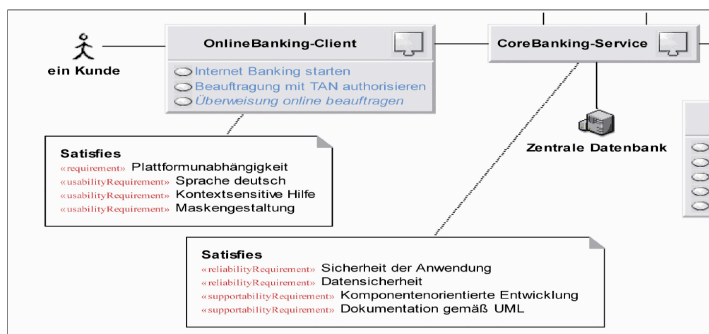


Abbildung 4 Visualisierung der Traceability im UML-Diagramm

Zusätzlich können auch Beziehungen der Requirements untereinander visualisiert werden. Innovator for Requirements bietet darüber hinaus das Word-Add-In und eine Applikation zur Verwaltung von Requirements inklusive Traceability Views zur Impact-Analyse.

Das löst allerdings das Problem noch nicht vollständig: Nehmen wir an, wir verknüpfen ein Requirement mit einem Anwendungsfall (Use Case). Dabei sind mehrere Varianten möglich:

- Der Use Case kann ein funktionales Requirement umsetzen – das ist die einfachste und meist verbreitetste Form.

- Das Requirement kann ein nicht-funktionales Requirement für die Implementierung eines Use Cases sein – in diesem Fall haben Requirement und Use Case eine andere Art der Beziehung.
- Der Use Case selbst kann ein Requirement sein. In diesem Fall wird ein funktionales Requirement nicht durch Text, sondern durch ein kleines Modell selbst ausgedrückt. Das heißt, dass der Use Case selbst eine Priorität und all die anderen Eigenschaften haben muss, die ein Requirement normalerweise hat. Das heißt auch, dass die Benutzeroberfläche für das Requirements Management nicht zwischen text- und modellbeschriebenen Requirements unterscheiden soll.

In Innovator for Requirements kann jedes geeignete Modellelement ein Requirement sein, so dass es dem Benutzer freisteht, ob er Requirements in Form von Modellen oder in Form von Texten beschreibt.

Im Vortrag wird anhand eines konkreten Beispiels der gesamte Prozess von der textuellen Anforderungserhebung in MS Word bis zur Abbildung im Modell anschaulich durchgespielt und nachvollziehbar dargestellt.

## **Geschäftsregelmanagement und Fachmodellierung – SBVR in der Praxis anwenden**

(Extended Abstract)

Dr. Jürgen Pitschke

BCS – Dr. Jürgen Pitschke

Bautzner Str. 79

01099 Dresden

jpitschke@enterprise-design.eu

**Abstract:** Die OMG hat eine Vielzahl von Standards für die Modellierung und Softwareentwicklung hervorgebracht. Neben den populären Standards UML und BPMN ist der Standard „Semantics of Business Vocabulary and Rules (SBVR)“ in den Fokus gerückt.

Der Beitrag stellt einen Ansatz vor, SBVR in der Fachmodellierung zu nutzen. Weiterhin wird die Verbindung von Geschäftsregelanalyse, Anforderungsanalyse, und Geschäftsprozessmodellierung diskutiert. Abschließend werden Möglichkeiten der Werkzeug-Unterstützung des Vorgehens aufgezeigt.

### **1 Semantics of Business Vocabulary and Rules (SBVR)**

Der OMG Standard „Semantics of Business Vocabulary and Rules“ standardisiert ein Metamodell für die Entwicklung semantischer Modelle für die Darstellung eines Geschäftsvokabulars und von Geschäftsregeln. *Nicht* Gegenstand des Standards ist die Definition einer einzelnen Sprache oder Diagrammkonvention für die Darstellung von Vokabular und Regeln. Ebenso ist eine Methode nicht im Standard enthalten.

Der Standard dient der Fachmodellierung und nicht der Beschreibung einer Implementation in einem IT-System. In [SBVR08] heißt es dazu: „Business models, including the models that SBVR supports, describe businesses and not the IT systems that support them.“

Die systematische Darstellung von Begriffen, Fakten und Geschäftsregeln erfährt angesichts der wachsenden Bedeutung des Themas „Compliance“ zunehmende Beachtung. Nur mit einem systematischen Geschäftsregelmanagement lässt sich nachweisen, dass gesetzliche Vorgaben und Regularien eingehalten werden. Gleichzeitig resultiert die explizite Betrachtung von Geschäftsregeln in besseren Unternehmensmodellen. Besser bedeutet vollständigere Unternehmensmodelle, bessere Qualität und bessere Überprüfbarkeit der Modelle.

Verkürzt gesagt folgt der SBVR-Standard dem „Mantra der Geschäftsregeln“: „Regeln basieren auf Fakten. Fakten bauen auf Konzepten auf. Konzepte werden durch Begriffe repräsentiert“.

Wichtiges Grundprinzip ist also die Trennung von Bedeutung (meaning) und Darstellung (expression).

Grundbaustein ist ein Vokabular, das durch Begriffe (Terms) und Fakttypen (Fact Types) gebildet wird. Abbildung 1 zeigt einen Ausschnitt aus dem Metamodell von SBVR zur Beschreibung der Bausteine eines Vokabulars.

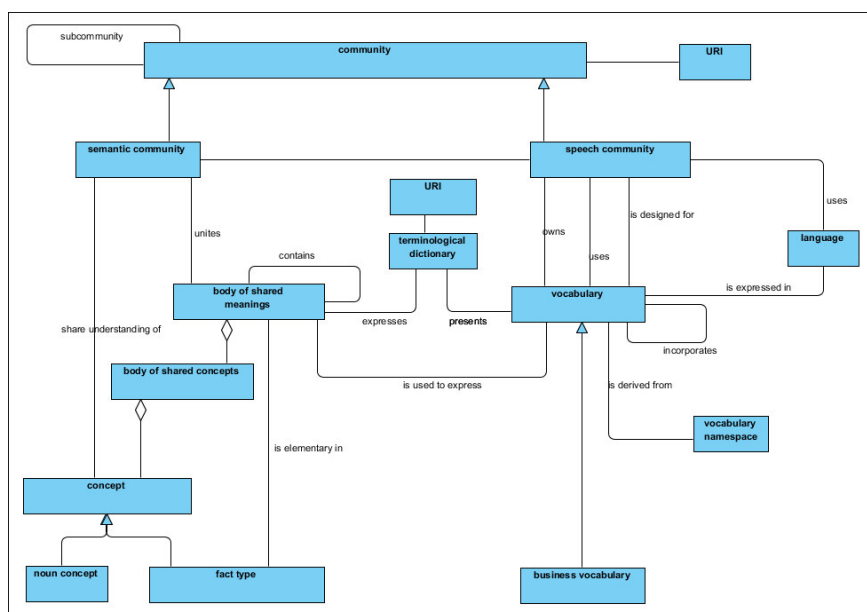


Abbildung 1: Metamodell Vokabular nach [SBVR08]

Die Trennung von Bedeutung und Darstellung gestattet es einem Konzept mehrere Begriffe zuzuordnen und „Communities“ zu definieren. Eine Community ist eine Gruppe von Menschen, die in der täglichen Kommunikation dasselbe Vokabular benutzen. Verschiedene Communities nutzen verschiedene Begriffe für dasselbe Konzept.

Fakttypen definieren Beziehungen zwischen Konzepten und gehen damit über die Definition eines Glossars hinaus. Fakttypen definieren Verbkonzepte in einem Unternehmensmodell. Fakttypen und Fakten können textuell oder grafisch dargestellt werden.

### 5.3 Geschäftsregelmanagement und Fachmodellierung

Beispiele für Fakttypen:

Nutzer berichtet Incident.  
Nutzer berichtet Incident für Gerät.  
Land benutzt Währung.

Beispiele für Fakten:

Deutschland benutzt Euro.  
Schweiz benutzt Schweizer Franken.  
UK benutzt Pound Sterling.

Die Definition der Substantiv-Konzepte und das Faktenmodell sind Grundlage für die Definition von Geschäftsregeln und anderen Arten von „Führung“ (Guidance). Abbildung 2 zeigt das Metamodell „Guidance“ des SBVR-Standards.

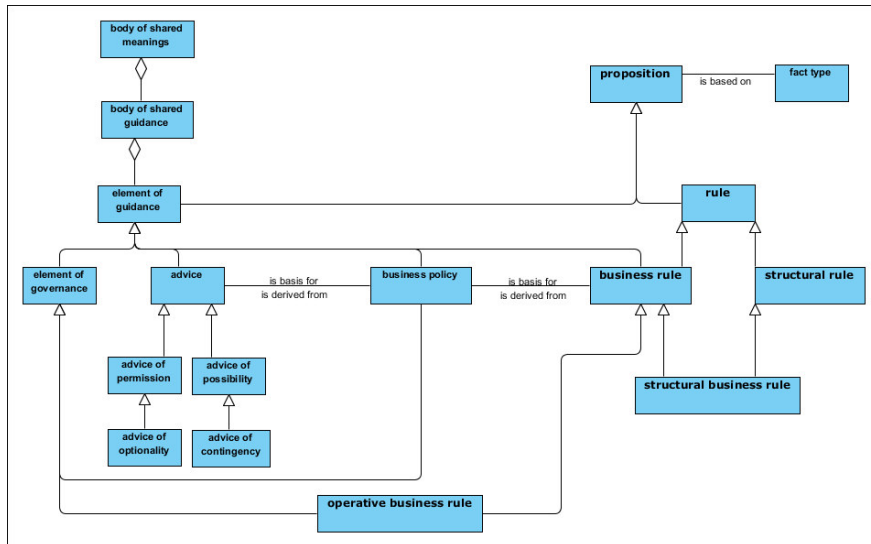


Abbildung 2: Metamodell "Guidance" nach [SBVR08]

Führung wird durch Geschäftsregeln und Empfehlungen beschrieben. Beide basieren auf der Geschäftsstrategie des Unternehmens.

Geschäftsregeln sind dabei Regeln, die sich „in der Verantwortung des Geschäfts“ befinden. Geschäftsregeln schränken immer einen Freiheitsgrad im möglichen Handeln ein.

Beispiele für Geschäftsregeln:

Eine Bestellung muss einem Kunden zugeordnet sein.  
Ein Kunde darf eine Bestellung nur platzieren, wenn der Kunde ein Konto besitzt.

Neben Regeln werden in SBVR Empfehlungen als Elemente der Führung betrachtet. Empfehlungen schränken keinen Freiheitsgrad ein, stellen jedoch ein gewünschtes Verhalten klar.

Beispiele für Empfehlungen:

Eine Kreditprüfung braucht für einen Gold-Kunden nicht durchgeführt werden.  
Eine Kreditprüfung braucht nicht ausgeführt werden, wenn der Bestellwert weniger als 1.000 € beträgt.

## 2 Geschäftsregeln in natürlicher Sprache darstellen

Die Diskussion der Darstellung von Vokabular und Geschäftsregeln erfolgt oft vorrangig im Kontext von Rule-Engines. Schwerpunkt ist dabei die Implementation von Geschäftsregeln, die meist in Form von „If ... then ...“ Statements dargestellt werden.

Dabei werden mehrere Tatsachen übersehen:

- Bevor Regeln in einer Rule-Engine implementiert werden können, müssen sie im Fachmodell erfasst und systematisiert werden.
- Nicht alle Geschäftsregeln lassen sich in der „If ... then ...“-Form darstellen.
- Nicht alle Geschäftsregeln werden in einer Regelmaschine implementiert.
- Fachanwender kommunizieren untereinander in vielen verschiedenen Formen. Geschäftsregeln sind wichtiger Bestandteil der täglichen Kommunikation.
- Menschliche Kommunikation ist nicht immer mathematisch-logisch formalisierbar.

Wichtigstes Ziel der Darstellung von Geschäftsregeln auf der Fachebene ist die Kommunikation zwischen Fachanwendern, um ein einheitliches Verständnis des Unternehmens über Abteilungen und Prozesse hinweg zu erreichen. Das schließt auch die Darstellung von Regeln als Vorlage für die Implementation ein.

### 2.1 RuleSpeak® Deutsch – Geschäftsregeln in natürlicher deutscher Sprache

Im Anhang F des SBVR-Standards ist die RuleSpeak® - Notation für die Darstellung von Geschäftsregeln in natürlicher englischer Sprache enthalten. Dieser Ansatz wurde von Ron Ross und seinen Mitarbeitern über viele Jahre entwickelt. Er gestattet die Darstellung von Geschäftsregeln in reglementierter, natürlicher englischer Sprache für die Kommunikation zwischen Fachanwendern. Einen Überblick ist in [Ross09] zu finden.



Der Ansatz wurde vom Autor in 2009 gemeinsam mit Ron Ross ins Deutsche übertragen. Damit steht ein Ansatz zur Darstellung von Geschäftsregeln in natürlicher deutscher Sprache zur Verfügung (siehe [Pit09a], [Pit09b], [Pit09c]).

Grundlage des Ansatzes sind folgende Regeln:

- Jede Regel muss ein Regelschlüsselwort enthalten.
- Regelschlüsselwörter sind
  - „muss“
  - „darf nicht“
  - „nur“ (nur, wenn).
- Jede Regel muss mit dem Regelsubjekt beginnen.
- Eine Empfehlung muss ein Empfehlungsschlüsselwort enthalten.
- Empfehlungsschlüsselwörter sind
  - „kann“
  - „braucht nicht“

### 2.2 Geschäftsregelanalyse mit RuleSpeak®

Neben den Basis-Richtlinien existieren weitere Vorgaben für die Darstellung von Geschäftsregeln. Folgt man diesen Richtlinien entsteht eine Methode zur Darstellung und Verwaltung von Geschäftsregeln. Der Ansatz gibt zugleich eine Anleitung zur textuellen Analyse von Regelausdrücken. Es erfolgt eine Normalisierung von Regeln. Durch diese Normalisierung ist eine heuristische Überprüfung der Konsistenz und der Regelmenge möglich. Die folgenden Beispiele illustrieren diese Herangehensweise.

Beispiel 1: Ein Geschäftsausdruck beginnt mit dem Subjekt der Regel.

Regel: *Ein Kunde darf eine Abhebung nur ausführen, wenn das Konto aktiv ist.*

Es ist zu hinterfragen, ob „Kunde“ das Subjekt der Regel ist. Gilt die Regel nur für den Kunden? Gilt die Regel auch für andere Akteure? Was ist der Gegenstand des Regelausdrucks?

Revidierte Regel: *Eine Abhebung für ein Konto darf nur ausgeführt werden, wenn das Konto aktiv ist.*

Beispiel 2: Die Form eines Geschäftsausdrucks muss der Form der zugrunde liegenden Fakten folgen.

Fakten: *Kunde platziert Bestellung*  
*Kunde besitzt Konto*

Regel: *Ein Kunde darf eine Bestellung nur platzieren, wenn der Kunde ein Konto besitzt.*

Fehlen die im Regelausdruck benutzten Konzepte und Fakttypen, so ist das Vokabular entsprechend zu ergänzen. Hier bietet sich ein iteratives Vorgehen bei der Entwicklung von Regelmodellen an.

Regel: *Eine Bestellung darf nicht verschickt werden, wenn die ausstehenden Zahlungen den Kreditrahmen übersteigen.*

Dieser Regelausdruck ist missverständlich. Es ist unklar, worauf sich die Begriffe „ausstehende Zahlungen“ und „Kreditrahmen“ beziehen. Der Regelausdruck wird daher revidiert, die fehlenden Begriffe und Fakten werden im Vokabular ergänzt.

Fakten: Kunde platziert Bestellung  
Kunde hat Kreditrahmen (Eigenschaft)  
Kunde besitzt Konto  
Konto hat ausstehende Zahlungen (Eigenschaft)

Regel: *Eine Bestellung darf nicht versandt werden, wenn die ausstehenden Zahlungen des Kontos, das der Kunde besitzt, der die Bestellung platziert, den Kreditrahmen des Kunden übersteigt.*

Beispiel 3: Regeln sollen keine eingebetteten Zahlen, Berechnungen oder Regeln beinhalten:

Regel: *Die Selbstbeteiligung des Kunden für eine bestätigte Forderung muss berechnet werden als 25% der Forderungssumme, wenn die gesamte Selbstbeteiligung der Familie 500 € in dem entsprechenden Jahr nicht übersteigt.*

Der Ausdruck beinhaltet den numerischen Schwellwert „500 €“. Derselbe Schwellwert wird wahrscheinlich in weiteren Geschäftsregelausdrücken auftreten. Wenn der Schwellwert sich ändert, muss er in allen Ausdrücken geändert werden. Werden nicht alle Werte geändert oder erfolgt die Änderung nicht zur selben Zeit, sind Inkonsistenzen die Folge.

Neue Regel: *Die maximale Selbstbeteiligung pro Familie und Jahr muss 500 € betragen.*

Revidierte Regel: *Die Selbstbeteiligung des Kunden für eine bestätigte Forderung muss berechnet werden als 25% der Forderungssumme, wenn die gesamte Selbstbeteiligung der Familie die maximale Selbstbeteiligung pro Familie und Jahr nicht übersteigt.*

Durch Werkzeuge wie die Lösung „RuleXpress“ wird die systematische Umsetzung dieses Ansatzes praktisch unterstützt.

### 3 Geschäftsregelmodell und Unternehmensmodell

Vokabular und Geschäftsregeln sind essentiell für die Gestaltung von Unternehmensmodellen. Ein definiertes Vokabular ist Grundlage für jedes (Teil-)Modell in einem Unternehmensmodell. Besonders eng sind Geschäftsregeln mit Geschäftsprozessmodellen verbunden. Aber auch zu anderen Modellen, wie dem Anforderungsmodell, bestehen enge Zusammenhänge.

#### 3.1 Vokabular und Geschäftsprozessmodelle

Um die Beziehungen zwischen Geschäftsregelmodell und Geschäftsprozessmodell zu analysieren, müssen beide auf demselben Vokabular aufbauen. Eine gängige Praxis der Benennung von Aktivitäten in einem Geschäftsprozessmodell ist die Verwendung von <<Substantiv>><<Verb>>-Konstrukten. Z.B. wird eine Aktivität in einem Helpdesk-Prozess mit „Fälligkeit bestimmen“ bezeichnet. Erste Voraussetzung ist, dass der Begriff „Fälligkeit“ im Vokabular definiert ist.

Wenden wir die SBVR-Konzepte an, erkennen wir, dass diese Namensgebung unvollständig ist und zu Missverständnissen führen kann. Im Beispiel ist nicht klar, worauf sich „Fälligkeit“ bezieht. Die Benennung soll daher ebenfalls den referenzierten Fakt korrekt wiedergeben. Statt „Fälligkeit bestimmen“ ist die Aktivität mit „Fälligkeit des Incidents bestimmen“ zu benennen. Der zugrundeliegende unäre Fakt lautet „Incident hat Fälligkeit“ (Eigenschaft).

Eine Überprüfung der verwendeten Begriffe sollte durch das Modellierungswerkzeug unterstützt werden.

#### 3.2 Geschäftsregeln und Geschäftsprozessmodelle

Geschäftsregeln und Geschäftsprozesse sind zwei Seiten derselben Medaille. Geschäftsregeln beschreiben Aktivitäten und Elemente des Geschäftsprozesses im Detail.

Es ist wichtig, Geschäftsprozesse und Geschäftsregeln unabhängig voneinander zu analysieren und darzustellen, da es sich um unabhängige Konzepte handelt. Prozesse sind vergleichsweise stabil, Regeln ändern sich schneller und öfter als Prozesse. Zugleich sind beide eng verknüpft.

Abbildung 3 zeigt einen Ausschnitt eines in der Praxis typischen Prozessmodells. Dabei wurden Geschäftsregeln implizit im Prozessmodell umgesetzt. Das Modell zeigt einen Entscheidungsbaum, der eigentliche Inhalt des Prozessmodells geht verloren. Änderungen in der Regelmenge erfordern eine Änderung des Prozessmodells.

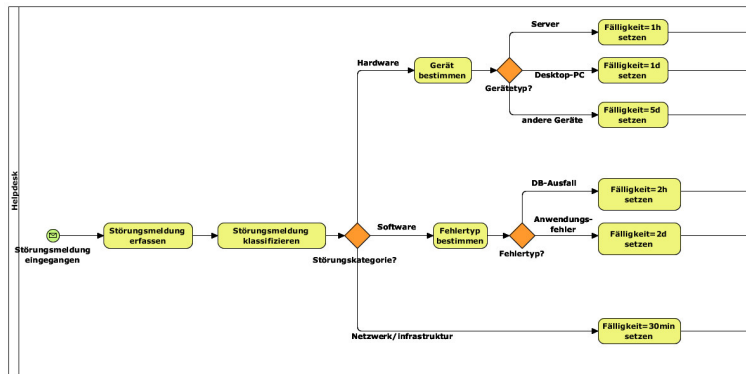


Abbildung 3: Ausschnitt aus einem Prozessmodell

Abbildung 4 zeigt das revidierte Prozessmodell. Es zeigt ausschließlich den fachlichen Inhalt des Prozesses. Die Regelmenge wird der entsprechenden Aktivität zugeordnet. Eine Änderung der Regelmenge führt nicht mehr zu einer Änderung des Prozessmodells.

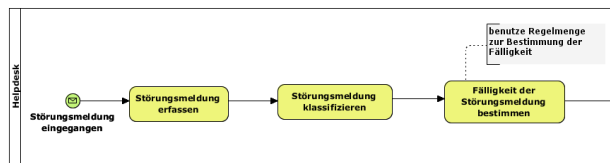


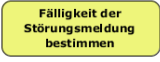
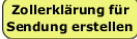


Abbildung 4: Revidiertes Prozessmodell

Für die Umsetzung dieses Ansatzes ist es notwendig, Geschäftsregelmenen mit Aktivitäten in einem Prozessmodell verbinden zu können.

Obiges Beispiel zeigt eine mögliche Verbindung zwischen Geschäftsregeln bzw. Gruppen von Geschäftsregeln und Elementen in einem Geschäftsprozessmodell. Eine Gruppe operativer Regeln wurde einer Aktivität im Geschäftsprozessmodell zugeordnet. Dieselbe Gruppe von Regeln kann natürlich auch für weitere Aktivitäten relevant sein.

Eine Klassifizierung des Regelinhalts mit Bezug auf die Elemente eines Geschäftsprozessmodells zeigt weitere Möglichkeiten auf. Tabelle 1 gibt einen Überblick:

### 5.3 Geschäftsregelmanagement und Fachmodellierung

Element im Geschäftsprozessmodell	Regelart	Beispiel-Element	Beispiel-Regel
Aktivität	Berechnungsregel Datenregel		Die Fälligkeit eines Incidents muss auf 3 Stunden gesetzt werden, wenn das vom Incident betroffene Gerät ein Server ist.
Aktivität	Steuerungsregel		Die Aufgabe „Zollerklärung abgeben“ darf nur ausgeführt werden, wenn alle folgenden Informationen vorliegen: - Warenbeschreibung - Herkunftsland - Ursprungszertifikat
Sequence Flow	Prozess-Trigger		Die Lösung eines Incidents muss dokumentiert werden, wenn der Incident als „gelöst“ gekennzeichnet wird.
Pool / Lane	Zuordnungsregel		Alle Aufgaben des Teilprozesses „Aufnahme des Incidents“ müssen durch denselben Bearbeiter ausgeführt werden.
Event / Gateway	Strukturelle Regel Ableitungsregel		Ein Kunde muss als Neukunde angesehen werden, wenn der Kunde keine Bestellung platziert hat.

Durch die Verbindung von Geschäftsregeln und Geschäftsprozessmodell, lassen sich Geschäftsprozesse vollständiger definieren. Das Prozessmodell wird stabiler gegen Veränderungen.

Die Darstellung von Geschäftsregeln nach SBVR beinhaltet nicht die Darstellung der Durchsetzung (Enforcement) der Regeln. Hier können Geschäftsprozessmodelle Inhalt zu einem Regelmodell beitragen. Das Prozessmodell kann mögliche Reaktionen auf eine Verletzung einer Geschäftsregel zeigen.

Es bestehen Zusammenhänge zwischen Geschäftsregeln und weiteren Artefakten in einem Unternehmensmodell, z.B. zwischen Anforderungen und Geschäftsregeln.

## Literaturverzeichnis

- [BPMN09] Business Process Model and Notation (BPMN), FTF Beta 1 for Version 2.0, OMG, August 2009, OMG Document Number: dtc/2009-08-14
- [Halp06] Halpin, Terry, Objectification of Relationships in Advanced Topics in Database Research, Volume 5, Seiten 106-123, Idea Group Pub, 2006
- [Pit09A] Pitschke, Jürgen, RuleSpeak<sup>®</sup> Satzformen, Business Rules in natürlich sprachlichem Deutsch spezifizieren, 2009
- [Pit09B] Pitschke, Jürgen, RuleSpeak<sup>®</sup> Guidelines- Grundlagen, Geschäftsregeln in natürlich sprachlichem Deutsch darstellen – was man tun sollte und was nicht, 2009
- [Pit09C] Pitschke, Jürgen, RuleSpeak<sup>®</sup>, Kommentare zu den Basisdokumenten, Version 1.2
- [Ross09] Ross, Ronald G. , Business Rule Concepts, Third Edition, Business Rule Solutions, LLC., 2009
- [SBVR08] Semantics of Business Vocabulary and Business Rules (SBVR), v1.0, OMG Document Number: formal/2008-01-02



# Geschäftsregelmanagement und Fachmodellierung – SBVR in der Praxis anwenden

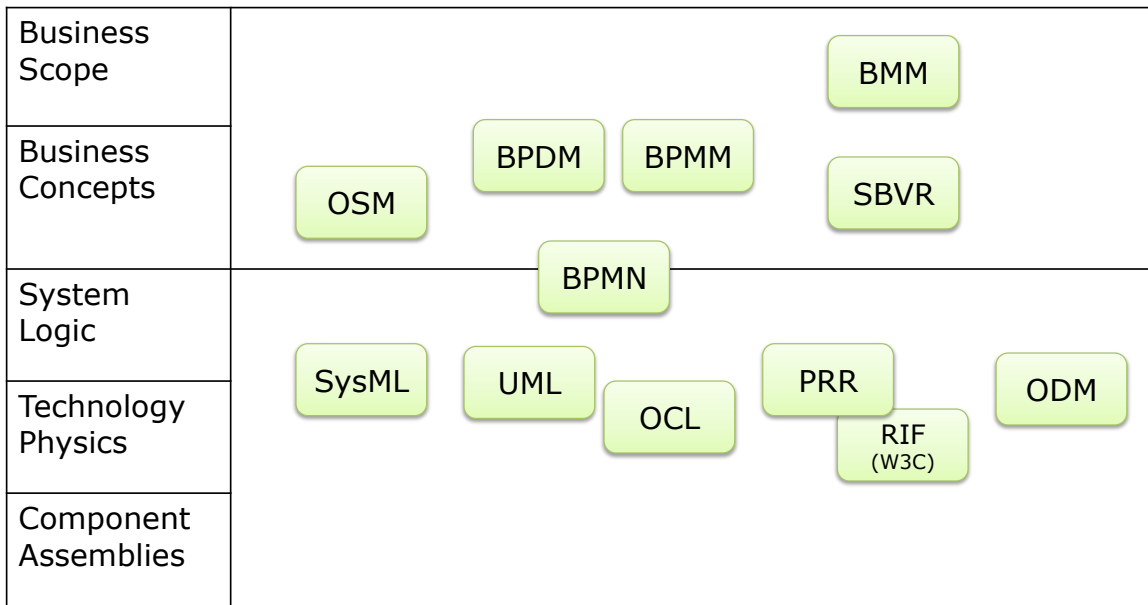
SEE 2010, Köln  
3. Mai 2010



Dr. Jürgen Pitschke  
BCS – Dr. Jürgen Pitschke  
[www.enterprise-design.eu](http://www.enterprise-design.eu)



## OMG-Standards und Unternehmensmodelle





## Semantics of Business Vocabulary and Rules - Grundzüge

- Semantics of Business Vocabulary and Rules – Grundzüge
- RuleSpeak® - Deutsch
- Vokabular, Geschäftsregeln, Geschäftsprozesse und Unternehmensmodelle

## SBVR – Übersicht

- Semantics of Business Vocabularies and Rules
  - Unterstützt Entwicklung / Spezifikation von
    - Geschäftsvokabularien (Begriffe und Definitionen, Fakttypen)
    - Geschäftsregeln (und anderen Führungselementen)
  - ist ein Metamodell für die Entwicklung semantischer Modelle von Geschäftsvokabularien und Geschäftsregeln
  - ist keine Sprache zur Definition von Regeln
  - ist keine Methode, standardisiert keine Methode
  - enthält keine Abbildung von Geschäftsregeln auf Produktionsregeln für die Ausführung

## Das Business Rule – Mantra

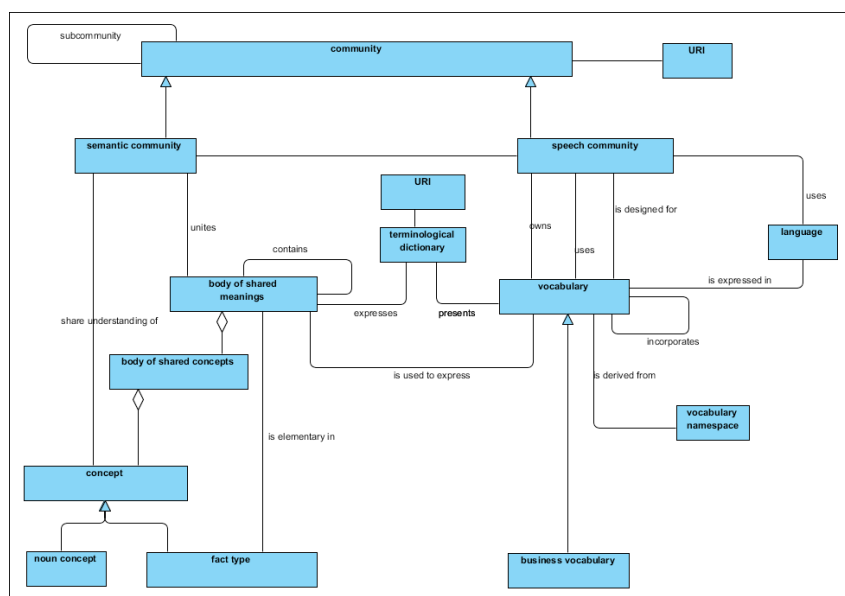
Regeln basieren auf Fakten.



Fakten bauen auf Konzepten auf.

Konzepte werden durch Begriffe repräsentiert.

## SBVR – Metamodell Vokabular



## SBVR – Übersicht

- Geschäftsvokabular:
  - SBVR: A business vocabulary contains all the specialized terms, names, and fact type forms of concepts that a given organization or community uses in their talking and writing in the course of doing business.
- Term
  - Wort oder Wortgruppe, die für die Kommunikation benutzt wird
  - *"a verbal designation of a general concept in a specific field"*

## Fakttypen, Fakten und Faktenmodell

- Fakttypen etablieren Beziehungen (Assoziationen) zwischen Konzepten
  - Z.B.  
Nutzer berichtet Incident  
(Nutzer hat Incident berichtet)  
Ware ist verladen in Container  
Land benutzt Währung
- Fakten beschreiben Beziehungen zwischen individuellen Konzepten
  - Z.B.  
Deutschland benutzt Euro  
Schweiz benutzt Schweizer Franken



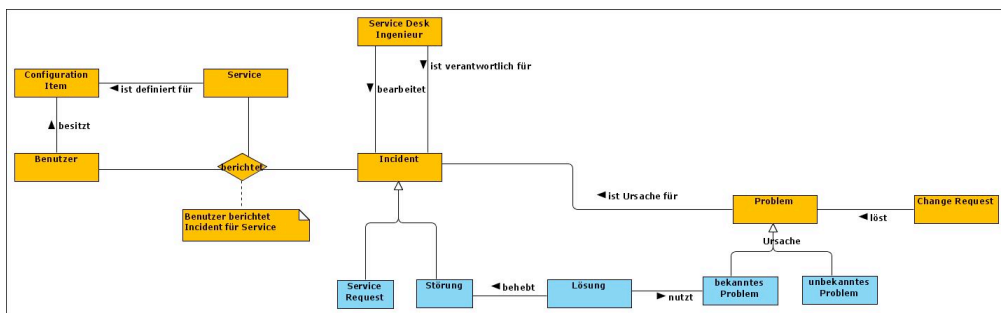
# Fakttypen

Fakttyp	Konstrukt	Beispiel
Unärer Fakttyp	Subjekt Prädikat	Service-Desk-Ingenieur <u>ist verfügbar</u> .
Binärer Fakttyp	Subjekt Prädikat Objekt	Anwender <u>berichtet</u> Incident.
Ternärer Fakttyp	Subjekt Prädikat Objekt Präpositionales Attribut	Anwender <u>berichtet</u> Incident für Asset.
Quaternärer Fakttyp	Subjekt Prädikat Objekt Attribut Adverbiale Bestimmung	Anwender <u>berichtet</u> Incident für Asset am Datum.

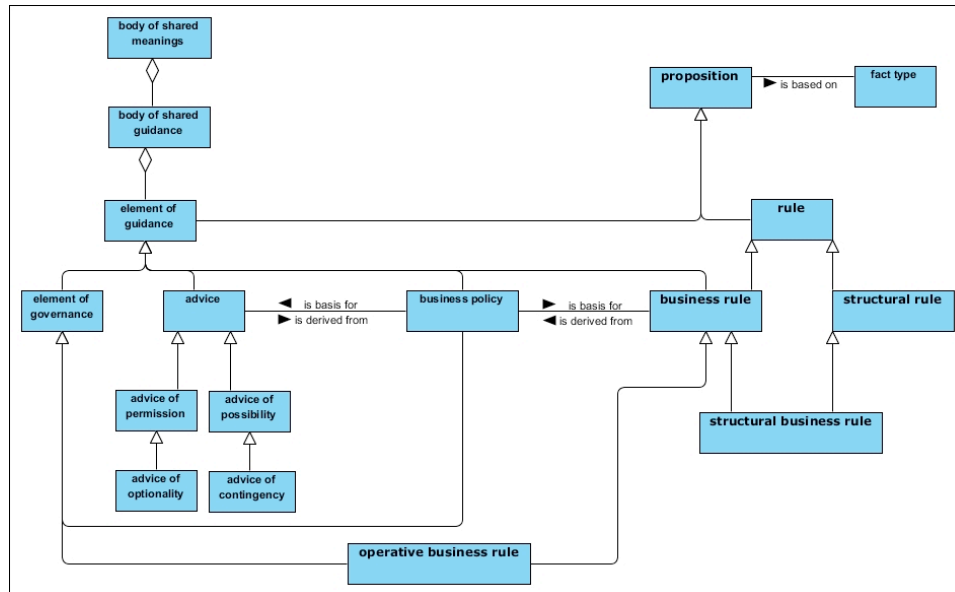


# Faktenmodell

- Darstellung von Fakttypen:
  - Textuell: Anwender berichtet Incident
  - Visuell



## SBVR – Metamodell Guidance



## SBVR – Business Rules

- Führungsarten:
  - Geschäftsregeln
  - Empfehlungen (Klarstellungen)
- Was ist eine Geschäftsregel?
  - Regel, die in der Zuständigkeit des „Geschäfts“ ist
    - Das Gravitationsgesetz gehört nicht zur Zuständigkeit des Unternehmens.
    - Ein (legislatives) Gesetz gehört nicht in die Zuständigkeit des Unternehmens.  
Die Umsetzung des Gesetzes ist dagegen in der Zuständigkeit des Unternehmens.
  - Eine Regel schränkt immer einen Freiheitsgrad ein



## RuleSpeak® - Deutsch

- Semantics of Business Vocabulary and Rules – Grundzüge
- **RuleSpeak® - Deutsch**
- Vokabular, Geschäftsregeln, Geschäftsprozesse und Unternehmensmodelle



## Was ist RuleSpeak®?

- *RuleSpeak* ist eine Zusammenfassung praktischer Richtlinien für
  - Die Darstellung von Geschäftsregeln in wohl- strukturiertem Deutsch
  - Die Vermeidung häufiger Fehler bei der Darstellung von Regeln und Empfehlungen
- “Business models, including the models that SBVR supports, describe businesses and not the IT systems that support them.”
  - Source: Semantics of Business Vocabularies and Business Rules (SBVR) V1.0, OMG, January 2008

## Was ist RuleSpeak®?

- RuleSpeak definiert Satzformen.
  - Eine *Satzform* ist ein elementares Muster oder eine elementare Vorlage in natürlicher Sprache. Diese wird benutzt, um eine bestimmte Art einer Geschäftsregel in einer konsistenten und wohlorganisierten Art und Weise auszudrücken.  
Jede Satzform dient einer bestimmten Art der Führung.

## Geschäftsregeln und Schlüsselworte

- Eine Geschäftsregel reduziert immer einen Freiheitsgrad.
- Jede Geschäftsregel wird unter Nutzung eines der folgenden Schlüsselworte für Regeln dargestellt:



- muss
- darf nicht
- nur



## Schlüsselwort für Regeln – „muss“

- Beispiele:



– Regel: *Für eine Bestellung ist es unbedingt erforderlich, den Kunden anzuzeigen, der sie platziert hat.*



– Revidierte Regel: *Eine Bestellung muss den Kunden anzeigen, der sie platziert hat.*



## Schlüsselworte für Regeln

- Beispiele:



– Regel: *Eine Bestellung darf nicht verschickt werden, wenn die ausstehenden Zahlungen des Kundenkontos den Kreditrahmen dieses Kunden übersteigen.*



– Regel: *Ein Kunde darf eine Bestellung nur platzieren, wenn dieser Kunde ein Konto besitzt.*



– Regel: *Eine Versandgebühr darf nur für eine Bestellung erlassen werden, die von einem inländischen Kunden platziert wurde.*



## Empfehlungen und Schlüsselworte



- Eine Empfehlung soll genau eines der folgenden Schlüsselworte für Empfehlungen enthalten:
  - „kann“
  - „braucht nicht“
- Eine Empfehlung stellt klar,
  - dass etwas erlaubt ist
  - dass etwas nicht erforderlich ist

## Empfehlungen – Beispiele



- Beispiele:
  - Empfehlung: *Eine Person jeden Alters kann ein Bankkonto besitzen.*
  - Empfehlung: *Eine Kreditprüfung braucht nicht ausgeführt werden, wenn der Bestellwert weniger als 1.000 Euro beträgt.*
  - Empfehlung: *Ein Kunde mit Gold-Status braucht eine Bestellung mit internationalem Ziel nicht in bar bezahlen.*



## Subjekte von Direktiven



- RuleSpeak Direktiven beginnen immer mit dem Subjekt.
  - Was ist der Gegenstand der Regel?
  - Das Subjektiv soll sich immer im Singular befinden.



## Fakten und Regeln

- Die Form des Geschäftsregelausdrucks soll der Form der Fakten folgen.
- Beispiel:
  - Fakten: Kunde platziert Bestellung
  - Kunde besitzt Konto



- Regel: Ein Kunde darf eine Bestellung nur platzieren, wenn der Kunde ein Konto besitzt.

## Agilität und Wiederverwendung

- Eingebettete Werte sind nicht gut.
- Beispiel:



– Regel: *Die Selbstbeteiligung des Kunden für eine bestätigte Forderung muss berechnet werden als 25% der Forderungssumme, wenn die gesamte Selbstbeteiligung der Familie 500 € in dem entsprechenden Jahr nicht übersteigt.*



– Neue Regel: *Die maximale Selbstbeteiligung pro Familie und Jahr muss 500 € betragen.*

– Revidierte Regel: *Die Selbstbeteiligung des Kunden für eine bestätigte Forderung muss berechnet werden als 25% der Forderungssumme, wenn die gesamte Selbstbeteiligung der Familie die maximale Selbstbeteiligung pro Familie und Jahr nicht übersteigt.*

## Agilität und Wiederverwendung

- Eingebettete Bedingungen sind nicht gut.
- Beispiel:



– Regel: *Ein Kunde, dessen ausstehende Zahlungen 1.000 € für jede der drei letzten aufeinanderfolgenden Rechnungen übersteigt, darf keine Bestellung für einen Artikel, dessen Kosten 500 € übersteigen, platzieren.*

– Neue Regel: *Ein Kunde muss als Risikokunde angesehen werden, wenn die ausstehenden Zahlungen 1.000 € für jede der drei letzten aufeinanderfolgenden Rechnungen übersteigt.*

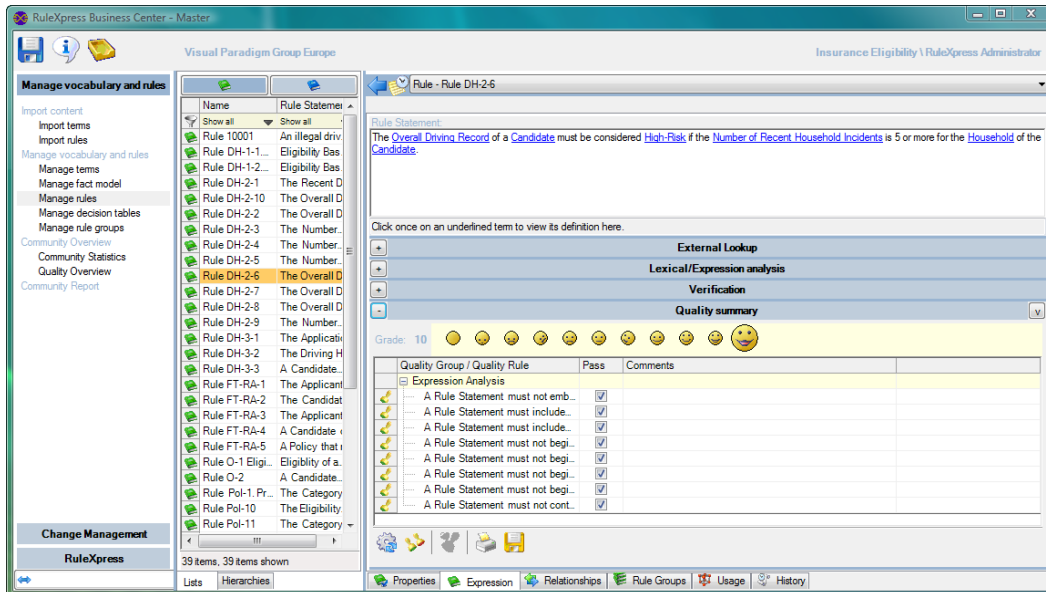


– Neue Regel: *Ein Artikel muss als hochpreisiger Artikel angesehen werden, wenn dessen Kosten 500 € übersteigen.*

– Revidierte Regel: *Ein Risiko-Kunde darf keinen hochpreisigen Artikel bestellen.*



## Werkzeugunterstützung - RuleXpress



## Vokabular, Geschäftsregeln, Geschäftsprozesse und Unternehmensmodelle

- Semantics of Business Vocabulary and Rules – Grundzüge
- RuleSpeak® - Deutsch
- Vokabular, Geschäftsregeln, Geschäftsprozesse und Unternehmensmodelle

## SBVR und visuelle Modellierung

- Wie hängen Vokabular und Regeln mit Geschäftsprozessen, Geschäftsaktivitäten, Anwendungsfällen, Anforderungen und anderen Artefakten zusammen?



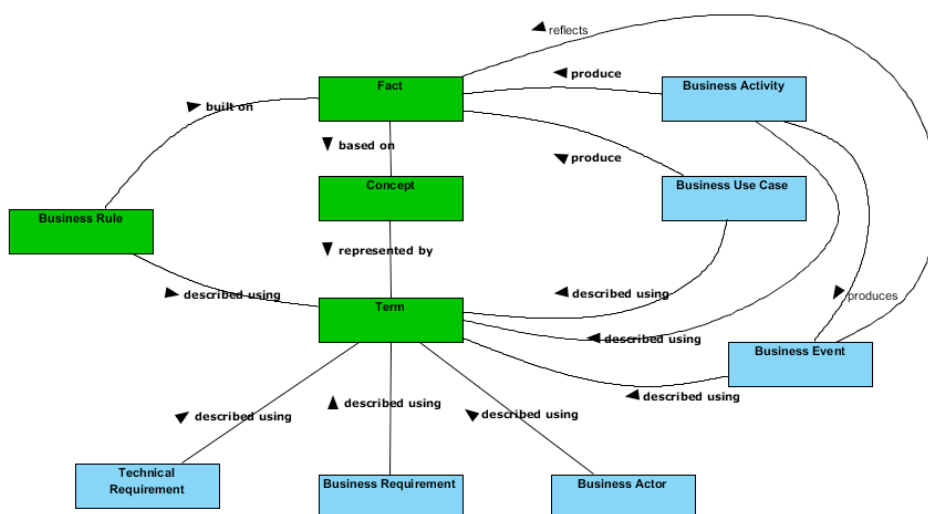
## SBVR und visuelle Modellierung

- Begriffe und Fakten sind Grundlage für JEDES Modell.
  - Definition eines Vokabulars
  - Definition eines Faktenmodells
- Nutzung von Vokabular und Faktenmodell in anderen Modellen
  - Verwendung von Begriffen
  - Namenskonventionen

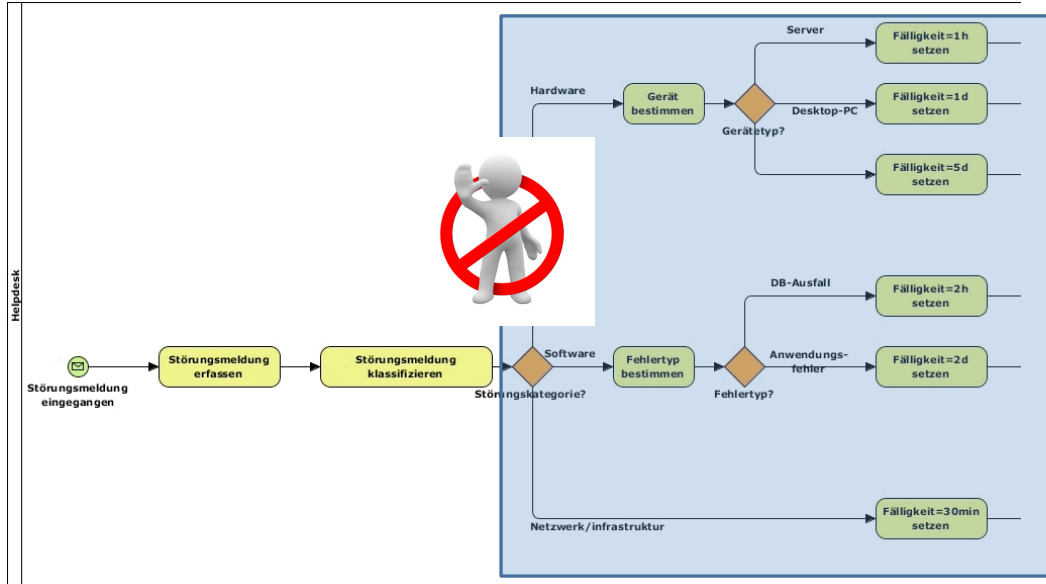
## Vokabular und Geschäftsprozessmodelle

- Übliche Namenskonvention in Geschäftsprozessmodellen:
  - <<Verb>><<Substantiv>>
  - oder
  - <<Substantiv>><<Verb>>
- Die verwendeten Begriffe müssen definiert sein.
- Die verwendeten Verben müssen als Faktsymbol definiert sein.
- Die Namenskonvention ist zu ändern: Die Bezeichnung der Aktivitäten muss dem unterliegenden Fakttyp entsprechen.

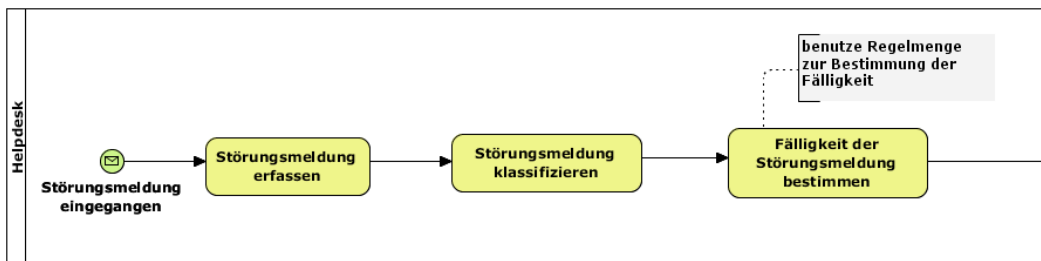
## Artefakte im Modellierungsprozess - I



# Ist das ein Geschäftsprozessmodell?



# Geschäftsprozess reduziert





## Geschäftsregeln und Geschäftsprozesse

Element im GP-Modell	Regelart	Beispiel
Aktivität	Berechnungsregel Datenregel	Die Fälligkeit eines Incidents muss auf 3 Stunden gesetzt werden, wenn das vom Incident betroffene Gerät ein Server ist.
Aktivität	Steuerungsregel	Die Aufgabe „Zollerklärung abgeben“ darf nur ausgeführt werden, wenn alle folgenden Informationen vorliegen: - Warenbeschreibung - Herkunftsland - Ursprungszertifikat
Sequence Flow	Prozess-Trigger	Die Lösung eines Incidents muss dokumentiert werden, wenn der Incident als „gelöst“ gekennzeichnet wird.
Pool/Lane	Zuordnungsregel	Alle Aufgaben des Teilprozesses „Aufnahme des Incidents“ müssen durch denselben Bearbeiter ausgeführt werden.
Event/ Gateway	Strukturelle Regel Ableitungsregel	Ein Kunde muss als Neukunde angesehen werden, wenn der Kunde noch keine Bestellung platziert hat.

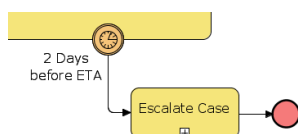
© BCS - Dr. Jürgen Pitschke 2003-2010, www.enterprise-design.eu

Seite 33



## Geschäftsregeln und Geschäftsprozesse

- Regeln sind deklarativ und unabhängig von einem Geschäftsprozess.
- In einem Prozess kann die Reihenfolge für die Anwendung von Regeln definiert werden.
- Regeln werden unabhängig von der Durchsetzung definiert.
- Die Durchsetzung einer Regel kann im Geschäftsprozessmodell dargestellt werden.



© BCS - Dr. Jürgen Pitschke 2003-2010, www.enterprise-design.eu

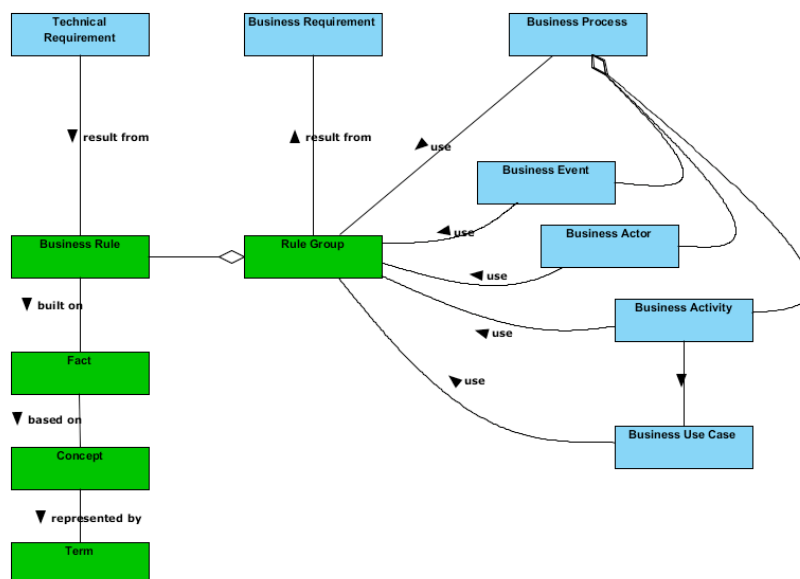
Page 34



## Geschäftsregeln im Unternehmensmodell

- Regeln sind mit weiteren Artefakten im Unternehmensmodell verknüpft:
  - Geschäftsregel / Geschäftsanforderung
  - Geschäftsregel / technische Anforderung
  - Geschäftsregel / Geschäftsanwendungsfall
- Das Vokabular ist Grundlage jedes Modells.

## Artefakte im Modellierungsprozess - II





## Literatur / Quellen

- Ronald G. Ross, Business Rules Concepts, 3<sup>rd</sup> Edition, Business Rules Solutions, LLC
- Dr. Jürgen Pitschke, Ronald G. Ross, RuleSpeak®- Satzformen, Version 1.2
- Dr. Jürgen Pitschke, Ronald G. Ross, RuleSpeak®- Guidelines, Version 1.2
- Dr. Jürgen Pitschke, RuleSpeak®, Kommentare zu den Basisdokumenten, Version 1.2
- Business Process Model and Notation (BPMN), FTF Beta 1 for Version 2.0, OMG, August 2009, OMG Document Number: dtc/2009-08-14
- Semantics of Business Vocabulary and Business Rules (SBVR), v1.0, OMG Document Number: formal/2008-01-02
- Terry Halpin, Objectification of Relationships in Advanced Topics in Database Research, Volume 5, Seiten 106-123, Idea Group Pub, 2006



## Danke!



Dr. Jürgen Pitschke  
[www.enterprise-design.eu](http://www.enterprise-design.eu)  
[jpitschke@enterprise-design.eu](mailto:jpitschke@enterprise-design.eu)

## Nichts ist Beständiger als der Wandel – Unterstützung der Anforderungsanalyse durch Simulation

André Pflüger

SOPHIST GmbH  
Vordere Cramergasse 13  
90478 Nürnberg  
andre.pflueger@sophist.de

**Abstract.** Designentscheidungen basieren auf vielen verschiedenen Faktoren, welche sich im Verlaufe eines Projektes ändern können. Tritt dieser Fall ein, muss der Architekt die getroffenen Entscheidungen erneut validieren und gegebenenfalls anpassen. Der vorgestellte Ansatz unterstützt den Architekten bei dieser Arbeit und ermöglicht ihm eine durchgängig in UML gehaltene Dokumentation seiner Entscheidung, welche gleichzeitig als Grundlage für eine automatische Validierung und Übernahme von Änderungen ins System-Modell dient.

### 1 Motivation

Zwei entscheidende Faktoren bei der Entwicklung eines neuen Systems sind die Kosten und die zur Verfügung stehende Zeit. Für beide ist vor dem eigentlichen Projekt eine Abschätzung notwendig, welche auf wenig detaillierten Informationen beruhen. Die Wissenslücken werden meist mit Erfahrungswerten gefüllt. Unter diesen Voraussetzungen wird ein Architektentwurf erstellt, dessen Güte entscheidend für den Erfolg des Projektes ist.

Im Verlauf des Projektes vervollständigt sich das Bild des zu erstellenden Systems, das Wissen über das System wird detaillierter und es ergeben sich immer wieder neue und angepasste Randbedingungen. Die Architektur muss gegenüber den neuen Randbedingungen validiert und gegebenenfalls angepasst werden. Dazu bedarf es einer guten Dokumentation und der Gewissheit, dass alle Bedingungen bei der Validierung berücksichtigt werden. Die modellbasierte Dokumentation von Systemen in Verbindung mit der Modellierungssprache Unified Modelling Language (UML) ist ein weit verbreitetes und häufig verwendetes Werkzeug. Die Festlegung der zu verwendenden Notationselemente und deren Semantik ermöglicht die Etablierung einer domänenspezifisch einheitlichen Dokumentationssprache, was sich positiv auf das Verständnis und die Verständigung aller Projektbeteiligten auswirkt.

Im Projektalltag ist aber gerade die durchgängige Verwendung einer Dokumentationsprache von der Anforderungsanalyse bis zur Architektur selten bis gar nicht vorzufinden. Auch die Unterstützung des Architekten bei der Wahl einer validen Architektur, vor allem unter der Voraussetzung sich häufig ändernden Bedingungen, lässt zu wünschen übrig.

Wir stellen Ihnen einen Ansatz vor, der eine durchgängige Verwendung der UML als Dokumentationssprache ermöglicht und dem Architekten ein Werkzeug an die Hand gibt, welches die Wahl und Validierung der Architektur mit geringem Aufwand ermöglicht.

## 2 Idee

Die Aufgabe des Architekten ist das Erstellen einer Architektur, welche allen Kundenanforderungen genügt. Um dies sicherzustellen, muss jede Kundenanforderung für eine ausgewählte Architektur validiert werden. Aus den Kundenanforderungen folgen unterschiedliche Untersuchungsziele, z.B. die Auslastung von Prozessoren oder die Übertragungsrate von Kommunikationskanälen.

Einen Überblick über die konsequente Integration der Validierung in die Systementwicklung gibt die Abbildung 1. Ausgangspunkt für den dargestellten Prozess sind die Anforderungen und ein domänenspezifisches Modell (DSM), das im Rahmen der konventionellen Entwicklung entstanden ist.

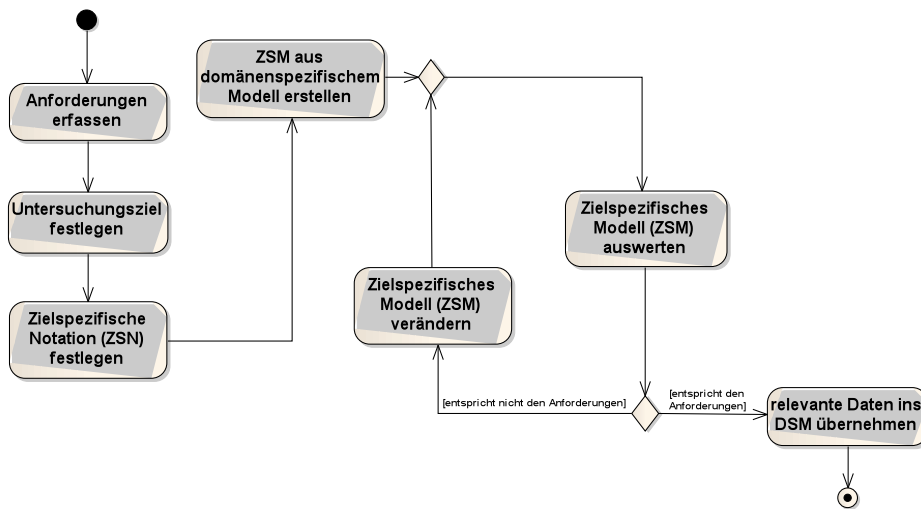


Abbildung 1: Validierung eines Untersuchungszieles

Nach der Festlegung des Untersuchungszieles ermittelt der Architekt die zur Validierung eines Untersuchungszieles benötigten Daten und erstellt eine zielspezifische Notation (ZSN), um diese Daten modellieren zu können. Alle nicht mehr für das Untersuchungsziel benötigten Informationen werden aus dem Modell entfernt. Es entsteht das zielspezifische Modell (ZSM).

Ausgehend von diesem Modell werden Simulationen durchgeführt, deren Ergebnisse mit denen für das Untersuchungsziel festgelegten Werte verglichen werden. Falls die Validierung fehlschlägt, ändert der Architekt das zielspezifische Modell und führt die Simulation erneut durch. Nach einem positiven Ergebnis werden die für das domänenspezifische Modell relevanten Daten des ZSMs in das DSM übernommen.

Durch die Trennung der für die Simulationen notwendigen Daten von den domänenspezifischen Daten, wird das domänenspezifische Modell nicht mit unnötigen Informationen angereichert. Gleichzeitig entsteht eine vollständig in UML gehaltene Dokumentation der Lösungsmöglichkeit für jedes Untersuchungsziel. Bei auftretenden Veränderungen können alle dokumentierten Untersuchungsziele, welche in ihrer Gesamtheit die Designentscheidung bilden, automatisch gegenüber den neuen Anforderungen validiert werden. Der Architekt kann dem Wandel im Projekt gelassen entgegensehen.



**Nichts ist beständiger als  
der Wandel**  
Unterstützung der Anforderungsanalyse  
durch Simulation

SOPHIST GmbH  
Vordere Cramergasse 13  
90478 Nürnberg

Tel.: +49 (0)911 40 900 - 0  
Fax: +49 (0)911 40 900 - 99

www.sophist.de  
heureka@sophist.de

Wer bin ich?



## André Pflüger

- Berater im Bereich Analyse und Architektur mit der UML
- Doktorand im Bereich Unterstützung der Entwicklung durch Modellierung

# Wer schreibt der bleibt

## Die Bücher der SOPHISTen



REQUIREMENTS ENGINEERING & MANAGEMENT	UML 2 GLASKLAR	AGILE SOFTWARE-ENTWICKLUNG	BASISWISSEN REQUIREMENTS ENGINEERING	SYSTEMANALYSE KOMPAKT	AGILITY KOMPAKT
Chris Rupp & die SOPHISTen	Chris Rupp, Dr. Stefan Queins & Barbara Zengler	Chris Rupp & Peter Hruschka	Chris Rupp & Klaus Pohl	Chris Rupp & die SOPHISTen	Chris Rupp, Peter Hruschka & Gernot Starke
5. Auflage	3. Auflage			2. Auflage	2. Auflage
seit Juli 2009 im Handel	seit August 2007 im Handel	seit Februar 2002 im Handel	seit März 2009 im Handel	seit März 2008 im Handel	seit April 2009 im Handel
ISBN: 978-3-446-41841-7	ISBN: 978-344-641-1180	ISBN: 978-344-621-9977	ISBN: 978-389-864-6130	ISBN: 978-382-741-9361	ISBN: 978-382-742-0923

SOPHIST GmbH

Nichts ist beständiger als der Wandel

Seite 2

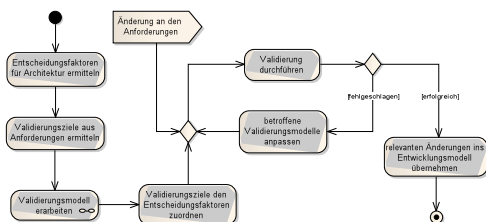


1.1

Die Welt ist im Wandel

1.2

Mit der Zeit gehen



1.3

...den Weg beschreiten

1.4

Fazit

Nichts ist beständiger als der Wandel





- Wandel im Projekt
- Auswirkungen auf die Architektur

## Die Welt ist im Wandel

### Wandel im Projekt

Die Welt ist im Wandel

#### Vorstellungen des Kunden verändern sich im Laufe der Zeit

Abschätzung von Kosten und Aufwand gerade bei großen Projekten sehr schwierig

Anforderungsänderungen können große Auswirkungen auf den Grobentwurf haben

Anforderungsqualität oftmals schlecht  
→ viel Interpretations-spielraum

Nachweis des zusätzlichen Aufwandes durch die Änderungen schwierig



Quelle Bild: Daniel Schwen, Wikimedia Commons



## Wandel im Projekt

- Arbeitsweisen verändern sich
  - Änderungen erfolgen in verschiedenen Abteilungen selten zur gleichen Zeit
- Verständnis des Produktes unterschiedlich
  - einheitliche (Dokumenations-) Sprache hilfreich
- Ausfall von wichtigen Informationsträgern muss kompensiert werden
  - schwierig bei lückenhafter Dokumentation



## Auswirkungen auf die Architektur

### Auftretende Probleme

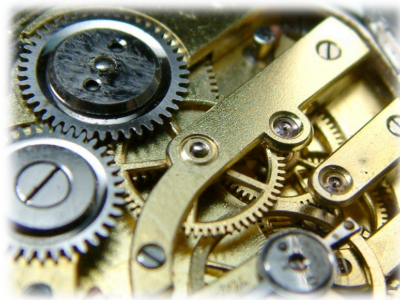
- Validierung des Architekturentwurfs gegenüber Anforderungen
  - Oftmals durch die Methode des „scharfen Hinsehens“
    - Aufwändig, keine Automatisierung möglich
    - Keine belastbaren Aussagen (Zahlen)
    - Übersehen von Faktoren durch einen getrübbten Blick
- Dokumentation in Protokollen
  - Medienbruch gegenüber der Modellierung mit UML

# Auswirkungen auf die Architektur

## Problemursachen

Architekturwahl hängt von vielen Faktoren ab

Gefahr des  
Überblicksverlusts



Kleine Änderungen,  
große Wirkung

Faktoren können sich  
gegenseitig beeinflussen

Quelle Bild: <http://www.pixelio.de>

SOPHIST GmbH

Nichts ist beständiger als der Wandel

1.1 – Seite 9



- Die Idee
- Vorgehensweise

Mit der Zeit gehen

## Die Idee

Zerlegung des Vorgangs der Architekturvalidierung

Entscheidungsfaktoren beeinflussen Wahl der Architektur

Validierungsziele sind Entscheidungsfaktor zugeordnet

Validierungsziel = allgemein formulierte Anforderung(en)



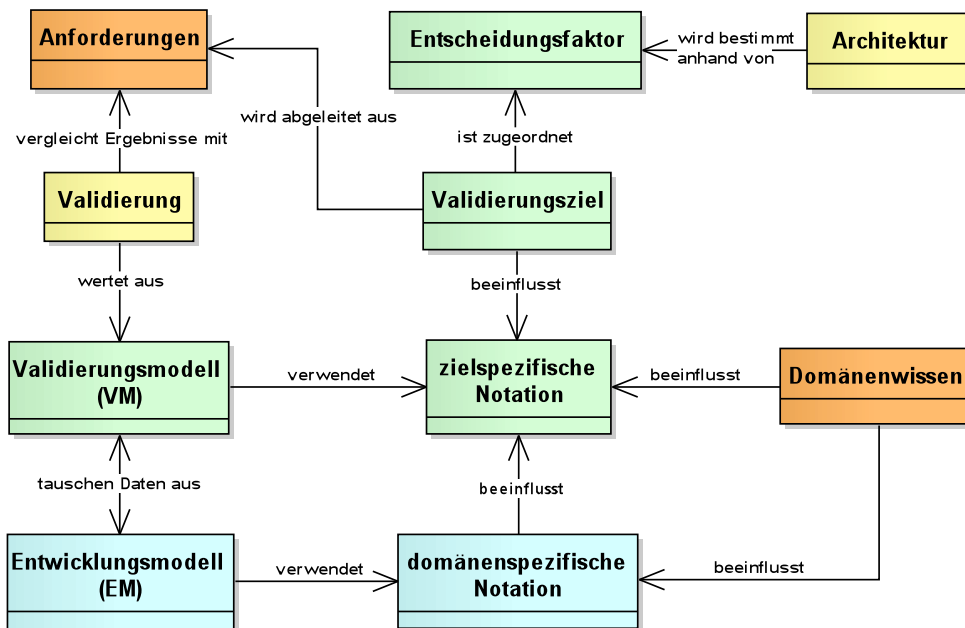
Trennung von Validierungs- und Entwicklungsdaten

Bidirektionaler Austausch von Daten um Aufwand zu reduzieren

Einheitliche Dokumentationsprache:  
UML

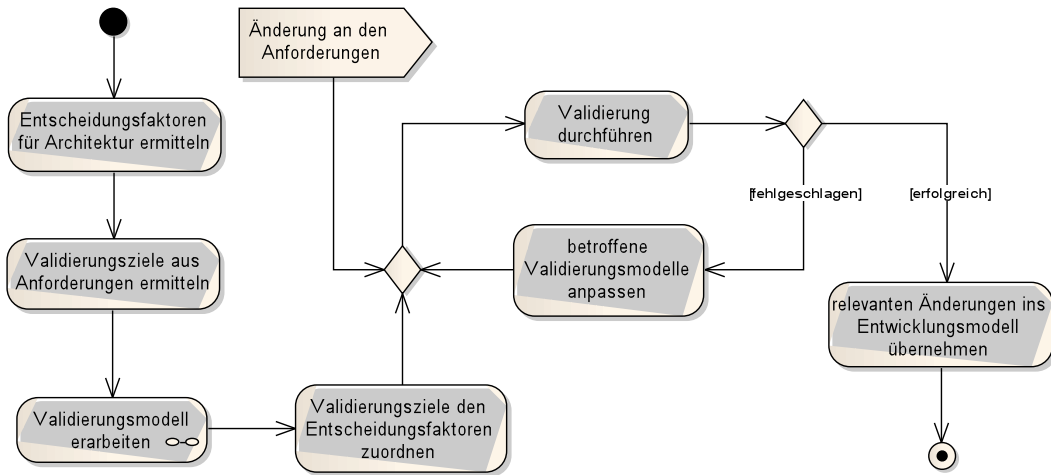
## Die Idee

Zusammenhänge der Begrifflichkeiten



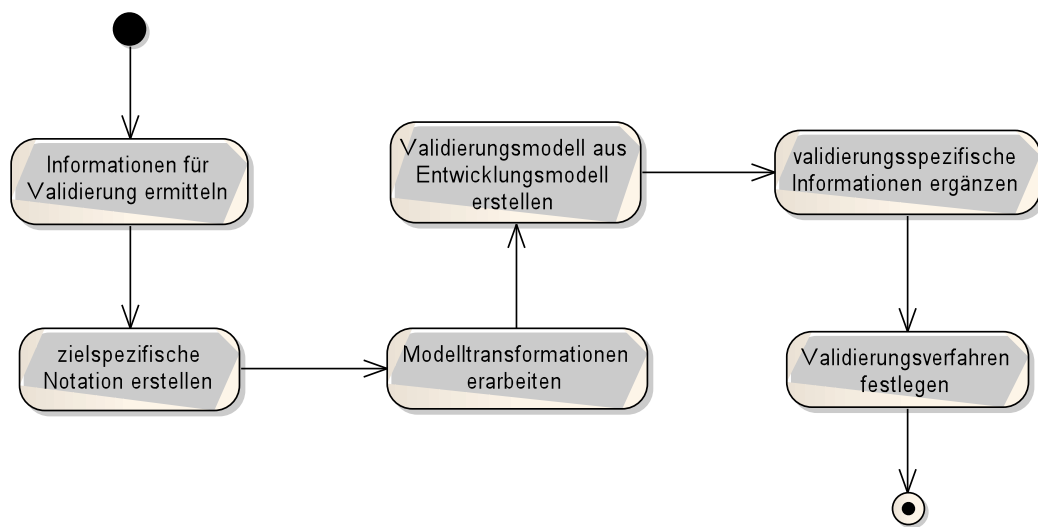
## Vorgehensweise

### Architektur validieren



## Vorgehensweise

### Validierungsmodell erarbeiten



1.3



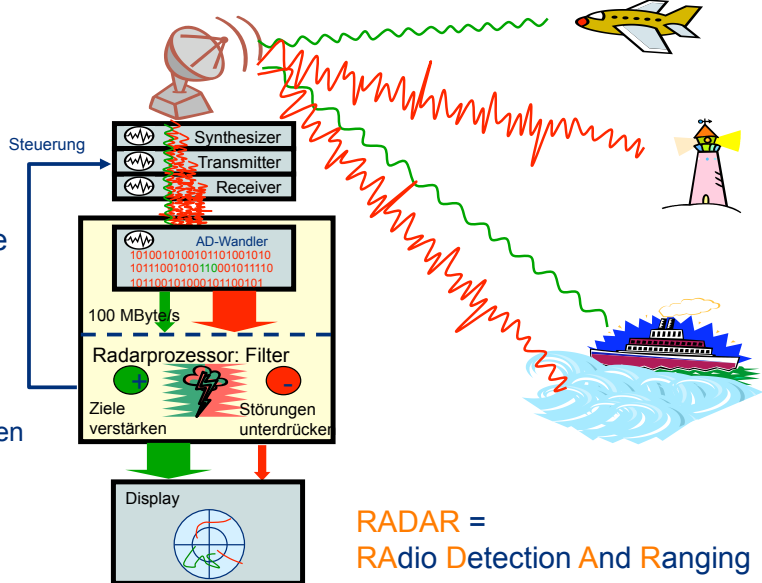
- Radarsysteme
- Validierungsziel
- Entwicklungsmodell
- Validierungsmodell
- Validierung
- Ergebnis

...den Weg beschreiten

...den Weg beschreiten

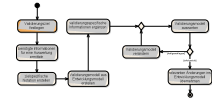
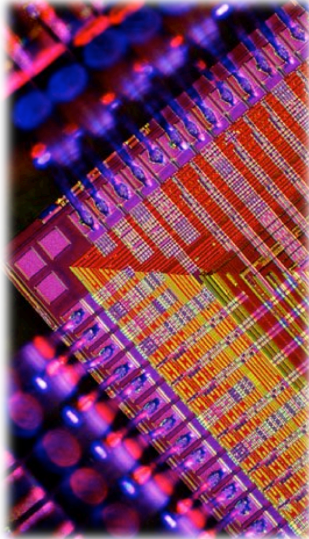
Radarsysteme  
Funktionsweise

- Signale aussenden und empfangen
- Verarbeitung der empfangenen Signale
  - A/D-Wandlung
  - Filterung
  - Verstärkung
  - Störungen unterdrücken
  - Daten verarbeiten



**RADAR =**  
**R**adio **D**etection **A**nd **R**anging

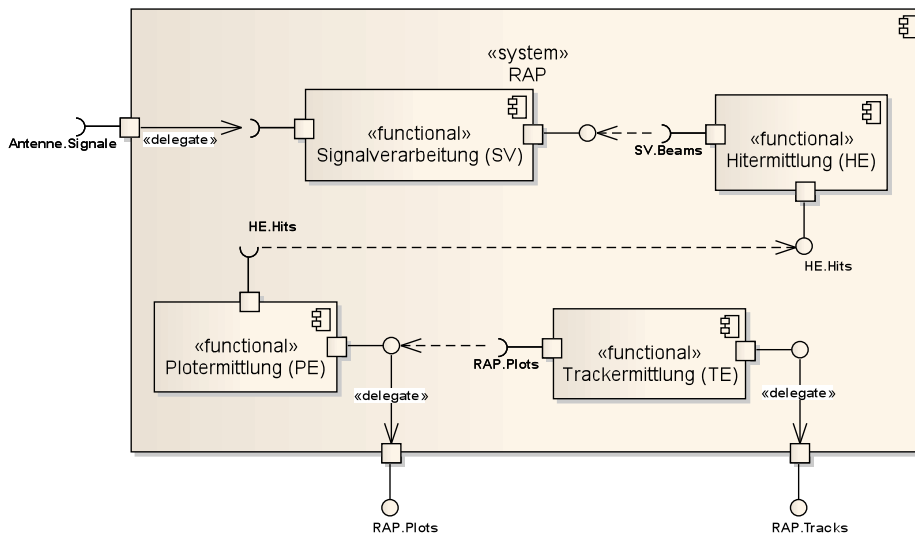
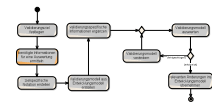
# Validierungsziel



- Auswirkungen der Datenkommunikation von SW-Komponenten auf die Hardware-Infrastruktur
- Validierungsziel:  
Die Architektur muss den Transport der bei der Verarbeitung der Radardaten auftretenden Datenmengen erlauben.

# Entwicklungsmodell

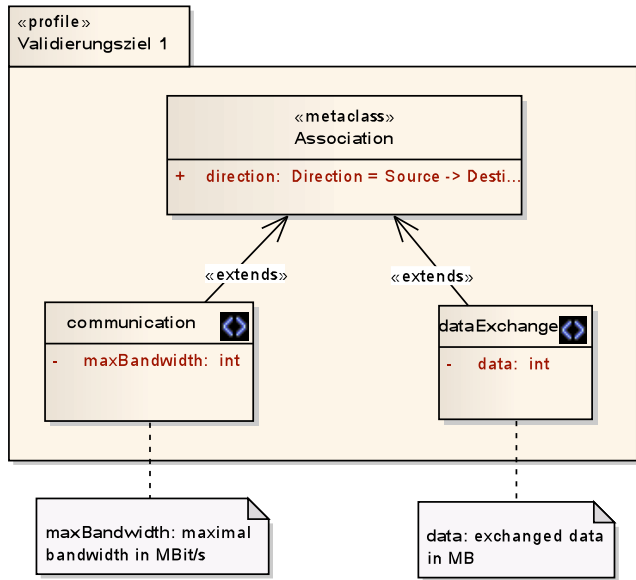
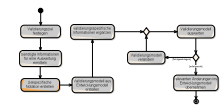
## Funktionale Sicht



...den Weg beschreiben

# Validierungsmodell

## Zielspezifische Notation

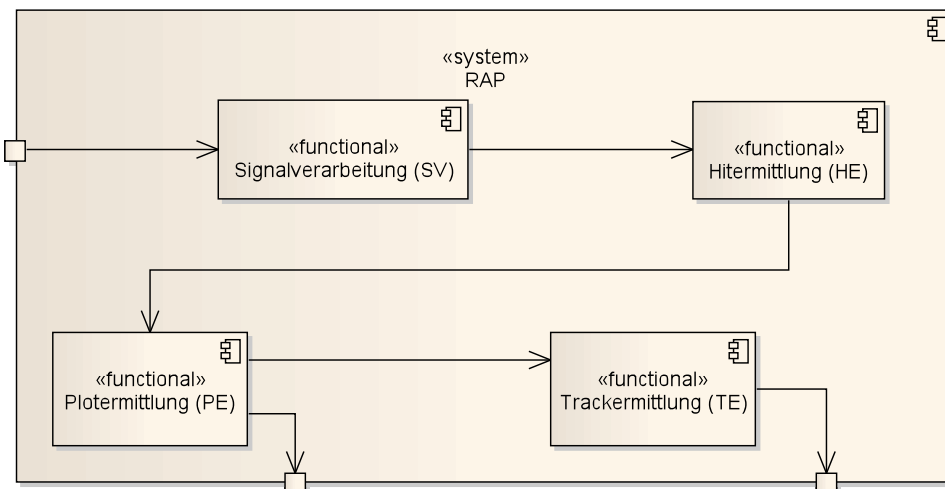


- UML wird leichtgewichtig durch ein UML-Profil erweitert
- 2 Stereotypen, 2 TaggedValues
- Anwendung auf Associations

...den Weg beschreiben

# Validierungsmodell

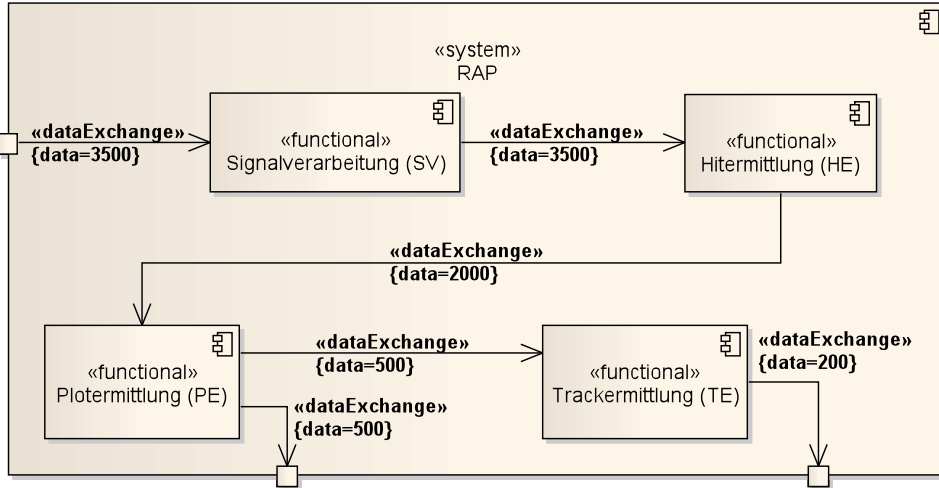
## Transformierte Daten aus dem EM



...den Weg beschreiben

# Validierungsmodell

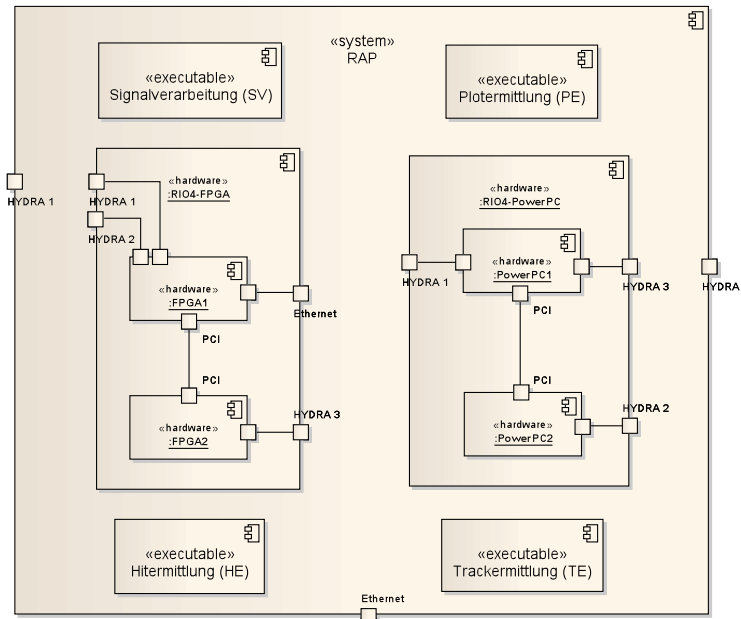
## Funktionale Sicht



...den Weg beschreiben

# Validierungsmodell

## Transformierte Daten aus dem EM

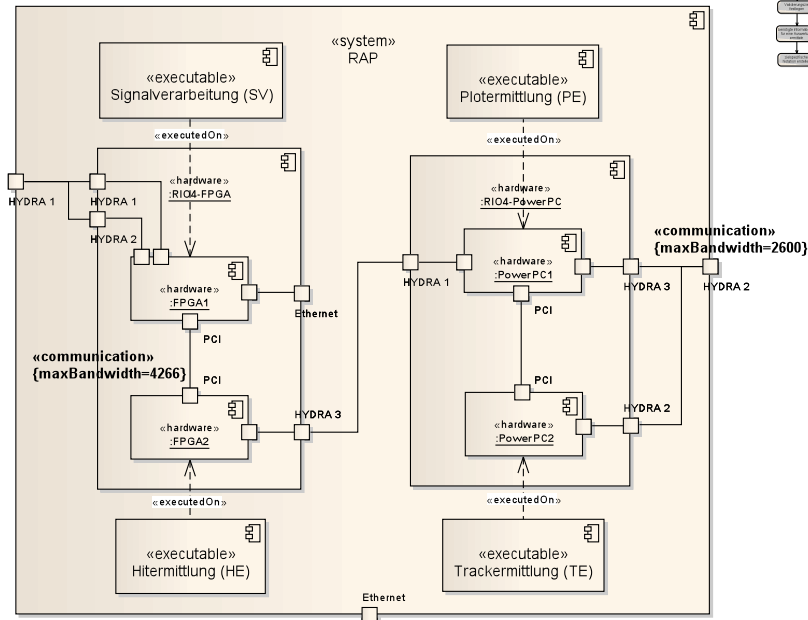




## Validierungsmodell

## Zielspezifische Daten

...den Weg beschreiben



SOPHIST GmbH

Nichts ist beständiger als der Wandel

1.3 – Seite 23

## Validierung

## Manuelle Auswertung

...den Weg beschreiben

- Datenannahme: 3500 Gbit/s  
→ 2 x 2600 Gbit/s = 5200 Gbit/s ✓
- Signalverarbeitung → Hitermittlung: 3500 Gbit/S  
→ 1 x 4266 Gbit/s ✓
- Hitermittlung → Plotermittlung: 1800 Gbit/S  
→ 1 x 2600 Gbit/s ✓
- Plotermittlung → Trackermittlung: 500 Gbit/S  
→ 1 x 2600 Gbit/s ✓
- Plotermittlung → Datenausgabe: 500 Gbit/S  
→ 1 x 2600 Gbit/s ✓
- Trackermittlung → Datenausgabe: 200 Gbit/S  
→ 1 x 2600 Gbit/s ✓

SOPHIST GmbH

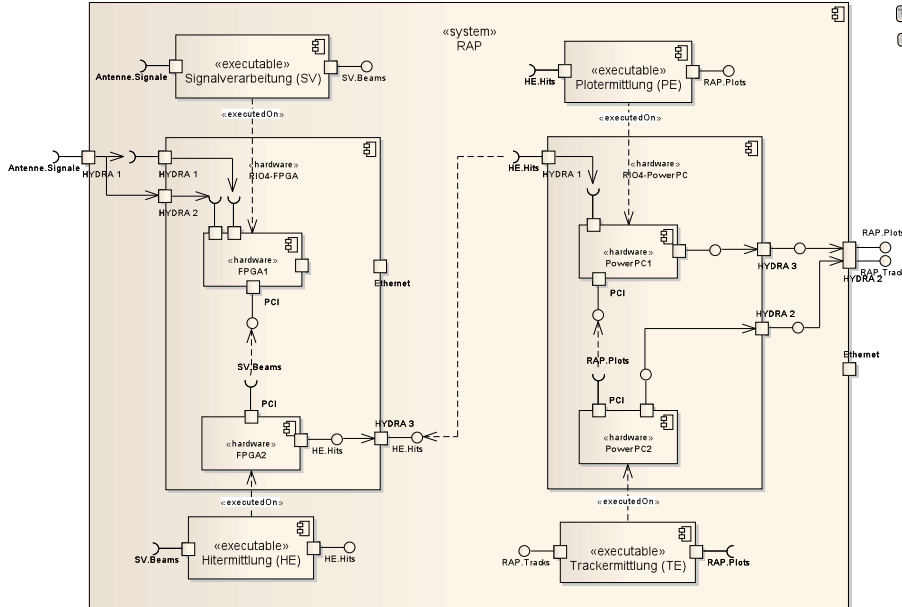
Nichts ist beständiger als der Wandel

1.3 – Seite 24

...den Weg beschreiten

# Ergebnis

## Entwicklungsmodell – System-Architektur



SOPHIST GmbH

Nichts ist beständiger als der Wandel

1.3 – Seite 25



- Zusammenfassung
- Vorteile

## Fazit

# Zusammenfassung

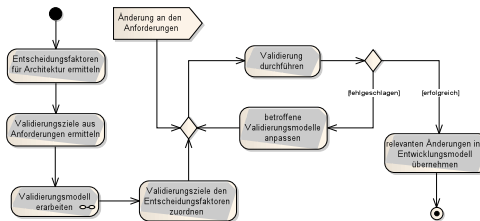


## Der vorgestellte Ansatz

Fazit

dokumentiert die Entscheidungsfaktoren, welche Grundlage für die Auswahl der Architektur sind

nutzt bereits vorhandene Informationen im UML-Entwicklungsmodell



validiert die Architektur anhand der Entscheidungsfaktoren gegenüber den Anforderungen

verwendet UML-konforme Elemente

# Vorteile



## Der vorgestellte Ansatz

verringert Modellierungsaufwand durch einen bidirektionalen Informationsaustausch zwischen Entwicklungs- und Validierungsmodell

unterstützt bei der Suche nach Auswirkungen von Anforderungsänderungen

Fazit

fördert die Ermittlung aller notwendigen Entscheidungsfaktoren

fördert automatisierte Validierungen

unterstützt die Verwendung von UML als einheitliche Dokumentationssprache



hält das Entwicklungsmodell frei von validierungsspezifischen Daten

erlaubt die Nachvollziehbarkeit von Architekturentscheidungen

## Haben Sie weitere Fragen?

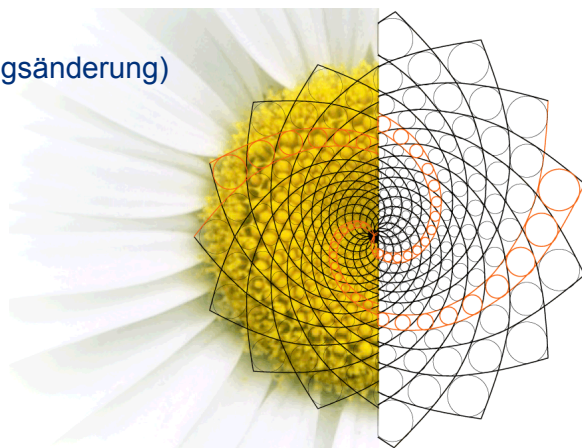


## Infos gefällig?

Die wichtigste Zeit eines Vortrags ist nach dem Vortrag

- ausführlicher Artikel zum Thema
- pdf des Vortrags (inkl. Anforderungsänderung)
- Login in den Downloadbereich
- Newsletter zu UML, OO und RE

Einfach Visitenkarte  
abgeben oder Mail an  
[heureka@sophist.de](mailto:heureka@sophist.de)  
Schicken.



Wir erkennen die Struktur Ihrer Projektanforderungen!

Infos unter [www.SOPHIST.de](http://www.SOPHIST.de)

## 6. Testen

### Sessionüberblick

---

6.1. Abbildung von Standard-Testprozessen in modernen Testmanagementwerkzeugen .	259
6.2. Erfolgsfaktoren von Testprozessbewertungsmodellen . . . . .	277
6.3. Erweiterung des TPI-Modells zur Reifegradbewertung unter Berücksichtigung von projekteigenen Anforderungen . . . . .	299
6.4. Beide Zweige des V aus einer Hand . . . . .	315

---



# Abbildung von Standard-Testprozessen in modernen Testmanagementsystemen

Matthias Zieger

Developer Platform & Strategy Group  
Microsoft Deutschland GmbH  
Siemensstrasse 27  
61325 Bad Homburg  
mzieger@microsoft.com

**Abstract:** Testprozesse im benötigen spezielle Standards, genau wie Softwareentwicklungsprojekte im Allgemeinen. Es werden kurz agile und formale Testprozessstandards erläutert, wie z.B. ISTQB, TMAP, TPI und Test Driven Development. Danach wird gezeigt, wie diese Testprozesse in modernen Testmanagementwerkzeugen geplant, gesteuert, ausgeführt und ausgewertet werden. Dabei werden auch zahlreiche Qualitätsmetriken vorgestellt und wie diese möglichst automatisch gewonnen werden können. Am Ende werden Ansätze gezeigt, wie die Testergebnisse möglichst einfach an die Entwicklung zurückgespielt werden können und der Code-Build-Test-Repro-Debug-Fix Zyklus massiv verkürzt werden kann.

## 1 Überblick über Testprozesse

Es werden kurz agile und formale Testprozessstandards erläutert, wie z.B. ISTQB, TMAP, TPI und Test Driven Development. Dabei werden kurz die Vorteile der Prozesse dargestellt.

## 2 Umsetzung von Testprozessen in modernen Testmanagement-Lösungen

### 2.1 Grundlagen

Erklärt werden ausgehend von den fachlichen Anforderungen Konzepte wie Testsuiten, Testpläne und Testfälle. Es wird gezeigt, wie ein durchgängiger Workflow aussehen kann, um diese nahtlos mit Themen wie Build-Management, Versionskontrolle, Change-Management und Fehlerverwaltung zu integrieren.

Es werden verschiedene automatisierte Workflows vorgestellt, die entsprechende Testprozessstandards automatisiert abbilden können.

## **2.2 Testautomatisierung**

Testautomatisierung ist der Schlüssel zur Effizienzsteigerung im Testlabor. Es wird gezeigt, wie aus manuellen Tests sehr effizient Testautomaten erstellt werden können und wie mächtig diese Testautomaten sein können, wenn kein proprietäres Testframework sondern ein Standard-Applikationsframework als Grundlage für die Testautomaten hergenommen wird.

## **2.3 Informationsgewinnung beim Testen**

Nachdem in den Testmanagementprozessen zahlreiche Aktivitäten durchgeführt wurden, stehen zahlreiche Metriken zum Abrufen bereit. Diese Metriken lassen sich mit anderen Kennzahlen aus Entwicklung (z.B. Code-Analyse, Profiling), Anforderungsmanagement, Versionierung und Build zu einem Gesamtbild über das Projekt verknüpfen, welches dann wiederum fundierte Entscheidungen (z.B. Freigabe, Terminverschiebung, Risikomanagement) erlaubt.

## **2.4 Aufbau virtueller Testfabriken**

Testvirtualisierung ist einer der großen Trends im Testmanagement-Umfeld. Der Autor stellt vor, wie die Testvirtualisierung helfen kann, die Fehlerreproduzierbarkeit massiv zu verbessern. Ausserdem wird erklärt, warum einfache Fehler-Tickets den heutigen Ansprüchen an ein modernes Testmanagement nicht mehr gerecht werden können und was man stattdessen mit „Rich Information Bugs“ anfangen kann.





SEE Conference, Köln

# Quo Vadis, Tester ?




Abbildung von Testprozessen in modernen Testmanagementlösungen

Matthias Zieger  
Business Development Test Solutions  
Microsoft GmbH

Microsoft

# Agenda

- Kommunikation zwischen Fachseite, Test und Entwicklung verbessern
- Auswahl der Richtigen Tests
- Aufbau eines Virtualisierten Testcenters
- Reporting und Auswertung von Tests

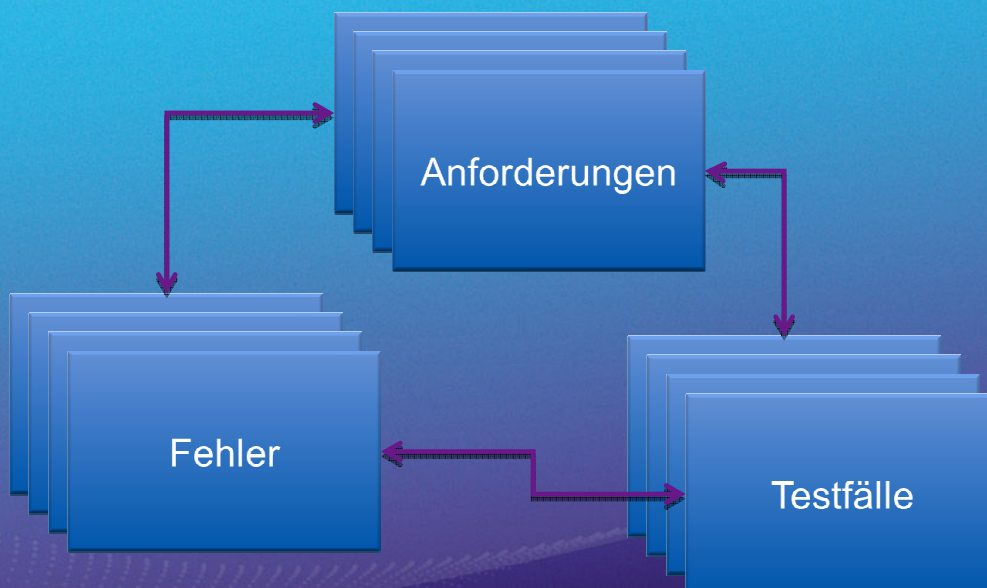


## Worin manifestiert sich Qualität ?

- Artefakte aus der Fachabteilung, der Entwicklung und der Qualitätssicherung
  - Anforderungen und Projektplan
  - Architektur und Modelle
  - Code
  - Testfälle und Testdaten
- Das alles ergibt (mehr oder weniger)
  - Funktionalität
  - Performance
  - Wartbarkeit

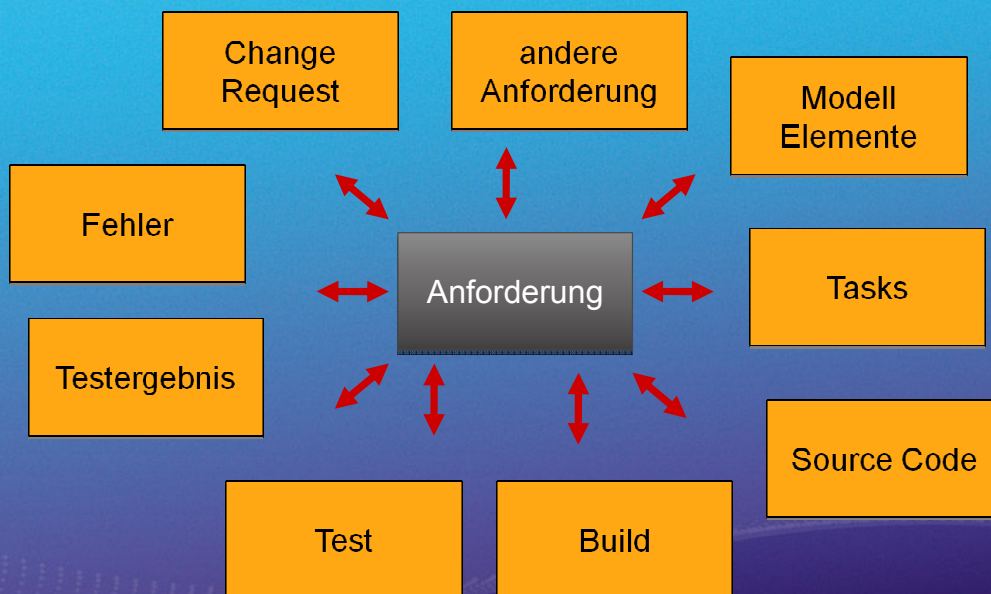


## Nachvollziehbarkeit beim Testen





## Mehr Bedarf an Traceability ?



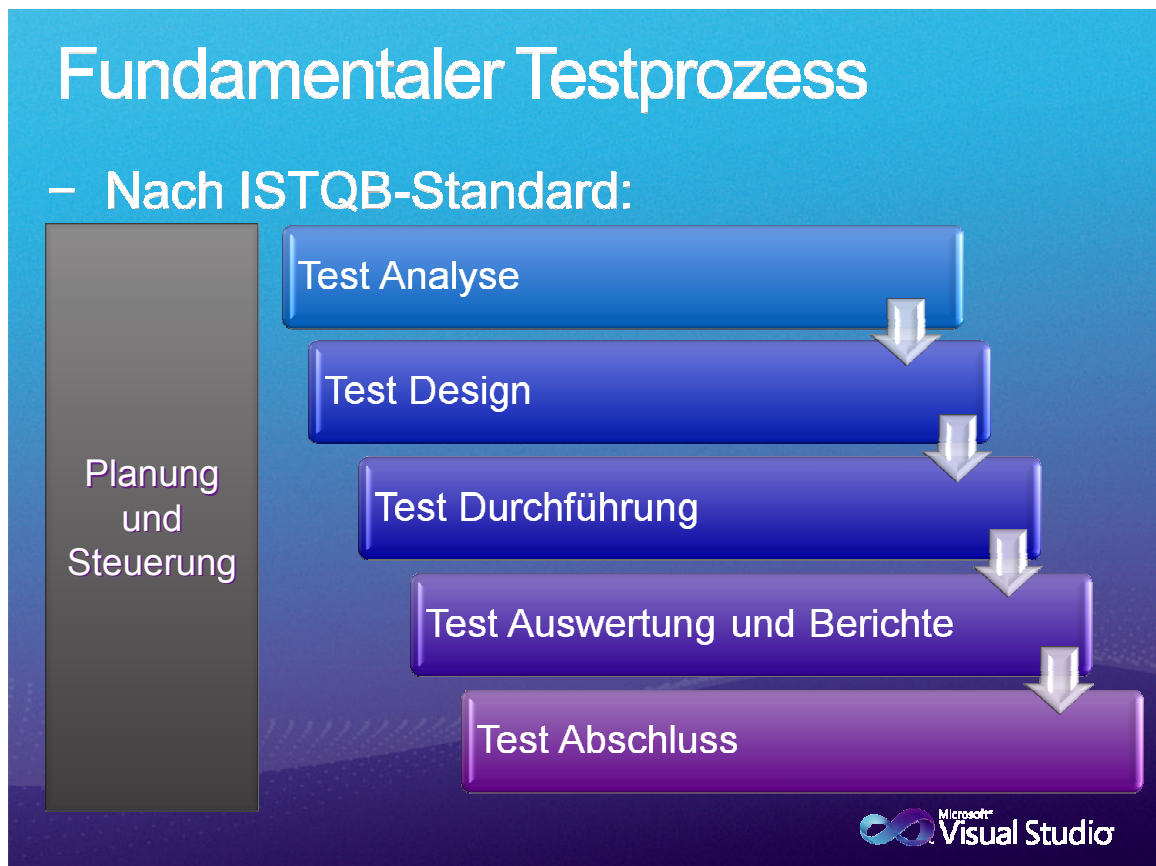
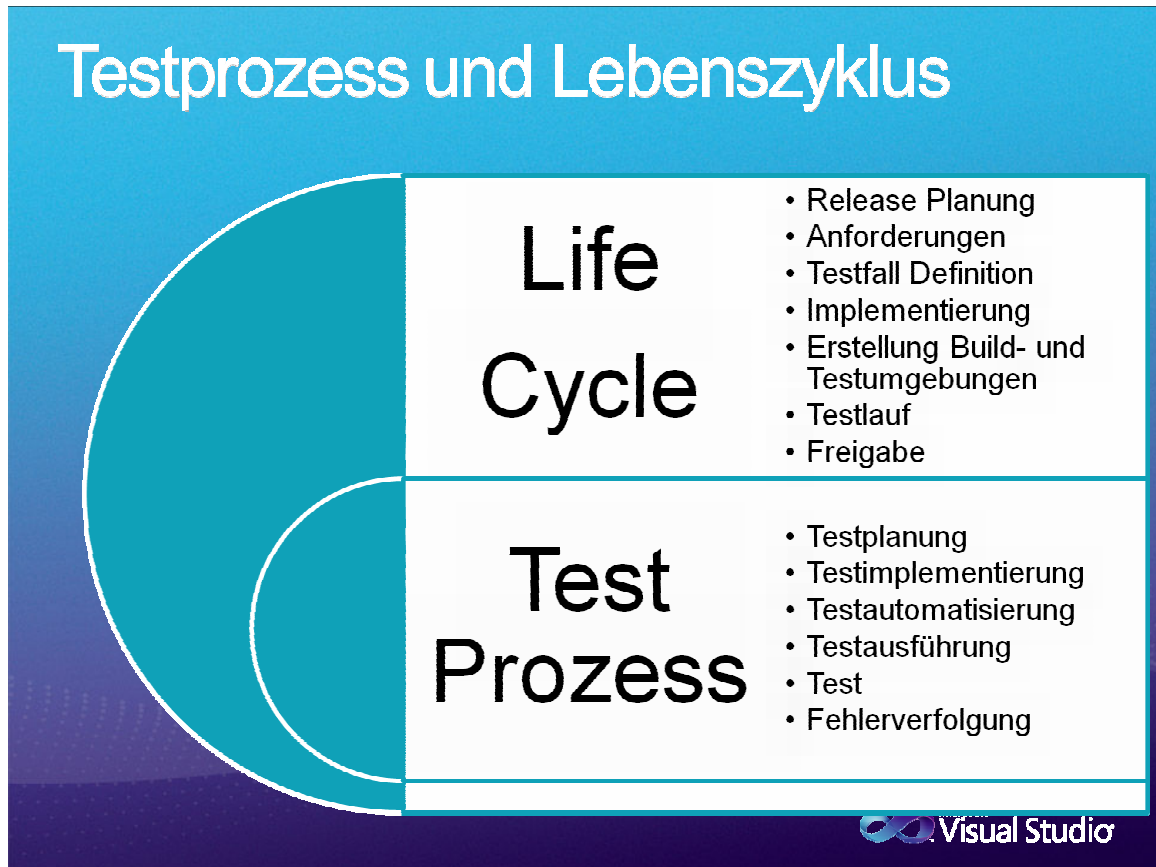
Microsoft  
Visual Studio

## Was wird benötigt ?

- Der Klassiker: People, Process, Technology
  - Ausgebildete Tester, Testmanager, Testanalysten
    - z.B. nach ISTQB\* Standard
  - Ein Testprozess
    - Agil, Iterativ, Wasserfall, immer Kunden-spezifisch
  - Testwerkzeuge zum effizienten Testen
    - Testmanagement
    - Testausführung
    - Testanalyse
- Hilfsmittel, damit sich Tester und Entwickler besser verstehen

ISTQB: International Software Testing Qualifications Board – kümmert sich um standardisierte Ausbildung von Testern und Testmanagern

Microsoft  
Visual Studio





## Basis-Anforderungen an Testwerkzeuge

- Ermöglicht anforderungsbasiertes Testen
- Steuerung für manuelle und automatisierte fachliche Tests
- Verwaltung der Testumgebung
- Kommunikation mit der Entwicklung



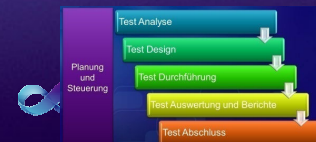
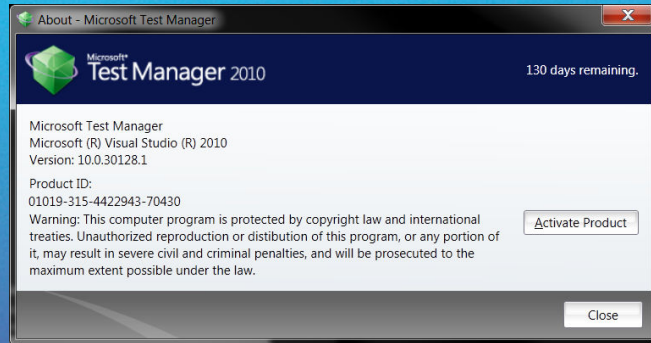
## Moderne Testlösungen

- Testplanung
- Testdefinition
- Testausführung
- Test Reports
- Rich Information Bugs
- Testlaborverwaltung
- Verwaltung aller Artefakte in einem Repository
- Benötigte Links
  - Anforderungsmanagement
  - Change Management
  - Configurations Management
  - Build-System
  - ...



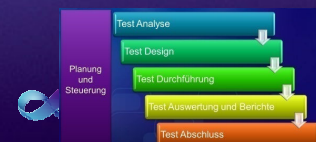
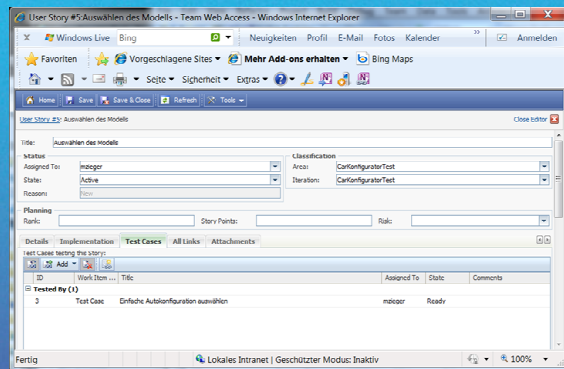
# Visual Studio 2010 für Tester !

- Orientiert an Testaufgaben
  - Testplanung
  - Testausführung
  - Testauswertung
  - Umgebungsmanagement
- Testautomatisierung für GUIs
- Verbindet Test und Entwicklung
  - „Rich Bugs“
  - Testvirtualisierung



# Testanalyse: Anforderungen

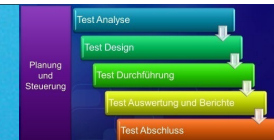
- Echtes Anforderungsmanagement
  - Web Client, Office, MS Project
- Agil oder Formal
  - User Stories, Use Cases, Anforderungen, ...
- Vorteile
  - Bidirektionale Verbindungen mit Testfällen, Arbeitsaufgaben, Quellcode usw. sichern die Nachvollziehbarkeit
  - Definierbare Workflows





# Testplanung

- Was soll getestet werden ?
  - Testplan basierend auf verlinkten Anforderungen
    - Bidirektionale Traceability
    - Testabdeckung ist direkt sichtbar



Testing Center | Plan | Test | Track | Organize

Contents | Properties

Test suite: TestPlanCarKonfig  
Default configurations (1): win 7 x64 IE 8

State: In progress

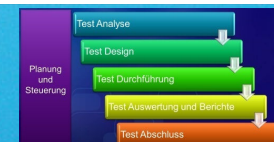
Drag a column header here to group by that column.

Order	ID	Title	Priority	Conf...	Testers	Area Path
1	1	Starte Anwendung	2	1	mzieger	CarKonfiguratorTest
2	3	Einfache Autokonfiguration auswählen	2	1	mzieger	CarKonfiguratorTest
3	2	Schliesse Anwendung	2	1	mzieger	CarKonfiguratorTest

Microsoft  
Visual Studio

# Testdesign

- Testsuiten
- Logische Testfälle
  - Testschritte
  - Testschrittbibliothek
- Testdaten
  - Testausprägungen
  - Äquivalenzklassen
- regelbasierte Test-Suiten



Testing Center | Plan | Test | Track | Organize

Contents | Properties

Test Case 3\*: Einfache Autokonfiguration auswählen

Title: Einfache Autokonfiguration auswählen

Status: Assigned To: mzieger, State: Ready, Priority: 2, Automation Status: Automated

Classification: Area: CarKonfiguratorTest, Iteration: CarKonfiguratorTest

Steps: Summary | Tested User Stories | All Links | Attachments | Associated Automation

Action	Expected Result
Selektiere @Autotyp	Preis ändert sich
Selektiere @Extra	Preis ändert sich
Wähle Rabatt @Rabattsstufe	
Überprüfe Preis	Gesamtpreis: @Preis

Parameter Values:

Autotyp	Preis	Rabattsstufe	Extra
IS	53370	10	Zentralverregelung

Microsoft  
Visual Studio

# Test- und Regressionsplanung

## Regression Tests

- R1 – Critical Regression Tests
- R2 – Software Update Tests
- R3 – Software Release Tests
- R4 – Full Tests

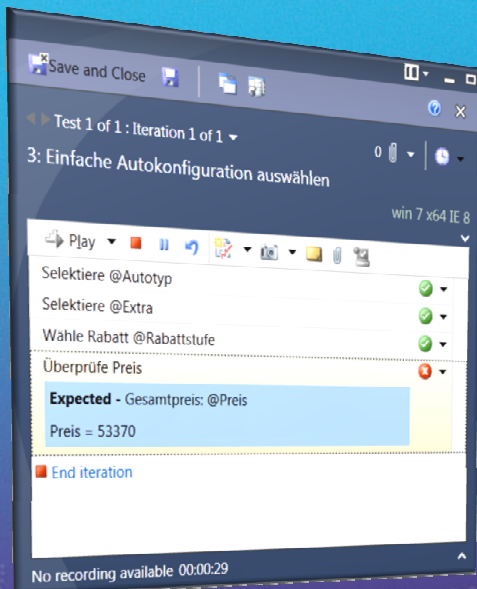
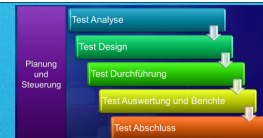
# Testplanung: Regression

- Was soll erneut getestet werden ?
  - Test-Impact Analyse für Regressionstests
  - Basierend auf Quellcodeänderungen zwischen zwei Versionen bzw. Releases der Anwendung

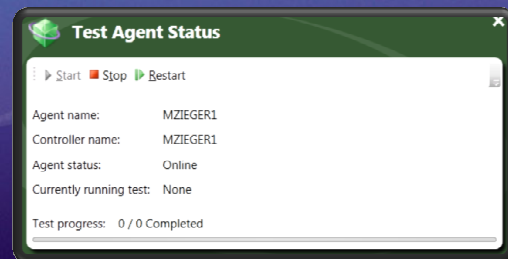
ID	Title	State	Priority	Last run date	Suite	Test configuration
Passed (2)						
16	Test Add	Passed	2	10.11.2009 13:44:59	Userinterface	win 7 x64 IE 8
18	Test Mult	Passed	2	06.11.2009 09:42:23	Userinterface	win 7 x64 IE 8



# Testdurchführung



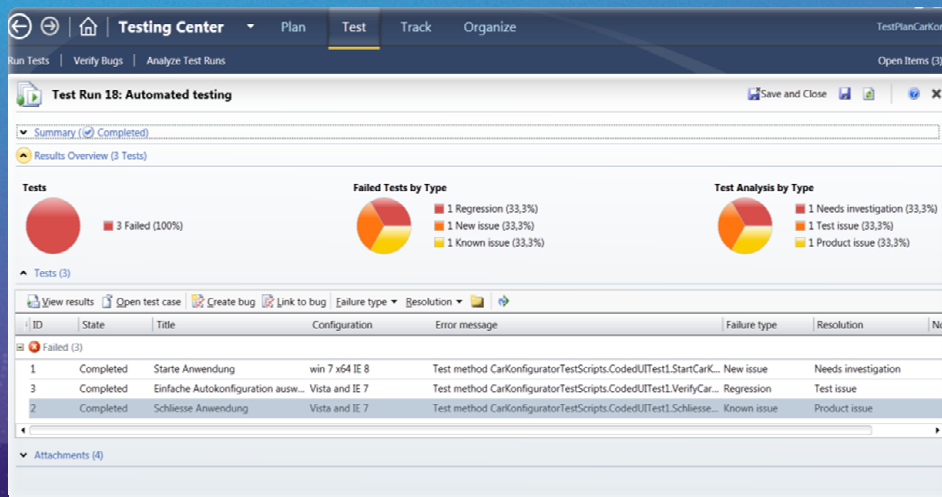
- Einfaches Werkzeug
- Testsuiten oder Testfälle
- Inkl. Test-Daten-Parameter
- Manuell oder automatisiert



# Fehlerverfolgung




- Direkt während der Testausführung lassen sich Fehler anlegen und auswerten
- Vorteil: Kein Medienbruch




Studio

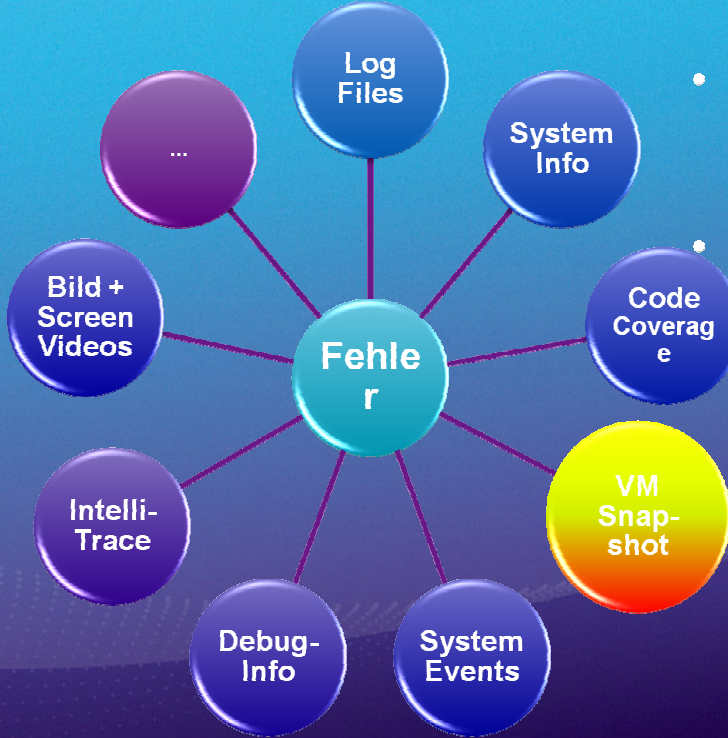

# Moment mal ...




- Ziel: Bug-Ping Pong vermeiden
- Wie ?
  - Automatisch
  - Transparent
  - Zuverlässig
- Was will der Entwickler ?
  - Fehler brauchen Zusatzinformationen
  - Am besten soll die ganze Testumgebung am Fehler hängen 😊



# Rich Information Bugs



- Bug-Ping Pong vermeiden
- Zusatzinformationen
  - Automatisch
  - Transparent
  - Zuverlässig
- Am besten soll die ganze Testumgebung am Fehler hängen 😊





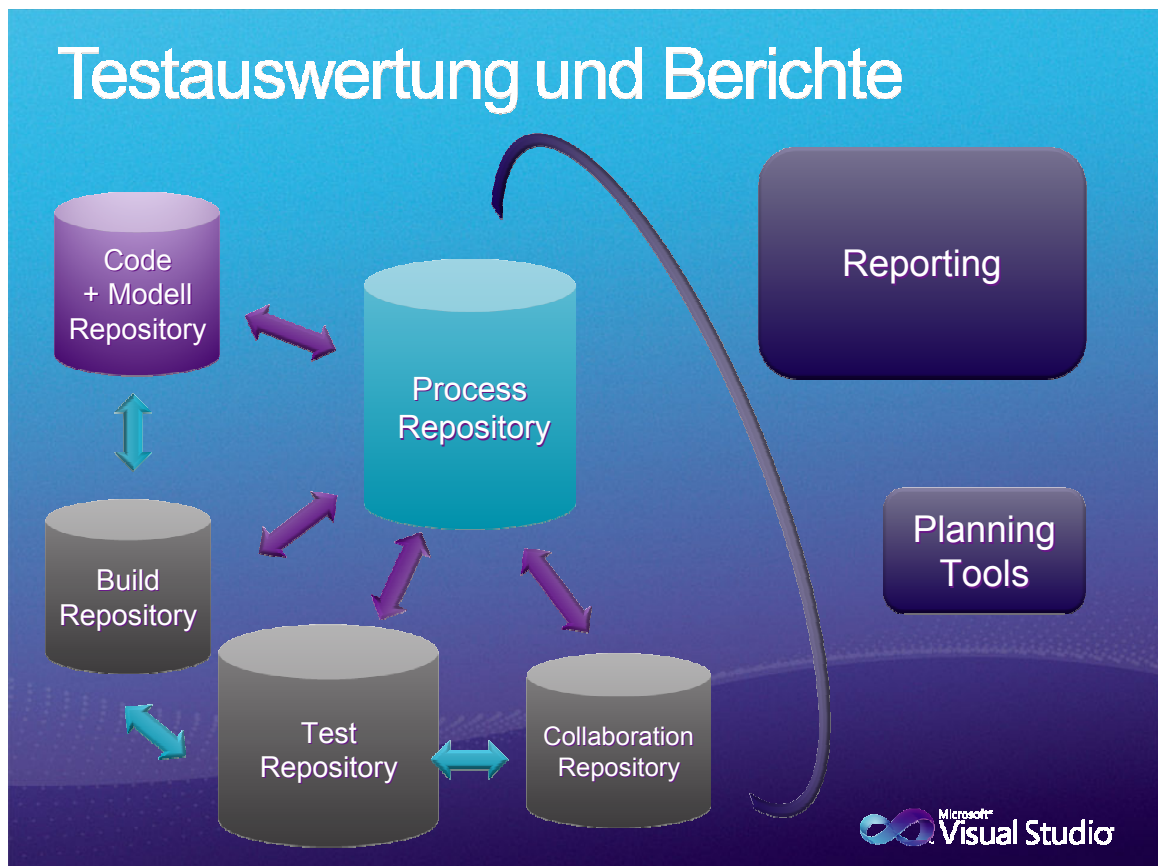
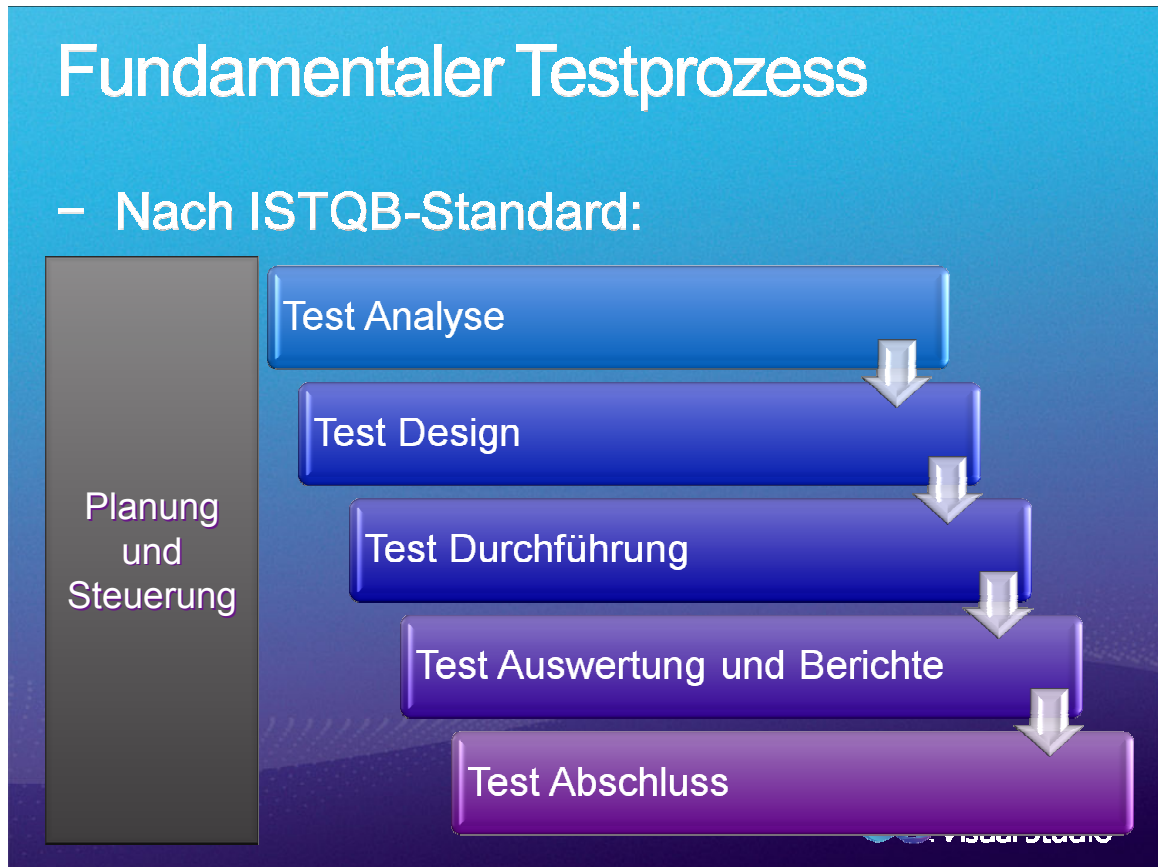
## Testumgebungen virtualisieren

- Mehrere Konfigurationen gleichzeitig testbar
- Einfaches Vorbereiten und Aufräumen von Testumgebungen
- **Eingefrorene Test-Umgebungen können direkt mit Fehlerverfolgung verlinkt werden**

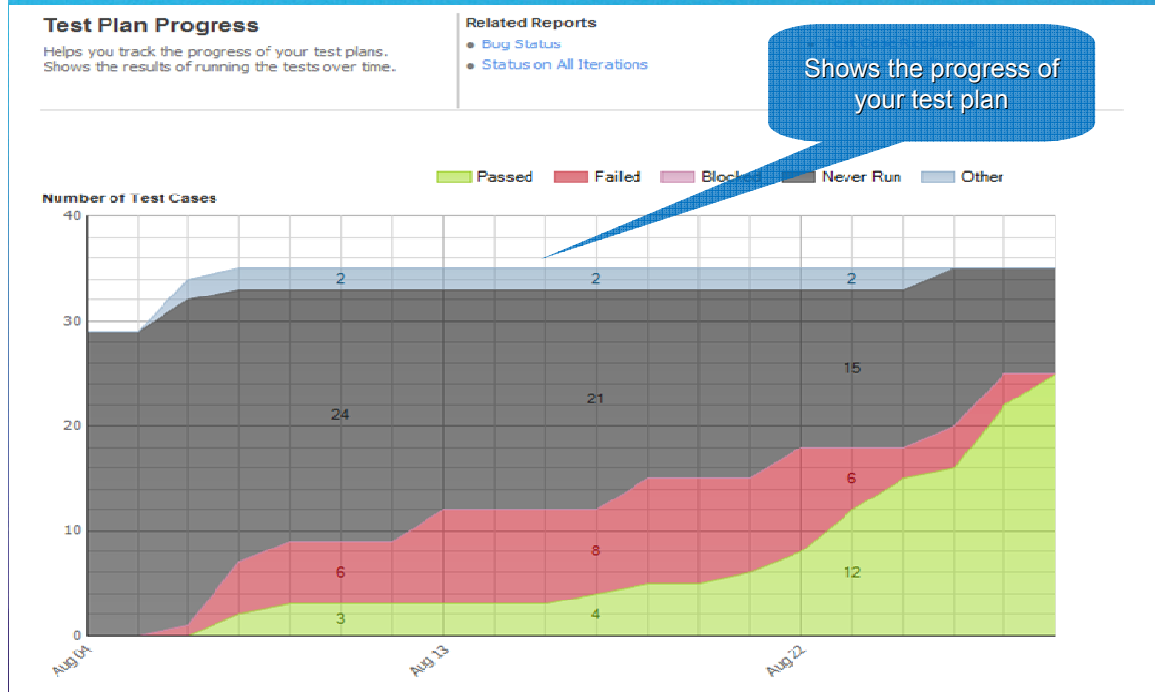


## Vorteile der Rich Information Bugs

- Extrem einfache Fehlerreproduktion
- Weniger HW/SW Kosten durch virtuelle Test-Labore
- Weniger Fehler mit Status „Nicht reproduzierbar“



# Welche Tests sind wie verlaufen ?



# Testergebnisse je Anforderungen

**Stories Overview**  
Helps you track how far each user story has been implemented. Shows each story's actual number of hours of work remaining and completed, its acceptance test results, and the number of bugs that are linked to each story.

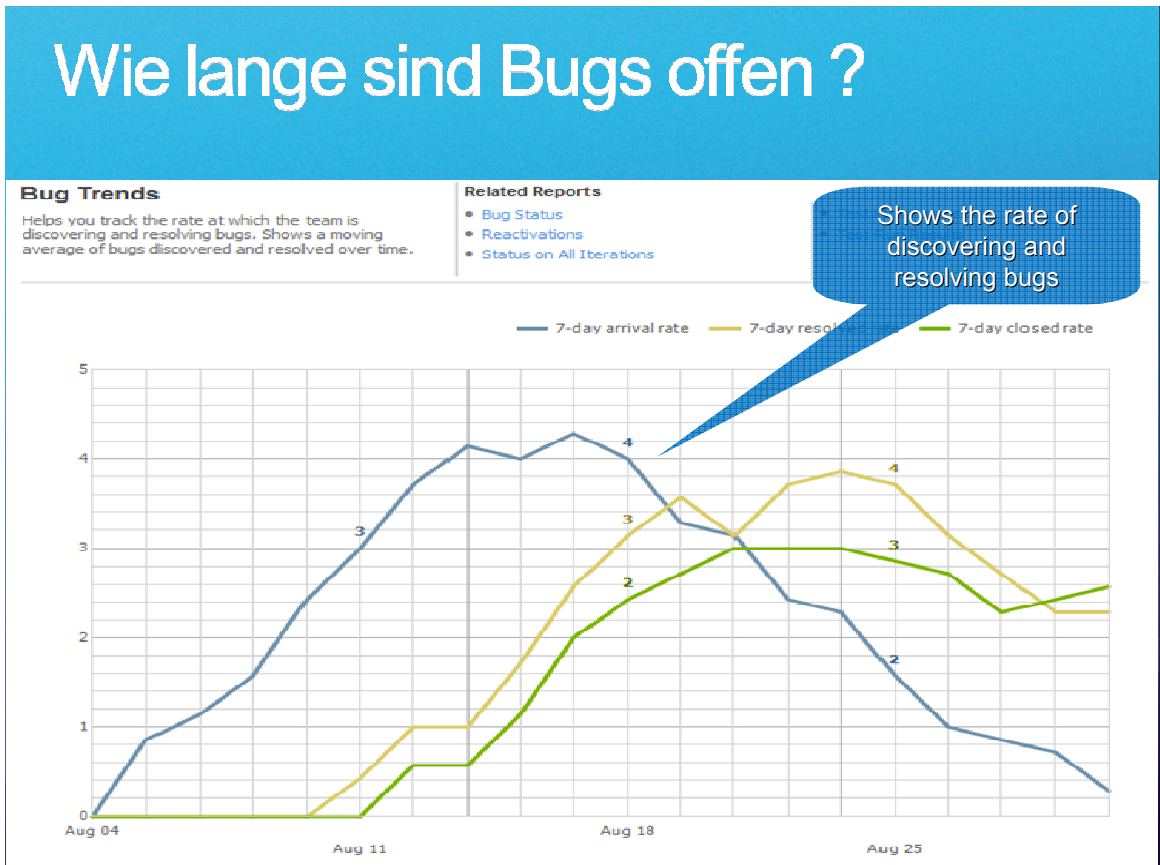
**Related Reports**

- Bug Status
- Status on All Iterations
- Stories Progress
- Test Case Readiness
- Test Plan Progress

Shows how far each user story has been implemented

Title	Work Progress		Test Status		
	% Hours Completed	Hours Remaining	Tests	Test Results	Bugs
Customer can see new functionality in their profile as they are introduced	100 %	0	3	33 % 67 %	0
Customer enters personal reviews	100 %	0	5	56 % 44 %	0
Admin/editor publishes customer submitted review	100 %	0	3	33 % 67 %	0
Customer rates review	100 %	0	3	40 % 40 %	0
<b>Customer comments on review</b>	13 %	112	5	100 %	0
Customer searches for reviews	100 %	0	3	33 % 67 %	0
Customer finds restaurant from review	100 %	0	2	33 % 67 %	0
Customer finds reviews for restaurant	100 %	0	2	50 % 50 %	3





## Zeit für Ihre Fragen ...

- Kontakt:
  - [mzieger@microsoft.com](mailto:mzieger@microsoft.com)
- Informationen:
  - [www.microsoft.de/visualstudio](http://www.microsoft.de/visualstudio)



## www.microsoft.de/visualstudio

- Die erste Anlaufstelle für alle Informationen rund um die professionellen Microsoft-Entwicklerwerkzeuge und Application Lifecycle Management
- Ausführliche Informationen zu allen Produkten, aktuelle News, Whitepaper, Kundenreferenzen und technische Ressourcen sowie kostenlose Testversionen und kurze Videos, die Ihnen in jeweils 15 Minuten einen guten Überblick geben
- Download von Visual Studio 2010





**Microsoft®**  
*Your potential. Our passion.®*

[www.microsoft.de/visualstudio](http://www.microsoft.de/visualstudio)

© 2009 Microsoft Corporation. All rights reserved. Microsoft, Visual Studio, the Visual Studio logo, and [list other trademarks referenced] are trademarks of the Microsoft group of companies.  
The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.  
MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

## Erfolgsfaktoren von Testprozessbewertungsmodellen

Frank Brüseke<sup>1</sup>, Yavuz Sancar<sup>2</sup>, Enes Yigitbas<sup>3</sup>

Universität Paderborn  
Software Quality Lab – s-lab  
33098 Paderborn, Deutschland

<sup>1</sup>fbrueseke@s-lab.upb.de  
<sup>2</sup>ysancar@s-lab.uni-paderborn.de  
<sup>3</sup>enes@mail.uni-paderborn.de

### 1 Motivation

Softwarebasierte Produkte sind ein wichtiger Bestandteil unserer Gesellschaft geworden. Die Qualität der zugrundeliegenden Software ist in vielen Märkten der entscheidende Wettbewerbsvorteil. Um den ständig steigenden Anforderungen an die Softwarequalität gerecht zu werden, muss der zugehörige Softwareentwicklungsprozess ebenfalls eine hohe Qualität aufweisen, welches maßgeblich durch den begleitenden Testprozess mitbestimmt wird. Einen international standardisierten Testprozess hat die ISTQB aufgestellt. Mittlerweile existieren unterschiedliche Testprozessbewertungsmodelle, um die Stärken und Schwächen eines Testprozessen zu identifizieren und gegebenenfalls Verbesserungsvorschläge zu geben. Die Verwendung solcher Modelle kann in der Phase „Abschluss“ stattfinden (Abbildung 1).

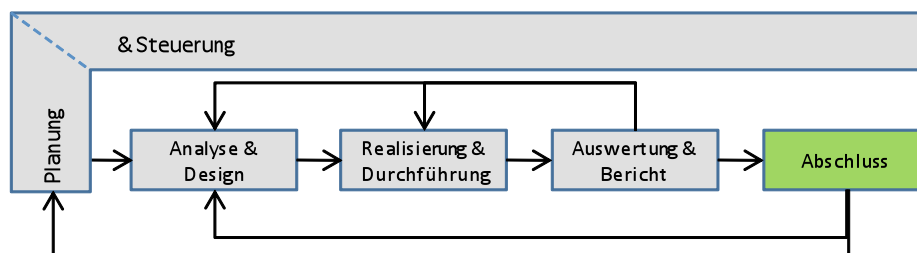


Abbildung 1: Fundamentaler Testprozess nach ISTQB [SL05]

In der Praxis haben sich TPI [PKS00], TMM [BUR03] und TMMi [Vee09] als Bewertungsmodelle für Testprozesse bewährt [FD08c]. Sie ermöglichen auf Grundlage von Befragungen der Testmanager und Tester im Rahmen von Interviews eine qualitative Reifegradstufung für den vorliegenden Testprozess.

Da sich die genannten Bewertungsmodelle im Aufbau und bei der Vorgehensweise unterscheiden, ist es wichtig von einem passenden Bewertungsmodell Gebrauch zu machen. Dazu sollen dem Testmanager Kriterien für die Auswahl eines Bewertungsmodells an die Hand gegeben werden. So kann er für sein Projekt oder sein Unternehmen, das Bewertungsmodell auswählen, dass am ehesten verspricht erfolgreich mit Standards zu sein.

Essentiell für eine erfolgreiche Testprozessbewertung ist also die richtige Wahl eines Bewertungsmodells. Deshalb wurden in unserer Studie [Yig09] Auswahlkriterien für die Analyse bzw. Vergleich der Testprozessbewertungsmodelle aufgestellt. Die genannten Bewertungsmodelle wurden mithilfe dieser Kriterien untersucht und miteinander verglichen. Dabei wurden Gemeinsamkeiten und Unterschiede, sowie Vor- und Nachteile der jeweiligen Bewertungsmodelle verdeutlicht. Mit Hilfe dieser Untersuchung sind wir in der Lage einen Leitfaden für die systematische Auswahl und Einsatz eines Bewertungsmodells zu entwickeln. Diese Ergebnisse werden wir präsentieren.

## 2 Ansatz

Der Testprozess ist ein wichtiger Bestandteil des Softwareentwicklungsprozesses. Die Auswahl eines Testprozessbewertungsmodells hat einen großen Einfluss auf den gesamten Entwicklungsprozess und damit auf das entstehende Softwareprodukt. Bei der Bestimmung von Auswahlkriterien wurden andere wissenschaftliche Arbeiten berücksichtigt, die sich mit Vergleichsansätzen für Testprozessbewertungsmodelle [FD08a] und dem Vergleich von *Software Process Improvement* (SPI) Frameworks ([HC01], [SC99]) beschäftigen. Ausgehend von diesen Vergleichsansätzen wurde mit Hilfe der in [PKS00] beschriebenen Anforderungen für Testprozessbewertungsmodelle ein feinerer Vergleichsansatz entwickelt. Hierbei wurde beachtet, dass wichtige testrelevante Aspekte abgedeckt und Testmanagern ein breites Spektrum an Auswahlkriterien zur Verfügung gestellt wurde, welches in unterschiedlichen Praxissituationen angewandt werden kann. Bei der Bestimmung von Auswahlkriterien wurden also sowohl theoretische (Testprozess nach ISTQB) als auch praktische Aspekte (beispielsweise Zeit und Kosten) berücksichtigt, sodass ein sinnvoller Einsatz der Auswahlkriterien ermöglicht wurde. Als Ergebnis ist ein Auswahlkriterienkatalog entstanden, der aus den folgenden zwei Kategorien besteht:

*Anwendungsbezogene* Auswahlkriterien beziehen sich auf allgemeine Aspekte bei der Anwendung eines Testprozessbewertungsansatzes.

*Bewertungsbezogene* Auswahlkriterien beziehen sich auf die Art und Weise wie ein bestimmter Ansatz die Bewertung des Testprozesses vornimmt.

Im Einzelnen handelt es sich um folgende Auswahlkriterien:

Anwendungsbezogene Auswahlkriterien	Bewertungsbezogene Auswahlkriterien
Zeit	Phasenabdeckung und Prozessgebiete
Kosten	Granularität
Akzeptanz im Unternehmen und Praxisbezug	Optionen und Prioritäten
Benutzerfreundlichkeit	Objektivität und Zuverlässigkeit
Anpassbarkeit	Rasche Einsicht in aktuelle Situation
Unabhängigkeit bzgl. anderer Modelle	Einhalten der Norm ISO 15504
Zugänglichkeit	Berücksichtigung der Norm ISO 9126
Zertifizierung	
Dokumentationsstand und Weiterentwicklung	

### 3 Bewertung

Die Untersuchung der Bewertungsmodelle TPI, TMM und TMMi gemäß der oben beschriebenen Auswahlkriterien zeigt die erheblichen Unterschiede der Bewertungsmodelle hinsichtlich ihrer Vorgehensweise, Struktur und Eigenschaften. Hierbei wird deutlich, dass bestimmte Bewertungsmodelle bestimmte Kriterien besonders gut erfüllen und umgekehrt einige Kriterien nicht so stark berücksichtigen. So konnte zum Beispiel festgestellt werden, dass das TMMi Bewertungsmodell gut bei den Kriterien *Objektivität und Zuverlässigkeit* sowie *Einhaltung der Norm ISO 15504* abschneidet. TMMi kann allerdings, aufgrund der engen Verbindung zum CMMI, keine gute *Anpassbarkeit* bieten, hier bietet das TPI-Modell mehr Möglichkeiten.

In der Präsentation werden wir einen weitreichenderen Überblick über die Vor- und Nachteile der einzelnen Bewertungsmodelle im Bezug auf die Auswahlkriterien geben. Wir werden diese Bewertungen der verschiedenen Bewertungsmodelle auch einander gegenüberstellen. Auf Grundlage dieser Vergleiche haben wir einen Leitfaden zur Auswahl von Bewertungsmodellen erarbeitet, der im Rahmen der Präsentation vorgestellt werden soll. Der Leitfaden soll Testmanagern konkrete Hilfestellung bieten um schnell das richtige Bewertungsmodell zur Testprozessbewertung für sein Projekt oder Unternehmen zu finden. Der Leitfaden gibt Empfehlungen welches Bewertungsmodell für welchen Zweck geeignet ist, in dem er die Bewertungen der einzelnen Auswahlkriterien in Betracht zieht.

Testprozessbewertungsmodelle werden durch Rückmeldungen aus der Praxis und auf Basis der Analyse ihrer Stärken und Schwächen ständig aktualisiert und weiterentwickelt. So wurde erst im November 2009 das TPI Modell durch eine neue Generation, das *Business Driven TPI* [Ham09], verbessert. Einige Nachteile des TPI-Modells, die auch in unserer Evaluierung erkannt wurden, wurden bei der Entwicklung des Business Driven TPI beachtet. Die Auswahl und der Einsatz eines bestimmten Bewertungsmodells können und sollten also immer wieder evaluiert werden, um mit dem gewählten Standard möglichst erfolgreich zu sein.

Ausblickend ist noch zu erwähnen, dass es neben den qualitativen Bewertungsansätzen (z.B. TPI, TMM und TMMi) auch quantitative Ansätze zur Testprozessbewertung gibt. Quantitative Ansätze, wie zum Beispiel das *Software Test Process Evaluation Framework* (STP-EF) [FD08b], bewerten einen Testprozess durch die Erhebung von Testmetriken innerhalb des Testprozesses. Eine Untersuchung dieses Bewertungsansatzes im Kontext der oben beschriebenen Vorgehensweise (ggf. auch mit zusätzlichen Kriterien) und die Unterscheidung dieser zum qualitativen Bewertungsansatz sind anstehende interessante Schritte aus der Sicht der Entwicklungen im Bereich der Testprozessbewertungsansätze.

### Literaturverzeichnis

[BUR03] BURNSTEIN, Ilene: *Practical Software Testing*. Springer-Verlag New York, Inc., 2003. – ISBN 0–387–95131–8

[FD08a] FAROOQ, Ayaz ; DUMKE, Reiner R.: *Developing and Applying a Consolidated Evaluation Framework to Analyze Test Process Improvement Approaches*. Version: 2008. <http://www.springerlink.com/content/328423m58600w43p/>, Abruf: 04.06.2009

## 6.2 Erfolgsfaktoren von Testprozessbewertungsmodellen

- [FD08b] FAROOQ, Ayaz; DUMKE, Reiner R.: A quantitative Evaluation Framework for the Software Test Process. In: *Setting Quality Standards*, CONQUEST 2008, S.1-14
- [FD08c] FAROOQ, Ayaz; DUMKE, Reiner R.: *Evaluation Approaches in Software Testing*. Version: 2008. [http://ivs.cs.uni-magdeburg.de/~farooq/publications\\_files/TR\\_Farooq.pdf](http://ivs.cs.uni-magdeburg.de/~farooq/publications_files/TR_Farooq.pdf)  
Abruf: 02.12.2009, Technical Report.
- [Ham09] HAMBURG, Matthias: *Das neue TPI-Erste Erfahrungen*. Version 1.  
[http://www.gm.fh-koeln.de/~winter/tav/html/tav29/TAV29P05Hamburg\\_Das%20neue%20TPI%20V10.pdf](http://www.gm.fh-koeln.de/~winter/tav/html/tav29/TAV29P05Hamburg_Das%20neue%20TPI%20V10.pdf), Abruf: 02.12.2009.
- [HC01] HALVORSEN, Christian P. ; CONRADI, Reidar: *A Taxonomy to Compare SPI Frameworks*. Version: 2001. <http://www.springerlink.com/content/22ljfvy6mhe5qbmc/>, Abruf: 04.06.2009
- [PKS00] POL, Martin; KOOMEN, Tim; SPILLNER, Andreas: *Management und Optimierung des Testprozesses*. dpunkt.verlag, 2000.  
ISBN 3-932588-65-7
- [SC99] SAIEDIAN, Hossein ; CHENNUPATI, Kalyani: *Towards an evaluative framework for software process improvement models*. Version: 1999.  
[Journal of Systems and Software, Volume 47, Issues 2-3](#), 1 July 1999, Pages 139-148
- [SL05] SPILLNER, Andreas ; LINZ, Tilo: *Basiswissen Softwaretest*.  
dpunkt.verlag, 2005. – ISBN 3-89864-358-1
- [Vee09] VEENENDAAL, Erik van: *Test Maturity Model Integration (TMMi)*.  
Version: 2009. <http://www.tmmifoundation.org/downloads/tmmi/TMMi%20Framework.pdf>, Abruf: 02.12.2009.
- [Yig09] YIGITBAS, Enes: *Evaluierung von Bewertungsmodellen für den Testprozess*,  
Universität Paderborn – Software Quality Lab (s-lab), Bachelorarbeit, Juli 2009.



## Erfolgsfaktoren von Testprozessbewertungsmodellen

Yavuz Sancar, Frank Brüseke, Enes Yigitbas  
SEE 2010 – 03. Mai 2010



### Software Quality Lab



- Offenes Multi Private Public Partnership (Multi-PPP) Institut (Gründung 2005)
- Kompetenz- & Technologietransfer zu Themen der Softwaretechnik
- Erhöhung der Qualität in der industriellen Softwareentwicklung
- Impulse und damit die qualitative Verbesserung der universitären Forschung und Lehre
- Fünf Hochschullehrer
- Acht assoziierte Industriepartner + zehn Projektpartner



## Kooperationsformen



- Forschungs- und Entwicklungsprojekte
  - Auftragsforschung
  - Gemeinsame Förderprojekte (BMBF, EU, usw.)
- Kleinprojekte (Studien, Beratung, Entwicklung)
- Bachelor-/Masterarbeiten
- Schulungen/Workshops
- Praktikum
- gemeinsame Marketing-Aktionen (Messe, usw.)
  
- Bilaterale und Multilaterale Projekte

3

## Die Partner des s-lab



### Assoziierte Partner



### Projektpartner



Bauhaus-Universität Weimar

4



## Softwarequalität



Ein entscheidender Wettbewerbsvorteil!



Yavuz Sancar, Frank Brüseke, Enes Yigitbas – SEE 2010 – 03. Mai 2010

5

## Konstruktive QS



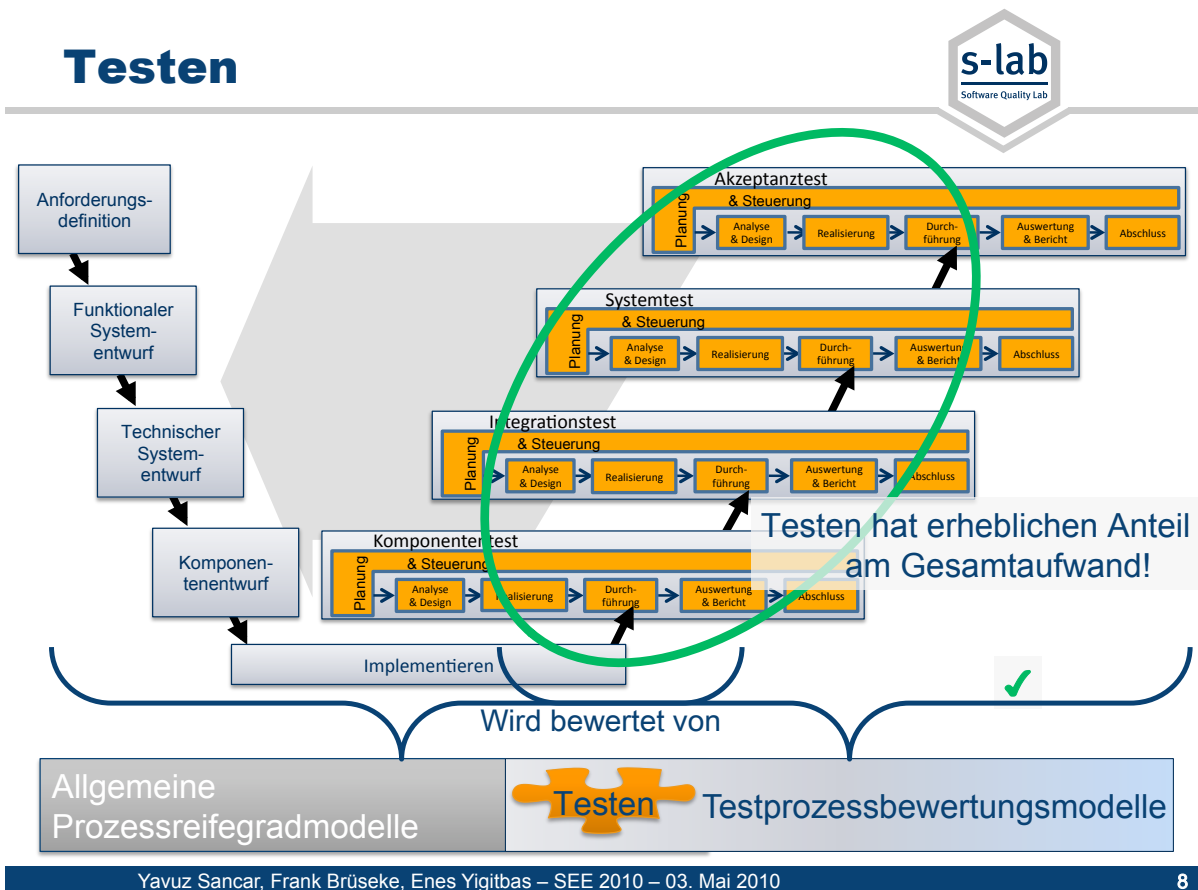
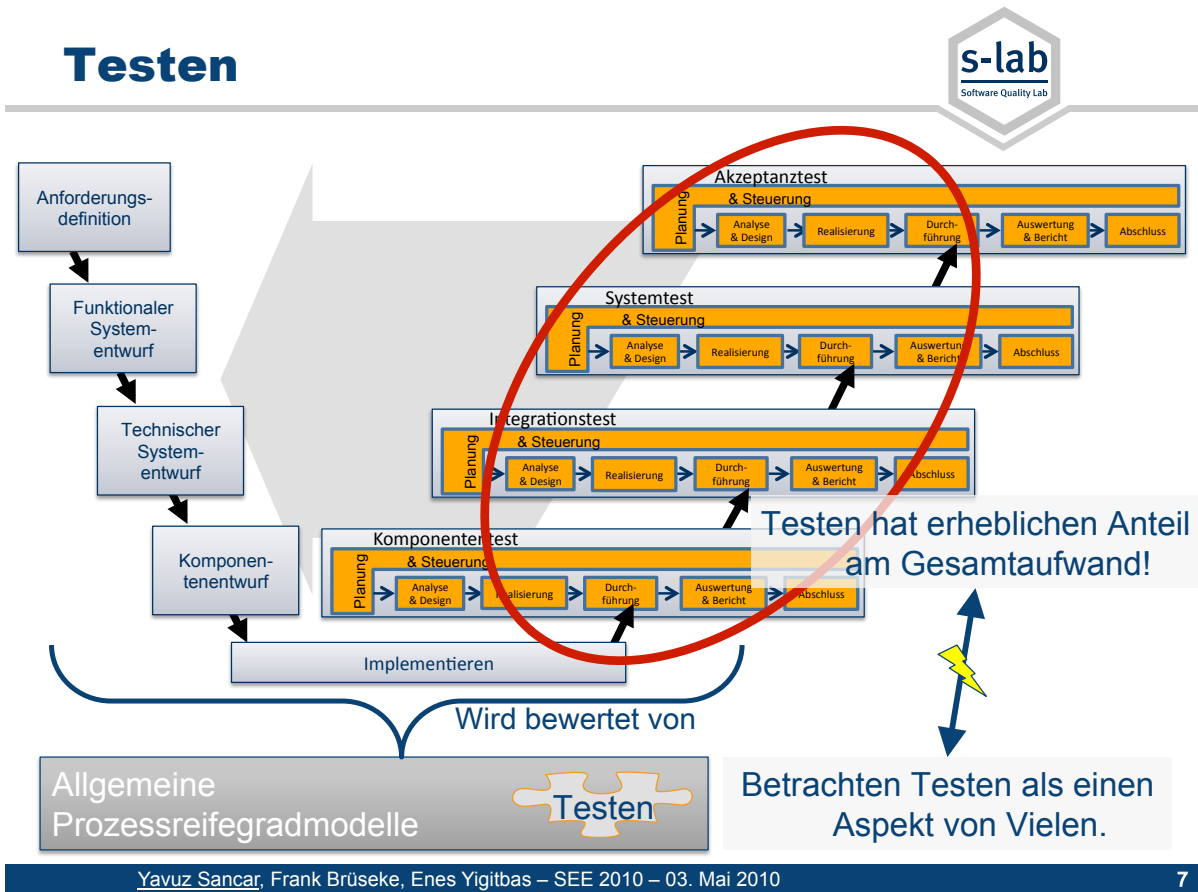
Allgemeine  
Prozessreifegradmodelle



- Bewerten den Softwareentwurfsprozess
- Decken Verbesserungspotential auf
- Tragen so zur Softwarequalität des Endprodukts bei

Yavuz Sancar, Frank Brüseke, Enes Yigitbas – SEE 2010 – 03. Mai 2010

6



## Testprozessbewertungsmodelle



Es existieren unterschiedliche Bewertungsmodelle:

- Test Process Improvement (TPI)
- Testing Maturity Model (TMM)
- Test Maturity Model Integration (TMMi)
- Test Improvement Model (TIM)
- Test Process Assessment Model (TPAM)
- Test Organization Maturity Model (TOM)
- ...

Wie kann die Entscheidung für den Einsatz eines Testprozessbewertungsmodell unterstützt werden?


## Evaluierung notwendig



### Kriterien für eine Vorauswahl:

- Hoher Praxisbezug
- Betrachtung des Testprozesses in seiner Gesamtheit
- Gängiges und lebendiges Bewertungsmodell
- Detailliertes Informationsmaterial verfügbar

### Evaluierungskandidaten:

- 
- Test Process Improvement (TPI)
  - Testing Maturity Model (TMM)
  - Test Maturity Model Integration (TMMi)



# Vorstellung der Bewertungsmodelle

## TPI



### Herkunft und Verbreitung:

- Vom Unternehmen IQUIP (seit 2002: Sogeti) im Jahre 1997 entworfen
- Wird weltweit eingesetzt und vermehrt in folgenden Ländern: Niederlande, USA, Deutschland und Finnland

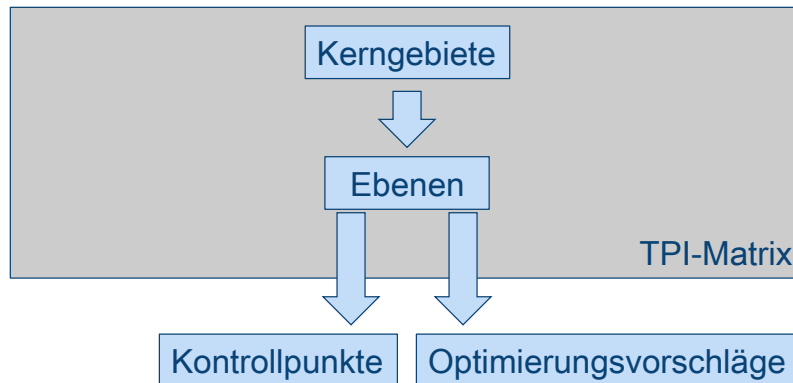
### Ursprung und Submodelle:

- Wurde auf der Grundlage des Einsatzes von TMap und den Erfahrungen einer Vielzahl professioneller Tester entworfen

# TPI



## Aufbau und Struktur:



Der Ist-Zustand des Testprozesses wird anhand der Kontrollpunkte festgestellt.

Abbildung nach:

Pol, M. ; Koomen, T. ; Spillner, A.: Management und Optimierung des Testprozesses. dpunkt.verlag, 2000.

Yavuz Sancar, Frank Brüseke, Enes Yigitbas – SEE 2010 – 03. Mai 2010

# TPI-Matrix



Kernbereich / Stufe	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
1 Teststrategie		A					B				C		D		
2 Einsatz des Phasenmodells		A			B										
3 Zeitpunkt der Beteiligung			A				B				C		D		
4 Kostenvoranschlag und Planung				A							B				
5 Test-Spezifikationstechniken		A		B											
6 Statische Testtechniken					A		B								
7 Metriken						A		B				C		D	
8 Testautomatisierung					A			B			C				
9 Testumgebung				A				B						C	
10 Testarbeitsplatz				A											
11 Engagement und Motivation		A				B							C		
12 Testfunktionen und Ausbildung				A			B				C				
13 Reichweite der Methodik					A						B			C	
14 Kommunikation			A		B								C		
15 Berichterstattung		A			B		C						D		
16 Dokumentation der Abweichungen		A				B		C							
17 Testware-Management			A			B				C				D	
18 Testprozessmanagement		A		B									C		
19 Prüfen							A			B					
20 Low-Level-Tests					A		B		C						
		beherrschbar					effizient					optimierend			

Abbildung nach:

Pol, M. ; Koomen, T. ; Spillner, A.: Management und Optimierung des Testprozesses. dpunkt.verlag, 2000.

Yavuz Sancar, Frank Brüseke, Enes Yigitbas – SEE 2010 – 03. Mai 2010

# TMM



## Herkunft und Verbreitung:

- Wurde von Ilene Burnstein et al. im Jahre 1996 am Illinois Institute of Technology in Chicago entwickelt
- Große Verbreitung in den USA

## Ursprung und Submodelle:

- CMM
- Softwaretestpraktiken aus der Industrie
- Testmodelle (Gelperin, Hetzel, Beizer)

# TMM



## Aufbau und Struktur:

- Levels legen den Reifegrad fest und indizieren Testqualität
- Maturity Goals bzw. Prozessgebiete müssen „gelebt“ werden
- Maturity Subgoals sind konkreter formulierte Hinweise
- ATRs sind ausführlich beschrieben und müssen durchgeführt werden

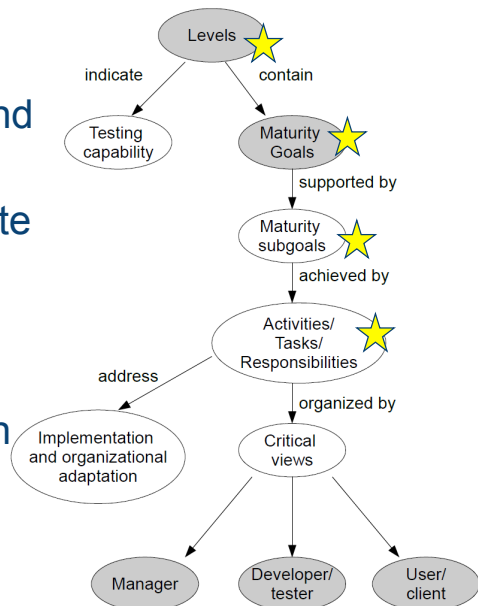


Abbildung nach:

Burnstein, Ilene: Practical Software Testing. 1st ed. Springer-Verlag New York, Inc., 2003.

## TMM Reifegradstufen und -ziele

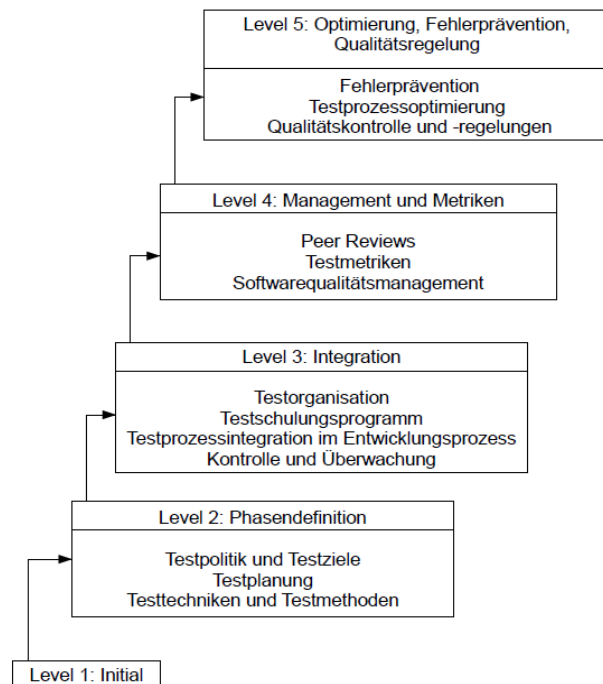


Abbildung nach:

Burnstein, Ilene: Practical Software Testing. 1st ed. Springer-Verlag New York, Inc., 2003.

Yavuz Sancar, Frank Brüseke, Enes Yigitbas – SEE 2010 – 03. Mai 2010

17

## TMMi



### Herkunft und Verbreitung:

- Wurde von einer gemeinnützigen Organisation TMMi Foundation, sesshaft in Irland, entwickelt
- 2008: Veröffentlichung TMMi Version 1.0
- 2009: Veröffentlichung TMMi Version 2.0
- Steigende Verbreitung in Europa, Asien und USA

### Ursprung und Submodelle:

- Grundlagen und Ideen von TMM bzw. CMM wurden übernommen
- Entwicklungen zu CMMI wurden berücksichtigt
- IEEE 829 Standard für Software Test Dokumentation wurde berücksichtigt
- ISTQB-Terminologie für Softwaretest wurde eingehalten

Yavuz Sancar, Frank Brüseke, Enes Yigitbas – SEE 2010 – 03. Mai 2010

18



# TMMi



## Aufbau und Struktur:

- Geforderte Komponenten
  - Was muss die Organisation erreichen, um das Prozessgebiet zufriedenstellend abzudecken.
  
- Erwartete Komponenten
  - Was wird Organisation typischerweise tun, um eine geforderte Komponente zu erfüllen.
  
- Informative Komponenten
  - Bieten detaillierte Hilfestellung im Umgang mit geforderten und erwarteten Komponenten

# TMMi Struktur

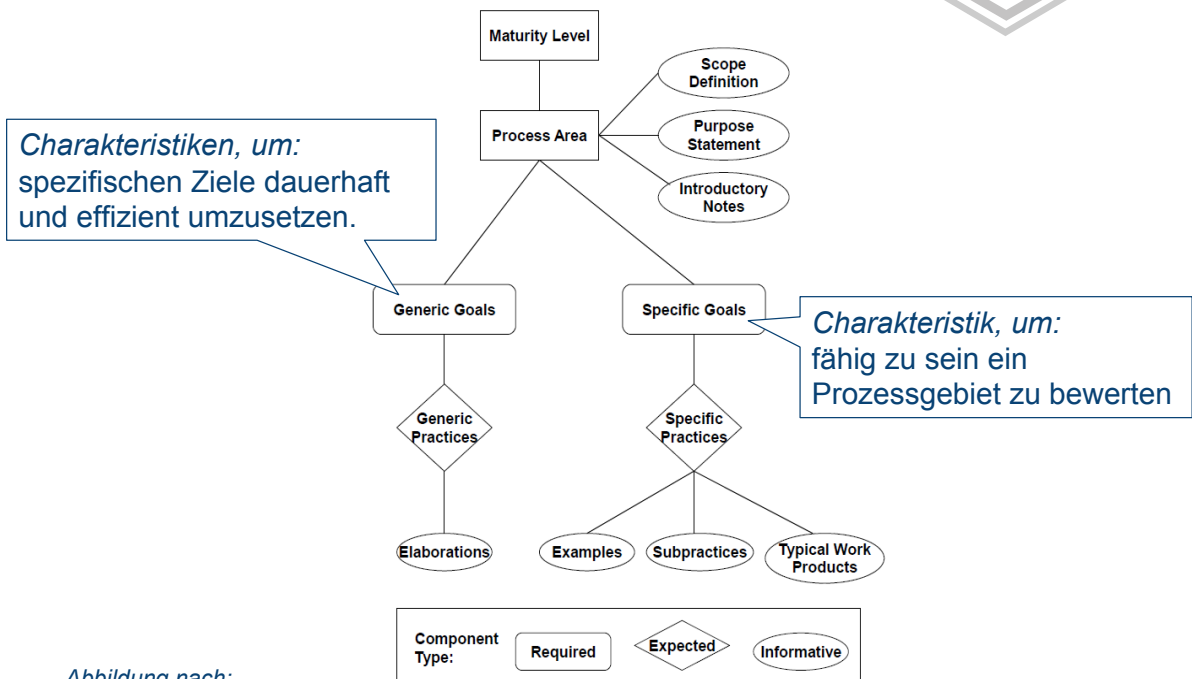


Abbildung nach: Farooq, Ayaz ; Dumke, Reiner R.: Evaluation Approaches in Software Testing, 2008.

## TMMi Reifegradstufen und -ziele

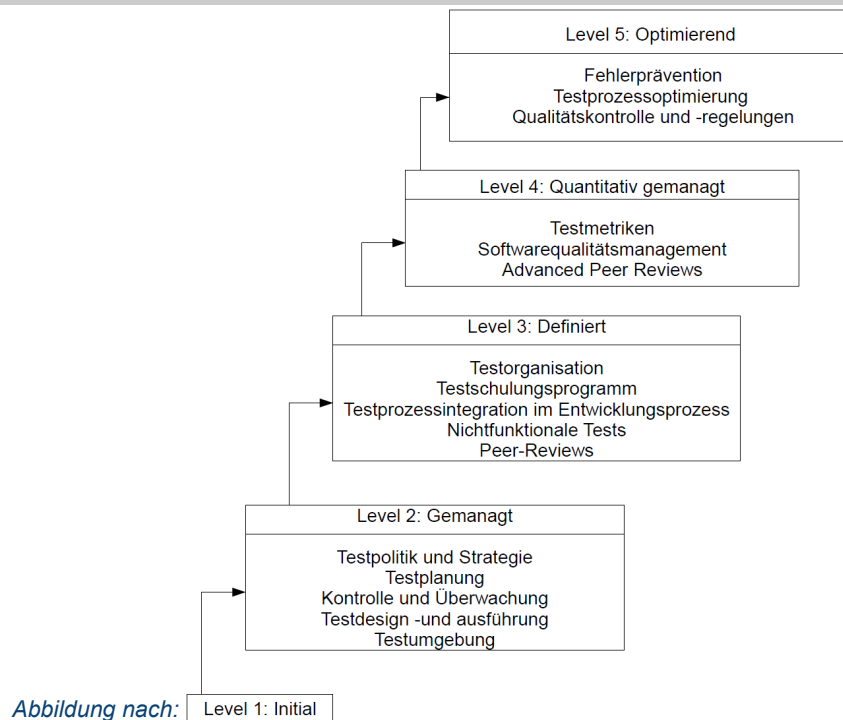


Abbildung nach:

Level 1: Initial

Veenendaal, Erik van: Test Maturity Model Integration (TMMi). Version: 2009.

Yavuz Sancar, Frank Brüseke, Enes Yigitbas – SEE 2010 – 03. Mai 2010

21

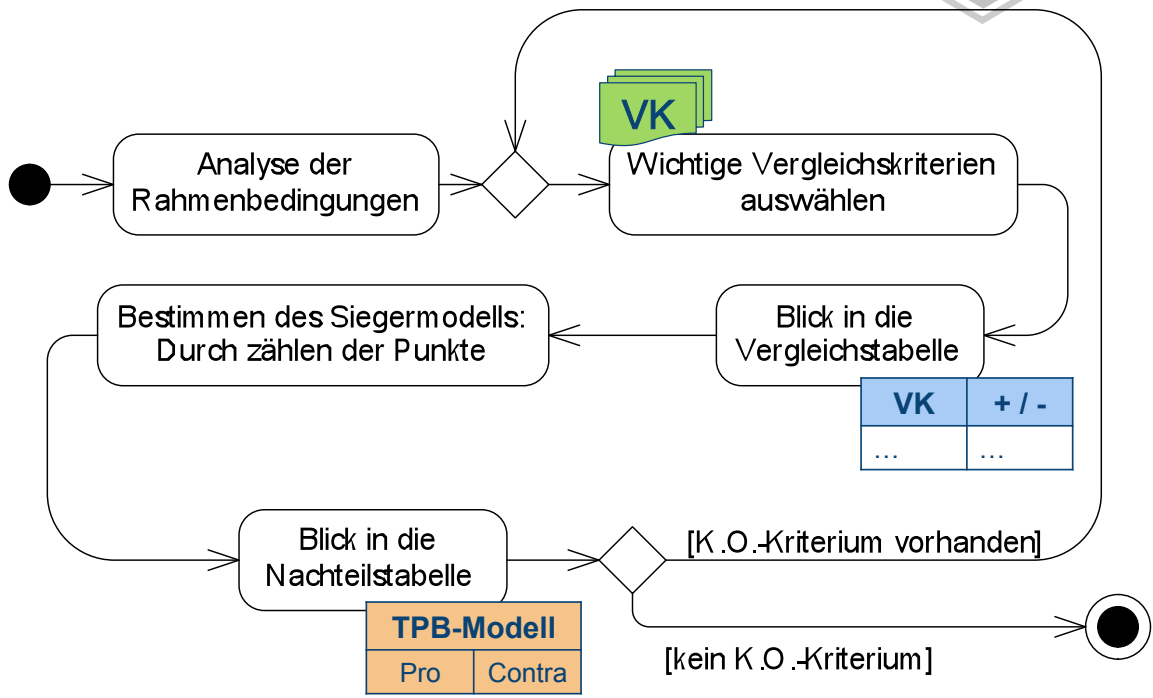


## Erfolgsfaktoren für die Auswahl

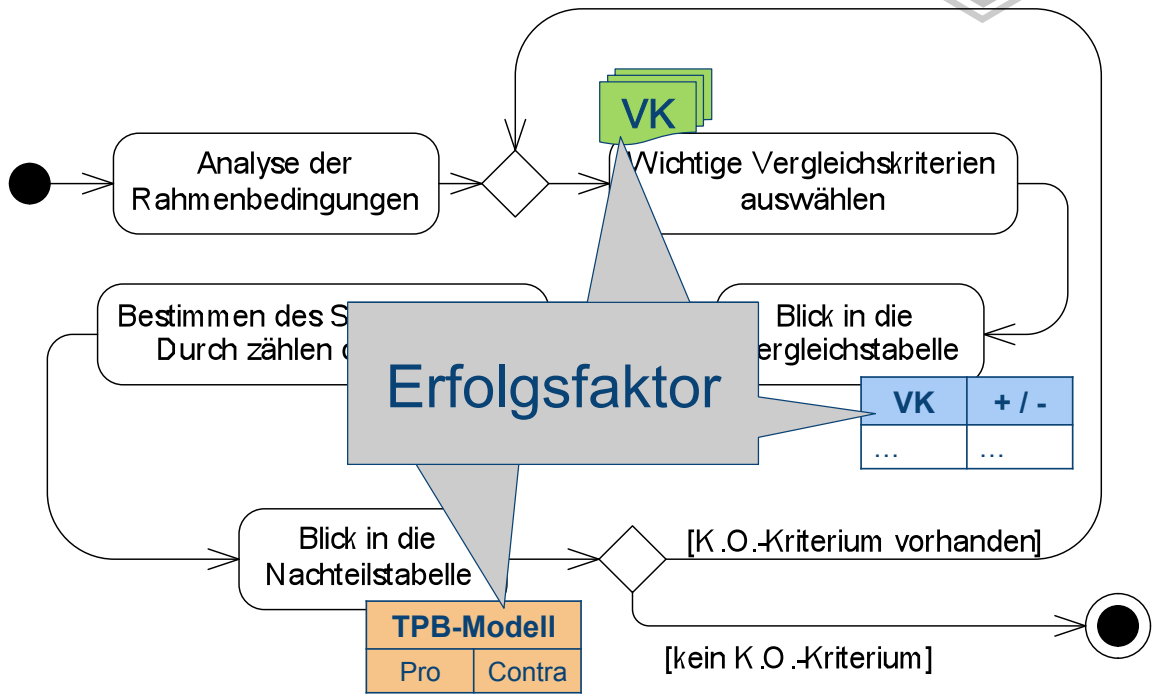
Yavuz Sancar, Frank Brüseke, Enes Yigitbas – SEE 2010 – 03. Mai 2010

22

# Auswahlprozess



# Auswahlprozess



## Vergleichskriterien



### Ergebnisse unserer Evaluierung:

Gefundene Vergleichskriterien (in zwei Kategorien) und deren Bewertung

### Anwendungsbezogene Vergleichskriterien:

Dienen zum Vergleich von Aspekten, die bei der Anwendung bzw. beim Einsatz eines Bewertungsmodells eine entscheidende Rolle spielen.

### Bewertungsmodellbezogene Vergleichskriterien:

Diese Kategorie enthält Vergleichskriterien, die sich beim Vergleich der einzelnen Bewertungsmodelle speziell mit der Art und den jeweiligen Eigenschaften des Bewertungsmodells beschäftigen.

Yavuz Sancar, Frank Brüseke, Enes Yigitbas – SEE 2010 – 03. Mai 2010

25

## Vergleichstabelle (1)

VK	+ / -
...	...



Anwendungsbezogene Vergleichskriterien	TPI	TMM	TMMi
Zeit (Dauer der Bewertung/ Einführung)	ca. 1 Monat	ca. 3 Monate	abhängig von der akkreditierten Assessment-Methode
Kosten	mittel	mittel	hoch
Akzeptanz im Unternehmen und Praxisbezug	hoch	hoch	mittel
Benutzerfreundlichkeit (Verständlichkeit / Erlernbarkeit)	hoch	mittel	hoch
Anpassbarkeit	hoch	mittel	mittel
Unabhängigkeit bzgl. anderer Modelle	hoch	mittel	mittel
Zugänglichkeit	frei	frei	eingeschränkt
Zertifizierung	nein	nein	ja
Dokumentationsstand und Weiterentwicklung	o	-	+

Legende	+	o	-
	Kriterium wird voll bzw. in positiver Weise erfüllt	Kriterium wird nur teilweise erfüllt	Kriterium wird nicht oder schlecht erfüllt

Yavuz Sancar, Frank Brüseke, Enes Yigitbas – SEE 2010 – 03. Mai 2010

26

# Vergleichstabelle (2)

VK	+ / -
...	...

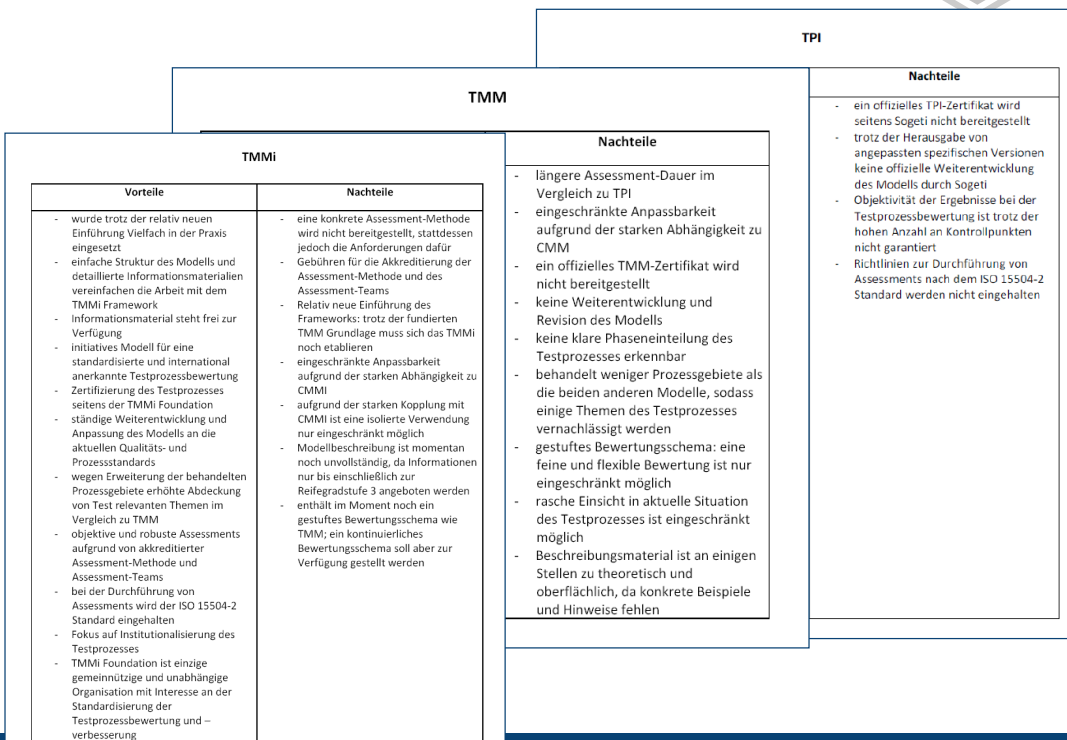


Vergleichskriterien bzgl. der Bewertung	TPI	TMM	TMMi
Phasenabdeckung und Prozessgebiete	+	o	+
Granularität	fein	grob	mittel
Optionen und Prioritäten	+	o	o
Objektivität und Zuverlässigkeit	o	o	+
Rasche Einsicht in aktuelle Situation	+	o	o
Einhaltung von Assessment-Prozess-Standards (ISO/IEC 15504-2)	-	o	+
Berücksichtigung von SW-Qualitätsmerkmalen (ISO 9126)	+	o	o

Legende	+	o	-
	Kriterium wird voll bzw. in positiver Weise erfüllt	Kriterium wird nur teilweise erfüllt	Kriterium wird nicht oder schlecht erfüllt

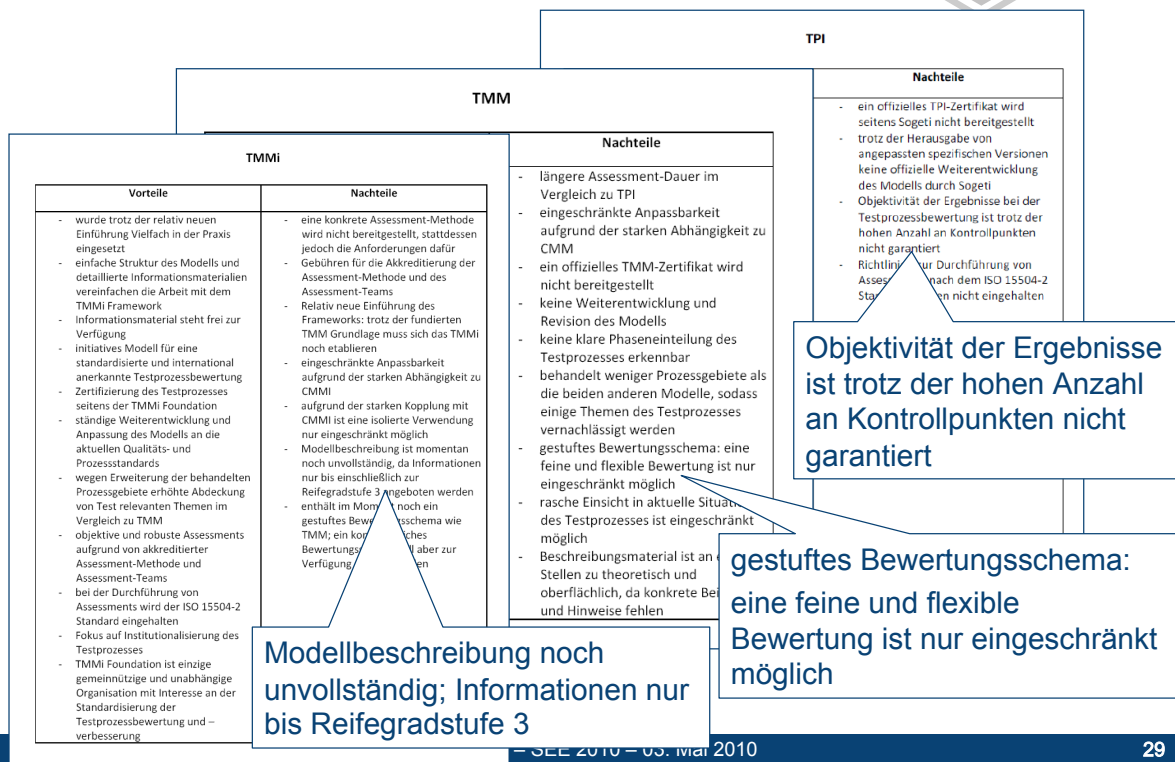
# Nachteile-Tabellen

TPB-Modell
Pro Contra



# Nachteile-Tabellen

TPB-Modell	
Pro	Contra



# Beispiele

## Szenario A






- Unternehmen mit ca. 10.000 Mitarbeiter weltweit in der Finanzbranche tätig
- Marketing- und Wettbewerbsvorteil durch Einsatz eines zertifizierten Testprozesses
- In diesem Projekt: Testprozess nach ISTQB
  - Projekt ist Pilotprojekt für die Einführung eines Testprozessbewertungsmodells
- Unternehmensweite Einführung des Testprozessbewertungsmodells wird angestrebt
  - ➔ praxiserprobtes Testprozessbewertungsmodell benötigt



## Szenario A



- Auszug aus der Vergleichstabelle, enthält nur die ausgewählten Vergleichskriterien:

Anwendungsbezogene Vergleichskriterien	TPI 	TMM 	TMMi 
Akzeptanz im Unternehmen und Praxisbezug	hoch	hoch	mittel
Zertifizierung	nein	nein	ja
Phasenabdeckung und Prozessgebiete	+	o	+
Objektivität und Zuverlässigkeit	o	o	+
Einhaltung von Assessment-Prozess-Standards (ISO/IEC 15504-2)	-	o	+

- In der Nachteiltabelle finden sich keine K.O.-Kriterien
  - ➔ Ergebnis: TMMi



## Szenario B



- Mittelständisches Unternehmen mit 120 Mitarbeitern
- Stellt Individualsoftware für Office-Anwendungen her
- Aktueller Projektauftrag incl. Wartungsvertrag:
  - Kunde verlangt genauen Nachweis über die erbrachte Testqualität
    - Testqualität muss schnell erhoben werden können
    - Einsatz des ISTQB-Testprozesses



## Szenario B



- Auszug aus der Vergleichstabelle, enthält nur die ausgewählten Vergleichskriterien:

Anwendungsbezogene Vergleichskriterien	TPI <b>1.</b>	TMM <b>3.</b>	TMMi <b>2.</b>
Benutzerfreundlichkeit (Verständlichkeit / Erlernbarkeit)	hoch	mittel	hoch
Unabhängigkeit bzgl. anderer Modelle	hoch	mittel	mittel
Phasenabdeckung und Prozessgebiete	+	o	+
Granularität	fein	grob	mittel
Rasche Einsicht in aktuelle Situation	+	o	o

- In der Nachteiltabelle finden sich keine K.O.-Kriterien  
➔ Ergebnis: TPI

# Zusammenfassung



## Testprozessbewertungsmodelle

- Keine generelle Empfehlung möglich
- Auswahl für ein Modell ist abhängig von dem jeweiligen Szenario

## Auswahlprozess

- Vergleichskriterien
- Vergleichstabelle
- Nachteileitabelle



VK	+ / -
...	...

TPB-Modell	
Pro	Contra

## Ausblick

- Berücksichtigen weiterer/neuer Modelle (Business TPI, TMMi)
- Berücksichtigung von Tailoring im Auswahlprozess

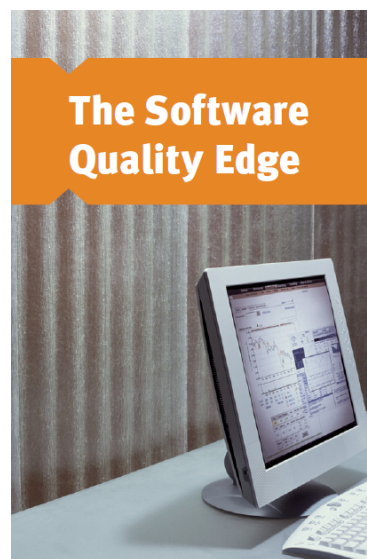


# Vielen Dank für Ihre Aufmerksamkeit.

### Software Quality Lab (s-lab)

Universität Paderborn  
 Warburger Str. 100  
 33098 Paderborn  
 Tel.: (05251) 60 5390 / 5391

<http://s-lab.upb.de>  
[s-lab-info@s-lab.upb.de](mailto:s-lab-info@s-lab.upb.de)



## Erweiterung des TPI-Modells zur Reifegradbewertung unter Berücksichtigung von projekteigenen Anforderungen

Yavuz Sancar<sup>1</sup>, Claudia Schumacher<sup>2</sup>

Universität Paderborn  
Institut für Informatik  
33098 Paderborn, Deutschland

<sup>1</sup>ysancar@s-lab.upb.de

<sup>2</sup>clasch@mail.uni-paderborn.de

### 1 Motivation

Die Anforderungen an die Qualität von Software und softwarebasierten Systemen steigen ständig. Um die angestrebte Softwarequalität zu erreichen, existieren sowohl konstruktive als auch analytische Maßnahmen. Die konstruktiven Maßnahmen sind vorbeugend und versuchen, Fehler zu vermeiden – ein Beispiel für diese Art ist das V-Modell. Das Testen stellt eine analytische Qualitätssicherung dar [BALZ98] und ist ein Bestandteil gängiger Entwicklungsprozesse (Abbildung 1). Dabei ist die Erkenntnis wichtig, dass auch das Testen selbst ein eigenständiger zu optimierender Prozess ist. Zu diesem Zweck existieren Bewertungs- und Verbesserungsmodelle. Exemplarisch für ein Bewertungs- und Verbesserungsmodell ist das *Test Process Improvement Model* (TPI-Modell) nach Sogeti. Dieses teilt den Testprozess in Kernbereiche ein und ermittelt seinen Reifegrad. Dem ermittelten Reifegrad entsprechend bietet das TPI-Modell dann Optimierungsvorschläge an, um den Testprozess zu verbessern [POL00].

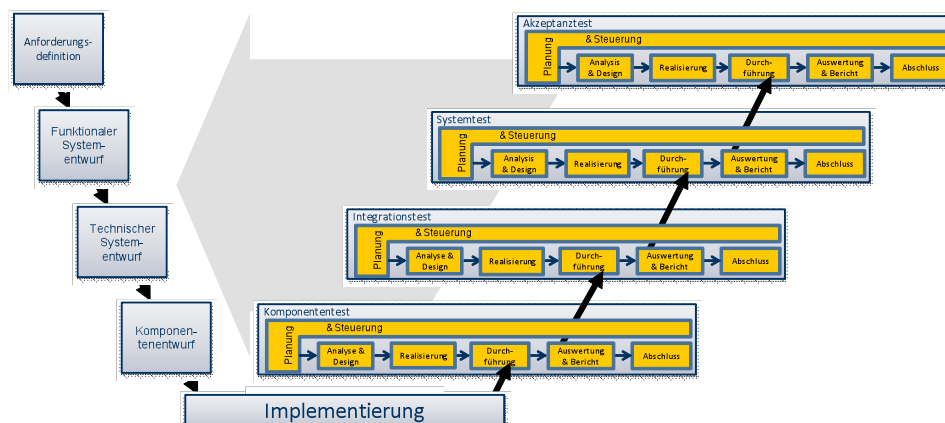


Abbildung 1: Der Testprozess als eigenständiger Prozess innerhalb des Entwicklungsprozesses

Das TPI-Modell hat sich durch objektive Bewertungskriterien und reproduzierbare Ergebnisse in der Praxis bewährt und wird weiterhin vielfach angewendet. Nach unserer Erfahrung in dem Einsatz des TPI-Modells hat es sich als schwierig erwiesen projekteigene Anforderungen an einen Testprozess in der Bewertung durch das TPI-Modell zu berücksichtigen. Beispielsweise kann es sinnvoll sein, von Projekt zu Projekt eine unterschiedliche Priorisierung der Kernbereiche vorzunehmen oder zusätzliche Kernbereiche

je nach Projekt oder je nach Domäne in der sich ein Unternehmen befindet, hinzuzufügen.

In unserer Untersuchung haben wir festgestellt, dass das TPI-Modell erweitert werden muss, um die oben genannten projekteigenen Anforderungen besser zu berücksichtigen [SCHU09].

## 2 Ansatz

Das Priorisieren der TPI-Kernbereiche erreichen wir durch eine Verschiebung der zugehörigen Ebenen innerhalb der Entwicklungsstufen. Bei diesem Vorgang müssen die im TPI-Modell definierten Abhängigkeiten zwischen den Ebenen der Kernbereiche berücksichtigt werden. Für das Hinzufügen von TPI-Kernbereichen haben wir das zugrundeliegende fachliche Datenmodell der TPI-Matrix herausgearbeitet, welches alle zu definierten Daten angibt. Auf dieser Grundlage haben wir ein Werkzeug implementiert, welches unseren Ansatz der Erweiterung realisiert, sodass die Bewertung und Auswertung des Testprozesses softwaregestützt durchgeführt werden kann.

## 3 Bewertung

Unsere Erfahrungen in der Praxis haben gezeigt, dass es sinnvoll ist projekteigene Anforderungen an einen Testprozess bei der Bewertung und Optimierung des betrachteten Testprozesses zu berücksichtigen. In unserer Präsentation zeigen wir Fallbeispiele, in denen die oben beschriebenen Erweiterungen des TPI-Modells Sinn ergeben. Wir zeigen den Mehrwert des Priorisierens und des Hinzufügens von Kernbereichen und stellen anschließend unseren Lösungsansatz vor. Ein Vorteil dieses Ansatzes ist, dass der Aufbau des ursprünglichen TPI-Modells bestehen bleibt.

Ergänzend zeigen wir in unserer Präsentation die softwaregestützte Bewertung und Auswertung nach dem erweiterten TPI-Modell sowie die zugrundeliegende Abhängigkeiten-Matrix und das fachliche Datenmodell.

## Literaturverzeichnis

[BALZ98] BALZERT, Helmut: Lehrbuch der Software-Technik, Spektrum-Akademischer Verlag, 1996. ISBN 3-8274-0065-1

[POL00] POL, Martin; KOOMEN, Tim; SPILLNER, Andreas: *Management und Optimierung des Testprozesse*, dpunkt.verlag, 2000. ISBN 3-932588-65-7

[SCHU09] SCHUMACHER, Claudia: *Analyse und Erweiterung eines Testprozessbewertungsmodells zur Nutzung projektspezifischer Testprozesse*, Universität Paderborn – Software Quality Lab (s-lab), Bachelorarbeit, 2009.



## Erweiterung des TPI-Modells zur Reifegradbewertung unter Berücksichtigung von projekteigenen Anforderungen

Yavuz Sancar, Claudia Schumacher

SEE 2010 – 03. Mai 2010



## Softwarequalität



Ein entscheidender Wettbewerbsvorteil!



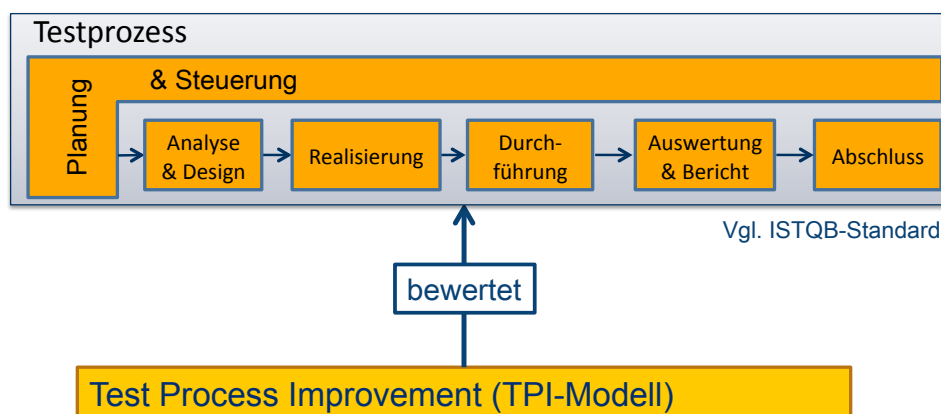
## Prozessbewertung



- Im SW-Entwicklungsprozess wird ein signifikanter Anteil an Ressourcen und Aufwand für Testaktivitäten verbraucht.
- Testprozessbewertungsmodelle sollen durch eine detailliertere Betrachtung Stärken und Schwächen verdeutlichen.

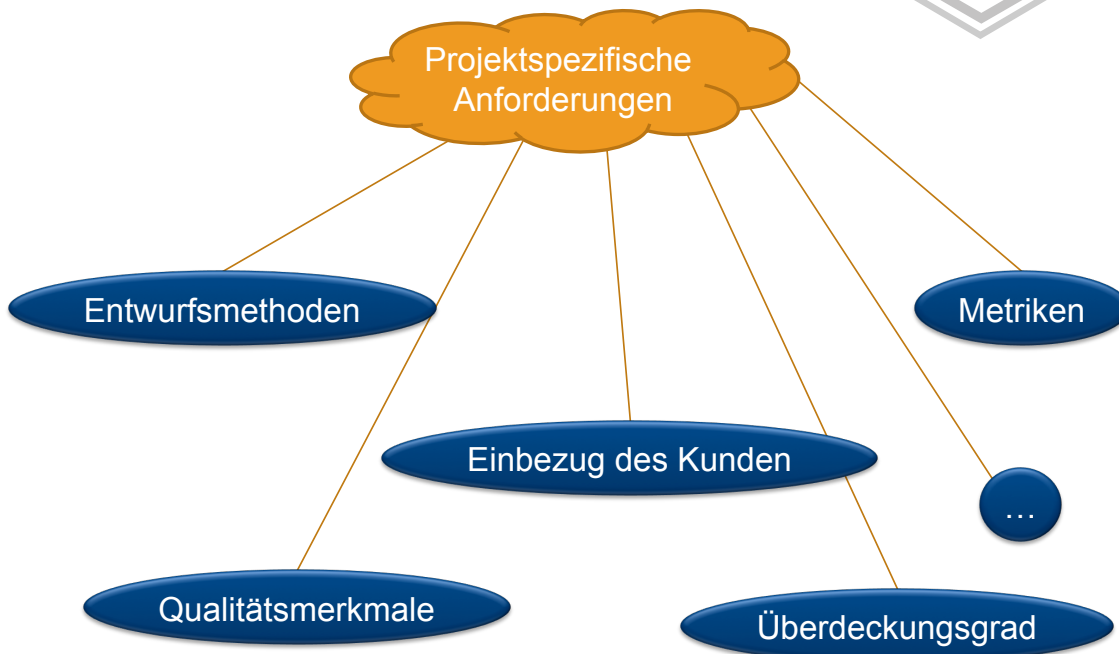


## Ein Beispiel

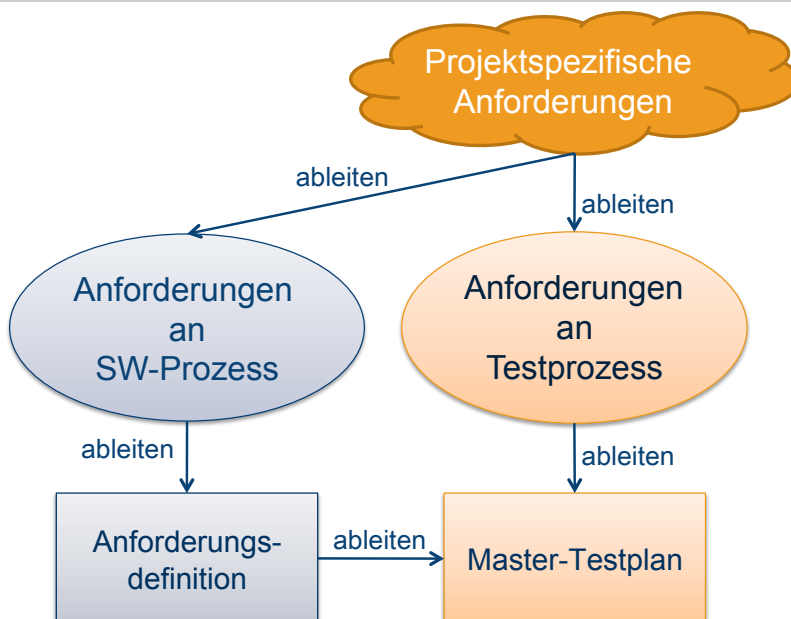


Ist die Ausführung eines Testprozesses immer gleich?

## Spezifische Anforderungen

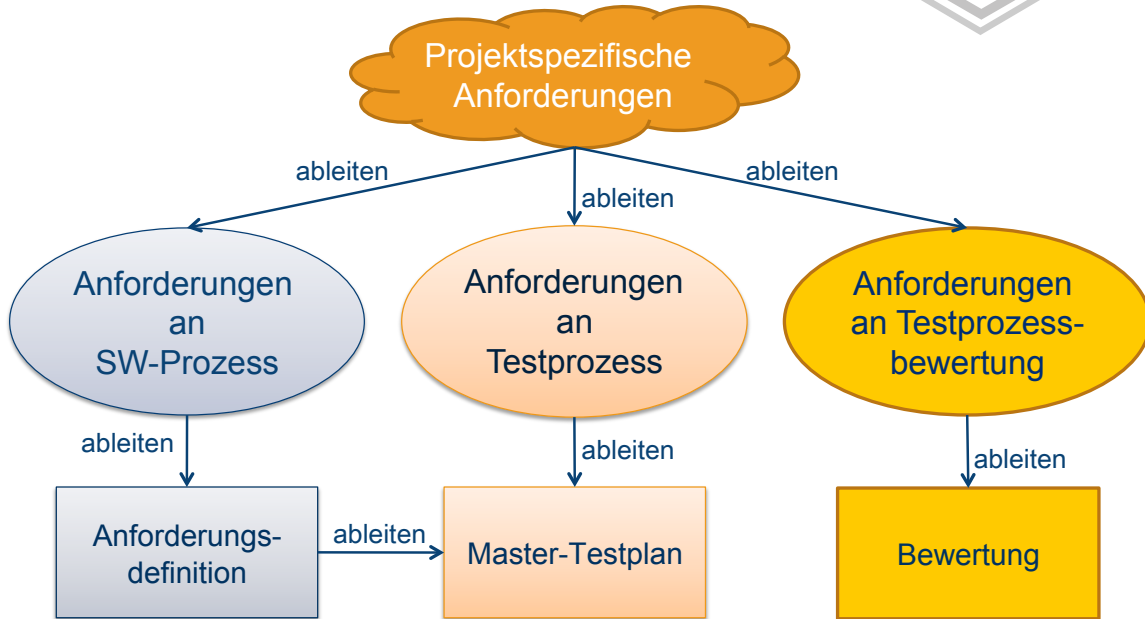


## Spezifische Anforderungen





## Spezifische Anforderungen

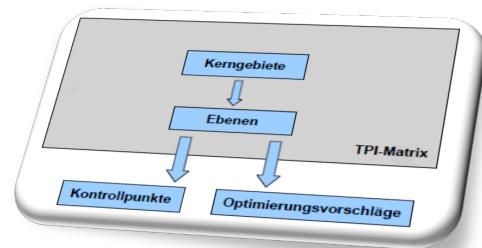


## Zielsetzung



### Erweiterung des TPI-Ansatzes

- Berücksichtigung von projektspezifischen Anforderungen
- Erhaltung der Grundstruktur des TPI-Modells



Hinzufügen von Kernbereichen

Priorisieren von Kernbereichen

## Hinzufügen von Kernbereichen



- Sicherstellung der Überprüfung der Qualitätsmerkmale im Testprozess.
- Qualitätsmerkmale werden in der Testprozessbewertung entsprechend den Anforderungen priorisiert.

Yavuz Sancar, Claudia Schumacher – SEE 2010 – 03. Mai 2010

9

## Hinzufügen von Kernbereichen



### Hinzufügen der Qualitätsmerkmale als Kernbereiche

18	Testprozessmanagement		A		B									C		
19	Prüfen						A			B						
20	Low-Level-Tests				A		B		C							
		beherrschbar					effizient					optimiert				
21	Qualitätsmerkmal Effizienz									A						
22	Qualitätsmerkmal Zuverlässigkeit				A											



TPI-Struktur



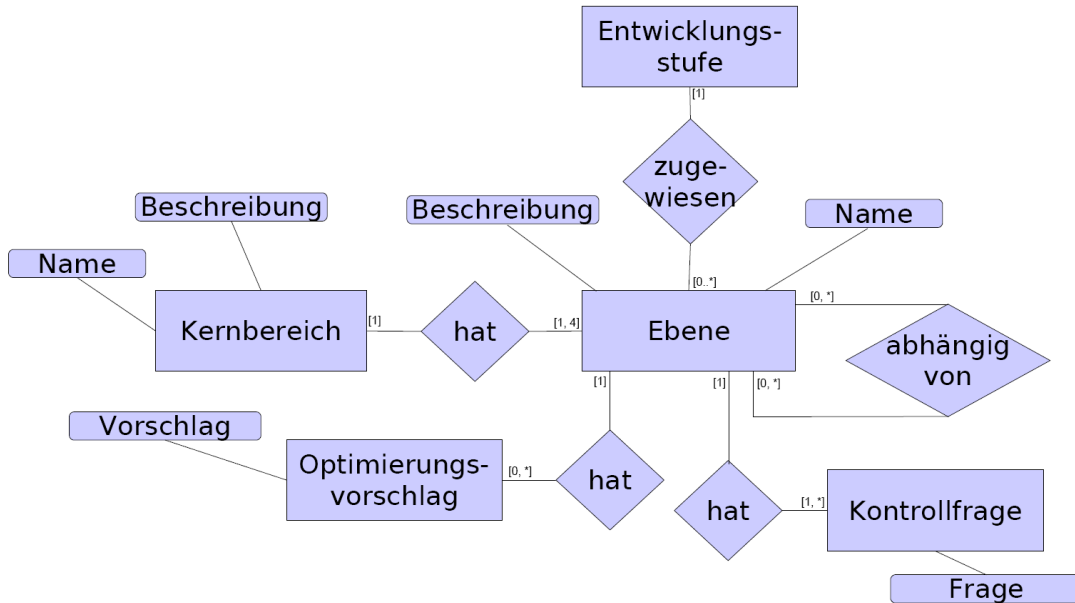
Yavuz Sancar, Claudia Schumacher – SEE 2010 – 03. Mai 2010

10

# TPI-Struktur



## Datenmodell eines Kernbereichs



## Beispiel Qualitätsmerkmale 1/4



<b>Kernbereich: Qualitätsmerkmal Zuverlässigkeit</b>	
Name	Qualitätsmerkmal Zuverlässigkeit
Beschreibung	Der Kernbereich Qualitätsmerkmal Zuverlässigkeit beschreibt die Überprüfung der Zuverlässigkeit nach ISO9126
Anzahl der Ebenen	1

Ebene A	
Name	Testen des Qualitätsmerkmals Zuverlässigkeit
Beschreibung	In dieser Ebene wird die Fähigkeit der Software überprüft, ihr Leistungsniveau unter vorgegebenen Bedingungen über eine bestimmte Zeitspanne oder über eine bestimmte Anzahl von Aktionen zu bewahren.

## Beispiel Qualitätsmerkmale 2/4



Kontrollpunkte	<ul style="list-style-type: none"> <li>- Die Zuverlässigkeit wird in der Risikoanalyse berücksichtigt.</li> <li>- Es wird ein Nutzungsprofil erstellt.</li> <li>- Es wird ein Zuverlässigkeitswachstumsmodell ausgewählt</li> </ul>
Optimierungsvorschläge	
Entwicklungsstufe	4
Abhängigkeiten	Test-Spezifikationstechniken Ebene: A

## Beispiel Qualitätsmerkmale 3/4



<b>Kernbereich: Qualitätsmerkmal Effizienz</b>	
Name	Qualitätsmerkmal Effizienz
Beschreibung	Der Kernbereich Qualitätsmerkmal Effizienz beschreibt die Überprüfung der Effizienz nach ISO9126
Anzahl der Ebenen	1

Ebene A	
Name	Testen des Qualitätsmerkmals Effizienz
Beschreibung	Das Qualitätsmerkmal Effizienz beschreibt das Zeitverhalten unter bestimmten Bedingungen und das Ressourcenverhalten bei der Ausführung des Systems. Dieses Qualitätsmerkmal muss entsprechend den Anforderungen geprüft werden.

## Beispiel Qualitätsmerkmale 4/4



Kontrollpunkte	<ul style="list-style-type: none"> <li>- Es werden Performance-, Last- und Stresstests durchgeführt, welche sich an einem Nutzungsprofil orientieren.</li> <li>- Es werden Tests zur Überprüfung des Ressourcenverhaltens auf jeder Teststufe durchgeführt.</li> <li>- Es werden Maximal-Antwortzeiten des Systems definiert</li> </ul>
Optimierungsvorschläge	
Entwicklungsstufe	9
Abhängigkeiten	Test-Tools Ebene: A Test-Spezifikationstechniken Ebene: A

## Priorisieren von Kernbereichen



Verschieben von Ebenen, um deren Prioritäten zu verändern.

		0	1	2	3	4	5	6	7	8
1	Teststrategie		A			<b>B</b>				
2	Einsatz des Phasenmodells		A			B				
3	Zeitpunkt der Beteiligung			A				B		

Erhöhen der Priorität → Verringern der Entwicklungsstufe  
 Verringern der Priorität → Erhöhen der Entwicklungsstufe



# Beispiel



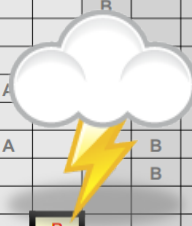
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1 Teststrategie		A				<b>B</b>					C		D	
2 Einsatz des Phasenmodells		A			B									
3 Zeitpunkt der Beteiligung			A				B				C		D	
4 Kostenvoranschlag und Planung				A							B			
5 Test-Spezifikationstechniken		A		B										
6 Statische Testtechniken					A		B							
7 Metriken						A			B			C		D
8 Test-Tools					A			B			C			
9 Testumgebung				A				B						C
10 Testarbeitsplatz				A										
11 Engagement und Motivation		A				B						C		
12 Testfunktionen und Ausbildungen				A			B				C			
13 Reichweite der Methodik					A						B			C
14 Kommunikation			A		B							C		
15 Berichterstattung		A			B		C					D		
16 Dokumentation der Abweichungen		A				B		C						
17 Testware-Management			A			B				C				D
18 Testprozessmanagement		A		B								C		
19 Prüfen							A			B				
20 Low-Level-Tests					A		B		C					
					beherrschbar			effizient			optimiert			

2 Stufen

# Beispiel



	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1 Teststrategie		A			<b>B</b>						C		D	
2 Einsatz des Phasenmodells		A			B									
3 Zeitpunkt der Beteiligung			A				B				C		D	
4 Kostenvoranschlag und Planung				A							B			
5 Test-Spezifikationstechniken		A		B										
6 Statische Testtechniken					A									
7 Metriken						A			B			C		D
8 Test-Tools					A		B				C			
9 Testumgebung				A				B						C
10 Testarbeitsplatz				A										
11 Engagement und Motivation		A				<b>B</b>						C		
12 Testfunktionen und Ausbildungen				A			B				C			
13 Reichweite der Methodik					A						B			C
14 Kommunikation			A		B							C		
15 Berichterstattung		A			B		C					D		
16 Dokumentation der Abweichungen		A				B		C						
17 Testware-Management			A			B				C				D
18 Testprozessmanagement		A		B								C		
19 Prüfen							A			B				
20 Low-Level-Tests					A		B		C					
					beherrschbar			effizient			optimiert			



## Priorisieren von Kernbereichen



Regel notwendig:

Sei Ebene X die zu verschiebende Ebene:  
 Für alle Ebenen Y für die gilt:  
 X ist abhängig von Y  
 muss gelten:  
 Entwicklungsstufe(Y) <= Entwicklungsstufe (X)

## Explizite Abhängigkeiten



	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1 <b>Teststrategie</b>		A					B				C		D	
2 <b>Einsatz des Phasenmodells</b>		A			B									
3 <b>Zeitpunkt der Beteiligung</b>			A				B				C		D	
4 <b>Kostenvoranschlag und Planung</b>				A							B			
5 <b>Test-Spezifikationstechniken</b>		A		B										
6 <b>Statische Testtechniken</b>					A		B							
7 <b>Metriken</b>						A			B			C		D
8 <b>Test-Tools</b>					A			B			C			
9 <b>Testumgebung</b>				A				B						C
10 <b>Testarbeitsplatz</b>				A										
11 <b>Engagement und Motivation</b>		A				B						C		
12 <b>Testfunktionen und Ausbildungen</b>				A			B				C			
13 <b>Reichweite der Methodik</b>					A						B			C
14 <b>Kommunikation</b>			A		B							C		
15 <b>Berichterstattung</b>		A			B		C					D		
16 <b>Dokumentation der Abweichungen</b>		A				B		C						
17 <b>Testware-Management</b>			A			B				C				D
18 <b>Testprozessmanagement</b>		A		B								C		
19 <b>Prüfen</b>							A			B				
20 <b>Low-Level-Tests</b>					A		B		C					

beherrschbar                      effizient                      optimiert



# Implizite Abhängigkeiten



	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1 Teststrategie		A					B				C		D	
2 Einsatz des Phasenmodells		A			B									
3 Zeitpunkt der Beteiligung			A				B				C		D	
4 Kostenvoranschlag und Planung				A							B			
5 Test-Spezifikationstechniken		A		B										
6 Statische Testtechniken					A		B							
7 Metriken					A				B			C		D
8 Test-Tools					A			B			C			
9 Testumgebung				A				B						C
10 Testarbeitsplatz				A										
11 Engagement und Motivation		A				B						C		
12 Testfunktionen und Ausbildungen				A			B				C			
13 Reichweite der Methodik					A						B			C
14 Kommunikation			A		B							C		
15 Berichterstattung		A			B		C					D		
16 Dokumentation der Abweichungen		A				B		C						
17 Testware-Management			A			B				C				D
18 Testprozessmanagement		A		B								C		
19 Prüfen							A			B				
20 Low-Level-Tests					A		B		C					
					beherrschbar			effizient			optimiert			

Yavuz Sancar, Claudia Schumacher – SEE 2010 – 03. Mai 2010

21

# Beispiel



	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1 Teststrategie		A			B						C		D	
2 Einsatz des Phasenmodells		A			B									
3 Zeitpunkt der Beteiligung			A				B				C		D	
4 Kostenvoranschlag und Planung				A							B			
5 Test-Spezifikationstechniken		A		B										
6 Statische Testtechniken					A		B							
7 Metriken					A				B			C		D
8 Test-Tools					A			B			C			
9 Testumgebung				A				B						C
10 Testarbeitsplatz				A										
11 Engagement und Motivation		A			B							C		
12 Testfunktionen und Ausbildungen				A			B				C			
13 Reichweite der Methodik					A						B			C
14 Kommunikation			A		B							C		
15 Berichterstattung		A			B		C					D		
16 Dokumentation der Abweichungen		A				B		C						
17 Testware-Management			A			B				C				D
18 Testprozessmanagement		A		B								C		
19 Prüfen							A			B				
20 Low-Level-Tests					A		B		C					
					beherrschbar			effizient			optimiert			

Yavuz Sancar, Claudia Schumacher – SEE 2010 – 03. Mai 2010

22

## Beispiel: Chemische Vorgänge



Priorisierung entsprechend den Anforderungen an das Projekt

- *Unterbrechungsfreie Kontrollsoftware für chemische Vorgänge*

20	Low-Level-Tests				A		B		C					
21	Qualitätsmerkmal Effizienz									A				
22	Qualitätsmerkmal Zuverlässigkeit				A									
					beherrschbar			effizient			optimiert			

## Beispiel: Spiele



Priorisierung entsprechend den Anforderungen an das Projekt

- *Redering Software von Spielen*

20	Low-Level-Tests				A		B		C					
21	Qualitätsmerkmal Effizienz				A	←								
22	Qualitätsmerkmal Zuverlässigkeit				→					A				
					beherrschbar			effizient			optimiert			

## Zusammenfassung



### Vorteile

- Berücksichtigung von individuellen Anforderungen an einen Testprozess für die Testprozessbewertung und -verbesserung
- Erhaltung und Erweiterung eines bewährten Konzepts

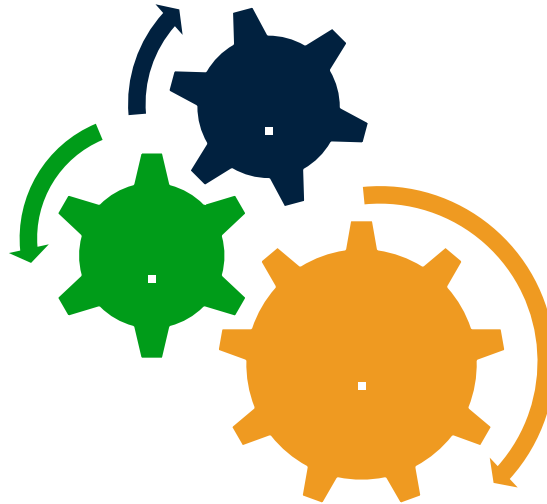
## Zusammenfassung



### Nachteile

- Projektübergreifende Vergleichbarkeit geht verloren.
- Psychologie des Testens: Kernbereiche könnten so gewählt und/oder priorisiert werden, dass das Bewertungsergebnis positiver als in der Realität ausfällt.

## Werkzeug-Demo

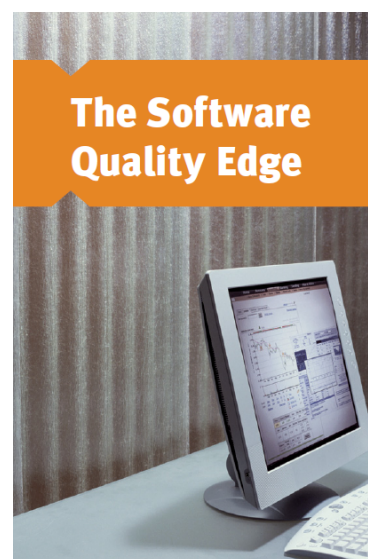


**Vielen Dank für Ihre  
Aufmerksamkeit.**

**Software Quality Lab (s-lab)**

Universität Paderborn  
Warburger Str. 100  
33098 Paderborn  
Tel.: (05251) 60 5390 / 5391

<http://s-lab.upb.de>  
[ysancar@s-lab.upb.de](mailto:ysancar@s-lab.upb.de)



## Beide Zweige des V aus einer Hand

Marco Goeldner, Albert Wilhelm

IBM Software Group Rational Services  
IBM Deutschland GmbH  
Otto-Brenner-Straße 247  
33604 Bielefeld  
marco.goeldner@de.ibm.com  
albert.wilhelm@de.ibm.com

**Abstract:** In vielen Softwareentwicklungsprojekten müssen gegebene Vorgehensmodelle (z.B. V-Modell) und branchenspezifische Standards eingehalten und der Reifegrad des Entwicklungsprozesses (z.B. CMMI, Spice) nachgewiesen werden. Zusätzlich werden die Anforderungen an die zu entwickelnde Systeme immer umfangreicher und komplexer. Die Qualität der auszuliefernden Ergebnisse bemisst sich aber gerade am Nachweis der Erreichung dieser Anforderungen, deshalb ist ihre nahtlose Nachverfolgbarkeit ein wesentlicher Erfolgsfaktor für jedes Projekt – beginnend mit der Anforderungsdefinition auf Stakeholderebene, über das Herunterbrechen auf implementierungsfähige Systemanforderungen, bis hin zum Test ihres Erfüllungsgrads. Häufig liegen die dafür nötigen Informationen in unterschiedlichen Systemen und in verschiedenen Formaten vor, erst eine durchgängige und auswertbare Traceability im Sinne eines angewandten Risikomanagements erlaubt eine angemessene Handhabung der Komplexität.

Der Ansatz des Requirements-driven-Testing ermöglicht eine flexible Abbildung gegebener Vorgehensmodelle, die Steuerung der Einhaltung von Standards und die Unterstützung für den Nachweis bestimmter Reifegrade. Durch eine geeignete Werkzeugunterstützung können Best Practices umgesetzt und eine Integration zwischen den Disziplinen Anforderungs- und Qualitätsmanagement erreicht werden.

Ein präsentiertes Beispiel aus der Praxis macht die Vorteile einer integrierten Lösung aus IBM Rational DOORS und IBM Rational Quality Manager offensichtlich. Der Einsatz von DOORS auf der *Requirements-Management-Seite* des V-Modells ermöglicht eine frühzeitige Beurteilung der Einflüsse geänderter Anforderungen, verhindert unnötige Iterationen und reduziert damit Risiken und die Zeit bis zur Markteinführung. Die übersichtliche und leicht verständliche Darstellung der Compliance der Anforderungen auf Basis der Traceability bietet für die Projektbeteiligten ein zusätzliches Differenzierungsmerkmal gegenüber Mitbewerbern. Auf der *Test-Seite* kann mit IBM Rational Quality Manager kundenspezifisch eine anpassbare Lösung für Testplanung, Workflow-Management, Test-Tracking und die Erstellung von Metriken eingesetzt werden, die sicherstellt, dass alle in DOORS geführten Anforderungen angemessen getestet sind. Anforderungsautoren, Entwickler, Tester und Qualitätsmanager haben dabei kontrollierten Zugriff auf die jeweils erforderlichen Daten. Die praktikable Zuordnung der Anforderungen zu den Projekt- und Geschäftszielen und die Qualifizierung von Projektscheidungen bieten dem Projekt zusätzliche Sicherheit.



# Beide Zweige des V-Modells aus einer Hand.

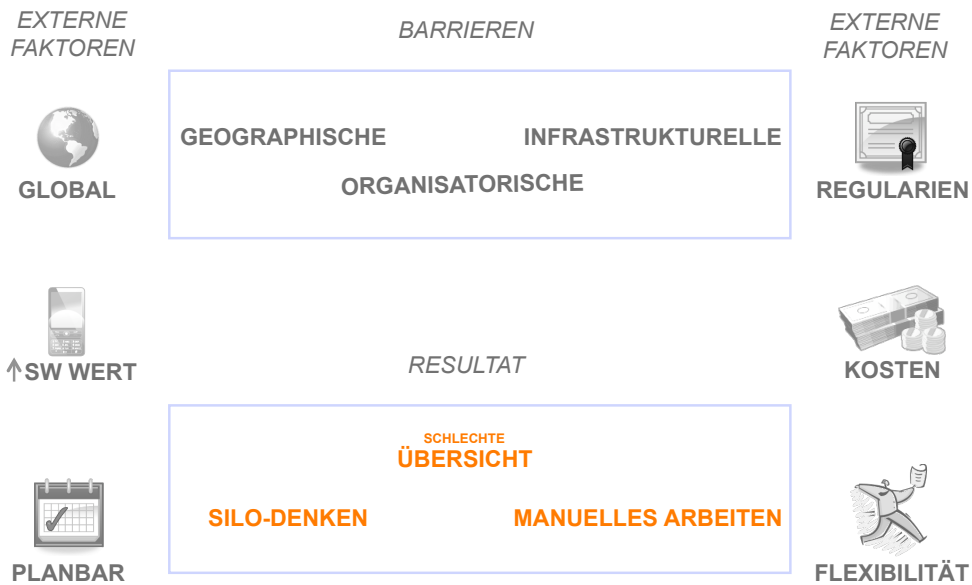


© 2009 IBM Corporation

Beide Zweige des V-Modells aus einer Hand.



## Barrieren und Faktoren im Entwicklungsprozess

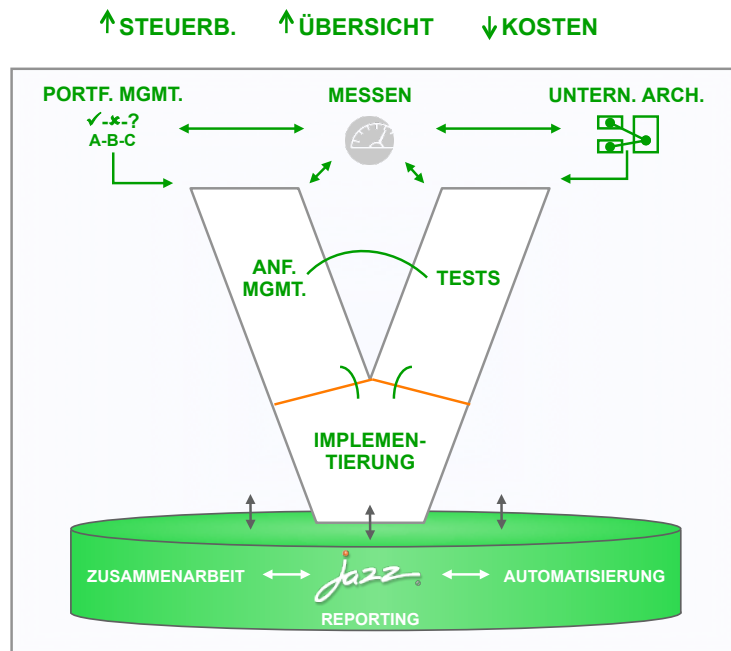


© 2009 IBM Corporation

Beide Zweige des V-Modells aus einer Hand.



## Eine integrierte Plattform für den gesamten Entwicklungszyklus

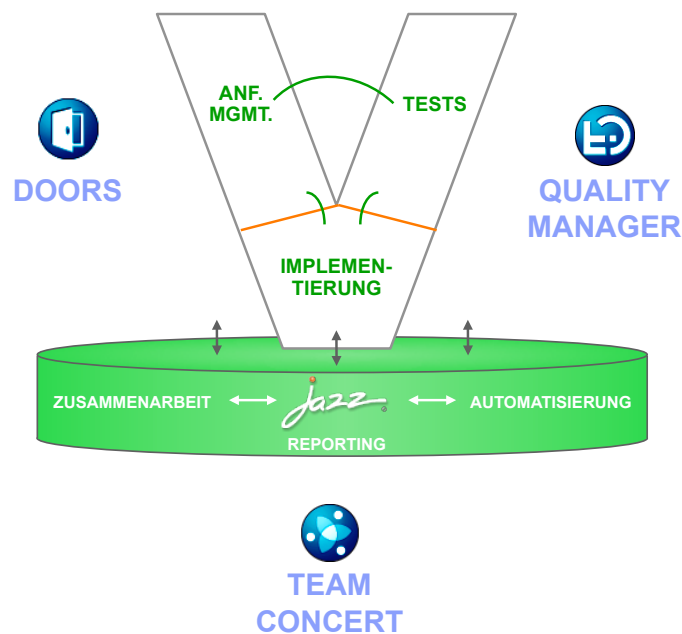


© 2009 IBM Corporation

Beide Zweige des V-Modells aus einer Hand.



## Jazz - Lösung für beide Zweige des V-Modells



© 2009 IBM Corporation

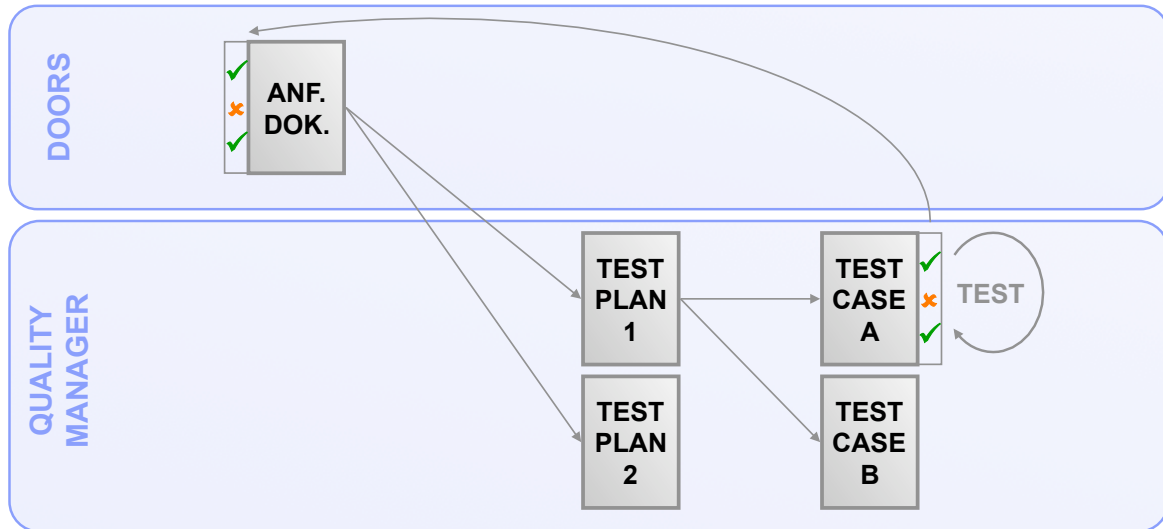


## 6.4 Beide Zweige des V aus einer Hand

Beide Zweige des V-Modells aus einer Hand.



### Workflow zwischen dem Anforderungs- und Testmanagement



© 2009 IBM Corporation

Beide Zweige des V-Modells aus einer Hand.



## Vielen Dank

Für weitere Informationen, besuchen Sie unsere Webseiten

- **IBM Rational Software**  
<http://www-01.ibm.com/software/de/rational/>
- **jazz-Produkte**  
<http://jazz.net/projects/>

© 2009 IBM Corporation

**II.**  
**Tag 2**



## 7. Entwicklung und Einführung von Vorgehensmodellen

### Sessionüberblick

---

- 7.1. Gibt es ein Leben nach dem V-Modell? . . . . . 323
  - 7.2. Entwicklung und Einführung eines standardisierten Softwareentwicklungsprozesses . 347
  - 7.3. Die Entwicklung eines Vorgehensmodells für E-Government-Projekte . . . . . 379
-



## Gibt es ein Leben nach dem V-Modell? Ein pragmatischer Ansatz in einer großen Organisation.

Birgit Geißler

IuK-Controlling und Qualitätsmanagement, Bayerisches Landesamt für Steuern  
Vordere Cramergasse 11  
90478 Nürnberg

Birgit.Geissler@lfst.bayern.de

**Abstract.** Im Bayerischen Landesamt für Steuern (BayLfSt) wurde im letzten Jahr nach einer europaweiten Ausschreibung ein prozessbasiertes Multiprojektmanagement-Tool beschafft, das bereits Standard-Vorgehensmodelle wie V-Modell XT beinhaltet. Für das BayLfSt ist die Akzeptanz der Anwender ein Schlüssel für die erfolgreiche Einführung einer so hochgradig integrierten Lösung. Das bisherige, BayLfSt-eigene Vorgehensmodell war nur für Teilbereiche der Organisation gültig und wurde manchmal nicht ganz konsequent eingehalten. Die neue, toolgestützte Version des Vorgehensmodells wurde deshalb ganz konsequent „verschlankt“. Gleichzeitig wurden die durch Standard-Vorgehensmodelle bisher nicht so gut unterstützten Bereiche (IT-Infrastrukturprojekte und Organisationsprojekte) integriert. Hierzu wurden unterschiedliche Projekttypen abgebildet und ein organisationspezifisches Tailoring hinterlegt, ebenso wie eigene Vorlagen, Prozesse und Meilensteine, die jeweils mit Vorlagen verknüpft sind. Vom ursprünglichen Standard-Vorgehensmodell ist also inhaltlich nur noch wenig erhalten geblieben, lediglich die Methodik wurde genutzt und auf eigene Bedürfnisse angepasst. Ein weiterer Schritt zur Akzeptanz ist die Vermeidung der Doppelerfassung von Informationen: Eine Schnittstelle zur Kosten- und Leistungsrechnung erspart dem Anwender den Wechsel nach SAP.

### 1 Ausgangssituation

Das Bayerische Landesamt für Steuern (BayLfSt) ist eine Mittelbehörde mit ca. 1200 Mitarbeitern und sehr heterogenen Tätigkeitsfeldern. Es beinhaltet sowohl einen IT- als auch einen Fachbereich. Im IT-Bereich wird die Software für alle bayerischen Finanzämter und für Länderverbünde und länderübergreifende Projekte und Verfahren (z. B. ELSTER) erstellt. Daneben beinhaltet das BayLfSt ein Rechenzentrum, welches Hardware und technische Infrastruktur für die Finanzverwaltung und andere Ressorts zur Verfügung stellt.

Durch die zunehmende Größe und Bedeutung des IT-Bereichs nimmt auch die Anzahl der Projekte immer mehr zu. Um diese besser und erfolgreicher durchführen und steuern zu können, beschloss das BayLfSt, ein Multiprojektmanagement-Tool zu beschaffen.

## 7.1 Gibt es ein Leben nach dem V-Modell?

Dieses Tool sollte von allen Projekten im Bereich des Landesamtes eingesetzt werden, unabhängig vom konkreten Tätigkeitsfeld. Man erhoffte sich dadurch mehr Einheitlichkeit und Standards in den Projekten und wollte gleichzeitig auch das Problem der unterschiedlichen Ablageorte für Projektinformationen in den Griff bekommen. Die Beschaffung erfolgte nach Durchführung einer europaweiten Ausschreibung Ende letzten Jahres. Der Zuschlag wurde dem Tool in-Step von der Fa. microTOOL erteilt.

## 2 Einführung des Tools

Im Landesamt entwickelte man Software schon seit Jahren nach einem Vorgehensmodell. Dieses Vorgehensmodell basierte auf dem V-Modell 97, war jedoch in einigen Teilbereichen konkretisiert worden. Um die Projektleiter hierbei optimal zu unterstützen, hat man ein Tool beschafft, in dem Vorgehensmodelle bereits standardmäßig integriert sind.

### 2.1 Entrümpelung des bisherigen Vorgehensmodells

Das bisherige Vorgehensmodell hatte jedoch zwei Nachteile: zum einen war es sehr idealistisch, so dass in der Realität nicht immer alle Anforderungen konsequent umgesetzt werden konnten. Zum anderen konzentrierte es sich ausschließlich auf die Softwareentwicklung, so dass sich viele Bereiche der Organisation (wie z. B. das Rechenzentrum oder der Fachbereich) darin nicht wiederfanden und daher im Laufe der Zeit eigene Prozesse und Vorlagen entwickelt haben.

Diese Probleme löste man mit einem sehr pragmatischen Ansatz: alle Teile, die bisher nicht gelebt wurden, wurden rigoros entfernt.

### 2.2 Erweiterung des bisherigen Vorgehensmodells zum „VM BayLfSt“

Im Gegenzug wurden Bereiche, die im bisherigen Vorgehensmodell keine Berücksichtigung fanden, im Vorgehensmodell aufgenommen. Ziel sollte sein, dass sich jeder Arbeitsbereich im Vorgehensmodell wiederfindet, nicht nur die Softwareentwicklung, sondern zum Beispiel auch Organisationsprojekte.

### 2.3 Anpassung des Prozessmodells in in-Step

Das neue Vorgehensmodell musste auch entsprechend im Tool integriert werden. Hierbei wurde zunächst ein Großteil des enthaltenen V-Modell XT-Inhalts entfernt, danach wurden eigene Vorlagen und Prozesse integriert. Die Möglichkeit des Tailorings wird als Technik genutzt, inhaltlich jedoch komplett mit eigenen Fragen und Antworten ausgestattet. Um alle Bedürfnisse abdecken zu können, wurden drei verschiedene Projekttypen abgebildet, die ggf. unterschiedliche Vorlagen und Meilensteine enthalten. Hierbei hat man versucht, auf einen allzu strengen Workflow zu verzichten. Man wollte den Projektleiter in seiner Freiheit möglichst wenig einschränken.



### **2.4 Verknüpfung von Vorlagen mit in-Step**

Es wurde intensiv von der Möglichkeit Gebrauch gemacht, Projektinformationen automatisiert in die entsprechenden Vorlagen zu generieren. Dazu gehören neben den Standard-Projektinformationen wie Projektname, Projektleiter, Änderungshistorie usw. auch ganze Reports (z.B. aktueller Projektplan, Status des Projekts, aktuelle Risikoliste, Gegenüberstellung von Plan- und Ist-Zeiten usw.). Die Dokumente enthalten damit automatisch immer die zum Bearbeitungszeitpunkt aktuellen Projektinformationen, ohne dass sie vom Projektleiter hineinkopiert werden müssen.

### **2.5 Eigenes Schulungskonzept**

Um eine möglichst hohe Akzeptanz zu erreichen, wurde ein eigenes Schulungskonzept entwickelt, das nicht nur die Bedienung des Tools, sondern auch Projektmanagementtheorie schult. Hierzu gehört neben den allgemeinen Projektmanagement-Grundlagen auch die Vorstellung des organisationsspezifischen Qualitätsmanagementsystems.

### **2.6 Pilotierungsprojekte**

Nach dem Customizing eines ersten unternehmenseigenen Prozesses wurde mit einigen Pilotprojekten gestartet. Basierend auf deren Erfahrungen und Anregungen hat sich der Prozess weiter verändert und angepasst. Das Einführungsteam und die Pilotprojektleiter coachen sich damit gegenseitig.

### **2.7 Schnittstelle zur Kosten- und Leistungsrechnung (SAP)**

Um möglichst wenige Informationen doppelt (in verschiedenen Tools) erfassen zu müssen, wurde eine Schnittstelle zur Kosten- und Leistungsrechnung erstellt. Eine Erfassung der Arbeitszeiten in SAP ist daher nicht mehr nötig, sofern der Bearbeiter die Zeiten bereits in in-Step eingepflegt hat.

### **2.8 Multiprojektmanagement**

Das Tool ermöglicht, dass man sich jederzeit aktuell über alle Projekte informieren kann, hierzu gehören neben bestimmten Projektinformationen auch veröffentlichte Projektdokumente. Für das Management können außerdem zusätzliche Managementberichte erstellt werden, die gezielt auf Probleme oder Handlungsbedarf aufmerksam machen.

### **2.9 Projektübergreifendes Ressourcenmanagement**

Durch die Verwaltung aller Mitarbeiter und deren Linienaufgaben ist ein übergreifendes Ressourcenmanagement möglich geworden. So erkennt man auf den ersten Blick, welche Bereiche bereits ausgelastet sind und wo daher Verzögerungen zu erwarten sind.

### **3 Bewertung und Ausblick**

Der Verzicht auf zu viel Formalismus und strenge Prozesse hat sich bewährt: Projektleiter nutzen das Tool, weil sie selbst einen Nutzen davon haben. Die Akzeptanz der von den Anwendern als Einheit wahrgenommenen Kombination aus Prozess und Tool führte außerdem zu einem schnellen Nutzen für die Organisation.

Trotzdem bewirkt auch ein Tool nicht, dass sich alle vorherigen strukturellen Probleme in Luft auflösen. Mit dem Werkzeug werden jedoch Probleme und kritische Projekte schneller sichtbar. Wichtig ist, dass das Management die Nutzung des Tools offensiv unterstützt und einfordert, aber auch bei offensichtlichen Missständen entsprechend reagiert.

Inzwischen planen alle neuen Projekte mit in-Step, eine Ausdehnung auf Linienaufgaben erfolgt im Jahr 2010.



## Gibt es ein Leben nach dem Vorgehensmodell?



**Warum** ein Tool?  
**Welches** Tool?  
**Wozu** ein Vorgehensmodell?  
**Warum** ein eigenes Vorgehensmodell?  
**Wie** war die Einführung?



**Wer sind wir?**



BayLfSt



BayLfSt

Ein Tool bietet:



**Bestmögliche  
Unterstützung**

BayLfSt

Ein Tool ermöglicht:



**Multi-  
Projektmanagement**

BayLfSt



**bisher:  
Projektmanagement  
„zu Fuß“**

BayLfSt



Ein Tool fördert:



BayLfSt

## Beschaffung des Tools



# Welches Tool?

BayLfSt



## Abteilungsübergreifende Abstimmung

BayLfSt



## Wunschliste



BayLfSt

## Welches Tool?

**„Man kann nicht alles haben,  
den **Rahm**  
und die **Butter**“**

BayLfSt



Welches Tool?



**Ranking**

**KO-  
Kriterien**



BayLfSt

Umfangreiche Leistungsbeschreibung



BayLfSt

## Finale



BayLfSt

## Teststellung



**Tarnen  
und  
Täuschen?**



BayLfSt



BayLfSt

Welches Tool?

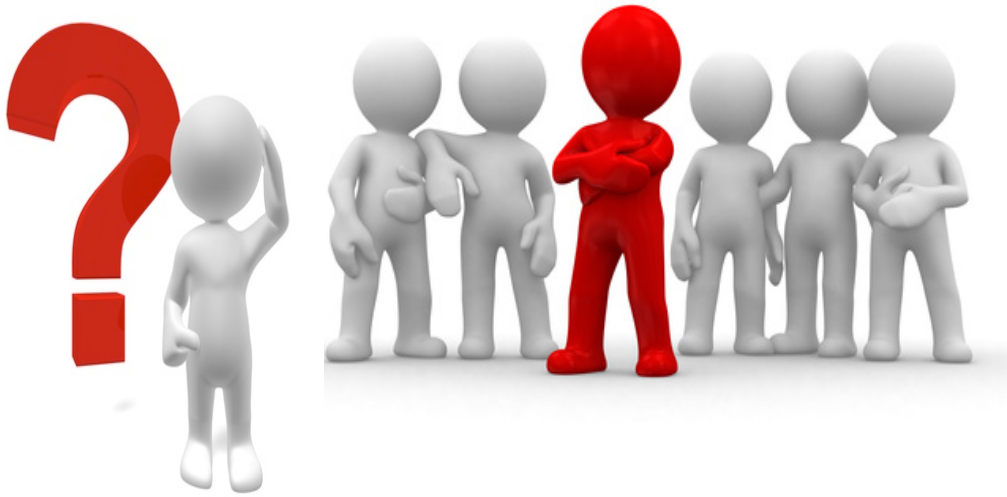


**... ist auch  
„vergaberechtlicher“  
Gewinner:**

**in-Step**

BayLfSt

Warum in-Step?



BayLfSt

Warum in-Step?

**Integriertes  
Vorgehensmodell**



BayLfSt

## Warum in-Step?



**Vorgehensmodell  
an eigene  
Prozesse  
anpassbar**

BayLfSt



**Warum ein  
Vorgehensmodell?**

BayLfSt



# Maß- geschneiderte Projekte



BayLfSt



## Ergebnis- Orientierung

BayLfSt





## Einheitliche Produkte

BayLfSt



BayLfSt



## „Entrümpelung“ des bisherigen Vorgehensmodells

BayLfSt



## Erweiterung zum „VM BayLfSt“

BayLfSt





## alle Projekttypen abgebildet



BayLfSt

Wie eingeführt?



**„Mache es so einfach wie möglich -  
aber nicht einfacher!“**

Albert Einstein

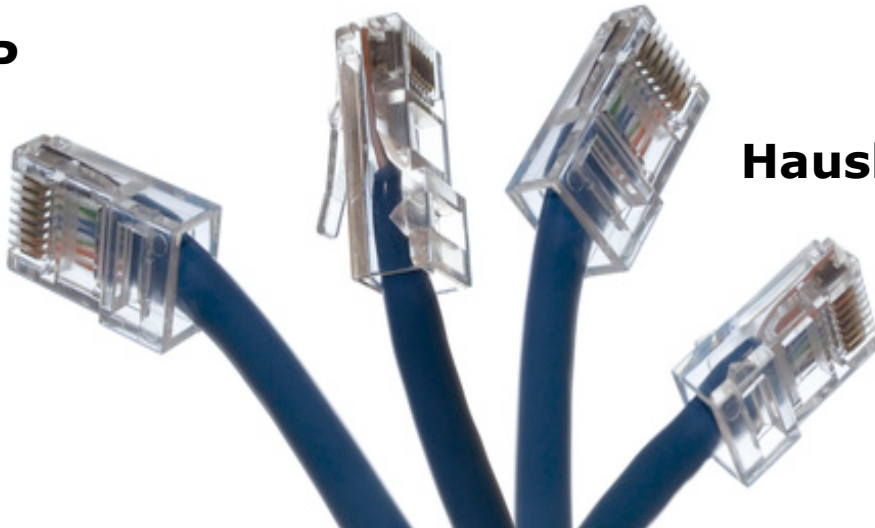
BayLfSt

## Keine Doppeleingaben



### KONSENS

**SAP**



**Haushalt**

BayLfSt

**Einbeziehung  
Personalrat**



BayLfSt



BayLfSt



## **Tool-übergreifende Schulungen**

BayLfSt



BayLfSt

Wie geht's weiter?



**„Nichts ist beständiger als der  
Wandel“**

BayLfSt

Wie geht's weiter?



**„Der Schlüssel  
zum Erfolg sind  
nicht  
Informationen.**

**Das sind  
Menschen.“**

Lee Iacocca

BayLfSt



BayLfSt

## 7.1 Gibt es ein Leben nach dem V-Modell?

# **Entwicklung und Einführung eines standardisierten Softwareentwicklungsprozesses – ein Erfahrungsbericht**

Dr. Stefan Walburg

IT-Systemhaus der Bundesagentur für Arbeit  
SEA – Systementwicklung Anwendungsarchitektur, Regensburger Straße 104  
90478 Nürnberg  
stefan.walburg@arbeitsagentur.de

## **1 Motivation**

In großen IT-Häusern weisen die praktizierten Softwareentwicklungsprozesse häufig eine große Heterogenität hinsichtlich Vorgehen, Methodik und Werkzeugen auf. Ursachen hierfür sind historisch gewachsene Systeme, methodische und tooltechnische Fortentwicklungen, mangelnde bzw. unscharfe interne Vorgaben und – insbesondere bei IT-Projekten im öffentlichen Dienst – übernommene Vorgehensweisen externer Dienstleister. Diese Heterogenität führt jedoch zu hohen Entwicklungs- und Instandhaltungskosten sowie zu unerwünschten langfristigen Abhängigkeiten.

Im IT-Systemhaus der Bundesagentur für Arbeit (BA) wurde daher ein standardisierter Softwareentwicklungsprozess konzipiert, auf dessen Grundlage neue IT-Projekte realisiert bzw. laufende Verfahren selektiv migriert werden. Nachdem im IT-Systemhaus der BA täglich 2.000 Mitarbeiterinnen und Mitarbeiter an der Entwicklung und dem Betrieb von Software mitwirken, müssen Konzeption und Einführung des Standards systematisch erfolgen, um eine breite Akzeptanz zu erlangen und somit die Vorteile der Standardisierung auch tatsächlich ausschöpfen zu können.

## **2 Ansatz**

Der Vortrag gliedert sich in zwei Teile. Im ersten Teil wird das methodische Vorgehen zur Entwicklung des Standards skizziert. Dieses basiert auf einem hybriden Top-Down und Bottom-Up-Ansatz, der strategische (Management-)Vorgaben mit etablierten Standards und empirischen Projekterfahrungen kombiniert und daraus einen tragfähigen und anerkannten Standard ableitet. Dieser wird vor dem Rollout über interne Quality Gates abgesichert und über einen definierten Aktualisierungsprozess kontinuierlich optimiert sowie an geänderte Rahmenbedingungen adaptiert.

Die Definition eines Standards ist jedoch nur eine Seite der Medaille. Diesen in eine komplexe und historisch gewachsene Organisation einzuführen, stellt eine mindestens ebenso große Herausforderung dar. Nur selten werden "Standardbringer" wie Heilsbrin-



ger empfangen. Daher muss die Einführung dieser Veränderungen sehr weitsichtig und umfassend geplant und umgesetzt werden.

Mit dieser Aufgabenstellung beschäftigt sich der zweite Teil des Vortrags. Darin werden die erfolgskritischen Handlungsfelder einer systematischen Einführungsstrategie aufgezeigt und auf der Grundlage der bisherigen Erfahrungen bewertet. Konkret adressieren die einzelnen Handlungsfelder folgende Fragen:

- *Umfeldanalyse*: Welche Partner sind in welcher Rolle in welcher Form in die Entwicklung respektive Einführung des Standards miteinzubeziehen?
- *Marketing*: Welche Informationen sind in welcher Form wann an welche Zielgruppe zu kommunizieren?
- *Qualifikation*: Welche Rollen sind in welcher Intensität über welche Schulungsmodule zu qualifizieren?
- *Leitfaden*: Wie werden die Inhalte des Standards den Betroffenen aktuell, zielgruppen- und praxisorientiert bereitgestellt?
- *Coaching*: Welche Coaches werden in welcher Quantität und Qualität gefordert und gefördert?
- *Pilotierung*: Wie erfolgt die Auswahl der Pilotprojekte? Wann und wie erfolgt die Pilotierung?
- *Rollout*: Wann und wie erfolgt eine systematische und flächendeckende Einführung des Standards in die Organisation?
- *Kennzahlen*: Welche Metriken eignen sich, um qualitative und quantitative Aussagen über den Standard zu treffen?

### 3 Bewertung

Die Heterogenität und Komplexität der Softwareentwicklungsprojekte im IT-Systemhaus der Bundesagentur für Arbeit lässt einen „One-Size-Fits-All“-Ansatz für die Systementwicklung nicht zu. Der standardisierte Softwareentwicklungsprozess folgt daher dem Baukastenprinzip und bietet einen Pool von standardisierten Modulen, die in Abhängigkeit von den gegebenen Projekteigenschaften kombiniert werden (Tailoring). Durch dieses Vorgehen wird Flexibilität und Akzeptanz erhöht ohne die Vorteile der Standardisierung einzubüßen.

Bei der Entwicklung des Standards hat sich der „Meet-in-the-Middle“-Ansatz bewährt, obgleich sich die unterschiedlichen Ziele der Beteiligten nicht immer friktionsfrei ineinander überführen lassen.

Bei der Einführung des Standards muss das Spannungsfeld zwischen visionären Management-Zielvorgaben und operativen Projekt-Constraints überbrückt werden. Eine systematische Einführungsstrategie mit den genannten Handlungsfeldern kann helfen, den Standard zu einem unternehmensweit akzeptierten Werkzeug dauerhaft zu etablieren.

Dr. Stefan Walburg – Entwicklung und Einführung eines standardisierten Softwareentwicklungsprozesses - 4. Mai 2010



Entwicklung und Einführung eines  
standardisierten Softwareentwicklungs-  
prozesses – ein Erfahrungsbericht



## Agenda

---

- Informationstechnik der Bundesagentur für Arbeit
- Softwareentwicklung im IT-Systemhaus
- Vorgehen zur Entwicklung des Standards
- Handlungsfelder der Einführungsstrategie



## Informationstechnik der Bundesagentur für Arbeit

Kurzprofil

Standorte

IT-Kennzahlen



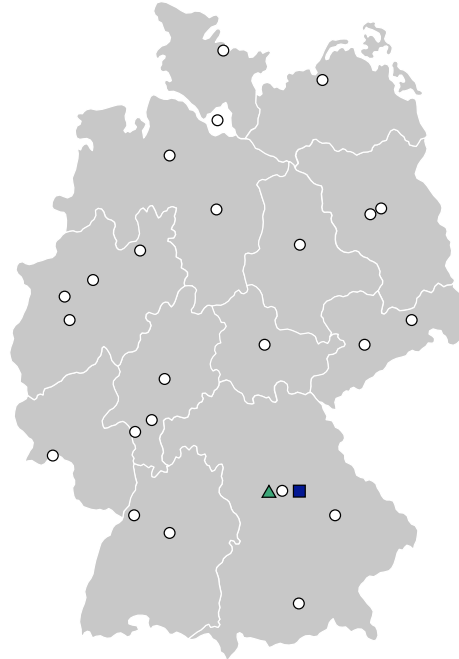
## Kurzprofil – BA-Informationstechnik

- **Hauptsitz:** Nürnberg
- **CIO:** Klaus Vitt
- **IT-Mitarbeiter/innen:** 2.000
- **Vernetzte PC:** 170.000
- **Server:** 11.500

## Standorte

---

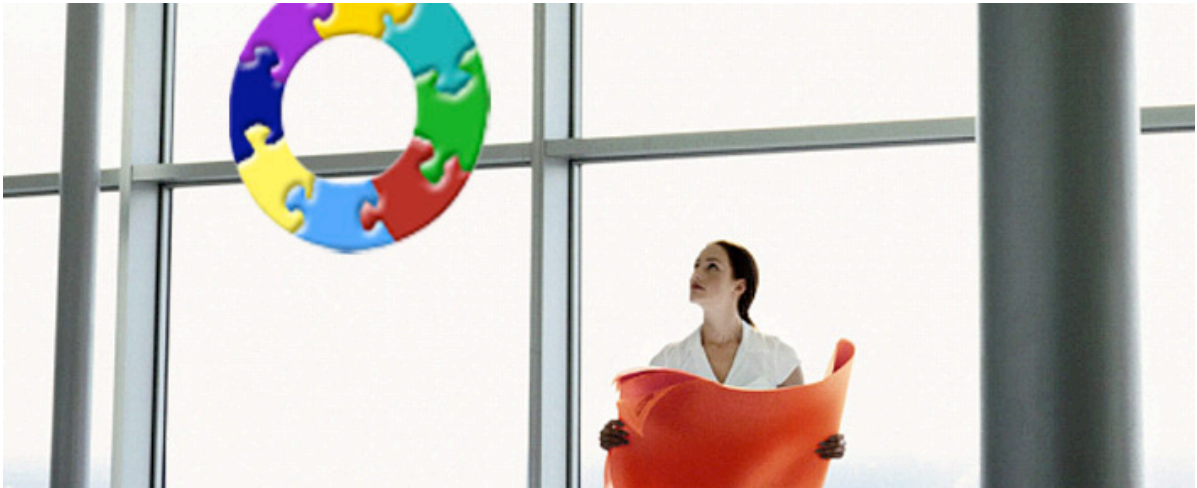
- ▲ IT-Steuerung - Zentrale der BA
- IT-Systemhaus
- Regionaler IT-Service  
24 Standorte



## Output – IT-Kennzahlen

---

- E-Mail-Volumen
  - Extern: 10 Mio. E-Mails monatlich
  - Intern: 22 Mio. E-Mails monatlich
- Überweisungen
  - 14 Mio. Überweisungen monatlich
- Postsendungen
  - 7,5 Mio. Sendungen monatlich
- Druckseiten
  - 38 Mio. Seiten monatlich



## **Softwareentwicklung im IT-Systemhaus**

---

Status Quo

Ziele der Standardisierung

Der SWEP im Überblick



## **Softwareentwicklung im IT-Systemhaus – Status Quo**

---

### **Ausgangssituation**

- Große Heterogenität hinsichtlich Vorgehen, Methodik und Tools

### **Ursachen**

- Historisch gewachsene Systeme und Prozesse
- Kontinuierliche methodische und tooltechnische Weiterentwicklungen
- Fehlende bzw. unscharfe interne Vorgaben und Richtlinien
- Übernommene Vorgehensweisen externer Dienstleister

### **Folgen**

- Hohe Komplexität
- Hohe Ressourcenbindung
- Hohe Abhängigkeiten zu Dritten
- Breites Qualifikationsspektrum

## Sachziele des standardisierten Softwareentwicklungsprozesses (SWEP)

---

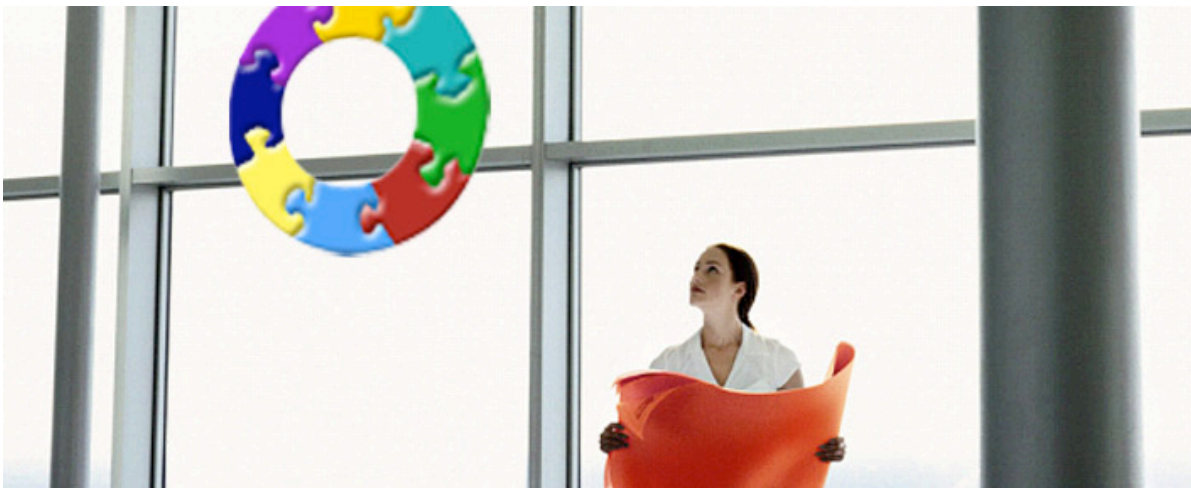
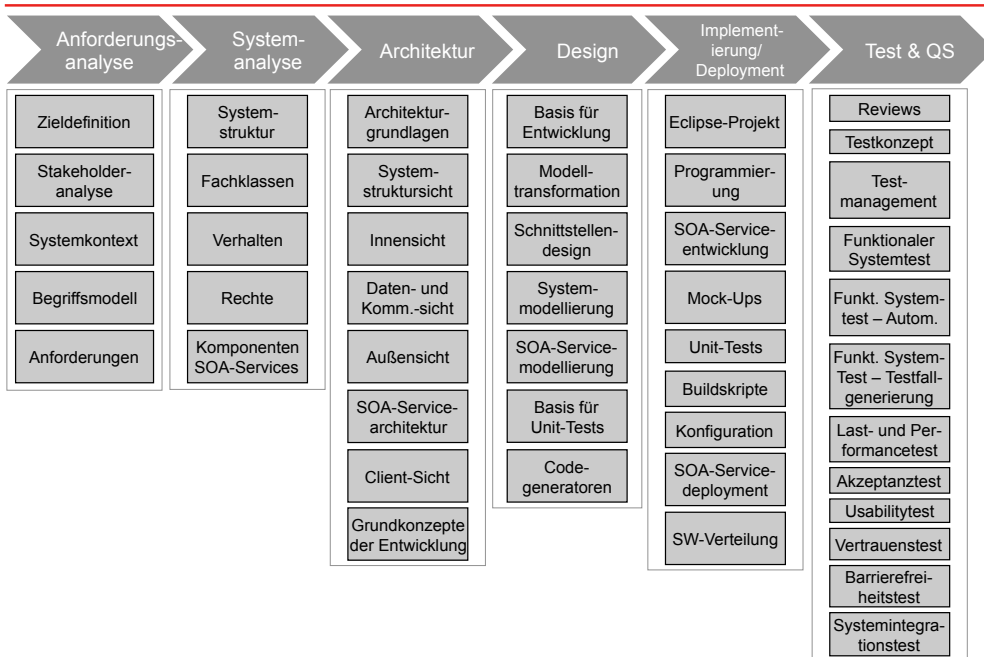
- **Standardisierung** des Softwareentwicklungsprozesses hinsichtlich
  - Vorgehen,
  - Methodik und
  - Werkzeuge
- **Transparente und rollenorientierte Qualifizierungspfade**
  - SWEP als Qualifizierungsraaster
  - Minimierung der Fluktuationsfraktionen (intern/extern)
- **Ausschreibungsvorgabe** bei externer Projektvergabe
- **Einführungsstrategien** zur systematischen Integration des SWEP in die Organisation

## Formalziele des SWEP

---

- Qualitätssteigerung von Produkten und Prozessen
- Wirtschaftlichkeit
- Akzeptanz
- Ganzheitliche methodische Durchgängigkeit
- Homogenität
- Einfachheit
- Wartbarkeit
- Planbarkeit (→ u.a. Ressourcen)

## SWEP-Disziplinen und -Module



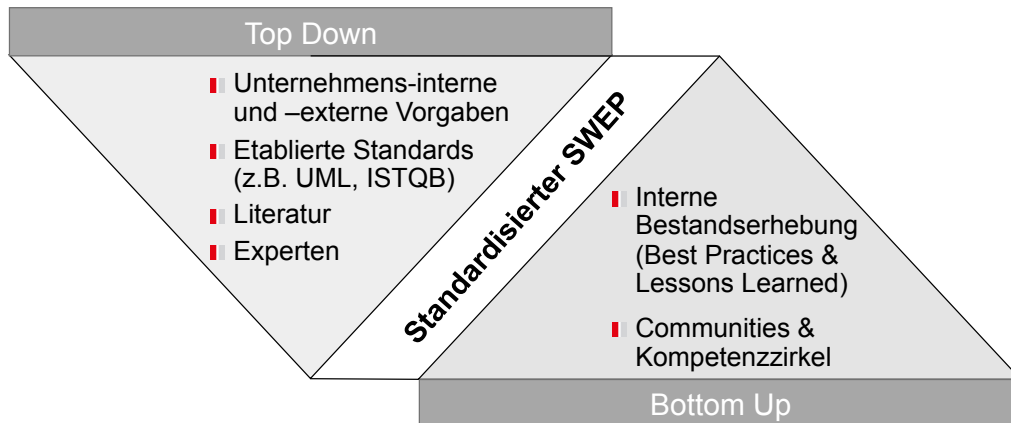
## Vorgehen zur Entwicklung des Standards

Top Down

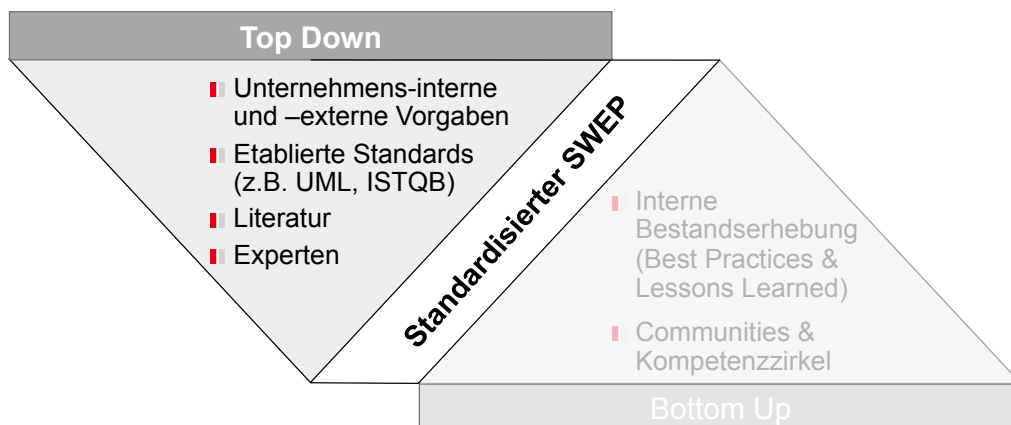
Bottom Up



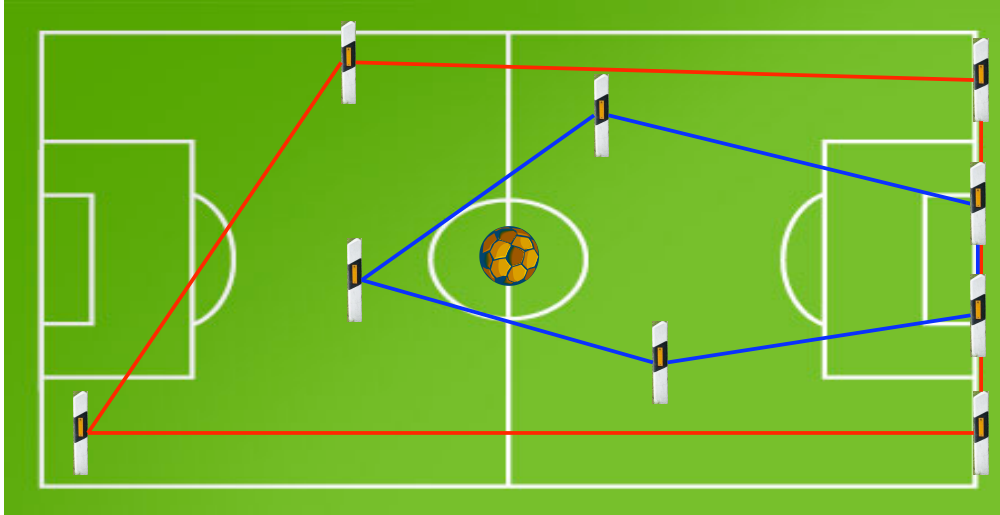
## Hybrider Top-Down und Bottom-Up-Ansatz





## Top-Down-Views

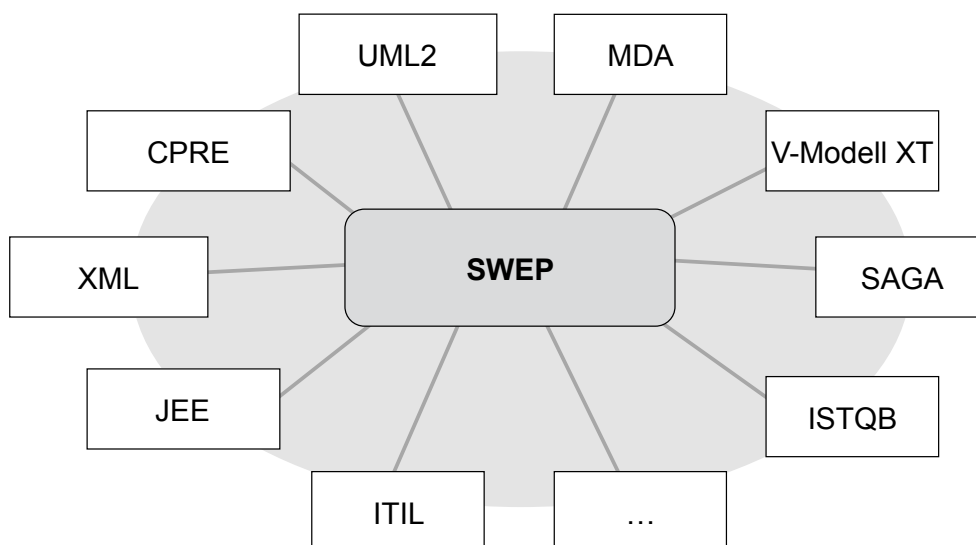


## Das Spielfeld des SWEP ist eingeschränkt

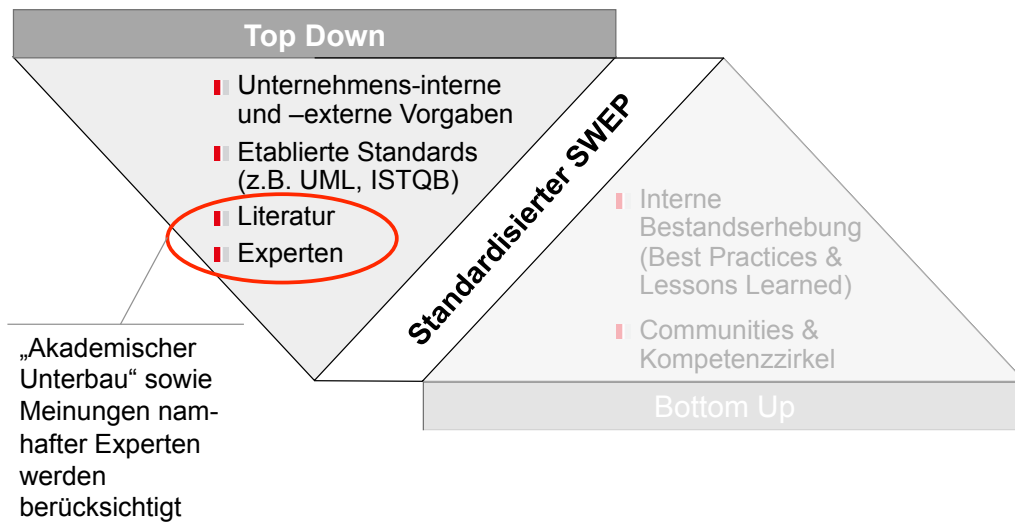


-  : Unternehmens-externe Constraints (z.B. Branchenstandards, Technologieentwicklung etc.)
-  : Unternehmens-interne Constraints (z.B. IT-Strategie, Lizenzen, Wertschöpfungstiefe etc.)

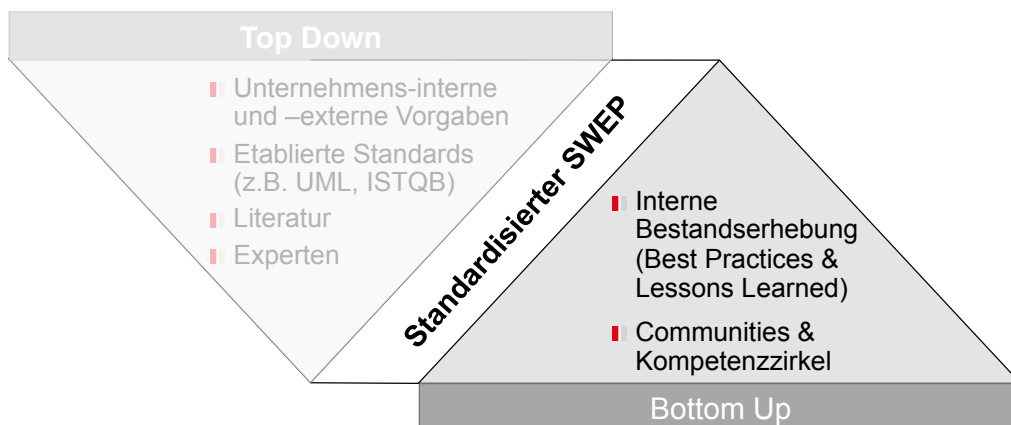
## Etablierte Standards im Kontext des SWEP



## Literatur und Experten



## Bottom-Up-Views

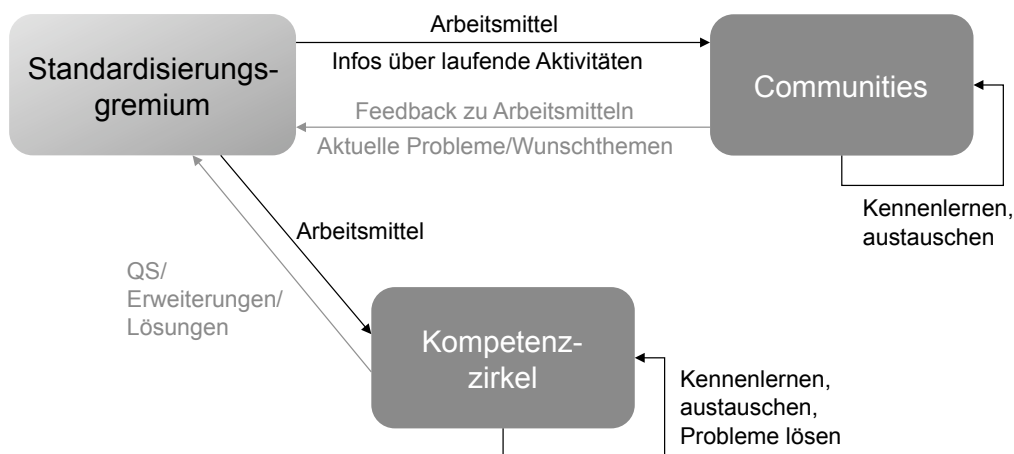


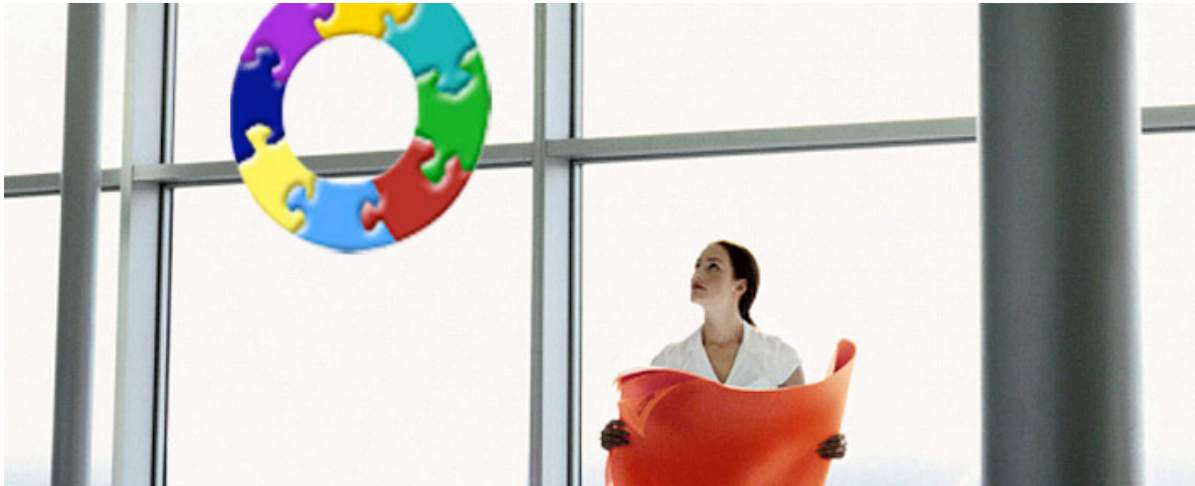
## Warum in die Ferne schweifen... Interne Bestandserhebung

- Systematische Befragung repräsentativer Inhouse-Projekte
- Kernfrage: Was hat sich im Praxis(!)-Alltag bewährt, was weniger?
- Identifikation von Best Practices und Lessons Learned
- Herausforderungen:
  - Mengengerüst (über 150 Verfahren im Haus)
  - Ergebniskonsolidierung (große Heterogenität)
  - Detaillierungstiefe (adäquate Aufwand-Nutzen-Relation)
  - Generalisierungsschritt proprietärer Lösungen (viele ist nicht generalisierbar)
  - Erwartungsdruck („wieso wurde nicht unsere Lösung in den Standard überführt?“)

## „Partizipativ-kybernetische“ Entwicklung von Standards: Communities und Kompetenzzirkel

- Feedback-Loops
  - mit allen Beteiligten, gruppiert nach Disziplin: **Communities**
  - mit den Experten zu einzelnen Themen: **Kompetenzzirkel**

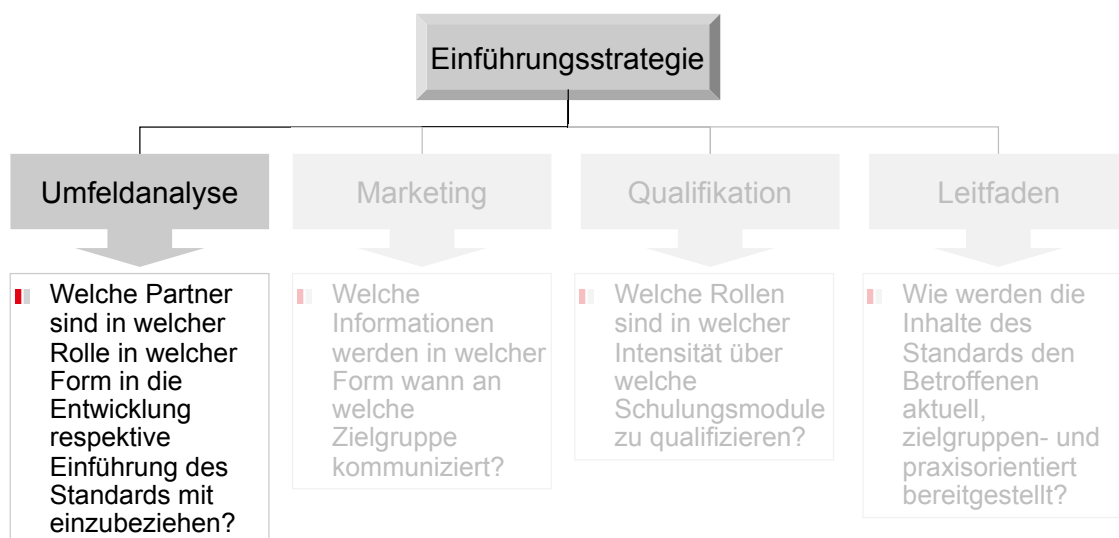




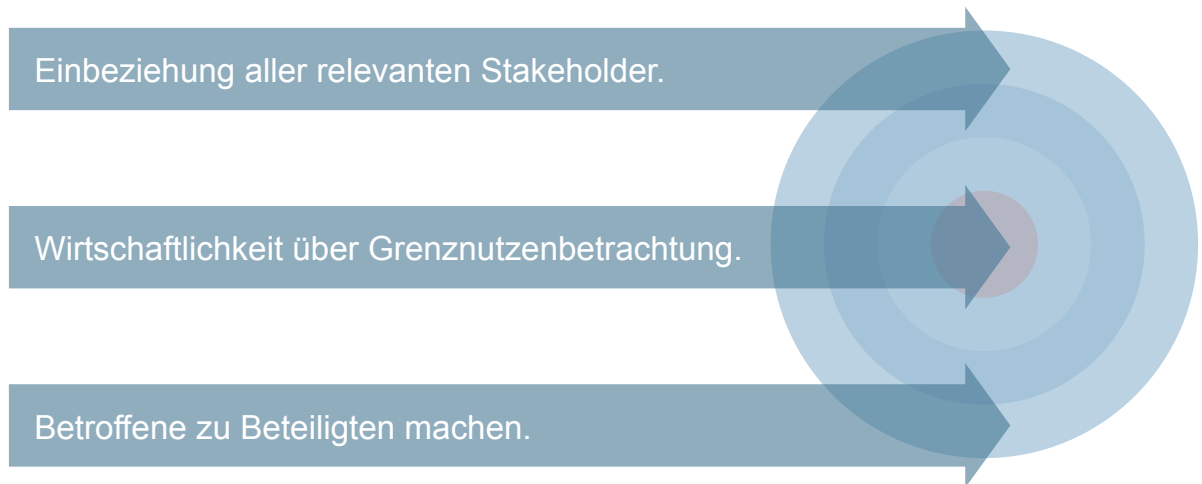
## Handlungsfelder der Einführungsstrategie

Umfeldanalyse	Coaching
Marketing	Pilotierung
Qualifikation	Rollout
Leitfaden	Kennzahlen

## Handlungsfeld <Umfeldanalyse>

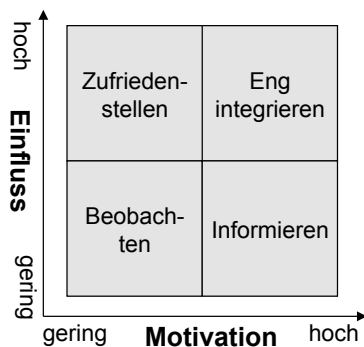


## Ziele der Umfeldanalyse



## Aufgaben innerhalb einer Umfeldanalyse

- Stakeholder-Analyse (vgl. [Rupp09])

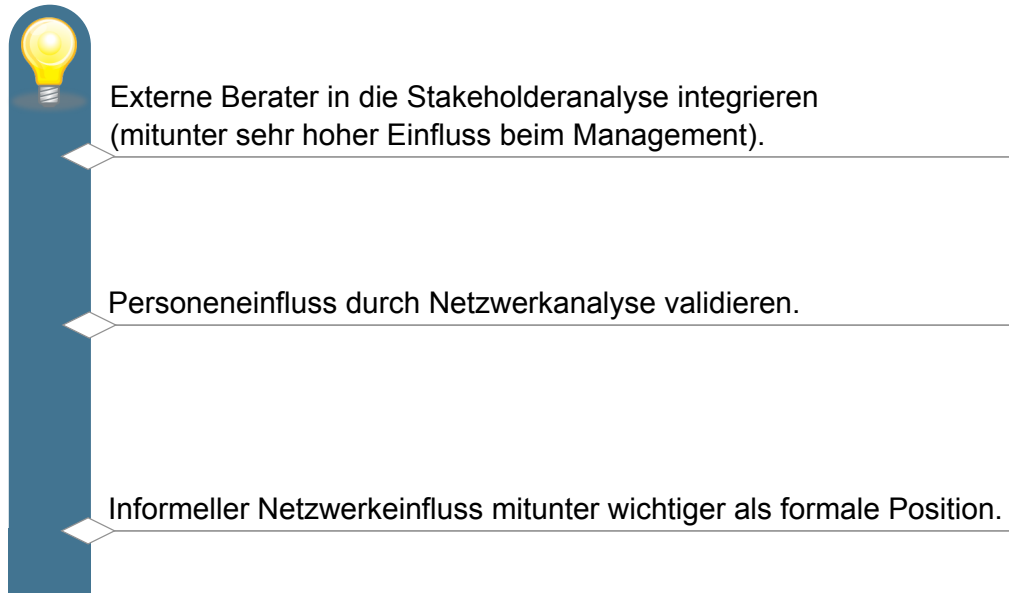


→ Klassische RE-Disziplin

- Stakeholder-Dokumentation (einheitlich, kontinuierlich)
- Stakeholder-Pflege (Stakeholder-Relationship-Management)

→ Enge Verzahnung mit allen weiteren Handlungsfeldern

## Best Practices & Lessons Learned

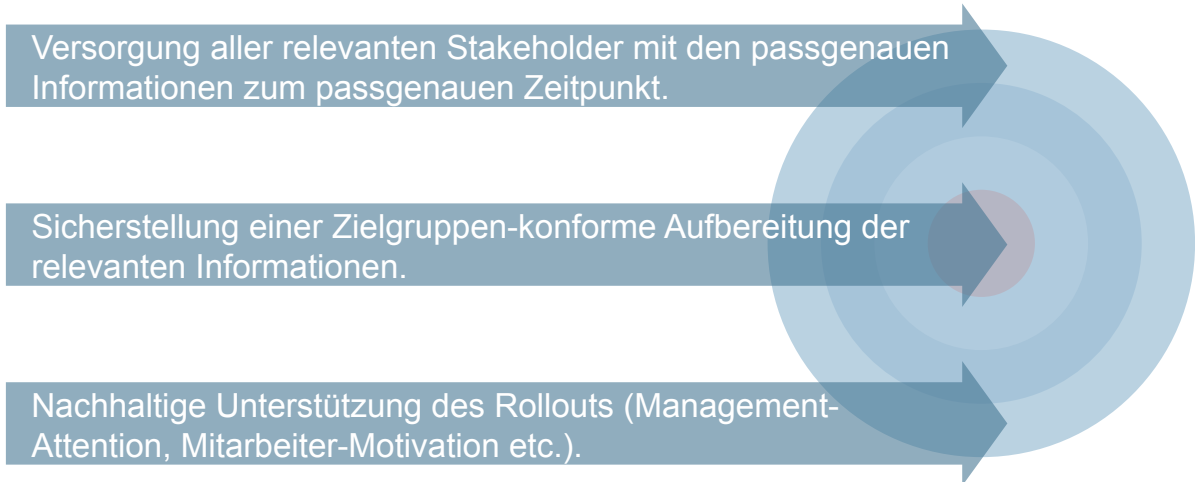


## Handlungsfeld <Marketing>





## Ziele des Marketingkonzepts

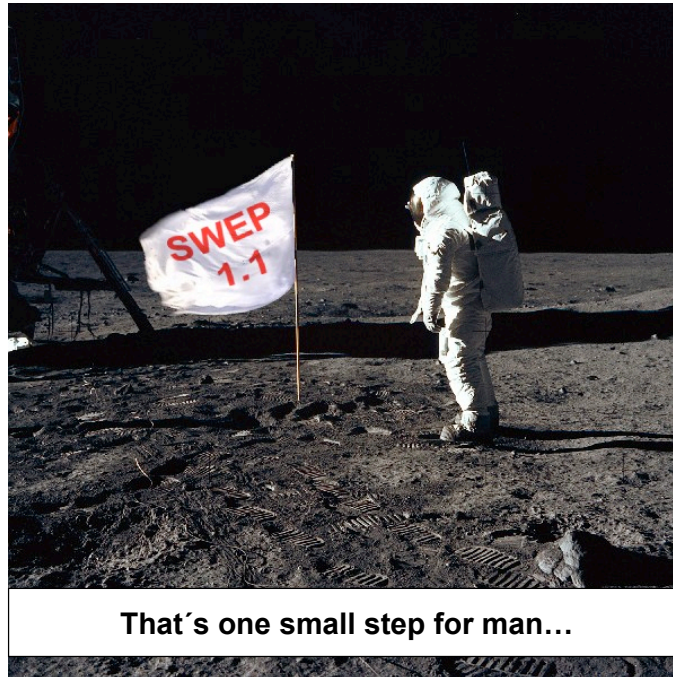


## Exemplarischer Ergebnistyp des Marketingkonzepts: Marketingm(atr)ix

Zielgruppe \ Kanal	Erforderlicher Informationsgrad					
	Interne BA-Mitarbeiter (Ausführungsebene)	Externe BA-Dienstleister	Azubis	Mittleres Management	Top Management	BA-Mitarbeiter außerhalb IT
Sachstandsbericht	○	○	○	◐	●	○
Teammeeting	◐	○	○	◐	○	○
Kompetenzwoche	◐	○	○	◐	○	○
Kompetenz Jour-Fixe	◐	○	○	●	○	○
SEK-Kompetenztag	●	○	○	◐	◐	○
Kompetenzzirkel	●	○	○	○	○	○
Wissensbörse	◐	◐	◐	◐	◐	○
Standardqualifizierung	●	○	○	○	○	○
Wissensportal	●	●	◐	◐	◐	○
Intranet	◐	◐	◐	◐	◐	●
SWEP-Newsletter	◐	◐	◐	◐	◐	○
IT-Newsletter	◐	◐	◐	◐	◐	◐
Poster/Flyer	◐	◐	◐	◐	◐	◐
Infomesse	●	◐	◐	◐	○	○
Ausbildungshandbuch	○	○	●	○	○	○
Juniorenfirma-Spielprojekt	○	○	●	○	○	○
Dialog (Zeitschrift)	◐	◐	◐	◐	◐	●

Gemäß dem erforderlichen Informationsgrad ● gut geeignet ○ ungeeignet

## Beispielplakat mit unerwartetem Interpretations- spielraum



Dr. Stefan Walburg – SEE 2010

Seite 29

## Best Practices & Lessons Learned



Marketing kostet Geld...

Marketing ist eine hohe Kunst → Experten einbeziehen!

Projekt-spezifisches Marketing muss konform zum Unternehmens-spezifischen Marketing sein (keine Alleingänge).

Zielgruppen-spezifisches Marketing ist unerlässlich: Programmierer haben beispielsweise andere „Reiz-Reaktions-Muster“ als Requirements Engineers.

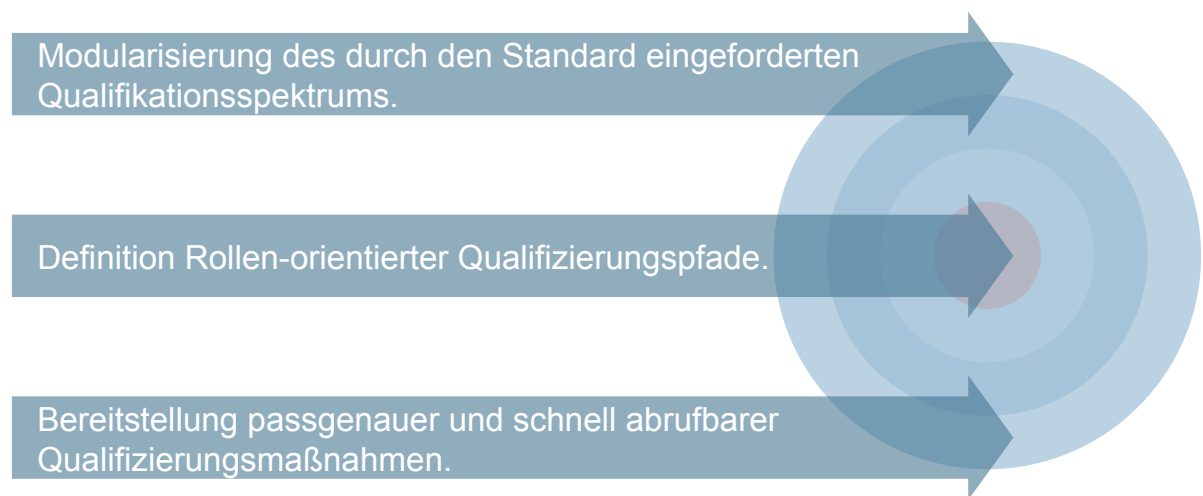
Dr. Stefan Walburg – SEE 2010

Seite 30

## Handlungsfeld <Qualifikation>



## Ziele des Qualifikationskonzepts



## Exemplarischer Ergebnistyp des Qualifizierungskonzepts: Qualifizierungsmatrix

Qualifikationsmodul \ Rolle	Anforderungs-analytiker	Modell-analytiker	Fachlicher Tester
Afo-Spezifikations-Templates	●	◐	◐
Sophist REgelwerk	●	◐	◐
UML 2	◐	●	◐
Review-Techniken	◐	◐	●
Model Driven Architecture	◐	●	○
Objektorientierte Analyse	◐	●	◐
ERM/SERM	◐	●	◐
V-Modell XT	◐	◐	◐
...			

○ keine Kenntnisse erforderlich ... ● Expertenwissen erforderlich

## Best Practices & Lessons Learned



„Mainstream-Module“ über Rahmenverträge absichern (Größendegressionseffekte, SLAs!).

Zeitpunkt der Qualifizierung ist entscheidend für den Lernerfolg (kein Modell der „kognitiven Bevorratung“).

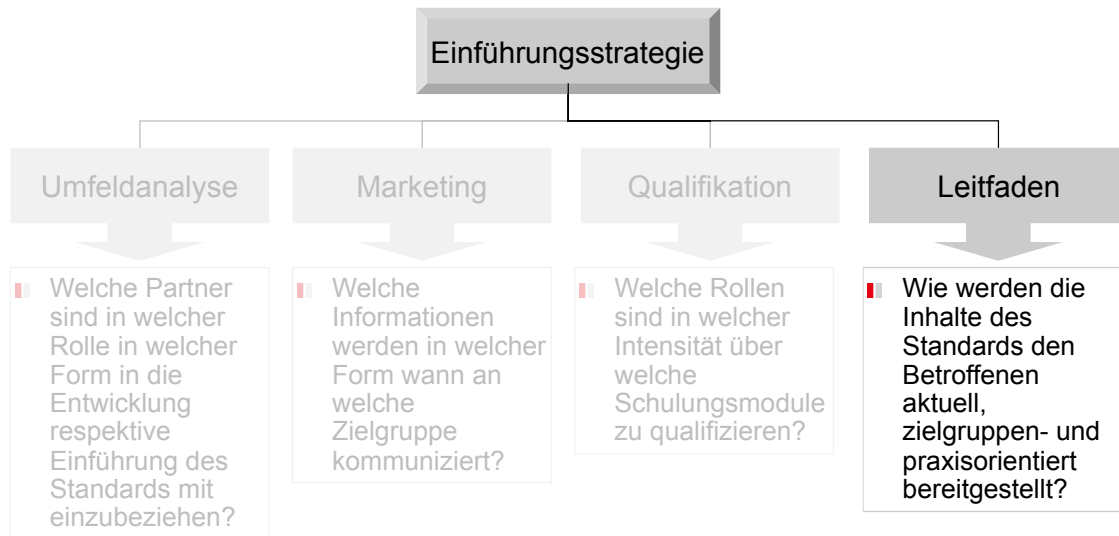
Sorgfältige Planung – Abstimmung der Maßnahmen mit den Produktverantwortlichen (Produkt-Piks berücksichtigen).

Coach als dedizierte Rolle vorsehen (Coach-the-Coach-Konzept).

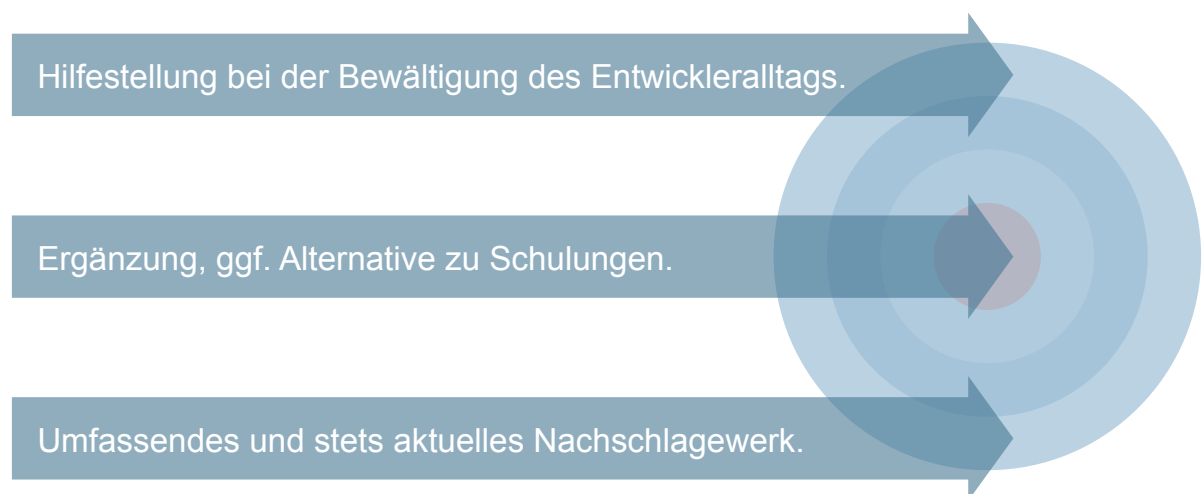
Kompaktkurs für Führungskräfte anbieten.

Selbst ein guter Qualifikationsmix ersetzt kein Coaching!

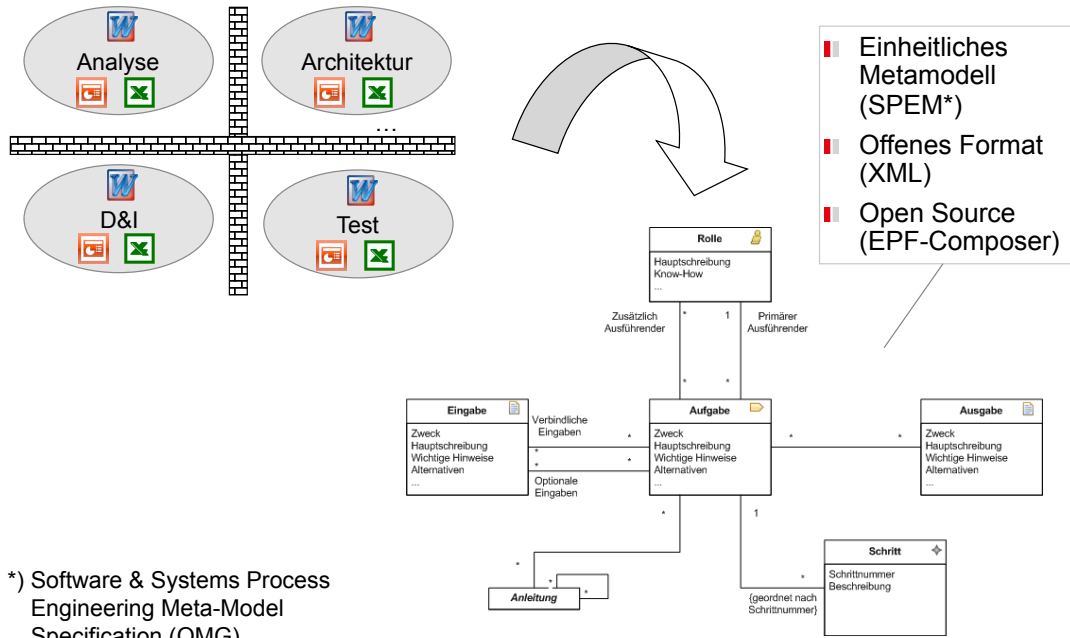
## Handlungsfeld <Leitfaden>



## Ziele des Leitfadens




## Von „Dokumentationsinseln“ zu einer integrierten und konsistenten Gesamtdokumentation



\*) Software & Systems Process Engineering Meta-Model Specification (OMG)

## Screenshot gemäß SPEM aus Anwendersicht

## Best Practices & Lessons Learned

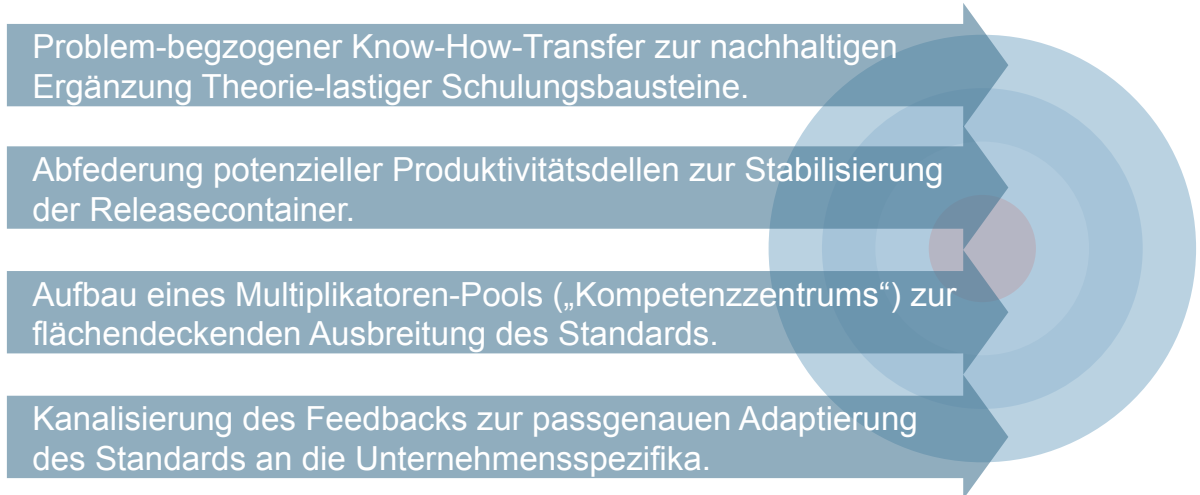


- Neue Dokumentationsformen benötigen eine „Mini-Einführungsstrategie“ (Vorteilsübersetzung, Handlingschulung, Releasezyklen etc.).
- Publikation neuer Dokumentationsformen darf erst bei angemessenem Reifegrad erfolgen.
- Akzeptanzkurve korreliert mit Anzahl operativer Hilfsmittel und Qualität der Aufbereitung.
- Print-fähige Dokumentation ist nach wie vor unerlässlich.

## Handlungsfeld <Coaching>

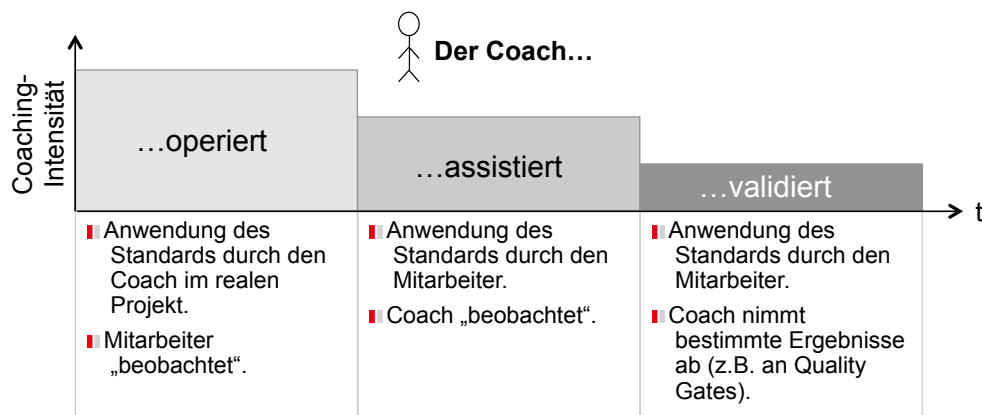


## Ziele des Coachingkonzepts



## Determinanten des Coaching-Aufwands


### ■ Das 3-Phasen-Modell des Coachings



### ■ Weitere Einflussfaktoren: Gap Soll-/Ist-Profil, Mitarbeiter-/Coach-Skillprofil, Disziplin (Analyse, Test etc.), Zieltermin etc.



## Best Practices & Lessons Learned

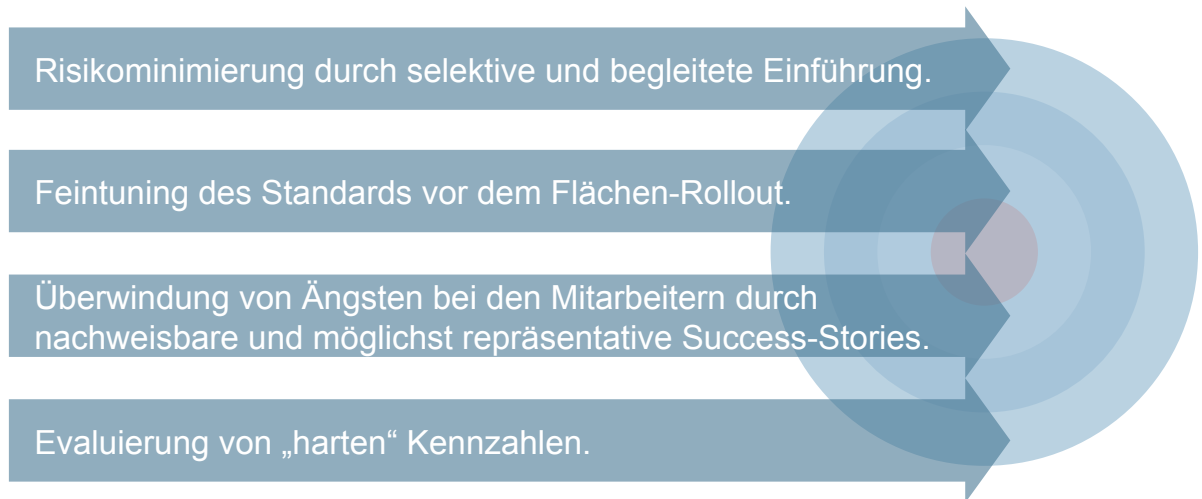


- Schriftliche Zielvereinbarung schützt vor „Coaching-Missbrauch“.
- Push-Prinzip zu Beginn zielführender als Pull-Prinzip.
- Coaching-Phase 3 sollte zur Vermeidung „böser Überraschungen“ an den Quality Gates um Zwischen-Reviews angereichert werden.
- Auswahl der Coaches entscheidend für die Akzeptanz des Standards.
- Gratwanderung zwischen Pragmatismus und Dogmatismus.

## Handlungsfeld <Pilotierung>



## Ziele des Pilotierungskonzepts

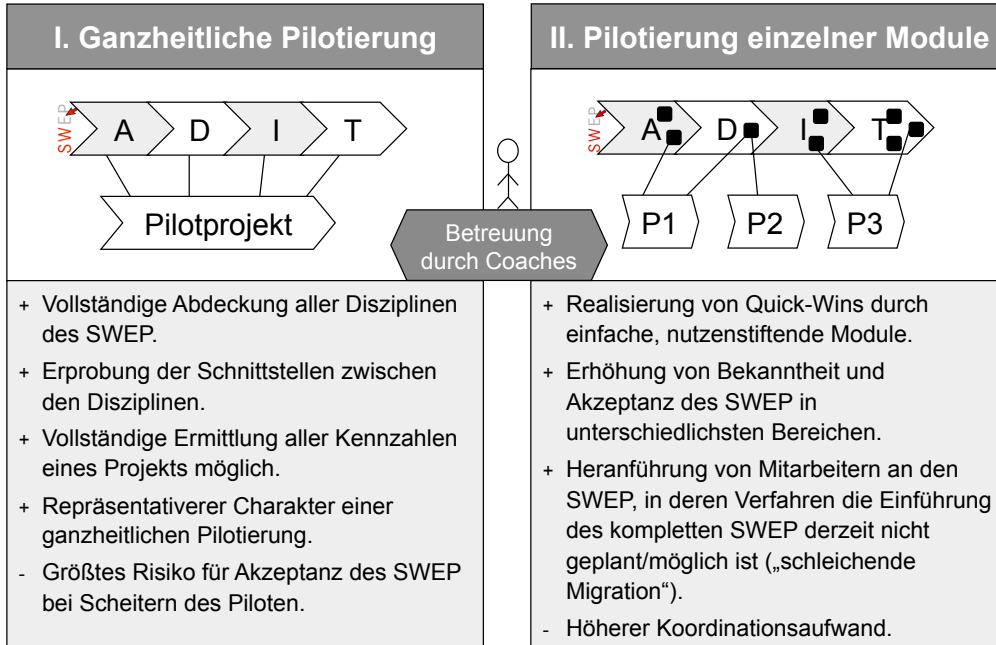


## Werkzeug zur Auswahl geeigneter Piloten: Pilotierungsmatrix (Ausschnitt)

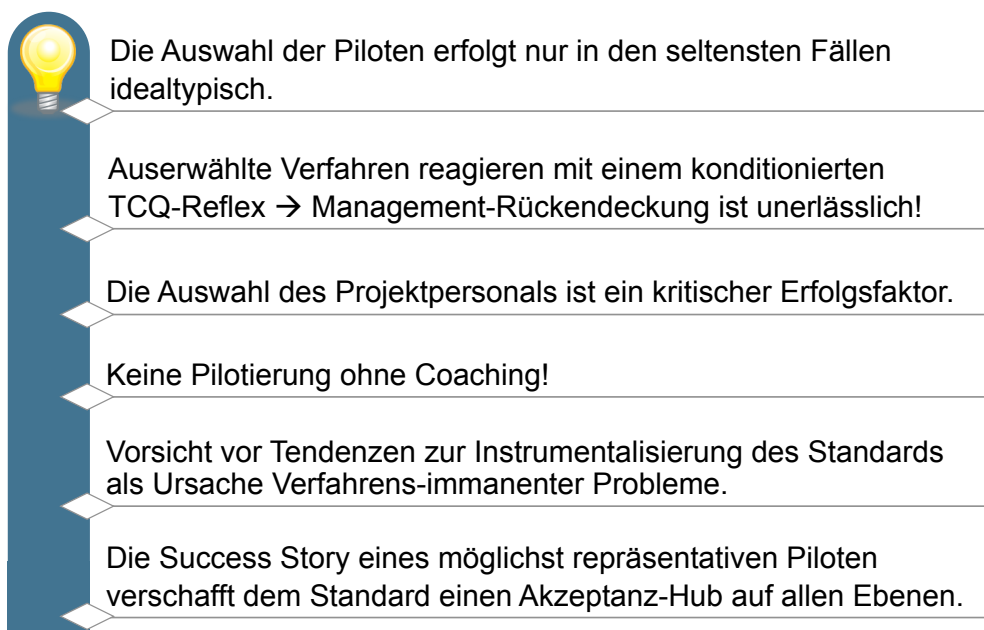
Kriterien	Faktor	Projekt 1	Projekt 2	Projekt 3	Projekt 4
Realistischer Kosten- und Zeitrahmen	5	0	1	1	2
„Standard-affine“ Mitarbeiter	5	3	2	2	1
Kurze Laufzeit (6-12 Monate)	5	3	2	3	0
Erfahrener Projektleiter mit Rückendeckung durch Vorgesetzten	4	2	2	2	2
Geringe Kritikalität	4	0	3	0	0
<b>Summe:</b>		<b>38</b>	<b>45</b>	<b>38</b>	<b>23</b>

- Kriterien werden gewichtet (Faktor 1=relativ unwichtig bis 5=essentiell).
- Einzelne Projekte werden hinsichtlich der Kriterien bewertet.
- Ordnungsrelation entlang des Produktes über alle Projekte.

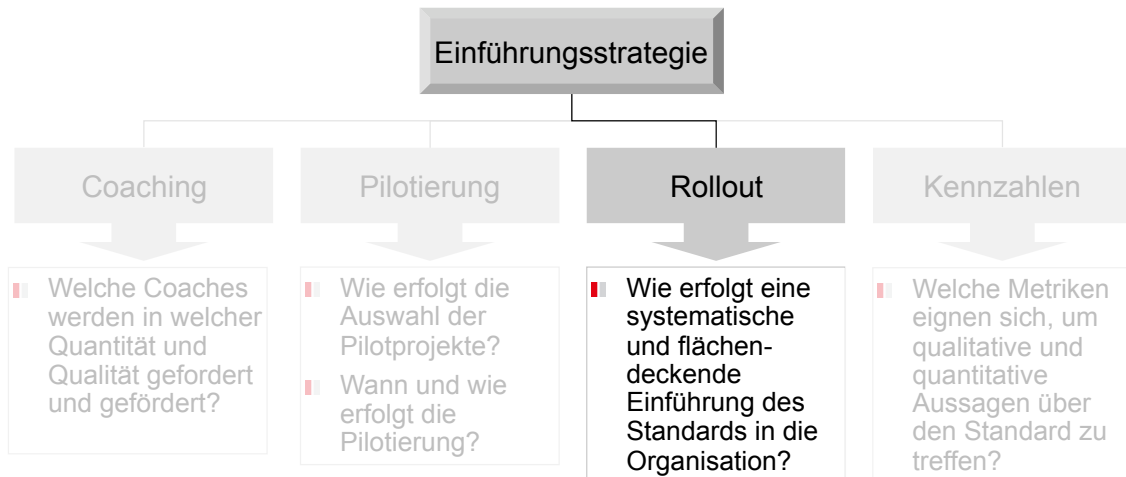
## Pilotierungsszenarien



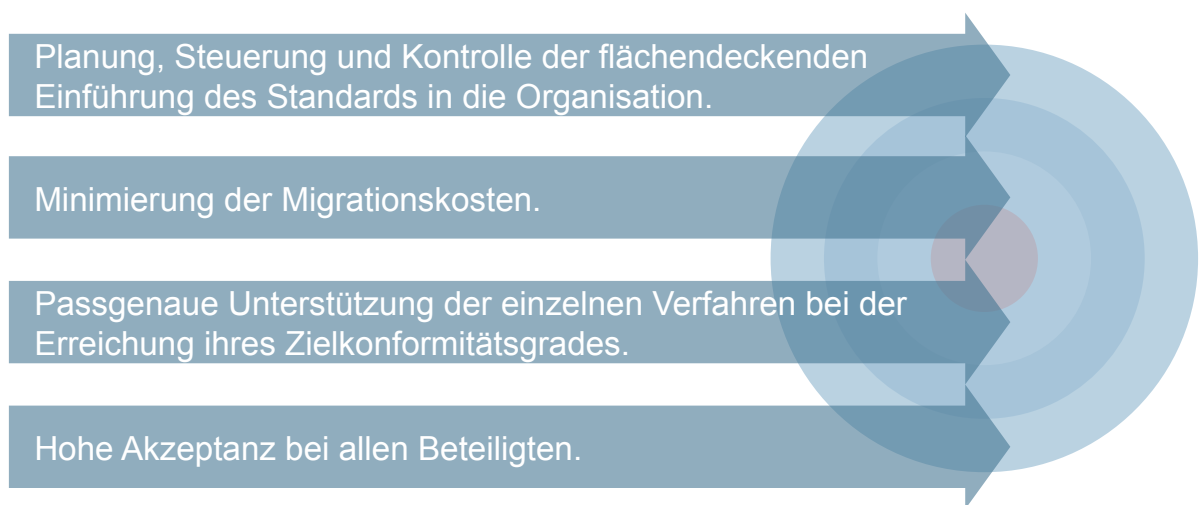
## Best Practices & Lessons Learned



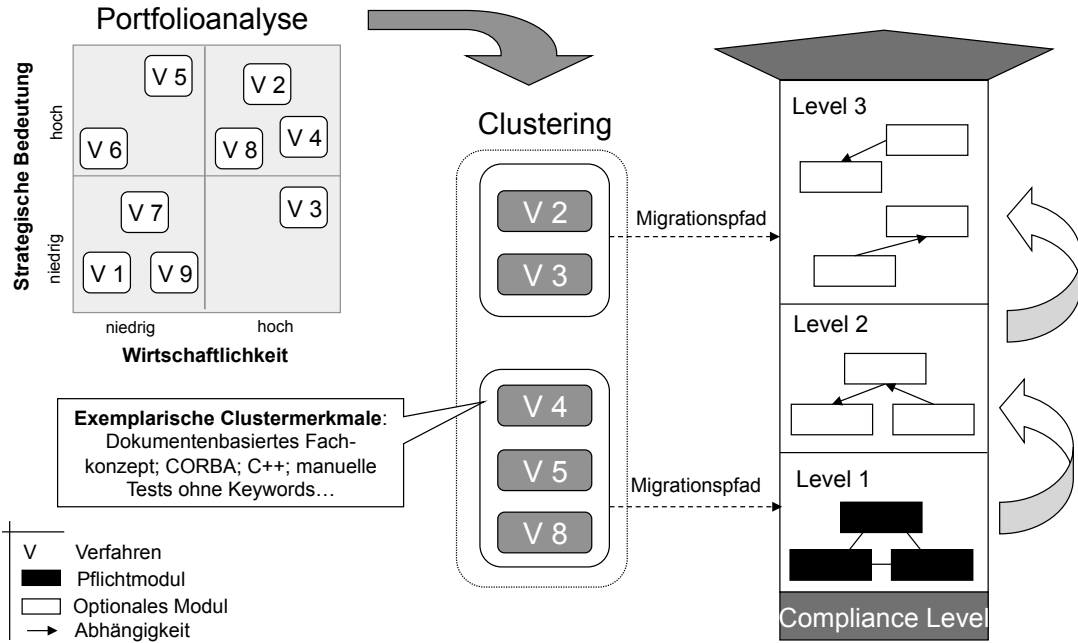
## Handlungsfeld <Rollout>



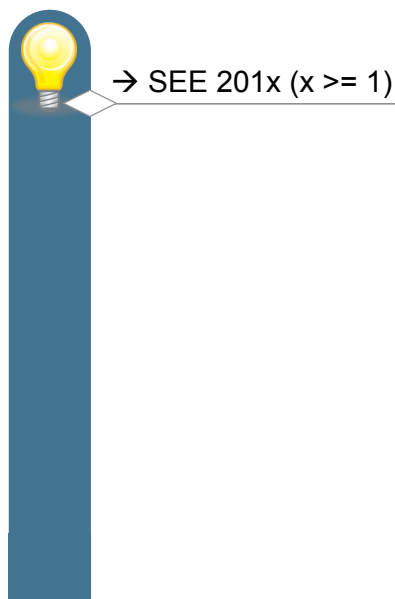
## Ziele des Rolloutkonzepts



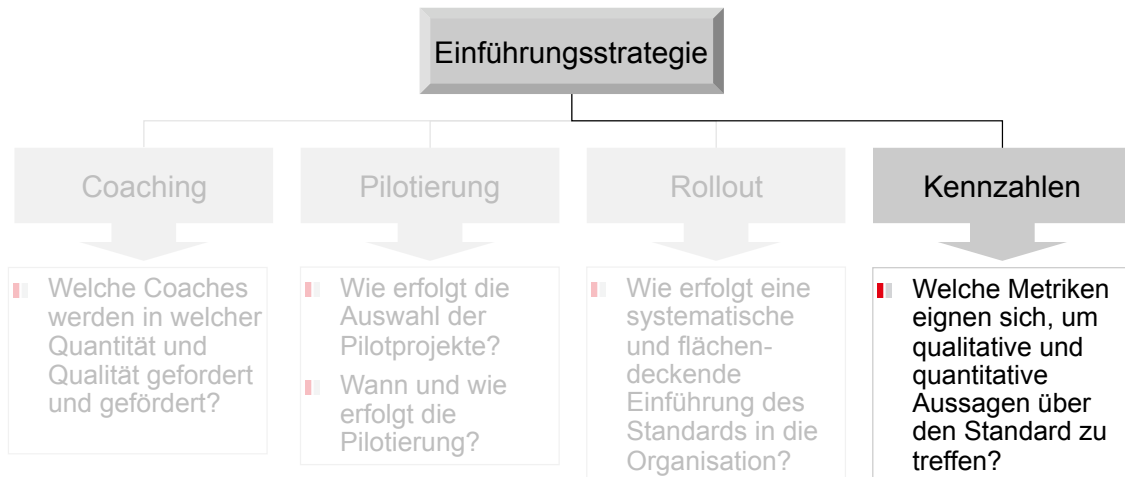
## Migrationsstrategie: Grundidee



## Best Practices & Lessons Learned



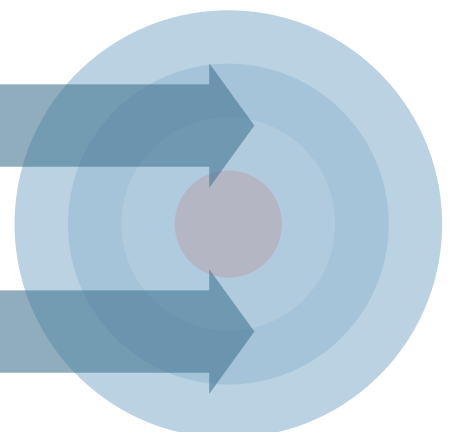
## Handlungsfeld <Kennzahlen>



## Ziele des Kennzahlensystems

Grundlage für die Planung und Steuerung einer flächendeckenden Einführung.

Schaffung von Transparenz auf „Ampelebene“.



## SWEP-Konformitätskennzahl: Grundidee

		← SWEP-Breite →					
		Verfahren 1	Verfahren 2	Verfahren 3	Verfahren 4	VK	Gewicht
SWEP-Tiefe	Analyse	M1			M1	1	5
	Architektur		M2			2	4
	D&I					3	2
	Test	M4			M4 M5	4	1
	Soll (=100%)	M1, M2, M4	M2, M5	M2	M1, M4, M5	Modul	Gewicht
Konformität	75%	50%	0%	100%	M1	1	
Verfahrenskl.	3	4	2	1	M2	1	
Gesamt	58%				M3	3	
					M4	2	
					M5	1	

$$SWEP-Konformität_{gesamt} = \frac{\sum SWEP-Konformität_{verfahren} \times Gewicht_{verfahren}}{\sum Gewicht_{verfahren}}$$

$$SWEP-Konformität_{verfahren} = \frac{\sum Modulexistenz_{Ist} \times Modulgewicht}{\sum Modulgewicht_{Soll}}$$

VK: Verfahrensklasse  
M: SWEP-Modul

## Best Practices & Lessons Learned



→ SEE 201x (x >= 1)

## Herzlichen Dank für Ihre Aufmerksamkeit !





## *7.2 Entwicklung und Einführung eines standardisierten Softwareentwicklungsprozesses*

## Die Entwicklung eines Vorgehensmodells für E-Government-Projekte der IHK-Organisation

Michael Tonndorf<sup>1</sup>, Franz Goschi<sup>2</sup>, Armin Saalmann<sup>2</sup>

<sup>1</sup>CSC Deutschland Solutions GmbH Sandstr. 7-9  
80335 München  
michael.tonndorf@csc.com

<sup>2</sup>Stabstelle QM, IHK-GfI mbH, Emil-Figge-Str. 86  
44227 Dortmund  
goschi@comnetmedia.de  
saalmann@comnetmedia.de

**Abstract.** In der IHK-Organisation<sup>1</sup> wird das Thema E-Government seit 2008 in Form eines Programms von Einzelprojekten systematisch entwickelt. Bis Ende 2009 waren 4 Projekte abgeschlossen, 7 Projekte laufend und es sind weitere Projekte für 2010 und Folgejahre geplant. Diese Projekte können verschiedenen Typen zugeordnet werden: Infrastruktur, Konzeption, Realisierung, Organisation. Daraus konkretisierte sich 2009 die Anforderung, für alle Projekte im Umfeld von E-Government eine einheitliche Begriffswelt und Vorgehensweise einzuführen.

### 1 Ausgangslage und Zielsetzung

Die 80 Industrie- und Handelskammern in Deutschland vertreten ca. 3,6 Mio. Unternehmen und sind das Selbstverwaltungsorgan der Deutschen Wirtschaft. Eine ihrer Stärken ist die regionale Verankerung und damit die Nähe zu den Unternehmen. Der Deutsche Industrie- und Handelskammertag (DIHK) ist der Dachverband der Industrie- und Handelskammern. Die IHKs stehen an der Schnittstelle zwischen dem öffentlichen Sektor und der Privatwirtschaft in Deutschland. Für die IHKs gilt, ganz analog zu den Verwaltungen des Bundes und der Länder, die Vorgabe der weitergehenden Digitalisierung interner und externer Prozesse zur Verbesserung des Dienstleistungsangebots für Bürger, Wirtschaft und Verwaltung. Diese Disziplin wird, vereinfacht gesagt, als E-Government bezeichnet. Das Programm IHK E-Government wurde im Jahr 2008 mit 4 Projekten gestartet:

- IT-gestützte Gewerbemeldungen,
- Sachverständigenverzeichnis,
- automatisierte Handelsregister-Eintragungsmeldungen,
- Unternehmensverzeichnis.

---

<sup>1</sup> Darunter sollen im Kontext dieses Papiers der Deutsche Industrie- und Handelskammertag (DIHK) und die 80 deutschen Industrie- und Handelskammern (IHKs) verstanden werden, die Auslandshandelskammern (AHKs) sind an dem Vorhaben nicht beteiligt

### 7.3 Die Entwicklung eines Vorgehensmodells für E-Government-Projekte

Bereits 2007 wurde mit Vorarbeiten zur Bereitstellung einer IT-Infrastruktur begonnen, die mit der Inbetriebnahme eines Enterprise Service Bus (ESB) sowie eines Enterprise Service Hub im Jahr 2009 beim zentralen IT-Dienstleister der IHKs, der IHK-GfI in Dortmund, abgeschlossen wurden. Es wird im Folgenden vorausgesetzt, dass die Rolle Auftraggeber (AG) konstant vom DIHK ausgeübt wird. Operativ wird dies durch das Projektbüro E-Government beim DIHK geleistet. Das Projektbüro bündelt die Aktivitäten der IHK-Organisation und schafft die Voraussetzungen, dass die Anforderungen des DIHK und der 80 IHKs zum E-Government tatsächlich bei einer Instanz zusammenlaufen und dadurch der AG als eine sichtbare Rolle wahrgenommen wird.

In der Rolle Auftragnehmer (AN) ist grundsätzlich jeder Lieferant zu sehen, der in einem E-Government-Projekt für den AG DIHK tätig wird. Die hier vorgestellten spezifischen Projekttypen wurden anhand konkreter Projektsituationen aus den Jahren 2008 und 2009 mit der IHK-GfI als Auftragnehmer modelliert. Die IHK-GfI ist der größte IT-Dienstleister der IHKs in Deutschland und betreibt ein gemeinsames Rechenzentrum für 56 IHKs. Jedoch kann prinzipiell jeder IT-Lieferant die hier beschriebene AN-Rolle einnehmen, der ein E-Government Projekt im Auftrag des DIHK durchführt.

2008 wurde beim AG die grundsätzliche Entscheidung für eine Orientierung der Vorgehensweise am V-Modell XT getroffen. Vorlagen auf der Grundlage des V-Modell XT 1.2.1 wurden manuell angepasst und für die Projektarbeit zur Verfügung gestellt. Eine erste Projektrichtlinie wurde erstellt. Es stellte sich jedoch bald heraus, dass diese Vorlagen die konkrete Projektsituation nur partiell widerspiegeln und ein hoher Coaching- und Einarbeitungsaufwand nötig wurde in einer Umgebung, in der das V-Modell noch nicht flächendeckend als Vorgehensstandard in Projekten eingeführt war.

Deshalb wurde 2009 damit begonnen, eine organisationspezifische Anpassung des V-Modells XT für die E-Government Projekte der IHK-Organisation vorzunehmen.

Ziel des Ansatzes ist es, eine Projektmanagement-Umgebung für die E-Government-Projekte der IHK-Organisation zu entwickeln, die die folgenden Anforderungen erfüllt:

- einheitliche Begriffswelt,
- angepasste Projekttypen und -varianten AG und AN,
- spezifische Projektdurchführungsstrategien,
- flexibler, aber sicherer Tailoring-Mechanismus,
- automatische Generierung der initialen Dokumentenvorlagen,
- automatisierte Initialisierung der Planung,
- Modellierung der AG-AN-Schnittstelle zwischen dem Auftraggeber DIHK und IT-Dienstleistern.

Ein weiteres Ziel der IHK-GfI ist über die Thematik E-Government hinaus, die Eignung des V-Modells XT als Grundlage des Projektmanagement-Standards der Organisation festzustellen. Gesteuert wird das Vorgehen auf AG-Seite vom Projektbüro, auf Seiten des AN durch die Stabsstelle QM und die Arbeitsgruppe „Einführung Projektmanagement“. CSC hat die Arbeiten zur Einführung des V-Modells seit 2008 sowohl auf Seiten des AG (DIHK) als auch auf Seiten des AN (IHK-GfI) in einer beratenden Rolle unterstützt und begleitet.

## 2 Die Rolle des V-Modell XT

Seit 2005 wird in der IHK-GfI das V-Modell XT im Sinne des Vorgehensbausteins Einführung und Pflege eines organisationsspezifischen Vorgehensmodells systematisch zum Aufbau eines Prozessmanagementsystems und zur Weiterentwicklung und Standardisierung des Qualitätsmanagements eingesetzt. Aufbauend auf den dabei gemachten positiven Erfahrungen mit dem V-Modell XT wird dieses auf zwei Ebenen zu Grunde gelegt:

- a. Aufbau einer Projektmanagement-Umgebung des Unternehmens IHK-GfI auf Basis eines organisationsspezifischen Vorgehensmodells,
- b. Entwicklung eines Vorgehensmodells für Projekte der IHK-Organisation im E-Government Umfeld.

Bei der Anpassung des V-Modells XT in dem hier beschriebenen Kontext stellt sich somit die Aufgabe, alle Elemente des V-Modells daraufhin zu überprüfen, ob diese das Vorgehen in den E-Government-Projekten adäquat widerspiegeln, und bei den Elementen, bei denen das nicht der Fall ist, mit dem V-Modell XT Editor eine organisationsspezifische Anpassung durchzuführen. Wenn dieser Prozess abgeschlossen ist, dann wird daraus ein angepasstes Vorgehensmodell resultieren, auf dessen Grundlage mit dem Werkzeug V-Modell XT Projektassistent die Initialisierung eines konkreten Projekts durchgeführt werden kann. In den nachfolgenden Kapiteln 4. bis 6. werden deshalb die V-Modell-Elemente vorgestellt, die im Anpassungsprozess für das IHK Vorgehensmodell geändert wurden. Dies betrifft nach dem generellen Tailoring-Verständnis des V-Modells XT

- Projekttypen und Projekttypvarianten,
- Vorgehensbausteine, darin definiert
- Produkte, Aktivitäten und Rollen.

## 3 Spezifische Besonderheiten der Organisation

Die IT-Dienstleister der IHK Organisation verfügen über eine Prozessinfrastruktur, die bei der Einführung des Vorgehensmodells für E-Government Projekte möglichst nahtlos integriert werden soll. Bei der IHK-GfI sind beispielhaft die Softwareentwicklungsprozesse gemäß RUP (Rational Unified Process) sowie die Betriebsprozesse gemäß ITIL (IT Infrastructure Library) zu nennen. Die künftige Lösung bei IHK-GfI setzt auch hier bei hausinternen Entwicklungsaufträgen auf eine Zweiteilung (in Anlehnung an [LM09]), nämlich eine Makro-Planungsebene gemäß V-Modell XT und eine Mikro-Planungsebene gemäß RUP bzw. ITIL.

Der Übergang von der Makro- zur Mikro-Planungsebene der Entwicklungsstrategie erfolgt im Anschluss an den Entscheidungspunkt Iteration geplant. Hier sind nacheinander die RUP-Disziplinen Anforderungserstellung (Pflichtenheft), Analyse & Design, Implementierung und Test zu absolvieren. Nach Durchlauf durch die RUP-Disziplinen geht der getestete Liefergegenstand über den Entscheidungspunkt Lieferung durchge-

führt zurück zur Makro-Ebene mit den aufeinander folgenden Entscheidungspunkten Abnahme erfolgt und Inbetriebnahme erfolgt, bei dem basierend auf den erreichten Stand des Produkts die nächste Iteration durch eine Entscheidung des Projektleitungsausschusses festgelegt wird. Diese Iterationsschleife zur Mikro-Planungsebene RUP und zurück wiederholt sich mit Zwischenabnahmen solange, bis der finale Durchlauf durch den Entscheidungspunkt Inbetriebnahme erfolgt ansteht.

Der Übergang zur Mikro-Planungsebene ITIL ist an den neu eingeführten Entscheidungspunkten in der Projektdurchführungsstrategie vorgesehen, wird jedoch erst in einer späteren Iteration des IHK Vorgehensmodells feiner ausgearbeitet.

Eine weitere spezifische Besonderheit der Organisation ist die Einrichtung des IHK E-Government IT-Boards. Das IT-Board hat den Auftrag, die E-Government-Projekte aus einem technischen Blickwinkel zu begleiten und auf einen möglichen Handlungsbedarf zu den Fragen der Gesamtarchitektur und der Standardisierung z.B. von Basiskomponenten und Datenmodellen zu reagieren und Expertenmeinungen zu aktuellen E-Government-Themen zu bündeln. In Personalunion fungiert das IT-Board außerdem als XIHK Datenkonferenz, in Analogie zur XÖV-Anwenderkonferenz im öffentlichen Sektor. Damit soll die kontinuierliche Weiterentwicklung des Datenaustauschformats XIHK sichergestellt werden.

## **4 Entwicklung von Projekttypen und Projekttypvarianten für das IHK Vorgehensmodell**

Auf Seiten des AG werden zwei neue Projekttypvarianten definiert und modelliert:

- AG-Projekt mit mehreren Teilprojekten (Programm),
- AG-Projekt IHK E-Government.

Auf Seiten des AN wurden aus den im Jahr 2009 laufenden 7 Projekten mit Schwerpunkt E-Government 4 als Pilotprojekte für die Entwicklung des IHK Vorgehensmodells ausgewählt. Diese decken ein charakteristisches Spektrum an Projekttypvarianten des V-Modells XT ab:

- AN-Projekt mit komponentenbasierter Entwicklung,
- AN-Projekt mit inkrementeller Entwicklung,
- AG-AN-Projekt mit inkrementeller Entwicklung,
- AG-AN-Entwicklungsprojekt mit Beschaffungsanteil.

Die Projektdurchführungsstrategien für die Projekttypen und Projekttypvarianten sind in das bestehende Prozessmanagement der IHK-GfI zu integrieren. Weitere Projekttypen können nach Bedarf aus dem V-Modell XT generiert werden. Alle o. g. Projektdurchführungsstrategien (PDS) des AN weisen folgende Gemeinsamkeiten auf:

- Entwicklungsstrategie gemäß RUP, mit Übergang von der Makro- zur Mikro-Planungsebene,

- neuer Vorgehensbaustein Betriebskonzept basierend auf der ITIL-Phase Service Design, mit dem zugehörigen Entscheidungspunkt Betriebskonzept festgelegt, der vor der Angebotsabgabe eingefügt wurde,
- neuer Vorgehensbaustein Inbetriebnahme gemäß Schritt Deployment der ITIL-Phase Service Transition mit dem zugehörigen Entscheidungspunkt Inbetriebnahme erfolgt im Anschluss an eine Abnahmeentscheidung.

Im Folgenden werden die Unterschiede zwischen dem IHK Vorgehensmodell und dem V-Modell XT Vers. 1.3 aus Sicht der Projekttypvarianten beschrieben.

Die generelle Intention des V-Modells XT ist es, einen idealisierten Projektablauf vom Entscheidungspunkt Projekt genehmigt auf Seiten des AG bis hin zum Entscheidungspunkt Projekt abgeschlossen abzubilden und den Projektbeteiligten damit einen „roten Faden“ durch das Projekt an die Hand zu geben. In der Realität wird ein solch idealisiertes Projekt schwer zu finden sein, die konkrete Situation in einem IHK E-Government Projekt wird dadurch nicht erfasst. Nun liegt die Stärke des V-Modells XT gerade darin, dass es an eine konkrete Branche bzw. Organisation und darüber hinaus an individuelle Projektsituationen angepasst werden kann. Dadurch fällt es der Organisation erfahrungsgemäß um Einiges leichter, sich in dem spezifischen Vorgehensmodell wiederzufinden, da hier ohne Einschränkungen „die Sprache des Kunden“ gesprochen werden kann.

Die Unterschiede zwischen dem IHK Vorgehensmodell und dem ursprünglichen V-Modell XT werden in den nachfolgenden 6 Abschnitten konkretisiert:

### 4.1 Projekttypvariante AG-Projekt mit mehreren Teilprojekten ( Programm)

Ein IHK E-Government Projekt läuft im Kontext des Programms IHK E-Government ab. Im V-Modell XT wird die Projekttyp-Variante AG Projekt mit mehreren Auftragnehmern als eine Ausprägung für Multi-Projektmanagement definiert. Eine andere mögliche Ausprägung ist ein Programm. Ein Programm ist dadurch gekennzeichnet, dass das Programm eine Anzahl von Einzel- bzw. Teilprojekten umfasst. Im Gegensatz zum Multi-Projektmanagement ist aber kein Gesamt-Lastenheft definiert, das ein Multi-Projekt insgesamt kennzeichnet. Die Projekte stehen zwar in einem inhaltlichen Zusammenhang, werden aber unabhängig von Projektleitern des AG geplant und durchgeführt. Die Projekte können vom gleichen AN oder von verschiedenen Auftragnehmern realisiert werden. Analog zum Multi-Projekt wird für das Programm ein Start-Termin und ein Ende-Termin festgelegt. In diesem Zeitrahmen können Projekte unabhängig voneinander gestartet und beendet werden. Zu einem bestimmten Zeitpunkt kann ein Projekt im Zustand geplant, in Bearbeitung oder abgeschlossen sein. Die Statusänderungen eines Projekts von

- geplant → in Bearbeitung und von
- in Bearbeitung → abgeschlossen

werden jeweils anlässlich einer Sitzung des Projektlenkungsausschusses festgestellt (Entscheidungspunkt). Zu einem bestimmten Zeitpunkt befindet sich daher immer eine definierte Menge von Projekten im Portfolio, die sich aus den geplanten, in Bearbeitung

befindlichen und abgeschlossenen Projekten zusammensetzt. Natürlich ist es möglich, dass nach Programmstart weitere Projekte in das Portfolio aufgenommen werden, so dass die Gesamtzahl der Projekte in der Regel dynamisch wächst.

Der Unterschied zwischen einem Multiprojekt und einem Programm liegt also darin, dass die Projekte des Multiprojekts nach V-Modell XT in jedem Fall von Anfang bis Ende durchlaufen werden, da diese einen Teil der Gesamtfunktionalität des Multiprojekts beisteuern. Im Programm hingegen wird für jedes laufende Projekt im Entscheidungspunkt Projektfortschritt überprüft jeweils festgestellt, ob das Projekt wie geplant weitergeführt wird, ob es im Gesamtportfolio neu bewertet wird, höher oder niedriger priorisiert, oder aber unterbrochen oder ggf. abgebrochen wird.

#### **4.2 Projekttypvariante AG-Projekt IHK E-Government**

In Ergänzung zur unverändert übernommenen Struktur der Projektdurchführungsstrategie im IHK Vorgehensmodell gibt es einen weiteren Entscheidungspunkt Dienst online gestellt, allerdings sind einige Produktbezeichnungen anders gewählt (siehe Kap. 5.2.). Auch hat ein einzelnes E-Government Projekt keinen eigenen Lenkungsausschuss, diese Rolle wird vom Lenkungsausschuss des Gesamtprojekts wahrgenommen.

#### **4.3 Projekttypvariante AN-Projekt mit komponentenbasierter Entwicklung**

Diese Projekttypvariante betrifft vor allem die IHK-GfI als Betreiber der zentralen E-Government Infrastruktur. Zum Ausbau der Betriebsumgebung sowie zur Optimierung der über die E-Government Infrastruktur (ESB und Hub) ablaufenden Prozesse sind unterstützende Komponenten unterschiedlicher Größenordnung erforderlich. Diese Komponenten können sowohl hausintern erstellt als auch in Form externer Software-Module hinzugekauft werden. Der Systemausbau anhand derartiger Komponenten wird im Rahmen des Gesamtauftrags zum Betrieb der E-Government Infrastruktur beauftragt, wobei im Falle technischer Komponenten die Erstellung auch ohne gesonderte Beauftragung durch den AG erfolgen kann.

Die Entwicklungsstrategie basiert auf dem bei der IHK-GfI etablierten Entwicklungsstandard RUP, wobei bei externen Komponenten die Disziplinen Analyse & Design sowie Implementation entfallen und der Schwerpunkt auf Schnittstellenanforderungen und integrative Tests gelegt wird. Zum Ende jedes Iterationsdurchlaufs wird zum Entscheidungspunkt Abnahme erfolgt wieder die Makro-Planungsebene erreicht.

Mit dem neu eingeführten Vorgehensbaustein Betriebskonzept müssen die Betriebsanforderungen der neuen Komponenten (Liefergegenstände) verbindlich festgelegt und mit den Betriebsanforderungen des bestehenden Gesamtsystems (ESB und Hub) in Einklang gebracht werden.

Die neuen Komponenten müssen so ausgelegt sein, dass sowohl die Inbetriebnahme (neuer Vorgehensbaustein Inbetriebnahme) als auch der nachfolgende kontinuierliche

Betrieb ohne Störung bereits vorhandener Komponenten und bereits eingeführter E-Government Prozesse erfolgt.

### 4.4 Projekttypvariante AN-Projekt mit inkrementeller Entwicklung

Diese Projekttypvariante eignet sich für den Standardfall der iterativen Produktentwicklung eines AN auf Basis eines Projektauftrags eines AG.

Die PDS unterscheidet sich, wie bereits das AN-Projekt mit komponentenbasierter Entwicklung, nur in den folgenden Punkten vom Standard:

- a. die iterative Entwicklungsstrategie gemäß RUP und
- b. die zusätzlichen Entscheidungspunkte Betriebskonzept festgelegt vor der Angebotsabgabe sowie Inbetriebnahme erfolgt nach jeder Abnahme.

### 4.5 Projekttypvariante AG-AN-Projekt mit inkrementeller Entwicklung

Diese Projekttypvariante stellt die Basis für Projekte dar, die nicht vom AG beauftragt werden, sondern im Zuge des Portfoliomanagements der IHK-GfI aus eigener Initiative entwickelt werden. Hierbei kann es sich sowohl um E-Government als auch anderweitige Projekte handeln, die im Hause IHK-GfI gehostet werden. Die Anpassungen leiten sich von der Prämisse ab, möglichst gleichartige Verfahren im Hause zu etablieren. Damit ist gewährleistet, dass Mitarbeiter, die von einem E-Government Projekt zu einem Projekt wechseln, das nicht im E-Government Umfeld angesiedelt ist, die gleichen Rahmenbedingungen zur Projektdurchführung vorfinden.

In diesem Fall muss das Betriebskonzept des zu erstellenden Liefergegenstands zwingend vor Einstieg in die erste Iteration festgelegt und mit den Betriebsanforderungen des bestehenden Gesamtsystems abgestimmt werden. Nach Abnahme wird der Liefergegenstand gemäß Vorgehensbaustein "Inbetriebnahme" in das Betriebsumfeld eingeführt.

Auch diese Projektdurchführungsstrategie enthält somit wie oben beschrieben zwei zusätzliche Entscheidungspunkte Betriebskonzept festgelegt und Inbetriebnahme erfolgt. Die Entwicklungsstrategie auf Basis RUP ist bei der IHK-GfI vorgegeben. Die Erstellung des Liefergegenstands erfolgt regelmäßig in mehreren Iterationen.

### 4.6 Projekttypvariante AG-AN-Projekt mit Beschaffung

Diese Projekttypvariante ist gegenüber allen vorgenannten Varianten von einem erheblich reduzierten Entwicklungsanteil gekennzeichnet, der sich auf die Anpassung des zu beschaffenden Fertigprodukts an die Gegebenheiten der E-Government Infrastruktur bzw. der bei IHK-GfI etablierten Hosting-Umgebung beschränkt. Die Anpassung kann mittels individueller Konfiguration oder durch Hinzunahme neuer oder erweiterter E-Government Komponenten erfolgen. Die Erstellung zusätzlich benötigter Komponenten kann in Form eines Unterauftrags intern oder extern vergeben werden. Alle Aspekte der



Beschaffung entsprechen denen des unverändert übernommenen Projektmerkmals "Fertigprodukte".

Gerade bei der Auswahl von Fertigprodukten ist es extrem wichtig, die Einhaltung der Betriebsanforderungen exakt zu eruieren. Das konfigurierte Fertigprodukt inkl. ggf. erforderlicher E-Government Komponenten muss ohne Störung der bereits vorhandenen E-Government Prozesse in Betrieb genommen werden können.

Diese Projektdurchführungsstrategie enthält, analog zum oben Beschriebenen, gegenüber der Standard-Projektypvariante AG-AN Entwicklungsprojekt zwei zusätzliche Entscheidungspunkte Betriebskonzept festgelegt und Inbetriebnahme erfolgt. Die Entwicklungsstrategie reduziert sich im Wesentlichen auf Integrationsaspekte.

## 5 Anpassung der Modellelemente des IHK Vorgehensmodells

In den folgenden Abschnitten 5.1 bis 5.3 werden die geänderten Modellelemente für das IHK Vorgehensmodell übersichtsartig dargestellt. Die referenzmäßige Ausgestaltung erfolgt im V-Modell XT Editor und ist mit den Projektbeteiligten bei AG wie AN im Detail abzustimmen.

### 5.1 Anpassung der Vorgehensbausteine

Es werden im IHK Vorgehensmodell die neuen Vorgehensbausteine

- Portfoliomanagement AG,
- Portfoliomanagement AN,
- Datenmodellierung (gemäß XIHK Datenmodell),
- Betriebskonzept,
- Inbetriebnahme

definiert, die Vorgehensbausteine

- Anforderungsmanagement (AG) und
- Anforderungsmanagement (AN)

auf Basis des Vorgehensbausteins Anforderungsfestlegung neu definiert. Die Vorgehensbausteine umfassen nach Maßgabe des V-Modells XT zusammengehörige Produkte, Aktivitäten und Rollen. Diese werden im Folgenden summarisch vorgestellt.

### 5.2 Anpassung der Produkte und Aktivitäten

Die Produkte im IHK Vorgehensmodell werden gemäß den Projektrichtlinien IHK E-Government benannt und als Vorlagen bereitgestellt. Grundlage sind die Produktdefinitionen aus dem V-Modell XT. Individuelle Änderungen bzw. Ergänzungen sind durch die folgenden Produkte gegeben:

- (E-Government) Dienstleistungsportfolio,
- Portfoliobewertung,
- Bedarfsanalyse,
- Projektskizze (ersetzt den Projektvorschlag),
- Projektrahmendokument 1 („Projekthandbuch“),
- Projektrahmendokument 2 (Projektrahmendokument 1 plus Lastenheft),
- Projektrahmendokument 3 (Projektrahmendokument 1 plus Pflichtenheft),
- Betriebskonzept (gemäß ITIL-Phase Service Design, unterteilt in SLA (Service Level Agreement), Kapazitäts-, Verfügbarkeits-, Sicherheitsplan),
- Konfigurationsmanagement-Datenbank (CMDB),
- Inbetriebnahmeplan (gemäß ITIL-Phase Service Transition, Schritt Deployment, unterteilt in Ablauf-, Einsatz-, Info-, Notfallplan).

Da die Aktivitäten und Produkte inhaltlich im Verhältnis 1:1 zueinander stehen, wird hier auf die Aufzählung der entsprechenden Aktivitäten verzichtet.

### 5.3 Anpassung der Rollen

Ein wesentliches Element des IHK Vorgehensmodells sind die neuen oder geänderten Rollen, da die Rollen im IHK Vorgehensmodell genau so benannt werden können, wie sie in der Projektpraxis eingeführt wurden. Die Rollen ergeben sich entweder durch eine Neudefinition oder aber eine Umbenennung existierender Rollen. Die Bezeichnungen der AN-Rollen entsprechen denen, die bei IHK-GfI eingeführt sind. Diese sind noch im Detail mit den Rollen, die im bereits in Kraft gesetzten IT-Prozessmanagement Handbuch der IHK-GfI definiert sind, abzustimmen.

- Projektlenkungsausschuss (AG),
- Projektarbeitsgruppe (AG),
- IT-Board (technisches Steuerungsgremium, von AG-, AN- und weiteren Interessenvertretern besetzt),
- Projektmanager Programm (AG),
- Projektreferent (AG),
- Projektassistent (AG),
- Controller (AG),
- Anforderungsanalytiker (AG),
- Projektleiter (AG),
- Projektleiter (AN),
- Produktmanager (AN) (Produktbetreuung sowie Anforderungsanalyse und Erstellung),
- Release Manager (AN).

## 6 Erste Erfahrungen

Mit einem projektspezifischen Tailoring der Auslieferungsversion des V-Modells Version 1.3 konnten zunächst die ersten individuellen Erfahrungen mit der werkzeuggestütz-

ten Arbeit mit dem V-Modell XT gewonnen werden. In diesem Zusammenhang wurden die beiden einschlägigen Werkzeugfamilien zum V-Modell XT evaluiert.

Es hat sich in Folge der bisherigen Arbeiten gezeigt, dass das V-Modell XT tatsächlich die Flexibilität aufweist, die benötigt wird, um von einem allgemeinen, organisationsneutralen Standard zu einem organisationsspezifischen Vorgehensmodell zu kommen, in dem sich die Organisation wiederfinden kann. Dadurch kann die Akzeptanz des Vorgehensstandards enorm gesteigert werden und die damit durchgeführten Projekte mit einer wesentlich gesteigerten Prozesssicherheit durchgeführt werden. Der Wert der Arbeiten wird im Laufe der Zeit weiter zunehmen, wenn einmal alle benötigten Vorlagen erstellt worden sind und das IHK Vorgehensmodell mehr an Wiedererkennungswert erlangt hat. Die Funktionalitäten, die die Werkzeuge V-Modell XT Editor und V-Modell XT Projektassistent mit sich bringen, machen diese organisationsspezifischen Anpassungen eigentlich erst möglich.

Zusammengefasst stellen sich die bisherigen Erfahrungen bei der Entwicklung des IHK Vorgehensmodells wie folgt dar:

- Für eine zielorientierte Vorgehensweise müssen die Beteiligten ein gutes Verständnis vom V-Modell XT sowie Erfahrungen mit den Werkzeugen des V-Modells XT haben, was nur im Rahmen einer
- V-Modell XT Schulung erworben werden kann.
- Die Modellierung der AG- und AN-Projekttypvarianten sollte „aus einem Guss“ erfolgen, so kann am effizientesten eine Abstimmung über die gemeinsame Projektvorgehensweise von AG und AN herbeigeführt werden.
- Durch die Möglichkeiten des V-Modells XT bei organisationsspezifischen Anpassungen können die individuellen Prozesse, Rollen und Produkte der Organisation direkt so modelliert werden, dass sich die Beteiligten im IHK Vorgehensmodell wiederfinden und nicht die „Sprache des V-Modells XT“ lernen müssen.
- Das V-Modell XT kann somit über die Thematik E-Government hinausgehend als Grundlage für ein Projektmanagement-System der Organisation entwickelt werden.

**Gesamtfazit:** Das V-Modell XT ist für organisationsspezifische Anpassungen bestens geeignet. Die Werkzeuge bieten eine gute Unterstützung für die Nutzung im Projektalltag. Ohne Kenntnisse und hinreichende Dokumentation des V-Modell XT Metamodells kann eine solch weitgehende individuelle Anpassung des V-Modells XT kaum ohne externe Unterstützung geleistet werden. Das Werkzeug V-Modell XT Projektassistent ist hingegen so einfach in der Bedienung, dass ein IT-Projektleiter damit die drei Schritte der Projektinitialisierung

- Tailoring des Projekt-Vorgehensmodells,
- Initialisierung der Planung und
- Erzeugung der Projektvorlagen

selbständig erfolgreich durchführen kann, ohne dass dabei vertiefte V-Modell-Kenntnisse erworben werden müssen.

### 7 Ausblick, Weitere Pläne

Im nächsten Schritt sollen die vorgestellten Projekttypvarianten der Pilotprojekte im ausgewählten Werkzeug modelliert werden, so dass diese beim Tailoring für ein neues Projekt direkt zur Verfügung stehen. Hierbei erweist sich das Beispiel „Vorgehensmodell Bayern“ [Sch09] als gute Orientierungshilfe. Die Anpassung der Dokumentenvorlagen ist ein kontinuierlicher Prozess, der eher bedarfsgesteuert abläuft, was sich aber problemlos in den Gesamtprozess der Entwicklung des organisationsspezifischen Vorgehensmodells integrieren lässt.

Mit den bisherigen Arbeiten konnte auf Grundlage der Pilotprojekte die Tragfähigkeit des Konzepts uneingeschränkt nachgewiesen werden. Darauf basierend werden im nächsten Schritt die ersten Iterationen beim „Feintuning“ des Vorgehensmodells durchlaufen. Dies umfasst schwerpunktmäßig die vollständige Abbildung des IHK Vorgehensmodells im V-Modell XT Editor. Daraus ist eine Strategie zu entwickeln, wie das Vorgehensmodell auch flächendeckend für IT-Projekte in der IHK-Organisation verfügbar gemacht werden kann. Eine beispielhafte, erfolgreiche Durchführung der laufenden Pilotprojekte auf Basis des hier vorgestellten Vorgehensmodells wird eine entscheidende Voraussetzung dafür sein.

### Literaturverzeichnis

- [LM09] Lewitz, O.; Meseberg, U.: Sprinten mit dem V-Modell XT (WWW-Seite, Stand: 26.11.2009). Inter net: <http://www.computerwoche.de/software/softwareinfrastruktur/1911008> (Zugriff: 27.11.2009, 11.32 MEZ).
- [Scho09] Schober, Chr., Mück, A.: Das Vorgehensmodell Bayern im neuen Gewand, SEE 2009, Berlin.



# Die Entwicklung eines Vorgehensmodells für E-Government-Projekte der IHK-Organisation

Michael Tonndorf

CSC Deutschland Solutions GmbH

Franz Goschi, Armin Saalmann

ComNetMedia AG

SEE 2010

Köln, 4. Mai 2010



Vorgehensmodell für E-Government-Projekte der IHK-Organisation

SEE 2010 1

## Ausgangslage

- Die 80 Industrie- und Handelskammern sind Teil des öffentlichen Sektors in Deutschland mit Schnittstellen zu den Bundes-, Landes- und Kommunalverwaltungen.
- Die IHKs sind das Selbstverwaltungsorgan der deutschen Wirtschaft und erfüllen zahlreiche gesetzliche Aufgaben z.B. in den Bereichen Sachverständigenwesen, Berufsbildung, Außenwirtschaft.
- 2008 erfolgte der Start des Programms IHK E-Government analog zu den E-Government Initiativen im öffentlichen Sektor.
- Damit war der Rahmen für eine Reihe von Projekten mit einem ähnlichen fachlichen, technischen und organisatorischen Umfeld gegeben.
- In einer Vorphase wurde bereits die Basisinfrastruktur (Enterprise Service Bus und Service Hub), die alle IHKs verbindet, entwickelt und in Betrieb genommen.
- Es ergaben sich günstige Vorbedingungen für die Entwicklung einer einheitlichen Vorgehensweise in den Projekten als „IHK Vorgehensmodell“



Vorgehensmodell für E-Government-Projekte der IHK-Organisation

SEE 2010 2

## Motivation und Zielsetzung

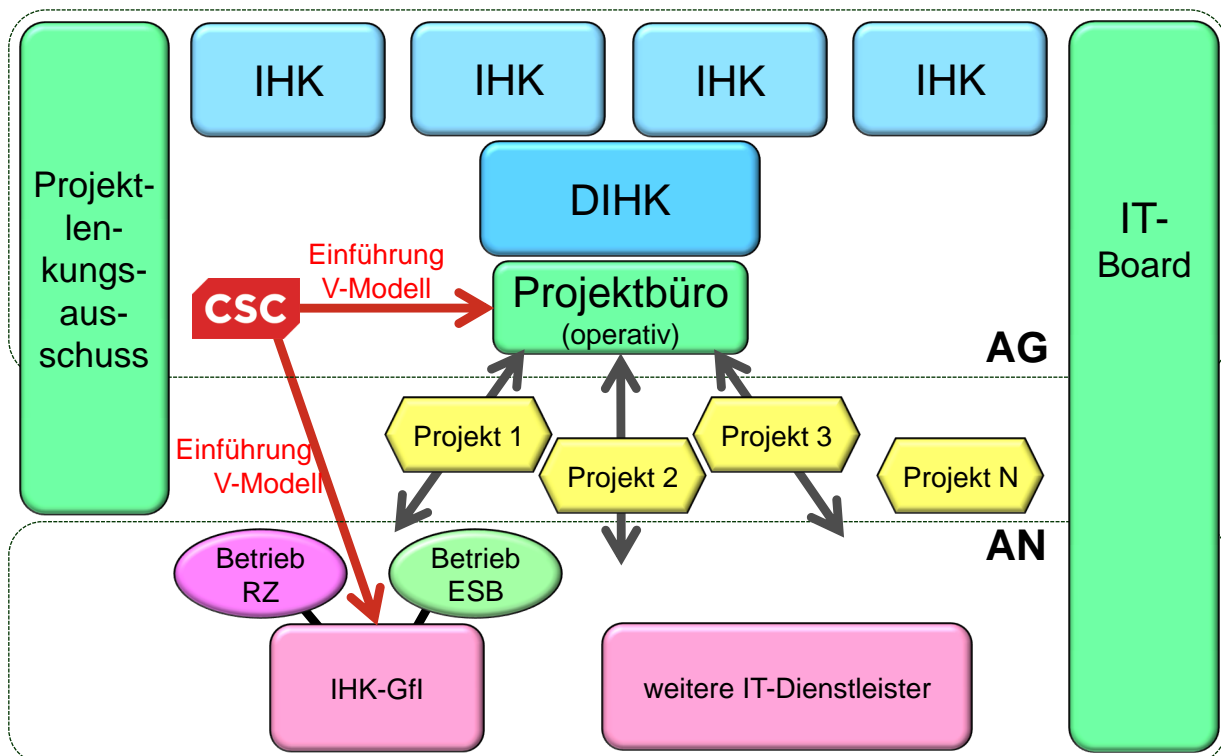
- Entwicklung eines organisationspezifischen Vorgehensmodells für die E-Government Projekte der IHKs auf Grundlage des V-Modells XT<sup>1</sup>
- kurzfristige Unterstützung für Projekte aus dem Kontext E-Government
- mittelfristige Unterstützung für alle Typen von IT-Projekten
- Vorgehensstandard als Grundlage für das Projektmanagement

### Prämissen:

- Orientierung an den vorhandenen Begriffen und Strukturen der IHKs
- Aufteilung in Projekttypen für AG und AN, so wie im Programm IHK E-Government konkret gelebt
- Verträglichkeit mit vorhandenen Projektmanagementverfahren und Prozessmodellen in der IHK-Organisation

<sup>1</sup> Das V-Modell® XT ist urheberrechtlich geschützt. Copyright © 2006 V-Modell® XT Autoren und andere. Alle Rechte vorbehalten.

## Organisatorisches Umfeld



## Anforderungen

### Organisationsspezifische Anpassung des V-Modells XT für die E-Government Projekte der IHK-Organisation:

- einheitliche Begriffswelt
- angepasste Projekttypen AG und AN
- spezifische Projektdurchführungsstrategien
- flexibler, aber sicherer Tailoring-Mechanismus
- automatische Generierung der initialen Dokumentenvorlagen
- automatisierte Initialisierung der Planung
- Modellierung der AG-AN-Schnittstelle zwischen dem Auftraggeber DIHK und IT-Dienstleistern

## Verlauf der bisherigen Arbeiten

### Einführung V-Modell XT für die E-Government-Projekte der IHKs

- 2008 grundsätzliche Entscheidung für die Verwendung des V-Modells XT im Programm IHK E-Government
  - Vorlagen aus dem V-Modell XT 1.2.1 wurden manuell angepasst
  - erste Projektrichtlinie herausgegeben
  - 4 initiale Projekte
- Praktische Erkenntnisse:
  - Vorlagen des V-Modells XT für Projektsituation nur bedingt geeignet
  - hoher Coaching- und Einarbeitungsaufwand
- 2009 Zielsetzung einer organisationsspezifischen Anpassung des V-Modells XT für die E-Government Projekte der IHK-Organisation

## Projekttypen / Projektvarianten

### Projekttypvarianten beim AG

- Zwei neue Projekttypvarianten definiert und modelliert:
  - AG-Projekt mit mehreren Teilprojekten (Programm)
  - AG-Projekt IHK E-Government

### Projekttypvarianten beim AN

- Vier neue Projekttypvarianten definiert und modelliert:
  - AN-Projekt mit komponentenbasierter Entwicklung
  - AN-Projekt mit inkrementeller Entwicklung
  - AG-AN-Projekt mit inkrementeller Entwicklung
  - AG-AN-Projekt mit Beschaffung

## AG-Projekte

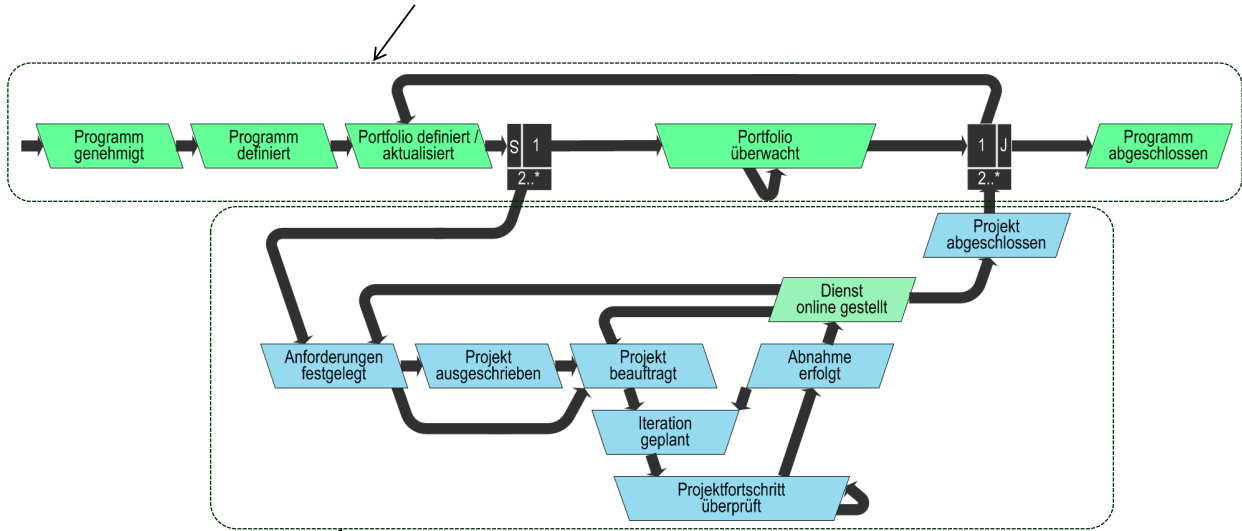
### Projekttypvariante: AG-Projekt mit mehreren Teilprojekten (Programm)

- Ein Projekt ist ein **IHK E-Government Projekt**, wenn es im Kontext des Programms IHK E-Government durchgeführt wird (pragmatische Def.)
- Ein **Programm** umfasst eine Anzahl von (Teil-)Projekten
- Kein Gesamt-Lastenheft
- Inhaltlicher Zusammenhang, jedoch unabhängige Planung und Durchführung seitens Projektleiter des AG
- Projektvergabe an verschiedene AN möglich
- Start- und Endetermin des Programms vorgegeben
  - Die Projekte können unabhängig voneinander gestartet und beendet werden
  - Entscheidungspunkt **Portfolio überwacht**:  
Projekt wie geplant weiterführen, im Gesamtportfolio neu bewerten, höher oder niedriger priorisieren, unterbrechen oder ggf. abbrechen



## Projekttypvarianten AG

### AG: Programm- / Portfoliomanagement - IHK E-Government



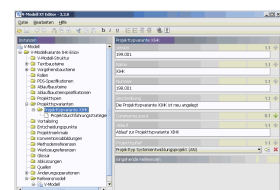
### Projekttypvariante: AG-Projekt IHK E-Government



## Die Rolle des V-Modells bei der IHK-GfI

### Vorreiterrolle der IHK-GfI bei der Einführung des V-Modells

- Seit 2005 systematischer Einsatz des V-Modell XT bei IHK-GfI
- Vorgehensbaustein „Einführung eines organisationspezifischen Vorgehensmodells“
  - Aufbau des *IT-Prozessmanagementsystems* der IHK-GfI
  - Weiterentwicklung des *Qualitätsmanagements*
- E-Government
  - Entwicklung eines *Vorgehensmodells für Projekte* der IHK-Organisation im E-Government Umfeld auf Basis V-Modell XT 1.3
  - Aufbau einer *Projektmanagement-Umgebung*
- V-Modell Editor → organisationspezifische Anpassung
  - Initialisierung eines konkreten Projekts



## Spezifische Besonderheiten in der Organisation

### Strukturen / Vorgaben des DIHK (AG):

- E-Government IT-Board
- XIHK Datenkonferenz
- Softwareentwicklungsrahmen gemäß SAGA

### Einbindung bestehender Verfahren beim IHK-GfI (AN):

- Forderung: Nahtlose Einbindung vorhandener Prozessinfrastrukturen
  - RUP (Rational Unified Process)
  - ITIL (IT Infrastructure Library)
- Bei IHK-GfI im **IT-Prozessmanagement Handbuch** festgelegt:
  - Softwareentwicklungsprozesse gemäß RUP
  - Betriebs- und Wartungsprozesse gemäß ITIL
- Einführungsszenario:
  - Integration in das Projektmanagement der IHK-GfI anhand des Vorgehensbausteins *Einführung und Pflege eines organisationsspezifischen Vorgehensmodells*

## Spezifische Modifikationen für das IHK Vorgehensmodell

### Anpassung der Projektdurchführungsstrategie

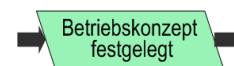
Alle Projektdurchführungsstrategien (PDS) des AN weisen folgende Gemeinsamkeiten auf:

- Entwicklungsstrategie gemäß **RUP** (Mikro-Planungsebene)



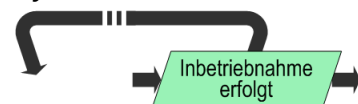
- Neuer Vorgehensbaustein **Betriebskonzept**

- Basis: ITIL-Phase „Service Design“
- Entscheidungspunkt **Betriebskonzept festgelegt**
- Übergang Makro-Mikroplanungsebene vorgesehen

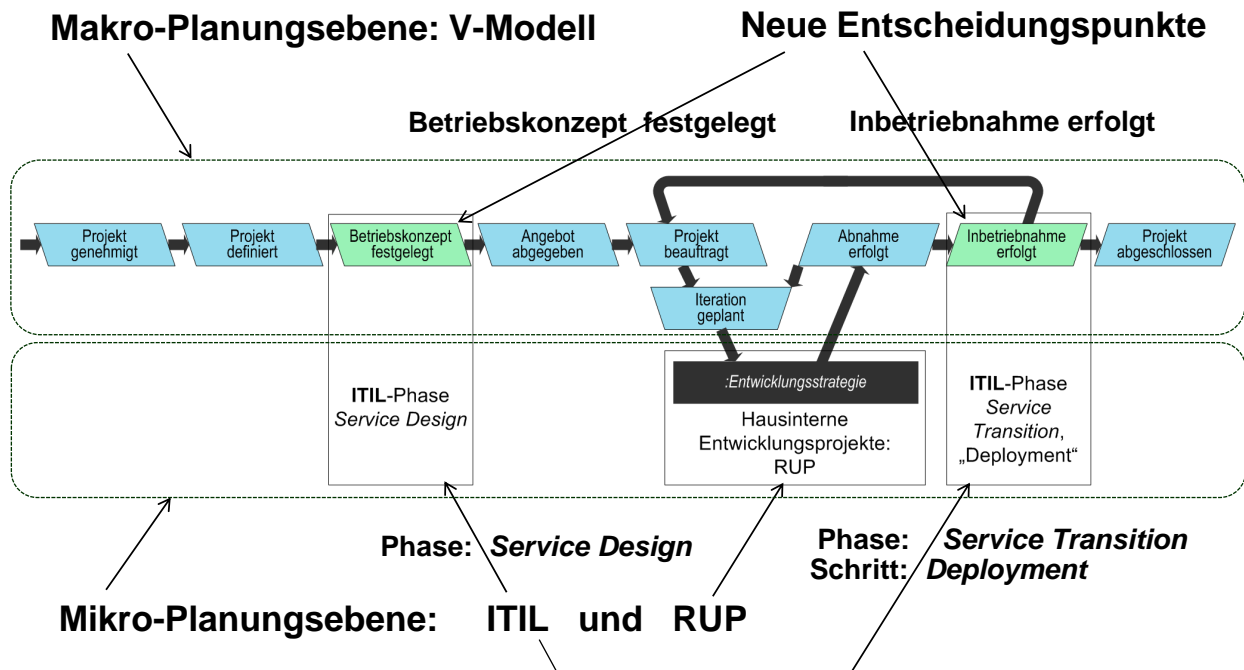


- Neuer Vorgehensbaustein **Inbetriebnahme**

- Basis: ITIL-Phase „Service Transition“, Schritt „Deployment“
- Entscheidungspunkt **Inbetriebnahme erfolgt**
- Übergang Makro-Mikroplanungsebene vorgesehen



## Spezifische Modifikationen für das IHK Vorgehensmodell: Bsp. Projekttypvariante AN-Projekt mit inkrementeller Entwicklung



## AN-Projekte

### Projekttypvariante: AN-Projekt mit komponentenbasierter Entwicklung

- Erstellung Infrastruktur-unterstützender Komponenten
  - Betrieb der Infrastruktur-Basis für E-Government der IHKs bei IHK-GfI: ESB (Enterprise Service Bus) und Enterprise Service Hub
  - Komponentenerstellung bzw. -einbindung im Rahmen des Gesamtauftrags zum Betrieb der E-Government Infrastruktur
  - Technische Komponenten auch ohne explizite Beauftragung durch den AG
  - Anwendung auch für andere Infrastruktur- / Hosting-Plattformen bei IHK-GfI
- Entwicklungsstrategie gemäß RUP
- Erarbeitung Betriebskonzept auf Basis des *Service Design* (ITIL)
- Inbetriebnahme und nachfolgend kontinuierlicher Betrieb im Umfeld bereits vorhandener Komponenten bzw. E-Government Prozesse auf Basis der Phase *Service Transition, Deployment* (ITIL)

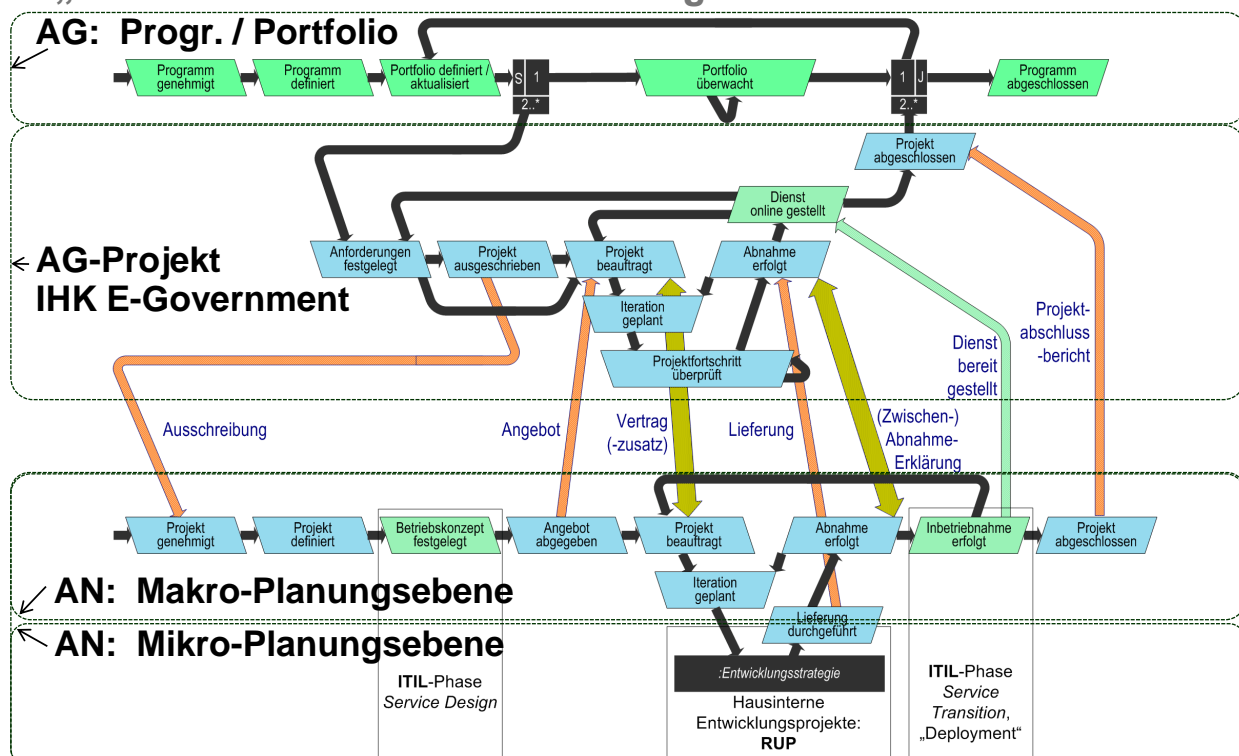
## AN-Projekte

### Weitere Projekttypvarianten beim AN:

Beim AN IHK-GfI werden Projekte auf Basis der folgenden Projekttypvarianten durchgeführt. Nebst Einbindung von RUP und ITIL waren keine weiteren Anpassungen im Modell erforderlich.

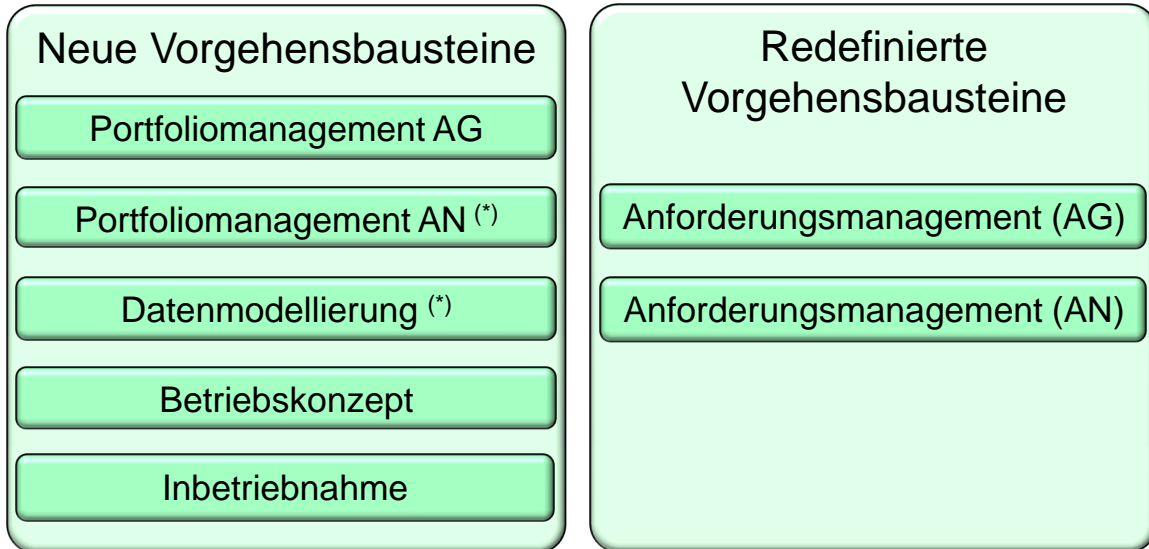
- **AN-Projekt mit inkrementeller Entwicklung**
  - Keine weiteren Anpassungen ggü. Standard V-Modell
- **AG-AN-Projekt mit inkrementeller Entwicklung**
  - Bedarf nach einem *einheitlichen Vorgehensmodell* für alle Projekte bei IHK-GfI, daher *Harmonisierung* mit dem Vorgehensmodell für E-Government Projekte.
- **AG-AN-Projekt mit Beschaffung**
  - Verpflichtende Hinzunahme des Projektmerkmals *Fertigprodukte*
  - Hauptaugenmerk auf Anpassung des zu beschaffenden Fertigprodukts an E-Government Infrastruktur bzw. Hosting-Umgebung der IHK-GfI

### Zusammenwirken von AG-Projekt und AN-Projekt „iterativ-inkrementelle Entwicklung“



## Vorgehensbausteine

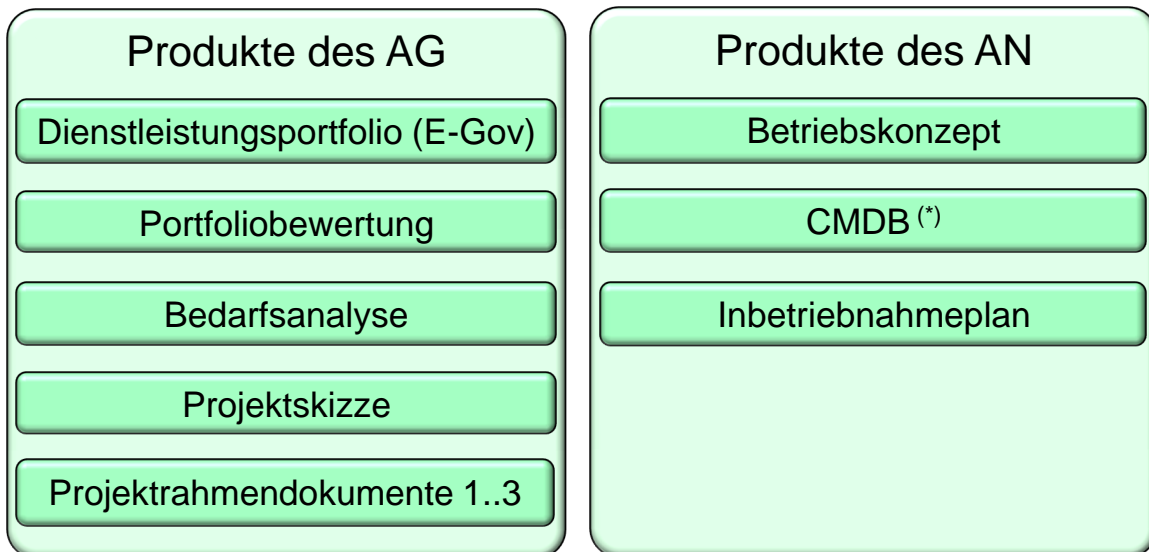
### Anpassung der Vorgehensbausteine



(\*) für spätere Iterationsstufe vorgesehen

## Vorgehensbausteine

### Anpassung der Produkte und Aktivitäten

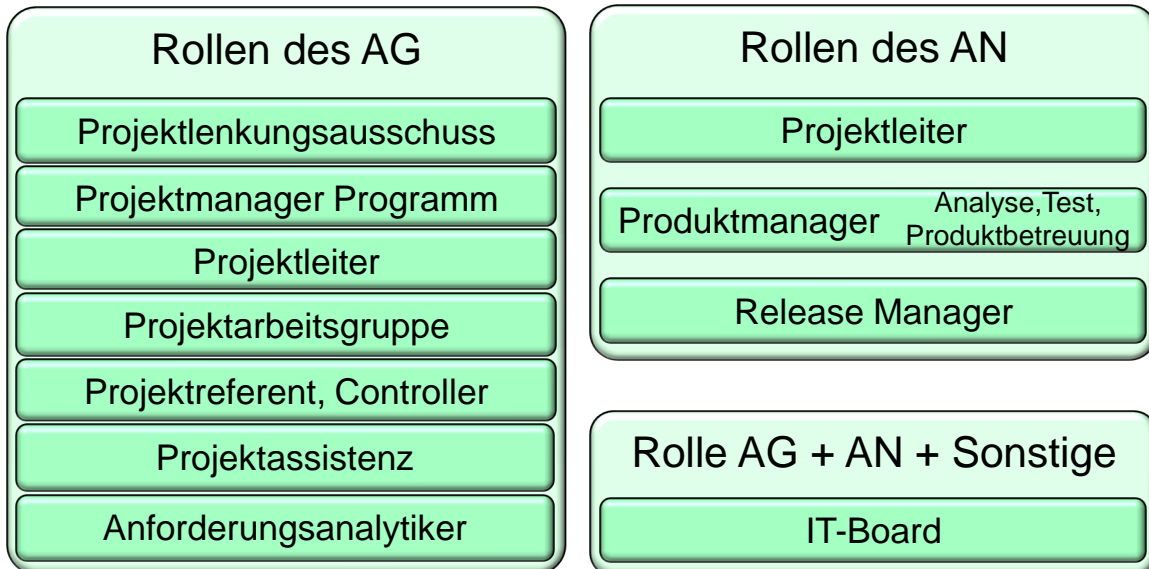


Aktivitäten und Produkte inhaltlich im Verhältnis 1:1 zueinander

(\*) CMDB = Configuration Management Database nach ITIL

## Vorgehensbausteine

### Anpassung der Rollen



Benennung der Rollen gemäß Konvention DIHK (AG) bzw. IHK-GfI (AN)

### Erste Erfahrungen

- Erst die Version V-Modell XT erlaubt eine **individuelle und gezielte** Anpassung als Vorgehensstandard einer Organisation.
- Die Anpassung ist nur mit dem **Einsatz von Werkzeugen** möglich.
- Ein zweistufiges Vorgehen **organisationsspezifische** Anpassung und **projektspezifische** Anpassung ist praxisgerecht
  - für die **organisationsspezifische** Modellierung ist ein hohes Abstraktionsvermögen und Expertenwissen zum V-Modell XT und zur Organisation erforderlich,
  - das **projektspezifische** Tailoring kann von Projektleitern ohne Expertenwissen zum V-Modell durchgeführt werden.
- Die Ausgangslage ist günstig, wenn **AG und AN** bei der Entwicklung eines Vorgehensstandards **gemeinsam vorgehen** können.

## Chancen und Risiken

### Chancen ...

- Die Einführung des V-Modells XT als Vorgehensstandard ist eine Voraussetzung für eine **gesteigerte Prozesssicherheit** bei AG wie AN,
- **bewährte Prozess-Standards** brauchen nicht geändert zu werden.

### ... und Risiken

- aufgrund der **Freiheitsgrade** bei den Anpassungen haben die Modellierer eine hohe Verantwortung,
- ohne eine **Einführungsstrategie** bei AG und AN kann die Gefahr aufkommen, dass die Standards nicht gelebt werden.

## Folgeschritte / Ausblick

### Folgeschritte, Ausblick

- Projekttypvarianten der Pilotprojekte im ausgewählten Werkzeug ausmodellieren
- Dokumentenvorlagen kontinuierlich anpassen
- Pilotprojekte kontinuierlich begleiten
- **Akzeptanzmanagement:**  
Einführungsstrategie des IHK Vorgehensmodells entwickeln und weitere Projekte unterstützen

Vielen Dank für Ihre Aufmerksamkeit!





### *7.3 Die Entwicklung eines Vorgehensmodells für E-Government-Projekte*

## 8. Standardisierung mit XÖV

### Sessionüberblick

---

8.1. Standardisierung auf allen Ebenen: XÖV und V-Modell XT in XPolizei-Vorhaben . . .	405
8.2. Ein Musterkonzept zur Wartung und Pflege von XÖV-Standards . . . . .	419
8.3. Interoperabilität durch Standardisierung – Zertifizierte XÖV-Konformität für die Verwaltung . . . . .	439
8.4. Vom Meldewesen zu XÖV – Erfolgsfaktoren der Standardisierung im eGovernment .	455
8.5. Semantische Repositories in Europa . . . . .	469
8.6. Ein Standard für Registeranwendungen: die Register Factory . . . . .	491

---



## Standardisierung auf allen Ebenen: XÖV und V-Modell XT in XPolizei-Vorhaben

Matthias Memmesheimer<sup>1</sup>, Thomas Reiter<sup>1</sup>, Klaus Bergner<sup>2</sup>

<sup>1</sup>L/ZIR / ITD-S, Bundeskriminalamt  
65173 Wiesbaden  
matthias.memmesheimer@bka.bund.de  
thomas.reiter@bka.bund.de

<sup>2</sup>4Soft GmbH, Mittererstraße 3  
80336 München  
klaus.bergner@4soft.de

**Abstract:** Der XÖV-Standard XPolizei definiert Datenaustauschformate zur Integration der polizeilichen IT-Systeme in Deutschland. Ein begleitender Leitfaden regelt die effektive Nutzung in IT-Vorhaben sowie die Pflege und Fortschreibung des Standards. Dazu definiert der Leitfaden aufbauend auf dem Vorgehensmodell-Standard V-Modell XT spezielle XPolizei-relevante Organisationseinheiten, Rollen und Prozesse und standardisiert die zu erarbeitenden Ergebnisse und deren Prüfkriterien. Der Leitfaden ist speziell für öffentliche Auftraggeber und auch Auftragnehmer im XPolizei-Kontext relevant. Darüberhinaus kann er über den XPolizei-Kontext hinaus allgemein als Mustervorgehen für XÖV-Anwenderprojekte dienen.

### 1 Motivation

Das Bundeskriminalamt und die deutsche Polizei insgesamt stehen vor großen Aufgaben: Die polizeilichen IT-Systeme sollen bundesweit integriert und für den Datenaustausch mit anderen Behörden wie z.B. den Meldeämtern ausgebaut werden. Dazu erarbeiten das BKA und die Landeskriminalämter den XML-Datenaustausch-Standard XPolizei [XPÖ] auf Basis des XÖV-Standardisierungsansatzes [XÖV].

Die initiale Erarbeitung des Standards XPolizei basierend auf den Anforderungen der INPOL-Teilnehmer ist dabei nur der erste Schritt: Letztendliches Ziel ist ja, dass der Standard in den relevanten polizeilichen IT-Vorhaben eingesetzt und zudem kontinuierlich weiterentwickelt wird. Hierfür sind Prozessvorgaben, Methoden und Werkzeuge erforderlich, die den Entwicklungs- und Wartungsprozess von polizeilichen IT-Vorhaben hinsichtlich seiner XPolizei-relevanten Aspekte regeln.

## 2 Ansatz

In einem Vorbereitungsprojekt [IMP] ist deshalb ein Leitfaden für XPolizei-Vorhaben entstanden, der diese Regelungen und Inhalte enthält und folgende Themen behandelt:

- Der Leitfaden beschreibt detailliert die beteiligten *Organisationen* und die ihnen zugeordneten Verantwortlichkeiten und *Rollen*. Aufgrund der Größe des Gesamtvorhabens und des föderalen Aufbaus der Polizei ergibt sich eine Struktur aus drei Ebenen mit einer zentralen Informationsmodell-Redaktion (ZIR) zur Pflege und Fortschreibung des Standards auf der obersten Ebene sowie einer Zwischenebene aus lokalen Koordinationsstellen in den Ländern, welche die eigentlichen IT-Vorhaben auf der untersten Ebene koordinieren, begleiten und prüfen.
- Da nicht jedes polizeiliche IT-Vorhaben XPolizei-relevant ist und unterschiedliche Vorhaben unterschiedliche Anforderungen haben, unterscheidet der Leitfaden analog zu den Projekttypen des V-Modell XT [VMX] zwischen vier *Vorhabentypen* mit jeweils abgestuften Regelungen für das anzuwendende Vorgehen und die durchzuführenden Prüfungen.
- *Prozesse* regeln, wie bei der Neuentwicklung polizeilicher IT-Verfahren im Rahmen von Projekten bzw. während des Betriebs von Bestandsverfahren vorgegangen werden muss, um die XPolizei-Konformität zu gewährleisten. Dazu gibt der Leitfaden Ergebnisse und Kriterien an, die bei bestimmten Meilensteinen erarbeitet bzw. erfüllt sein müssen, und stellt die zugehörigen Vorlagen und Checklisten zur Verfügung. Die entsprechenden Prozesse lehnen sich eng an das V-Modell XT an, gestatten aber im Prinzip auch die Anwendung anderer Vorgehensmodelle.
- Eine kurze Beschreibung der *Infrastruktur* aus Werkzeugen und Basis-Systemen zeigt IT-Vorhaben, auf welche Hilfsmittel sie zurückgreifen können.

Zielgruppe des Leitfadens sind hauptsächlich die Beteiligten bei der ZIR und den lokalen Koordinationsstellen in den Ländern, andererseits die konkreten IT-Vorhaben. Bei letzteren sind wiederum zwei Ebenen relevant:

- *Auftraggeberprojekte beim öffentlichen Auftraggeber* kommunizieren mit den lokalen Koordinationsstellen und der ZIR. Auftraggeberprojekte liefern XPolizei-relevante Ausschnitte aus Ergebnissen wie Projekthandbuch oder Lastenheft sowie die in den entsprechenden Auftragnehmerprojekten erarbeiteten XML-Austauschformate an die lokalen Koordinationsstellen, welche sie dann auf XPolizei-Konformität prüfen.
- *Auftragnehmerprojekte bei Auftragnehmern der öffentlichen Hand* sind zwar nicht direkt von den Vorgaben des Leitfadens betroffen, indirekt aber sehr wohl, da sie von ihren jeweiligen Auftraggeberprojekten zur XPolizei-Konformität verpflichtet und daraufhin überwacht werden. Kenntnis und Verständnis der betreffenden Regelungen sind also auch für sie ratsam.

Zusammen bilden die Inhalte des Leitfadens ein kleines Vorgehensmodell (oder eher

„Vorgehens-Framework“), das sich ausschließlich auf die XPolizei-relevanten Aspekte von polizeilichen IT-Vorhaben beschränkt. Die restlichen Aspekte werden durch existierende Vorgehensmodelle abgedeckt. Insbesondere wurde auf eine nahtlose Integration in das V-Modell XT bzw. das parallel entstehende, organisationspezifische V-Modell XT Bund [VMB] geachtet.

### 3 Bewertung

Der Leitfaden zeigt anhand des XPolizei-Standards beispielhaft eine Organisationsstruktur und ein Vorgehensmodell zur Verankerung eines übergreifenden Datenmodells und der zugehörigen Austauschformate in IT-Vorhaben. Da es sich beim XPolizei-Standard um einen der komplexesten XÖV-Standards handelt, lässt sich erwarten, dass die Ergebnisse auch für die XÖV-Standardisierung im allgemeinen anwendbar sind. Ggf. können auch Change-Programme in größeren Unternehmen Anleihen bei den vorgestellten Strukturen und Prozessen nehmen.

Auf Basis des erarbeiteten Leitfadens sind weitere Schritte möglich. Beispielsweise scheint es möglich, den Leitfaden oder Teile davon in Form eines Vorgehensbausteins für das V-Modell XT oder das V-Modell XT Bund zur Verfügung zu stellen. Dadurch würden polizeiliche IT-Projekte ein übergreifendes Vorgehensmodell erhalten, in dem die XPolizei-relevanten Aspekte nahtlos in das zugrundeliegende Vorgehensmodell integriert sind.

Denkbar wäre in diesem Rahmen auch eine Trennung der allgemeineren XÖV-Aspekte von den spezielleren XPolizei-Aspekten. Ergebnis wären dann zwei Vorgehensbausteine: Ein XÖV-Vorgehensbaustein mit den Regelungen für XÖV-Anwenderprojekte und aufbauend darauf ein XPolizei-Vorgehensbaustein mit zusätzlichen Regelungen für XPolizei-relevante polizeiliche IT-Vorhaben.

### Literaturverzeichnis

- [IMP] Andreas Rausch, Edward Fischer (TU Clausthal), Klaus Bergner (4Soft GmbH): *abschlussbericht EkoBKA – Einführung des V-Modell XT am Beispiel des Projekts IMP*, 2009.
- [VMX] V-Modell XT V1.3, verfügbar über [www.v-modell-xt.de](http://www.v-modell-xt.de), 2009.
- [XPÖ] XÖV-Projektsteckbrief zur Standardisierung von XPolizei, Website von Deutschland Online, [http://www.standardisierung.deutschland-online.de/Standardisierung\\_Internet/broker.jsp?uMen=7ad30f32-d664-ba11-d88e-f1ac0c2f214a](http://www.standardisierung.deutschland-online.de/Standardisierung_Internet/broker.jsp?uMen=7ad30f32-d664-ba11-d88e-f1ac0c2f214a)
- [XÖV] Deutschland Online, [www.standardisierung.deutschland-online.de](http://www.standardisierung.deutschland-online.de), 2009.



## Standardisierung auf allen Ebenen

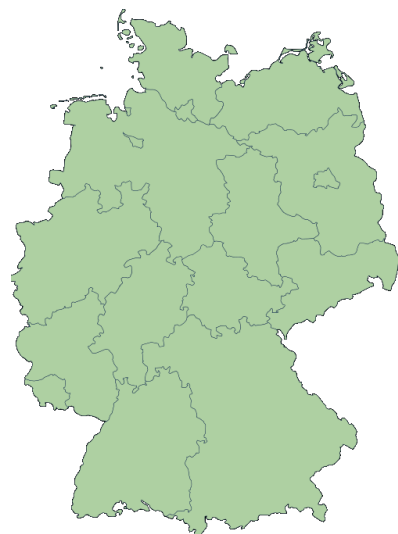
### XÖV und V-Modell XT in XPolizei-Vorhaben

Thomas Reiter, Bundeskriminalamt  
Klaus Bergner, 4Soft GmbH

Software & Systems Engineering Essentials  
4. Mai 2010, Köln

## Strategische Ziele von INPOL

- Unterstützung der polizeilichen Zusammenarbeit
- Unterstützung der polizeilichen Erkenntnisgewinnung
- Flexibilität für neue bzw. sich ändernde Sicherheitslagen
- Einmalerfassung und Mehrfachnutzung
- ➔ Integration der polizeilichen IT-Systeme der INPOL-Teilnehmer: BKA, Zoll, Bundespolizei, LKA, Landespolizeien



## Ziele der XPolizei-Standardisierung



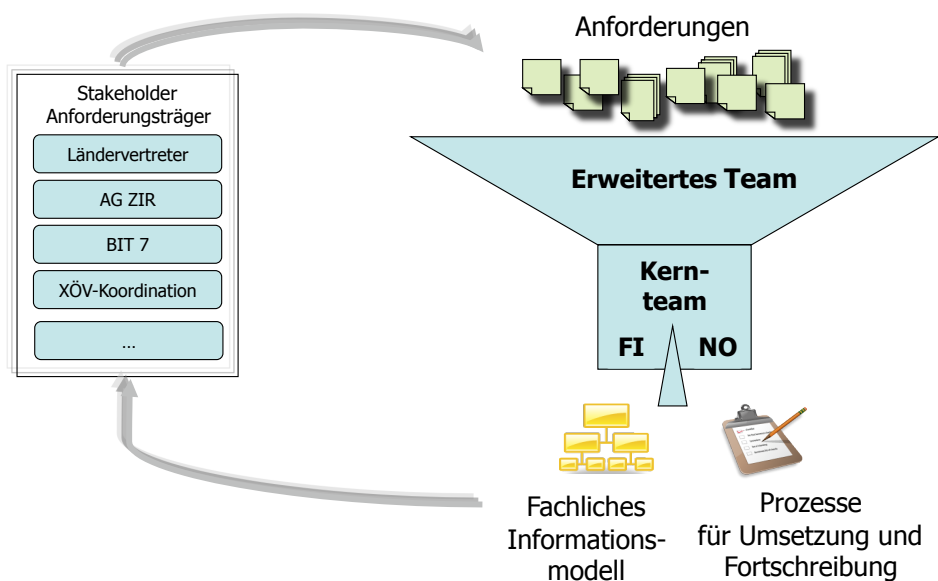
Standardisierte polizeiliche Daten und Prozesse

- **Informationsmodell Polizei** als Grundlage für polizeiliche IT-Systeme
  - UML-Datenmodell als fachliche Basis
  - XML-Formate für Austauschdaten
- **XPolizei-Standard** konform zu XÖV-Standard-Richtlinien

Standardkonforme Umsetzung in polizeilichen Vorhaben

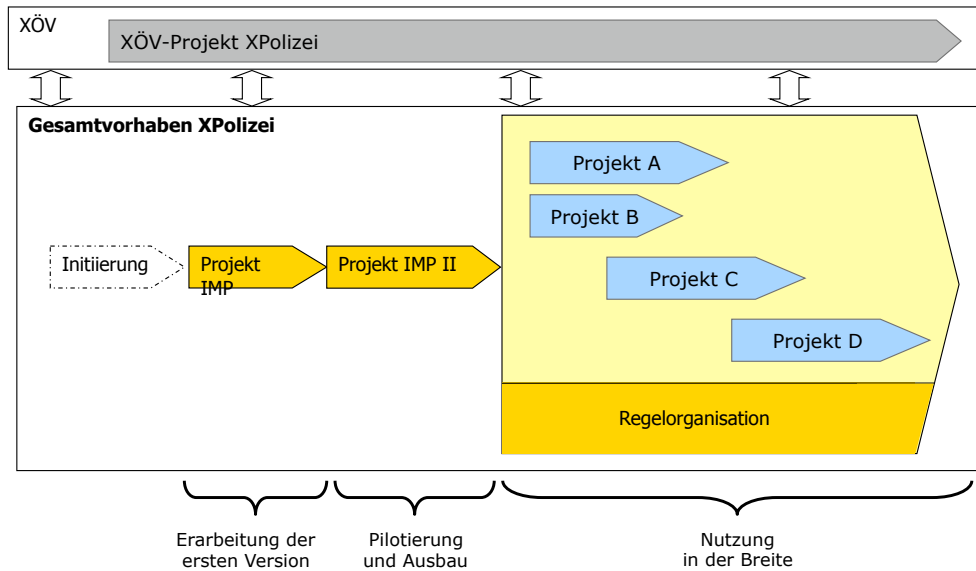
- **Prozessvorgaben** für die Erarbeitung polizeilicher IT-Verfahren
  - Neuentwicklung von IT-Verfahren
  - Weiterentwicklung von Bestandsverfahren
- **XPolizei-Prozessleitfaden** kompatibel mit dem Standard V-Modell XT

## Vorgehen bei der Standardisierung





## Status und weitere Planung

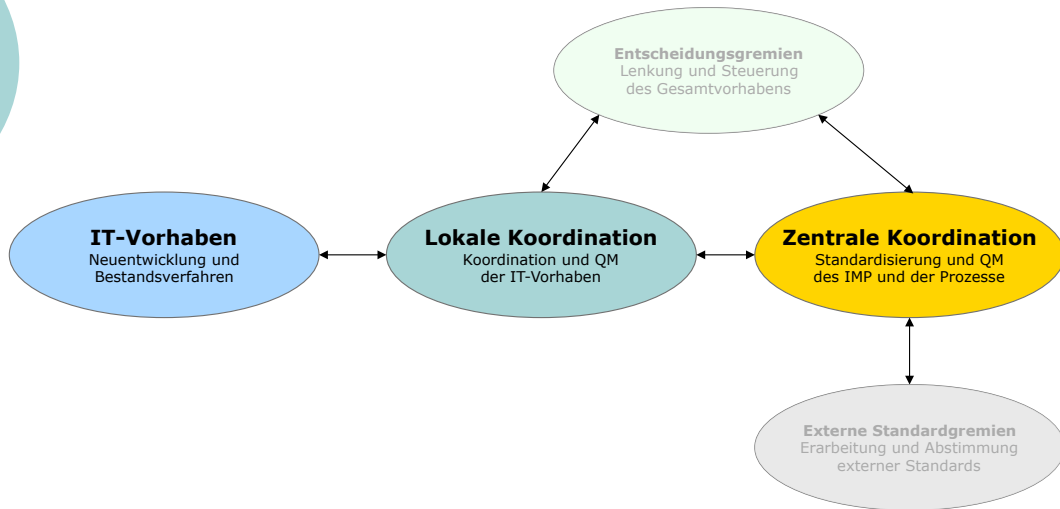


## Ausgangssituation und Anforderungen

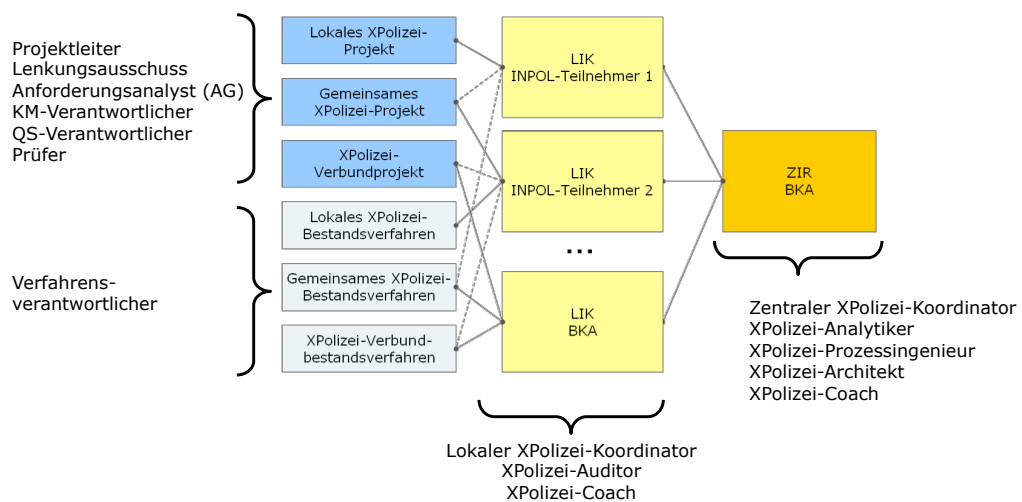
- Polizei ist föderal organisiert
  - BKA, BPol, Zoll im Bund
  - LKA als Zentralen in Ländern
  - Landespolizeien in der Fläche
- Vielzahl unterschiedlicher IT-Verfahren
  - Große/kleine, zentrale/lokale IT-Systeme
  - Unterschiedliche Vorgaben für IT-Entwicklung
    - V-Modell XT als Bundesstandard
    - Lokale Vorgaben der Länder und Behörden



## Grundsätzliche Aufgaben

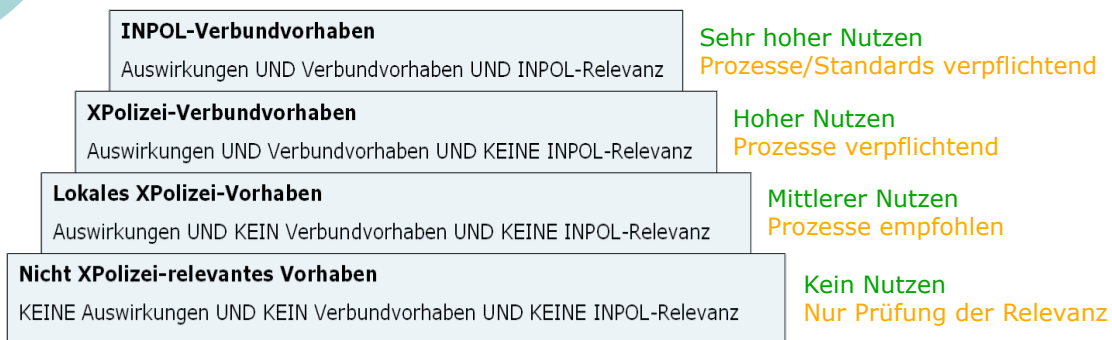


## Organisation und Rollen



## Vorhabentypen

- Je mehr Vorhabensmerkmale erfüllt sind
  - desto höher ist der Nutzen der XPolizei-Konformität
  - desto stärker sind die Vorgaben an das Vorhaben
- Hochstufung von Vorhaben auf Antrag oder durch LIK



## Nicht XPolizei-relevantes Vorhaben

- Kein Vorhabensmerkmal erfüllt
  - Kein Datenaustausch, keine Auswirkungen auf Standards, kein wirtschaftlicher Vorteil, kein Pilotprojekt oder begrenzte Lebensdauer
  - Beispiele: kurzfristig erforderliches Auswertesystem, System begrenzter Lebensdauer, System ohne Datenaustausch
- Erforderliche Maßnahmen
  - Nur Prüfung auf XPolizei-Relevanz
  - Im Normalfall **keine Anwendung des Frameworks**
- Kosten/Nutzen
  - Marginaler Aufwand für Prüfung durch LIK
  - Keine Zusatzkosten im Projekt



## Lokales XPolizei-Vorhaben

---

- Vorhabensmerkmal *Auswirkungen* erfüllt
  - Datenaustausch, Auswirkungen auf Standards, wirtschaftlicher Vorteil oder Pilotprojekt, keine begrenzte Lebensdauer
  - Beispiele: Systeme bei einzelnen INPOL-Teilnehmern
- Erforderliche Maßnahmen
  - Prüfung auf XPolizei-Relevanz
  - **Empfehlung zur Einhaltung des Frameworks**
    - Lieferung der definierten Ergebnisse durch das Vorhaben
    - Prüfung der XPolizei-Konformität durch LIK
    - Besetzung der Rollen
- Kosten/Nutzen
  - Aufwände durch Erarbeitung und Prüfung der Ergebnisse
  - Vorhaben „auf dem Radar“, Querbeziehungen werden sichtbar
  - Prüfungen erhöhen Qualität, Integrationsfähigkeit, Wiederverwendbarkeit sowie Kompatibilität der Schnittstellen und Formate



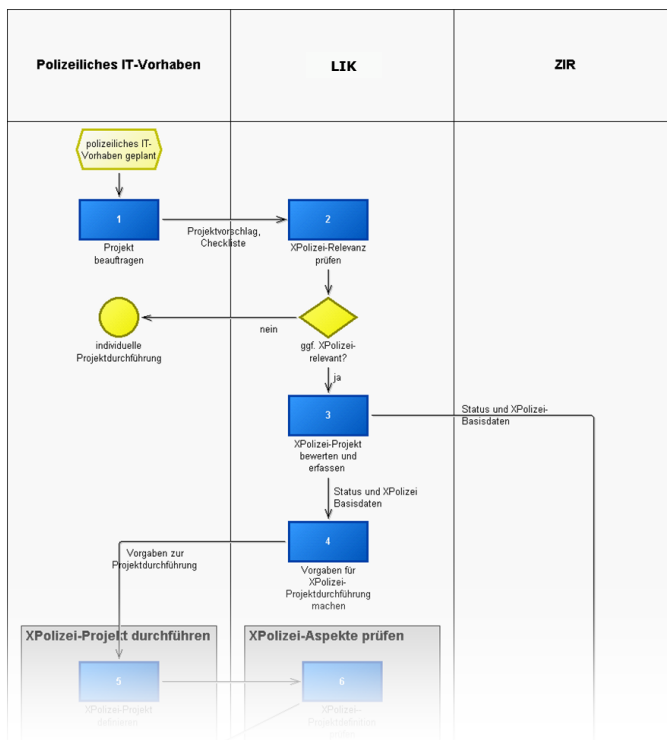
## XPolizei-Verbundvorhaben

---

- Vorhabensmerkmale *Auswirkungen* und *Verbundvorhaben* erfüllt
  - Wie bei lokalem XPolizei-Vorhaben, zusätzlich aber mehrere INPOL-Teilnehmer oder mindestens ein externer Teilnehmer beteiligt
- Erforderliche Maßnahmen
  - Prüfung auf XPolizei-Relevanz
  - **Verpflichtung zur Einhaltung des Frameworks**
    - Lieferung der definierten Ergebnisse durch das Vorhaben
    - Prüfung der XPolizei-Konformität durch LIK
    - Besetzung der Rollen
  - Federführende LIK wird bestimmt
- Kosten/Nutzen
  - Analog zum lokalen XPolizei-Vorhaben
  - Zusätzlich Aufwände für Koordination zwischen beteiligten LIK

## INPOL-Verbundvorhaben

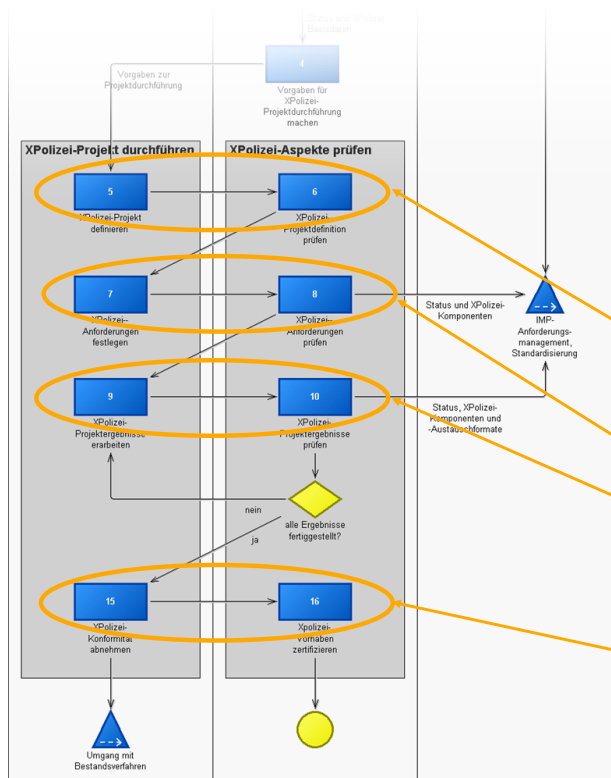
- Vorhabensmerkmale *Auswirkungen* und *Verbundvorhaben* erfüllt
  - Wie bei XPolizei-Verbundvorhaben, jedoch ist ein INPOL-Vorhaben betroffen
- Erforderliche Maßnahmen
  - Prüfung auf XPolizei-Relevanz
  - **Verpflichtung zur Einhaltung des Frameworks**
    - Lieferung der definierten Ergebnisse durch das Vorhaben
    - Prüfung der XPolizei-Konformität durch LIK
    - Besetzung der Rollen
  - Verpflichtung zur Einhaltung zusätzlicher Standards
    - V-Modell XT, UML-Modellierung mit standardkonformem Werkzeug, begleitende Prozessvorgaben, Checklisten und Vorlagen des Frameworks
  - Als federführende LIK wird die LIK der BKA bestimmt
- Kosten/Nutzen
  - Analog zum XPolizei-Verbundvorhaben
  - Kosten/Nutzen durch Nutzung zusätzlicher Standards



### Vorbereitung von Projekten

- Durchführung bis zur Beauftragung des Projekts
- Schritte zur
  - Bewertung der XPolizei-Relevanz
  - Erfassung des Projekts im XPolizei-Repository
- LIK regelt Gestaltung des Projekts

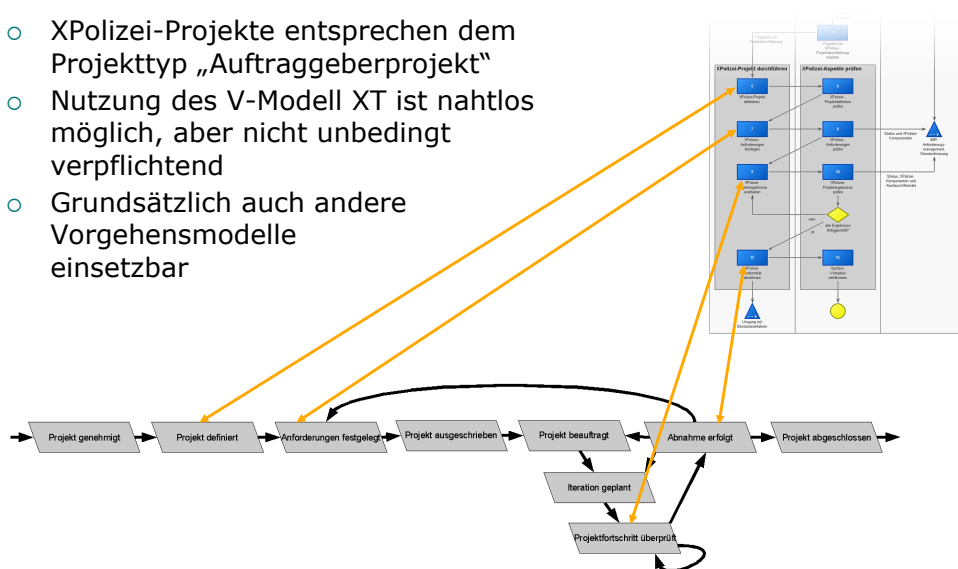
## Durchführung von Projekten

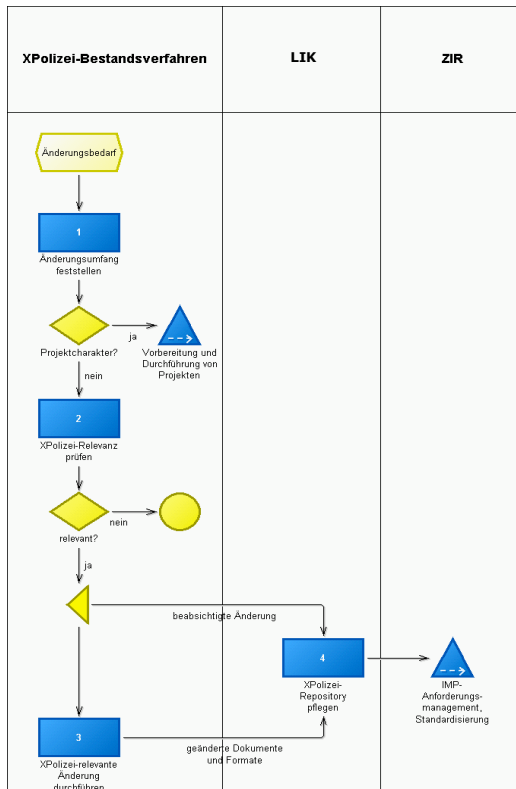


- Dualität aus
  - Erarbeitung durch das Projekt
  - Prüfung durch die LIK
- Haupt-Ergebnisse
  - Projekthandbuch, Projektplan, QS-Handbuch, Werkzeug-Infrastruktur
  - Lastenheft
  - Pflichtenheft, XML-Austauschformate, Datenmodell, Betriebsdokumente
  - XPolizei-Zertifikat

## Beziehung zum V-Modell XT

- XPolizei-Projekte entsprechen dem Projekttyp „Auftraggeberprojekt“
- Nutzung des V-Modell XT ist nahtlos möglich, aber nicht unbedingt verpflichtend
- Grundsätzlich auch andere Vorgehensmodelle einsetzbar





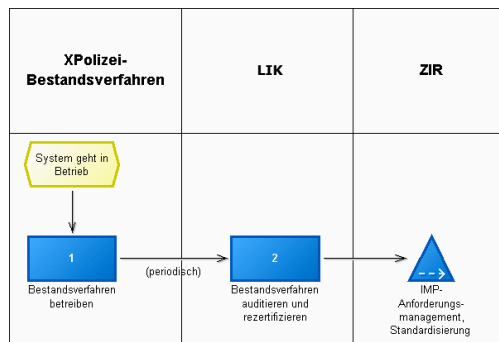
## Änderungsmanagement

- Umgang mit Änderungen von Bestandsverfahren
  - Typischerweise fallen sehr viele Änderungen an
  - Größere Änderungen erfolgen in Projekten
- Leichtgewichtiges Verfahren
  - Weitergabe XPolizei-relevanter Änderungen durch Verfahrensverantwortlichen
  - LIK gibt relevante Änderungen in XPolizei-Repository ein



## Periodische Rezertifizierung

- Periodische Überprüfung von Bestandsverfahren nach Maßgabe der zuständigen LIK
- Zu prüfende Ergebnisse und Prüfkriterien analog zu XPolizei-Projekten





## Infrastruktur

---

- System zum verteilten Anforderungsmanagement
- XPolizei-Repository
  - Bereitgestellt über System Track+
- IMP-Modellverwaltung
  - Zentrales Modell-Repository bei der ZIR
- Publikationsmedium
  - Veröffentlichung der Standards auf INPOL-Infoserver
- UML-Modellierungswerkzeuge und Generatoren
  - Durch ZIR insbesondere für INPOL-Verbundvorhaben definiert



## Ausblick

---

- Pilotierung und Weiterentwicklung des Frameworks in IMP2
- Weitere Ausarbeitung der Arbeitshilfen
  - Vorlagen, Mustertexte, Checklisten
- Möglichkeit: Vorgehensbausteine bereitstellen
  - Vorgehensbausteine für V-Modell XT Bund
    - XÖV-Projektdurchführung
    - XPolizei-Projektdurchführung
  - **Vorteil für XPolizei:** XPolizei-Vorhaben nutzen bestehende Inhalte, Arbeitshilfen und Werkzeuge
  - **Vorteil für BVA/BIT:** XÖV-Projekte können auf Vorarbeiten aus XPolizei-Kontext zurückgreifen



## *8.1 Standardisierung auf allen Ebenen: XÖV und V-Modell XT in XPolizei-Vorhaben*

## Ein Musterkonzept zur Wartung und Pflege von XÖV-Standards

Jan-Peter Baas

Abteilung  
BearingPoint GmbH, Speicherstraße 1  
60327 Frankfurt am Main  
jan-peter.baas@bearingpointconsulting.com

**Abstract:** Das XÖV-Handbuch und seine Konformitätskriterien sind in Zukunft der zentrale Mechanismus zur Steuerung der Standardisierung im XÖV-Umfeld. Während die Entwicklungsphase eines XÖV-Standards durch das XÖV-Handbuch heute schon gut abgedeckt ist, müssen andere Lebenszyklusphasen noch stärker mit Konformitätskriterien ausgestaltet werden. Eine der wichtigsten Phasen ist die Wartung von XÖV-Standards. Als Ergänzung zu den Konformitätskriterien des XÖV-Handbuchs stellt der Vortrag den Ansatz eines Musterwartungskonzeptes vor. Der aktuelle Stand der Erfahrungen bei der Wartung einzelner XÖV-Standards wird dargestellt und daraus der aktuelle Diskussionsstand zu zukünftigen Konformitätskriterien für XÖV-Standards abgeleitet.

### 1 Motivation

XÖV steht für XML in der öffentlichen Verwaltung. Ziel von XÖV ist es, den Datenaustausch innerhalb und mit der öffentlichen Verwaltung zu standardisieren. Hierdurch wird die Interoperabilität verbessert und in der Folge werden elektronische Prozesse einheitlicher und effizienter gestaltet.

Aktuell gibt es mehr als 20 XÖV-Standards. Jeder dieser Standards hat seine eigene Historie und seine eigenen organisatorischen und technischen Besonderheiten. Um den übergreifenden Nutzen von XÖV-Standards zum Beispiel in den Bereichen Wirtschaftlichkeit, Risikominimierung und Qualitätsverbesserung zu erhöhen, wurden mit dem XÖV-Handbuch [XHB] einheitliche Konformitätskriterien für XÖV-Standards eingeführt. Aufgrund der Erfahrungen der unterschiedlichen bereits bestehenden XÖV-Standards sind für die technische Ausgestaltung von XÖV-Standards umfangreiche Kriterien definiert worden. Darüber hinaus gibt es weitere Konformitätskriterien in Form von Bereitstellungs- und Auskunftspflichten. Die Entwicklung und Veröffentlichung eines XÖV-Standards ist dadurch gut beschrieben.

Für andere Phasen des Lebenszyklus eines XÖV-Standards bestehen diese umfangreichen Erfahrungen aktuell noch nicht. Insbesondere die langfristige Wartung und Pflege ist bisher nur in einzelnen XÖV-Standards verbindlich geregelt. Das XÖV-Handbuch fordert für jeden XÖV-Standard ein Betriebs- bzw. Wartungskonzept, welches die Wartung und Pflege des Standards langfristig sicherstellt. Konformitätskriterien vergleichbar

zu den Kriterien zur technischen Modellierung eines XÖV-Standards sind aktuell noch nicht definiert.

Die Wartung und Pflege der XÖV-Standards wird langfristig ein wichtiges Kriterium für den Erfolg von XÖV sein. Nur wenn auch für diese Phase des Lebenszyklus Hilfsmittel angeboten und Konformitätskriterien definiert werden, kann damit ein einheitliches und vergleichbares Qualitätsniveau etabliert werden. Dazu zählt auch die Wartung und Pflege von standardübergreifenden XÖV-Elementen wie zum Beispiel Kernkomponenten und Codelisten.

### **2 Ansatz**

Mit dem XÖV-Handbuch wurde ein Ansatz gewählt, um die Standardisierung durch Konformitätskriterien zu erhöhen. Um die Wartung und Pflege von Standards sicherzustellen, könnten auch hierfür Konformitätskriterien definiert werden. Ein alternativer Ansatz ist die Erstellung eines entsprechenden Musterwartungskonzepts, welches die Anforderungen an die Wartung eines XÖV-Standards festlegt. Der Vorteil dieses Ansatzes ist, dass neben der impliziten Festlegung von Kriterien (zum Beispiel an die Gliederung eines solchen Konzepts) gleichzeitig ein Hilfsmittel bereitgestellt wird, welches dem Verantwortlichen für einen XÖV-Standards die Erstellung eines Wartungskonzepts deutlich erleichtert. Dabei ist es wichtig, dass das Musterkonzept eine Balance zwischen umfassender Standardisierung auf der einen Seite und einer maximalen Flexibilität auf der anderen Seite findet. Das Konzept muss einerseits die individuellen Besonderheiten einzelner XÖV-Standards zulassen, andererseits möglichst umfassende Kriterien festlegen um ein hohes Maß an Wiederverwendung zu erreichen. Ein vollständig generisches Musterkonzept ist aufgrund der unterschiedlichen Situationen in den einzelnen XÖV-Standards nicht möglich. Trotzdem können eine Reihe unterschiedlicher Kriterien in einem solchen Konzept verbindlich geregelt und für andere Kriterien zumindest Anforderungen und Good-Practice Beispiele definiert werden.

Um diese Ziele zu erreichen wurden erste Erfahrungen aus bestehenden Wartungskonzepten einzelner XÖV-Standards analysiert und daraus ein allgemeines Wartungskonzept entwickelt. Dieses Konzept dient auf der einen Seite als Hilfsmittel für die Erstellung von XÖV-Wartungskonzepten, bietet andererseits aber auch die Möglichkeit, als weiteres Konformitätskriterium des XÖV-Handbuchs zu dienen.

### **2 Bewertung**

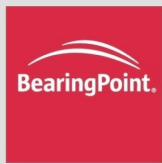
Das XÖV-Handbuch und seine Konformitätskriterien sind in Zukunft ein zentraler Mechanismus zur Steuerung der Standardisierung im XÖV-Umfeld. Während die Entwicklungsphase eines XÖV-Standards heute schon gut abgedeckt wird, müssen andere Lebenszyklusphasen eines XÖV-Standards noch stärker ausgestaltet werden. Eine der wichtigsten Phasen ist dabei die Wartung von XÖV-Standards. Als Ergänzung zu den

Konformitätskriterien des XÖV-Handbuchs wird hierbei der Ansatz eines Musterwartungskonzeptes gewählt.

Der Vortrag stellt den aktuellen Stand der Erfahrungen bei der Wartung einzelner XÖV-Standards dar und leitet daraus den aktuellen Diskussionsstand zu Konformitätskriterien und Musterwartungskonzept ab. Als Ausblick wird auf die standardübergreifende Wartung und Pflege von XÖV eingegangen. Beispielsweise werden Ansätze vorgestellt, wie zukünftig Änderungen an Kernkomponenten, die in unterschiedlichen XÖV-Standards verwendet werden über Standardgrenzen hinweg gepflegt werden können.

### **Literaturverzeichnis**

[XHB] DOL Standardisierung: XÖV-Handbuch (version 1.0 draft), Fassung vom 17. August 2009.



Management  
& Technology  
Consultants

## Ein Musterkonzept zur Wartung und Pflege von XÖV-Standards

Jan-Peter Baas  
Frankfurt, 04.05.2010

Welche Fragen stellen sich (unter anderem) in der Betriebsphase eines XÖV-Standards?

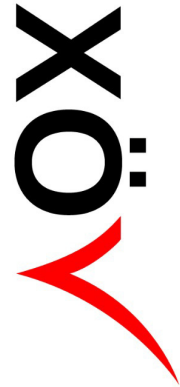


- Wer und wie wird über mögliche Änderungen bzw. Erweiterungen des Standards entschieden?
- Wie werden Pflege und Wartung sowie Erweiterungen eines Standards finanziert?
- Wie werden Bereitstellung, Versionierung und Problemmanagement sichergestellt?
- Wie werden alle Beteiligten über Änderungen informiert?

## Agenda



- Einführung XÖV
  - Begriffsdefinitionen und Anforderungen an den Betrieb eines XÖV-Standards
  - IST- und SOLL-Situation
  - Rollen und Organisation
  - Die Produktvorlage als Musterkonzept
  - Fazit



© 2010 BearingPoint GmbH

3

## Zielsetzung von XÖV



*„XÖV (XML in der öffentlichen Verwaltung) verfolgt das Ziel, die Interoperabilität des elektronischen Datenaustauschs innerhalb und mit der öffentlichen Verwaltung zu verbessern.“*

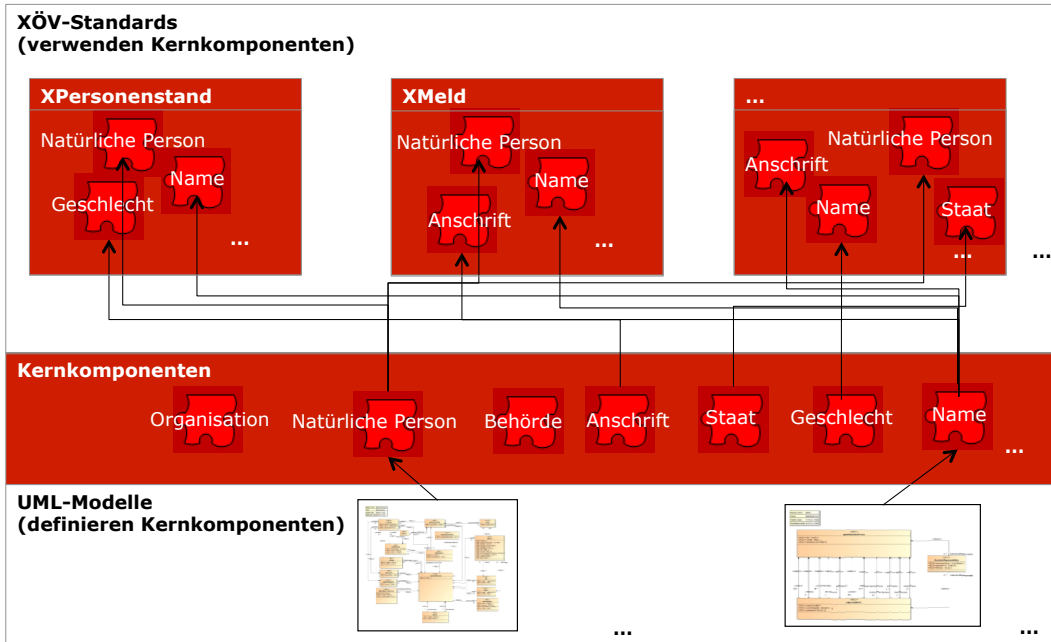
Quelle: XÖV-Handbuch (Version 1.0); Fassung vom 2. März 2010

- Einheitliche Organisation der deutschlandweiten elektronischen Kommunikation zwischen Behörden
- Verfahrensübergreifende Einigung über verbindliche Datenstrukturen
- Verbindliches Vorgehen zur Projektdurchführung
- Verbindliche Regeln zur Dokumentation der XÖV-Standards
- Verbindliches, prüfbares Qualitätsniveau
  
- Übergreifende Festlegungen für Bund, Länder und Kommunen

© 2010 BearingPoint GmbH

4

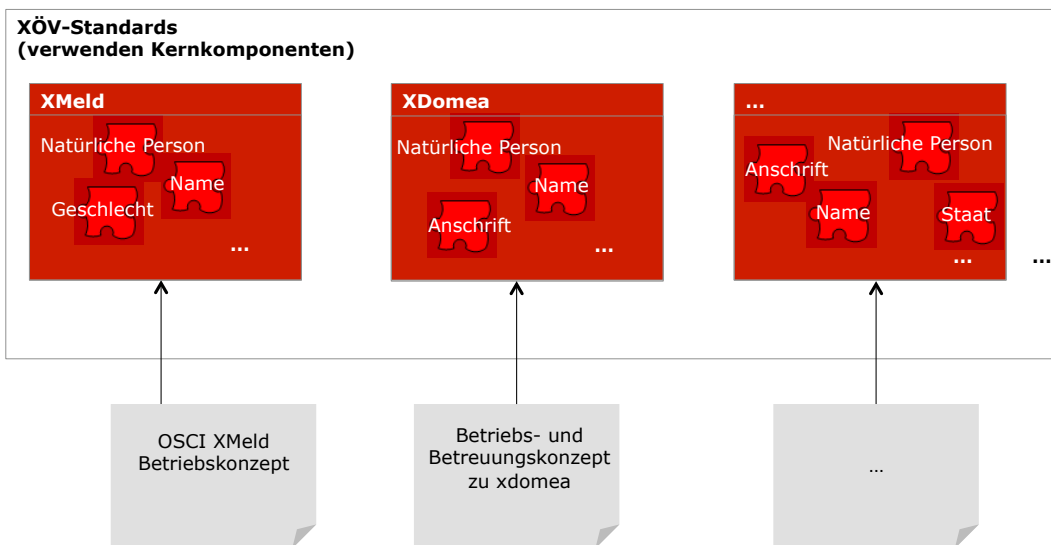
XÖV-Standards werden aus XÖV-Kernkomponenten aufgebaut



© 2010 BearingPoint GmbH

5

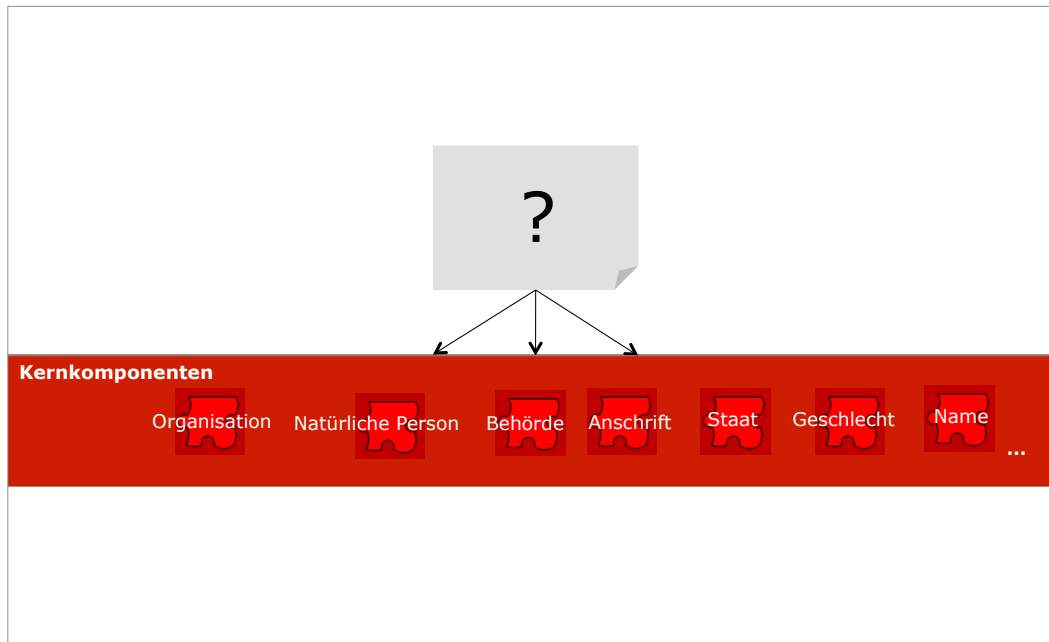
Einige XÖV-Standards haben bereits eigene Konzepte zur Wartung und Pflege entwickelt



© 2010 BearingPoint GmbH

6

Für die Wartung und Pflege der XÖV-Kernkomponenten ist noch kein entsprechendes Konzept veröffentlicht



© 2010 BearingPoint GmbH

7

## Agenda



- Einführung XÖV
- Begriffsdefinitionen und Anforderungen an den Betrieb eines XÖV-Standards
- IST- und SOLL-Situation
- Rollen und Organisation
- Die Produktvorlage als Musterkonzept
- Fazit



© 2010 BearingPoint GmbH

8



## Begriffsdefinitionen (gemäß XÖV-Handbuch)



### XÖV-Standard

„Als XÖV-Standard wird ein Standard bezeichnet, dessen XÖV-Konformität von der XÖV-Koordination festgestellt wurde.“

### XÖV-Projekt

„Als XÖV-Projekt wird die Entwicklungsphase eines XÖV-Standards bezeichnet, es ist Teil eines XÖV-Vorhabens.“

### XÖV-Vorhaben

„Im Rahmen eines XÖV-Vorhabens wird ein XÖV-konformer Standard zunächst entwickelt und dann **betrieben**. Das Vorhaben umfasst damit den gesamten Lebenszyklus des Standards...“

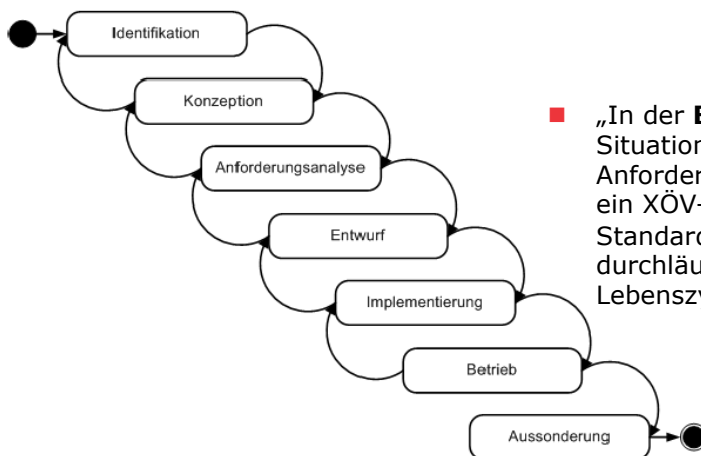
Ein XÖV-Vorhaben umfasst den gesamten Lebenszyklus eines Standards. Ein XÖV-Projekt umfasst die Phasen Anforderungsanalyse, Entwurf und Implementierung eines Standards. Der beschriebene Lebenszyklus legt kein Vorgehensmodell für XÖV-Vorhaben fest.

Quelle: Quelle: XÖV-Handbuch (Version 1.0); Fassung vom 2. März 2010  
© 2010 BearingPoint GmbH

## Der Lebenszyklus von XÖV-Standard sieht eine Betriebsphase vor



- „In der **Betriebsphase** wird der Standard in informationstechnischen Systemen verwendet. Weiterhin wird in dieser Phase die dauerhafte Verwendbarkeit des Standards durch Prozesse zum Problem-, Änderungs- und Konfigurationsmanagement sichergestellt. „



- „In der **Betriebsphase** kann es zu Situationen kommen, in denen sich die Anforderungen an den Standard ändern und ein XÖV-Projekt zur Weiterentwicklung des Standards notwendig wird. Der Standard durchläuft dann die vorhergehenden Lebenszyklusphasen erneut.“

## Forderungen des XÖV-Handbuchs zu Pflege und Betrieb von XÖV-Standards



### K-4 (MUSS) Veröffentlichung



„Der Standard muss im XRepository mit seiner Dokumentation als pdf-Datei, seinen XML-Schema-Dateien, einer XMI-Repräsentation seines XÖV-UML-Modells sowie seinem **Pflegekonzept** veröffentlicht sein.“

### K-5 (MUSS) Nachhaltigkeit des Standards

„Für den XÖV-Standard muss ein **Pflegekonzept** vorliegen, aus dem erkennbar ist, dass ein **langfristige Wartung und Fortschreibung** gewährleistet wird.“

### NDR-30 (MUSS): Versionierung der Schemata

„ Jedes Schema eines XÖV-Standards muss versioniert sein.“

„... Versionen müssen Entwicklungsstände eines Schemas bzw. des XÖV-Standards eindeutig identifizieren. Dieses Vorgehen ist unter anderem **für den Versionswechsel im Betrieb** unentbehrlich.

© 2010 BearingPoint GmbH

11

## Begriffsdefinition: Betrieb



### Betrieb

„Nach der Systemeinführung beginnt der Systembetrieb, d.h. die Nutzung des Anwendungssystems, die den größten zeitlichen Anteil am Softwarelebenszyklus besitzt...“

*Quelle: Einführung in die Wirtschaftsinformatik; Peter Stahlknecht; Ulrich Hasenkamp*

## Betrieb eines Standards ???

- Betrieb der notwendigen Infrastruktur ?
- Betrieb des XRepositories ?
- Betrieb der Fachanwendungen, die den Standard nutzen ?

© 2010 BearingPoint GmbH

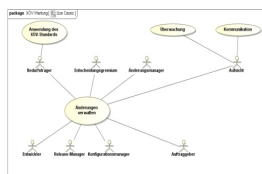
12

Wie definieren wir es?

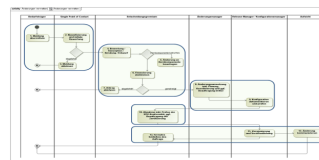


### Pflege- und Wartungskonzept für XÖV-Standards

- Beseitigung von Fehlern, Mängeln und Problemen
- Weiterentwicklung des XÖV-Standards
- Pflege und Wartung von Software und IT-Infrastruktur, sofern sie der Pflege und Wartung des XÖV-Standards dient (z.B. Software für das Änderungsmanagement)
- Ausrollen von neuen Releases



Rollen und Aufgaben



Prozesse

Verantwortlichkeiten (RACI)

© 2010 BearingPoint GmbH

13

### Agenda



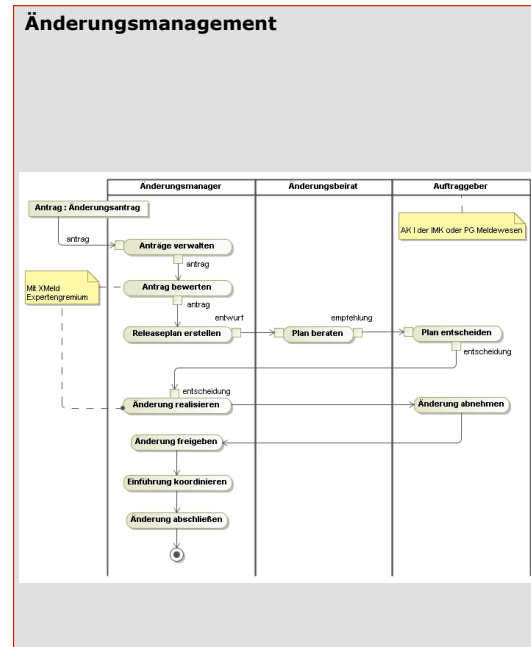
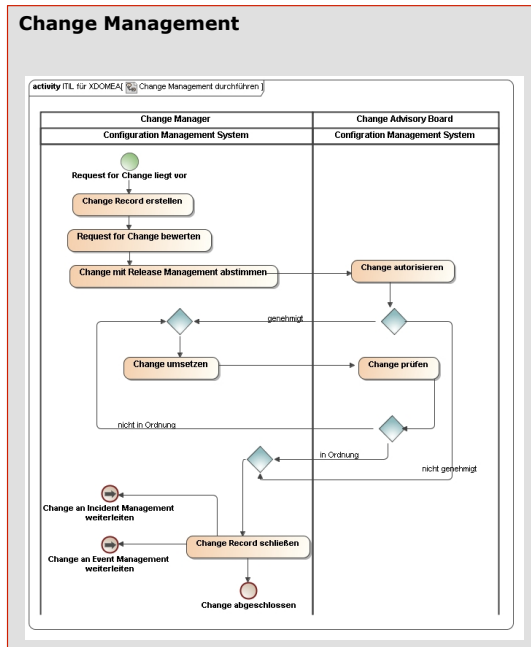
- Einführung XÖV
- Begriffsdefinitionen und Anforderungen an den Betrieb eines XÖV-Standards
- IST- und SOLL-Situation
- Rollen und Organisation
- Die Produktvorlage als Musterkonzept
- Fazit



© 2010 BearingPoint GmbH

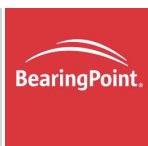
14

Bereits vorhandene Konzepte ähneln sich in der Darstellung der Prozesse



© 2010 BearingPoint GmbH

Beispiele aus den Inhaltsverzeichnissen zeigen, dass die Inhalte bestehender Konzepte nur schwer vergleichbar sind

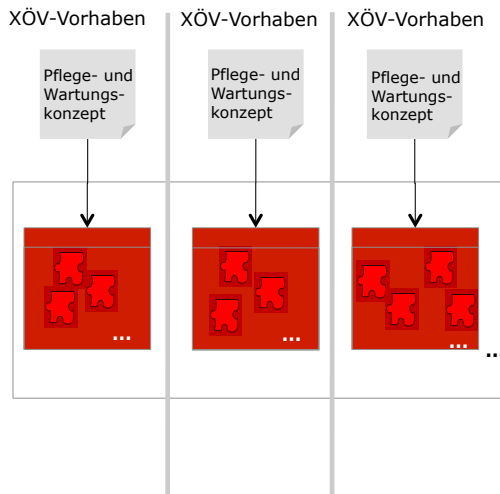


- Change Management**
- Überblick
  - Aktivität: Change Record erstellen
  - Aktivität: Request for Change bewerten
  - Aktivität: Change mit Release Management abstimmen
  - Aktivität: Change autorisieren
  - Aktivität: Change umsetzen
  - Aktivität: Change prüfen
  - Aktivität: Change Record schließen
  - Organisation

- Änderungsmanagement**
- Grundsätze
    - Keine Änderung ohne Auftrag
    - ...
  - Gegenstand von Änderungen
  - Erweiterungen des Standards
  - Änderungsantrag
  - Prozesse im Änderungsmanagement
    - Änderungen verwalten
    - ...
  - Rollen und Verantwortungen im Änderungsmanagement
    - Änderungsmanager
    - ...
  - Arbeiten mit Änderungsanträgen im Rahmen von Wartung und Pflege
  - ...

© 2010 BearingPoint GmbH

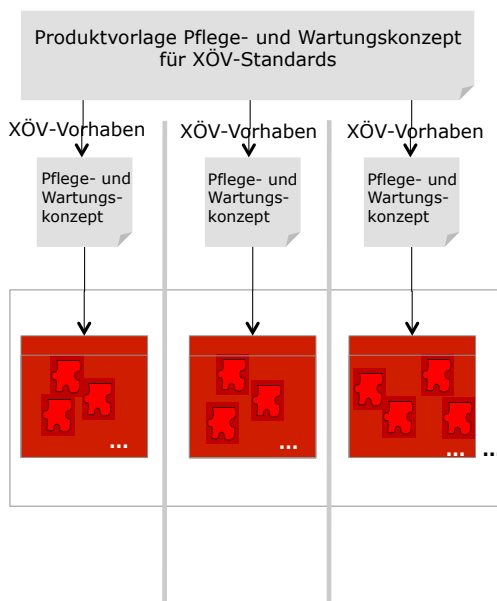
Die IST-Situation zeigt, dass die Pflege und Wartung von XÖV-Standards aktuell nicht standardisiert ist



- Einige XÖV-Standards (z.B. XMeld) haben bereits Pflege- und Wartungskonzepte
- Andere XÖV-Standards müssen diese in Zukunft entwickeln
- Ohne eine Koordination dieser Aufgaben, besteht das Risiko von mehrfacher Arbeiten und ggf. sogar inkompatibler Festlegungen für die Pflege und Wartung der Standards.
  - Nicht vergleichbare Konzepte
  - Keine einheitliche Struktur
  - Keine einheitlichen Inhalte
  - Keine einheitlichen Qualitätskriterien

➔ Keine Standardisierung

Die Betriebsphase als längste Phase im Lebenszyklus sollte durch entsprechende Werkzeuge (Produktvorlagen) standardisiert werden

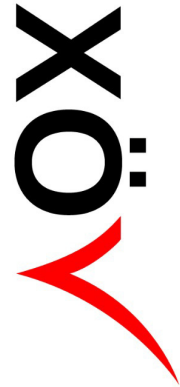


- XÖV-Standards nutzen eine einheitliche Produktvorlage für Pflege- und Wartungskonzepte
- Die Konzepte haben damit eine standardisierte Struktur und festgelegte Mindestinhalte
- Prozesse werden gemeinsam genutzt
- Checklisten und Hilfestellungen der Produktvorlage erleichtern in allen Vorhaben die Erstellung des Konzepts
- Auf aus Erfahrung bekannte Fehler und Probleme wird in der Vorlage hingewiesen
- Die Vorlage bietet eine hohe Flexibilität um vorhabenspezifische Inhalte und Regelungen innerhalb der gegebenen Struktur abzubilden
- Die Konzepte werden im XÖV-Repository veröffentlicht

## Agenda



- Einführung XÖV
- Begriffsdefinitionen und Anforderungen an den Betrieb eines XÖV-Standards
- IST- und SOLL-Situation
- Rollen und Organisation
- Die Produktvorlage als Musterkonzept
- Fazit



© 2010 BearingPoint GmbH

19

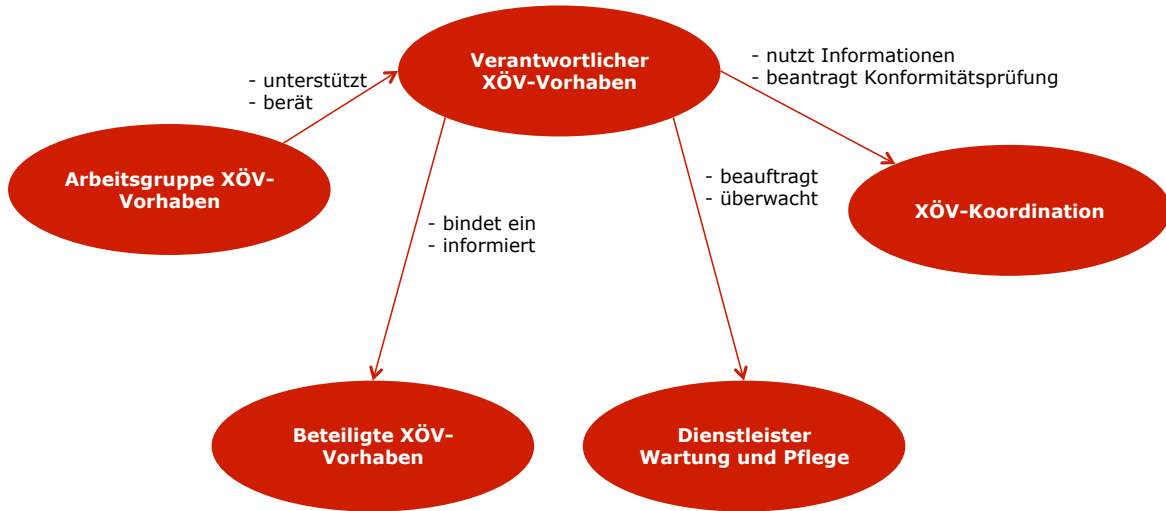
Bei der Pflege und Wartung sind eine Reihe unterschiedlicher Rollen zu berücksichtigen



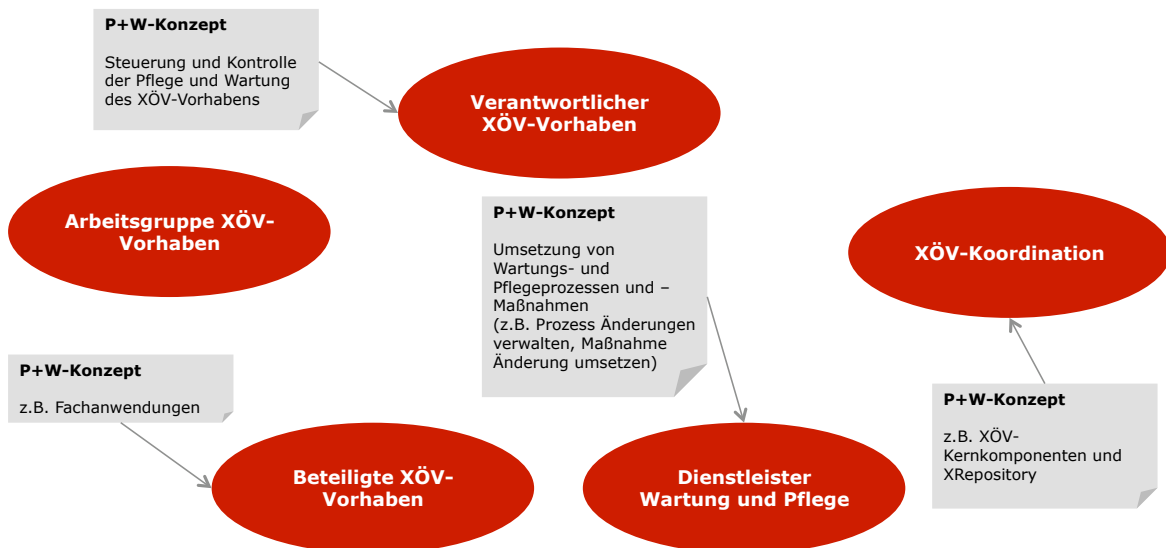
© 2010 BearingPoint GmbH

20

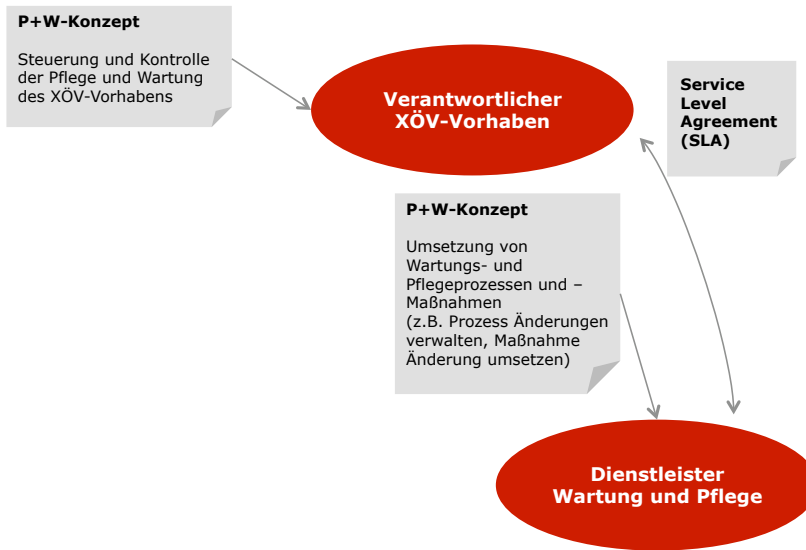
Die Organisation kann beispielsweise wie folgt aufgebaut werden



Für die Pflege- und Wartung sind unterschiedliche Konzepte notwendig



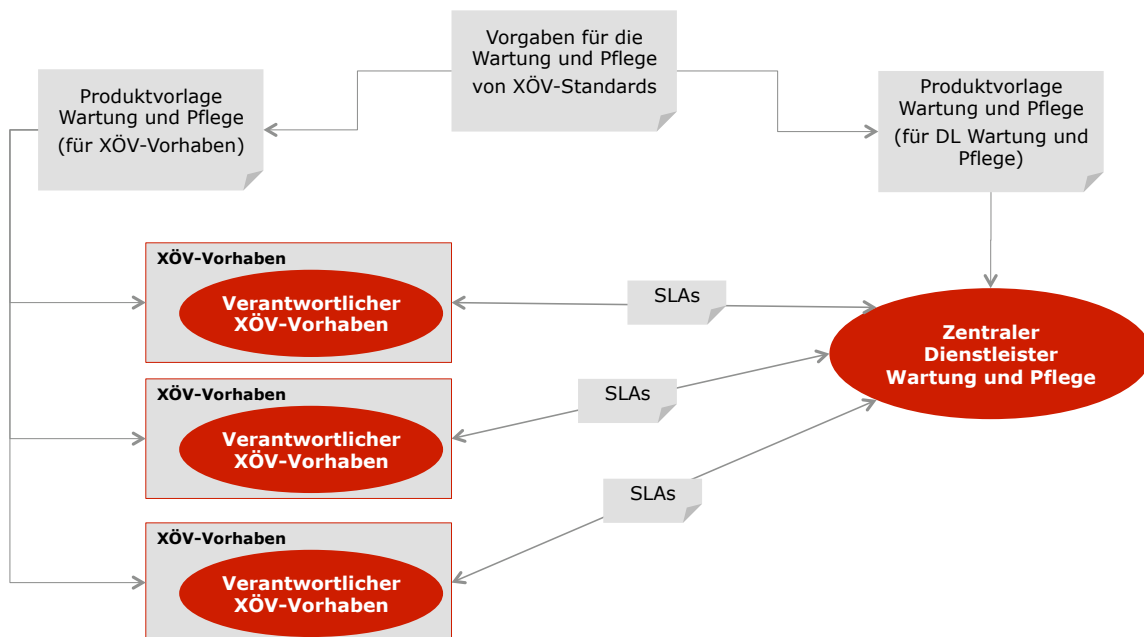
Im Rahmen der Produktvorlage wurde der Fokus auf die Pflege- und Wartungskonzepte für Vorhabenverantwortlichen und Dienstleister gelegt



© 2010 BearingPoint GmbH

23

Das Zusammenspiel zwischen Organisation und Pflege- und Wartungskonzepten kann wie folgt aussehen



© 2010 BearingPoint GmbH

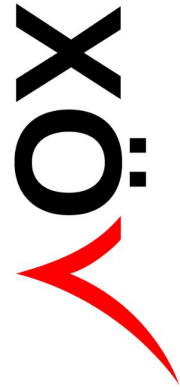
24



## Agenda



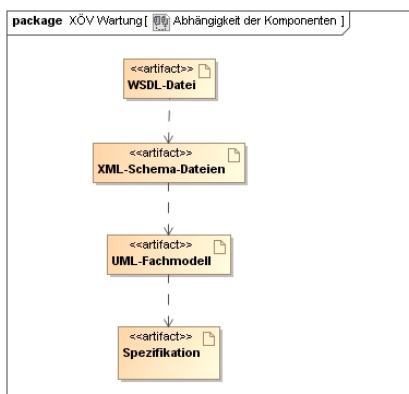
- Einführung XÖV
- Begriffsdefinitionen und Anforderungen an den Betrieb eines XÖV-Standards
- IST- und SOLL-Situation
- Rollen und Organisation
- Die Produktvorlage als Musterkonzept
- Fazit



© 2010 BearingPoint GmbH

25

## Was muss gepflegt werden?



- Steckbrief
- Spezifikation (Datenstrukturen, evtl. Prozesse)
- UML-Fachmodell als XMI-Datei, bestehend aus:
  - Informationsmodell
  - formale Beschreibung der kollaborativen Prozesse mit strukturierten Daten
  - XML-Schema-Dateien (automatisiert auf Basis des Fachmodells generiert)
- Pflege- und Wartungskonzept

### Optional

- Code-Listen, d. h. in den übermittelten Nachrichten zu nutzende Schlüsselstabellen
- Dokumentation der Bedingungen für die Inbetriebnahme der neuen Version des Standards
- Profilierung des Standards OSCI Transport
- WSDL-Vorlagendateien (zur Veröffentlichung im Deutschen Verwaltungsdienstverzeichnis DVDV)
- XML-Beispiel-Instanzen
- Liste gehaltener Vorträge
- Liste von Referenzprojekten
- Flyer und Artikel in Web- und Printmedien
- Gremienbeschlüsse
- Anforderungsdokumentation in dem aktuellen Stand
- Dokumentation von Lösungskonzepten und Lösungsumsetzung

© 2010 BearingPoint GmbH

26

## Aufbau der Produktvorlage Pflege- und Wartungskonzept für XÖV-Standards



- Festgelegtes Inhaltsverzeichnis mit muss- und soll-Kapiteln
  - Definition der Beteiligten
  - Funktionale Rollenbeschreibung
  - Definition der Prozesse (z.B. Änderungsmanagement, Fehlermanagement)
  - Definition zu pflegender Artefakte
  - Überwachung und Betrieb der Infrastruktur
  - Finanzierung und Ressourcen

Inhaltsverzeichnis	1
Version History	1
1 Erklärung	4
2 Ausgangssituation	5
2.1 Einordnung des Dokuments	5
2.1.1 Abgrenzung zur XÖV-Standardisierung	5
2.2 Problemstellung	6
2.3 Beteiligte	6
2.3.1 XÖV-Koordinator	6
2.3.2 XÖV-Zertifizierungsstelle	7
2.3.3 Bundeswerkstatt	7
2.3.4 Anwender	8
2.3.5 Softwarehersteller	8
2.3.6 Weitere Beteiligte	9
3 Vorüberweise und Rahmenbedingungen	20
3.1 Leitlinien und Grundsätze	20
3.1.1 XÖV-Standards anliegen zu entwickeln	20
3.1.2 Weiterentwicklung der Standards fördern	20
3.1.3 Investitionsbedarf und Risikoprüfung zu gewährleisten	21
3.1.4 Orientierung an anerkannten Strukturen und Techniken	21
3.1.5 Verlässlichkeit des Betriebes	22

- Festgelegte Textpassagen, die so zu übernehmen sind
- Optionale oder bedingte Textteile sowie Arbeitshinweise
  - *[Optional: ...]*
  - *[Arbeitshinweis: ...]*
  - *Der Hinweis definiert, wie mit dem Textteil umzugehen ist, ob und ggf. unter welchen Bedingungen er angepasst, gelöscht oder übernommen werden muss.*

© 2010 BearingPoint GmbH

27

## Einordnung der Produktvorlage Pflege- und Wartungskonzept für XÖV-Standards



- **Enthält** Definitionen, Regelungen, Prozesse und Aktivitäten, die für die Nutzung, Wiederverwendung und Weiterentwicklung eines XÖV-Standards nach seiner erstmaligen Veröffentlichung benötigt werden.
- **Bildet** den Rahmen und die Grundlage für die nachhaltige Bereitstellung und Nutzung des XÖV-Standards
- **Dient** sowohl der Aufgaben- und Anforderungsbeschreibung als auch der Festlegung der notwendigen Maßnahmen und Organisationsstrukturen im Rahmen der Pflege und Wartung des XÖV-Standards.

© 2010 BearingPoint GmbH

28

## Anforderungen an die Produktvorlage Pflege- und Wartungskonzept für XÖV-Standards



- Regelungen müssen stets konkret auf den in Rede stehenden XÖV-Standard bezogen sein
- Rechte und Pflichten von Bund, Ländern und Kommunen müssen im Einzelfall festgelegt werden
- Die Produktvorlage ist ein erläutertes Musterkonzept für XÖV-Standards, die als Vorlage für konkrete Konzepte dient.
- Der Erfahrungsaustausch zwischen den XÖV-Standards und dadurch die Vermeidung von Wiederholungsfehlern wird gefördert



### Ziele:

Reduzierung des Aufwands für die Entwicklung eines konkreten Pflege- und Wartungskonzepts

Vergleichbare Strukturen und einheitlicher Regelungsstand für alle Konzepte

Möglichkeit Prozesse und Infrastruktur gemeinsam zu nutzen und an zentrale Dienstleister auszulagern

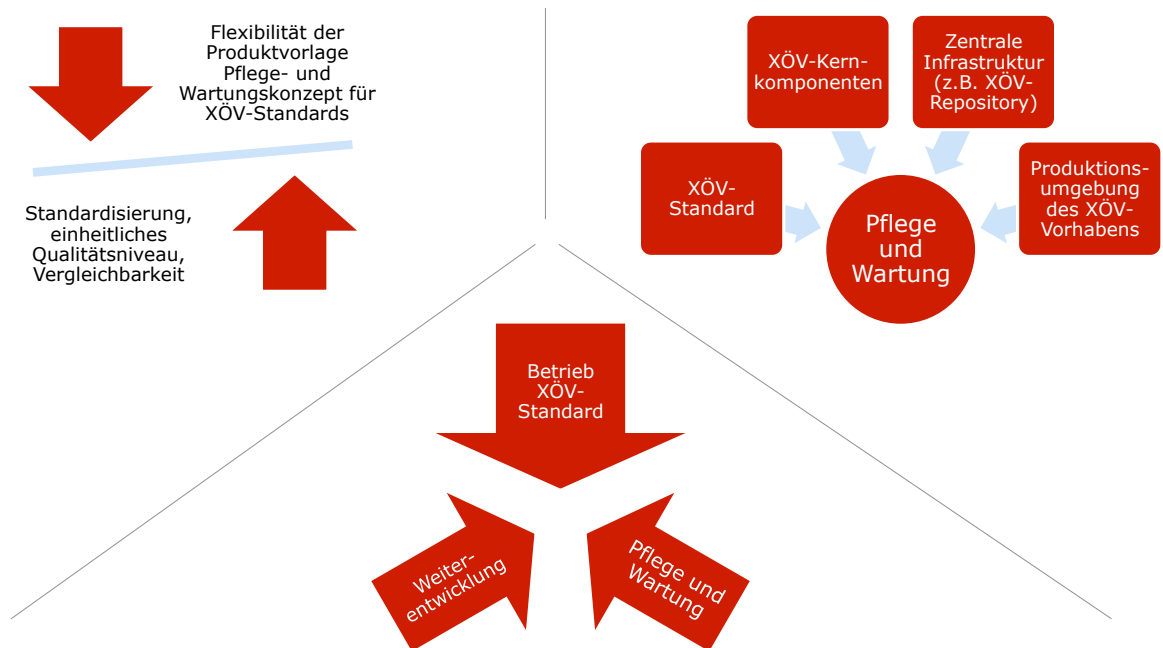
## Agenda



- Einführung XÖV
- Begriffsdefinitionen und Anforderungen an den Betrieb eines XÖV-Standards
- IST- und SOLL-Situation
- Rollen und Organisation
- Die Produktvorlage als Musterkonzept
- Fazit



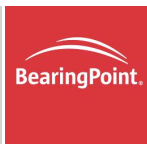
Mit der Produktvorlage hat BearingPoint eine Grundlage zur einheitlichen Pflege und Wartung von XÖV-Standards geschaffen



© 2010 BearingPoint GmbH

31

Vielen Dank für Ihre Aufmerksamkeit



▪ Haben Sie noch Fragen?

▪ Kontaktieren Sie mich:



**Jan-Peter Baas**  
Technology Architect

**BearingPoint GmbH**  
Speicherstrasse 1  
60327 Frankfurt  
[www.bearingpoint.de](http://www.bearingpoint.de)

Tel: +49.69.13022.5286  
Mobile: +49.173.3004948  
Jan-Peter.Baas@bearingpointconsulting.com

© 2010 BearingPoint GmbH

32

## *8.2 Ein Musterkonzept zur Wartung und Pflege von XÖV-Standards*

# Interoperabilität durch Standardisierung – Zertifizierte XÖV-Konformität für die Verwaltung

Cornelia Crome, Lars Lämmerhirt, Christian Lange

Bundesverwaltungsamt - Bundesstelle für Informationstechnik -  
50728 Köln

{cornelia.crome, lars.laemmerhirt, christian.lange}@bva.bund.de

**Abstract:** XÖV ist die Standardisierungsmethode der öffentlichen Verwaltung in Deutschland zur Förderung der semantischen Interoperabilität im elektronischen Datenaustausch. Um die Ziele der XÖV-Standardisierung zu erreichen, müssen methodische, technische und organisatorische Regeln bei der Entwicklung von fachlichen Datenschnittstellen beachtet werden. In diesem Vortrag werden die im April 2010 finalisierten XÖV-Konformitätskriterien und das zugehörige Zertifizierungsverfahren vorgestellt. Außerdem werden erste Erfahrungen aus der Anwendung der Konformitätskriterien und ein Ausblick auf die Entwicklung von Handlungsanweisungen für V-Modell-XT-basierende XÖV-Entwicklungsprojekte gegeben.

## 1 Einleitung

Die Bedeutung des elektronischen Datenaustauschs innerhalb und mit der öffentlichen Verwaltung ist groß und im Zuge steigender E-Government-Aktivitäten weiterhin steigend. Die Vielzahl miteinander kommunizierender Fachverfahren stellt hohe Ansprüche an die Interoperabilität ihrer Datenschnittstellen. Da unterschiedliche Fachverfahren häufig unabhängig in unterschiedlichen Behörden entwickelt werden, stellt insbesondere die Sicherstellung der semantischen Interoperabilität ihrer Schnittstellen oft ein Problem dar. Ein simples Beispiel ist der Name einer Person: „Otto; Graf Lamsdorff“ vs. „Otto; Graf; Lamsdorff“. Das prioritäre Deutschland-Online (DOL) Vorhaben „Standardisierung“<sup>1</sup> hat Methoden, Werkzeuge und Infrastruktur entwickelt, um die Einheitlichkeit im elektronischen Datenaustausch zu fördern. Das Ergebnis ist die XÖV-Standardisierung<sup>2</sup>.

Technisch betrachtet wird bei der Entwicklung einer XÖV-Schnittstellenspezifikation zunächst ein fachliches UML-Modell entwickelt, das mittels Modelltransformation durch das Werkzeug „XGenerator“ (das von DOL Standardisierung bereitgestellt wird) in ein XSD-Schema transformiert wird. Die Grundlagen hierfür bilden die technischen

<sup>1</sup> [www.standardisierung.deutschland-online.de](http://www.standardisierung.deutschland-online.de). Gemeinsames Vorhaben von Bund und Ländern, vertreten durch das Bundesministerium des Innern und dem Land Bremen

<sup>2</sup> XML-basierte fachliche Standards für den elektronischen Datenaustausch der öffentlichen Verwaltung

Spezifikationen der UN/CEFACT<sup>3</sup>. Methodisch werden zum Beispiel so genannte Kern- und Fachkomponenten, das sind standardisierte Informations- und Datenobjekte für häufig wiederkehrende Datensätze wie z.B. Name, Adresse oder Anschrift, wiederverwendet. Organisatorisch findet ein Abstimmungsprozess mit fachlichen Interessensvertretern und Gremien der XÖV-Standardisierung statt, um das Ziel der Interoperabilität zu erreichen. Das im April 2010 zu finalisierende XÖV-Handbuch [1] ist das umfassende Dokument, das die technischen, organisatorischen und methodischen Aspekte beschreibt und das Referenzdokument für XÖV-Entwickler ist. Insbesondere beschreibt das XÖV-Handbuch die Konformitätskriterien.

## 2 XÖV-Konformität

Derzeit sind 21 Projekte bei DOL Standardisierung registriert, die die XÖV-Methodik verfolgen. Die Projekte stammen aus unterschiedlichen Fachlichkeiten, wie dem Justizwesen, dem Meldewesen oder dem Dokumentenmanagement. Bisher wenden die Projekte die Bestandteile der XÖV-Standardisierung auf unterschiedliche Art und Weise an. Daher können die Ziele der XÖV-Standardisierung (Verbesserung von Interoperabilität, Qualität und Wirtschaftlichkeit in der Entwicklung) noch nicht vollständig ausgeschöpft werden. Die nun vorgelegten XÖV-Konformitätskriterien aus dem Handbuch [1] legen das für die Erreichung der Ziele erforderliche Regelwerk eindeutig fest. Neben den technischen Konformitätskriterien, die die Interoperabilität und technische Qualität einer zu entwickelnden Schnittstelle adressieren, gibt es Bereitstellungs- und Auskunftspflichten, die z.B. die Wiederverwendbarkeit für andere Standardisierungsprojekte und die Nachvollziehbarkeit der Nachhaltigkeit eines XÖV-Standards sicherstellen. Beispiele für Konformitätskriterien sind die korrekte Anwendung des XÖV-UML-Profiles, die Veröffentlichung der entwickelten Artefakte im XRepository [2] und die Bereitstellung eines Betriebs-Konzeptes für einen XÖV-Standard. Insgesamt gibt es 15 Konformitätskriterien, die in MUSS-Kriterien und SOLL-Kriterien unterteilt sind, wobei SOLL-Kriterien nachvollziehbar begründete Abweichungen zulassen, um z.B. Bestandsprojekten den Weg zur XÖV-Konformität zu ermöglichen. Damit Entwicklungsprojekte ihre XÖV-Konformität feststellen lassen können, wird voraussichtlich im Mai 2010 die zentrale XÖV-Zertifizierungsstelle ins Leben gerufen. Gegenstand der Zertifizierung ist die XÖV-Konformität einer entwickelten Datenschnittstelle. In diesem Vortrag wird die XÖV-Konformität erläutert und der Prozess der Zertifizierung von XÖV-Standards durch die Zertifizierungsstelle.

## 3 Fazit und Ausblick

Die XÖV-Methodik ist die Kombination eines modellgetriebenen Ansatzes mit methodischen und organisatorischen Aspekten der Abstimmung und Entwicklung. XÖV wurde von der öffentlichen Verwaltung für die öffentliche Verwaltung entwickelt, um bei der Entwicklung von Datenschnittstellen die Ziele Interoperabilität, technische Qualität und Wirtschaftlichkeit zu erreichen. Im XÖV-Handbuch werden den Entwicklern von

---

<sup>3</sup> United Nations Centre for Trade Facilitation and Electronic Business, <http://www.unece.org/cefact/>

Schnittstellen Konformitätskriterien bereitgestellt, damit diese Ziele erreicht werden können. Die XÖV-Zertifizierungsstelle überprüft und bestätigt die Konformität von entwickelten Schnittstellen. Derzeit werden außer den Konformitätskriterien auch (unverbindliche) Handlungsempfehlungen entwickelt, die den Entwicklern die Best-Practices der XÖV-Standardisierung anhand eines V-Modell XT-basierenden Vorgehens aufzeigen. Der Vortrag wird auch einen Ausblick auf die XÖV-Handlungsempfehlungen und Erfahrungen mit der XÖV-Konformität präsentieren.

### Referenzen

- [1] Deutschland-Online Standardisierung. XÖV-Handbuch, Version 1.0 (draft). August 2009.
- [2] XRepository: [URL:http://www.xrepository.deutschland-online.de](http://www.xrepository.deutschland-online.de). Bundesverwaltungsamt, Köln, 2009.





## Interoperabilität durch Standardisierung – Zertifizierte XÖV-Konformität für die Verwaltung

**Dr. Christian Lange**  
**Cornelia Crome**  
**Lars Lämmerhirt**  
BIT7 – Standards und Methoden  
Bundesstelle für Informationstechnik  
Köln

Seite: 1



## Ziele Aktionsplan Deutschland-Online

- ▶ Innovative, leistungsfähige und effiziente öffentliche Verwaltung als wesentliche Voraussetzung für den wirtschaftlichen Erfolg Deutschlands.
  - ▶ Der Einsatz moderner Informationstechnik und die Nutzung des Internets leisten hierzu einen entscheidenden Beitrag
  - ▶ Ziel: Umsetzung durchgängiger, elektronisch unterstützter und medienbruchfreier Verwaltungsprozesse über alle Verwaltungsebenen hinweg und die Bereitstellung von Verwaltungsdienstleistungen für die Wirtschaft und die Bürger im Internet.



**Braucht Standards für Daten und Kommunikation**

Seite: 2

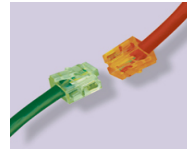


## Ausgangssituation

Vom gewollten politischen Föderalismus ...

.. zum unvermeidbaren Verwaltungsföderalismus ...

.. zum informationstechnischen Föderalismus?



### Die Folgen:

Heterogene Standardisierungsprojekte in Bund, Ländern und Kommunen

Kaum Nutzung einheitlicher Standardisierungskonzepte

Probleme (fachliche Inkompatibilität, Kosten, Verbindlichkeit)

Potenziale nicht oder nur unzureichend realisiert

Bessere Koordination der Standardisierungsaktivitäten

XÖV



## Ziele

Das Deutschland-Online-Vorhaben „Standardisierung“

- ▶ soll die **Entwicklung und Bereitstellung von fachlichen Standards** für den elektronischen Datenaustausch („XÖV-Standards“)
- ▶ **unterstützen und koordinieren**,
- ▶ so dass elektronische Prozesse innerhalb (G2G) und mit der Verwaltung (G2G, G2B) **effizient und in einheitlicher Weise** abgesehen werden können



## Überblick XÖV-Projekte

- ▶ XAgro
- ▶ XAusländer
- ▶ XBafög
- ▶ XBau
- ▶ XDomea
- ▶ XFinanz
- ▶ XGewerberegister
- ▶ XHydro
- ▶ XJustiz
- ▶ XSozial
- ▶ DatMLRaw/  
XGewerberegister

DEUTSCHLAND-ONLINE  
XAUSLÄNDER

XDOMEA  
XÖV Standard

XHydro

- ▶ XKfz
- ▶ XMeld
- ▶ XPass
- ▶ XPersonenstand
- ▶ XPlanung
- ▶ XPolizei
- ▶ XStatistik
- ▶ XUmwelt
- ▶ XVemags
- ▶ XVergabe

DEUTSCHLAND-ONLINE  
KFZ-WESEN

OSCI  
XMeld

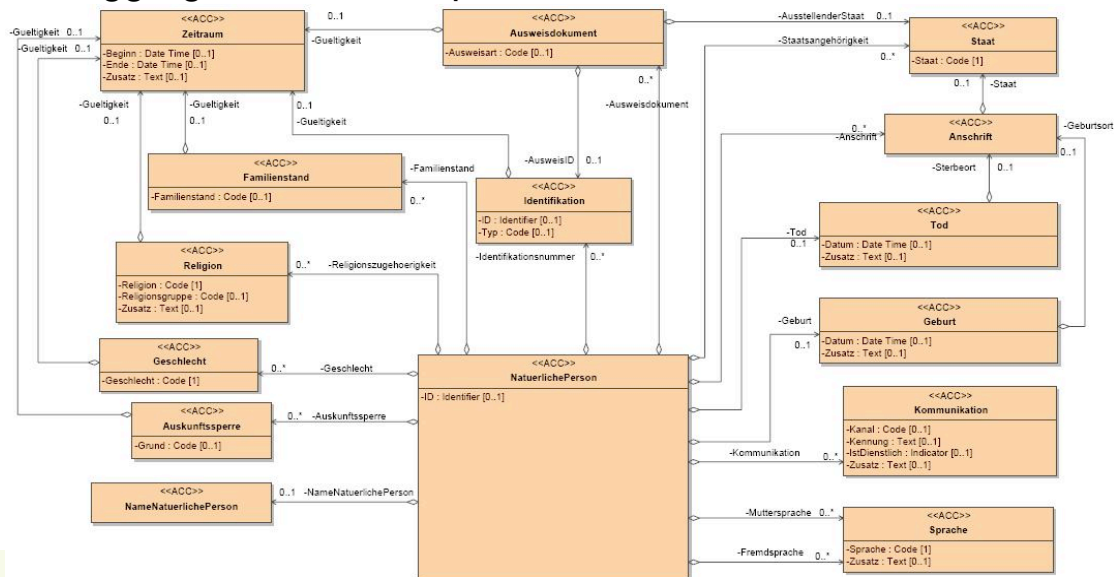
DEUTSCHLAND-ONLINE  
PERSONENSTAND

XPlanung

eSTATISTIK .core  
Common Online Rawdata Entry

## XÖV-Kernkomponenten

- ▶ Aggregierte Kernkomponente „Person“



## Konformitätskriterien und das XÖV-Handbuch 1.0

- ▶ Warum XÖV-Konformität?
  - Motivation
  - Ziele
  
- ▶ Konformitätskriterien
  - Bereitstellungspflichten
  - Auskunftspflichten
  - Technische Kriterien
  
- ▶ Wie wird ein Vorhaben XÖV-konform?

## XÖV-Konformität: Motivation

$X = X\ddot{O}V?$

## XÖV-Konformität: Motivation

Nicht überall wo ein „X“ drauf steht, steckt (richtiges) XÖV  
drin

## XÖV-Konformität: Motivation



## XÖV-Konformität: Ziele

...erforderlich sind Regeln zur Erreichung der  
**Standardisierungsziele:**

- ▶ Sicherstellung der **Interoperabilität**
- ▶ **Wirtschaftlichkeit** der Entwicklung
- ▶ **Risikominimierung**
- ▶ **Qualitätsverbesserung**

Diese Ziele dienen

- ▶ dem jeweiligen Standardisierungsvorhaben
- ▶ anderen Vorhaben
- ▶ den Anwendern des Standards
- ▶ der Öffentlichen Verwaltung im Allgemeinen

## Der Weg zur neuen XÖV-Konformität

- ▶ **Ausgangspunkt (2006):**
  - XÖV Framework 1.0



- ▶ **Ziel (2008):**
  - Überarbeitung



- ▶ **Ergebnis (2010):**
  - 15 Kriterien
  - vollständig neues Dokument
  - Konformitätszertifizierung

Herangehensweise

- Pragmatischer Ansatz
  - Relevanz zur Zielerreichung
  - Machbarkeit
- Anwendbares und gut verständliches Dokument
- Einbeziehung der bisherigen Erfahrungen



## Die drei Kategorien der Konformitätskriterien

- ▶ Bereitstellungspflichten
- ▶ Auskunftspflichten der Standardentwickler und –betreiber
- ▶ Technische Kriterien



## Die „Spielregeln“: MUSS und SOLL

- ▶ **MUSS-Kriterien**
  - sind zwingend einzuhalten
  - erforderlich zur Erreichung der XÖV-Ziele
  
- ▶ **SOLL-Kriterien**
  - sind einzuhalten
  - Abweichungen müssen schriftlich begründet werden
    - Rahmenbedingungen mit dringenden Gründen, wie z.B. wirtschaftliche oder fachliche Gründe, die eine Umsetzung NOCH nicht ermöglichen
  - Mittelfristig wird bei einigen Regeln aus SOLL ein **MUSS**



## Die drei Kategorien der Konformitätskriterien

### ▶ **Bereitstellungspflichten**

- Wer stellt den Standard wie bereit?
- Ziele
  - Sicherstellung ÖV als Eigentümer
  - Sicherstellung Recht zur (Wieder-)Verwendung
  - Sicherstellung der Investitionssicherheit
- ▶ Auskunftspflichten der Standardentwickler und –betreiber
- ▶ Technische Kriterien



## Bereitstellungspflichten I

1. **Ein Standard der öffentlichen Verwaltung (MUSS)**
  - ÖV ist „Eigentümer“
  - Keine Abhängigkeit von kommerziellen Interessen
  - Prüfinhalt: Entscheidungsgremien sind von der ÖV besetzt.
2. **Freie Verwendung (MUSS)**
  - Einstellung in das XRepository
  - Lizenz als Grundlage für Freiheit von rechten Dritter
3. **Dokumentation (MUSS)**
  - Weiterentwicklung ermöglichen
  - Vorhandensein von : UML, XML, Spezifikation, evtl. Code-Listen und WSDL-Vorlagen





## Bereitstellungspflichten II

### 4. Veröffentlichung im XRepository (MUSS)

- Zentrale Anlaufstelle,
- Einfache Auffindbarkeit
- UML/XMI, XML, Spezifikation (pdf)

### 5. Nachhaltigkeit des Standards (MUSS)

- Betriebskonzept muss vorhanden sein
- Wg. Investitionssicherheit für implementierende Verfahrenshersteller und andere Standards, die auf dem Standard aufbauen
- Plausibilitätsprüfung: Betriebskonzept enthält Angaben zu
  - Betreiber
  - Rollen und Verantwortlichkeiten
  - Finanzierung



## Die drei Kategorien der Konformitätskriterien

### ▶ Bereitstellungspflichten

### ▶ Auskunftspflichten der Standardentwickler und –betreiber

– Informationen über Standard bzw. Vorhaben

– Ziele

- Transparenz gegenüber XÖV-Koordination und Community
- Beurteilung der Nachhaltigkeit
- Vermeidung von Parallelentwicklungen

### ▶ Technische Kriterien



## Auskunftspflichten

### 6. Anzeige der Entwicklungsabsicht (SOLL)

- Vermeidung redundanter Entwicklungen
- Frühzeitiges Erkennen von Synergiepotential
- „Initialer Steckbrief“ und Kontaktaufnahme zur XÖV-Koordination

### 7. Informationen zum status quo des Standards (MUSS)

- Transparenz der XÖV-Standards
- „Steckbrief“
  - Aktuell
  - Regelgerecht



## Die drei Kategorien der Konformitätskriterien

- ▶ Bereitstellungspflichten
- ▶ Auskunftspflichten der Standardentwickler und –betreiber
- ▶ **Technische Kriterien**
  - Eigenschaften des UML-Fachmodells
  - Eigenschaften des XML-Schema (Darstellung des Fachmodells)
  - Ziele
    - Semantische Interoperabilität
    - Korrekte Anwendung der XÖV-Infrastruktur und –Methoden
  - Prüfbarkeit weitestgehend automatisiert
    - XGenerator 2



## Intermezzo - die XÖV-Produktionsumgebung -

- ▶ **UML-Modellierungswerkzeug**
  - Erzeugung UML-Fachmodell
  - Export im XMI-Format (geeignete Version)
- ▶ **XÖV-UML-Profil**
  - Stereotypen und Constraints zur XÖV-konforment Modellierung
- ▶ **XGenerator 2**
  - Modelltransformation (Fachmodell → XSD)
  - Kontrolle der Einhaltung der technischen Kriterien
  - Dokumentationserzeugung
- ▶ **Werkzeuge zur Dokumentationserzeugung** (optional)



## Technische Kriterien I

8. **Modellierung der Prozesse in UML (SOLL)**
  - Verständlichkeit der Prozesse
9. **Modellierung der Datenstrukturen in UML (MUSS)**
  - Verständlichkeit des Fachmodells
  - Fachliche Prüfbarkeit
  - Grundlage für Verarbeitung in XÖV-Produktionsumgebung
  - UML2 Klassendiagramm muss vorliegen
10. **Einhaltung der Namens- und Entwurfsregeln (MUSS)**
  - Sicherstellung der Interoperabilität
  - Sicherstellung der Konsistenz zwischen UML und XSD
  - Validierung mittels XGenerator 2

## Technische Kriterien II

### **11. Nutzung von Kern- und Fachkomponenten (SOLL)**

### **12. Nutzung der Basisdatentypen (SOLL)**

### **13. Nutzung von Code-Listen (SOLL)**

- Verbesserung der Interoperabilität
- Wiederverwendung von bewährten Strukturen
  - Vereinfachung der Entwicklung
  - Verringerung von Risiken

## Technische Kriterien III

### **14. Erfolgreiche Verarbeitung des UML-Fachmodells durch die XÖV-Produktionsumgebung (MUSS)**

- Einhaltung der technisch implementierten Namens- und Entwurfsregeln

### **15. Nutzung einer sicheren Infrastruktur für den elektronischen Datenaustausch (SOLL)**

- Empfohlen wird die Verwendung der Infrastrukturkomponenten
  - Public-Key Infrastruktur PKI-1 Verwaltung
  - OSCI-Transport
  - DVDV



Vielen Dank für Ihre Aufmerksamkeit.

	<b>Bundesverwaltungsamt</b> - Bundesstelle für Informationstechnik -	
<b>Dr. Christian Lange</b>		
Referatsleiter Standards und Methoden; Kompetenzzentrum Open Source Software		
POSTANSCHRIFT	Bundesverwaltungsamt, 50728 Köln	
TELEFON	+49 (0) 22899 358-1627 oder +49 (0) 221 758-1627	
FAX	+49 (0) 22899 10 358-1627	
E-MAIL	christian.lange@bva.bund.de	
INTERNET	www.bundesverwaltungsamt.de, www.bit.bund.de	

## Vom Meldewesen zu XÖV Erfolgsfaktoren der Standardisierung im eGovernment

Frank Steimke

OSCI-Leitstelle  
Senatorin für Finanzen  
R. Hilferding Platz 1  
28195 Bremen

frank.steimke@finanzen.bremen.de

Im Jahr 2002 hat in der öffentlichen Verwaltung Deutschlands eine nachhaltige Entwicklung des E-Government begonnen. Kernprozesse des staatlichen Meldewesens wurden auf die elektronische Datenübermittlung umgestellt. Angesichts der heterogenen IT-Ausstattung, der besonderen datenschutzrechtlichen Anforderungen und der vielfältigen Übermittlungsvorschriften war dies eine große Herausforderung. Sie konnte durch eine enge und konstruktive Zusammenarbeit der Fachleute aus den Kommunen, der IT-Industrie und des Gesetzgebers erfolgreich gemeistert werden. Seit dem 1. Januar 2007 finden Datenübermittlungen im Meldewesen ausschließlich elektronisch statt, unter Nutzung einer einheitlichen Schnittstelle. Wegen des konsequenten Einsatzes von Standards blieben die Produktvielfalt für IT-Verfahren des Meldewesens erhalten..

Inzwischen wurden weitere Standardisierungsprojekte durch die öffentliche Verwaltung initiiert. Das Vorgehensmodell zur Entwicklung, Einführung und Betrieb elektronischer Datenübermittlungen auf der Basis standardisierter Lösungen, welches sich im Meldewesen bewährt hat, konnte dem Grunde nach auch in anderen Bereichen beibehalten werden. Die Existenz vieler Standardisierungsvorhaben des E-Government, ermöglicht einen systematischen Vergleich. Das bewährte Vorgehensmodell für die Entwicklung und den Betrieb von Fachstandards des E-Government wird vorgestellt.

Bei der Entwicklung von Fachstandards für die öffentliche Verwaltung gibt es von Beginn an eine starke Wechselwirkung zwischen den rechtlich-organisatorischen Vorgaben einerseits, und der fachlich-technischen Entwicklung andererseits. Damit das Potenzial einer *Gestaltung* der Prozesse (anstatt der unreflektierten technischen Umsetzung von Vorgaben) genutzt werden kann, bedarf es einer adäquaten Projektorganisation. Anhand konkreter Beispiele wird gezeigt, wie diese Erkenntnis umgesetzt worden ist.

Dargestellt wird auch die besondere Rolle der Sicherheitsanforderungen und der Mechanismen zu deren Umsetzung. Flächendeckende E-Government Projekte in Deutschland müssen Heterogenität der IT-Infrastrukturen in den Ländern und Kommunen angemessen berücksichtigen, und dennoch den reibungslosen Datenaustausch gewährleisten.

Das im Rahmen der Standardisierungsinitiative „Deutschland Online“ entstandene „XÖV Handbuch“ soll es laufenden und zukünftigen Standardisierungsvorhaben erleichtern, bewährte Lösungen wiederzuverwenden. So soll die Chance auf Interoperabilität unterschiedlicher fachlicher Standards erhöht werden. Es wird dargestellt, in welcher Form die verallgemeinerten Erkenntnisse des Meldewesen-Projektes Eingang in das Handbuch gefunden haben. Dabei werden auch die noch nicht gelösten organisatorischen, methodischen und technischen Fragestellungen angesprochen.

## Vom Meldewesen zu XÖV Erfolgsfaktoren der Standardisierung im eGovernment

*SEE 2010  
Köln, 4. Mai 2010*

Frank Steimke  
OSCI Leitstelle, Bremen

- Vorstellung
- Standardisierung im Meldewesen
- Standardisierung in der öffentlichen Verwaltung
- Fazit



- Gegründet 1998 im Rahmen von MEDIA@Komm
- Organisationseinheit der Bremer Verwaltung
  - Vollständig fremdfinanziert
- Derzeit vier MitarbeiterInnen in Bremen, sowie:
  - Kooperation mit dem Innenressort in Bremen
  - Zusammenarbeit mit der Universität Bremen (TZI)
  - Kooperation mit dem Bundesministerium des Innern
  - Unterstützt durch externe Dienstleister
- Aufgabenfelder
  - Querschnittsaufgaben der Standardisierung
  - (Unterstützung bei) Entwicklung von Fachstandards
- Ab 2012: Koordinierungsstelle für IT-Standards

Folie 3

- Dezentrale Datenhaltung auf kommunaler Ebene
  - Ca. 5.400 Meldebehörden und Melderegister
  - Ständiger Datenabgleich bei Veränderungen
- Ca. 20 verschiedene IT-Verfahren im Einsatz
- Technische Schnittstellen: kaum vorhanden, nicht kompatibel
- Rückmeldungen wurden per Brief übermittelt
  - Kosten betragen ca. 2,70 Euro / Rückmeldung
- Melderegisterauskünfte: Hoher Bedarf, Schlechter Service
  - Ca. 80 Mio Melderegisterauskünfte pro Jahr
  - Prozessdauer: mehrere Tage ... mehrere Wochen
- Bundesbehörden nutzen proprietäre Schnittstellen
  - Hoher Aufwand für überregional tätige Hersteller

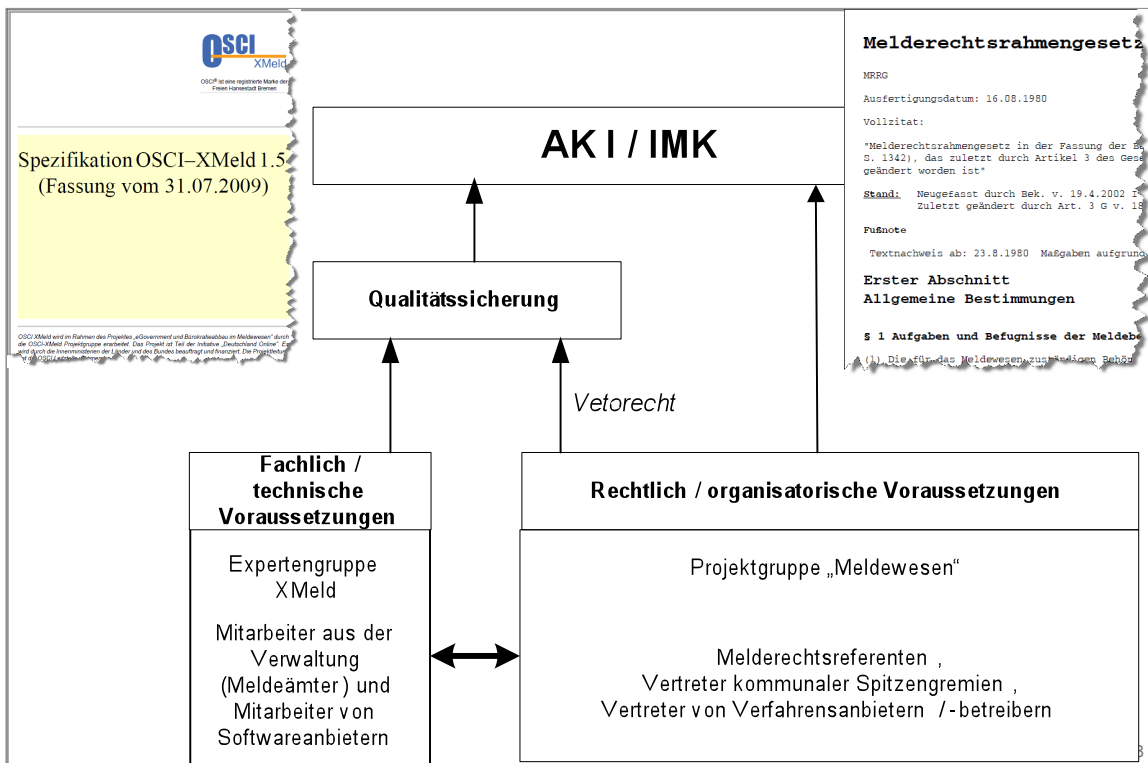
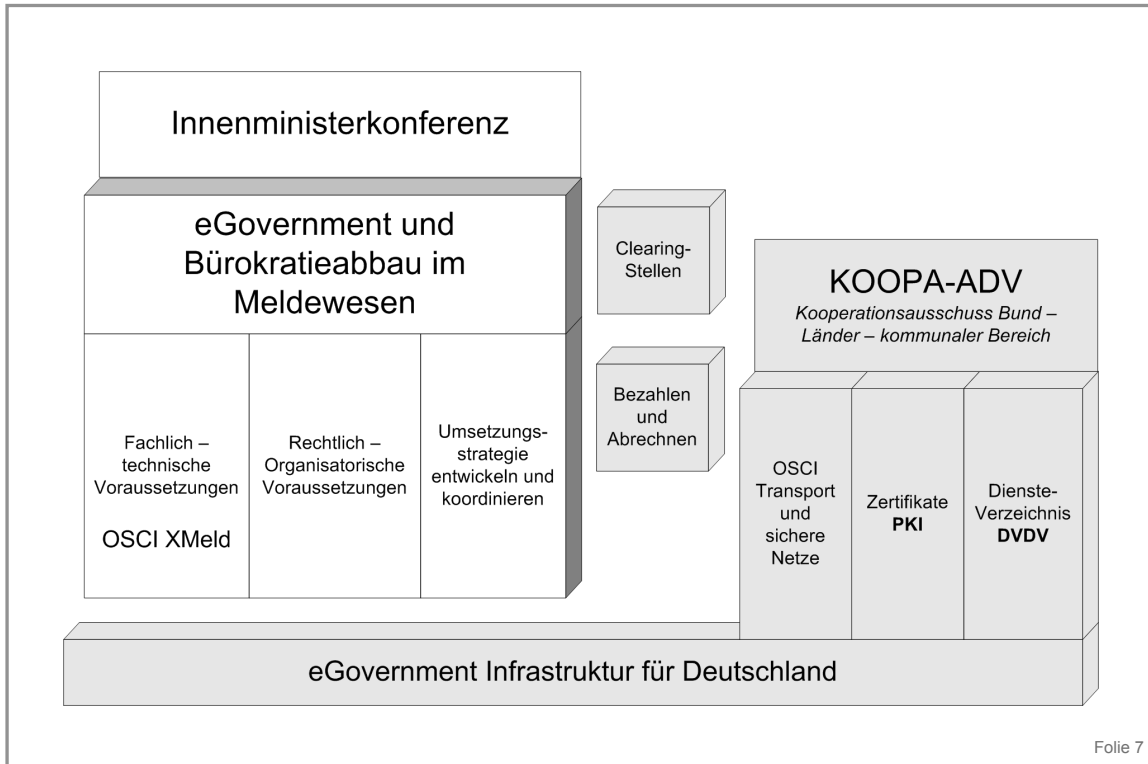
Folie 4

- Die elektronische Anmeldung wird erlaubt
  - Bürgerfreundlichkeit und Bürokratieabbau
  - Qualifizierte elektronische Signatur
- Die Abmeldepflicht entfällt
  - Wird substituiert durch die Vernetzung der Register
- Die elektronische Melderegisterauskunft wird möglich
  - Geringere Kosten, Höhere Qualität
- Die Meldebehörden werden vernetzt
  - Die elektronische Übermittlung wird erlaubt / gefordert
  - Die Frist zur Übermittlung wird auf drei Tage begrenzt
  - Fortgeschrittene elektronische Signatur

Folie 5

- Schaffung rechtlich-organisatorischer Voraussetzungen
- Entwicklung eines fachlichen Standards
- Einrichtung und Betrieb der technischen Infrastruktur
- Koordination der Umsetzung

Folie 6



- Standard OSCI-XML
  - Entwickelt und modelliert in (profilierem) UML
  - Beschreibt Datenstrukturen und Nachrichten
  - Transformiert in XML-Schema
  - Mit halbautomatisch erstellter Dokumentation
- Standard OSCI-Transport 1.2
  - *Secure Web Services 2002*
  - Sichere Datenübermittlung über das Internet
  - Gewährleistet Vertraulichkeit, Integrität und Authentizität
  - Elektronische Signaturen, X.509 Zertifikate, SigG-konform
  - Referenzimplementierung für Clients kostenfrei erhältlich
- Verzeichnisdienst für elektronische Services (DVDV)
  - Verteilte Infrastruktur (Bund / Länder)
  - Servicebeschreibung WSDL (Erweiterung für OSCI-Transport)

Folie 9

- Die flächendeckende Infrastruktur ist in Betrieb
  - Alle Meldebehörden sind vernetzt
  - XMeld ist die einheitliche fachliche Schnittstelle
- Positive Effekte sind eingetreten
  - Kosten wurden gesenkt (geschätzt 10...20 Mio. € / Jahr)
  - Die Datenqualität wurde erhöht
  - Die Geschwindigkeit wurde gesteigert
- Weiterentwicklung und Erweiterung
  - Zentrale Melderegister, Rentenmitteilungen, ElsterLohn II, ...
- Offener, frei verfügbarer Standard
  - Verbindlich vereinbartes Betriebskonzept
  - Prozesse der Änderung sind transparent

Folie 10

- Vorstellung
- Standardisierung im Meldewesen
- Standardisierung in der öffentlichen Verwaltung
- Fazit

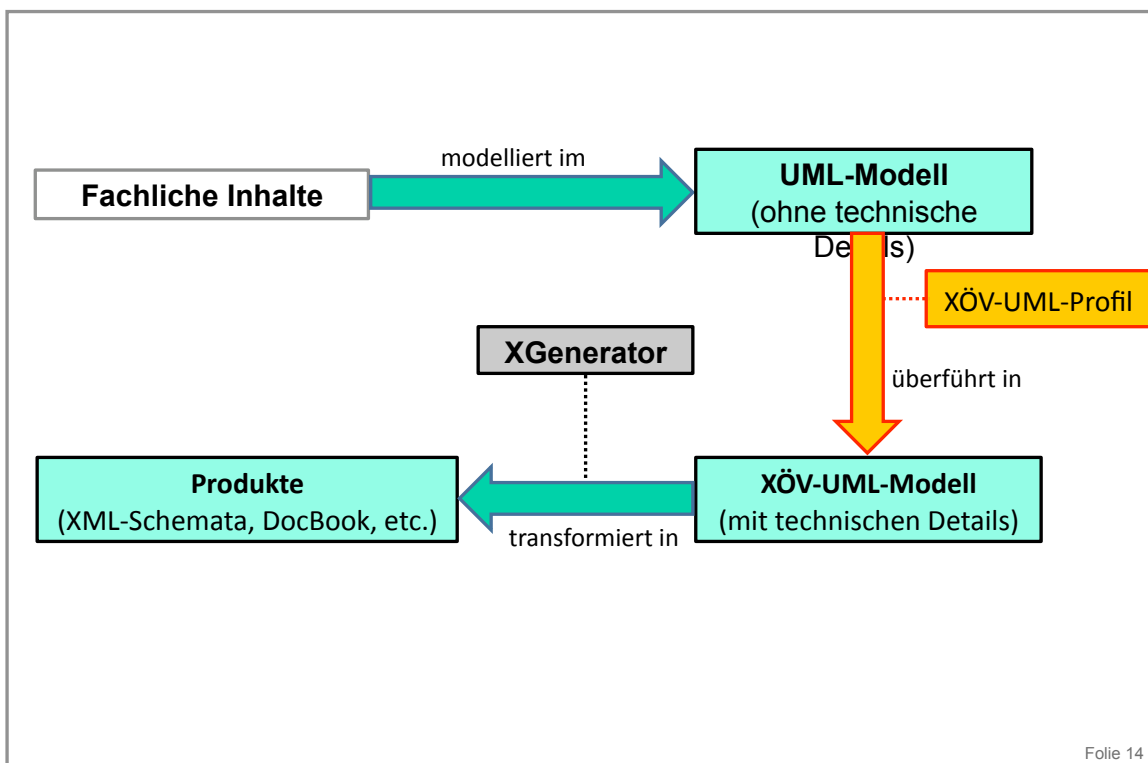
Folie 11

- Elektronischer Rechtsverkehr
- Austausch und Aussonderung behördlichen Schriftgutes
- Pass- und Ausweiswesen
- Kraftfahrzeugwesen
- Personenstandswesen
- Straßenverkehrswesen
- Ausländerwesen
- Finanzwesen
- Öffentliches Ausschreibungswesen
- Deutschland Online „Standardisierung“
  - **Koordination** der Standardisierungsaktivitäten
  - **Entwicklung** fachunabhängiger / fachübergreifender Lösungen (XÖV)

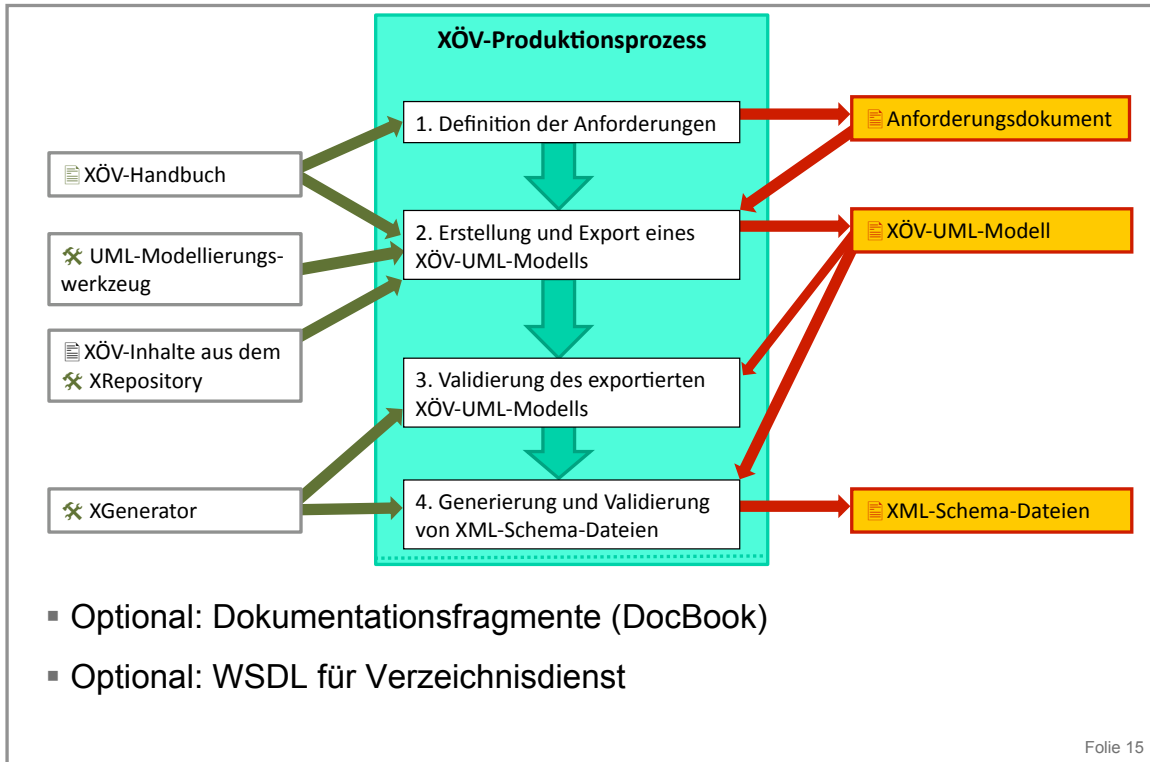
Folie 12

- Start small, think big
- Recht, Organisation und Technik ganzheitlich bearbeiten
  - Unterschiedliche Zuständigkeiten, aber ein Projekt
  - Rechtsgrundlagen schaffen Investitionssicherheit
  - **Die Verwaltung bestimmt ihre Standards selbst! (Regel K-1)**
- Kein Projekt ohne Projektauftrag!
  - Klären der Projektziele und der Beteiligten
  - Verantwortung festlegen, Ressourcen ehrlich klären und überwachen
- Akzeptanz nur bei offener und transparenter Entwicklung (Regeln K-2 ... K-4)
- Auch Standards haben einen Lebenszyklus
  - Der Betrieb muss bereits bei der Entwicklung mit bedacht werden (Regel K-5 ... K-7)
- Kein E-Government ohne eine sichere Infrastruktur
  - Unentbehrlich für medienbruchfreie Prozesse (Regel K-15)
- Teilautomatisierte Entwicklung unverzichtbar (Regeln K-8 ... K-14)
  - Geringere Entwicklungskosten, Höhere Qualität, Stärkere Einheitlichkeit

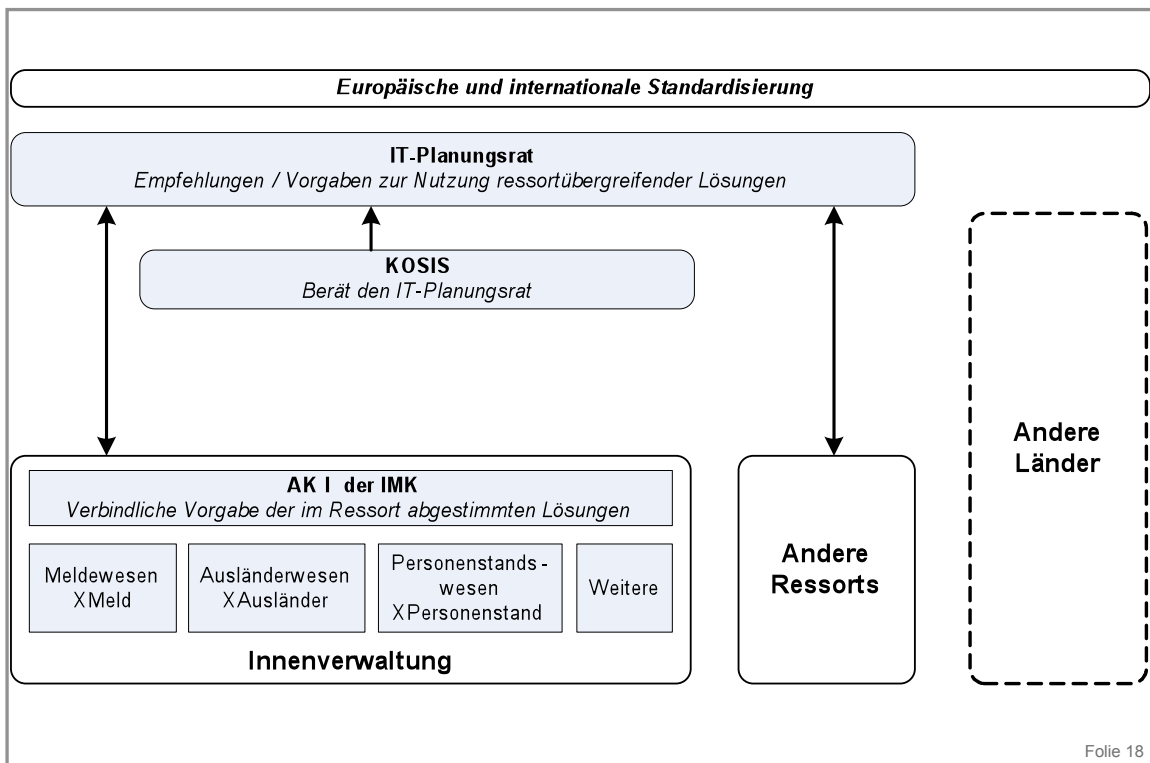
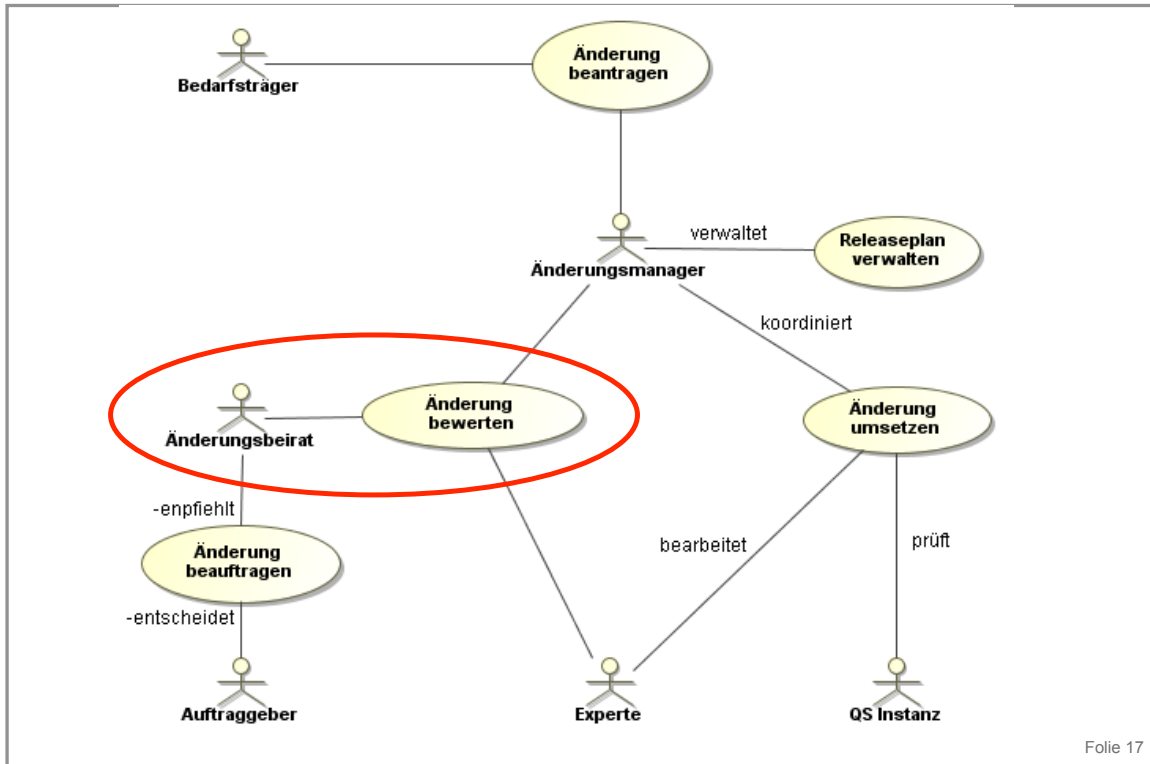
Folie 13



Folie 14



- OSCI-Transport 2: secure web services profiled for German needs
    - Bases on international standards from WS\* and WS-Security
    - Profiling is done to meet German (and European) laws
    - Some extensions for issues known from Version 1 Experiences
    - Developed with leading Experts from Industry as well as Government
    - Brought in EU Large Scale Pilots (SPOCS, PEPPOL)
  - SAFE: Secure Access for Federated eGovernment
    - Standardized interfaces to identity management techniques
    - Registration, authentication, authorization of communication participants
    - Based on WS\*-Stack, profiled to improve interoperability
    - Basic part: Application independent
    - Further profiling for applications in eJustic in a second part
  - E-Identity, Neuer Personalausweis ...
- Folie 16





- Vorstellung
- Standardisierung im Meldewesen
- Standardisierung in der öffentlichen Verwaltung
- Fazit

Folie 19

- Praxisbezogenes „Bottom Up“ Vorgehen hat sich bewährt
- Einrichtung der technischen Infrastruktur war erfolgreich
  - Enabler für weitere Projekte
  - Deutschland hat Praxiserfahrung mit Secure Webservices
- Organisation der Entwicklung hat sich bewährt
- Technischer Ansatz „Modelltransformation“ hat sich bewährt
  - Unverzichtbar für Entwicklung und Prüfung
- Seit März 2010: XÖV-Konformitätskriterien
  - Trennung der Spreu vom Weizen
  - Jedenfalls bezüglich der Formalkriterien

Folie 20

- Fachliches
  - Was nützen Kernkomponenten, wenn es schon am Zeichensatz hapert?
- Technik und Methodologie
  - Umgang mit wiederverwendbaren Datentypen (Atomar vs. Restriction)
  - Modularisierung
  - Semantic Constraints und XML-Schema?
  - CEFACT Core Components: überaus komplex
  - Brauchen wir die gleichen Datentypen oder die selben?
- Organisation
  - Deutschland-weite Abstimmung: schwierig bis unmöglich
  - Wahrnehmung einer **Architekturverantwortung** für den IT-Planungsrat

Folie 21

Vielen Dank für Ihre Aufmerksamkeit

Frank Steimke  
OSCI Leitstelle, Bremen

Folie 22



## Semantische Repositorys in Europa

Renke Fahl-Spiewack

Team Consulting  
]init[ AG für digitale Kommunikation, Köpenicker Str. 9  
10997 Berlin  
renke.fahl-spiewack@init.de

**Abstract:** Um unnötige Mehrfacharbeit und daraus resultierende Inkompatibilitäten bei der Erstellung von E-Government Anwendungen zu vermeiden werden Repositorys benötigt, die Arbeitsergebnisse sichern, Standards bereitstellen und bei der Wiederverwendung unterstützen. Sowohl die EU als auch einzelne Mitgliedsstaaten wie Deutschland haben dies erkannt und bauen derzeit eine Repository-Landschaft in Europa auf. Diese Landschaft soll hier beschrieben und die weitere Entwicklung aufgezeigt werden.

### 1 Bedeutung von Repositorys für den pan-europäischen Datenaustausch

Für viele Aufgabenstellungen werden E-Government-Anwendungen mehrfach entwickelt, sei es weil mehrere Anbieter konkurrierende Produkte erstellen oder die gleiche Aufgabe in unterschiedlichen Verantwortlichkeiten auftritt, z.B. in allen Bundesländern gleichermaßen. Soll Interoperabilität der Anwendungen sichergestellt werden, ist eine zentrale Bereitstellung von Standards über ein zentrales Repository notwendig. Werden gleichzeitig Arbeitsergebnisse abgeschlossener Projekte zur Verfügung gestellt, lassen sich durch Wiederverwendung dieser Ergebnisse Einsparungen in laufenden Projekten erzielen.

Zusätzlichen Nutzen erfahren Repositorys, wenn sie durch begleitende Initiativen flankiert werden und ihre Zielgruppen durch zusätzliche Informations- und Schulungsangebote unterstützen.

### 2 Stand der Entwicklung: Europäische und nationale Repositorys

Für die Europäische Union wird die Rolle des zentralen Repositorys für semantische Bausteine durch SEMIC.EU [S08] ausgefüllt. Dabei verfügt SEMIC.EU nicht nur über ein Repository zur zentralen Bereitstellung von semantischen Bausteinen (z.B. Code-Listen, Datenmodelle, XML-Schemata, etc.). Es bietet darüber hinaus weiterführende Informationen, Coachingangebote und weitere Dienstleistungen rund um das Thema semantische Interoperabilität.

In Deutschland wird auf einer ähnlichen technischen Basis das XRepository [X09] betrieben, das im Rahmen des Deutschland-Online Vorhabens Standardisierung vor allem für eine zentrale Bereitstellung der sogenannten X-Standards sorgt; Standards für den Datenaustausch, die im Rahmen von Deutschland-Online Standardisierung erstellt bzw. gefördert werden.

Andere EU-Mitgliedstaaten wie Finnland, Frankreich oder der Beitrittskandidat Türkei, erwägen die Schaffung von Repositories auf Basis der SEMIC.EU Softwarearchitektur. Weitere Mitgliedsstaaten wie Dänemark, Österreich oder Großbritannien betreiben eigene Plattformen auf unterschiedlicher technischer Basis.

Trotz einiger Anstrengungen handelt es sich bei fast allen Repositories bisher um Insellösungen für eine Fachdomäne oder eine nationale Anlaufstelle. Mittelfristig wird eine pan-Europäische Intergration dieser Repositories notwendig werden.

### 3 Repository Zukunft

Ein vereintes Europa kann nur nach dem Motto „global denken – lokal handeln“ funktionieren. Perspektivisch werden daher die nationalen Repositories weiterhin ihre nationalen Zielgruppen bedienen, während SEMIC.EU als zentrales europäisches Repository eigene europäische Inhalte bereitstellt und darüber hinaus über eine Föderation mit den nationalen Repositories deren Inhalte auf europäischer Ebene auffindbar macht. Der Weg dorthin wird nicht leicht und birgt einige technische und organisatorische Herausforderungen.

Für Deutschland und das XRepository bestehen die nächsten Schritte darin, die Plattform um weitere Inhalte und neue Funktionen zu ergänzen. Damit sollen weitere Anwendungsgebiete erschlossen werden, die Bedienbarkeit wird weiter verbessert und die Grundlagen für höhere Akzeptanz werden geschaffen.

Auf europäischer Ebene steht SEMIC.EU vor der Aufgabe, im neuen ISA-Programm der Kommission nicht nur die etablierten Services weiterzuführen, sondern sich auch weiter den nationalen Repositories gegenüber zu öffnen, um eine echte Transparenz über die Grenzen in Europa hinweg zu schaffen.

Dabei stehen alle Repository-Initiativen vor der Herausforderung, in den kommenden Jahren den Spagat zwischen Management-Attention und technischer Nutzbarkeit zu organisieren. Sowohl Entscheider als auch Techniker müssen die Plattformen akzeptieren, damit sie erfolgreich die Interoperabilität in Europa voran bringen können.

### Literaturverzeichnis

- [S08] SEMIC.EU: [URL:http://www.semic.eu](http://www.semic.eu), European Commission, Brüssel, 2008.  
[X09] XRepository: [URL:http://www.xrepository.deutschland-online.de](http://www.xrepository.deutschland-online.de), Bundesverwaltungssamt, Köln, 2009.



## Semantische Repositorys in Europa

SEMIC.EU –  
Service für Interoperabilität und  
Datenaustausch in Europa

Dr. Renke Fahl-Spiewack  
]init[ AG für digitale Kommunikation

21. April 2010



]init[

Semantische Repositorys in Europa

### ]init[ im Profil

- Gründung 1995 in Berlin
- Kompetenzzentrum für eGovernment
- > 220 Mitarbeiter in sechs Abteilungen:  
Communications, Design, Software Development,  
Operations, Consulting, Customer Service
- Standorte in Berlin, Mainz, München sowie in  
Abu Dhabi und Brüssel

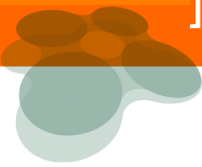
21. April 2010



### Full-Service Geschäftsbereiche der ]init[ AG

<b>Internetportale</b> Individuelle Lösungen für komplexe Online-Angebote.	<b>Consulting</b> Beratung und Entwicklung langfristiger IT- und E-Government-Strategien.
<b>Fachanwendungen</b> Realisierung elektronischer Geschäftsprozesse und Transaktionen.	<b>IT-Services</b> Sicherer Betrieb und Managed Services im ]init[ Rechenzentrum.

21. April 2010



### ]init[ Consulting

<b>Interoperabilität &amp; eGovernment</b> Die Vordenker für Daten- und Prozessintegration	<b>Consulting</b> Beratung und Entwicklung langfristiger IT- und E-Government-Strategien.
<ul style="list-style-type: none"><li>→ Spezialist für Cross-Border Interoperabilität</li><li>→ Betreiber von Clearinghouse Services</li><li>→ Moderator von Standardisierungsprozessen</li><li>→ Kompetensträger für Datenaustausch in Deutschland und Europa</li></ul>	

21. April 2010

**]init[** Semantische Repositys in Europa

Die Wegstrecke festlegen

Was ist SEMIC.EU ?

Was ist Semantik?  
Was heißt Integration & Standardisierung?

Was ist ein Interoperability Clearinghouse?

Warum „Interoperability Assets“?

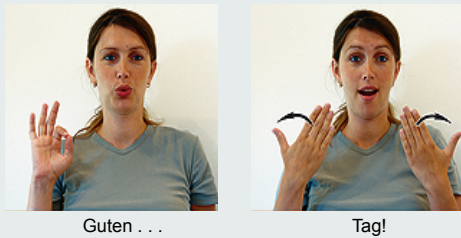
Standardisierung in Europa:  
Data Interoperability oder Information Interoperability?

**SEMIC.EU**  
Föderation von Repositys  
Semantische Repositys  
in Europa

21. April 2010

**]init[** Semantische Repositys in Europa

Wo immer Verständigung stattfindet . . .



. . . wurde Semantik geklärt und Interoperabilität geschaffen.

21. April 2010





### Service für Interoperabilität in Europa

[www.semic.eu](http://www.semic.eu)

Die Plattform für semantische Interoperabilität

[coaching@semic.eu](mailto:coaching@semic.eu)

Kostenfreier Coaching-Service der Europäischen Kommission

[collaboration@semic.eu](mailto:collaboration@semic.eu)

Experten Netzwerk für europäisches eGovernment

[community@semic.eu](mailto:community@semic.eu)

Anwenderforum – Projektpraxis auf dem kurzen Dienstweg



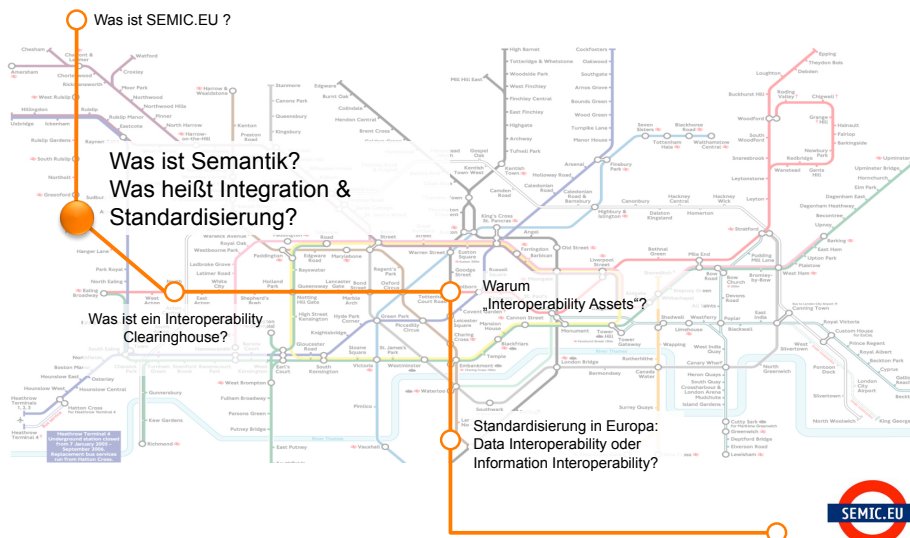
**Verlässlicher Service für eGovernment in Europa**

 **Servicebereitschaft bis 2015**

21. April 2010



### Die Wegstrecke festlegen



21. April 2010

Föderation von Repositories  
Semantische Repositories  
in Europa



### Wo bitte trifft man „semantische Interoperabilität“?

Der einfache Fall: Adresdatenaustausch . . .



. . . wie gleich müssen Adressen eigentlich sein?

21. April 2010

21. April 10



### Semantische Interoperabilität

*what's the meaning of this?!*

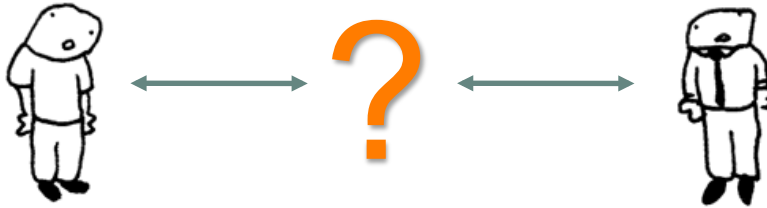


Semantische Interoperabilität:  
Schön, wenn wir die Antworten bekommen,  
nach denen wir auch gefragt haben.

21. April 2010

21. April 10

### Kernfragen von Semantik & Interoperabilität

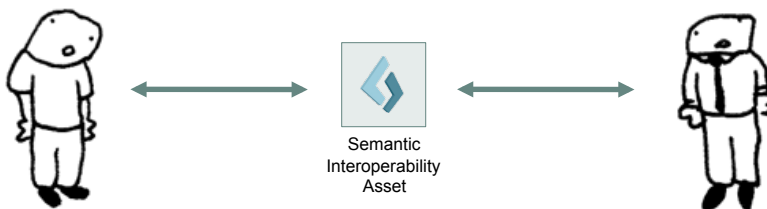


- 1. Werden die richtigen Daten bereitgestellt?
- 2. Werden die Daten richtig bereitgestellt?
- 3. Werden nicht-funktionale Anforderungen berücksichtigt?

21. April 2010

21. April 10

### Das Konzept Interoperability Assets



**Interoperability Assets:**  
Datenaustausch auf Basis geprüfter Austauschformate.  
Bereitgestellt im Asset-Repository [www.semic.eu](http://www.semic.eu)

21. April 2010

21. April 10

**]init[** Semantische Repositories in Europa

Das Konzept Interoperability Assets



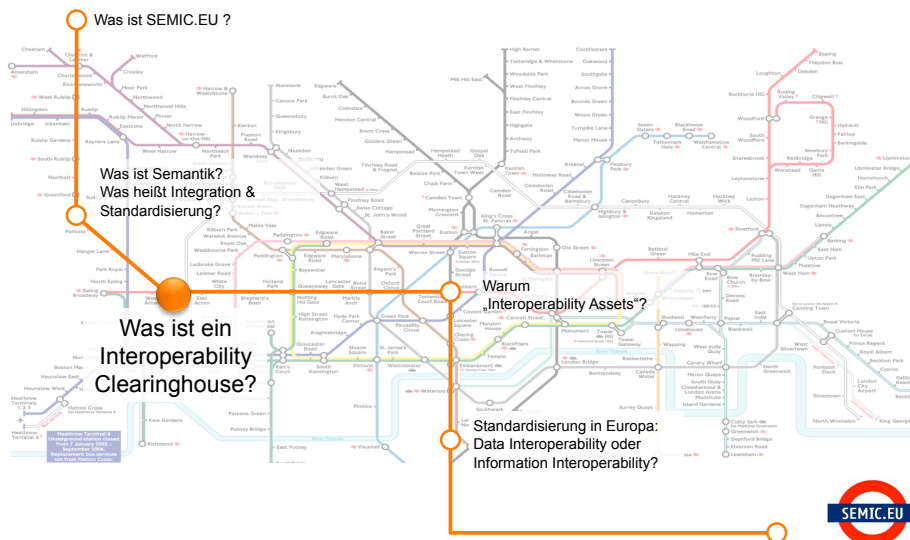
**Interoperability Assets:**  
 Beschleunigte Cross-Border Erweiterbarkeit  
 Ausbau des Datenaustauschs in Europa [www.semic.eu](http://www.semic.eu)

21. April 2010

21. April 10

**]init[** Semantische Repositories in Europa

Die Wegstrecke festlegen



21. April 2010

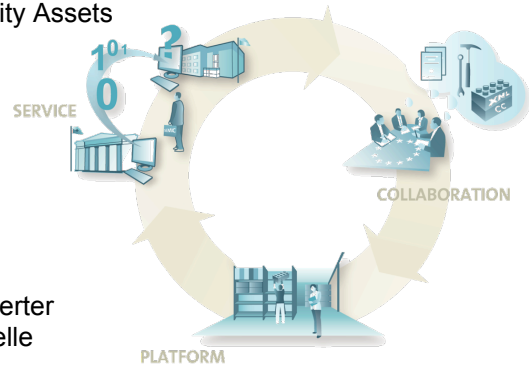
Föderation von Repositories  
 Semantische Repositories  
 in Europa

**]init[** Semantische Repositories in Europa



Leistungskomponenten

- **Service**  
Coaching für die Entwicklung und Wiederverwendung von Interoperability Assets
- **Collaboration**  
Fachliche Community für den europäischen Datenaustausch
- **Repository**  
Für die Bereitstellung qualitätsgesicherter Konzepte, Methoden und Datenmodelle



21. April 2010

**]init[** Semantische Repositories in Europa

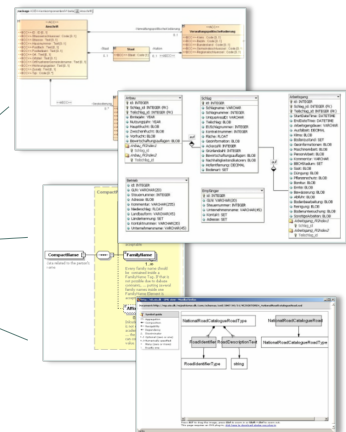
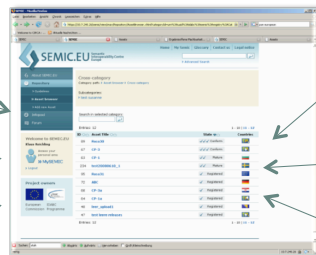
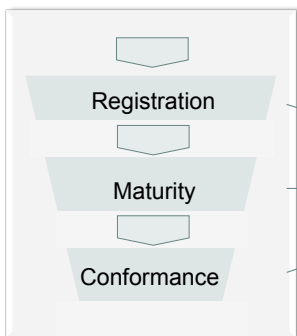


Das ‚Semantic Clearinghouse‘

Clearing Process

SEMIC.EU Repository

Semantic Assets

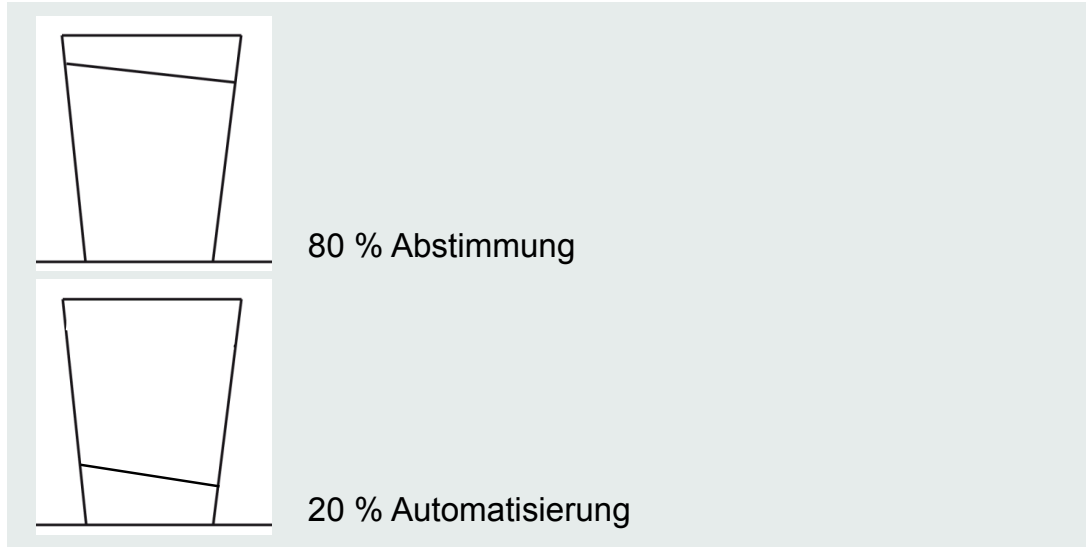


21. April 2010

**]init[** Semantische Repositorys in Europa



Wie schafft man eigentlich Interoperabilität?  
Eine Zwischenbilanz



21. April 2010

**]init[** Semantische Repositorys in Europa



SEMIC.EU – Netzwerk der Europäischen Zusammenarbeit



21. April 2010

**]init[ Semantische Repositories in Europa**

Plattform Funktionalität und Interaktion

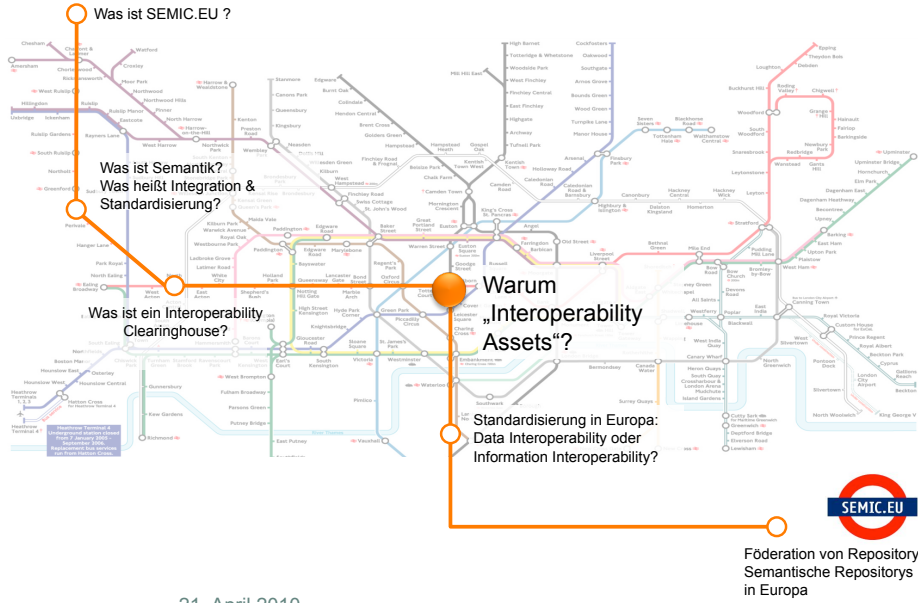
21. April 10

**]init[ Semantische Repositories in Europa**

Service beginnt immer mit einer konkreten Frage

**]init[** Semantische Repositories in Europa

Die Wegstrecke festlegen



**]init[** Semantische Repositories in Europa

Integration „one by one“

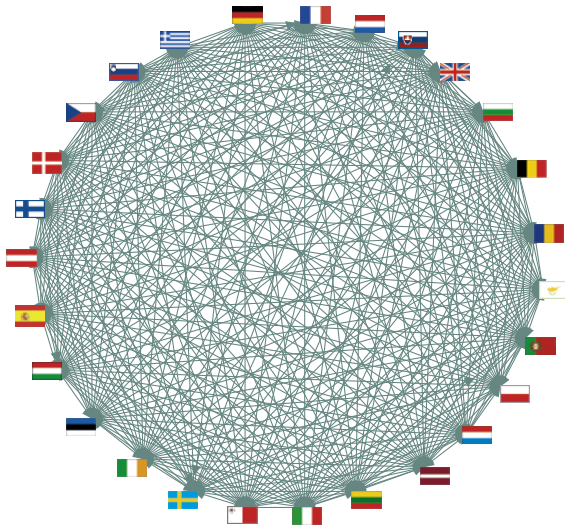




**]init[** Semantische Repositories in Europa



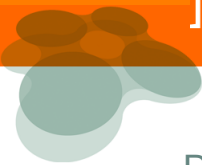
### Vollständige bilaterale Integration



Anzahl bilateraler Austauschbeziehungen bei 27 Mitgliedsstaaten: 702

21. April 2010

**]init[** Semantische Repositories in Europa



### Das Konzept "Interoperability Assets"

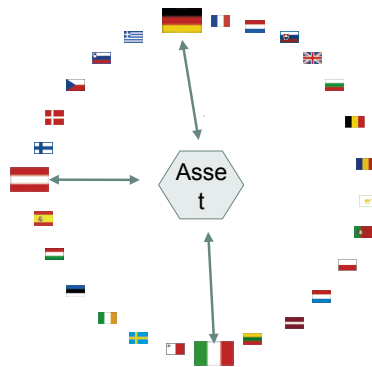


SEMIC.EU Support  
Interoperabilität erzeugen

- Coaching – Asset Entwicklung
- Clearing – Asset Optimierung
- Reusing – Asset Integration

21. April 2010

Verwendung von Interoperability Assets – Vorgehen

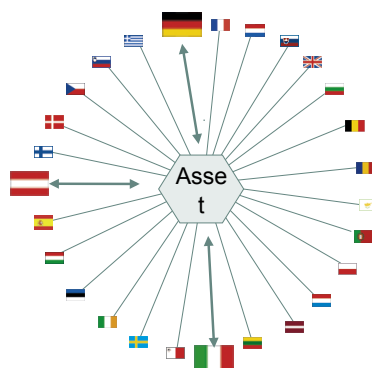


SEMIC.EU Coaching  
Services für eGovernment Projekte

- Koordination der “First Mover”
- Konzept- und Methoden-Support für das Interoperability Asset
- Begleitung des Roll out

21. April 2010

Verwendung von Interoperability Assets – Vorgehen



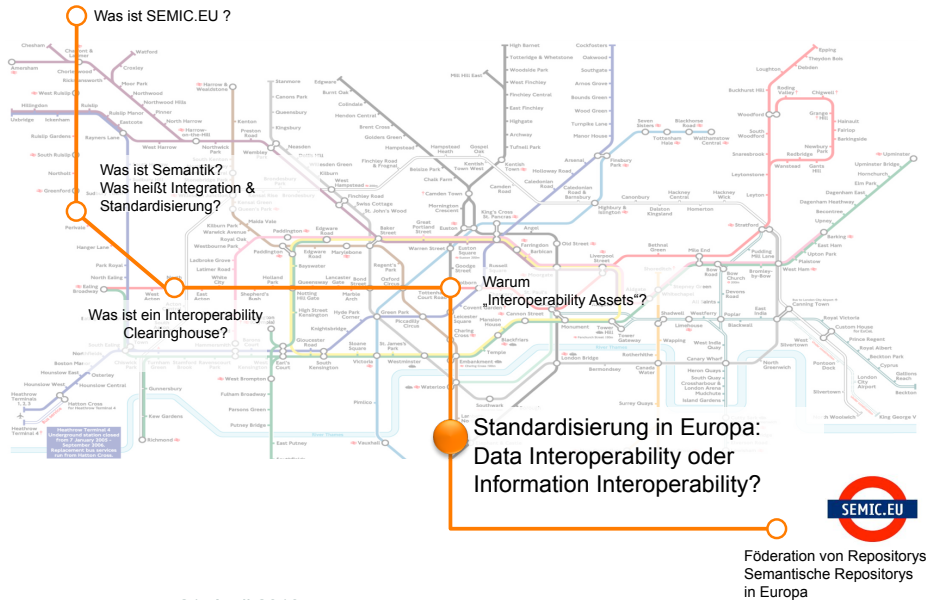
SEMIC.EU Coaching  
Services für eGovernment Projekte

- Koordination der “First Mover”
- Konzept- und Methoden-Support für das Interoperability Asset
- Begleitung des Roll out

21. April 2010

**]init[ Semantische Repositories in Europa**

Die Wegstrecke festlegen



21. April 2010

**]init[ Semantische Repositories in Europa**

SEMIC.EU – Lessons learnt

**Der Befund:**

Europa ist unendlich föderalistisch.

**Die Schlussfolgerung:**

Anstatt an Standardisierung zu verzweifeln, ist es sinnvoller, Formate zu schaffen, die nationale Datenmodelle unbeeinträchtigt lassen.

21. April 2010

**]init[** Semantische Repositorys in Europa



Gedankenexperiment 1:  
Müssen wir uns wirklich so anstrengen?



Interoperabilität, ja aber . . .

Was ist, . . .

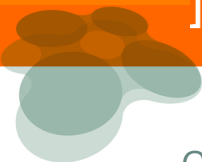
. . . wenn nicht alle immer mit allen Daten austauschen wollen,  
sondern nur Manche mit Manchen?

Was ist, . . .

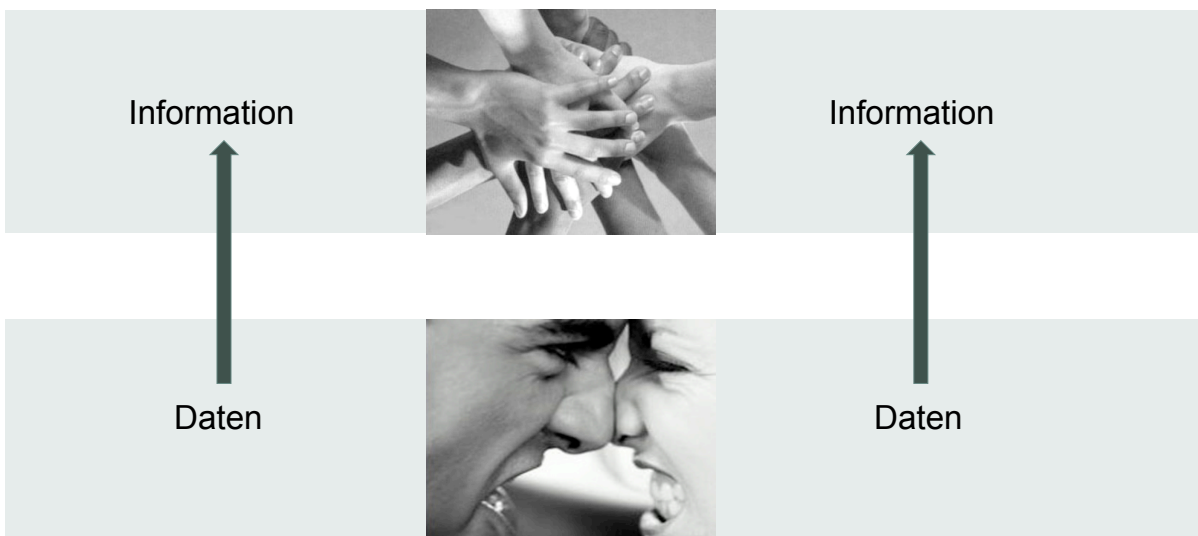
. . . wenn man Interoperabilität und Wirtschaftlichkeit wieder  
zusammen denkt?

21. April 10

**]init[** Semantische Repositorys in Europa

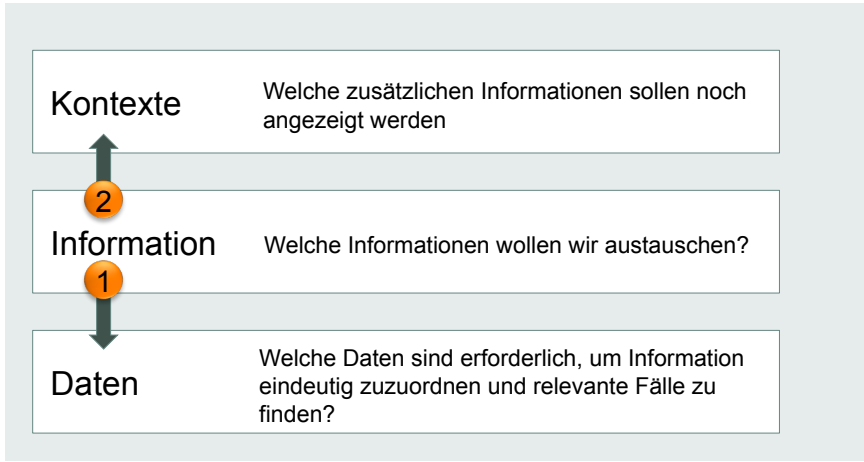


Gedankenexperiment 2:  
Können wir nicht einfach vom Zuhören lernen?



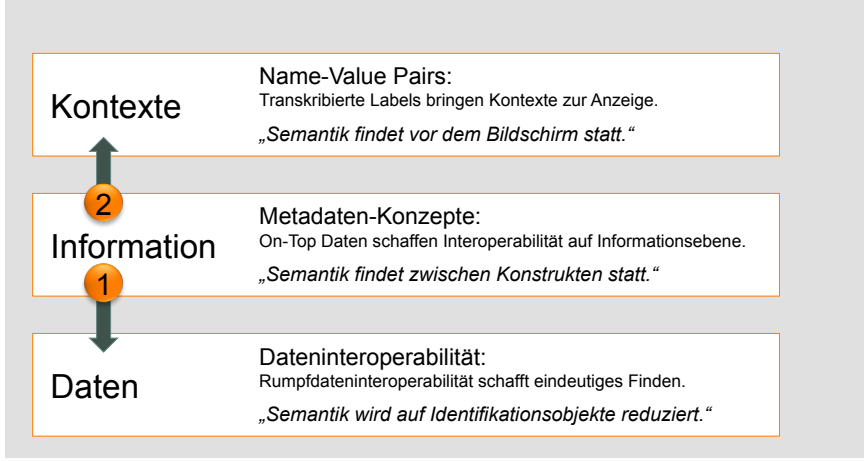
21. April 10

Gedankenexperiment 3:  
Das Latte Macchiato-Prinzip



21. April 10

Konzept der zielorientierten Umsetzung:  
Platz schaffen für das Wesentliche



21. April 10

**]init[** Semantische Repositories in Europa



Bedarfsorientierter Ausbau:  
Mit wem will ich – mit wem muss ich?



Entscheidbar machen:

- Wo reicht der Informationsaustausch?
- Mit wem ist Datenaustausch möglich?
- Wo ist Datenaustausch sinnvoll?

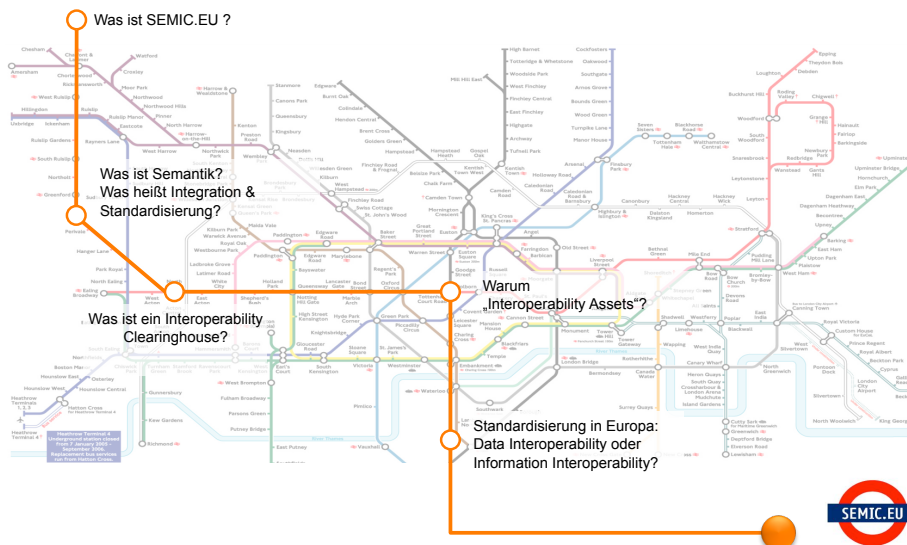
*Interoperabilität  
wirtschaftlich entscheidbar machen*

21. April 10

**]init[** Semantische Repositories in Europa



Die Wegstrecke festlegen



21. April 2010

Föderation von Repositories  
Semantische Repositories  
in Europa

]init[ Semantische Repositories in Europa



Zurück zum Repository



21. April 2010

21. April 10

]init[ Semantische Repositories in Europa



Was genau findet man bei SEMIC.EU ?

The screenshot shows the SEMIC.EU website interface. Red dashed boxes with arrows point to the following areas:

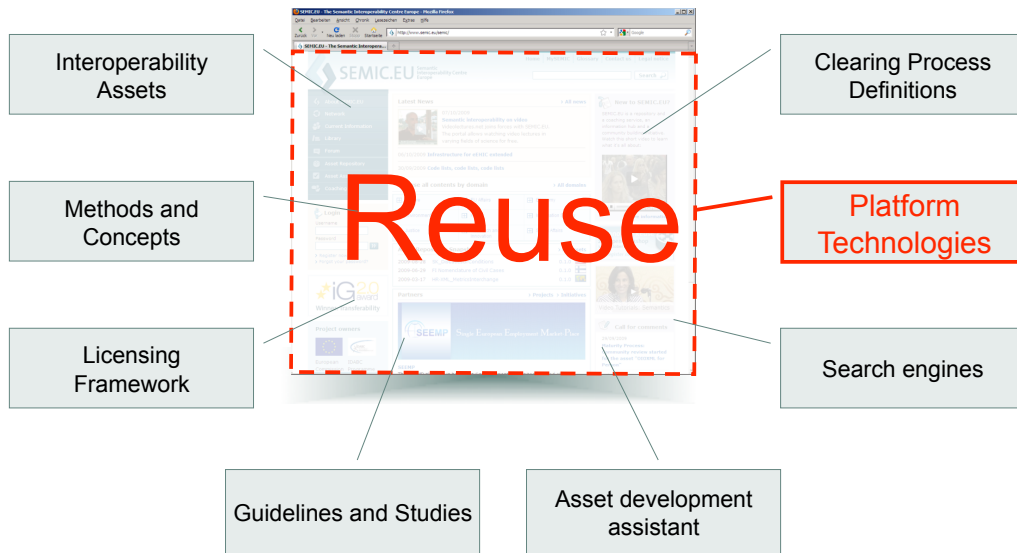
- Interoperability Assets** (labeled **Reuse**)
- Asset Clearing and development** (labeled **Reuse**)
- Community building** (labeled **Reuse**)
- XML Schema** (labeled **Reuse**)
- Code lists** (labeled **Reuse**)
- Taxonomies** (labeled **Reuse**)

21. April 2010

21. April 10

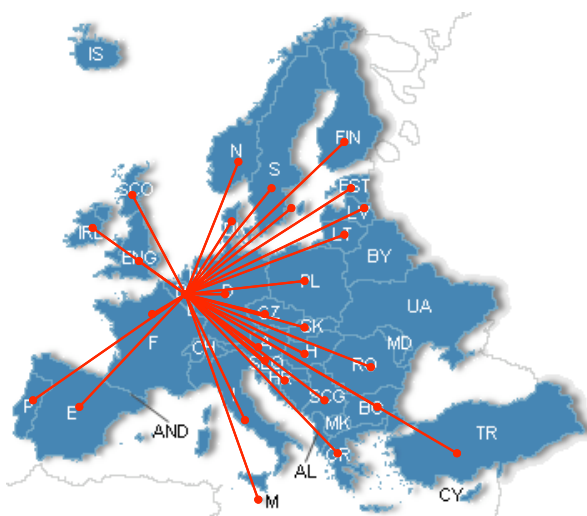


Wiederverwendung – Grundprinzip von Repositorys



21. April 2010

Vision – Die europäische „Föderation von Repositorys“



21. April 2010

1. Zentrale Partner & Interessenten
  - Deutschland
  - Finnland
  - Frankreich
  - Niederlande
  - Spanien
  - Türkei
2. Dezentrale Interessenten
  - Region Piemont (Italien)
  - BKA (Deutschland)
3. Federation Partner
  - CNIPA (Zentralregierung Italien)
  - FAO (United Nations)
  - NATO (Brüssel)



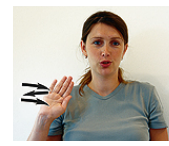


Vielen Dank für Ihre Aufmerksamkeit!



Dr. Renke Fahl-Spiewack  
]init[ AG für digitale Kommunikation

21. April 2010



Tschüss & Auf Wiedersehen

## Ein Standard für Registeranwendungen: die Register Factory®

Dr. Gerhard Pews, Ralf Leonhard, Stefan Brombach

Bundesverwaltungsamt  
Projektgruppe "Reengineering der Plattformen Innere Sicherheit" (RISP)  
50728 Köln  
register-factory@bva.bund.de  
gerhard.pews@capgemini-sdm.com

**Abstract:** In diesem Beitrag gehen wir auf die Frage ein, welche Standards ein Auftraggeber verlangen kann und verlangen sollte, ohne den Auftragnehmer unnötig einzuschränken. Dies wird für die Systeme im Bereich Innere Sicherheit des Bundesverwaltungsamtes (BVA) konkret beantwortet und dazu dargestellt, welche Erfahrungen in den letzten Jahren beim Bau von IT-Systemen nach diesen Standards gemacht wurden.

Dazu beschreiben wir zunächst, welche Ziele das BVA für seine Softwarelandschaft erreichen will und wo die existierenden Standards SAGA, V-Modell® XT und XÖV diese Ziele unterstützen und wo nicht. Daraus leiten wir weiteren Standardisierungsbedarf ab und zeigen, wie diese Standards im Rahmen der dabei entwickelten Register Factory konkret aussehen. Abschließend bewerten wir, den Einsatz dieser Standards durch zwei unterschiedliche Dienstleister beim Bau von mehreren größeren und kleineren IT-Systemen.

### 1 Ziele des Bundesverwaltungsamtes für Registeranwendungen

Register spielen in der öffentlichen Verwaltung eine zentrale Rolle. Das Bundesverwaltungsamt (BVA) betreut - neben anderen IT-Systemen – mehrere umfangreiche Registeranwendungen im Bereich der Inneren Sicherheit, darunter mit dem Ausländerzentralregister eines der großen Register in Deutschland.

Registeranwendungen bestehen zum einen aus Registern, die einen strukturierten Datenbestand verwalten und zum anderen aus Geschäftsanwendungen, die höherwertige Geschäftsprozesse auf Registern abbilden. Der Datenbestand der Register wird durch Meldungen verändert und kann durch Auskünfte abgefragt werden. Eine Besonderheit von Registern ist dabei der Umstand, dass auf ein Register zwar von sehr vielen Nutzergruppen zugegriffen werden darf, die Informationen aber, die für einen konkreten Nutzer sichtbar sind, unterscheiden sich deutlich und müssen durch ein umfangreiches Berechtigungsregelwerk gefiltert werden. Innerhalb der Geschäftsprozesse werden die Daten eines Registers abgefragt, editiert und oft komplexe Workflows abgebildet, deren Abarbeitung sich sogar über mehrere Behörden verteilen kann.

Die Ziele des BVA sind dabei einfach den Punkt gebracht: die Registeranwendungen sollen wie gewünscht funktionieren und sowohl kurz- als auch langfristig wirtschaftlich sein. Beispielsweise sollen alle funktionalen Anforderungen korrekt umgesetzt werden, die Verarbeitung effizient ablaufen (Performance, Durchsatz), die Sicherheit der Anwendung gegeben sein oder die Software wartbar und erweiterungsfähig sein.

### 2 Einsatz bestehender Standards

Zur Erreichung dieser Ziele bietet es sich an Standards zu verwenden. Dazu wurden im ersten Schritt die „Standards und Architekturen für E-Government (SAGA)“, das V-Modell® XT und XML in der öffentlichen Verwaltung (XÖV) herangezogen. Dabei stellte sich heraus, dass die Einhaltung dieser Standards zwar eine notwendige Voraussetzung für das Erreichen der Ziele war, diese aber bei weitem noch nicht ausreichend sind.

SAGA gibt zwar einen Rahmen für Produkte und Technologien vor, dieser Rahmen ist aber so weit gesteckt, dass in diesem Fall das Ziel der langfristigen Wartbarkeit nicht sichergestellt werden kann. Es ist möglich SAGA-konforme Anwendungen zu bauen, die einen komplett unterschiedlichen Technologiestack verwenden und für deren Wartung unverhältnismäßig hohe Kosten entstehen würden. Hier waren weitere Einschränkungen und Konkretisierungen notwendig.

Das V-Modell® XT ist in seiner allgemeinen Form universell einsetzbar und kann auf verschiedene Anwendungsszenarien angepasst werden (Tailoring). Dabei ist dafür zu sorgen, dass ein adäquates Tailoring sowohl die Erstellung einer hochwertigen Software fördert als auch für die Entstehung aller Dokumente sorgt, die im weiteren Lebenszyklus der Software zur Wartung und Weiterentwicklung benötigt werden. Gerade hier sind die allgemeinen Dokumentvorgaben des V-Modell® XT noch zu konkretisieren. Ein weiterer Aspekt ist, dass es sich bei der Registerplattform des BVA um ein Projektportfolio und eine konsistente Anwendungslandschaft mit gemeinsamen Architekturkonzepten handelt. Die Einhaltung dieser Konsistenz muss im Tailoring des V-Modell® XT noch vorgesehen werden.

XÖV ist als universelles Datenaustauschformat oft sehr komplex, insbesondere für die Verarbeitung innerhalb einer Anwendung. Eine entsprechende Konvertierung zwischen internen und XÖV-Format erscheint daher notwendig. Das Mapping auf die XÖV-Datenstrukturen ist aber für die externen Schnittstellen entsprechend zu dokumentieren.

### 3 Standards der Register Factory®

Es war daher nötig, einen weitergehenden Standard zu definieren. Ergebnis war und ist die Register Factory®: ein Konzept, mit dem Registeranwendungen effizient entwickelt und betrieben werden können. Dazu definiert die Register Factory® unter anderem auch Standards für die Entwicklung in Form von Referenzarchitekturen für

- die technische Architektur einer Anwendung und eingesetzte Produkte,
- die technische Infrastruktur, d. h. für Middleware, Betriebsplattform und Hardware sowie
- die fachliche Architektur einer Anwendung, d. h. den Schnitt in fachliche Systeme und Komponenten.

Neben diesen Architekturstandards definiert die Register Factory® Standards für das Vorgehen im Projekt, z. B. Vorlagen für Systemspezifikationen und Systementwürfe und Vorgaben, wie diese Standards gehandhabt werden.

#### **4 Erfahrungen mit den Register Factory® Standards**

Die Standards der Register Factory® wurden in Zusammenarbeit und im Auftrag des BVA mit dem Dienstleister Capgemini sd&m entwickelt. Bis heute wurden sechs größere IT-Systeme nach diesen Standards gebaut sowie zusätzlich fünf kleinere IT-Systeme (Querschnittssysteme). Die Umsetzung erfolgte in einem Projekt auch durch einen anderen Dienstleister, der nicht an der Erstellung der Standards beteiligt war.

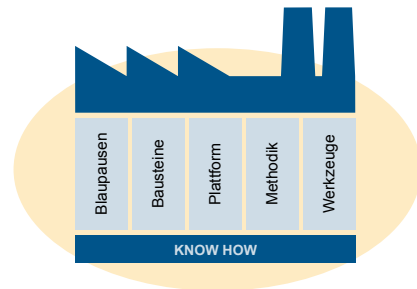
Die praktische Erprobung hat gezeigt, dass sich die Vorgaben bewährt haben. Ein kritischer Punkt ist es, dafür zu sorgen, dass die Vorgaben auch konsequent umgesetzt werden. Dazu müssen die Standards Teil der Ausschreibung und des Angebots sein und deren Einhaltung geprüft werden. Insbesondere in der Automatisierung der Überprüfung sehen wir noch Ausbaupotential. Weiterhin wird überprüft, einige der Standards auch über den eingegrenzten Bereich der Registeranwendungen hinaus für Entwicklungsprojekte im BVA zu verwenden.

Konzept und Komponenten der Register Factory® stellt das BVA auch anderen Behörden zur Verfügung.

## Ein Standard für Registeranwendungen: Die Register Factory



Stefan Brombach  
Dr. Gerhard Pews



## Vorstellung

**Stefan Brombach**  
*MSc - Dipl. Betriebswirt*  
*Projektleiter Register Factory*



- Bundesverwaltungsamt**
- zentraler Dienstleister des Bundes
  - über 100 Fachaufgaben
  - 2.300 Mitarbeiter an 10 Standorten
  - Bundesstelle für Informationstechnik

**Dr. Gerhard Pews**  
*Diplom-Informatiker*  
*Chefarchitekt AZR/Visa*  
*Geschäftsbereichsmanager*



- Capgemini sd&m**
- Leistungsfähige Prozess- und Softwarelösungen
  - für namhafte Kunden aller Branchen
  - 2.100 Mitarbeiter an 15 Standorten
  - Teil der Capgemini-Gruppe

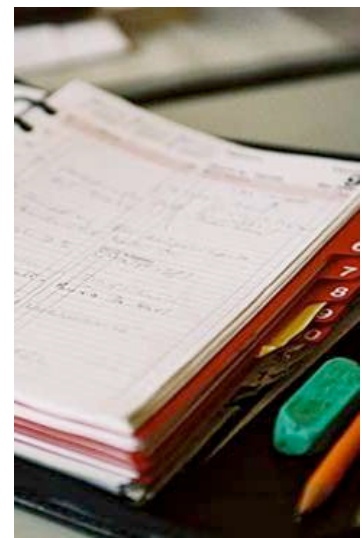
## Zusammenfassung

- Im Auftrag des BVA wurde durch Capgemini sd&m die Register Factory entwickelt, mit der sich effizient Registeranwendungen bauen lassen.
- Die Register Factory setzt weitergehende Standards für Registeranwendungen, die auf SAGA, V-Modell XT und XÖV aufsetzen und Konkretisierungen enthalten.
- Der Einsatz dieser Standards hat sich bewährt.



## AGENDA

- Registeranwendungen beim BVA
- Standards im öffentlichen Bereich
- Standards der Register Factory
- Zusammenfassung





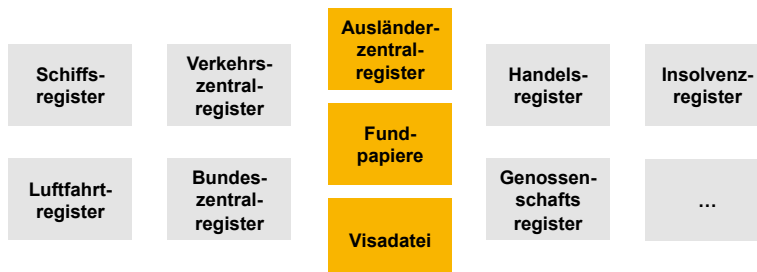
## AGENDA

- **Registeranwendungen beim BVA**
  - Standards im öffentlichen Bereich
  - Standards der Register Factory
  - Zusammenfassung



## Was ist ein Register?

Ein Register ist ein strukturiertes Verzeichnis und damit eine Form der standardisierten Dokumentation von Daten, die ein bestimmtes Merkmal verbindet.



→ Leistungsfähige Register sind die Basis einer erfolgreichen Verwaltung



## Register sind besonders anspruchsvolle IT-Anwendungen ...

### ▪ Komplexe fachliche Aufgaben

- Regelwerke für Berechtigung und Validierung
- Unschärfe Suchen (Phonetik + Biometrie)
- Nachvollziehbarkeit, Protokolle
- Beteiligung von Partnern

### ▪ Aufgabenstellungen

- Aufbau neuer Register
- Anpassung bestehender Register

### ▪ Schnelle Umsetzung

- Gesetzliche Änderungen zum Stichtag



### ▪ Herausforderungen

- Fachlicher + technischer Wandel
- Wirtschaftlichkeit
- Stabiler Betrieb
- Wartbarkeit

### ▪ Hohe Betriebsanforderungen

- Datenqualität
- Datensicherheit
- Verfügbarkeit
- Durchsatz (große Nutzerzahlen/ Datenmengen)

### ▪ Rahmenbedingungen

- Konformität zu Vorgaben und Standards (SAGA, BSI u.a.)

## ... mit dennoch ähnlichen Anforderungen

Fachlich		Technisch	IT-Betrieb	Methodik
Meldung	Mitteilung	Fehlerbehandlung	24/7-Verfügbarkeit	Projektvorgehen
Auskunft	Fristenkontrolle	Persistenz	Monitoring	Fachliche Modellierung
Suche	Auswertung	Dialogsteuerung	Sicherheit	Technische Modellierung
Protokollierung	...	Batch	...	Hilfsmittel
Datenpflege		...		...



## Anforderungen und Ziele

- Reengineering der Registerlandschaft im BVA
- Zukunftssicherheit
  - moderne, flexible, service-orientierte Architektur
  - gewachsene und neu anstehende Anforderungen effizient abbilden
  - Plattformunabhängigkeit
- Einfache Wartbarkeit
  - neue Anforderungen schneller und günstiger umsetzen
- Betriebssicherheit

→ Standardisierung und Modularisierung



## AGENDA

- Registeranwendungen beim BVA
- **Standards im öffentlichen Bereich**
- Standards der Register Factory
- Zusammenfassung



DEUTSCHLAND  
ONLINE



## Relevante Standards

- Für den Bau von Registeranwendungen sind die Standards SAGA, V-Modell und XÖV besonders relevant.
- Die Nutzbarkeit dieser Standards wird im Folgenden betrachtet.

### Standards

SAGA

V-Modell XT

XÖV

Seite: 11

## Erfahrungen beim Einsatz von SAGA

### Standards

SAGA

V-Modell XT

XÖV

### Erfahrungen

- Die Vorgaben von SAGA sind hilfreich, insbesondere für:
  - Infrastruktur/Zonen
  - Einschränkung der Produkt-/Technologievelfalt.
- Die Technologie-Festlegungen lassen noch große Freiräume.
- Durch den Ursprung E-Government via Internet sind native Clients nur grob umrissen.

→ Insbesondere für langfristige Wartbarkeit und nachhaltige Kostenkontrolle sind weitere Einschränkungen und Detaillierungen nötig.

Seite: 12

## Erfahrungen beim Einsatz des V-Modells

### Standards

SAGA

V-Modell XT

XÖV

### Erfahrungen

- Das V-Modell XT stellt einen gut nutzbaren Vorgehensrahmen dar.
- Das V-Modell hat noch Verbesserungspotential
  - Insbesondere in der „XT“-Version ist das V-Modell sehr komplex.
  - Die Dokumentvorlagen sind z. T. sehr generisch und bieten daher nur geringe Unterstützung.
  - Die Steuerung einer projektübergreifenden Gesamtarchitektur ist nicht beschrieben.

→ In der Nutzung des V-Modells für Registeranwendungen sind noch weitere Detaillierungen nötig.

## Erfahrungen beim Einsatz von XÖV

### Standards

SAGA

V-Modell XT

XÖV

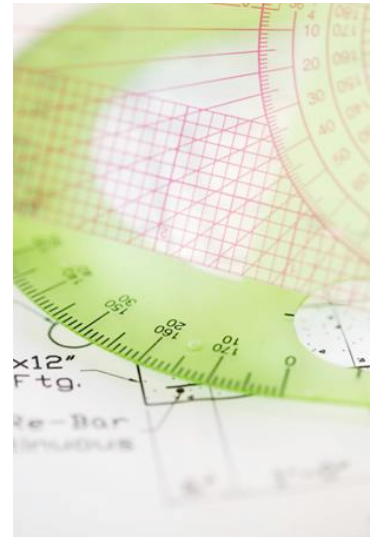
### Erfahrungen

- XÖV gibt (unter anderem) universelle Austauschformate für bestimmte Daten vor
- Universelle Formate tendieren dahin, komplex zu werden, dies liegt in der Natur der Sache.

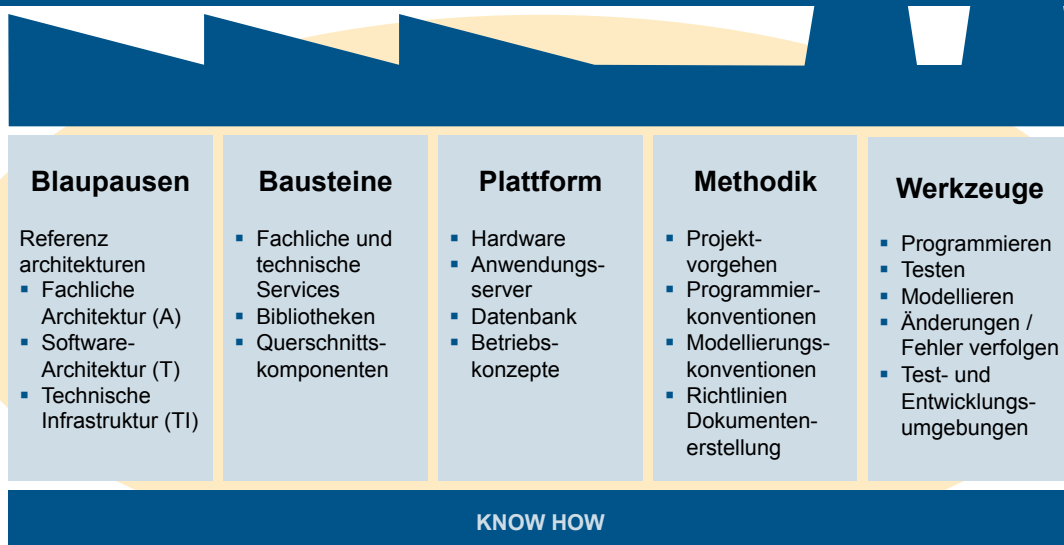
→ Es muss ein Weg gefunden werden, wie mit der Komplexität der Austauschformate umgegangen werden kann.

## AGENDA

- Registeranwendungen beim BVA
- Standards im öffentlichen Bereich
- **Standards der Register Factory**
- Zusammenfassung



## Die Säulen der Register Factory



## Verfeinerung der SAGA-Architekturvorgaben

### Standards

SAGA

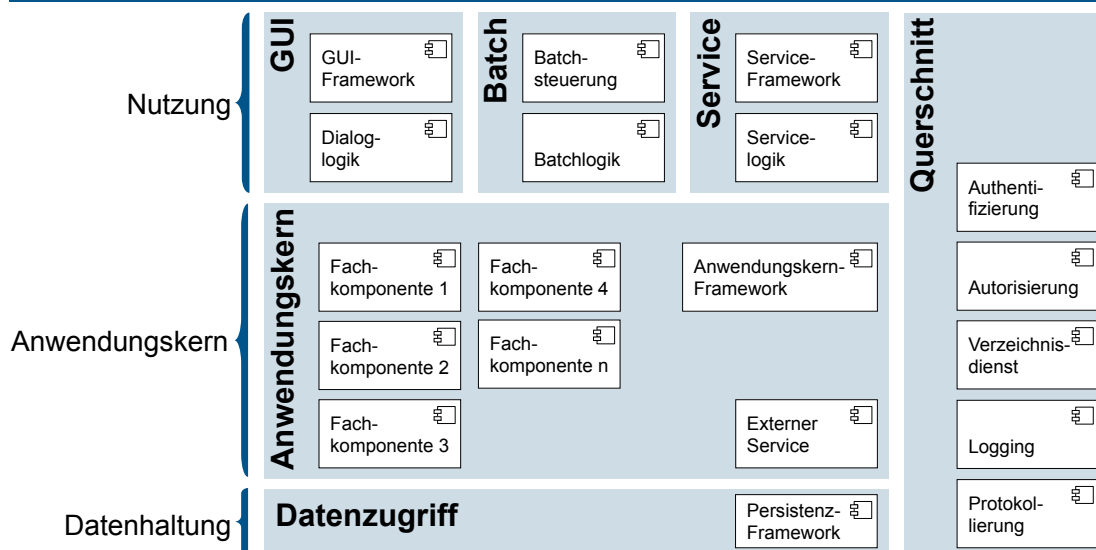
V-Modell XT

XÖV

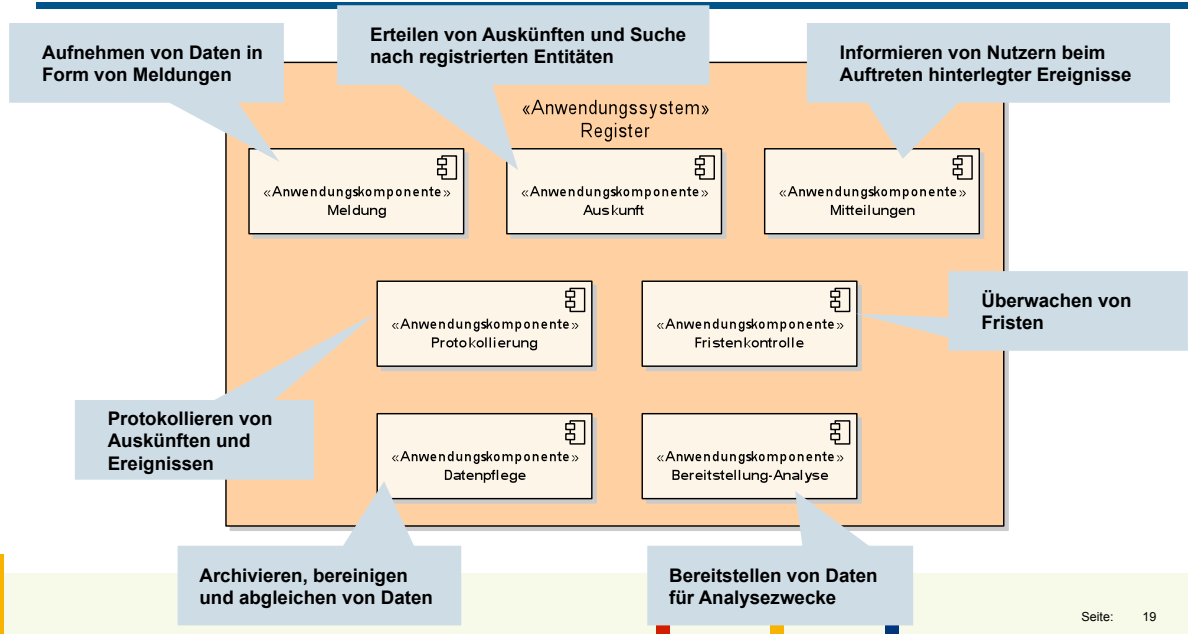
Die Vorgaben von SAGA werden in der Register Factory verfeinert und konkretisiert

- Verkleinerung des Produktkatalogs
- Vorgabe konkreter Referenzarchitekturen:
  - Technische Infrastruktur
  - Technische Softwarearchitektur
  - Fach-Architektur
- Konkretisierung der Vorgaben für eine Native GUI

## Technische Referenzarchitektur



## Fachliche Referenzarchitektur für Register



## Umgang mit dem V-Modell

### Standards

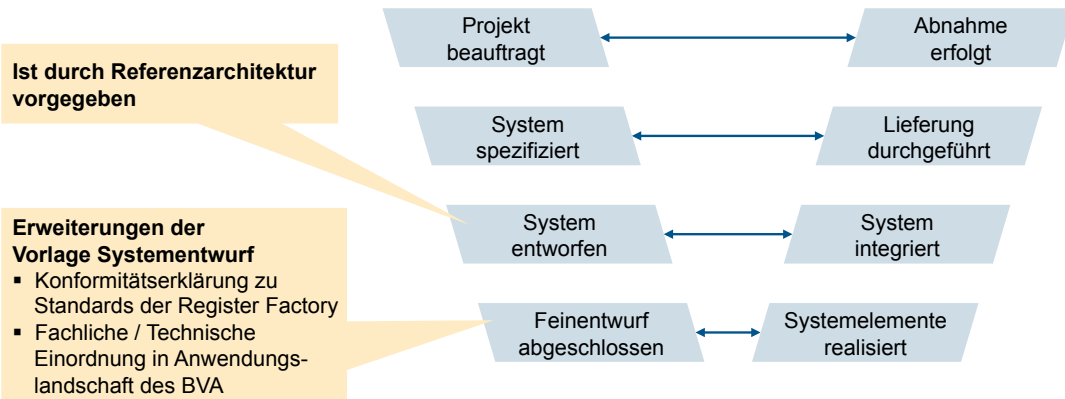
SAGA

V-Modell XT

XÖV

- Tailoring: Verzicht auf Tailoring für viele Fälle
  - Geringe fachliche und technische Risiken, da Referenzarchitekturen vorhanden
  - Registeranwendungen sind in vielen Fällen reine Datenpflege-Anwendungen, geringe Komplexität in der GUI
  - Größere Komplexität nur in Prozessanwendungen
- Dokumentvorlagen
  - Konkretisierung der Dokumentvorlagen, insbesondere Systementwurf
    - Einpassung in Systemlandschaft
    - Konformität zu Architektur- und Produktvorgaben
    - Fortschreiben zu Systemdokumentation

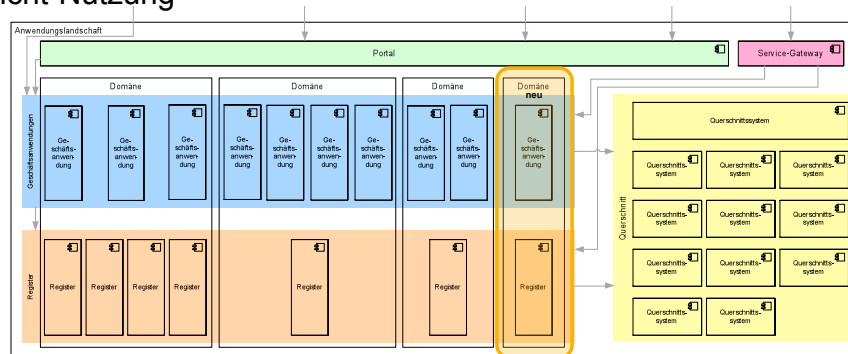
## Methodik: Vorgehen auf Basis von V-Modell® XT



## Methodik: Vorgehen auf Basis von V-Modell® XT

### Einpassung zu Standards der Architektursteuerung:

- Fachlich/Technische Einordnung in Anwendungslandschaft
- Liste aller Bausteine und Architekturvorgaben mit Erklärung des Projekts zu Nutzung bzw. Nicht-Nutzung



## Vom Systementwurf zur Systemdokumentation

- Systementwurf ist Arbeitsdokument im SW-Entwicklungsprozess
- Enthält sowohl Arbeitsvorgaben für die Entwickler als auch dokumentierende Anteile
- Arbeitsvorgaben werden nach Systemerstellung entfernt

Arbeitsvorgaben	Dokumentation
<ul style="list-style-type: none"> <li>▪ Implementierungsvorlagen</li> <li>▪ Klassen/Methodendesign</li> </ul>	<ul style="list-style-type: none"> <li>▪ Einordnung in Systemlandschaft</li> <li>▪ Schnitt in Subsysteme und Komponenten</li> <li>▪ Besondere Architekturpatterns und Detaillösungen</li> <li>▪ Logische Hardware-Architektur</li> </ul>

## Umgang mit XÖV

### Standards

SAGA

V-Modell XT

XÖV

### Interne Verarbeitung

- XÖV-Standard-Datenstrukturen oft zu komplex für Nutzung als interne Datenrepräsentation.
- Konvertierung XÖV  $\leftrightarrow$  internes Format in den Service-Komponenten der jeweiligen Anwendungen

### Schnittstellendokumentation

- Dokumentation der Semantik der Schnittstellen in leichter verständlichem Format, in der Regel angelehnt an das interne Verarbeitungsformat.
- Anschließend Dokumentation des Mappings auf XÖV-Datenstrukturen



## AGENDA

- Registeranwendungen beim BVA
- Standards im öffentlichen Bereich
- Standards der Register Factory
- **Zusammenfassung**



Seite: 25

## Erfahrungen

- Register Factory verfeinert öffentliche Standards als Richtlinie
- Bestandteil von Ausschreibung / Angebot um Einhaltung zu gewährleisten
- Verantwortlichkeit Architekturboard
  - Einhaltung oder Abweichung von den Standards
  - Weiterentwicklung der Standards
- Die Register Factory hat sich im Praxiseinsatz bewährt
  - 6 Systeme erfolgreich umgesetzt
  - Weiterentwicklung / Verfeinerung der Register Factory
  - Anwendung durch unterschiedliche Dienstleister



Seite: 26

**Vielen Dank für Ihre Aufmerksamkeit!**



[www.register-factory.de](http://www.register-factory.de)

Seite: 27


## Kontakt


**Stefan Brombach**



**Bundesverwaltungsamt**  
50728 Köln

 0228 99 / 358 - 3547


 stefan.brombach@bva.bund.de

 www.bundesverwaltungsamt.de

**Dr. Gerhard Pews**



**Capgemini sd&m AG**  
Berliner Str. 76  
63065 Offenbach

 069 / 82901 - 236

 gerhard.pews@capgemini-sdm.de

 www.de.capgemini-sdm.com

 [www.register-factory.de](http://www.register-factory.de)

Seite: 28

## 8.6 Ein Standard für Registeranwendungen: die Register Factory

# 9. Einführung und Nutzung von ITIL und CMMI

Sessionüberblick

---

9.1. ITIL bei der Optimierung von Serviceprozessen . . . . .	511
9.2. ITIL richtig gemacht . . . . .	527
9.3. Einführung von CMMI unter agilen Gesichtspunkten . . . . .	539

---



## **ITIL bei der Optimierung von Serviceprozessen: Qualität von Produkten und Services**

Folkert Jung

networks direkt GmbH  
Griegstraße 75  
22763 Hamburg  
Folkert.Jung@direkt-gruppe.de

**Abstract.** Zentrales Anliegen von ITIL ist es, IT-Prozesse beherrschbar zu machen. Es kommt darauf an, das Ausmaß und die Richtung von Änderungen der IT-Landschaft proaktiv zu steuern. Prozessmodelle wie ITIL liefern Vorbilder, an denen Unternehmen ihre Änderungsvorhaben ausrichten können.

Das gilt insbesondere dort, wo es um die Qualität von Produkten und Services geht. Folkert Jung (direkt gruppe) beschreibt anhand von Beispielen, welche Bedeutung ITIL bei der Optimierung von Serviceprozessen hat und wie auf Basis von ITIL eine erhebliche Verbesserung der Stabilität des Rechenzentrumsbetriebes herbeigeführt wurde.

### **1 Ausgangssituation und Projektziel**

Zunächst werden die Ausgangssituation und die Projektziele dargestellt. Mit der Einführung der Service-Prozesse sollen die Stabilität des Rechenzentrumsbetriebs verbessert, Störungen sowie Ausfall- und Fehlerbehebungszeiten verringert und die Anwenderzufriedenheit gesteigert werden.

Dazu wird eine Prozessorganisation neben der bestehenden Aufbauorganisation eingeführt. Gleichzeitig ist für eine durchgängige Toolunterstützung der Prozesse durch eine Configuration Management Database (CMDB) zu sorgen. Als Projektergebnis sollen alle Prozesse in akzeptiert und verbindlich gelebt werden.

Folgende Faktoren sind von besonderer Beachtung: Die Praxisorientierung der gesamten Prozessetablierung hat oberste Priorität. Neben den ITIL-Standards werden die spezifischen Rahmenbedingungen angemessen berücksichtigt und fließen in die Modellierung der Prozesse mit ein. Die beteiligten Mitarbeiter werden frühzeitig integriert. Dieses Vorgehen förderte die Akzeptanz der organisatorischen Veränderungen.

Eine durchgängige Toolunterstützung soll den effizienten Einsatz der Prozesse sicherstellen. Als zentraler Datenlieferant für alle Prozesse wird eine hochgradig automatisierte CMDB geplant und eingeführt.

## 2 Projektvorgehen und -ablauf

Der Projektplan erläutert das gewählte Vorgehen im Überblick. Anhand eines Big Pictures wird das Zielszenario dargestellt. Ein weiterer Aspekt ist die Übersicht der zu implementierenden Prozesse und deren Zusammenhänge.

Die Projektplanung erfolgt in Anlehnung an das Plan-Do-Check-Act-Modell (PDCA) gemäß ITIL. Das Projekt verläuft für alle Prozesse gleich. Zuerst modelliert das Team die Abläufe auf Basis der in ITIL beschriebenen Prozess-Schritte und der Ergebnisse der Ist-Aufnahme. Anschließend werden diese Prozesse um die spezifischen Anforderungen und die Erfahrungswerte erweitert und dokumentiert.

## 3 Prozess-Implementierung

Die Etablierung des Prozesses Configuration Management enthält neben diesen Schritten auch die Konzeption und Implementierung der CMDB. Ein innovativer und unkonventioneller Ansatz: Die Erweiterung der vorhandenen Workflow-Lösung um zusätzliche Tabellenstrukturen für die CMDB. So entsteht ein durchgängiges IT-Service-Management-System.

Die CMDB bildet den Dreh- und Angelpunkt aller Prozesse im ITIL-Modell und erfordert eine durchdachte Systemkonzeption und -entwicklung. Das Projektteam veranstaltet deshalb Workshops zur Systemanalyse und zum Softwaredesign. Automatisierte Inventarisierungstools minimieren den Pflegeaufwand. Die Möglichkeit, Servicebäume zu modellieren, schafft die Voraussetzung, die gesamte IT-Infrastruktur logisch abzubilden.

## 4 Wirksamkeitsüberprüfung

Die Wirksamkeitsüberprüfung dient dazu, um zu ermitteln, ob die Projektziele erreicht wurden.

Das Projektteam führt zur systematischen Bewertung der Projektergebnisse in der Abschlussphase Interviews durch mit Prozessmanagern, -Ownern, Bearbeitern und Service-Verantwortlichen. Das Team bewertet sämtliche Prozesse in standardisierten Reifegrad-Assessments, die an das Capability Maturity Model Integration (CMMI) angelehnt sind. Eine SWOT-Analyse fasst die Ergebnisse zusammen, ein Maßnahmenkatalog liegt vor. Der Vortrag schließt mit der Formulierung eines Fazits und einer Übersicht, wie gelebte Prozesse erzielt werden.

SEE 2010 Vortrag 27  
**ITIL bei der Optimierung von Serviceprozessen**  
**Qualität von Produkten und Services**

Folkert Jung, Wolfram Herz  
Köln, 2010



Agenda



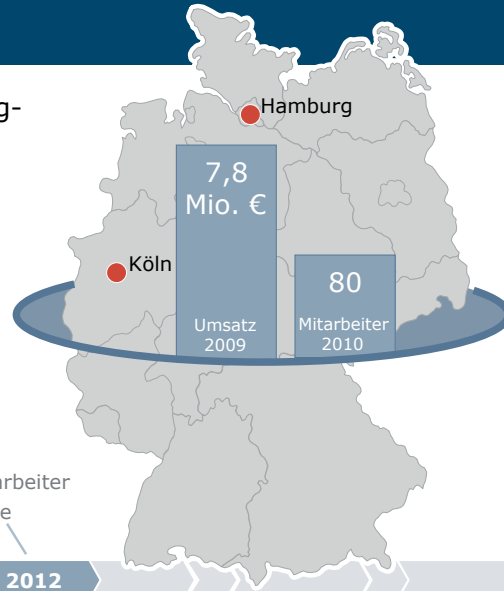
- 1 Kurzvorstellung
- 2 Ausgangssituation und Projektziel
- 3 Projektvorgehen und -ablauf
- 4 Prozess-Implementierung
- 5 Wirksamkeitsüberprüfung



## Die Unternehmensgruppe

### Die direkt gruppe ...

- ... bietet übergreifende IT- und Marketing-Services aus einer Hand
- ... verbindet die umfassende Beratung eines Gesamtdienstleisters mit der Flexibilität, Kundennähe und Qualität eines Spezialanbieters



## Die Unternehmensgruppe

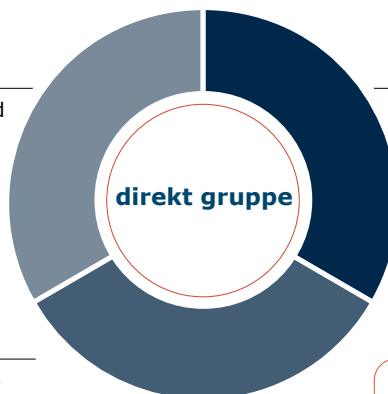
### Drei Dienstleister – eine Einheit

#### networks direkt

- Beratung, Implementierung und Betrieb komplexer IT-Infrastrukturen

#### solutions direkt

- Entwicklung kundenspezifischer Anwendungslösungen
- Neuentwicklung, Redesign und Portierung von Anwendungen für IT-Prozesse und Geschäftsabläufe



#### marketing solutions direkt

- Full-Service für Online-Marketing und -kommunikation
- Statistische Auswertung
- Marketingberatung
- Online-Schulungssysteme
- Anwendungsentwicklung

- ✓ Alle Unternehmenseinheiten arbeiten als Spezialisten oder als vernetzte Gesamtdienstleister
- ✓ Die individuelle Abstimmung der Service-Komponenten für jede Aufgabe sichert die Qualität und den Erfolg der Leistungen

## Die BGW

### Positionierung und Kurzdarstellung

- Berufsgenossenschaft für Gesundheitsdienst und Wohlfahrtspflege
- gesetzliche Unfallversicherung für nichtstaatliche Einrichtungen in Gesundheitsdienst und Wohlfahrtspflege
- Aufgaben:
  - Verhütung von Arbeitsunfällen, Berufskrankheiten und arbeitsbedingten Gesundheitsgefahren
  - Im Schadensfall Rehabilitation und Entschädigung
- Körperschaft des öffentlichen Rechts
- über 6 Millionen Versicherte in ca. 560.000 Unternehmen
- 12 Standorte, Hauptsitz: Hamburg

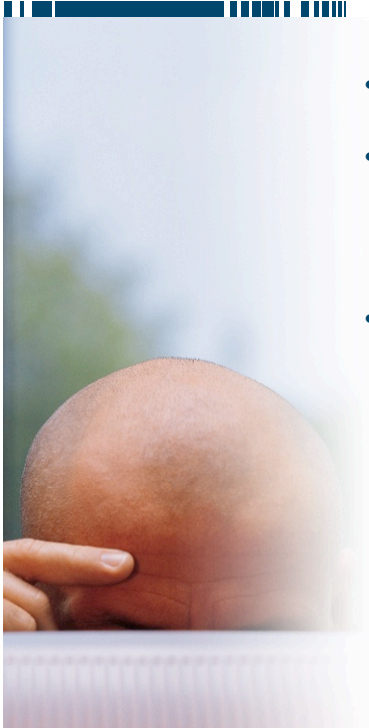


## Agenda



- 1 Kurzvorstellung
- 2 Ausgangssituation und Projektziel
- 3 Projektvorgehen und -ablauf
- 4 Prozess-Implementierung
- 5 Wirksamkeitsüberprüfung

## Ausgangssituation und Projektziel



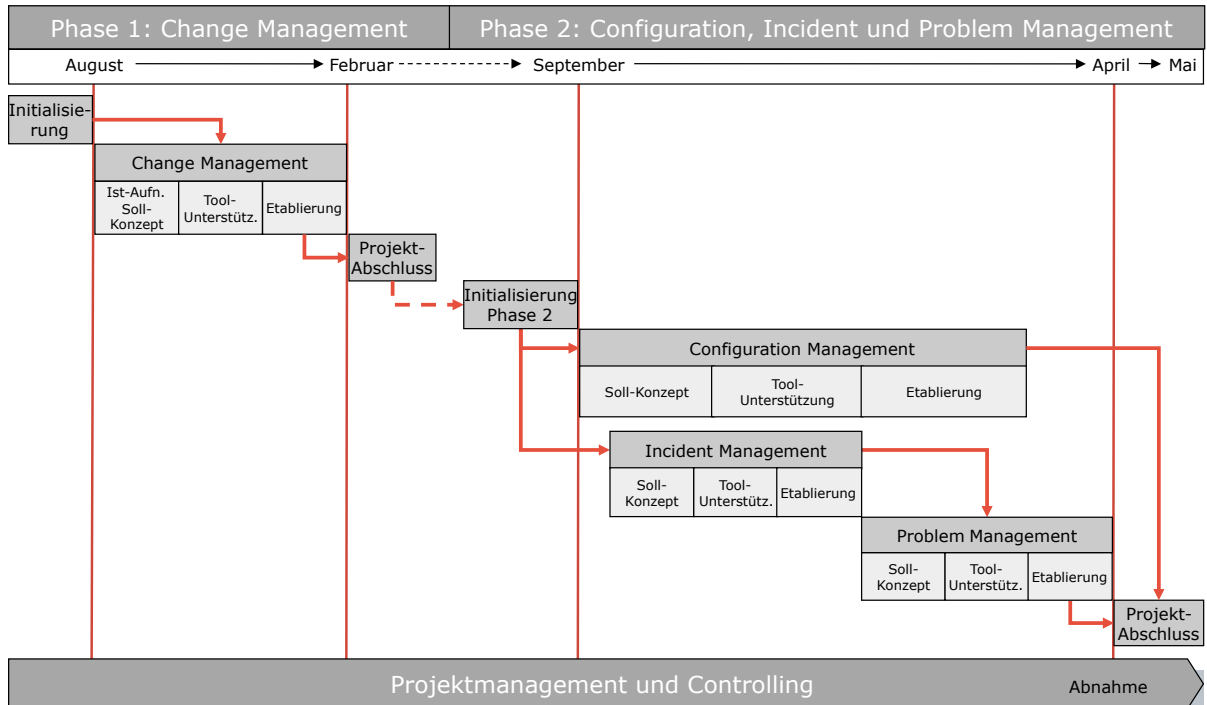
- IT-Dienstleistungen ausgerichtet an Geschäftsprozessen des Unternehmens
- Ausrichtung an ITIL-Best-Practices
  - vollständige Implementierung von ITIL noch nicht erreicht
  - ITIL-orientierte Change Management-Einführung umgesetzt
- Projektziele
  - Optimierung der Serviceprozesse
  - Verbesserung der Stabilität des Rechenzentrumsbetriebs
    - durch eine Reduzierung von Störungen sowie Ausfall- und Fehlerbehebungszeiten und
    - damit eine Verbesserung der Anwenderzufriedenheit

## Agenda

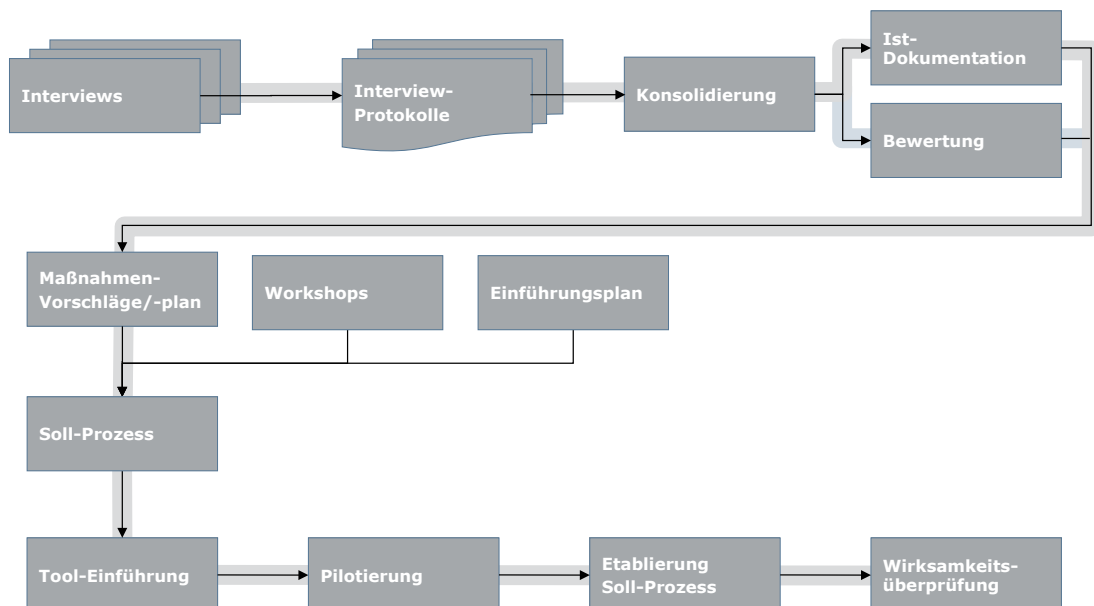


- 1 Kurzvorstellung
- 2 Ausgangssituation und Projektziel
- 3 **Projektvorgehen und -ablauf**
- 4 Prozess-Implementierung
- 5 Wirksamkeitsüberprüfung

## Projektvorgehen und -ablauf: Zeitplan



## Projektvorgehen und -ablauf: Ergebniskette

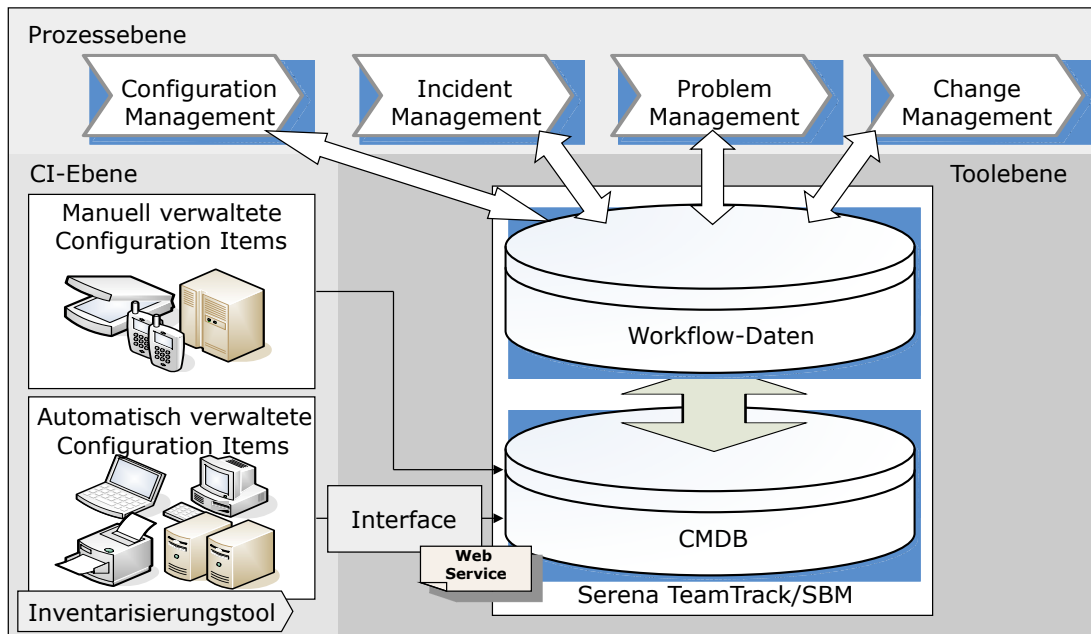


## Agenda

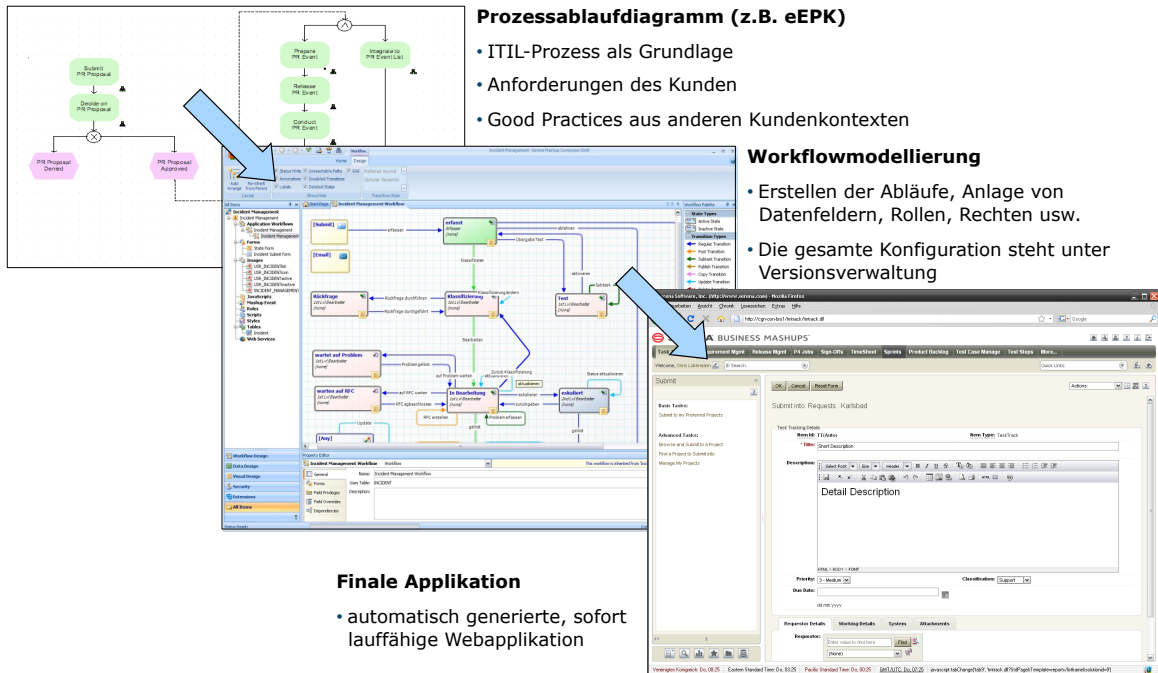


- 1 Kurzvorstellung
- 2 Ausgangssituation und Projektziel
- 3 Projektvorgehen und -ablauf
- 4 Prozess-Implementierung
- 5 Wirksamkeitsüberprüfung

## CMDB Big Picture

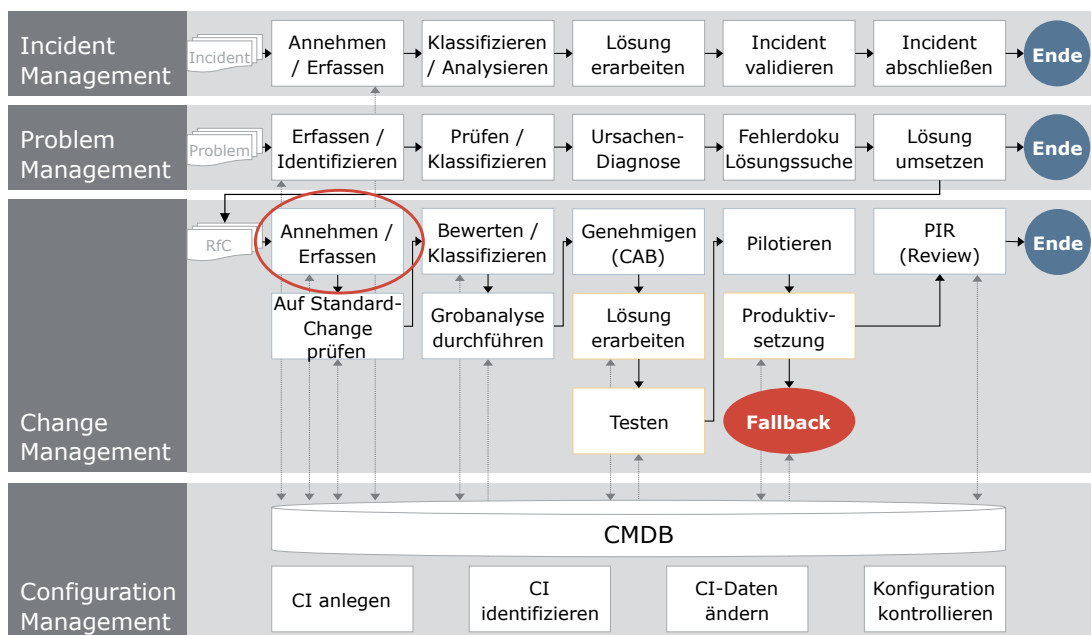


## Vom Workshop zur Toolimplementierung



© direkt gruppe 2010 | 21.04.2010 | SEE2010 - ITIL bei der Optimierung von Serviceprozessen

## Übersicht der Prozesse (verdichtet)



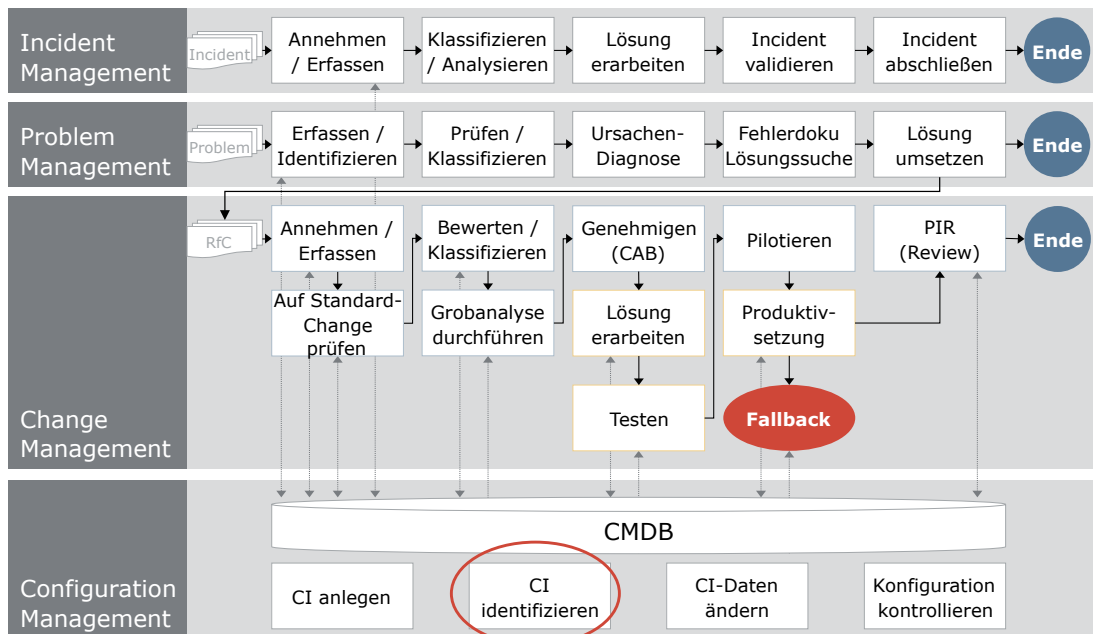
© direkt gruppe 2010 | 21.04.2010 | SEE2010 - ITIL bei der Optimierung von Serviceprozessen

Seite 14

## Beispiel Erfasster Change

The screenshot shows the Serena Business Mashups interface for a Change Management ChangeRequest. The main content area displays a status flow diagram with a step labeled 'Erfassung' (Recording) by 'admin' on '11.03.2009 16:29:51'. Below this, there are sections for 'Allgemeine Änderungsinformationen' (General Change Information) and 'Abhängige Vorgänge' (Dependent Processes). The 'Abhängige Vorgänge' section lists several related incidents, such as 'Posted from: INC-4000002: Druckauftrag wird zu langsam bearbeitet (gelöst)'. Callouts on the right side of the screenshot identify the 'Statusverlauf' (Status Flow) and 'Abhängige Objekte (z.B. Incidents)' (Dependent Objects, e.g., Incidents).

## Übersicht der Prozesse (verdichtet)



## Beispiel CMDB Objekt



The screenshot displays the SERENA BUSINESS MASHUPS interface. At the top, there are navigation tabs for Incident Mgmt, Problem Mgmt, Change Mgmt, CMDB, and Service Catalog. The main content area is divided into several sections: 'Statusverlauf', 'Interne Kennzeichnungen' (with fields for Item ID, Asset Name, Asset Tag, Seriennummer, Kosten, Kontakte, and Owner), 'Produkt Informationen' (with fields for Hersteller and Produkt), 'Technische Informationen' (with fields for IP Adresse, Standort, and Passwort), and 'Wartung' (with fields for Anschaffungsdatum, Ablaufdatum, and Active/Inactive status). At the bottom, there is a table with columns for 'Vorfalldatum', 'Kurzbeschreibung', 'State', and 'erfasst am'. The table contains three rows of data.

Aktionen auslösen  
(wie z.B. Incident erstellen)

Asset Detail-  
informationen

Anzeige von  
• abhängigen  
Services  
• gemeldeten  
Incidents, Problems

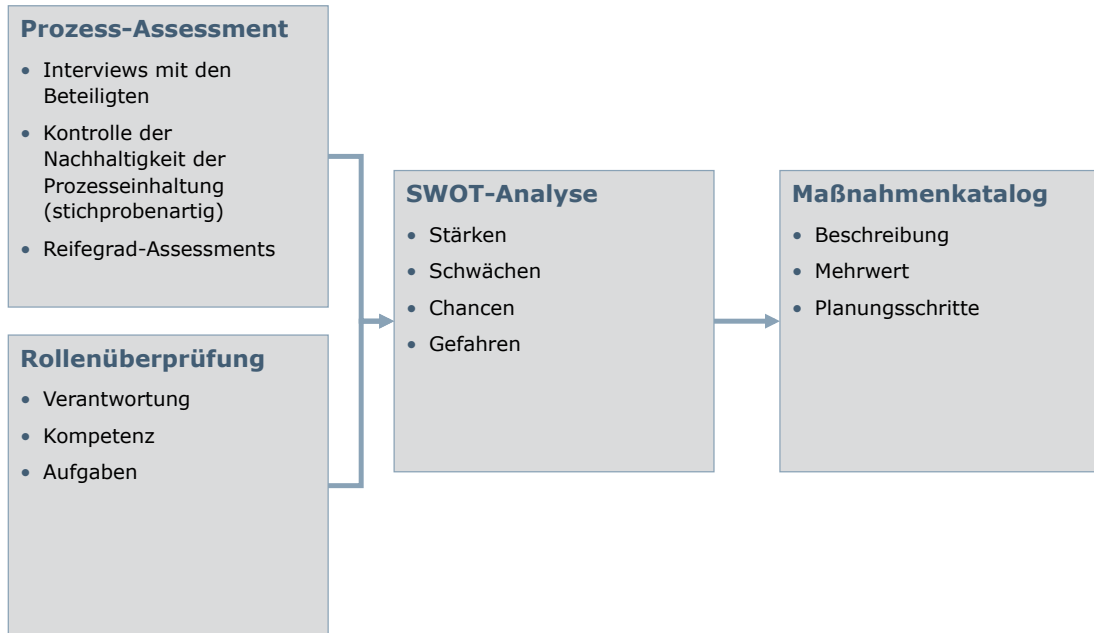
## Agenda



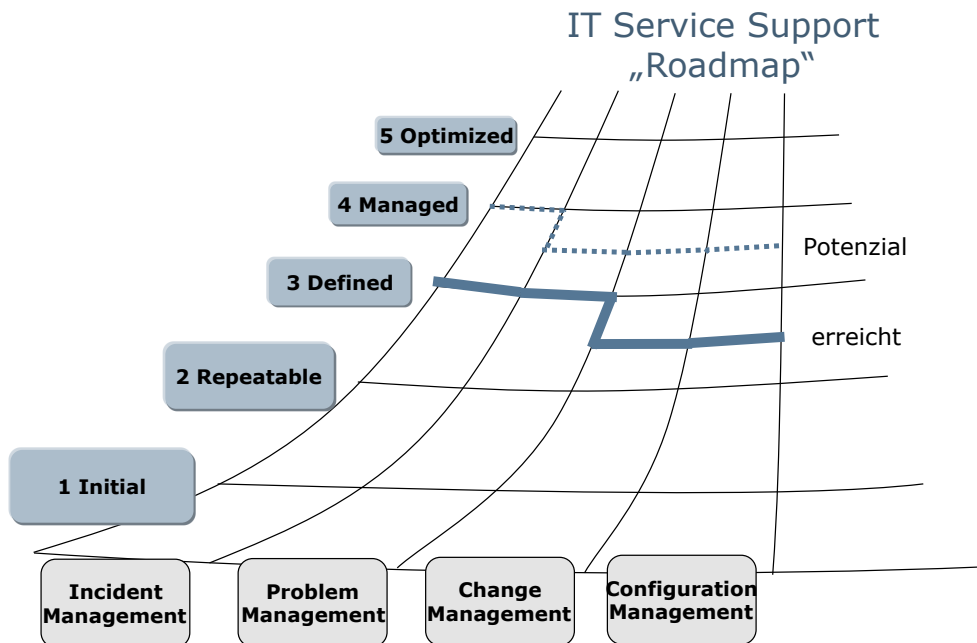
- 1 Kurzvorstellung
- 2 Ausgangssituation und Projektziel
- 3 Projektvorgehen und -ablauf
- 4 Prozess-Implementierung
- 5 Wirksamkeitsüberprüfung



## Methodenbaukasten zur Wirksamkeitsüberprüfung



## Ergebnisse Prozess-Assessment



## Ergebnisse des Reviews

### Erreichte Ergebnisse

- Stabilität und Performance des Service Managementtools
- Kontinuierliche Statusmeldungen für den Kunden
- Professionelle Wahrnehmung der Prozesse
- Integrität der Service-Support-Prozesse
- Unbürokratische Wahrnehmung der Prozesse
- Klarheit über die eigene Rolle in Bezug auf Verantwortung, Kompetenz und Aufgaben
- Schnelle Durchlaufzeit der Vorgänge im 2nd Level Support
- Schnelle Anpassbarkeit des Service Managementtools

### Handlungsfelder Zukunft

- Verbesserung der Incidentannahme und Erstbearbeitung M04, M05
- Anpassung Incident Management zur Bearbeitung von Service Requests M06, M35
- Nutzung der Knowledge Base M03, M13
- Kampagne zur Verbesserung der Datenqualität in der CMDB M11, M12, M27, M28
- Kunden- (Fachbereichs-)integration M14, M33

## Fazit



### „Lebbarkeit“ der Prozesse als oberste Maxime

- Immer im Augenmerk bei der Projektarbeit und die Entwicklung der Prozesse
- Maximaler Reifegrad ist nicht das Ziel.

“  
*ITIL ja, aber nur soweit, wie wir uns einen lebbareren Prozess vorstellen können bzw. sich dieser realisieren lässt, ohne die Organisation neu zu strukturieren.*  
 ”

Projektleiter der BGW

## Was sich vermeiden ließ ...



Quellenangabe: pixelio.de

## Neue Führung in der IT



- Service-Qualität und Kundenzufriedenheit sind ein gemeinsames Ziel von Management und Mitarbeitern
- Prozesse werden gemeinsam gestaltet
- Freiräume werden bewusst gelassen
- Mitarbeiter „verändern“ ⇒ *Mitarbeiter anspornen*
- Befehl und Gehorsam ⇒ *Ein eigenes Umfeld schaffen*
- Große Ideen ⇒ *Mutige Taten*
- „Mir ist’s egal („... was Sie denken“) ⇒ *„Ich weiß es nicht“ („... jedenfalls habe ich nicht alle Antworten“)*
- Schall und Rauch (Führung als Mysterium) ⇒ *Hand und Fuß (Führung als praktische Kunst)*
- Mitarbeiter „korrigieren“ ⇒ *Mitarbeiter „konnektieren“*



**Vielen Dank.**

direkt gruppe

Hauptsitz  
Griegstraße 75, 22763 Hamburg

Geschäftsstelle West  
Am Holzmarkt 2, 50676 Köln

Fon: +49 40 88155-0  
Fax: +49 40 88155-5200  
Mail: [Folkert.Jung@direkt-gruppe.de](mailto:Folkert.Jung@direkt-gruppe.de)  
Mail: [Wolfram.Herz@direkt-gruppe.de](mailto:Wolfram.Herz@direkt-gruppe.de)

## 9.1 *ITIL bei der Optimierung von Serviceprozessen*

# ITIL richtig gemacht: Über die richtige Einführungsstrategie und die Arbeit mit Menschen

Markus Bonk

Managing Consultant IT-ServiceManagement – Öffentliche Verwaltung  
HiSolutions AG, Bouchéstr. 12  
12435 Berlin  
bonk@hisolutions.com

**Abstract:** Um ITIL erfolgreich umzusetzen ist mehr als nur „Schrankware“ zu produzieren und ein neues Tool einzuführen. Vielmehr bedarf es unter anderem einer vernünftigen Einführungsstrategie und umfassender Erfahrungen in der IT-Organisationsentwicklung. Der Beitrag beschreibt die beiden Themenfelder und verdeutlicht die Herausforderungen und Problemstellungen. Es werden konkrete Erfolgsfaktoren benannt und Lösungen zur erfolgreichen ITIL-Einführung beschrieben.

## 1 Motivation

ITIL ist schon seit vielen Jahren aus in IT-Organisationen kaum noch wegzudenken. Nach einer motivierenden Start in ein „ITIL-Projekt“ mit hohen Erwartungen und dem Ziel durch die Einführung von ITIL möglichst alle Probleme einer IT-Organisation zu lösen folgt allzu häufig die Ernüchterung: Es ist viel Papier produziert, ein Tool ist eingeführt aber wirklich besser geworden ist nichts. Woher kommen solche Erfahrungen? Was ist die Ursache für das Scheitern von Projekten im IT-ServiceManagement? Und vor allem: Wie kann man es besser machen? Der Vortrag gibt auf diese Fragestellungen praxisnahe Antworten und schlägt konkrete Lösungen vor.

## 2 Ansatz

Die Praxiserfahrung zeigt, dass der Erfolg von Projekten im IT-ServiceManagement eher selten im Design von Prozessen zu suchen ist. Die Entwicklung von Abläufen im IT-ServiceManagement ist in der Regel schnell beherrschbar. Auch die Einführung unterstützender Tools und Software ist eine Herausforderung, die vor allem IT-Organisationen regelmäßig meistern. Probleme existieren jedoch aus unserer Beobachtung an zwei Stellen:

- 1) Einführungsstrategie: Es fehlt eine vernünftige Einführungsstrategie, die sich mit folgenden Fragen befasst:



- 2) Mitnahme von Mitarbeiterinnen und Mitarbeitern:
- ITIL führt zu erheblichen **Änderungen an verinnerlichteten Arbeitsabläufen**: Prozesse die bisher nicht oder über Linienfunktionen gesteuert wurden, sollen nun in definierten **Managementprozessen** gesteuert werden **Ein hoher technischer Fokus** während des Prozessdesigns vernachlässigt organisatorisches Veränderungsmanagement.
  - Wenn die Organisationsmitglieder die anstehenden Veränderungsmaßnahmen als **Bedrohung** empfinden, sich **nicht ausreichend eingebunden** oder informiert fühlen und so die Auswirkungen der Änderungen **nicht vollständig überblicken** können, besteht die Gefahr, dass die neue Ausrichtung der IT-Organisation an den ITIL-Prozessen nicht den gewünschten Erfolg hat und sogar scheitert.
  - Ansatz (Konkrete Kriterien für eine gute Lösung, Lösungsansatz, gewähltes Vorgehen, Konzeption, Herangehensweise)

### 3 Bewertung

In dem Beitrag sollen für die beiden genannten Problembereiche konkrete Handlungsempfehlungen und Lösungsmöglichkeiten gegeben werden:

- Wie gelange ich mit möglichst überschaubarem Aufwand zu einer validen Einführungsstrategie?
- Wie kann ich den Faktor Mensch in einem gesteuertem Veränderungsmanagement berücksichtigen.





## ITIL richtig gemacht

Über die richtige  
Einführungsstrategie und die Arbeit  
mit Menschen

Markus Bonk  
Managing Consultant  
ITSM – Public Sector  
HiSolutions AG

VBA™ HiSolutions AG



### Inhalt

#### 1. Über HiSolutions

#### 2. Erfolgsfaktoren in ITIL Projekten

- » Einordnung & Grunderkenntnis
- » Bedarf und Nutzen
- » Ziele, Orientierung und Führung
- » Zeit und Abschluss
- » Organisationsreife
- » Theorie und Praxis

#### 3. Start in die Veränderung

- » Veränderungsbedarfsanalyse™
- » Varianten



1

## HiSolutions

- ⇒ Überblick
- ⇒ Kernkompetenzen
- ⇒ Beratungsfelder
- ⇒ Kunden



## HiSolutions AG

**Business  
Felder**

Beratung

**Position +  
Kunden**

- ◆ **IT Governance . Risk . Compliance**
- ◆ **IT-Service & Portfolio Management**
- ◆ **Information Security Management**

- Führenden unabhängiger Beratungsspezialist im IT-Service Management und Information Security Management
- langjährige Umsetzungskompetenz in IT-Departments vieler Branchen und aller Größenordnungen
- Aktive Mitwirkung in der Entwicklung und Etablierung wichtiger IT-Standards

**Projekte**

über 250 jährlich in D/A/CH

**Mitarbeiter**

55

**Gründung**

1994

**Sitz**

Berlin

**Awards**

Fast50-Award Deutschland

Fast500-Award EMEA

mehrere Innovationspreise



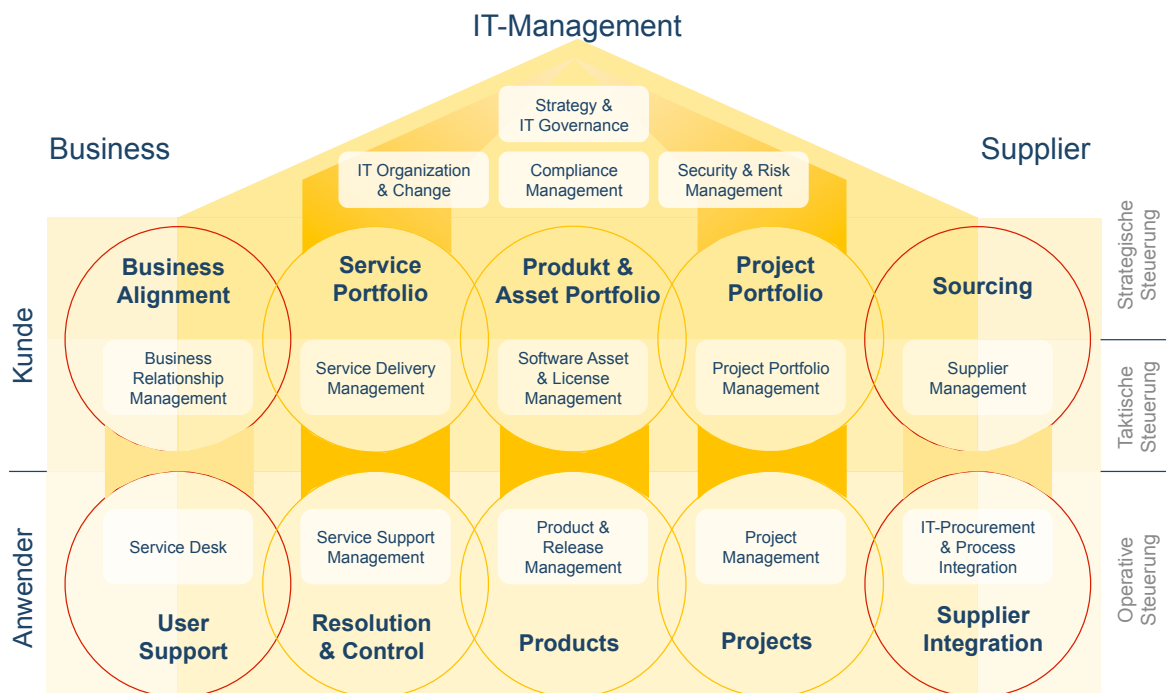
## Beratungsspezialist



Wir machen Ihre IT-Organisation „Fit für die Zukunft“

© 2010, HiSolutions AG | ITIL richtig gemacht – SEE Conf 2010 5

## IT-Management-Organisation



© 2010, HiSolutions AG | ITIL richtig gemacht – SEE Conf 2010 6

# 2 Erfolgsfaktoren in ITIL-Projekten

- ⇒ Einordnung & Grunderkenntnis
- ⇒ Bedarf und Nutzen
- ⇒ Ziele, Orientierung und Führung
- ⇒ Zeit und Abschluss
- ⇒ Organisationsreife
- ⇒ Theorie und Praxis

## Erfolgsfaktor 1: Einordnung & Grunderkenntnis



© Peter Kirchhoff / PIXELIO

**Gewohnheiten:** Liebgewonnene Gewohnheiten dauerhaft ändern



© ham/256 / PIXELIO

**Organisationsprojekt:** ITIL darf nicht zum Tool-Customizing-Projekt verkommen

## Erfolgsfaktor 2: Bedarf und Nutzen



© Stephan Bratek/geralt / PIXELIO

**Problembewußtsein:** Warum soll eigentlich etwas geändert werden?

**Nutzen:** Ist ITIL für mich das richtige und alleinige Heilmittel?



© Grace Winter / PIXELIO

## Erfolgsfaktor 3: Ziele, Orientierung und Führung

**Planung:** Zeigen, was wann verändert werden soll



© TRGeizler / PIXELIO



© Alfred Patuschka / PIXELIO

**Führung:** ITIL bedeutet IT-Service-MANAGEMENT



## Erfolgsfaktor 4: Zeit und Abschluss

**Zeit:** Veränderungen brauchen Ihre Zeit

© Essenia Deva / PIXELIO



© Maren Baßler / PIXELIO

**Abschluss:** Erst eine Veränderung abschließen und dann die nächste beginnen

## Erfolgsfaktor 5: Organisationsreife

**Realistisch bleiben:** Richtige Einschätzung der Fähigkeiten

© Rainer Sturm / PIXELIO



© halmackentreuter / PIXELIO

**Start kennen:** Ausgangspunkt ehrlich einschätzen

## Erfolgsfaktor 6: Theorie und Praxis

**Praxis beachten:** Ihre Kollegen sind keine Out-Of-The-Box-Clone



© Stephanie Hofschlaeger / PIXELIO

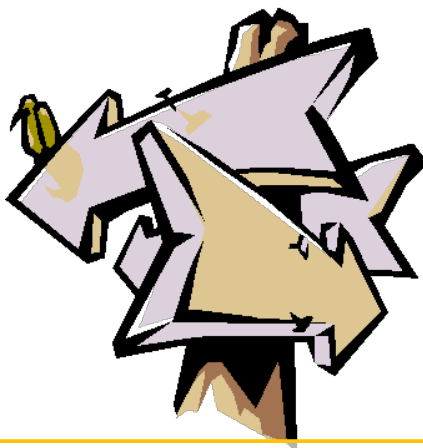


© Claudia Hautum / PIXELIO

**Ergebnisse:** Machen statt konzipieren, schreiben, reden...!

© 2010, HiSolutions AG | ITIL richtig gemacht – SEE Conf 2010 13

## Entscheidungsbedarf



**Was** verändern?

Wesentlicher Prozessfokus

**Warum** verändern?

Konkrete Erwartung

**Wann** verändern?

Passender Stufenplan

**Wie** verändern?

Strategie und Maßnahmen

**Veränderung**

© 2010, HiSolutions AG | ITIL richtig gemacht – SEE Conf 2010 14

## Welche Fragen bewegen?

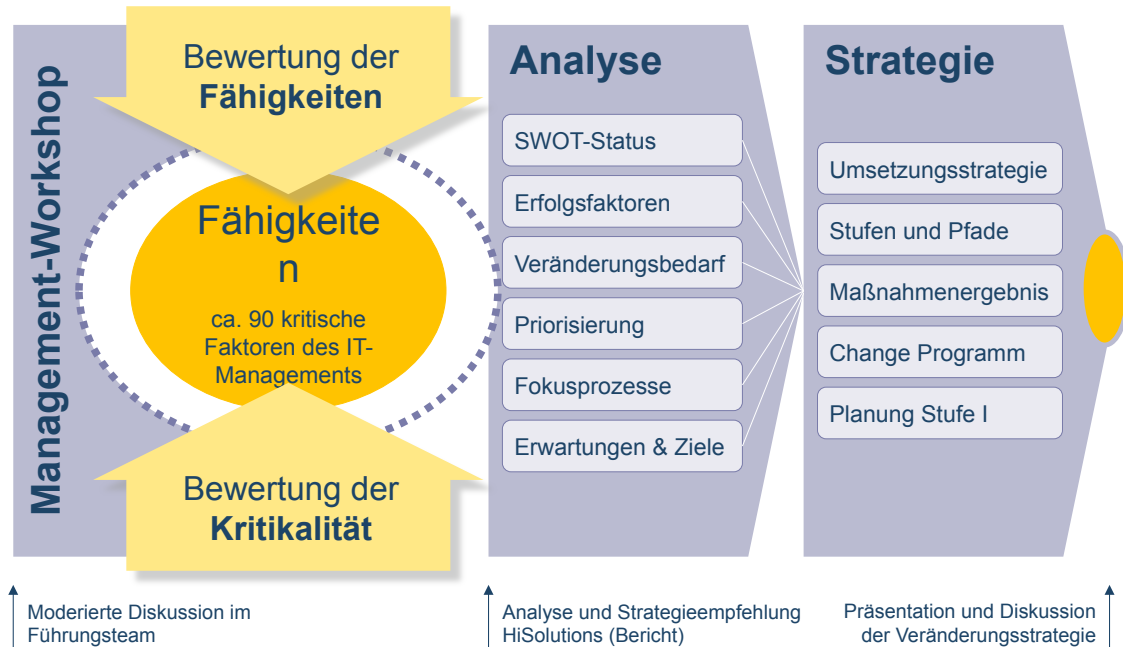


# 3 Start in die Veränderung

- ⇒ Veränderungsbedarfsanalyse™
- ⇒ Varianten

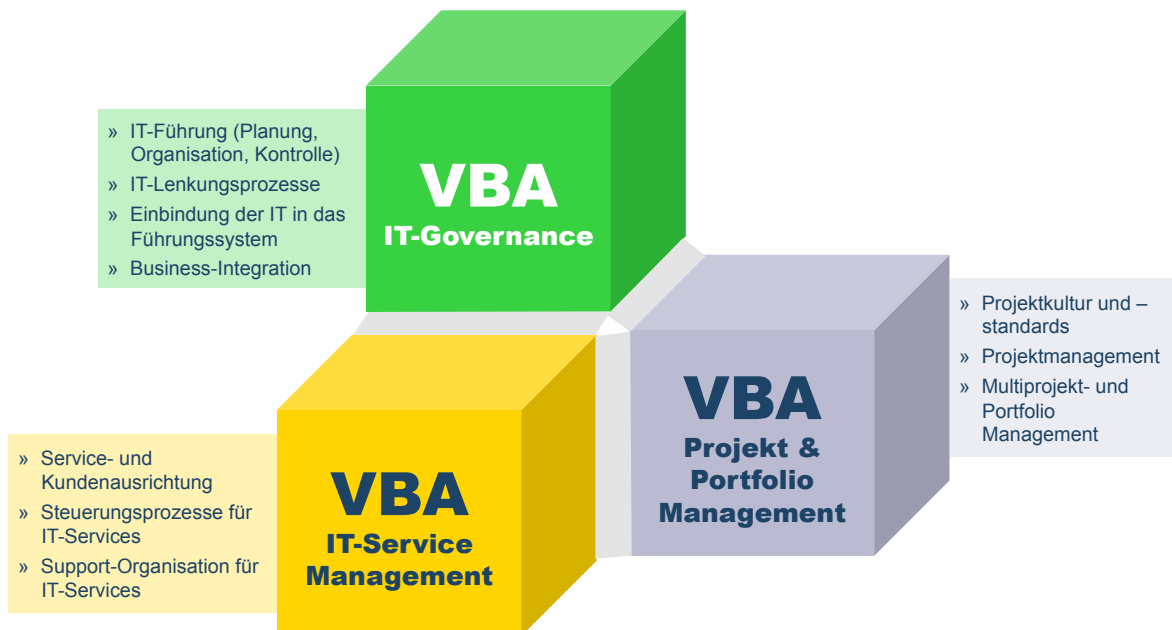
## Veränderungsbedarfsanalyse™

Ausgangspunkt ist das zu lösende Problem, nicht der konkrete Prozess



© 2010, HiSolutions AG | ITIL richtig gemacht – SEE Conf 2010 17

## Varianten



© 2010, HiSolutions AG | ITIL richtig gemacht – SEE Conf 2010 18





## Kontakt

Anschrift

**HiSolutions AG**  
Bouchéstraße 12  
D-12435 Berlin

Fon: +49 30 533289-0  
Fax: +49 30 533289-900  
[www.hisolutions.com](http://www.hisolutions.com)

Kontakt

**Markus Bonk**  
Managing Consultant ITSM  
Public Sector  
[bonk@hisolutions.com](mailto:bonk@hisolutions.com)

## **Einführung von CMMI unter agilen Gesichtspunkten – Wie agile Methoden die Umsetzung von CMMI in kleinen Organisationen erleichtern**

Matthias Stawinski<sup>1</sup>, Dr. Winfried Russwurm<sup>2</sup>

<sup>1</sup>Competence Center für Wirtschaftsinformatik, Hochschule München  
Lothstraße 64, 80335 München  
m.stawinski@hm.edu

<sup>2</sup>Siemens AG, Enterprise Processes, Corporate Technology  
Otto-Hahn-Ring 6, 81739 München  
russwurm@siemens.com

**Abstract:** Der Einsatz von CMMI-DEV bietet eine Menge praktischer Vorzüge für kleine Organisationen. Doch wie kann CMMI in kleinen Projekten effizient, also ohne erheblichen Mehraufwand, eingeführt werden?

Dieser Beitrag zeigt Beziehungen zwischen kleinen und mittleren Organisationen und agilen Vorgehensweisen auf und beschreibt, wie agile Methoden die Einführung von CMMI effizient unterstützen können. Darüber hinaus wird anhand eines Fallbeispiels eine exemplarische Umsetzung des CMMI und insbesondere der Prozessgebiete Projektplanung (PP) und Projektverfolgung und -steuerung (PMC) sowie deren Anknüpfungspunkte mit anderen Prozessgebieten beschrieben.

### **1 Motivation und Definitionen**

Das Prozessmodell CMMI wird häufig für die Anwendung in kleinen Organisationen – respektive kleinen Projekten – als nicht zweckmäßig und handhabbar wahrgenommen. Meist werden hierfür die mit CMMI verbundenen Aufwände und dessen Komplexität als wesentliche Gründe genannt. Die Praktiken des CMMI sind jedoch für Organisationen jeder Größenordnung sinnvoll und sehr hilfreich. Letztendlich unterstützt die Anwendung von CMMI Projekte auf unterschiedliche Art und Weise, insbesondere hinsichtlich der Erfüllung von Budget-, Termin-, Funktionalitäts- und Qualitätszielen. Insofern sollte CMMI auch in kleinen Projekten Anwendung finden.

Doch was ist ein kleines Projekt? In der hier betrachteten Domäne sei ein kleines Projekt durch annähernd 10 eingebundene Entwickler und Tester sowie einer geschätzten Projektdauer von sechs Monaten oder weniger definiert.

### 2 Agile, CMMI und KMUs – wie passt das zusammen?

Im Folgenden soll diskutiert werden, wie CMMI möglichst aufwandsneutral in kleinen Projekten eingeführt werden kann. Dabei soll ein generischer Ansatz verfolgt werden, welcher die Umsetzung der CMMI-Praktiken in kleinen Organisationen vereinfacht, in dem die agilen Prinzipien *Angemessenheit* und *Ergebnisorientierung* berücksichtigt werden.

#### 2.1 Typische Charakteristika von kleinen und mittelständischen Unternehmen

Für üblich weisen kleine und mittelständische Unternehmen (KMU) bestimmte charakteristische Eigenschaften auf. Zumeist lassen sie sich insbesondere durch Merkmale wie *geringere Finanzmittel, flache Hierarchien, lokale Nähe der Mitarbeiter, kundengetriebene Entwicklung* und *enge Zusammenarbeit mit dem Kunden* sowie *Innovationsführerschaft* definieren. Darüber hinaus hängt die Zukunft von KMU häufig davon ab, dass *Projektziele zeit- und kostengerecht erfüllt werden*. Insofern besteht meist auch die *Notwendigkeit, Arbeit und Projektfortschritt in angemessenen Maße zu dokumentieren*.

#### 2.2 Das Agile Manifest – Grundlage agiler Vorgehen

Das Agile Manifest definiert die grundlegenden Wesenszüge einer agilen Methode. Es sagt aus, dass *Menschen und klare Kommunikation* wichtiger als festgeschriebene Standardabläufe und -Tools sind, dass *einwandfreie Software* wichtiger als übertriebene Dokumentation ist, dass *enge Zusammenarbeit mit dem Kunden* wichtiger als reine Vertragsverhandlungen ist und dass *Änderungen am Plan angenommen werden sollen*.

#### 2.3 Gemeinsamkeiten von Agile und KMU

Bei genauem Vergleich lässt sich feststellen, dass sich agile Grundzüge sehr gut mit typischen Eigenschaften von KMU vereinigen lassen. So lassen sich bspw. flache Hierarchien und lokale Nähe der Mitarbeiter gut mit den agilen Grundsätzen Eingehen auf Menschen und starke Zusammenarbeit verbinden. Ebenso lassen sich kundenorientierte Entwicklung und kooperative Zusammenarbeit mit Kunden auf den Grundsatz Ausgeprägte Zusammenarbeit mit Auftraggebern abbilden.

#### 2.4 Agile & CMMI – ein Fallbeispiel

Nachdem festgestellt wurde, dass Agile und KMU eine gemeinsame Charakteristik besitzen, sollten agile Methoden als ein möglicher Ansatz, um CMMI in kleinen Organisationen einzuführen, diskutiert werden. Wie agile Vorgehen die Anforderungen des CMMI erfüllen und durch ihre Einführung zugleich die Verbesserung des Software-Engineering unterstützen, soll im folgenden Fallbeispiel erläutert werden: Es sei ein (Teil-)Projekt zur Entwicklung der grundlegenden Workflow-Services für ein Dokumen-

tenmanagementsystem angenommen. Die Projektdauer beläuft sich auf rund sechs Monate; neun Entwickler und ein Tester sind dem Projekt zugeteilt.

Anfänglich werden die Projektplanungsparameter und der Projektplan grob geschätzt. Während des iterativ-inkrementellen Projektfortschritts werden die Schätzungen immer genauer. Dies wird durch die grobe Schätzung der Komplexität (bspw. mit Planning Poker oder Delphi Methode) und Abgrenzung der Anforderungen (bspw. mit der MoS-CoW-Methode – Must, Should, Could, Won't have this time but Would like in the future) erreicht. Im Folgenden werden die Sprints und entsprechende Artefakte via Time und/oder Cost Boxing Methoden geschätzt und geplant. Dies wird durch die Spezifischen Ziele 1 und 2 des CMMI-Prozessgebiets Projektplanung adressiert. Sogenannte Backlog Meetings zu jedem Sprint liefern Rückmeldungen über Ergebnisse und Probleme und können als eine Funktion von „Fortschritts- und Meilensteinreviews durchführen“ in den Praktiken 1.6 und 1.7 des Prozessgebiets Projektverfolgung und -steuerung betrachtet werden. Dies führt zu Entscheidungen/Interventionen sowie möglicherweise auch neuen Anforderungen und einem entsprechend revidierten Plan, wie es in den Praktiken 2.2 und 2.3 der Projektverfolgung und -steuerung sowie 1.3 des Prozessgebiets Anforderungsmanagement erwartet wird.

### 3 Fazit

In Anbetracht, dass agile Methoden besonders Eigenheiten von KMU adressieren und, wie im Fallbeispiel dargestellt, geeignet sind um den Anforderungen des CMMI zu entsprechen, stellt die Anwendung von agilen Methoden einen guten und pragmatischen Ansatz dar, um CMMI in kleinen Organisationen einzuführen.

### Literaturverzeichnis

- [St07] Stawinski, M.: Die effiziente Anwendbarkeit von CMMI in kleinen Projekten. Untersucht an den Prozessgebieten Projektplanung und Projektverfolgung und -steuerung. Hochschule München/Siemens AG, München, 2008.
- [Ru09] Russwurm, W.: Agile und CMMI – “die Quadratur des Kreises”? OBJEKTSpektrum, Online-Themenspecial Agility 2009. <http://www.objektspektrum.de>.



## Einführung von CMMI unter agilen Gesichtspunkten

Wie agile Methoden die Umsetzung von CMMI in kleinen Organisationen erleichtern

Matthias Stawinski, Hochschule München  
Dr. Winfried Russwurm, Siemens AG

**SIEMENS**



In Kooperation mit:



## Agenda



1. Motivation
2. Definition kleiner Projekte
3. Gemeinsamkeiten von Agile und KMU
4. Abgeleitete Richtlinien
5. Ansatz um CMMI in KMU einzuführen
6. Fallbeispiel – Agile und CMMI
7. Zusammenfassung und Fazit



## Motivation – CMMI und kleinen Organisationen

### Ausgangssituation und Mythen:

**CMMI wird häufig als nicht passend für kleine Organisationen erachtet:**

- Aufwand für Einführung
- Komplexität und Umfang
- Nur für große Organisationen entwickelt



### Herausforderungen und Vorteile:

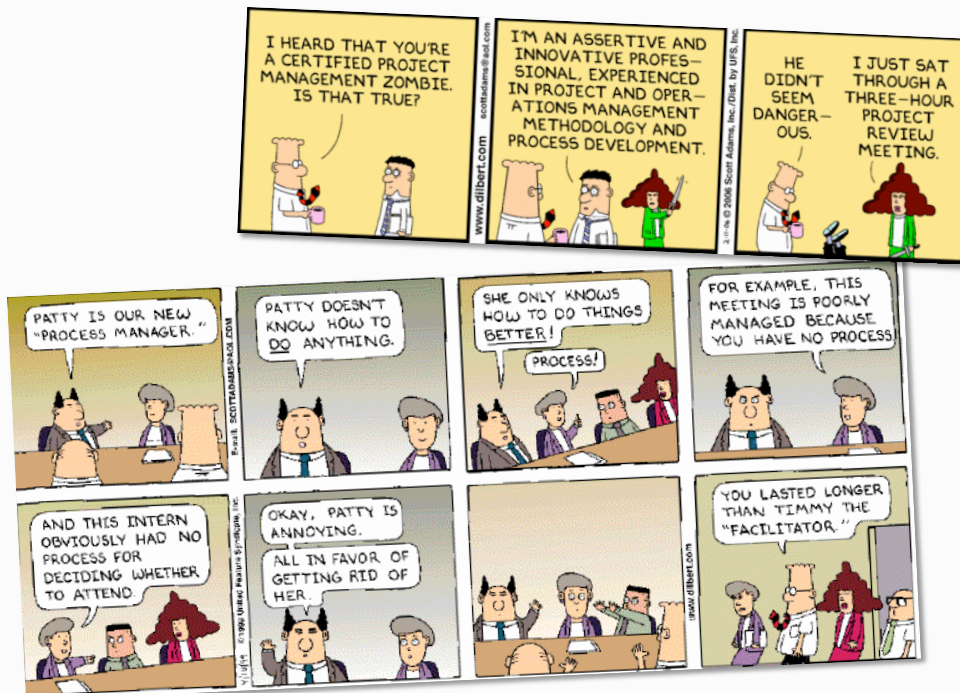
#### Herausforderungen:

- Geringe Prozessreife kann Nachteile am Markt bedeuten
- Vorteile der Nutzung reifer (CMMI) Praktiken nicht vorhanden
- Unzureichende Erfüllung der Projektziele hinsichtlich Kosten, Zeit und Qualität

#### Vorteile durch CMMI:

- Bessere Zielerreichung im Projekt
- Verbesserte Marktsituation gegenüber Wettbewerbern

## Prozessverbesserung – häufige Irrtümer



## Definition kleiner Projekte

Projektgröße lässt sich definieren über:

- Anzahl der Teammitglieder
- Voraussichtlicher Dauer
- Finanziellem Rahmen (Budget)



Größe	Personenanzahl	Laufzeit
Klein	bis 10 Personen	kurzfristig: bis zu 1 Monat langfristig: bis zu 6 Monaten
Mittel	bis 20 Personen	6 bis 12 Monate
Groß	> 20 Personen	> 12 Monate

## Typische Charakteristika von KMU



Merkmale kleiner und mittelständischer Unternehmen:

- Meist geringere Finanzmittel
- Meist flache Hierarchien
- Meist lokale Nähe der Mitarbeiter
- Meist kundengetriebene Entwicklung
- Häufig enge Zusammenarbeit mit dem Kunden
- Häufig Innovationsführer

Zukunft meist abhängig von:

- Zeit- und kostengerechte Erfüllung der Projektziele
- Angemessene Dokumentation von Arbeit und Projektfortschritt



## Agile Manifest – Grundlage agiler Methoden



### Wichtig:

- Arbeitsabläufe und Tools
- Dokumentation
- Vertragsverhandlungen
- Befolgen von Plänen



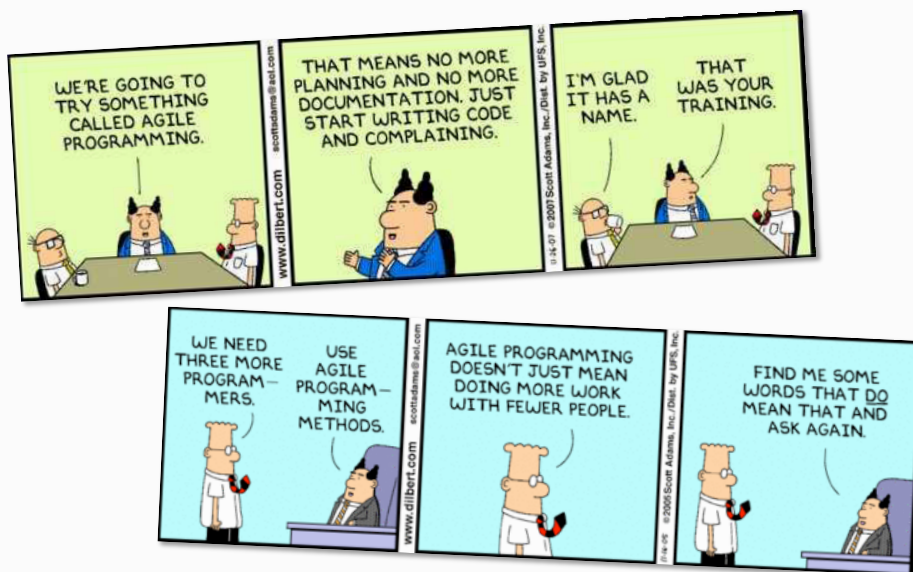
### Noch wichtiger:

- Menschen und ausgeprägte Kommunikation und Interaktion
- Funktionstüchtige Software
- Enge Zusammenarbeit mit Kunden
- Eingehen auf Änderungen

### Bedeutet:

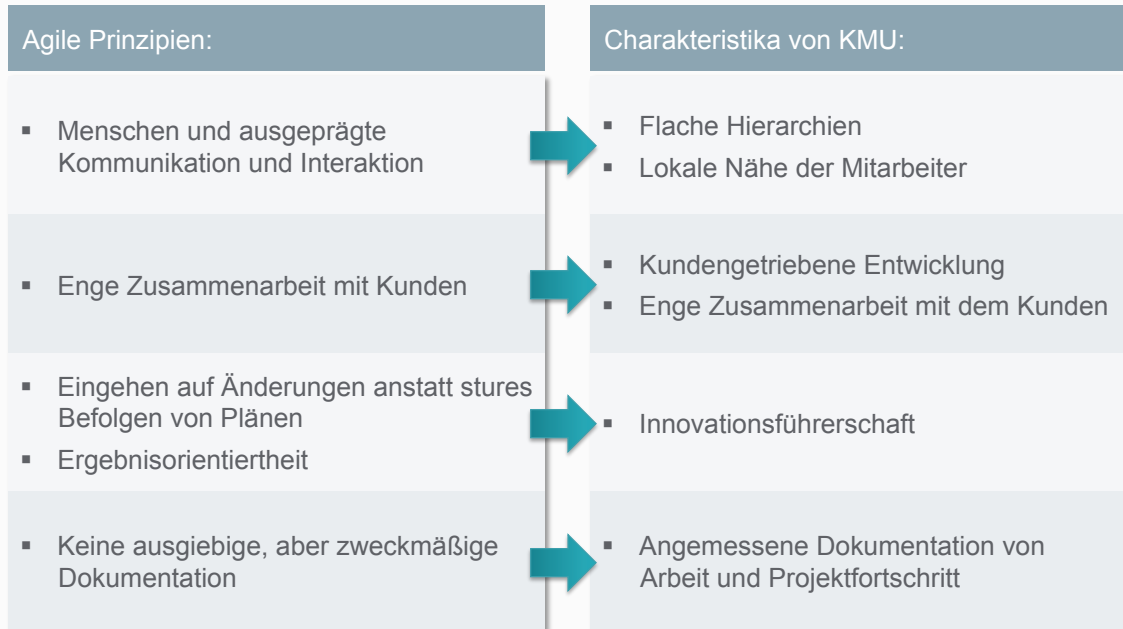
Prozesse, Tools, Dokumentation, Verträge und Pläne sind für die Erreichung von Projektzielen notwendig, sollten **aber** stets **angemessen** eingesetzt werden.

## Agile Softwareentwicklung – häufige Irrtümer





## Gemeinsamkeiten von Agile und KMU



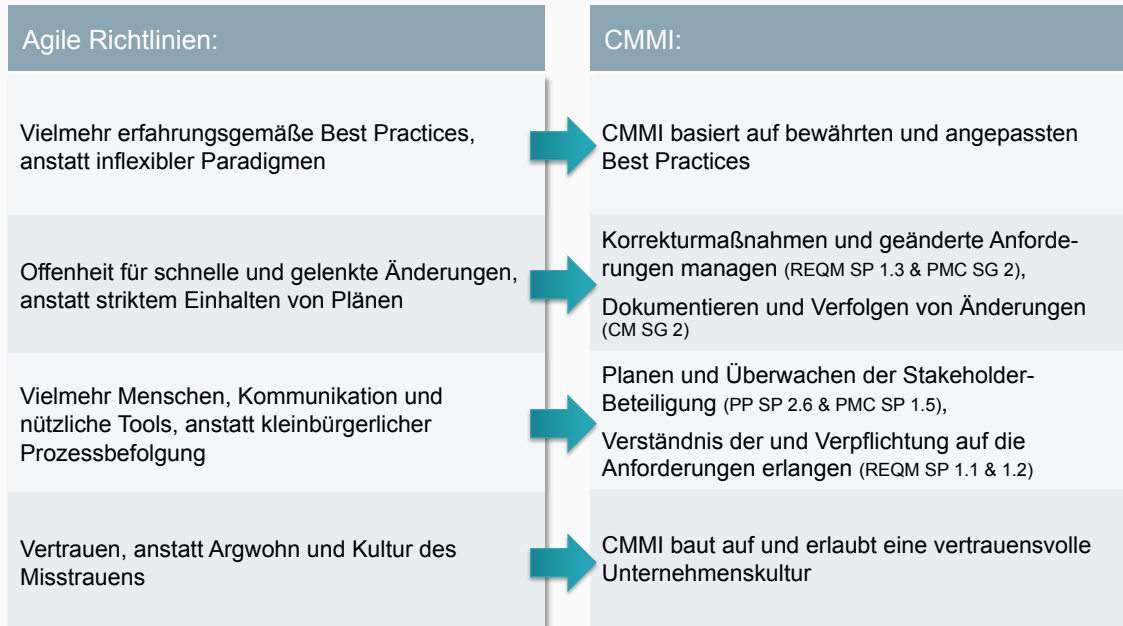
## Abgeleitete Richtlinien – Kombination von Agile und KMU



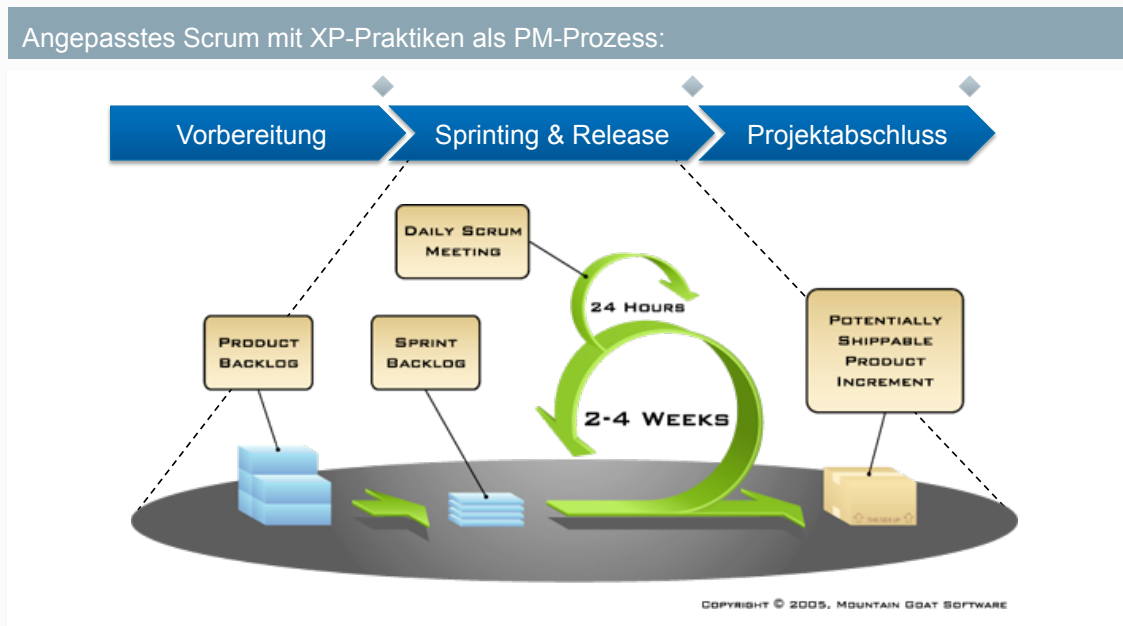
### Mögliche Basis für Prozesseinführung in KMU:

- Vielmehr **ergebnisorientiert**, anstatt unangemessen umfangreiche/detaillierte Prozesse
- Vielmehr **erfahrungsgemäße Best Practices**, anstatt striktes Verfolgen unflexibler, dem Geschäft ungenügender Paradigmen
- Vielmehr **offen für schnelle und gelenkte Änderungen**, anstatt rigides Einhalten eventuell veralteter Pläne / Vorgaben
- Vielmehr **Menschen, Kommunikation und nützliche Tools**, anstatt kleinlicher Prozessbefolgung
- Vielmehr **Zuversicht**, anstatt einer Kultur des Misstrauens

## Richtlinien als Ansatz zur Einführung von CMMI



## Fallbeispiel – Agiles Projektmanagement und CMMI



## Fallbeispiel – Agile Projektplanung und -steuerung



### Prozessgebiet „Projektplanung“ (PP):

- Anfangs grobe Schätzung der Projektplanungsparameter und des Projektplans [PP SG 1]
- Während des Projektfortschritts werden Schätzungen immer genauer (iterativ-inkrementell) [PP SG 1]
- Schätzung von Komplexität und Umfang (Planning Poker, Delphi Methode) [PP SG 1]
- Abgrenzung der Anforderungen (MoSCoW-Methode, Eisenhower-Prinzip, Economic Value) [PP SG 1, SG 2]
- Time- und / oder Costboxing zur Schätzung und Planung der Sprints [PP SG 1, SG 2]
- Verpflichtung der Stakeholder werden eingeholt [PP SG 3]

## Fallbeispiel – Agile Projektplanung und -steuerung



Planning Poker®

MoSCoW - Methode



## Fallbeispiel – Agile Projektplanung und -steuerung

**Prozessgebiet „Projektverfolgung und -steuerung“ (PMC):**

- Backlog Meetings zu jedem Sprint liefern Feedback zu Ergebnissen und Problemen [PMC SG 1]
- Führt zu Diskussionen/Interventionen, evtl. neuen Anforderungen und Anpassungen des Projektplans [PMC SG 1, SG 2, REQM SG 1]
- Projektrisiken können im Daily Scrum Meeting frühzeitig erkannt werden (Any Impediments?) [PMC SG 1, SG 2, RSKM SG 2]

## Zusammenfassung und Fazit

**Zusammenfassung:**

- Agile Methoden sind hinsichtlich der Eigenheiten von KMU und deren geschäftlichen Anforderungen geeignet für kleine Organisationen
- Agile Methoden, respektive der „agilen Richtlinien“, erfüllen die Anforderungen von CMMI weitestgehend

**Folgerung:**

- Anwendung agiler Methoden stellt guten und pragmatischen Ansatz dar, um CMMI in kleinen Organisationen einzuführen

- Ergebnisse konnten bereits in agilen Projekten erfolgreich eingesetzt werden, Benefits müssen noch evaluiert werden

---

## FRAGEN & DISKUSSION



## Kontaktinformationen

### Matthias Stawinski



Matthias Stawinski

Hochschule München  
Competence Center für Wirtschaftsinformatik  
Fakultät für Informatik und Mathematik  
Lothstraße 64  
80335 München  
[m.stawinski@hm.edu](mailto:m.stawinski@hm.edu)  
Mobil: 0178/ 290 29 20

### Dr. Winfried Russwurm



Dr. Winfried Russwurm

Siemens AG  
Corporate Technology  
Enterprise Processes  
Otto-Hahn-Ring 6  
81379 München  
[russwurm@siemens.com](mailto:russwurm@siemens.com)  
Telefon: 089/ 636 42 627

## Literaturhinweis



**Die effiziente Anwendbarkeit von CMMI® in kleinen und mittleren Projekten. Untersucht an den Prozessgebieten Projektplanung und Projektverfolgung und -steuerung.**

Stawinski, Matthias. Hochschule München / Siemens AG 2008.

**Agile und CMMI – “die Quadratur des Kreises”?**

Winfried Russwurm. OBJEKTSpektrum, Online-Themenspecial Agility 2009.

**Small Steps, Giant Leap.**

Srivastava, Nidhi. Murthy, Sathya. SEPG Conference 2006.

**Agile Project Management with Scrum.**

Schwaber, Ken. Microsoft Press 2004.

**Agility kompakt. Tipps für erfolgreiche Systementwicklung.**

Hruschka, Peter. Rupp, Chris. Starke, Gernot. Spektrum 2004.

**Agiles Projektmanagement. Riskogesetzte Softwareentwicklung.**

Gernert, Christiane. Hanser 2003.

**The First CMMI-based Appraisal in an Agile Environment at Siemens AG. Results and Experiences.**

Canditt, Sabine. Russwurm, Winfried. SEPG Conference 2008.



Vielen Dank für die Aufmerksamkeit!

**SIEMENS**



In Kooperation mit:



© Capability Maturity Model, Carnegie Mellon, CMM, and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

sm CMM Integration; IDEAL; Personal Software Process; PSP; SCAMPI; SCAMPI Lead Assessor/ Appraiser; SEPG; Team Software Process; and TSP are service marks of Carnegie Mellon University.

### 9.3 Einführung von CMMI unter agilen Gesichtspunkten

## 10. V-Modell XT und Projektmanagement

### Sessionüberblick

---

10.1. Projektmanagement im V-Modell XT . . . . .	555
10.2. Tailoring PRINCE2:2009 . . . . .	569
10.3. Steuerung von IT-Projekten durch ein Projekt Management Office . . . . .	585

---





## Projektmanagement im V-Modell XT

Dr. Karl Kollischan, Jan Friedrich, Edward Fischer, Thomas Ternité, Reinhard Wagner

Arbeitsgruppe Projektmanagement WEIT e.V.  
karl.kollischan@kobaxx.com

**Abstract:** Durch die zunehmende Verzahnung von Organisationen und einer internationalen Ausrichtung von Projekten gewinnen einheitliche Prozesse im Projektmanagement immer mehr an Bedeutung. Auf internationaler Ebene wird dieser Entwicklung durch die Spezifikation der einheitlichen Projektmanagementnorm ISO 21500 Rechnung getragen. Die Arbeitsgruppe Projektmanagement wurde gegründet, um die praktische Anwendbarkeit der Projektmanagementprozesse im V-Modell XT zu verbessern. Neben Anforderungen seitens der V-Modell Anwender werden dabei internationale Standards und aktuelle Entwicklungen im Projektmanagement berücksichtigt und in das V-Modell integriert. Die aktuellen Arbeitsergebnisse werden hier vorgestellt.

### 1 Motivation

Im Zuge einer immer stärkeren Verzahnung von Organisationen und einer internationalen Ausrichtung von Projekten gewinnen eine gemeinsame Sprache und einheitliche Prozesse immer mehr an Bedeutung. Dies gilt ganz besonders für das Projektmanagement, wo die Arbeit von oft räumlich getrennten Teams sowie von Menschen mit vielfältigem kulturellen Hintergrund und unterschiedlichsten Ausbildungswegen – man denke allein an die verschiedenen Zertifizierungen für Projektmanagement - zu koordinieren und auf gemeinsame Ziele auszurichten ist.

Seitens der führenden Organisationen für Projektmanagement wurde die Wichtigkeit von einheitlichen internationalen Standards erkannt. Derzeit entsteht auf internationaler Ebene die prozessorientierte Norm ISO 21500 „A Guide for Project Management“. Zum einen fließen darin existierende nationale Standards (z.B. das Prozessmodell der DIN 69901-2) ein, zum anderen werden sich die nationalen Standards zukünftig an der neuen Norm orientieren.

Die Arbeitsgruppe Projektmanagement wurde Anfang 2009 im Rahmen des WEIT. e.V. mit dem Ziel gegründet, die Verständlichkeit und Anwendbarkeit des Projektmanagements im V-Modell XT für die Anwender in der Praxis zu verbessern. Zu ihren Aufgaben gehört es, Anregungen und Kritik der V-Modell Anwender kontinuierlich zu evaluieren sowie aktuelle Standards und Entwicklungen im Projektmanagement im Auge zu behalten. Zusammen mit der GPM Deutsche Gesellschaft für Projektmanagement e.V. koordiniert sie Anwenderanforderungen und Projektmanagementnormen und erarbeitet Vorschläge zur Verbesserung der Darstellung des Projektmanagements im V-Modell XT.

## **2 Vorstellung Arbeitsgruppe und der aktuellen Arbeitsergebnisse**

Im Hauptteil werden die Arbeitsgruppe, ihre Ziele und ihre aktuellen Arbeitsergebnisse vorgestellt.

### **2.1 Vorstellung der AG**

Mitglieder, Sinn und Zweck, Ziele

### **2.2 Auswertung Fragebogen VMEA**

Auf der V-Modell Anwenderkonferenz 2009 fand in Zusammenarbeit mit dem ANS-STAND e.V. eine Fragebogenaktion statt, mit welcher die Zufriedenheit und Kritikpunkte der V-Modell Anwender evaluiert wurden. Die Ergebnisse sollen hier vorgestellt und bewertet werden.

### **2.2 Konventionsabbildung DIN 69901 - VMXT**

Seit Anfang 2009 ist die neue DIN 69901 freigegeben, in der Standards für das Projektmanagement beschrieben werden. Die Konventionsabbildung beschreibt eine Abbildung von den Prozessen der DIN 69901 auf Elemente des V-Modell XT. Dafür ist diese Konventionsabbildung gegliedert in die einzelnen Prozesse, die in DIN 69901-2 beschrieben sind. Zu jedem Prozess ist für alle Outputs beschrieben, welche Elemente des V-Modell XT ihnen entsprechen.

### **2.3 Projektmanagement-View**

Mit dem Projektmanagement - View wird dem Anwender ein Nachschlagewerk in die Hand gegeben, in dem sämtliche PM-relevanten Inhalte des V-Modells zusammengefasst und einfach auffindbar sind.

### **2.4 Ausblick ISO 21500**

Die ISO 21500 (vgl. Kapitel „Motivation“) soll 2012 erscheinen. Sofern Inhalte im Vorfeld freigegeben werden, sollen sie hier mit ihrer Relevanz für das V-Modell XT dargestellt werden.

## **3 Diskussion und Bewertung**

Die abschließende Diskussion mit den Teilnehmern soll Erkenntnisse über den praktischen Wert der bisherigen Arbeitsergebnisse und insbesondere Informationen für die zukünftige Weichenstellung der Arbeitsgruppe Projektmanagement liefern.



# Projektmanagement im V-Modell XT

Ziele und aktuelle Ergebnisse der  
Arbeitsgruppe Projektmanagement

---

Verein zur Weiterentwicklung des V-Modell XT e.V.

## Inhalt

---



- Projektmanagement und V-Modell XT
- Vorstellung und Ziele der Arbeitsgruppe Projektmanagement im WEIT e.V.
- Auswertung des Fragebogens
- Abbildung auf Projektmanagementstandards
- Konventions- und Nomenklaturabbildung auf DIN 69901
- Ausblick



## Sie drei „Säulen“ des PM



Dr. Karl Kollischan, SEE 2010

04. Mai 2010

3

## Die Projektmanagement-Welt



Dr. Karl Kollischan, SEE 2010

04. Mai 2010

4

## AG Projektmanagement im WEIT

---



- Gegründet Anfang 2009 mit dem Ziel:

Verbesserung der Verständlichkeit und  
Anwendbarkeit des  
Projektmanagements im V-Modell XT  
für den Anwender in der Praxis

- Aktuelle Aktivitäten:
  - Abbildung auf prozessorientierte PM-Standards (DIN 69901, PMBoK, zukünftig ISO 21500)
  - PM-View: Übersichtliche Darstellung der PM-Aufgaben im V-Modell
  - Fragebogen zum Thema Projektmanagement im VMXT in Zusammenarbeit mit dem ANSSTAND e.V.

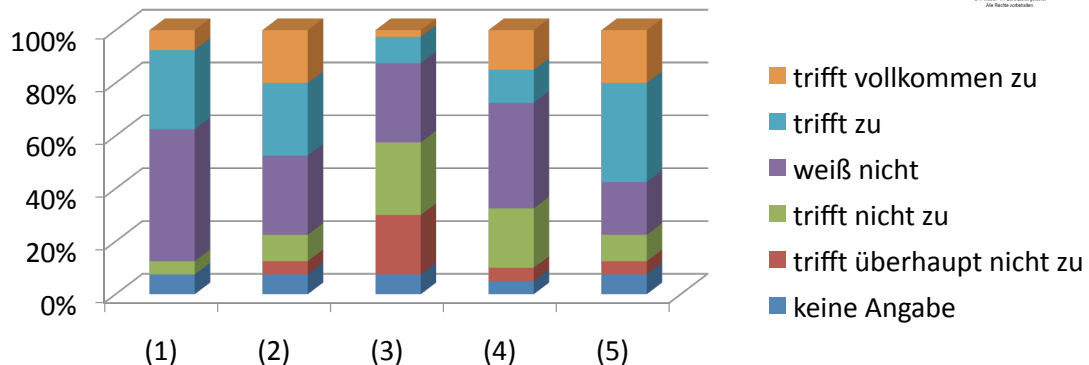
## Fragebogenaktion

---



- Durchgeführt bei der VMEA im November 2009 in München in Zusammenarbeit mit dem ANSSTAND e.V.
- Rücklauf: 40 Fragebögen
- Gefragt wurde nach:
  - Projektabdeckung durch das VMXT:
    - Der einzelnen Disziplinen der Systementwicklung
    - Des Projektmanagements im Besonderen
  - Anwendbarkeit des VMXT hinsichtlich:
    - Rollenmodell
    - Produktmodell
    - Tailoring
  - Standards im VMXT:
    - Die derzeit verwendete Begriffe und Methoden
    - Ausrichtung an etablierten Standards

## Auswertung Fragebogen Begriffe und Methoden



- (1) Die im V-Modell im Umfeld des Projektmanagements verwendeten Begriffe sind insgesamt verständlich und in sich konsistent
- (2) Die Begriffswelt des V-Modell XT sollte besser an allgemeinen Projektmanagementstandards ausgerichtet sein (z.B. DIN 69901)
- (3) Das V-Modell XT ist insgesamt zu „IT-lastig“
- (4) Im V-Modell XT werden moderne PM-Methodiken nicht adäquat berücksichtigt
- (5) Eine zusammenfassende Sicht auf die Projektmanagementaufgaben wäre sehr hilfreich

Dr. Karl Kollischan, SEE 2010

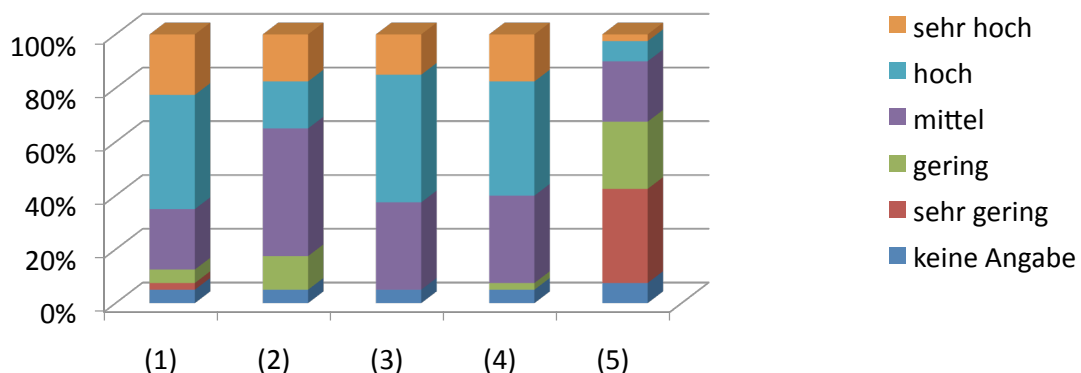
04. Mai 2010

7

## Auswertung Fragebogen Projektmanagementstandards



Eine Ausrichtung der Begriffe und Prozesse an etablierten Projektmanagementstandards...



- (1) ... fördert insgesamt die Verständlichkeit
- (2) ... verbessert die Projektmanagementprozesse
- (3) ... erleichtert den Einstieg für Projektleiter (die z.B. nach GPM zertifiziert sind)
- (4) ... erleichtert die Arbeit in internationalen Projekten bzw. Projektteams
- (5) ... bringt keine Vorteile

Dr. Karl Kollischan, SEE 2010

04. Mai 2010

8

## Fragebogen: Individuelle Anmerkungen (Auswahl)

---



- Praxisbezug fördern
- Vieles müsste konkreter sein, Bessere Unterstützung für kleine Projekte/Organisationen (Rollen)
- Anpassung an Praxis bei kleineren Projektteams
- deutlicher Ausbau bei Risikomanagement etc. sowie praktische Werkzeuge
- Darstellung der Zusammenhänge verbessern (z.B. zu welcher Disziplin gehört ein Produkt und welche VBs setzen es ein)
- Standardisierung von Begriffen notwendig (z.B. Begriff "Produkt führt immer zu Missverständnissen")
- Agile Methoden SAUBER in VM-XT integrieren!
- AG/AN Verhältnis bei unterschiedlichen Vorgehensweisen/-modellen z.B. AG arbeitet nach V-Modell und AN nach Scrum

---

Dr. Karl Kollischan, SEE 2010

04. Mai 2010

9

## Fazit Fragebogen

---



- Umfrage ist nicht repräsentativ, eher Stimmungsbild
- Aussagen sehr homogen, keine klare Erkenntnis „wo der Schuh drückt“
- Das V-Modell bietet im allgemeinen schon eine recht gute Unterstützung für Projektmanagementaufgaben
- Tendenzielle Zustimmung zu einer stärkeren Ausrichtung an PM-Standards
- Verbesserungspotenzial hinsichtlich Praxisbezug und Darstellung der Zusammenhänge
- Berücksichtigung agiler Methoden

---

Dr. Karl Kollischan, SEE 2010

04. Mai 2010

10



## Warum Abbildung auf PM-Standards?



Dr. Karl Kollischan, SEE 2010

04. Mai 2010

11

## Prozessorientierte PM-Standards



- DIN Deutsches Institut für Normung e.V.: Neue PM-Normen **DIN 69901-1 bis -5** mit Prozess- und Datenmodell
- ISO International Standardisation Organisation: Neue internationale PM-Norm **ISO 21500 Guide to Project Management** mit Prozessmodell (erscheint 2012)
- Project Management Institute (USA): **Project Management Body of Knowledge (PMBok)**
- OGC Office of Government Commerce (UK): Prozessorientierter PM-Standard **PRINCE 2**.
- ProSTEP iViP e.V. (Automobilindustrie) – Empfehlung: „**Collaborative Project Management**“, Referenz- und Datenmodell

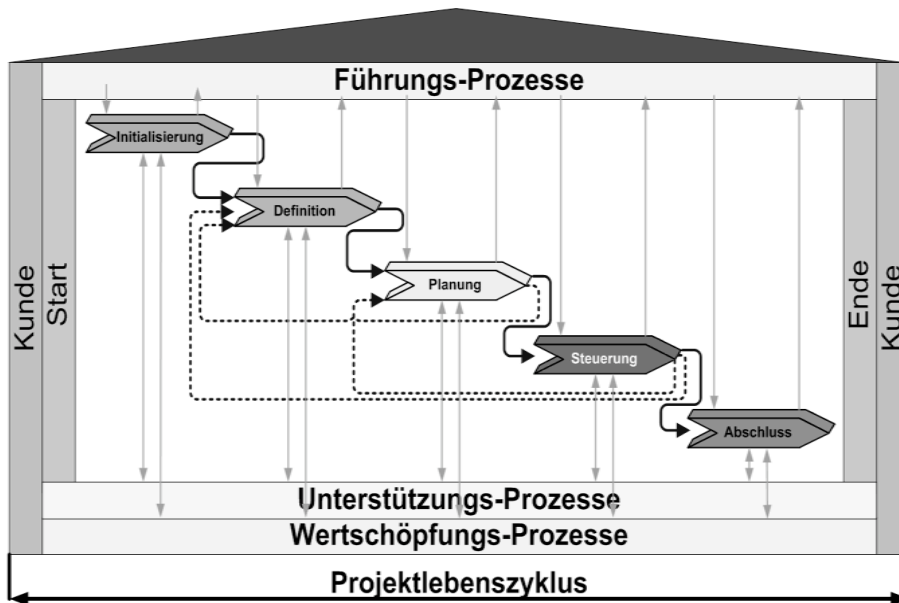
Dr. Karl Kollischan, SEE 2010

04. Mai 2010

12



## Prozessmodell der DIN 69901-2



Quelle: DIN 69901-2

Dr. Karl Kollischan, SEE 2010

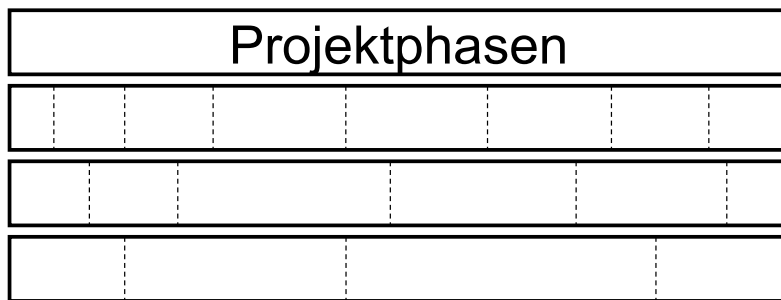
04. Mai 2010

13

## Projekt- und PM-Phasen



**5 Phasen**



**X Phasen**  
(Firmen- und/oder  
Branchenabhängig)



I = Initialisierung   D = Definition   P = Planung   S = Steuerung   A = Abschluss

Quelle: DIN 69901-2

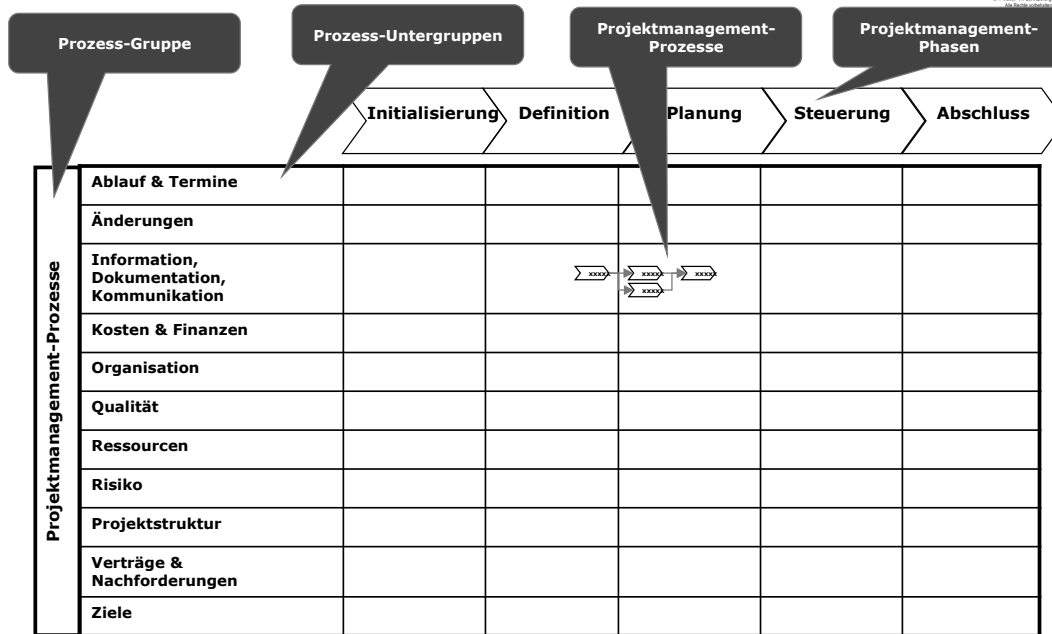
Dr. Karl Kollischan, SEE 2010

04. Mai 2010

14

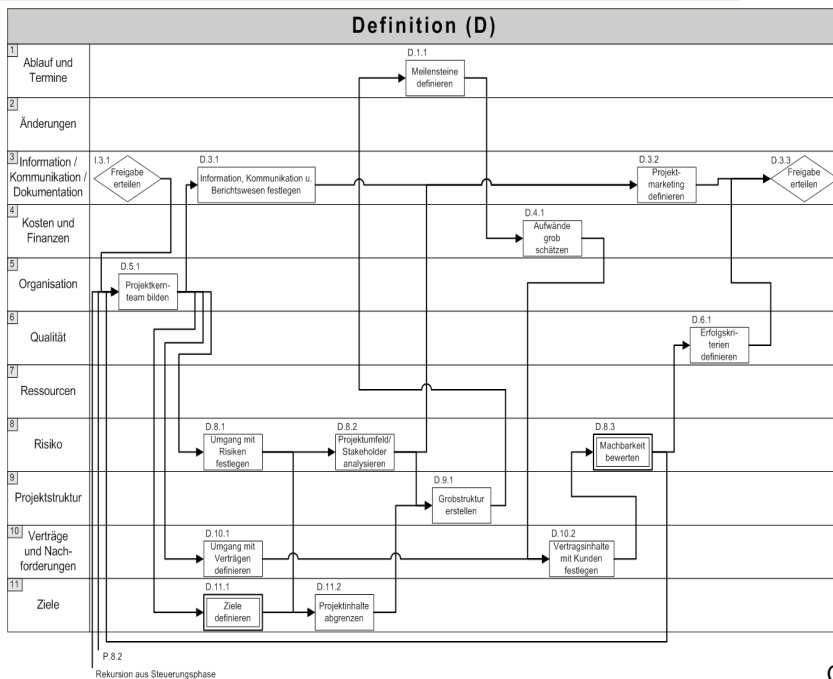


# Struktur des PM-Prozessmodells



Quelle: DIN 69901-2

# PM-Prozesse in der Definitionsphase



Quelle: DIN 69901-2



## DIN 69901 Prozessbeschreibung (Beispiel)

<b>Prozess Nr.:</b>	D 1.1		
<b>Prozessname:</b>	Meilensteine definieren		
<b>Vorgänger-Prozesse:</b>	D 9.1 Grobstruktur erstellen	<b>Nachfolge-Prozesse:</b>	D 4.1 Aufwände grob schätzen
<b>Zweck und Hintergrund:</b>	Nach der Abgrenzung der Projektinhalte und der Erstellung der Grobstruktur dient dieser Prozess nun dazu, die Zwischenereignisse/-ergebnisse in eine zeitliche Reihenfolge zu bringen. Dies ist u. a. Voraussetzung für die Abschätzung der Aufwände und die Bewertung der Machbarkeit sowie Grundlage für die Erstellung eines Terminplanes.		
<b>Prozessbeschreibung (Vorgehen):</b>	Zuerst werden die Meilensteine inhaltlich definiert (Meilenstein-Beschreibung), in eine zeitliche Reihenfolge gebracht und mit (groben oder vorläufigen) Terminen versehen (Meilensteinplan).		
<b>Input</b>	<b>PM-Methoden</b>	<b>Output</b>	
- Projektziele - Projektinhalte - Grobstruktur	- Meilenstein-Planung	- Beschreibung der Meilensteine - Meilensteinplan	

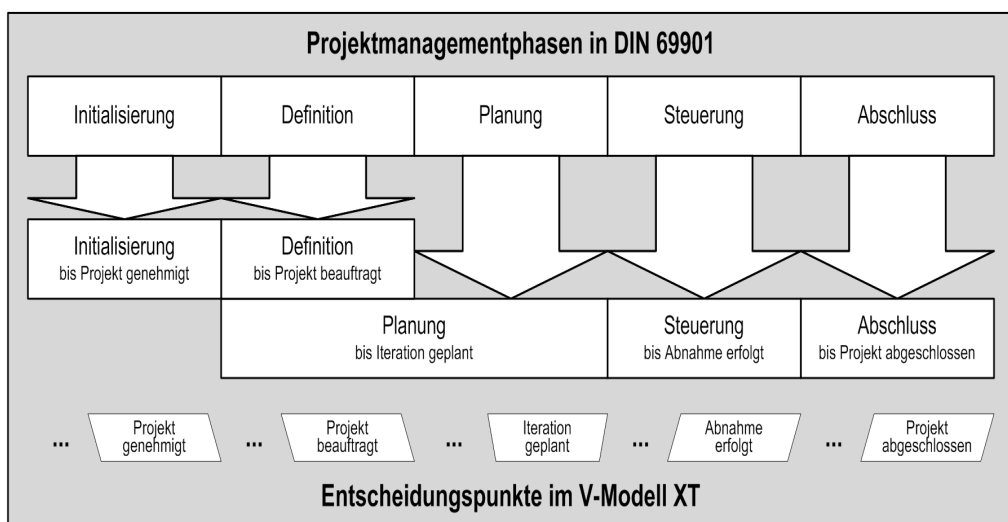
Quelle: DIN 69901-2

Dr. Karl Kollischan, SEE 2010

04. Mai 2010

17

## Konventionsabbildung DIN 69901



Dr. Karl Kollischan, SEE 2010

04. Mai 2010

18



# ISO 21500

Start im Oktober 2007, „Committee Draft“ Anfang 2010, Veröffentlichung geplant für 2012. Beteiligung von Experten aus über 35 Ländern. Basisdokumente u.a. BS 6079, DIN 69901, PMBOK-Guide

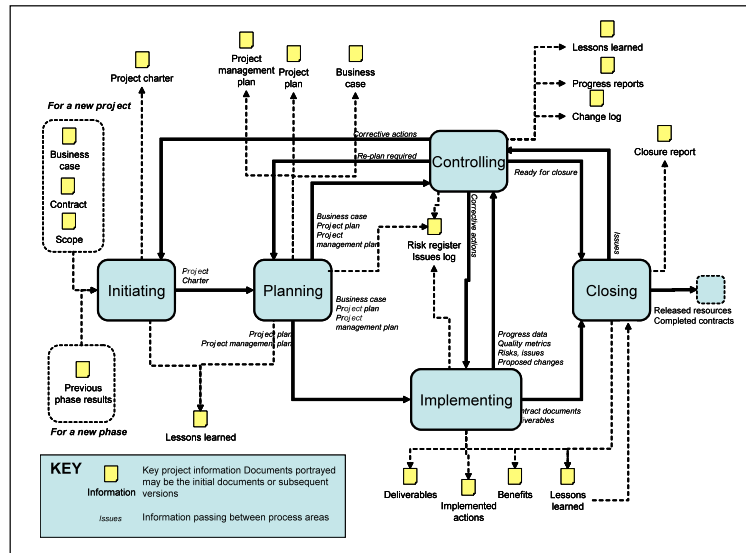
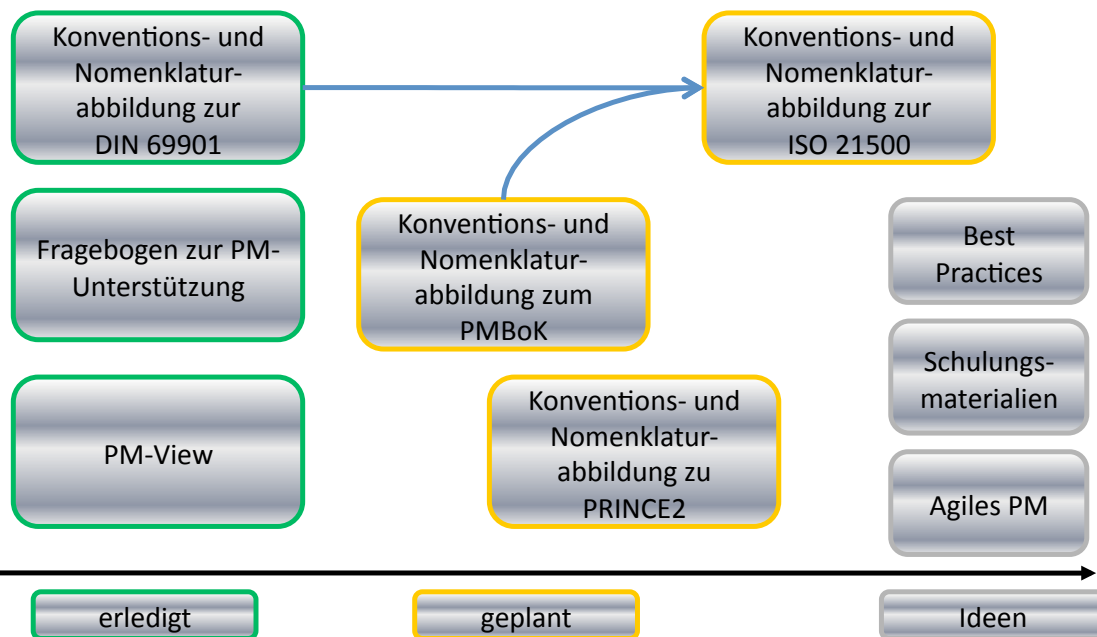


Abbildung: PM Process Group Interactions (Draft)

Quelle: ISO 21500 Clause CD

## Ausblick



Vielen Dank für Ihre Aufmerksamkeit!

---



Fragen  
und  
Anregungen  
?

Kontakt:

[weitagpm@tu-clausthal.de](mailto:weitagpm@tu-clausthal.de)

[karl.kollischan@kobaxx.com](mailto:karl.kollischan@kobaxx.com)



## Tailoring PRINCE2:2009™

Olaf Lewitz

microTOOL GmbH  
Voltastr. 5  
13355 Berlin  
Olaf.Lewitz@microTOOL.de

**Abstract:** PRINCE2™ ist wie das V-Modell® XT eine Methode, die in der Praxis im Allgemeinen angepasst, getailort werden muss. Im Gegensatz zum V-Modell® bringt PRINCE2™ keine Standard-Tailoring-Regeln und kein für dieses Tailoring geeignetes Metamodell mit. PRINCE2™ lässt sich jedoch durchaus im Metamodell des V-Modell® XT abbilden und mit organisationsspezifischen Tailoring-Regeln versehen. Diese Möglichkeiten erläutert der Vortrag an einem konkreten Beispiel.

### 1 Motivation

In diesem Jahr ist die englische Projektmanagement-Methode PRINCE2™ in einer neuer Version erschienen: PRINCE2:2009™. Neben inhaltlichen Änderungen wurden die Struktur vereinfacht und die Anpassbarkeit deutlicher hervorgehoben. Darüber hinaus müssen nicht in jedem Projekt alle Produkte tatsächlich schriftlich als Dokument erzeugt werden.

Das V-Modell® XT bietet ein Metamodell, in dem ein Vorgehen nicht nur in seinen Elementen und Abhängigkeiten, sondern auch in seinen projekt-spezifischen Anpassungsmöglichkeiten abgebildet werden kann. Diese Anpassungen werden durch ein Tailoring definiert. Bildet man die PRINCE2™-Aktivitäten und -Produkte in diesem Metamodell ab, so lassen sich diese zu V-Modell® XT-Vorgehensbausteinen gruppieren, für die Auswahlkriterien festgelegt werden. Ziel dabei ist, dass jedes Projekt nur die für sinnvoll erachteten Aktivitäten planen und Dokumente schreiben muss.

### 2 Tailoring – worum geht es?

Standard-Prozesse oder Vorgehensmodelle definieren Begriffe und Vorgehensweisen für wiederkehrende Arbeitsabläufe. Da Standardmodelle von konkreten Projekten und Organisationen abstrahieren, müssen sie für die Verwendung in Projekten und Organisationen wieder konkretisiert werden. Aus der Bandbreite von Themen, die die



Standardmodelle abdecken, sind vielleicht nicht alle in jedem Projekt gleich relevant, oder manche Organisationen benötigen gar nicht alle abgedeckten Bereiche.

Schon hier unterscheiden sich V-Modell XT und PRINCE2™ – das V-Modell gibt aufgrund seiner Spezialisierung auf IT-Projekte einige Themen vor (z.B. HW-Entwicklung), die eben nur in bestimmten IT-Projekten relevant sind, während PRINCE2™ solche Themen allgemein unter den Begriff „Spezialistenprodukte“ zusammenfasst. Spezialistenprodukte (im Ggs. zu „Managementprodukten“) kommen in jedem Projekt vor, da ja keines ausschließlich aus Management besteht. Daher ist der Tailoring-Ansatz des V-Modells, bestimmte Modellinhalte in manchen Projekten nicht zu benutzen, so nicht auf PRINCE2™ übertragbar.

### 3 Tailoring-Syntax vs. Semantik

Über die konkreten Tailoring-Vorgaben hinaus definiert das V-Modell jedoch auch ein Metamodell, um die Tailoring-Vorgaben abzubilden. Es bringt also für das Tailoring sowohl eine Syntax (eben das Metamodell) als auch eine Semantik (die enthaltenen Projektmerkmale, die Zusammenfassung von Produkten und Aktivitäten in vorgegebenen Vorgehensbausteinen) mit. Die Syntax definiert, dass es Vorgehensbausteine gibt, welche Produkte und Aktivitäten „enthalten“, und dass es Projektmerkmale gibt, deren Werte bestimmen, welche Vorgehensbausteine für ein Projekt zu verwenden sind. Die Semantik der konkreten V-Modell Vorgehensbausteine und Projektmerkmale geht davon aus, dass bestimmte Themen (z.B. HW-Entwicklung) nur in manchen IT-Projekten gebraucht werden und daher nur bei der entsprechenden Festlegung der Projektmerkmale benutzt werden müssen.

Um die Tailoring-Technik des V-Modell XT für PRINCE2™ benutzen zu können, müssen daher Syntax und Semantik getrennt betrachtet werden. Die Syntax lässt sich ohne weiteres verwenden: Auch die PRINCE2™-Produkte und -Aktivitäten lassen sich zu Vorgehensbausteinen zusammenfassen. Es ist jedoch eine andere Semantik zu finden, die nicht von dem einfachen „Weglassen“ bestimmter (nicht-relevanter) Prozess-Teile ausgeht – denn ein solches Weglassen von Prozessen ist in PRINCE2™ nicht vorgesehen.

### 4 Tailoring PRINCE2™

PRINCE2™ ist eine allgemeine, nicht auf eine bestimmte Art von Projekten festgelegte Methode. Die verschiedenen Prozesse und Disziplinen in PRINCE2™ sind eng miteinander verzahnt und dienen alle dem einen Ziel: dem Auftraggeber des Projektes den Business Case so transparent wie möglich zu machen. Er soll – im Idealfall zu jedem Zeitpunkt des Projektes – wissen, wie viel Geld er bereits investiert und welchen Nutzen er dafür bereits bekommen hat, wie viel Geld er noch investieren soll und welchen Nutzen er dafür noch zu erwarten hat.

Das Weglassen von PRINCE2™-Elementen würde unmittelbar zur Verschleierung dieser Transparenz führen und damit diesem Anspruch widersprechen. PRINCE2™ lässt allerdings offen, mit welchem Grad an Formalismus die Prozesse durchgeführt werden. Die Eskalation eines Problems kann beispielsweise mündlich erfolgen (ebenso wie die damit verbundene Klärung) oder schriftlich formalisiert mit einem Ausnahmebericht.

Habe ich Gäste zu Hause und stelle beim Kaffeekochen fest, dass mein Milchschaumer nicht funktioniert, sage ich ihnen einfach, dass ich heute leider keinen Latte Macchiato servieren kann. Passiert das gleiche im Restaurant, werde ich dies vermutlich schriftlich dokumentieren – vielleicht durch ein Hinweisschild. Tritt das Problem häufiger auf, so stelle ich einen schriftlichen Antrag an die Restaurantleitung, dass ich eine neue Maschine brauche, und lasse diesen auch schriftlich genehmigen, bevor ich sie bestelle.

Der Prozess ist in diesen Beispielen immer der gleiche: es wird ein Problem festgestellt, welches die gesetzten Management-Toleranzen überschreitet, es wird eine Entscheidung getroffen, mit welchem Alternativplan diesem Problem zu begegnen ist, und dieser anschließend in die Tat umgesetzt. Der Grad an Formalismus ist jedoch unterschiedlich.

## **5 Toolgestütztes Tailoring für PRINCE2™**

Für die reine Dokumentation des Prozesses ist diese Semantik des Tailorings nicht sehr ergiebig – anschauliche Beispiele wie das im letzten Absatz reichen vermutlich aus, um einen Anwender von PRINCE2™ mit der Fähigkeit auszustatten, diese Methode sinnvoll und pragmatisch einzusetzen. Eine Toolunterstützung des Prozesses bringt aber natürlich immer auch ein gewisses Maß an Formalisierung mit sich – und hier ist es für den Anwender sinnvoll, dieses bei Projektbeginn einstellen zu können.

Mit der Toolunterstützung wird auch ein anderer Aspekt der Formalisierung deutlich: ein Tool kann den Anwender des Prozesses natürlich immer nur so weit unterstützen, wie der Prozess formalisiert ist. Weniger Formalisierung erfordert also eine größere Souveränität des Anwenders, bei PRINCE2™ vor allem des Projektleiters, im Umgang mit der Methode.

Die konkreten Projektmerkmale, die in diesem Vortrag für das Tailoring von PRINCE2™ vorgeschlagen werden, legen also verschiedene Formalisierungsmöglichkeiten für unterschiedliche Prozesse fest. Beim Risikomanagement, dem Umgang mit offenen Punkten und Änderungen kann man zwischen den Optionen „formlos“ (keine Toolunterstützung), „Tool-gestützt“ (Formular-Workflow) und „dokumentiert“ (generierte Dokumente mit eigenem Genehmigungs-Workflow) wählen. Für die Planung und Berichtswesen gibt es ähnliche Optionen.

Zu jedem dieser Merkmale gibt es dann mehrere Vorgehensbausteine, welche die verschiedenen Prozessbestandteile enthalten. Bei dem „einfachsten“ Wert (z.B. „mündlich“) wird kein Vorgehensbaustein gewählt, so dass für dieses Thema dann keine Prozessunterstützung gewählt wird. Das entspricht im Extremfall der Entscheidung, gar

keine Tool-Unterstützung zu brauchen: für das Kochen einer Tasse Tee (ein beliebtes Beispiel in PRINCE2-Schulungen) braucht man wohl kein Projektmanagement-Tool.

## **6 Anwendungsbeispiele**

Zum Abschluss des Vortrags werden zwei praktische Anwendungsbeispiele gezeigt. Einmal wenden wir die beschriebenen Tailoring-Regeln für ein privates Dinner für zehn Personen an – ein Projekt, welches sicherlich keine großen Ansprüche an Formalismus stellt, aber andererseits auch anspruchsvoll genug ist, einen gewissen Planungsaufwand zu rechtfertigen. Anschließend tailorn wir PRINCE2™ für ein größeres Organisationsprojekt, in welchem mehrere Abteilungen einer größeren Firma umstrukturiert werden sollen. Die Gegenüberstellung dieser beiden Projekte macht deutlich, wie gut die Tool-Unterstützung von PRINCE2™ durch ein automatisiertes Tailoring an die konkreten Projektgegebenheiten angepasst werden kann.

Olaf Lewitz  
microTOOL GmbH · Berlin



Software & Systems  
Engineering  
Essentials

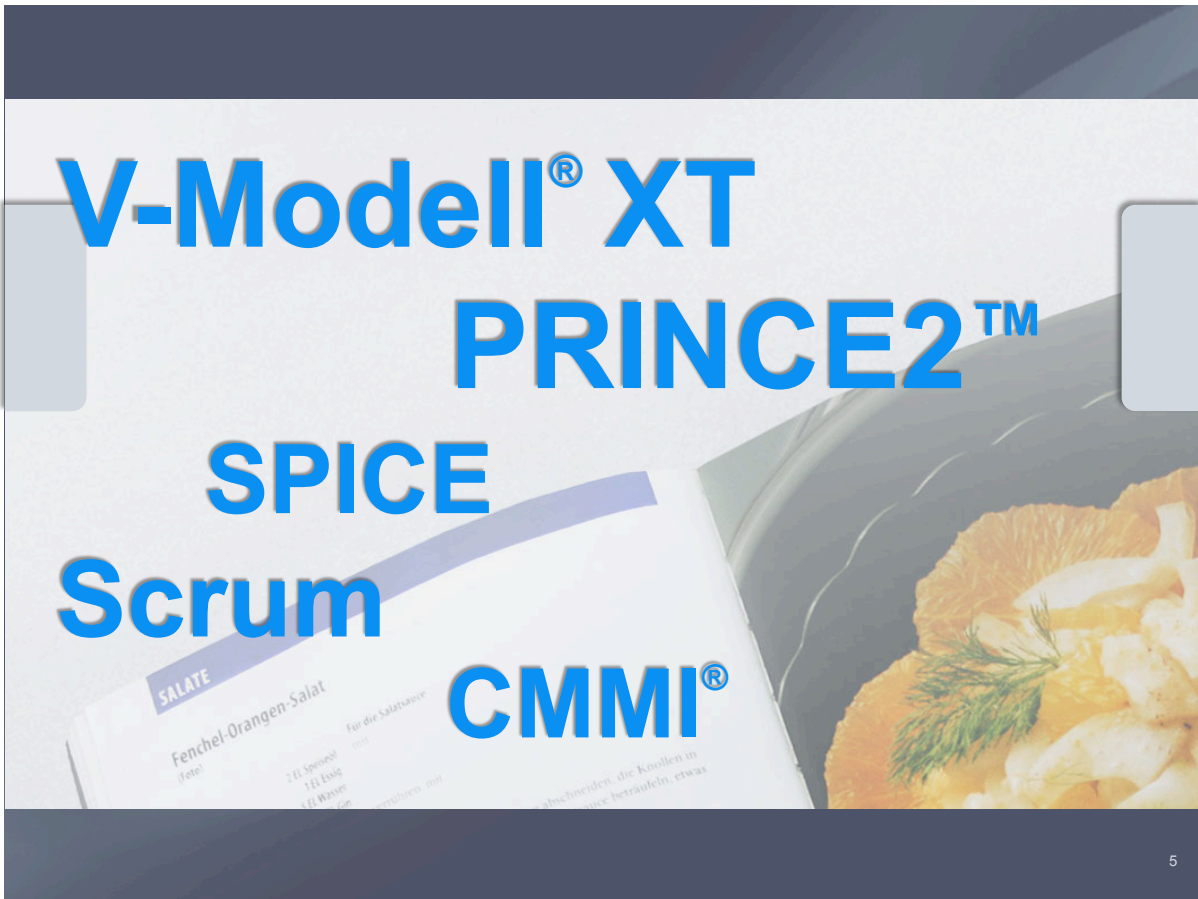
# Tailoring PRINCE2™

Eine Konzeptstudie

© 2010 microTOOL GmbH, Berlin. Alle Rechte vorbehalten.

# Tailoring Um was geht es?









## Tailoring im V-Modell® XT

7



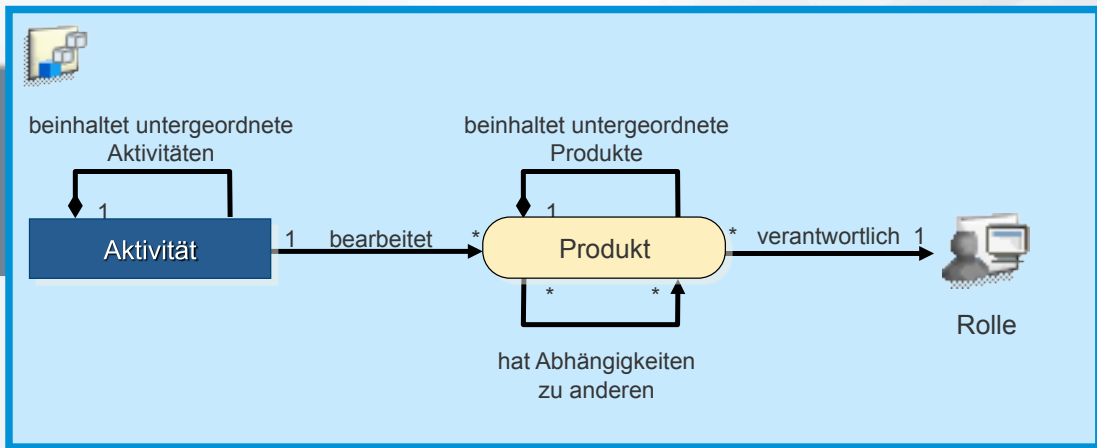
Differenzierung:

- **Syntax**  
Technik des Tailoring
- **Semantik**  
Was wird benutzt?  
Was wird weggelassen?

## Tailoring im V-Modell® XT

8

## Differenzierung: Vorgehensbausteine



## Tailoring im V-Modell<sup>®</sup> XT

9



10



## Semantik:

- Abdeckung verschiedener Disziplinen
- Projekte sind unterschiedlich spezialisiert
- Projekte brauchen andere Teilprozesse
- Beispiel Hardware-Entwicklung

## Tailoring im V-Modell® XT

11

## Tailoring PRINCE2™

12



13

- PRINCE2™ definiert keine Struktur für das Tailoring
- Tailoring bedeutet nicht, etwas wegzulassen!
- Tailoring wird durch den Grad des Formalismus bestimmt

**Tailoring PRINCE2™**

14

Lässt sich die Tailoring-  
Technik  
des **V-Modell® XT** auf das  
Tailoring von  
**PRINCE2™** übertragen?

Frage

15

- Verbinde die Syntax des V-Modell® XT mit der Tailoring-Semantik von PRINCE2™
- Wesentliches Merkmal:  
Grad der Formalisierung

**Idee**

16

- Mündlich
- Liste offener Punkte im Tool
- Dokument
- Ausnahme-Plan



## Eskalation

17



- Einfach
- Formal

## Entscheidungen

18



## Planung

- Grob, pragmatisch
- Fein granular



19

## Berichtswesen

- Ad hoc, bei Bedarf
- Elektronisch im Tool
- Regelmäßiges Dokument



20

## Je Thema zwei Vorgehensbausteine:

- Einen für Toolfunktion – Formulare
- Einen für Dokument

## Drei Optionen

- Formlos – kein Vorgehensbaustein
- Toolgestützt – Formular-Baustein
- Dokumentiert – beide Bausteine

## Tailoring nach Formalismus

21

## Private Dinner for Ten Organisationsänderung

## Zwei Beispiele – live

22



## Steuerung von IT-Projekten durch ein Projekt Management Office

Werner Achtert

TÜV Informationstechnik GmbH  
Leiter IT-Quality  
Langemarckstraße 20  
45141 Essen  
w.achtert@tuvit.de

**Abstract.** Durch verbesserte Methoden zum Projektmanagement können IT-Projekte in der öffentlichen Verwaltung wirtschaftlicher und mit weniger Risiken abgewickelt werden. Ein Projekt Management Office (PMO) trägt zur Bündelung von Knowhow über Projektmanagementmethoden bei und erleichtert die Einführung übergreifender Standards für die Projektsteuerung. Der vorliegende Beitrag stellt Erfahrungen beim Aufbau eines Project Management Office (PMO) für die Bundesverwaltung für Verkehr, Bau und Stadtentwicklung (BVBS) vor. Zentrales Element ist dabei die Integration von unterschiedlichen, in den verschiedenen Organisationseinheiten verwendeten Vorgehensmodellen.

### 1 Ausgangssituation

IT-Systeme in der öffentlichen Verwaltung müssen immer komplexere Geschäftsprozesse unterstützen. Durch die hohen Anforderungen der Anwender an die Funktionalität, die Integration von Verwaltungsabläufen und die Sicherheit steigen die Komplexität der IT-Systeme und damit auch der Aufwand sowie die Risiken in IT-Projekten.

Die Abwicklung von IT-Projekten in großen Organisationen (sowohl Behörden als auch Unternehmen) erfolgt unter vielfältigen Randbedingungen. Einerseits sollen Projekte mit möglichst einheitlichen Vorgehensmodellen abgewickelt werden, um den Aufwand für Schulung und unterschiedliche Werkzeuge zu begrenzen. Andererseits soll für unterschiedliche Projekte die Möglichkeit zur Verwendung der jeweils am Besten geeigneten Vorgehensweisen bestehen, um möglichst gut auf individuelle Anforderungen eingehen zu können.

Viele Behörden verfügen zwar über hervorragende Fachleute für ihre jeweiligen Verwaltungsaufgaben, jedoch nicht über ausreichende Personalressourcen mit spezieller Ausbildung und Erfahrung im Projektmanagement. Dies führt in Projekten einerseits zu Qualitätsmängeln, die z.B. durch einen Bericht des Bundesrechnungshofs deutlich kritisiert werden. Andererseits resultieren daraus teilweise erhebliche Terminverzögerungen mit Auswirkungen auf die Wirtschaftlichkeit und das Image der betroffenen Behörden.

Der Beitrag fasst Erfahrungen aus dem Aufbau eines PMO für die BVBS zusammen. Ausgangspunkt ist dabei eine verteilte Projektorganisation, in der verschiedene Fachbehörden innerhalb eines Ressorts IT-Projekte weitgehend eigenständig abwickeln.



## 2 PMO als Ansatz zur Verbesserung der Projektqualität

Ein Lösungsansatz zur Verbesserung der Projektqualität ist die Einrichtung eines PMO als Serviceeinheit für Projektmanagement und Qualitätsmanagement. In einem PMO werden Spezialisten für die verschiedenen Fachgebiete des Projektmanagements und Qualitätsmanagements zusammengefasst. Dies vereinfacht die systematische Nutzung von Erfahrungen und Best Practices auf der Basis anerkannter Standards wie dem V-Modell XT oder dem Project Management Body of Knowledge (PMBok). Darüber hinaus kann ein PMO die Rolle des unabhängigen Dritten für die Qualitätssicherung in einem IT-Projekt wahrnehmen.

In einem PMO können Mitarbeiter aus der Behörde zusammen mit externen Beratern ressortspezifische Regelwerke zum Projektmanagement und Qualitätsmanagement entwickeln und die Projektleiter bei Planung und Steuerung von IT-Projekten unterstützen. Durch die Zusammenarbeit von internen und externen Mitarbeitern kann ein Knowhow-Transfer zu den Mitarbeitern aus der Behörde stattfinden. Damit werden mittelfristig eigene Kapazitäten zum Projektmanagement und Qualitätsmanagement aufgebaut.

## 3 Aufgaben und Leistungen eines PMO

Ein PMO soll das Knowhow zum Projektmanagement und Qualitätsmanagement innerhalb eines Ressorts bündeln und einheitliche Hilfsmittel für die Projektleiter verfügbar machen. Dazu werden auf Basis von Best Practices und anerkannten Standards Vorlagen, Checklisten und Schulungsmaterial für Projektleiter entwickelt und bereitgestellt.

Grundsätzlich ist das PMO eine Organisationseinheit zur Unterstützung der Projektleiter; die Verantwortung für den Projektverlauf und die Projektergebnisse verbleibt beim Projektleiter. Das PMO unterstützt die Projektleiter in den IT-Projekten durch Schulung, Coaching sowie Moderation von wichtigen Sitzungen. Darüber hinaus kann das PMO im Einzelfall auch die Verantwortung für einzelne Prozesse im Projektmanagement und Qualitätsmanagement übernehmen.

## 4 Methodisches Konzept

Ein PMO definiert Prozesse und Vorgehensweisen zum Projektmanagement als Hilfestellung für IT-Projekte. Dabei ist zu berücksichtigen, dass die meisten Behörden bereits eigene Verfahren zum Projektmanagement eingeführt haben. Diese Vorgehensweisen unterscheiden sich im Detail und werden in den Behörden unterschiedlich konsequent angewendet. Das PMO hat das Ziel, hier ein Mindestmaß an Standardisierung zu erreichen und den Behörden dennoch im Detail Gestaltungsfreiräume zu lassen. Erst auf der Basis dieser Standardisierung kann ein zentrales IT-Projektportfoliomanagement eingeführt werden.

Das im Folgenden beschriebene PMO wurde von Mitte 2009 bis März 2010 beim Dienstleistungszentrum Informationstechnik (DLZ-IT) im Geschäftsbereich des BMVBS eingerichtet. TÜV Informationstechnik GmbH hat das DLZ-IT beim Aufbau des PMO beraten.

In dem hier beschriebenen PMO werden Spezialisten für die verschiedenen Fachgebiete des Projektmanagements zusammengefasst mit folgenden Aufgaben:

- Definition eines übergeordneten, generischen Phasenmodells,
- Definition und Weiterentwicklung der eingesetzten Vorgehensmodelle,
- Festlegung von Regeln für Auswahl und Tailoring eines Vorgehensmodells,
- Schulung der Projektleiter,
- Sammlung und systematische Aufbereitung von Erfahrungswerten
- Coaching der Projektleiter.

Grundsätzlich ist das PMO eine zentralisierte Organisationseinheit zur Unterstützung der Projektleiter; die Verantwortung für den Projektverlauf und die Projektergebnisse verbleibt beim jeweiligen Projektleiter. Das PMO unterstützt die Projektleiter in den IT-Projekten durch Schulung, Coaching sowie Definition von Prozessen und Vorlagen. Darüber hinaus kann das PMO im Einzelfall auch die Verantwortung für ausgewählte Prozesse im Projektmanagement und Qualitätsmanagement übernehmen.

Projektleiter und PMO verfolgen also unterschiedliche Ziele: Während der Projektleiter sein jeweiliges Projekt innerhalb der vorgegebenen Ziele mit einem konkreten Vorgehensmodell abwickelt, schafft das PMO die Voraussetzungen zur Anwendung der verschiedenen Vorgehensmodelle aus der übergeordneten Sicht der Gesamtorganisation.

Das PMO definiert Prozesse und Vorgehensweisen zum Projektmanagement als Hilfestellung für IT-Projekte. Dabei ist zu berücksichtigen, dass die meisten Organisationen bereits eigene Verfahren zum Projektmanagement eingeführt haben. Diese Vorgehensweisen unterscheiden sich im Detail und werden in den verschiedenen Projekten oft unterschiedlich konsequent angewendet. Die Einrichtung des PMO hat das Ziel, hier ein Mindestmaß an Standardisierung zu erreichen und den Projekten dennoch im Detail Gestaltungsfreiräume zu lassen.

Die Konzepte des PMO zur Gestaltung der Projektmanagementprozesse sind daher für die Projektleiter in erster Linie ein Hilfsmittel zur Prüfung der Vollständigkeit ihrer Vorgehensweise. Das methodische Konzept des PMO im hier behandelten Beispiel besteht im Wesentlichen aus den Komponenten

- Generisches Phasenmodell,
- Quality Gates,
- Standardisierte Projektmanagementprozesse sowie
- Zentrales IT-Projektportfoliomanagement.

Das **generische Phasenmodell** des PMO orientiert sich am V-Modell XT und definiert allgemeine Phasen, die in den meisten in der Praxis verwendeten Vorgehensmodellen enthalten sind. Damit wird ein allgemeiner Rahmen für eine Projektstruktur geschaffen, der ein Mindestmaß an Vergleichbarkeit von Projekten gewährleistet. Innerhalb dieser Projektstruktur haben die Projektleiter die Möglichkeit zur Ausgestaltung der jeweiligen Phasen.

Die Phasen eines Projektes werden jeweils abgeschlossen durch **Quality Gates**. Hierzu werden im Rahmen der Phasen-Planung konkrete und messbare Bedingungen für den Abschluss einer Phase definiert. Beim Abschluss der Phase werden diese Bedingungen

durch Maßnahmen der Qualitätssicherung vor dem Übergang in die nächste Phase überprüft.

**Projektmanagementprozesse** dienen der Steuerung von Projekten und können in jeder Phase unabhängig vom Vorgehensmodell durchgeführt werden. Das PMO definiert diese Prozesse in Anlehnung an den Project Management Body of Knowledge (PMBok) des Project Management Institute (PMI), einem international anerkannten und bewährten Standard für Projektmanagement.

Ein Projektportfolio ist eine Sammlung von Projekten mit dem Ziel, eine effiziente Abwicklung der Gesamtheit der Projekte zu ermöglichen und projektübergreifende Ziele einer Organisation zu erreichen. Der wesentliche Schritt vom isolierten Management einzelner Projekte in verschiedenen Behörden zu einem Portfolio ist die Festlegung übergeordneter Ziele für die IT. Im Rahmen eines **IT-Projektportfoliomanagements** werden die Ziele der zuständigen Fachbehörden für einzelne Projekte abgestimmt mit den strategischen Zielen des Ressorts.

## 5 Bewertung

Durch die Einrichtung des PMO wird die Effizienz der Projektausführung deutlich gesteigert. Dies ist derzeit schon in der Pilotphase erkennbar, da Hilfsmittel und Erfahrungen durch das PMO behördenübergreifend ausgetauscht werden.

Die Bündelung der Kapazitäten zum Projektmanagement trägt zu einer höheren Professionalität bei, die Erfahrungen aus Projekten können systematisch gesammelt und allen Projektleitern zur Verfügung gestellt werden. Damit werden Projektrisiken reduziert und die Qualität sowohl der Projektprozesse als auch der Projektergebnisse nachhaltig verbessert.

Die Einführung eines PMO und die damit verbundene Standardisierung von Prozessen stoßen jedoch in vielen Organisationen auf Widerstände. Zum Einen kann die Etablierung von standardisierten Prozessen von den Projektleitern als Einschränkung ihrer persönlichen Gestaltungsmöglichkeiten empfunden werden. Zum Anderen ist die höhere Transparenz und Vergleichbarkeit von Projekten zwar aus Sicht der Gesamtorganisation von Vorteil. Aus Sicht des einzelnen Projektleiters kann dies jedoch als Erhöhung des Leistungsdrucks empfunden werden.

Das hier beschriebene PMO wurde von Mitte 2009 bis März 2010 in einer initialen Version aufgebaut. Das PMO soll in den nächsten Jahren weiter ausgebaut werden durch die Integration weiterer Prozesse aus den Bereichen Anforderungsmanagement, Testmanagement und Lieferantenmanagement.

Parallel zum Aufbau der methodischen Komponenten wurden im Rahmen einer Roadshow die Konzepte in Behörden der BVBS vorgestellt und Pilotprojekte initiiert. Die bisherigen Erfahrungen zeigen, dass das PMO in der hier vorgestellten Form von den Behörden zur Unterstützung der eigenen Projektleiter gut angenommen wird. Bis Ende März 2010 konnten bereits eine Reihe von konkreten Projekten zum Coaching von Projektleitern durchgeführt werden. Mehrere Behörden nutzen mittlerweile die methodischen Konzepte des PMO zum Aufbau eigener Regelwerke für Projektmanagement. Die dabei gewonnenen Erfahrungen fließen wiederum zurück in die Weiterentwicklung des PMO.

Auf diese Weise wird mittelfristig eine stärkere Ausrichtung auf ressortweite Standards erreicht.

Die bisherigen Erfahrungen aus dem Aufbau des hier beschriebenen PMO zeigen allerdings auch die Bedeutung einiger zentraler Erfolgsfaktoren. Der zentrale Faktor bei jeder Standardisierung ist die Akzeptanz der betroffenen Bereiche, in diesem Falle der verschiedenen Behörden. Zur Verbesserung der Akzeptanz wurden die betroffenen Bereiche frühzeitig in den Aufbau des PMO einbezogen durch Abfrage der Bedürfnisse in der praktischen Projektarbeit. Im Gegenzug werden die betroffenen Bereiche regelmäßig über die methodischen Konzepte und das Leistungsangebot des PMO informiert.

Ein weiterer wichtiger Erfolgsfaktor sind Angebote zur Unterstützung in der Projektarbeit. Dabei unterstützen Mitarbeiter des PMO in konkreten Projekten die praktische Umsetzung der methodischen Konzepte durch Schulung und Coaching. Die Praxis der bisherigen Projekte zeigt, dass dieser intensive Austausch zwischen dem PMO und den Projekten wesentlich zur Akzeptanz des PMO beiträgt und ein direktes Feedback zur Anwendbarkeit der methodischen Konzepte liefert.

PMBoK hat sich bei der bisherigen Arbeit im PMO als sehr hilfreich erwiesen, durch die hohe Flexibilität konnten die Projektmanagementprozesse des PMO bisher in allen benutzten Vorgehensmodellen angewendet werden.



## **Integration von unterschiedlichen Vorgehensmodellen durch ein Project Management Office auf Basis des PMBoK**

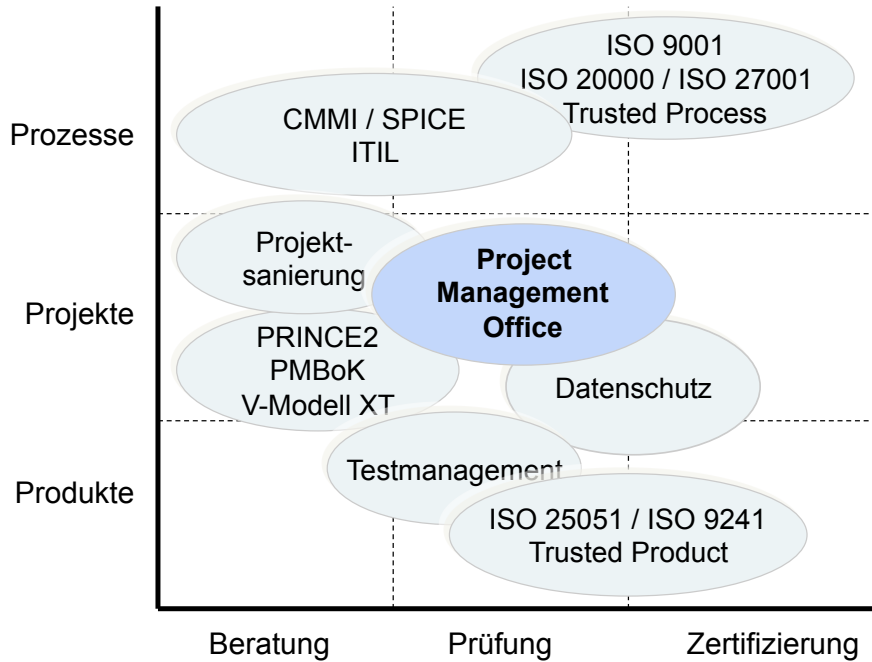
Werner Achtert



### **Agenda**

- Projektmanagement in IT-Projekten
- Einrichtung eines PMO in der BVBS
- PMBoK als methodische Grundlage für das PMO
- Methodische Konzepte
- Bisherige Erfahrungen

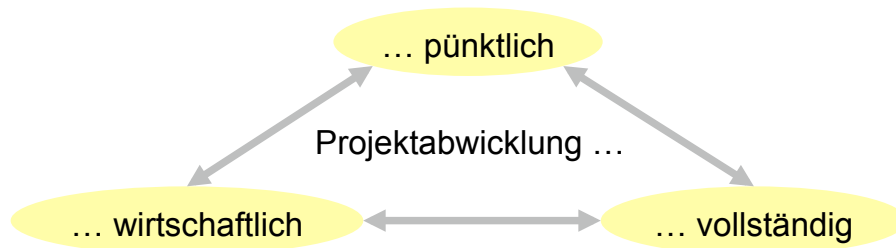
# Dienstleistungen der TÜViT



# Projektmanagement und Qualitätsmanagement

**Projektmanagement ist die Gesamtheit von Führungsaufgaben, -organisation, -techniken und -mitteln für die Abwicklung eines Projektes (DIN 69901)**

**Qualitätsmanagement bezeichnet grundsätzlich alle organisierten Maßnahmen, die der Verbesserung von Produkten, Prozessen oder Leistungen jeglicher Art dienen (ISO 9000)**



## Lösungsansatz zur Qualitätsverbesserung

- Beratung von IT-Projektleitern in Fragen des Projektmanagements
- Standardisierung von IT-Projektstrukturen und -steuerungsprozessen
- Zentrales Projektcontrolling
- Zentrales Portfoliomanagement

## Einrichtung eines Project Management Office (PMO) als Dienstleister für IT-Projekte

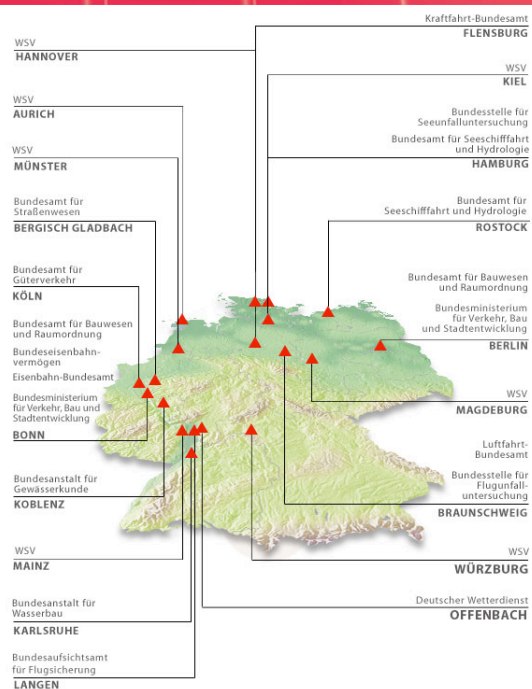
22.04.2010

© TÜV Informationstechnik GmbH – Unternehmensgruppe TÜV NORD

5

## Das PMO der Bundesverwaltung für Verkehr, Bau und Stadtentwicklung (BVBS)

- Die nachgeordneten Behörden in der BVBS entwickeln und betreiben umfangreiche IT-Systeme
- Das Dienstleistungszentrum Informationstechnik (DLZ-IT) nimmt zentrale IT-Aufgaben in der BVBS wahr
- Ab Mitte 2009 wurde im DLZ-IT ein PMO für die BVBS eingerichtet

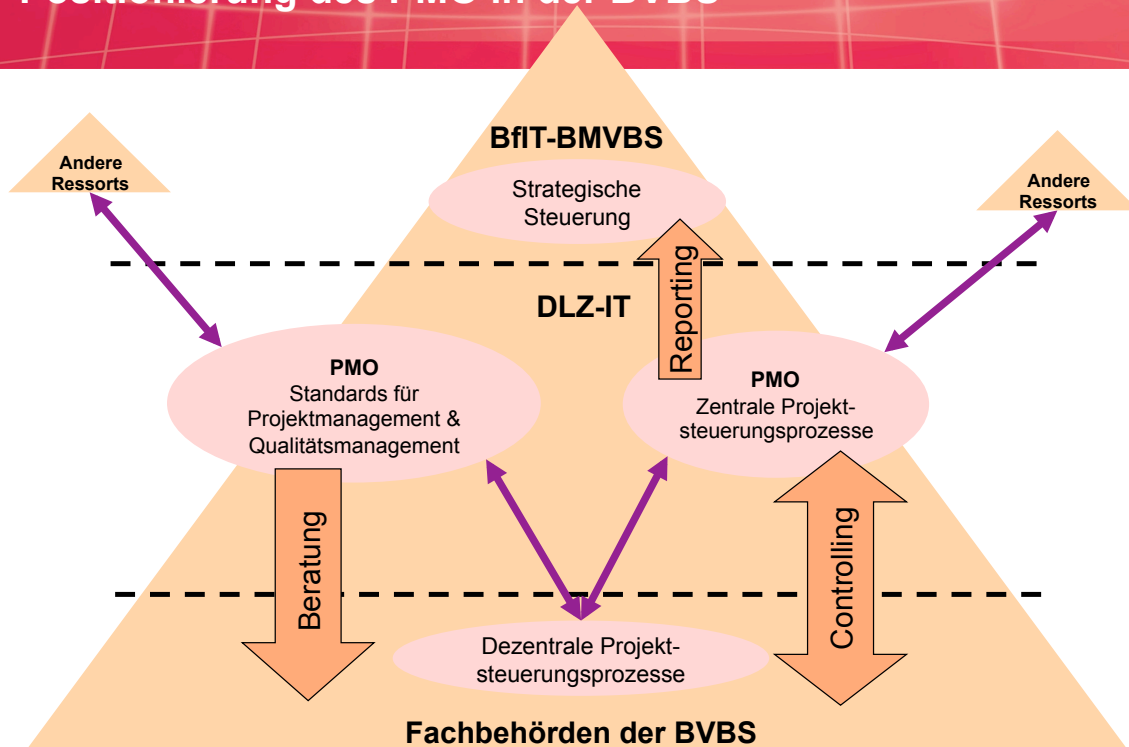


22.04.2010

© TÜV Informationstechnik GmbH – Unternehmensgruppe TÜV NORD

6

## Positionierung des PMO in der BVBS



22.04.2010

© TÜV Informationstechnik GmbH – Unternehmensgruppe TÜV NORD

7

## Leistungen des PMO für Projekte

- Definition von Methoden und Hilfsmitteln
- Schulung von Projektleitern
- Coaching von Projektleitern
- Projektcontrolling
- Unabhängige QS
- Unterstützung beim Aufbau von Regelwerken zum Projektmanagement

22.04.2010

© TÜV Informationstechnik GmbH – Unternehmensgruppe TÜV NORD

8



## Abgrenzung Projektleiter und PMO

- Der Projektleiter ist verantwortlich für die Erreichung der vorgegebenen Projektziele, das PMO schafft die projektübergreifenden Voraussetzungen für die Projektabwicklung.
- Der Projektleiter wendet im Projekt geeignete Methoden an, das PMO stellt Methoden zur Verfügung.
- Der Projektleiter steuert einzelne Projekte, das PMO koordiniert Projekte.

## Guide to the Project Management Body of Knowledge (PMBoK Guide)

- International weit verbreiteter Standard des Project Management Institute (PMI) zum Projektmanagement
- Prozessorientierter Ansatz: Ein Projekt basiert auf dem Zusammenspiel von Prozessen
- Für die Tätigkeiten im Verlauf einer Phase oder eines Projekts sind Prozessgruppen festgelegt
- Die Prozesse sind fachlich in Wissensgebiete gegliedert
- Jeder Prozess ist durch Input, Output und Methoden beschrieben

## Prozessgruppen des PMBoK

- Initiierung
- Planung
- Ausführung
- Überwachung und Steuerung
- Abschluss

## Wissensgebiete des PMBoK

- Integrationsmanagement
- Inhalts- und Umfangsmanagement
- Terminmanagement
- Qualitätsmanagement
- Personalmanagement
- Kommunikationsmanagement
- Risikomanagement
- Beschaffungsmanagement

## Methodische Konzepte des PMO Generisches Phasenmodell

### Initialisierung Gesamtprojekt

Phase 1:  
Definition der fachlichen Prozesse und Anforderungen

Phase 2:  
Definition Lastenheft für IT-System

Phase 3:  
Auswahl interner oder externer Realisierer

Phase 4:  
Definition Pflichtenheft für IT-System

Phase 5:  
Design und Implementierung

Phase 6:  
Test und Abnahme

Phase 7:  
Inbetriebnahme

Inkrementale

Das generische Phasenmodell integriert die unterschiedlichen Vorgehensmodelle der Projekte

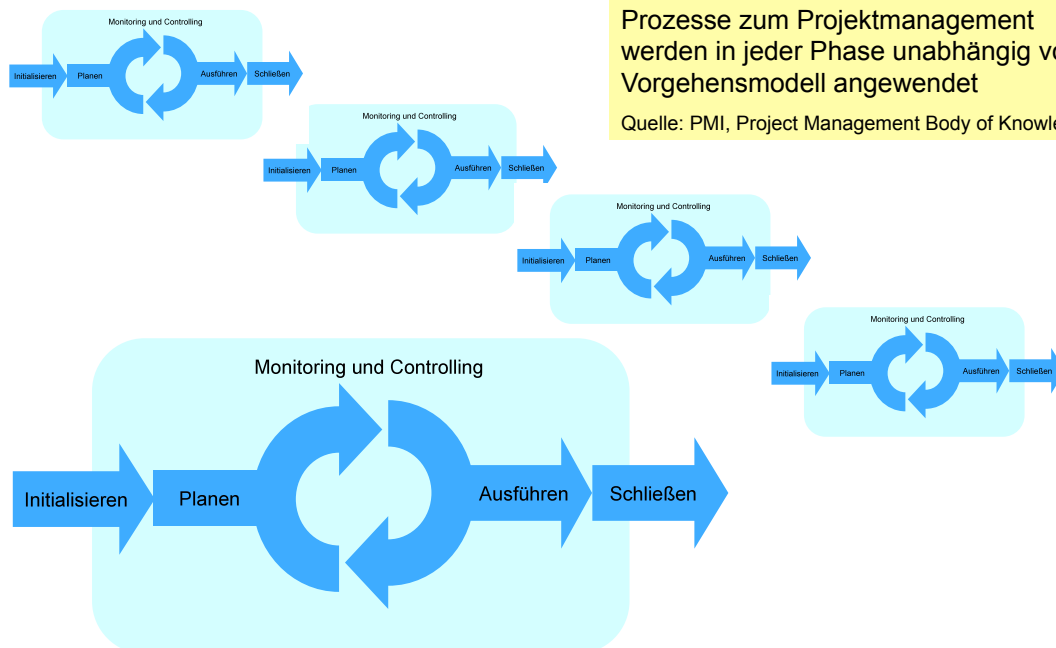
### Abschluss Gesamtprojekt

22.04.2010

© TÜV Informationstechnik GmbH – Unternehmensgruppe TÜV NORD

13

## Methodische Konzepte des PMO Projektmanagement innerhalb einer Phase



Prozesse zum Projektmanagement werden in jeder Phase unabhängig vom Vorgehensmodell angewendet

Quelle: PMI, Project Management Body of Knowledge

22.04.2010

© TÜV Informationstechnik GmbH – Unternehmensgruppe TÜV NORD

14

## Methodische Konzepte des PMO

### Prozesse zum Projektmanagement

Initialisieren	Planen	Management der Ausführung	Monitoring	Steuerung	Schließen
Projektauftrag entwickeln	Projektmanagementplan entwickeln	Projektausführung lenken und leiten	Terminplan und Kosten überwachen	Terminplan und Kosten steuern	Projekt abschließen
Projektbeteiligte identifizieren	Projekt-Umfang festlegen	Qualitätssicherung managen	Qualität überwachen	Qualität steuern	
	Projektzeitplan entwickeln	Personal managen	Risiken überwachen	Risiken steuern	
	Projektbudget planen	Kommunikation managen		Änderungen steuern	
	Qualität planen				
	Personal-bedarfsplan entwickeln				
	Kommunikation planen				
	Risikomanagement planen				

Prozesse in Anlehnung an PMBoK

22.04.2010

© TÜV Informationstechnik GmbH – Unternehmensgruppe TÜV NORD

15

## Methodische Konzepte des PMO

### Einführung zentraler Projektsteuerungsprozesse

- Einsatz von Standard-Kennzahlen und Ermittlung zusätzlicher spezifischer Kennzahlen („GQM-Methode“)
- Einsatz von Mess-Plänen und Mess-Vorlagen zur systematischen Erfassung der Kennzahlen
- Durchführung von strukturierten Dokumenten-Reviews und Jour Fixes
- Bewertung des Projektfortschritt mit Meilensteintrendanalyse (MTA) und Earned Value Analyse (EVA)
- Vorschlag von Maßnahmen zur Steuerung und Korrektur des Projektfortschritts auf der Basis der Kennzahlen, Dokumenten-Reviews und Jour Fixes

22.04.2010

© TÜV Informationstechnik GmbH – Unternehmensgruppe TÜV NORD

16

## Methodische Konzepte des PMO Beispiel für Balanced Scorecard für IT-Projekte

Ergebnisqualität				Kosten			
Ziel	Kennzahlen	Zielwerte	Maßnahmen	Ziel	Kennzahlen	Zielwerte	Maßnahmen
Erfüllung der definierten Anforderungen	Incident-meldungen	<10 pro Woche	Projektbegleitende QS	Einhaltung der Kostenziele	Kumulierte externe Kosten	+/-5%	Monatliche Kontrolle der Kosten und Stunden
	Restfehler bei Produkt-einführung	<5 pro 1000LOC	Reviews in allen Phasen Quality Gate		Kumulierte interne Stunden	+/-5%	Analyse von Abweichungen
Termineinhaltung				Risiken			
Ziel	Kennzahlen	Zielwerte	Maßnahmen	Ziel	Kennzahlen	Zielwerte	Maßnahmen
Erreichung der Terminziele	Fertigstellungsgrad	<10% Abweichung von Planung	Unabhängige Bewertung der Ergebnisse	Minimierung Projektrisiko	Hohe Risiken ohne Maßnahme	0	Regelmäßige Risikobewertung, Brainstorming
	Erreichung Meilensteine	<10% Abweichung von Planung	Meilenstein-trendanalyse		Unvorhergesehene Probleme	<10% der Probleme	Bildung von Risikoklassen

22.04.2010

© TÜV Informationstechnik GmbH – Unternehmensgruppe TÜV NORD

17

## Methodische Konzepte des PMO Quality Gates

- Quality Gates sind Entscheidungspunkte zwischen zwei aufeinanderfolgenden Projektphasen.
- Quality Gates werden durch den Projektmanager oder den Projektlenkungsausschuss durchgeführt
- Für jedes Quality Gate müssen Eingangsbedingungen und Kriterien zum Bestehen des Quality Gates vorab festgelegt werden.
- Wird ein Quality Gate bestanden, kann die nächste Phase begonnen werden. Ansonsten müssen die Ergebnisse der Vorphase nachgebessert werden und das Quality Gate erneut durchschritten werden.

22.04.2010

© TÜV Informationstechnik GmbH – Unternehmensgruppe TÜV NORD

18

## Methodische Konzepte des PMO

**Unabhängiges Risikomanagement**

- Regelmäßige Risikoerhebung durch Workshops mit den Projektbeteiligten
- Risikoanalyse und Bewertung der Risiken mit einer Risiko-Matrix
- Priorisierung der Risiken und Verfolgung der Top-Risiken
- Vorschlag von Maßnahmen zur Risikominderung
- Darstellung und Verfolgung der Risiken mit Kennzahlen und Ampeldarstellungen

## Methodische Konzepte des PMO

**Konzeption für zentrales Projektportfoliomanagement**

- Ein Projekt muss nicht nur unmittelbare Projektziele erreichen, sondern auch einen Beitrag zur Erreichung übergeordneter Geschäftsziele leisten.
- Prozesse des IT-Projektportfoliomanagement
  - Definition der IT-Strategie mit Festlegung der Ziele und Kriterien zur Messung ihrer Erreichung
  - Strategische Planung der IT-Projekte in den verschiedenen Organisationseinheiten
  - Bewertung und Planung einzelner Projekte aus Sicht der IT-Strategie
  - Durchführung der zentralen Projekt-Steuerungsprozesse auf Ebene des gesamten IT-Portfolios



## Bisherige Erfahrungen

- Durch den Austausch zwischen PMO und den IT-Projekten werden Methoden und Erfahrungen Behörden-übergreifend genutzt.
- Praktische Unterstützung wird in den Projekten gut angenommen
- Methodische Konzepte und Coaching in den Projekten führen schrittweise zur Standardisierung im Projektmanagement
- PMBoK ermöglicht flexible Gestaltung des Phasenmodells

**Vielen Dank für Ihre Aufmerksamkeit!**

## **TÜV Informationstechnik GmbH**

Unternehmensgruppe TÜV NORD

Werner Achtert  
Leiter IT Quality

Langemarckstr. 20  
45141 Essen

Telefon: +49 201 8999 – 504  
Telefax: +49 201 8999 – 544  
E-Mail: [w.achtert@tuvit.de](mailto:w.achtert@tuvit.de)  
URL: [www.tuvit.de](http://www.tuvit.de)

# 11. Agilität und Softwareprozesse

## Sessionüberblick

---

11.1. Agil trotz Standards . . . . .	603
11.2. Lean MDD – Minimale Modellierung und maximale Generierung . . . . .	605
11.3. Zwischen Wasserfall und Agilität . . . . .	617

---





## Agil trotz Standards – Agiles Vorgehen in standardisierten Umgebungen

Jens Coldewey

Coldewey Consulting, Toni-Schmid-Str. 10 b  
81825 München  
jens\_coldewey@acm.org

**Abstract:** Agile Verfahren stehen in dem – ungerechtfertigten – Ruf, „Cowboytum“ zu unterstützen. Die Wahrheit sieht ein wenig anders aus: Agile Vorgehen erfordern von den Teams ein hohes Maß an Disziplin und bieten dem Auftraggeber hohe Transparenz. Allerdings fügen sie sich nicht ohne weiteres in standardisierte Umgebungen ein, wie sie in hochkritikaln Projekten oder auch bei öffentlichen Auftraggebern anzutreffen sind. Hier müssen die Verfahren angepasst werden, aber auch die Erwartungshaltung und das Controlling des Auftraggebers. Dafür kann der Auftraggeber mit deutlich verringerten Projektrisiken und wirtschaftlicheren Projektentwicklung rechnen.

### 1 Warum sollten auch standardisierte Projekte agil sein?

Erfahrungsgemäß bietet agile Vorgehensweise dem Auftraggeber eine Reihe von Vorteilen: Der Projektstatus wird an fertiggestellter Software gemessen, wodurch er allen Beteiligten transparent wird. Dies verringert die Gefahr „böser Überraschungen“, wie sie sonst gerne während der Integrations- und Testphase auftreten. Zudem führen die Techniken der test-getriebenen Entwicklung regelmäßig zu wesentlich weniger Fehlern in Produktion und zu flexiblen Architekturen.

Gerade Projekte im öffentlichen Bereich stellen sich aufgrund der komplexen Fachlichkeit immer wieder als sehr riskant heraus, sie stehen unter besonderer Beobachtung der Öffentlichkeit, da hier erhebliche Steuergelder investiert werden. Hohe Transparenz und verringertes Risiko können hier wesentliche Beiträge sein, die Chancen auf einen Projekterfolg zu erhöhen.

Auch bei hochkritikaln Vorhaben sind Transparenz und hohe Code-Qualität wichtige Kriterien. Gerade im medizin-technischen Bereich werden mehr und mehr agile Ansätze verwendet, um diese Qualität zu erreichen.

Gemeinsam ist beiden Domänen, dass sie stark standardisiert und hohe Anforderungen an die Dokumentation der Abläufe stellen. Diesen Anforderungen genügen agile Verfahren „aus dem Buch“ nicht immer ohne Anpassungen.

## 2 Der Mythos Dokumentationsverweigerung

Durch eine Verkürzung des Agilen Manifests hält sich das Gerücht hartnäckig, agile Verfahren würden keine Dokumentation produzieren. Das ist in dieser Form falsch. Agile Verfahren verweigern sich lediglich *überflüssiger* Dokumentation und haben Wege gefunden, ohne einen großen Teil der *internen* Dokumentation auszukommen, um qualitativ hochwertige Software zu erstellen. Wird allerdings Dokumentation für *externe* Zwecke benötigt, zum Beispiel für den Nachweis der Sicherheit zur Abnahme von medizintechnischen Geräten, so wird diese Dokumentation in die „Definition of Done“ aufgenommen. Soweit diese Dokumentation redundant zum Code ist, verfolgt man Generierungsansätze. Soweit sie allerdings sicherheitskritische Bereiche betrifft, z.B. Impact Analysen, steht sie nicht zur Debatte. Ebenso Dokumentation, die zur Nachverfolgbarkeit des Entstehungsprozesses dient.

## 3 Worauf der Auftraggeber achten sollte

Agile Verfahren entfalten einen großen Teil ihrer wirtschaftlichen Attraktivität aus dem Umstand, dass der Auftraggeber ohne großen Aufwand auf neue Erkenntnisse reagieren kann („Change for free“). Dies setzt allerdings voraus, dass sich Auftraggeber und -nehmer auf sehr „kurzem Dienstweg“ einigen können. Der Auftraggeber muss zudem in der Lage sein, die Spezifikationen in Form von Akzeptanztests kurzfristig bereit zu stellen.

Zudem muss bei der Ausschreibung sicher gestellt werden, dass nicht Angebote, die sich erst im Rahmen des „Claim Managements“ wirtschaftlich für den Auftragnehmer lohnen verglichen werden mit Aufträgen, die Änderungen von vorne herein einkalkulieren, da die Ausschreibung sonst massiv zugunsten von Dumpingangeboten verzerrt wird.

## 4 Fazit

Auch in hoch standardisierten Umgebungen ist agiles Vorgehen möglich und empfehlenswert. Allerdings muss dabei auf allen Seiten darauf geachtet werden, dass die Verfahren sich in die Standards einpassen. Dass dies zu Lasten der Wirtschaftlichkeitsrechnungen geht, ist naheliegend, muss aber akzeptiert werden.

## **Lean MDD – Minimale Modellierung und maximale Generierung durch hohe Standardisierung**

Alexander Birke, Timo Nink

Accenture GmbH, Systems Integration & Technology, Campus Kronberg 1  
61476 Kronberg im Taunus  
alexander.birke@accenture.com  
timo.nink@accenture.com

### **1 Zusammenfassung**

Transaktionsorientierte Geschäftsanwendungen folgen in der Regel einem sehr ähnlichen Aufbau. Sie bestehen aus Bedienoberflächen, Verarbeitungskomponenten und zugehöriger Datenhaltung. Innerhalb dieser Bereiche weisen die Komponenten viele gemeinsame Eigenschaften auf und eröffnen damit Potential zur Standardisierung. Basierend auf starker Standardisierung bietet „Lean Model Driven Development“ (Lean MDD) ein konkretes, umfassendes und im Kundenprojekt erprobtes Vorgehen für die Realisierung der verschiedenen Bereiche einer Geschäftsanwendung an.

Die Komponenten jedes der drei Anwendungsbereiche weisen trotz individueller Funktionalität hohe Ähnlichkeiten in ihrer Struktur und ihrem Verhalten auf. Anhand der Ähnlichkeiten lässt sich eine begrenzte Anzahl an verschiedenen Komponententypen ableiten, die selbst wiederum auf wenige, parametrisierbare Grundelemente reduziert werden können.

Es ergibt sich dadurch ein technologieunabhängiges Meta-Modell, das drei sich gegenseitig referenzierende Sub-Meta-Modelle für die Beschreibung von Bedienoberfläche, Logikkomponenten und Datenhaltung beinhaltet.

Basierend auf dem Meta-Modell kann die gesamte Geschäftsanwendung mit einem Modellierungswerkzeug formal beschrieben und einer automatisierten Weiterverarbeitung zugänglich gemacht werden. Durch die Standardisierung des Ablaufverhaltens kommt Lean MDD dabei sogar ohne dynamische Modelle wie Aktivitätsdiagramme aus.

Mittels Generatoren werden die Modelle in unterschiedliche Artefakte transformiert. Zielartefakte im Lean MDD sind neben dem Applikationscode für Oberflächen, Geschäftslogik und Persistenzschicht auch die vollständige, fachliche und technische Dokumentation.

Die praktische Erfahrung hat gezeigt, dass sich der Applikationscode, der durch Transformation aus den drei formalen Modellen des Lean MDD erzeugt wird, durch sehr hohe Konsistenz im Hinblick auf Struktur und Verhalten auszeichnet. Dies wirkt sich positiv

## 11.2 Lean MDD – Minimale Modellierung und maximale Generierung

auf Aspekte der Software-Qualität, wie unter anderem die Wartbarkeit der Anwendung, aus. Sehr einfach können auch grundlegende Architekturprinzipien und Design Patterns in die Generate integriert werden.

Lean MDD setzt in hohem Maße auf Standardisierung und Formalisierung. Dies ist die Grundlage für eine schlanke und damit effiziente Modellierung. Das Ergebnis ist eine deutliche Reduzierung des Spezifikationsaufwandes bei gleichzeitig hohem Generierungspotential. Bei großen Applikationen, die hinsichtlich Oberfläche und internem Aufbau hohe Gemeinsamkeiten aufweisen, kann sich dieser Ansatz bereits bei der ersten Anwendung amortisieren.



**accenture**

*High performance. Delivered.*

**Lean MDD**  
Minimale Modellierung und maximale Generierung  
durch hohe Standardisierung

Alexander Birke und Timo Nink  
SEE 2010 Köln, 04. Mai 2010

Copyright © 2010 Accenture All Rights Reserved. Accenture, its logo, and High Performance Delivered are trademarks of Accenture.

Vorstellung

<p><b>Alexander Birke</b></p> <p><b>accenture</b></p> <p>Accenture GmbH Maximilianstraße 35 D-80539 München</p> <p>Telefon: +49 89 93081-68708 Mobil: +49 175 57-68708</p> <p>alexander.birke@accenture.com</p>	
<p><b>Timo Nink</b></p> <p><b>accenture</b></p> <p>Accenture GmbH Campus Kronberg 1 D-61476 Kronberg</p> <p>Telefon: +49 6173 94-67764 Mobil: +49 175 57-67764</p> <p>timo.nink@accenture.com</p>	

## Agenda

### Model Driven Development in Kürze

- Von MDA und MDSD zu Lean MDD

### Lean MDD

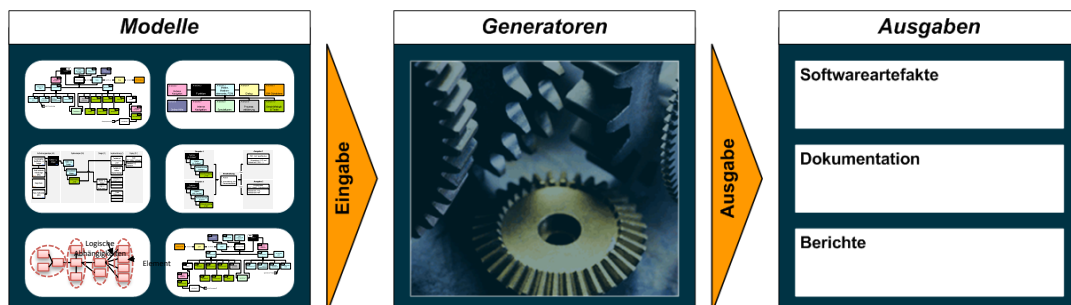
- Konzepte
- Modellierung
- Anwendung

### Zusammenfassung

- Erfahrungen und Ausblick

## Model Driven Development (MDD) in aller Kürze

In der modellgetriebenen Software-Entwicklung werden formale Modelle verwendet, um die zu entwickelnde Anwendung zu spezifizieren. Aus den Modellen werden Artefakte der Anwendung automatisiert erzeugt.



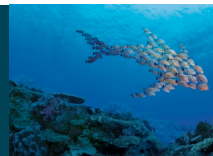
## Motivation für Lean MDD



MDD ist ein Oberbegriff für verschiedene Spielarten modellgetriebener Software-Entwicklung. In unserem Einsatzszenario waren die „gängigen“ Ansätze allerdings nur bedingt geeignet.

- Model Driven Architecture (MDA)
  - Starke Betonung der Plattformunabhängigkeit
  - Mehrere Modell-Ebenen
  - Modell-zu-Modell-Transformationen
  - Ausschließlich UML als Beschreibungssprache
- „Klassisches“ Model Driven Software Development (MDSD)
  - Entwicklung für konkrete Zielplattform
  - Viele verschiedene Modelle für die Spezifikation von Struktur und ggf. Verhalten
  - Modellierung sehr nah an den technischen Komponenten der Software
  - Beliebige Notationen für die Beschreibung möglich

## Lean MDD



Lean MDD ist eine spezielle Ausprägung von MDD, bei der der Umfang der Modellierung durch konsequente Standardisierung signifikant reduziert und gleichzeitig ein hoher Generierungsgrad erreicht wird.

- Identifikation von Mustern in Struktur und Verhalten
- Reduzierung der Modellierung auf das Notwendigste
  - Wenige Modelle und Beschreibungssprachen
  - Keine Modellierung von Ablauflogik
  - Modellierung mit höherwertigen, fachlichen Konzepten
- Umfassende Weiterverarbeitung der Modellinhalte



## Kontext für die Anwendung von Lean MDD



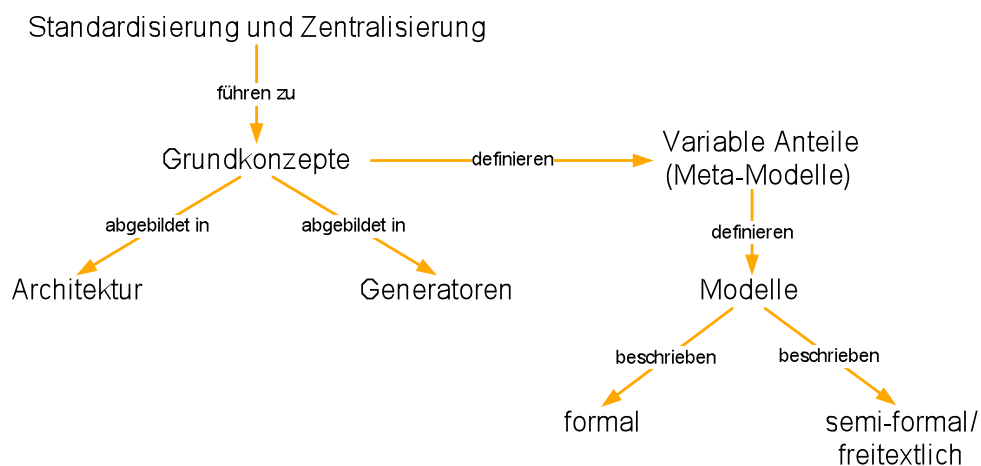
Lean MDD ist besonders für die Entwicklung von großen, datengetriebenen Geschäftsanwendungen geeignet.

- Geprägt durch Eingabe und Ausgabe von Daten
- Anlegen, Ändern, Anzeigen, Löschen und Suchen von Datensätzen als primäre Benutzerinteraktionen
- Keine oder nur wenige Algorithmen und komplexe Berechnungen
- Benutzerdialoge mit geringer Anzahl wiederkehrender Muster

## Standardisierung und Zentralisierung als Ausgangspunkt für Lean MDD



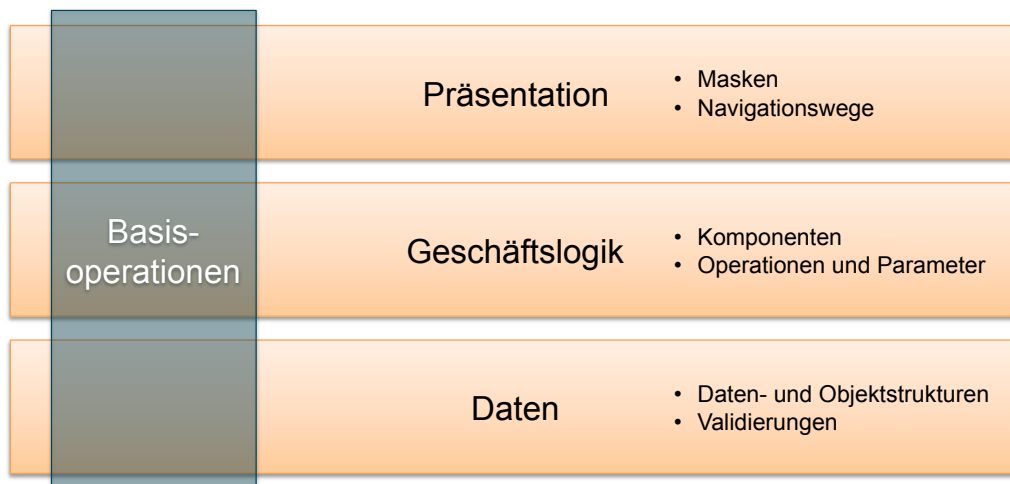
Grundkonzepte beinhalten Struktur- und Verhaltensmuster und werden in Architektur-Komponenten sowie Generatoren abgebildet. Die explizite Modellierung beschränkt sich auf die variablen Anteile.



## Basisoperationen und Architekturschichten als Rahmen für Grundkonzepte und Modelle



Ausgangspunkt für die Standardisierung sind die fünf Basisoperationen (Anlegen, Lesen, etc.) auf den Daten. Die Schichten der Software-Architektur geben die verwendeten Modelle vor.



Copyright © 2010 Accenture All Rights Reserved.

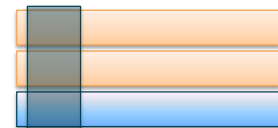
9

## Grundkonzepte und Modelle – Daten



### Grundkonzept:

- Zentralisierte Beschreibung von Validierungsregeln an den zugehörigen Daten
- Hierarchie von Datentypen



### Modell:

- Entitäten, Attribute, Beziehungen und Datentypen als Bausteine für die Struktur des Datenmodells
- Spezifische Beschreibungssprachen für Verhaltensaspekte, u.a. Validierungen



Copyright © 2010 Accenture All Rights Reserved.

10

## Grundkonzepte und Modelle – Geschäftslogik

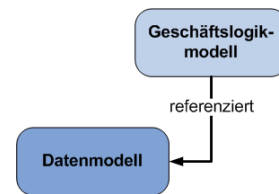
### Grundkonzept:

- Standardisierung der Verarbeitungslogik für Basisoperationen
- Aufteilung der Geschäftslogikschicht in Komponenten mit Operationen, die auf den Basisoperationen beruhen



### Modell:

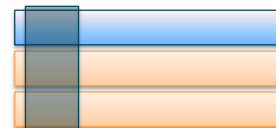
- Modellierung der Ein- und Ausgabedaten von Operationen
- Spezifikation von über die Basisoperationen hinausgehende Geschäftslogik



## Grundkonzepte und Modelle – Präsentation

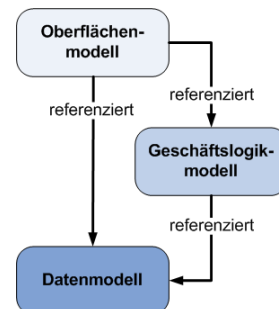
### Grundkonzept:

- Standardisierung von Struktur und Verhalten der Benutzerdialoge anhand der Basisoperationen
- Parametrisierbare Oberflächenelemente für die Ausgestaltung der Masken von Dialogen



### Modell:

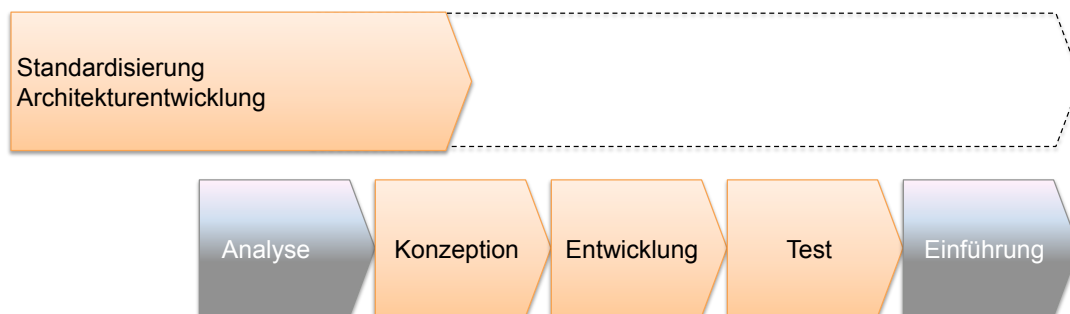
- Modellierung von Benutzerdialogen auf Basis des Oberflächenbaukastens
- Verknüpfung von Daten- und Aktionselementen mit korrespondierenden Elementen in Daten- und Geschäftslogikmodell
- Spezifikation von Navigationswegen zwischen Dialogen



## Lean MDD im Software-Entwicklungsprozess



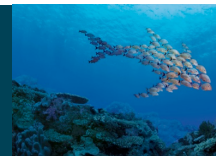
Die Grundlagen werden in einer vorgelagerten Standardisierungs- und Architekturphase gelegt. In den Phasen Konzeption, Implementierung und Test werden die Modelle erstellt und automatisiert weiterverarbeitet.



Copyright © 2010 Accenture All Rights Reserved.

13

## Lean MDD im Software-Entwicklungsprozess – Standardisierungs- und Architekturphase



Eine umfassende Analyse des Standardisierungspotentials und die Bereitstellung einer tragfähigen Entwicklungs- und Laufzeitarchitektur sind erfolgskritisch.



- Identifikation von Mustern und Ableitung von Standards
- Bereitstellung der Modellierungswerkzeuge und Modelle
- Entwicklung einer auf die Grundkonzepte abgestimmten Software-Architektur
- Entwicklung von Generatoren für unterschiedlichste Artefakte

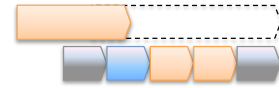
Copyright © 2010 Accenture All Rights Reserved.

14

## Lean MDD im Software-Entwicklungsprozess – Konzeption



Die fachlichen Anforderungen an die Anwendung werden in den Modellen spezifiziert. Der Spezifikationsaufwand ist durch den vorgegebenen Rahmen und komfortable Werkzeugunterstützung gering.

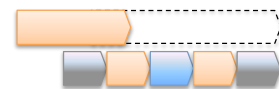


- Die Modellierung beschränkt sich in vielen Fällen auf den Benutzerdialog und die zu verarbeitenden Daten
- Individuelle Geschäftslogik wird nur in wenigen Ausnahmefällen explizit spezifiziert
- Abnahmedokumente und Prototypen der Oberfläche werden durch Generatoren automatisiert erstellt

## Lean MDD im Software-Entwicklungsprozess – Entwicklung



Die Modellinhalte werden durch plattformsspezifische Generatoren weiterverarbeitet. Die Standardisierung und der hohe Generierungsgrad steigern die Effektivität und Effizienz der Software-Entwicklung.

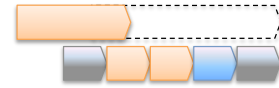


- Zusätzliche technische Designs müssen nur in Ausnahmefällen erstellt werden
- Große Teile der Anwendung werden aus den Modellen vollständig oder als Rumpfe generiert
- Technische Dokumentationen der Anwendung werden durch Generatoren automatisch erstellt

## Lean MDD im Software-Entwicklungsprozess – Test

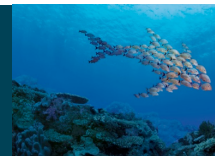


Der Fokus liegt auf dem Test nicht generierbarer Teile der Anwendung. Die Testvorbereitung und -durchführung wird durch Automatisierung unterstützt.

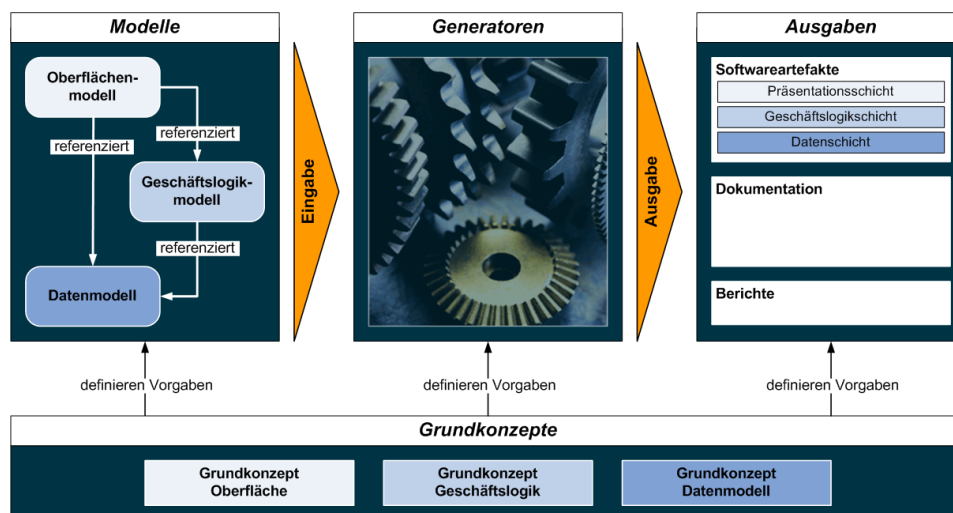


- Durch Beziehungen zwischen Modellelementen und Testbedingungen sowie Änderungsverfolgung kann der Testumfang leicht ermittelt werden
- Skripte und Vorlagen für die Testdurchführung können automatisch erstellt werden
- Testdaten können mit Hilfe generierter Routinen angelegt werden

## Lean MDD im Überblick



Auf Basis der Grundkonzepte und wenigen, schlanken Modellen werden mittels Generatoren in großem Umfang Artefakte für alle Phasen der Software-Entwicklung erzeugt.



## Erfahrungen mit Lean MDD



Lean MDD wurde von uns erfolgreich bei der Entwicklung eines großen, transaktionalen Internet-Portals eingesetzt.

- Mehr als 100 Masken
- Großes Datenmodell mit mehr als 100 Entitäten
- Hohe Anforderungen an Benutzerfreundlichkeit und Barrierefreiheit
- Hohe Performanz-Anforderungen
- Projektteam mit 50 Mitarbeitern

## Ausblick



Aufbauend auf den bisherigen Erfahrungen werden die Ideen und Konzepte von Lean MDD fortlaufend weiterentwickelt.

- Erhöhung des Standardisierungsgrades
- Optimierung der Architekturkonzepte
- Ausweitung auf andere Anwendungstypen

## Zwischen Wasserfall und Agilität – Softwareprozesse für die Entwicklung von Informationssystemen

Volker Gruhn<sup>a</sup>, Clemens Schäfer<sup>b</sup>

<sup>a</sup> Professur für Angewandte Telematik/e-Business, Universität Leipzig,

Klostergasse 3, 04109 Leipzig

<sup>b</sup> it factum GmbH,

Lichtenbergstr. 8, 85748 Garching b. München

gruhn@ebus.informatik.uni-leipzig.de

schaefer@it-factum.de

**Abstract.** Der Übergang von schwergewichtigen Softwareprozess-Modellen hin zu einfacheren Alternativen ist in der Praxis noch nicht durchgängig gelungen. Daher stellen wir mit dem Ansatz des No-Frills-Software Engineering eine ganzheitliche Herangehensweise für die Entwicklung von Informationssystemen vor, die auf den Vorteilen agiler Vorgehensweisen aufsetzt, zugleich aber auch in Situationen einsetzbar ist, in denen rein agile Verfahren üblicherweise nicht in Betracht kommen.

### 1 Motivation

Informationssysteme stellen das Rückgrat der Geschäftsprozessunterstützung von Finanzdienstleistern wie Versicherungen dar. Solche Anwendungen sind in der Regel das Ergebnis von Individualentwicklungen, sei es im eigenen Haus mit einer Abteilung, die die Anwendungsentwicklung übernimmt, oder durch externe Dienstleister. Dabei zeichnen sich die Systeme durch eine vergleichsweise geringe algorithmische Komplexität aus und werden von komplexen, historienbehafteten Datenstrukturen bestimmt. Zudem sind die Systeme in der Regel in heterogene, historisch gewachsene Anwendungslandschaften integriert.

Die Erfahrung zeigt, dass optimale Vorgehensmodelle zur Entwicklung derartiger Informationssysteme der jeweiligen Entwicklungssituation angepasst werden müssen. In bestimmten Situationen haben eher formale Prozessmodelle (wie z.B. der Rational Unified Process) ihre Stärken, in anderen Situationen sind agile Verfahren von Vorteil. Hieraus entsteht die Herausforderung, die Wahl von geeigneten Vorgehensmodellen projektabhängig objektiv und deterministisch zu gestalten, um situativen Festlegungen entgegenzuwirken. Zudem ergibt sich die Anforderung, die Integration der Anwendungsentwicklung in die Unternehmensprozesse (Schnittstellen zum Fachbereich, Dokumentation usw.) sich einheitlich gestalten sollte, egal, ob die eigentliche Entwicklung agil oder eher einem formaleren Modell folgend abläuft.



## 2 Ansatz

Diese Herausforderungen waren Ausgangspunkt für die Entwicklung des so genannten No-Frills Software Engineering (No Frills = ohne Schnickschnack). Dabei handelt es sich um eine Best-Practice Sammlung zur Entwicklung von Informationssystemen.

Hauptelement des No-Frills Software Engineering ist ein generischer Softwareprozess mit definierten Variationspunkten, die unterschiedliche Ausprägungen (Varianten) je nach Projektsituation erlauben, ohne den Gesamtprozess verlassen zu müssen. So ist es möglich, sowohl agile Projekte als auch eher wasserfallartige Projekte unter der Ägide eines Prozessmodells durchzuführen, was die Verankerung des Modells im Unternehmen erleichtert.

Wesentliches Element eines No-Frills Prozesses ist eine vorgeschaltete Triage mit eindeutig definierten Kriterien, nach welchen Prozessprinzipien ein Projekt durchgeführt werden soll (iterativ/agil, eher sequentiell usw.). Zu diesen Kriterien gehören unter anderem Innovationsgrad, Integrationsgrad mit Umsystemen, Projektumfang oder Sourcing-Optionen. Diese Triage führt dazu, dass anhand der Kriterien eine Auswahl einer Prozessvariante erfolgt.

Bei der Einführung des No-Frills Software Engineering werden die möglichen Prozessvarianten für ein Unternehmen definiert. Dabei wird auf die Sammlung von Prozessfragmenten zurückgegriffen, die das No-Frills Software Engineering bereithält. Darin enthalten sind Elemente eher formaler Verfahren wie RUP, aber auch agile Ansätze wie Extreme Programming oder SCRUM [SI07]. Ergänzt werden die Prozessfragmente durch Dokumentenvorlagen, die sich hinsichtlich Struktur und Inhalt bei dieser Systemklasse bewährt haben.

Die Fokussierung auf Informationssysteme zur Geschäftsprozessunterstützung hat zur Folge, dass der Behandlung von Ungewissheit im Entwicklungsprozess (z.B. späte Anforderungen) [Leh89] einen hohen Stellenwert eingeräumt werden muss. Praktisch umsetzbar werden solche Anforderungen durch die so genannten No-Frills Prinzipien [GS09], die in leicht verständlicher Form praktische Hilfestellung geben und von Teammitgliedern gut angenommen werden (Embodiment). Zudem stellt die Integration von wertorientierten Ansätzen [BS00] wichtige Hilfestellung für die Anwendungsentwicklung im kommerziellen Kontext zur Verfügung.

## 3 Bewertung

Erfahrungen mit Versicherungsunternehmen zeigen, dass sich eine ganzheitliche, aber schlanke Vorgehensweise bei der Entwicklung von Informationssystemen als nützlich erwiesen hat. Durch die Triage wird es möglich, nachvollziehbar geeignete Prozessvarianten zu wählen, die letztendlich dafür sorgen, dass qualitativ hochwertige Entwicklungsergebnisse erzielt werden, die sich durch eine gute Fokussierung auf den jeweiligen Einsatzzweck der Anwendung auszeichnen.

## Literaturverzeichnis

- [SI07] Schwaber, K.; Irlbeck, T.: Agiles Projektmanagement mit Scrum. Microsoft Press Deutschland, 2007.
- [Leh89] Lehman, M. M.: Uncertainty in computer application and its control through the engineering of software, *Journal of Software Maintenance: Research and Practice*, 1(1):3–27, September 1989.
- [BS00] Boehm, B.; Sullivan, K.: Software economics: A roadmap. In A. Finkelstein, editor, *The Future of Software Economics*. ACM Press, 2000.
- [GS09] Gruhn, V.; Schäfer, C.: No-Frills Software Engineering for Business Information Systems, *Proceedings of the 8th International Conference on Software Methodologies, Tools and Techniques (SoMeT\_09)*, Prague, IOS Press, 2009.

## Zwischen Wasserfall und Agilität – Softwareprozesse für die Entwicklung von Informationssystemen

Volker Gruhn  
Professur Angewandte Telematik / e-Business  
Universität Leipzig  
Stiftungslehrstuhl der  
Deutschen Telekom

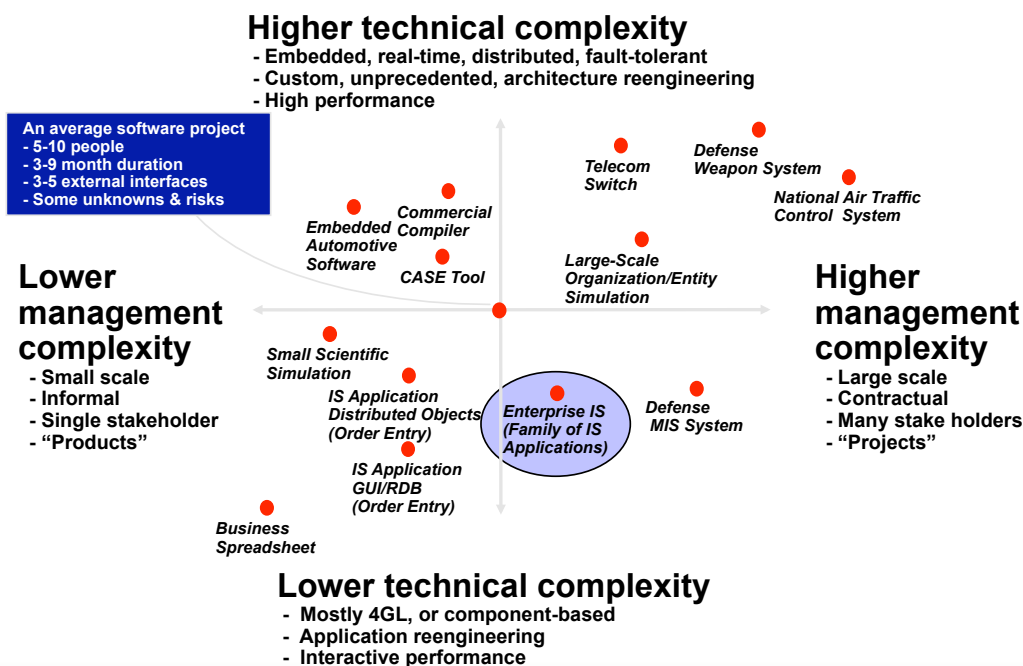
info@lpz-ebusiness.de  
www.lpz-ebusiness.de  
+49 (341) 97 323 30

Clemens Schäfer  
it factum GmbH  
Garching b. München

schaef@it-factum.de  
www.it-factum.de  
+49 (89) 5484 2751



## Dimensionen der Software-Komplexität



Grady Booch 2007

## Herausforderungen bei der Entwicklung von Informationssystemen

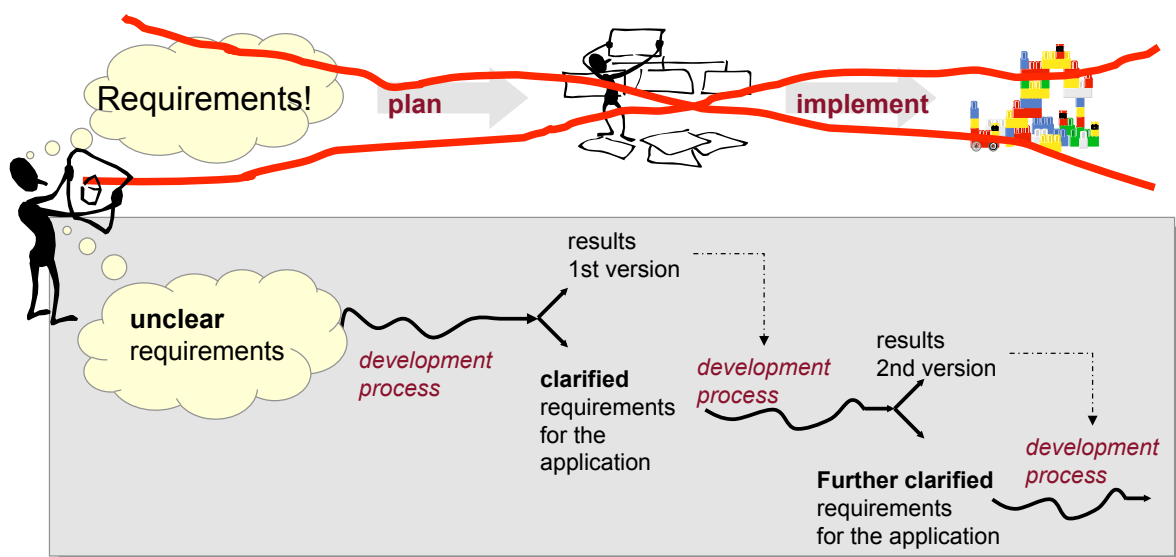
- Breite und diversifizierte Bedingungen und Umgebungen
  - Anpassung an Organisationen, Management- und Systementwicklungsmethoden
  - Projekte zunehmend unterschiedlich im Hinblick auf Größe, Anwendungsdomäne, zu Grunde liegende Technologie
- Erweiterung zu neuen, bislang nicht unterstützen Geschäftsprozessen und Anwendungsdomänen
- Sich schnell ändernde technische Möglichkeiten
  - Zunehmende Komplexität
  - Ablenkung von tatsächlich kritischen Aspekten

K. Kautz, S. Madsen, J.Nørbyerg. Persistent problems and practices in information system development. *Information Systems Journal*, 17(3):217-239, 2007

it factum

3

## Software Engineering ≠ Industrielle Produktion

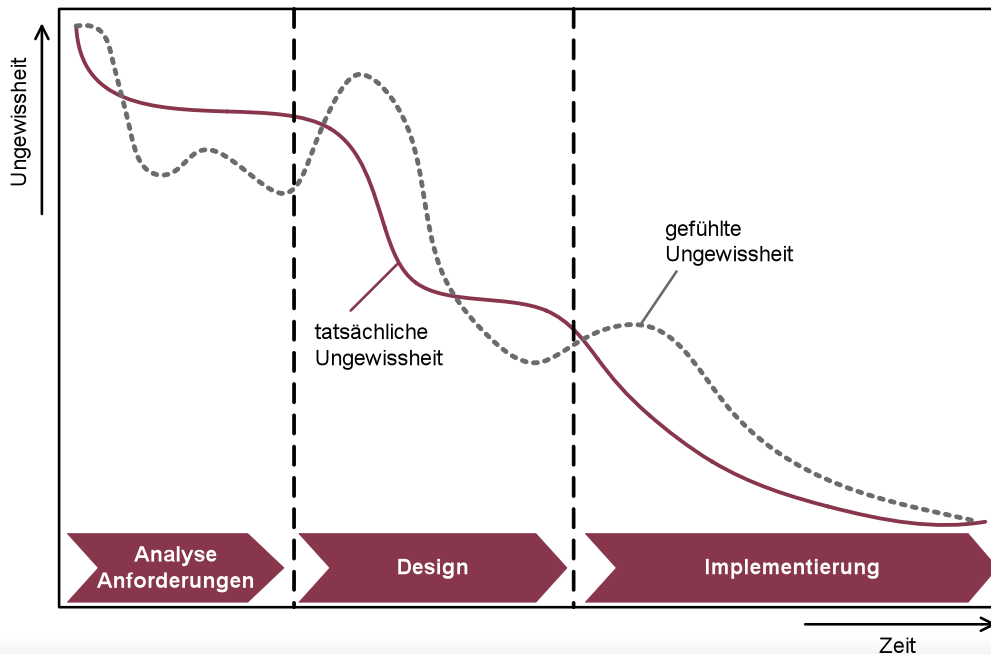


M.M. Lehman, The Uncertainty Principle of Computer Applications, ISPW 1998

it factum

4

## Ungewissheit im Softwareprozess



it factum

5

## Anforderungen an Softwareprozesse

- Anforderungen für erfolgreiche Softwareprozesse
  - Prominente Verankerung von Anwendungsdomänenwissen
  - Explizite Behandlung von Unsicherheit/Ungewissheit
  - Klare und verständliche Deliverables
  - Methodische Unterstützung
  - Pragmatismus und Zielorientierung
  
- Häufige Grenzen der Prozessunterstützung
  - Zu fokussiert auf „technische“ Prozessaspekte (Aktivitäten, Rollen, ...)
  - Zu wenig konkrete Hilfestellung für Artefakte (Dokumente, Templates)
  - Fehlen einer tiefgreifenden Unterstützung für angemessenen Inhalte, Abstraktionsniveaus, Konsistenz

it factum

6

## No-Frills Software Engineering®

UNIVERSITÄT LEIPZIG

 LPZ  E-BUSINESS  
 applied telematics

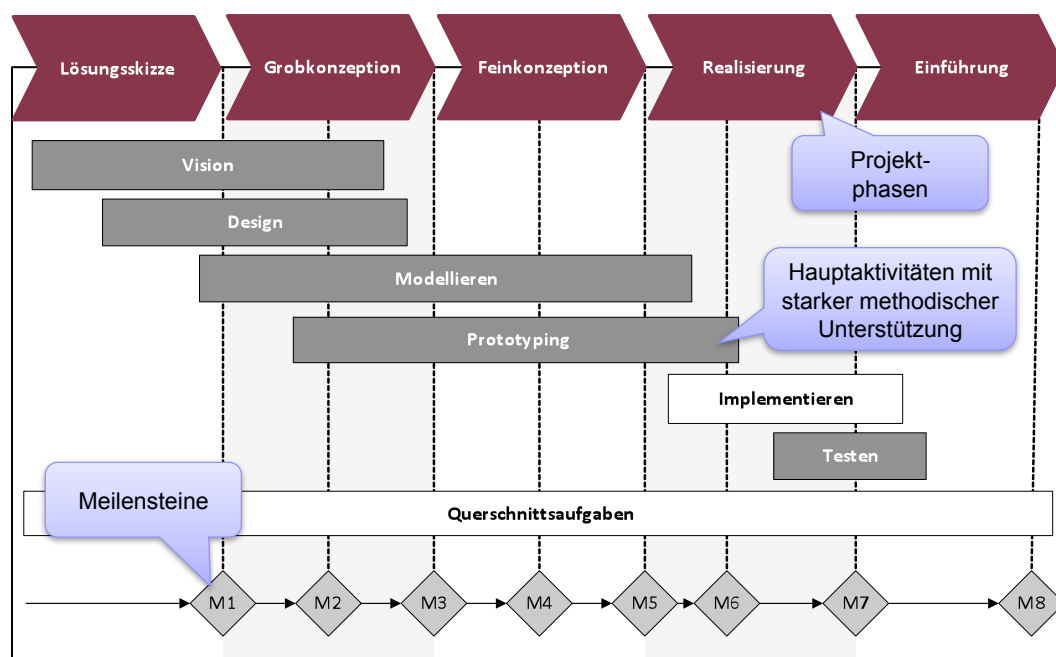
- Es gibt nicht **den** Softwareprozess für Informationssysteme
  - Unterschiedliche Ausprägungen für unterschiedliche Unternehmen
  - Varianten innerhalb eines Unternehmens (agilere, bürokratischere)
- NFSE ist Sammlung von Prinzipien und Methoden
  - „Off-the-shelf“-Softwareprozess bildet die Basis der Softwareentwicklung mit NFSE
  - Hauptaktivitäten des Softwareprozesses (WAS) werden durch NFSE-Methoden unterstützt (WIE)
- Motivation für NFSE: aktuelle Trends und Erkenntnisse
  - Prominente Verankerung von Anwendungswissen
  - Behandlung von Ungewissheit
  - Wertorientierung

it factum

7

## Hauptaktivitäten und Projektphasen

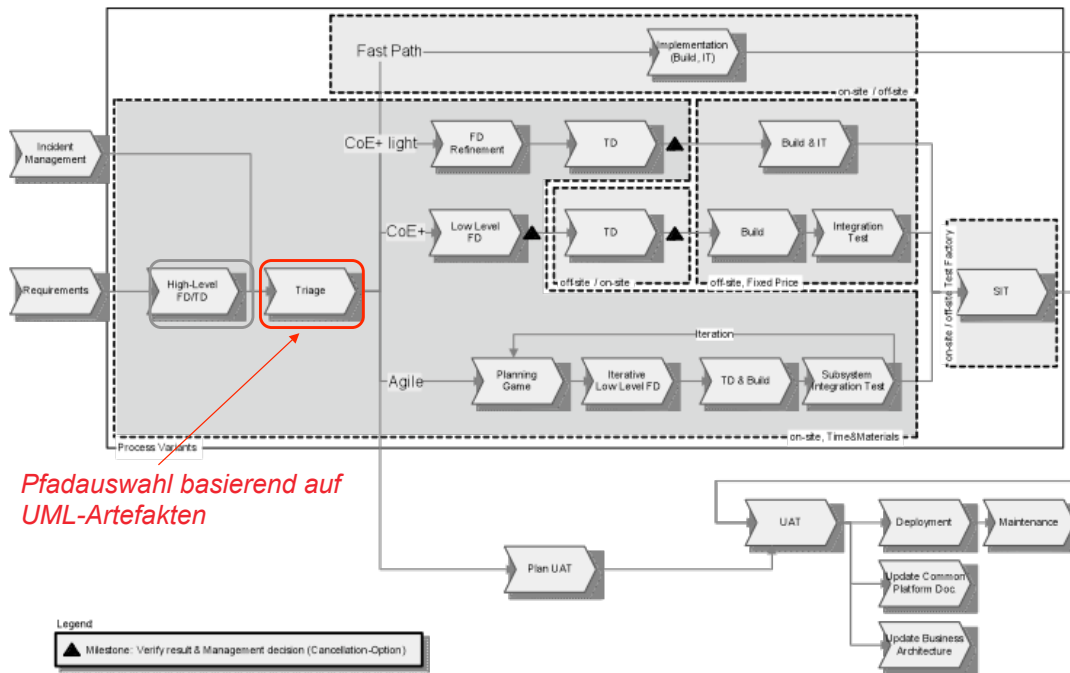
UNIVERSITÄT LEIPZIG

 LPZ  E-BUSINESS  
 applied telematics


it factum

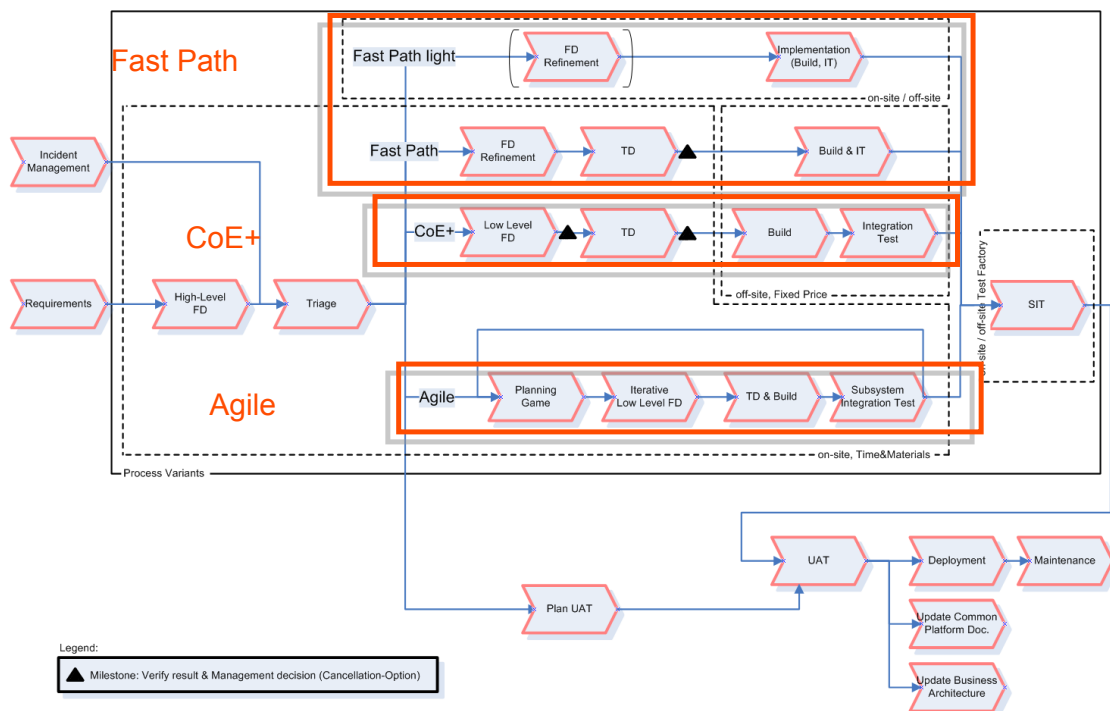
8

# Softwareprozess-Modell-Varianten

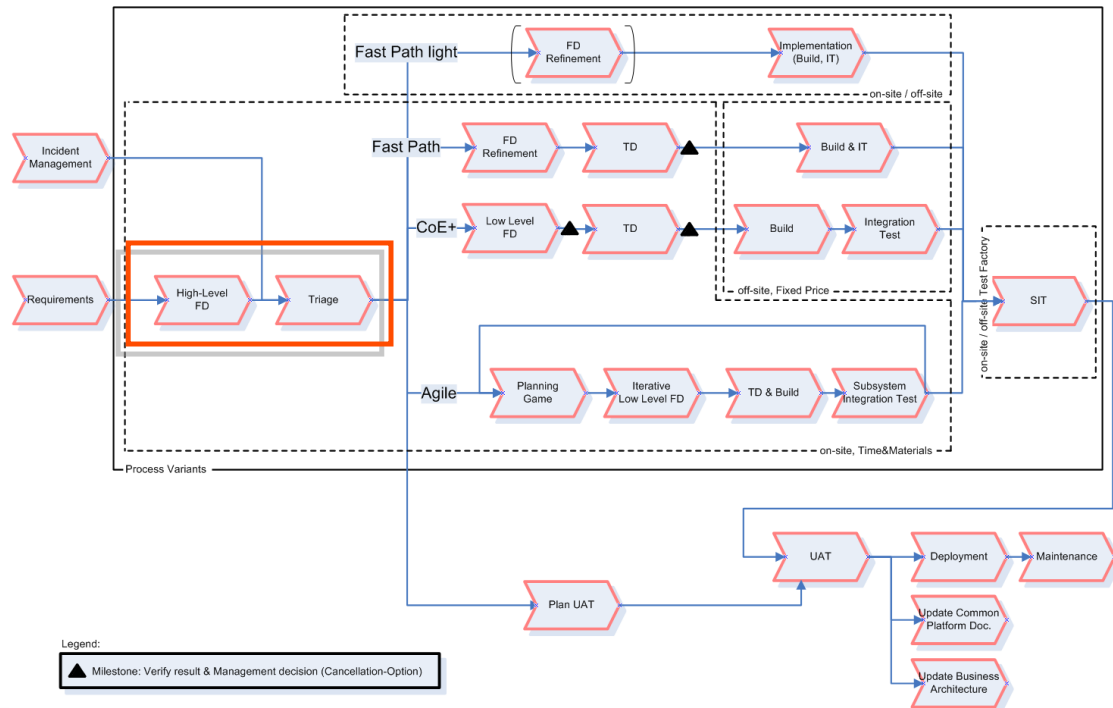


*Pfadauswahl basierend auf UML-Artefakten*

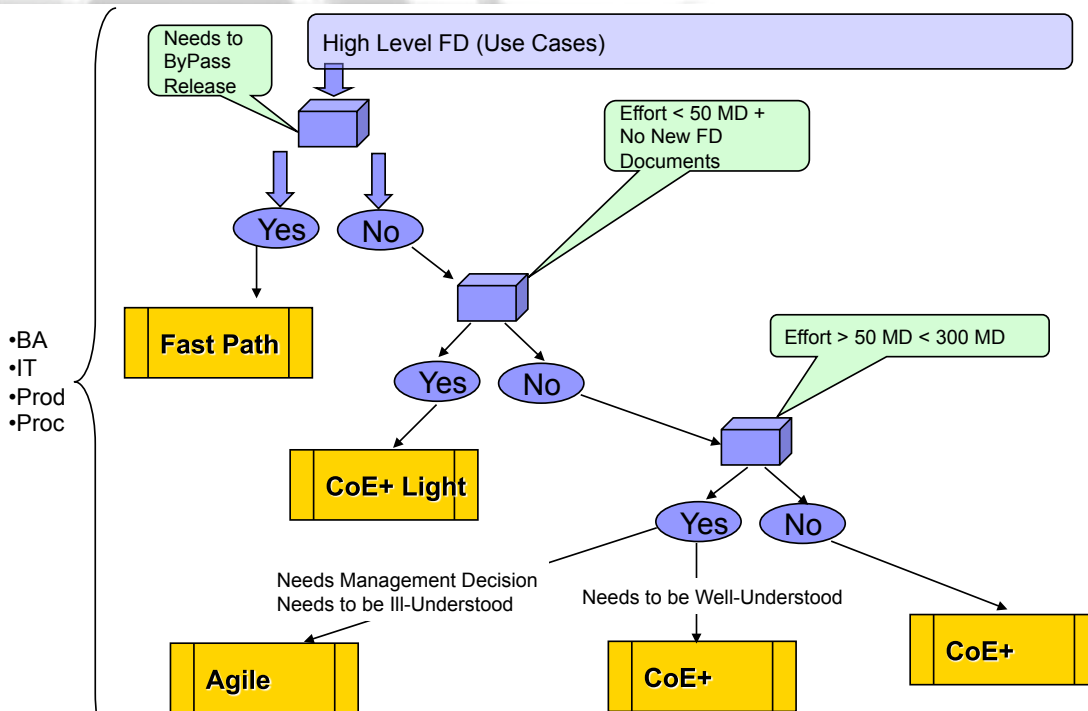
# Prozessvarianten



# High level Functional Design und Triage



# Triage



- BA
- IT
- Prod
- Proc



# No-Frills Prinzipien

<b>Problem Understanding</b>	<ul style="list-style-type: none"> <li>Understand the complete problem in breadth</li> <li>Identify the key document</li> <li>Design for change</li> <li>Choose the right abstraction level</li> <li>Separate requirements and design</li> </ul>	<b>Value Orientation</b>	<ul style="list-style-type: none"> <li>Test / validate and verify the substantial</li> <li>Challenge leaner solutions</li> <li>Productivity = Value / Effort</li> <li>Check for sourcing options</li> </ul>
<b>Technology</b>	<ul style="list-style-type: none"> <li>Integration engineering dominates : outside-in design</li> <li>Avoid technologically determined debates</li> <li>Consider deployment early</li> <li>Adhere to standards</li> </ul>	<b>Team &amp; Quality</b>	<ul style="list-style-type: none"> <li>Professional and focussed communication : who might be concerned</li> <li>Ensure : everybody involved is domain-knowledgeable</li> <li>Measure no-frills properties</li> </ul>

# Hauptaktivitäten und Prinzipien

	Entwicklung einer Vision	Sammeln von Anforderungen	No-Frills-Entwurf	Prototyping	Testen
<b>Problemverständnis</b>	Verstehe das ganze Problem in voller Breite	X	X	X	
	Identifiziere das Schlüsseldokument	X	X	X	
	Design for Change		X	X	X
	Wähle das richtige Abstraktionsniveau	X	X	X	
	Trenne Anforderungen und Design	X	X	X	
<b>Wertorientierung</b>	Teste, validiere und überprüfe das Wesentliche		X	X	X
	Fordere schlankere Lösungen	X	X	X	
	Produktivität = Nutzen / Aufwand	X	X	X	X
<b>Technik</b>	Design von außen nach innen		X	X	X
	Prüfe Sourcing-Möglichkeiten		X	X	
	Vermeide technologische Debatten		X	X	
<b>Team und Qualität</b>	Professionelle und fokussierte Kommunikation	X	X	X	X
	Wähle die passenden Sprachen und Werkzeuge	X	X	X	X
	Überprüfe auf No-Frills-Prinzipien	X	X	X	X

## Artefakte und Aktivitäten

UNIVERSITÄT LEIPZIG

**LPZ**  **E-BUSINESS**  
*applied telematics*

Lösungsskizze	Schritt 1	Schritt 2	Schritt 3	Schritt 4	Schritt 5	Schritt 6	Schritt 7				
	Geschäftsprozess aufnehmen	Grundlegende Begriffe erklären	Systemkontext aufnehmen	Use-Case Liste entwickeln	Nichtfunktionale Anforderungen skizzieren	Wichtige Use-Case ausformulieren	Zusätzl. funkt. Anforderungen spezifizieren	Dialoge skizzieren	Fachliches Klassenmodell entwerfen	Architekturüberblick erstellen	Einführungsstrategie beschreiben
Anwenderschnittstelle						Gate 1					
Architekturüberblick		Gate 1								Gate 2	
Fachliches Glossar	Gate 1										
Fachliches Klassenmodell						Gate 1				Gate 2	
Nichtfunktionale Anforderungen				Gate 1						Gate 2	
Systemkontext		Gate 1								Gate 2	
Testspezifikation					Gate 1						
Use-Case Modell			Gate 1							Gate 2	
Zusätzliche funktionale Anforderungen						Gate 1				Gate 2	

it factum

15

## Erfahrungen

UNIVERSITÄT LEIPZIG

**LPZ**  **E-BUSINESS**  
*applied telematics*

- No-Frills Software Engineering® bei verschiedenen deutschen Versicherungsunternehmen eingeführt
- Typischer Einführungsprozess
  1. Workshops zur Analyse / Verständnis des Status Quo
    - Hauptsächliche Schwächen der bisherigen Softwareentwicklung
    - Bekannte und unbekannte Anforderungen und Erwartungen
  2. Tailoring des neuen Softwareprozesses nach NF-Prinzipien
  3. Coaching-Phase, um NF-Prinzipien zu verankern
- Beobachtete Effekte
  - Prozesse werden als schlanker und zielorientierter erfahren
  - Commitment der Mitarbeiter, Akzeptanz der Prozesse
  - Verbesserte Artefaktqualität / Konsistenz
  - Verstärkte Wertorientierung / Ausrichtung an Business-Nutzen

it factum

16

it factum

UNIVERSITÄT LEIPZIG

LPZ  **E-BUSINESS**  
*applied telematics*

**Vielen Dank für Ihre Aufmerksamkeit!**

Ihre Fragen?

## 12. Architekturmanagement und der Architekturstandard SAGA

### Sessionüberblick

---

12.1. Automatisierte Architektur-Audits als Mittel zur Vermeidung von Architekturerosion	631
12.2. SAGA-Konformität: Verbindlichkeit von Standards . . . . .	655
12.3. SAGA-konforme Webapplikationen – Herausforderung und Perspektiven . . . . .	669

---



## **Automatisierte Architektur-Audits als Mittel zur Vermeidung von Architekturerosion**

Jan-Peter Baas

BearingPoint GmbH, Speicherstraße 1  
60327 Frankfurt am Main  
jan-peter.baas@bearingpointconsulting.com

**Abstract:** Um Investitionen im IT-Umfeld nachhaltig zu sichern, ist eine hohe Qualität bei der Durchführung von IT-Projekten unabdingbar. Insbesondere die Qualität der Softwarearchitektur bestimmt die langfristigen Kosten für Wartung und Weiterentwicklung von Software. Wird die Architekturqualität nicht regelmäßig überwacht, erodiert die Softwarearchitektur einer Anwendung sehr schnell. Damit verkürzt sich die Lebenszeit der Anwendung deutlich, d.h. es wird viel schneller ein Zeitpunkt erreicht, an dem Erweiterungen und Fehlerkorrekturen nicht mehr wirtschaftlich durchführbar sind und die Anwendung daher komplett ersetzt werden muss. Der Vortrag stellt Vorgehen und Mechanismen vor, mit denen die Architekturerosion deutlich abgeschwächt und damit die Wirtschaftlichkeit einer Anwendung erhöht werden kann.

### **1 Motivation**

Die Entwicklung von IT-Anwendungen steht heute unter einem hohen Kostendruck. Häufig fehlt das Bewusstsein oder das Wissen über notwendige Qualitätssicherungsmaßnahmen so dass diese nicht oder nicht in ausreichendem Maße durchgeführt werden.

Besonders problematisch ist die Situation bei der Qualitätssicherung der Architekturumsetzung. Während für die Durchführung von Tests unterschiedlichste Tools zur Unterstützung eingesetzt werden (z.B. für die Durchführung automatisierter Regressionstests), erfolgt die Prüfung der Umsetzung der Architektur einer Anwendung in der Regel „von Hand“ durch einen Softwarearchitekten. Die Prüfung kann daher maximal stichprobenartig durchgeführt werden. Dies führt häufig dazu, dass ein gutes Architekturkonzept nicht wie geplant umgesetzt wird sondern z.B. aufgrund von Zeit- und Termindruck oder aufgrund eines fehlenden Qualitätsbewusstseins im Entwicklungsteam nicht adäquat umgesetzt wird. Wartungs- und Pflegekosten steigen dadurch deutlich, die Lebenszeit der Anwendung wird verkürzt und die Wirtschaftlichkeit langfristig deutlich gesenkt. Um diesen Effekten entgegen zu wirken werden aufwendige und teure Qualitätssicherungsmaßnahmen z.B. in Form von Reviews eingeführt, mit denen sich die Architekturumsetzung aber trotzdem häufig nur stichprobenartig durchführen lässt. Eine vollständige und umfassende Prüfung der Architektur mit direktem Feedback an die Entwickler wird in der Regel nicht durchgeführt. Es entstehen daher weiterhin Anwendungen, die sich langfristig nicht weiterentwickeln lassen und bei denen die Wartungskosten explodieren.

## 2 Ansatz

Der Vortrag stellt am Beispiel der Prüfung der Softwarearchitektur einen Ansatz vor, mit dem ein IT-Verantwortlicher die Balance zwischen Kostendruck und Qualität herstellen kann. Es wird erläutert, wie frühzeitig, schnell und kostengünstig überprüft werden kann, ob Source Code und Architektur in hoher Qualität umgesetzt wurden.

Meist erfolgt eine Qualitätssicherung von IT-Projekten erst gegen Ende eines Projektes oder bei wichtigen Meilensteinen. Ein Gegensteuern bei grundsätzlichen Mängeln ist dann häufig nicht mehr möglich oder mit hohen Kosten verbunden. Aus diesem Grund ist es notwendig, die Architekturumsetzung regelmäßig und automatisiert zu prüfen. Auf diese Weise ist es sogar möglich, eine kontinuierliche Qualitätsprüfung über den gesamten Entwicklungsprozess zu etablieren, ein direktes Feedback für die Entwickler zu etablieren und sicherzustellen, dass alle Architekturverletzungen frühzeitig erkannt werden.

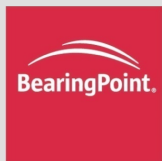
Neben einer generellen Vorgehensweise stellt der Vortrag unterschiedliche Architekturmetriken, Werkzeuge und Vorgehensweisen vor, mit denen sich das Ziel automatisierter Architektur-Audits oder eines kontinuierlichen Architekturcontrollings umsetzen lassen. Dabei werden anhand praktischer Beispiele die Möglichkeiten und Grenzen eines solchen Vorgehens vorgestellt.

Um die volle Wirkung der Architekturprüfung zu erreichen, muss diese in den Gesamtkontext der Qualitätssicherung eingebettet werden. Beispielweise müssen die Entwicklungsprozesse auf derartige Prüfungen ausgerichtet sein. Auch dieser Aspekt wird daher erläutert.

## 2 Bewertung

Eine zukunftsfähige Architektur ist ein wesentliches Merkmal um die Investitionssicherheit und Wirtschaftlichkeit einer Anwendung langfristig sicherzustellen. Um eine schnelle Erosion der Architektur zu vermeiden, ist es unumgänglich nicht nur die Architekturkonzepte zu prüfen sondern auch sicherzustellen, dass das geplante Architekturkonzept auch exakt umgesetzt wird. Verletzungen der Architektur bei der Umsetzung wirken sich in der Regel nicht unmittelbar aus. Bei späteren Änderungen führen sie jedoch zu Fehlern und Seiteneffekten, die letztendlich dazu führen, dass eine Weiterentwicklung der Anwendung nicht mehr wirtschaftlich möglich ist.

Eine automatisierte Architekturprüfung ist ein gutes Mittel, um mit geringem Aufwand eine deutliche Verbesserung der Architekturumsetzung zu erreichen. Entwickler erhalten (genau wie bei anderen Programmierfehlern auch) eine Fehlermeldung, wenn sie Architekturverletzungen begehen. Architekturfehler werden auf diese Weise verhindert, bevor sie begangen werden. Die hohen Kosten für manuell durchgeführte Architekturreviews entfallen.



Management  
& Technology  
Consultants

## Automatisierte Architektur-Audits als Mittel zur Vermeidung von Architekturerosion

Jan-Peter Baas  
Frankfurt, 04.05.2010

### Agenda



- Einführung
- Begriffsdefinitionen im Umfeld Architektur
- Architekturerosion: Ursache und Wirkungen
- Maßnahmen zur Vermeidung von Architekturerosion
- Metrikbasierte Architekturmessung
- Grenzen der Architekturmessung
- Permanentes Architekturcontrolling
- Beispiele für Metriken
- Fazit

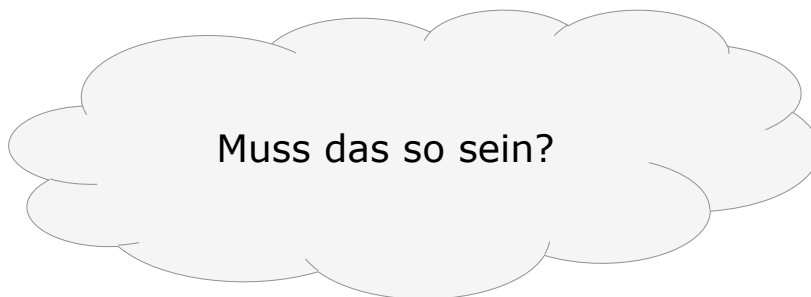
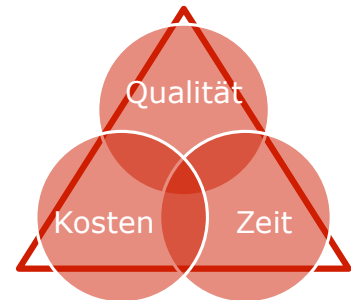


## Das magische Dreieck



### Häufig anzutreffende Aussagen:

- Eine Erhöhung der Qualität führt zu einer Erhöhung der Kosten
- Eine Verkürzung der Zeit führt zu einer Erhöhung der Kosten
- Kosteneinsparungen gehen zu Lasten der Qualität



© 2010 BearingPoint GmbH

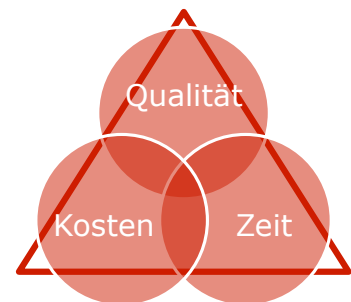
3

## Häufig anzutreffende Aussagen zum magischen Dreieck sind nicht immer korrekt



### Häufig anzutreffende Aussagen:

- Eine Erhöhung der Qualität führt zu einer Erhöhung der Kosten
- Eine Verkürzung der Zeit führt zu einer Erhöhung der Kosten
- Kosteneinsparungen gehen zu Lasten der Qualität



- Aktivitäten zur Qualitätssicherung kosten Zeit und Geld. Die Beseitigung von Qualitätsmängeln kostet ggf. weit mehr.
- Zeitdruck und eine Erhöhung der eingesetzten Ressourcen um Zeit zu sparen, kann zu höheren Kosten und Qualitätsmängeln führen. Probleme aufgrund langer Entwicklungszyklen, an deren Ende man feststellt, dass man in die falsche Richtung entwickelt hat, sind ggf. nicht mehr zu korrigierbar.
- Wenn QS-Maßnahmen ersatzlos gestrichen werden, geht das zu Lasten der Qualität. Optimiert man sie, können Kosteneinsparungen bei gleichzeitiger Erhöhung der Qualität erreicht werden.

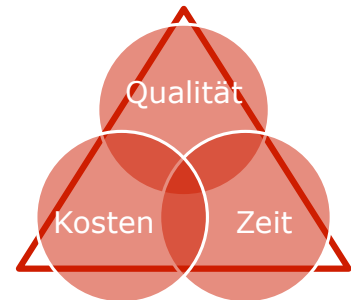
© 2010 BearingPoint GmbH

4

Eine Erhöhung der Qualität kann gleichzeitig zu niedrigeren Kosten und kürzeren Projektlaufzeiten führen



Das Ziel
<ul style="list-style-type: none"> <li>■ Qualität erhöhen</li> <li>■ Kosten senken</li> <li>■ Zeiten verkürzen</li> </ul>



- ➔ QS-Maßnahmen müssen die Qualität transparent und nachweisbar erhöhen.
- ➔ QS-Maßnahmen müssen schnell und kostengünstig durchführbar sein.
- ➔ QS-Maßnahmen müssen zu einer Verkürzung der Projektlaufzeit beitragen.

© 2010 BearingPoint GmbH

5

QS-Maßnahmen müssen frühzeitig während des gesamten Softwareentwicklungsprozess erfolgen



Je länger die Entwicklungsphase und je umfangreicher die Inhalte einer Entwicklungsphase, desto schwieriger lassen sich im anschließenden Test erkannte Probleme korrigieren

- ➔ QS-Maßnahmen müssen so gestaltet sein, dass Probleme möglichst beim Entstehen oder zumindest frühzeitig erkannt werden.

© 2010 BearingPoint GmbH

6

## Agenda



- Einführung
- **Begriffsdefinitionen im Umfeld Architektur**
- Architekturerosion: Ursache und Wirkungen
- Maßnahmen zur Vermeidung von Architekturerosion
- Metrikbasierte Architekturmessung
- Grenzen der Architekturmessung
- Permanentes Architekturcontrolling
- Beispiele für Metriken
- Fazit

## Begriffsdefinitionen



### Softwarearchitektur

„Eine **Softwarearchitektur** ... beschreibt die grundlegenden Komponenten und deren Zusammenspiel innerhalb eines Softwaresystems.“

Quelle: <http://de.wikipedia.org/wiki/Softwarearchitektur>, 6.4.2010, 11:45 Uhr

### Architekturmuster

„Im Bereich der Softwareentwicklung sind **Architekturmuster** (auch: Architekturstil engl. architectural style) in den Arten von Mustern auf oberster Ebene einzuordnen. Im Gegensatz zu Idiomen oder Entwurfsmustern bestimmen sie nicht ein konkretes (meist kleines oder lokales) Teilproblem, sondern die grundlegende Organisation und Interaktion zwischen den Komponenten einer Anwendung.“

Quelle: <http://de.wikipedia.org/wiki/Architekturmuster>, 6.4.2010, 11:45 Uhr

### Softwaremetrik

„Eine **Softwaremetrik**, oder kurz Metrik, ist eine (meist mathematische) Funktion, die eine Eigenschaft von Software in einen Zahlenwert, auch Maßzahl genannt, abbildet. Hierdurch werden formale Vergleichs- und Bewertungsmöglichkeiten geschaffen.“

Quelle: <http://de.wikipedia.org/wiki/Softwaremetrik>, 6.4.2010, 11:45 Uhr

## Begriffsdefinition: Architekturerosion



„Als **Erosion** (v. lat.: erodere = abnagen) bezeichnet man die Zerstörung der Formen der Erdoberfläche durch linienhafte oder flächenhafte Abtragung.

Die linienhafte Vertiefung der Erdoberfläche geschieht durch Fließgewässer oder Gletscher. Wind, Meeresbrandung und auch Niederschläge erzeugen flächenhafte Erosionserscheinungen.

... Quelle: [http://de.wikipedia.org/wiki/Erosion\\_%28Geologie%29](http://de.wikipedia.org/wiki/Erosion_%28Geologie%29), 6.4.2010, 11:45 Uhr

### Architekturerosion (Architectural Decay):

Die IST-Architektur (Sourcecode) stimmt mit der Zeit immer weniger mit der geplanten SOLL-Architektur (Architekturdokumentation) überein.

Die Qualität der SOLL-Architektur nimmt ab, wenn sie nicht ausreichend gepflegt wird.

## Agenda



- Einführung
- Begriffsdefinitionen im Umfeld Architektur
- Architekturerosion: Ursache und Wirkungen
- Maßnahmen zur Vermeidung von Architekturerosion
- Metrikbasierte Architekturmessung
- Grenzen der Architekturmessung
- Permanentes Architekturcontrolling
- Beispiele für Metriken
- Fazit

## Es gibt eine Vielzahl von Gründen für Architekturerosion



- Die Umsetzung der geplanten Architektur wird nicht konsequent überwacht
- Die Architekturdokumentation fehlt oder wird nicht aktuell gehalten
- Das Projektteam verfügt nicht über das notwendige Verständnis der Architektur
- Termindruck führt zu schnellen Workarounds
- Workarounds werden nicht beseitigt
- Das Projektteam verändert sich, Know How geht verloren, die Entwickler verstehen die Architektur nicht mehr vollständig
- Mit fortschreitender Weiterentwicklung nimmt die Anzahl der unbemerkten Architekturverletzungen zu
- Aufwände für Wartung, Pflege und Weiterentwicklung nehmen zu
- ...

## Ziele einer „guten“ Softwarearchitektur

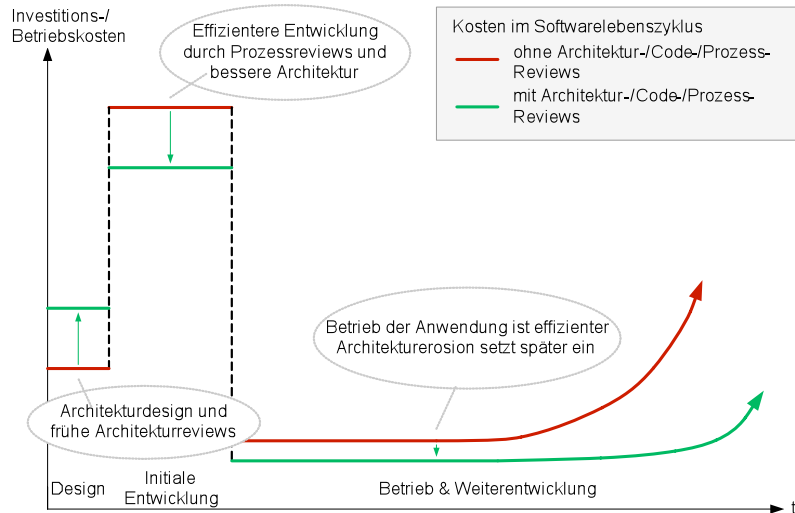


- Verstehbarkeit
- Erweiterbarkeit
- Wartbarkeit
- Testbarkeit
- Flexibilität
- Skalierbarkeit
- Sicherheit
- Performance
- Analysierbarkeit von Fehlern
- Zuverlässigkeit
- Verfügbarkeit



Architekturerosion gefährdet die genannten Ziele und reduziert damit die Investitionssicherheit nachhaltig

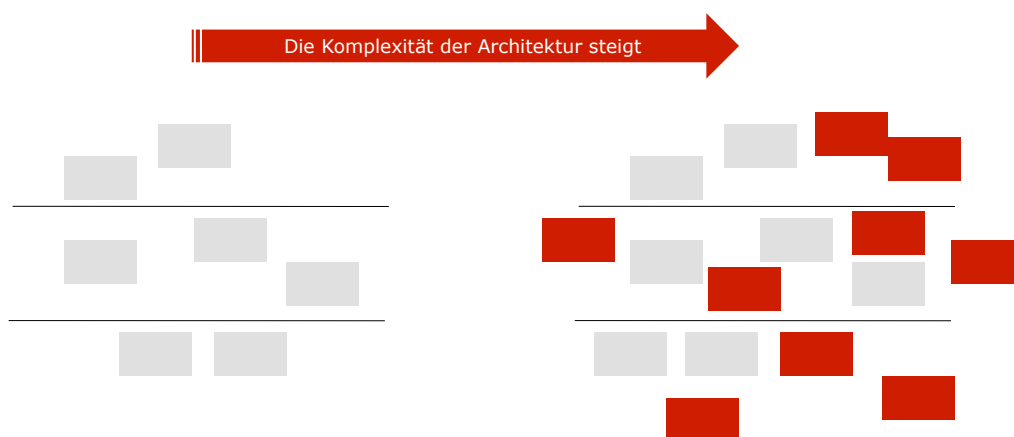
## Architekturerosion bedeutet, dass der Anwendungslifecycle verkürzt wird



- Wartungskosten steigen
- Budgets und Zeiten können nicht gehalten werden
- Es wird ein Redesign oder gar eine Neuentwicklung notwendig

➔ Die Vermeidung von Architekturerosion erhöht die mögliche Nutzungsdauer und verringert die Gesamtkosten

## Erweiterungen verändern die Softwarearchitektur, die Komplexität nimmt in der Regel zu

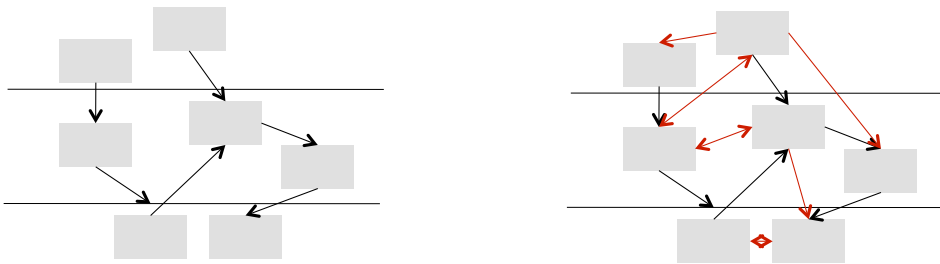


- ☑ Normal, wenn die Komplexität im Verhältnis zur zusätzlichen Funktionalität steigt
- ☒ Problematisch, wenn die Komplexitätszunahme nicht durch eine saubere Architekturdefinition gesteuert wird

Jede Änderung am Sourcecode birgt das Risiko neuer Abhängigkeiten und fördert damit ggf. die Architekturerosion

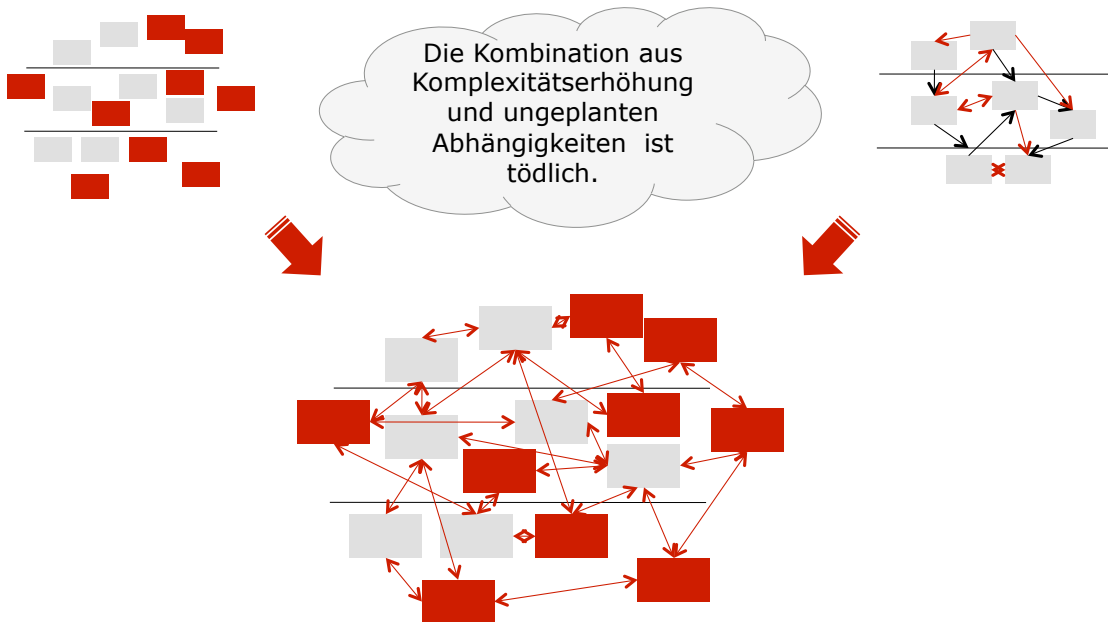


Die Abhängigkeiten und Zyklen steigen ebenfalls



- ☑ Akzeptabel, wenn aufgrund neuer Funktionalitäten eine geringe Zunahme der Abhängigkeiten in Einklang mit der geplanten Architektur erfolgt
- ☒ Problematisch, wenn ungeplante Abhängigkeiten oder gar Zyklen entstehen

Komplexe Systeme mit Abhängigkeiten und Zyklen sind nicht mehr beherrschbar



## Die Auswirkungen von Architekturerosion sind vielfältig



- Aufwände für Erweiterungen steigen stark an
- Treten Fehler auf, ist die Fehlerursache nur schwer zu ermitteln
- Es kommt immer häufiger zu überraschenden Seiteneffekten
- Die Anzahl Fehler pro Implementierungseinheit steigen
- Eine Weiterentwicklung ist unter Umständen nicht mehr mit vertretbarem Aufwand möglich
- Haben Abhängigkeiten zu stark zugenommen, lässt sich dieses Problem nicht mehr rückgängig machen.
- Mit der Zeit schleichen sich unbemerkte Architekturverletzungen in jeden Sourcecode ein. Sie müssen gezielt gesucht und behoben werden
- Durch eine „händische“ Analyse ist es nicht möglich, alle Architekturverletzungen zu erkennen

## Agenda


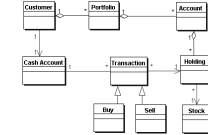


- Einführung
- Begriffsdefinitionen im Umfeld Architektur
- Architekturerosion: Ursache und Wirkungen
- Maßnahmen zur Vermeidung von Architekturerosion
- Metrikbasierte Architekturmessung
- Grenzen der Architekturmessung
- Permanentes Architekturcontrolling
- Beispiele für Metriken
- Fazit



Die Wahrheit liegt im Sourcecode

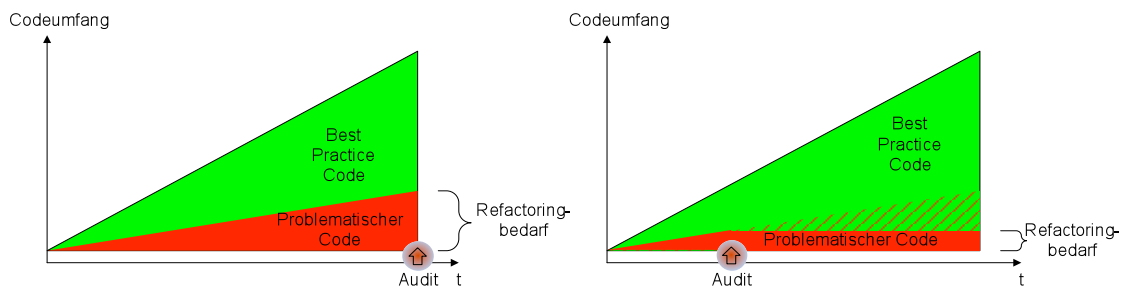


Es gibt Methoden zum Architekturmanagement	Es gibt Methoden zur Architekturdokumentation
 <p>Beispiel Architekturmanagement nach TOGAF</p>	 <p>Beispiel Unified Modeling Language (UML)</p>

Was nützt die beste Architektur, wenn sie im Sourcecode nicht eingehalten wird?

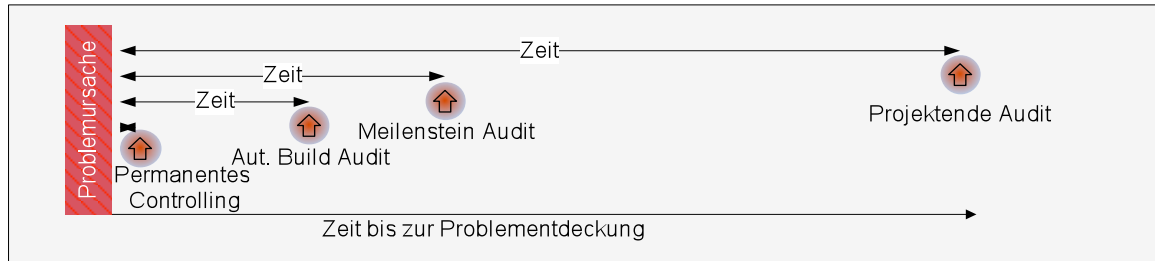
- ➔ Die Dokumentation der Softwarearchitektur muss aktuell und vollständig sein
- ➔ Die dokumentierte Softwarearchitektur muss im Sourcecode auch wie geplant umgesetzt sein

Je früher ein Problem erkannt wird, desto einfacher ist seine Behebung



- ➔ Probleme müssen möglichst frühzeitig erkannt und korrigiert werden

Nur mit häufigen, kurzfristig durchgeführten Audits lässt sich eine Architekturerosion wirksam bekämpfen



- Werden Probleme erst am Projektende erkannt, werden Korrekturen aufwändig und teuer
- Auch bei Meilensteinaudits werden Probleme (je nach Häufigkeit der Meilensteine) erst relativ spät bemerkt
- Automatisierte Build Audits bieten die Möglichkeit mit geringem Aufwand bei jedem Build Architekturprüfungen durchzuführen
- Ein permanentes Architekturcontrolling kann erreicht werden, wenn Architekturprüfungen direkt in die Entwicklungsumgebung integriert werden

Eine vollständige Prüfung der Architekturumsetzung ist nicht mit manuellen Audits möglich



Klassisches Architekturaudit („von Hand“)	Automatisiertes Architekturaudit
<ul style="list-style-type: none"> <li>▪ Hoher Aufwand</li> <li>▪ Nur stichprobenhaft möglich</li> <li>▪ Intuition und Erfahrung des Reviewers ausschlaggebend</li> <li>▪ Vergleichbarkeit der Ergebnisse nur schwer möglich</li> </ul>	<ul style="list-style-type: none"> <li>▪ Geringer Aufwand</li> <li>▪ Häufig durchführbar</li> <li>▪ Vollständige Prüfung möglich</li> <li>▪ Erfahrung des Reviewers notwendig, um die Messergebnisse zu interpretieren</li> <li>▪ Tendenzen über Qualitätserhöhung oder -Verringerung frühzeitig erkennbar</li> </ul>

➔ Grundlage eines Architekturaudits sollte eine metrikbasierte, automatisierte Architekturmessung sein

Auch für die Interpretation der Ergebnisse eines metrikbasierten Audits ist die Erfahrung eines Architekten notwendig



- Es gibt keine festen Standards, Normen oder Muster (vergleichbar mit Regeln zur Sourcecodeerstellung), wie eine gute Softwarearchitektur auszusehen hat.
- Design Pattern tragen zur Qualität der Softwarearchitektur bei, betrachten aber häufig nur einzelne Teilprobleme der Gesamtarchitektur
- Eine Qualitätsbeurteilung erfolgt daher häufig über subjektive Erfahrungswerte.



Die Qualität einer Architektur lässt sich nicht direkt messen

- Metriken untersuchen den Quelltext einer Anwendung mit mathematischen Formeln
- Eine Metrik liefert weder „gut“ noch „schlecht“ als Ergebnis. Vielmehr liefert sie Kennzahlen, die interpretiert werden müssen.
- Die kontinuierliche Anwendung von Metriken kann dabei helfen, Entwicklungen und Trends bei der Umsetzung von Architekturen zu beobachten und ggf. rechtzeitig Maßnahmen zu ergreifen.



Metriken können helfen eine Architektur messbar zu machen

© 2010 BearingPoint GmbH

23

## Agenda

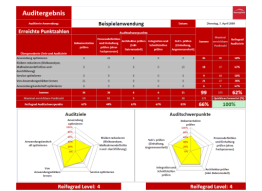


- Einführung
- Begriffsdefinitionen im Umfeld Architektur
- Architekturerosion: Ursache und Wirkungen
- Maßnahmen zur Vermeidung von Architekturerosion
- **Metrikbasierte Architekturmessung**
- Grenzen der Architekturmessung
- Permanentes Architekturcontrolling
- Beispiele für Metriken
- Fazit

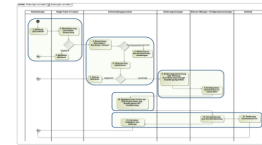
© 2010 BearingPoint GmbH

24

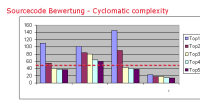
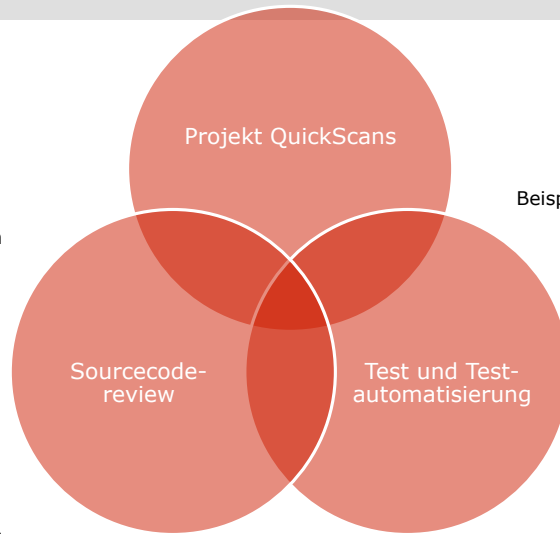
Die Architekturmessung ist keine Einzelmaßnahme sondern muss mit anderen Maßnahmen kombiniert werden



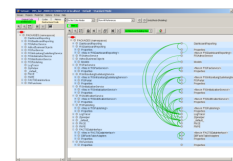
Beispiel Projekt Quick Scan



Beispiel Softwareentwicklungsprozesse



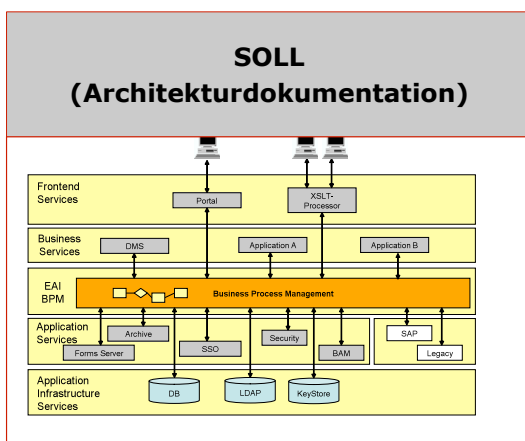
Beispiel Sourcecodemessung



Beispiel Architekturmessung

➔ Die Architekturmessung muss in das Projektvorgehen eingebettet sein. Ergebnisse der Messung müssen nach definierten Prozessen bearbeitet werden.

Ziel der Architekturmessung ist die Aufdeckung von Abweichungen zwischen dokumentierter SOLL- und implementierter IST-Architektur



**IST (Sourcecode)**

```
public class HelloWorld {
    public static void printHW(XScriptContext xSc) {
        // getting the text document object
        XTextDocument txtDocument =
            (XTextDocument) UnoRuntime.queryInterface(
                XTextDocument.class, xSc.getDocument());
        XText xText = txtDocument.getText();
        XTextRange xTextRange = xText.getEnd();
        xTextRange.setString( "Hello World (in Java)" );
    } // printHW
}
```



Stimmen SOLL und IST überein?

Für die Architekturmessung von Schichtenarchitekturen mittels Architekturmodell gibt es gute Werkzeuge



- Für eine toolgestützte Messung ist eine Architekturdokumentation nicht ausreichend
- Bevor eine Messung durchgeführt werden kann, muss die Architekturdokumentation in ein toolgestütztes Architekturmodell überführt werden
- Dabei kann gleichzeitig die Qualität der Architekturdokumentation überprüft werden

➔ Die systematische Messung der Qualität der Architektur anhand von Metriken erfordert ein Architekturmodell.

Zur Beschreibung der Architektur wird einfaches, inkrementelles und werkzeuggestütztes Verfahren eingesetzt:

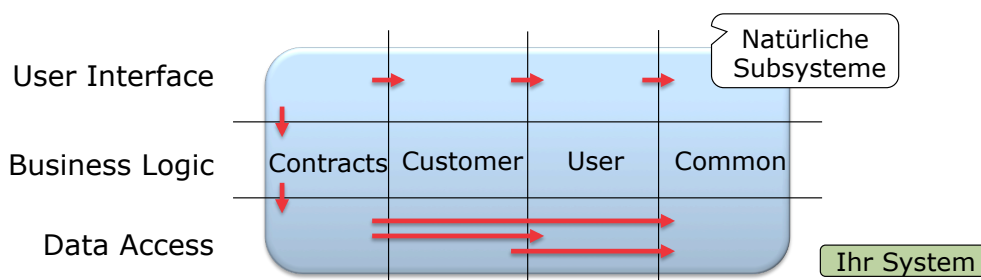
- **Schritt 1:** Ein existierendes System wird horizontal in Schichten aufgeteilt.
- **Schritt 2:** Im Anschluss erfolgt die vertikale Aufteilung in fachliche Schnitte.
- **Schritt 3:** Zuletzt werden die erlaubten Benutzungsbeziehungen definiert.

Das Architekturmodell bildet eine horizontale und vertikale Schichtung der Architektur inklusive Benutzungsbeziehungen ab

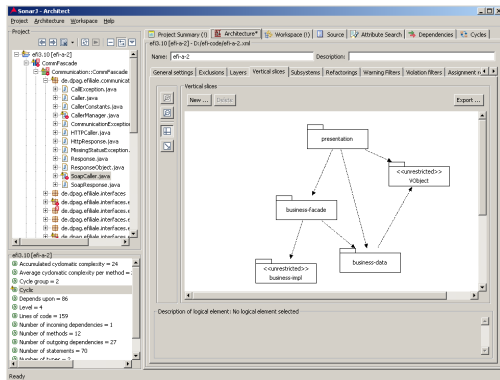


### Elemente der logischen Architektur

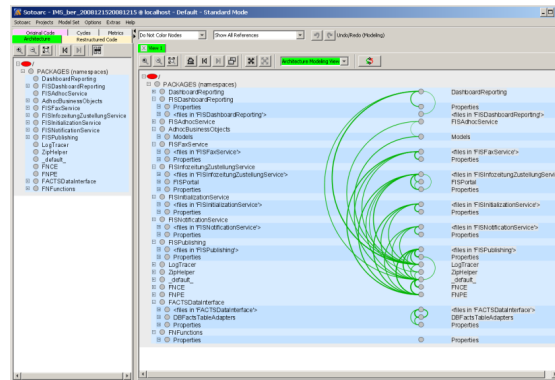
- Physikalische Ebene: Umfasst eine Menge von Komponenten (= Compilation Units), zum Beispiel Namespaces.
- Logische Ebene: Umfasst Subsysteme mit einer Menge von Packages sowie Schichten mit einer Menge von Subsystemen.
- Optional können fachliche Schnitte im Modell (z.B. nach Contracts) erfolgen.



Der Sourcecode muss dem Architekturmodell zugeordnet werden



Beispiel Architekturmodell



Beispiel Analyse der Benutzungsbeziehungen

## Agenda



- Einführung
- Begriffsdefinitionen im Umfeld Architektur
- Architekturerosion: Ursache und Wirkungen
- Maßnahmen zur Vermeidung von Architekturerosion
- Metrikbasierte Architekturmessung
- Grenzen der Architekturmessung
- Permanentes Architekturcontrolling
- Beispiele für Metriken
- Fazit

Metriken sind technologieunabhängig, die Ergebnisse müssen jedoch technologiespezifisch interpretiert werden



- Vergleichbare Metriken gelten für unterschiedliche Technologien (z.B. Java und .Net)
- Die Metriken müssen aber auf die eingesetzte Technologie und Architektur abgestimmt sein
- Kommt z.B. eine RulesEngine zum Einsatz, werden Benutzungsbeziehungen in die Konfiguration verlagert und damit ggf. nicht mehr erfasst (Messergebnisse zur Komplexität von Abhängigkeiten ändern sich)
- Kommen Standardprodukte und Frameworks zum Einsatz, muss entschieden werden, inwieweit diese mit gemessen werden sollen oder nicht

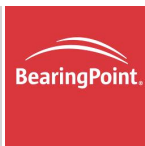


Werden unterschiedliche Technologien kombiniert, muss dies bei der Auswahl der Metriken und der Interpretation der Messergebnisse berücksichtigt werden

© 2010 BearingPoint GmbH

31

## Agenda



- Einführung
- Begriffsdefinitionen im Umfeld Architektur
- Architekturerosion: Ursache und Wirkungen
- Maßnahmen zur Vermeidung von Architekturerosion
- Metrikbasierte Architekturmessung
- Grenzen der Architekturmessung
- **Permanentes Architekturcontrolling**
- Beispiele für Metriken
- Fazit

© 2010 BearingPoint GmbH

32

## Von der Architekturmessung zum Architekturcontrolling ist es nur ein kleiner Schritt



- Automatisiertes Architekturcontrolling
  - Einführung automatisierter Builds (z.B. Nightly Build)
  - Durchführung der automatisierten Architekturprüfung im Rahmen des Nightly Builds
- Vermeidung von Architekturfehlern
  - Direkte Einbindung des Architekturmodells in die Softwareentwicklungsumgebung
  - Anzeige von Architekturfehlern analog zu anderen Fehlern (z.B. Syntaxfehlern) in der Entwicklungsumgebung
  - Sourcecode kann nicht eingchecked werden, solange er Architekturverletzungen enthält



Ein permanentes Architekturcontrolling erkennt bzw. vermeidet Probleme unmittelbar bei ihrer Entstehung

© 2010 BearingPoint GmbH

33

## Das automatisierte Architekturcontrolling bietet viele Vorteile



- Frühe Erkennung möglicher Qualitätsprobleme
- Stetige Qualitätssicherung und -Verbesserung der Architektur und Softwarequalität wird möglich
- Beschleunigung des Entwicklungsprozesses, da Probleme früher erkannt und behoben werden
- Sind entsprechende Werkzeuge und Tools einmal etabliert, ergeben sich kurze Rüstzeiten und eine einheitliche Durchführung von Softwareentwicklungsprojekten
- Ein aktives Change Management im Bereich Architektur wird unterstützt
- Workarounds etc. bleiben nicht unbemerkt
- Der Lebenszyklus der Software wird verlängert
- Kosten für Wartung und Weiterentwicklung werden reduziert
- Risiken in unterschiedliche Projektphasen z.B. beim Roll-Out werden reduziert
- Die Schätzgenauigkeit steigt (weniger Seiteneffekte usw.)

© 2010 BearingPoint GmbH

34



## Agenda



- Einführung
- Begriffsdefinitionen im Umfeld Architektur
- Architekturerosion: Ursache und Wirkungen
- Maßnahmen zur Vermeidung von Architekturerosion
- Metrikbasierte Architekturmessung
- Grenzen der Architekturmessung
- Permanentes Architekturcontrolling
- Beispiele für Metriken
- Fazit

## Beispiel 1: Cyclomatische Complexity – Gibt einen Hinweis auf zu komplexe Methoden



### **CCN (Cyclomatic Complexity Number)**

- Misst die Verschachtelung einer Methode durch Summierung der Verzweigungen (if, switch, throw, return) innerhalb einer Methode.
- Gibt direkt Rückmeldung über den Aufwand, der benötigt wird, um die Methode zu testen.
- Gut ist ein CCN < 10. Der Wert 25 sollte nicht überschritten werden.
- Lösung: komplexe Methoden in kleinere Methoden extrahieren.

## Beispiel 2: Average Component Dependency – Gibt einen Hinweis auf zu komplexe Abhängigkeiten zwischen Komponenten



### CCD (Cumulative Component Dependency)

- Summe aller Komponenten (= Compilation Units) von der ein Element (z.B. Package) direkt und indirekt abhängt, kumuliert über alle Komponenten des Elements.
- Ergibt auch die Anzahl von benötigten Komponenten, um das Element inkrementell zu testen.

### ACD (Average Component Dependency)

- Der durchschnittliche CCD der Komponenten eines Systems.
- Indikator für die Anzahl der zu testenden Komponenten bei der Umsetzung eines CR.
- Ein hoher ACD Wert weist auf Zyklen zwischen Elementen hin.

© 2010 BearingPoint GmbH

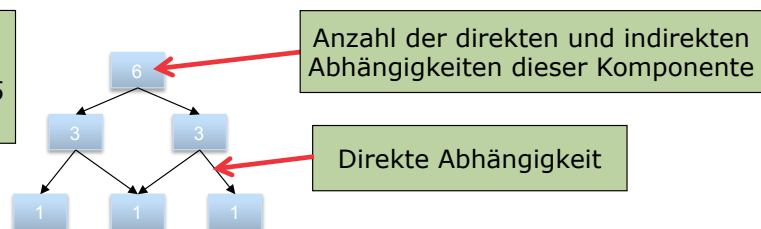
37

## Beispiel 2: Average Component Dependency – Insbesondere bei Zyklen steigt die ACD

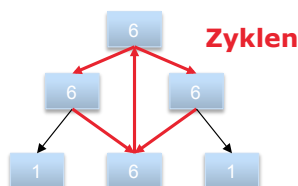


Beispiel 1:

$$ACD = \frac{6+3+3+1+1+1}{6} = 2,5$$



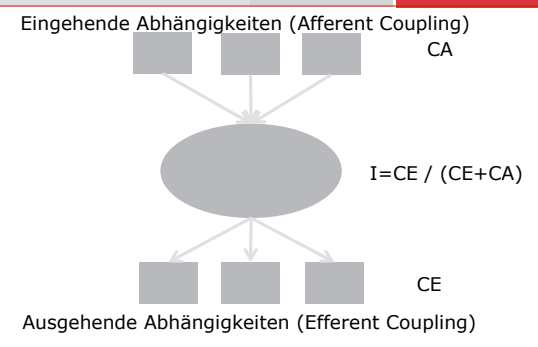
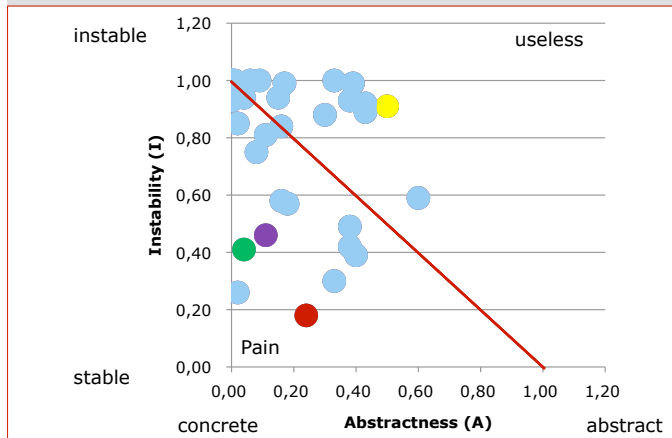
Beispiel 2: ACD = 4,33  
ACD wächst, wenn Zyklen zwischen den Komponenten existieren



© 2010 BearingPoint GmbH

38

Beispiel 2: Distance – Gibt einen Hinweis auf überflüssige Implementierungen und schwer zu beherrschende Komponenten



**Distance**

- Useless: Instabile Subsysteme, die nur von wenigen genutzt werden, benötigen kein aufwändiges Interface
- Pain: Stabile Subsysteme, die von vielen genutzt werden, benötigen abstrakte Interfaces, da Änderungen ansonsten Auswirkungen auf die zugreifenden Subsysteme haben

- Abstractness (A): Eine hohe Abstractness bedeutet viele abstrakte Interfaces oder Klassen
- Instability (I): Eine hohe Instabilität bedeutet, dass ein Subsystem viele andere Subsysteme nutzt. Stabil bedeutet, dass ein Subsystem von vielen anderen genutzt wird

Agenda



- Einführung
- Begriffsdefinitionen im Umfeld Architektur
- Architekturerosion: Ursache und Wirkungen
- Maßnahmen zur Vermeidung von Architekturerosion
- Metrikbasierte Architekturmessung
- Grenzen der Architekturmessung
- Permanentes Architekturcontrolling
- Beispiele für Metriken
- Fazit

## Durch den Einsatz automatisierter Architekturaudits lassen sich Budget- und Planungsrisiken bei gleichzeitiger Erhöhung der Qualität reduzieren

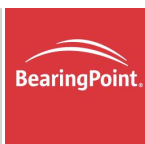


- Architekturerosion ist eine wesentliche Ursache für Budget- und Zeitüberschreitungen von Projekten
- Architekturerosion lässt sich nicht verhindern, aber verringern
- Mit Architekturmessungen und Architekturcontrolling lässt sich die Architekturerosion messbar machen
- Architekturmessungen können nicht manuell sondern müssen automatisiert durchgeführt werden
- Dazu ist ein toolgestütztes Architekturmodell notwendig
- Sowohl Architekturmessungen im Rahmen von Reviews als auch das Architekturcontrolling müssen sauber in die Softwareentwicklungsprozesse integriert werden
- Architekturmessungen sollten in andere QS-Maßnahmen (z.B. Messung der Sourcecode Qualität) eingebettet werden
- Der Messung müssen geeignete Maßnahmen zur Analyse der Schwachstellen und zur Optimierung folgen
- Architekturmessungen müssen regelmäßig, möglichst häufig oder im Rahmen eines Architekturcontrollings erfolgen

© 2010 BearingPoint GmbH

41

## Software-Architekturen und Projekterfolg (1)



Um den Einfluss der Qualität von Software-Architekturen auf Entwicklungsprojekte zu messen, hat das Software Technology Support Center (STSC) des Department of Defense (DoD) eine Studie<sup>1</sup> durchgeführt.

Im Rahmen der Studie wurde folgende Versuchssituation aufgebaut:

- Eine Anwendung mit 50 K Lines of Code (LOC) sollte um 3 K LOC erweitert werden.
- Ein Entwicklerteam startete auf der Basis einer unveränderten Architektur.
- Für ein zweites Entwicklerteam wurde die Architekturqualität verbessert, indem beispielsweise Zyklen entfernt wurden.

<sup>1</sup> B. M. Horowitz, The Importance of Architecture in DOD Software. Technical Report M91-35, The Mitre Corporation. July 1991.

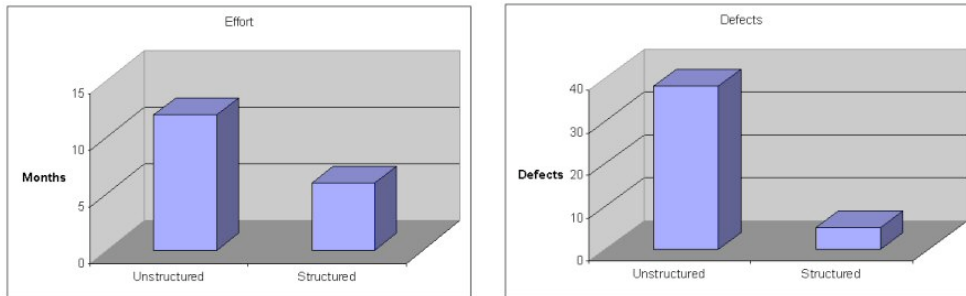
© 2010 BearingPoint GmbH

42

## Software-Architekturen und Projekterfolg (2)



Die Ergebnisse, die im Rahmen der Studie ermittelt wurden, sind deutlich:



- Das Team, das auf der unstrukturierten Architektur aufsetzen musste, hat mehr als doppelt so lange für die Entwicklung benötigt.
- Viel gravierender ist allerdings, dass bei der Arbeit mit der unstrukturierten Architektur die vierfache Menge an Fehlern erzeugt wurde.

© 2010 BearingPoint GmbH

43

Vielen Dank für Ihre Aufmerksamkeit



- Haben Sie noch Fragen?
- Kontaktieren Sie mich:



**Jan-Peter Baas**  
Technology Architect

**BearingPoint GmbH**  
Speicherstrasse 1  
60327 Frankfurt  
[www.bearingpoint.de](http://www.bearingpoint.de)

Tel: +49.69.13022.5286  
Mobile: +49.173.3004948  
Jan-Peter.Baas@bearingpointconsulting.com

© 2010 BearingPoint GmbH

44

## **SAGA-Konformität: Verbindlichkeit von Standards**

Rico Apitz

Team Consulting  
]init[ AG für Digitale Kommunikation  
Köpenicker Str. 9  
10997 Berlin  
rico.apitz@init.de

**Abstract:** Mit den Standards und Architekturen für E-Government-Anwendungen (SAGA) fördert der Bund die Interoperabilität, Plattformunabhängigkeit und Investitionssicherheit von Software-Systemen. Zur Erreichung der Ziele von SAGA ist es notwendig, einen Prozess zur Einhaltung und Überprüfung von Konformität zu etablieren. Dieser Prozess, die Schwächen in der Vergangenheit, die Stärken in der Gegenwart sowie die zukünftigen Perspektiven werden hier thematisiert.

### **1 Schwächen des Umgangs mit SAGA-Konformität in der Vergangenheit**

Mit SAGA [BMI08] verfolgt die Bundesverwaltung die Ziele Interoperabilität, Wiederverwendbarkeit, Offenheit, Skalierbarkeit sowie Reduktion von Kosten und Risiken von E-Government-Anwendungen. Um diese Ziele bestmöglich zu erreichen, muss SAGA eine hohe Akzeptanz erreichen und die Einhaltung muss gewährleistet sein. Herzstück von SAGA sind dabei insbesondere die Einstufungen technischer Standards als „Unter Beobachtung“, „Empfohlen“ und „Obligatorisch“.

Leider war in der Vergangenheit festzustellen, dass in Ausschreibungen der öffentlichen Verwaltung häufig „SAGA-Konformität“ pauschal gefordert wurde, was zu vertraglichen Unsicherheiten und zu teuren Angeboten führen kann.

### **2 Stärken des Umgangs mit SAGA-Konformität in der Gegenwart**

Um den vielfältigen Rahmenbedingungen bei der Entwicklung von E-Government-Anwendungen gerecht zu werden, sind die Klassifikationen in SAGA keine absoluten Vorgaben, sondern vielmehr Richtlinien, von denen in begründeten Ausnahmefällen z. B. aus Gründen der Wirtschaftlichkeit, abgewichen werden kann. Dies erhöht auf der einen Seite die Flexibilität der Auftragnehmer, auf der anderen Seite wird der Auftraggeber stärker in die Pflicht genommen, da er genauer spezifizieren muss, wie er die Vorgaben von SAGA umgesetzt sehen möchte. Um größtmögliche vertragliche Sicherheit zu gewährleisten, sollte SAGA-Konformität in den Ausschreibungsprozess integriert werden, wie in Abbildung 1 dargestellt.

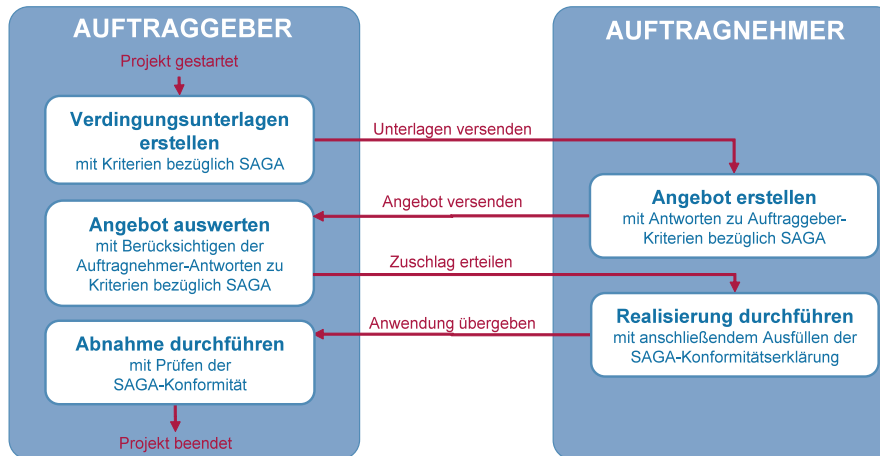


Abbildung 1: SAGA-Konformität im Ausschreibungsprozess

Auftraggeber und Auftragnehmer werden vom Bundesministerium des Innern in diesem Prozess unterstützt, beispielsweise durch Schulungen und Beispielunterlagen [BFIT08].

Die beschriebene Methode zur Prüfung und Bestätigung der SAGA-Konformität einer E-Government-Anwendung hat zwei grundlegende Stärken: Verzahnung mit bereits bestehenden Prozessen und flexible Modularisierung inklusive einer Unterscheidung zwischen eigenentwickelten und Produktkomponenten.

### 3 Ausblick auf die nächste SAGA-Version

Die nächste SAGA-Version 5, die für das Jahr 2011 geplant ist [BFIT09], soll durch Beschluss des IT-Rats für alle Bundesressorts verbindlich werden. In dem Zuge ist vorgesehen, die bisherige Klassifikation „Obligatorisch“ durch eine neue Klassifikation „Verbindlich“ zu ersetzen. Außerdem wird SAGA 5 nicht mehr nur für E-Government-Anwendungen Vorgaben machen. Der Geltungsbereich wird auf alle Software-Systeme der Bundesverwaltung ausgedehnt werden. Damit wird auch der SAGA-Konformität und ihrer Prüfung eine erhöhte Bedeutung zukommen.

### Literaturverzeichnis

- [BFIT08] Der Beauftragte der Bundesregierung für Informationstechnik (Hrsg.): SAGA-Konformität. URL: [http://www.cio.bund.de/DE/Standards/SAGA/Konformitaet/konformitaet\\_node.html](http://www.cio.bund.de/DE/Standards/SAGA/Konformitaet/konformitaet_node.html), 2008
- [BFIT09] Der Beauftragte der Bundesregierung für Informationstechnik (Hrsg.): Konzept für SAGA 5.0: URL: <http://www.cio.bund.de/saga>, BMI, Berlin, 2009.
- [BMI08] Bundesministerium des Innern (Hrsg.): Standards und Architekturen für E-Government-Anwendungen. URL: <http://www.cio.bund.de/saga>, BMI, Berlin, 2008.



## SAGA-Konformität Verbindlichkeit von Standards

Rico.Apitz@init.de  
Köln, 4. Mai 2010




]init[

SAGA-Konformität: Verbindlichkeit von Standards

1. Grundlagen der SAGA-Konformität
2. Schwächen der Vergangenheit
3. Stärken der Gegenwart
4. Perspektiven für die Zukunft



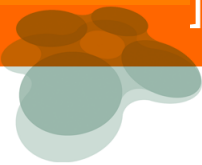


]init[

SAGA-Konformität: Verbindlichkeit von Standards

1. Grundlagen der SAGA-Konformität
2. Schwächen der Vergangenheit
3. Stärken der Gegenwart
4. Perspektiven für die Zukunft

3



]init[

SAGA-Konformität: Verbindlichkeit von Standards

### 1. Grundlagen der SAGA-Konformität

#### Definition der Konformität

- Die SAGA-Konformität einer E-Government-Anwendung wird anhand der in SAGA beschriebenen Modelle, Verfahren und Standards beurteilt:
  - Berücksichtigung standardisierter Prozessmodelle
  - Berücksichtigung standardisierter Datenmodelle
  - Einhaltung der in SAGA beschriebenen technischen Standards und Architekturen
  - Nutzung verwertbarer Einer-für-Alle-Angebote (EfA-Angebote)

4



## 1. Grundlagen der SAGA-Konformität

### Ziele der Beschäftigung mit SAGA-Konformität

- Erreichung der Ziele von SAGA
  - Interoperabilität
  - Wiederverwendbarkeit
  - Offenheit
  - Reduktion von Kosten und Risiken
  - Skalierbarkeit
- Vertragssicherheit
- Steigerung der Handhabbarkeit und Akzeptanz von SAGA
  - in der öffentlichen Verwaltung
  - bei Partnern der öffentlichen Verwaltung

5



1. Grundlagen der SAGA-Konformität
2. Schwächen der Vergangenheit
3. Stärken der Gegenwart
4. Perspektiven in der Zukunft

6



]init[

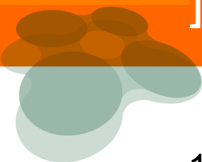
SAGA-Konformität: Verbindlichkeit von Standards

## 2. Schwächen der Vergangenheit

### Pauschale Forderung in Ausschreibungen

- Häufig praktiziert
- Nicht zielführend
  - Aufgrund hoher Komplexität von SAGA
  - Keine Vertragssicherheit durch weiche Formulierungen
  - Erfüllung nicht überprüfbar
  - Ggf. zu teure Angebote
  
- Pauschale Forderung nach SAGA-Konformität ist nicht gewollt

7



]init[

SAGA-Konformität: Verbindlichkeit von Standards

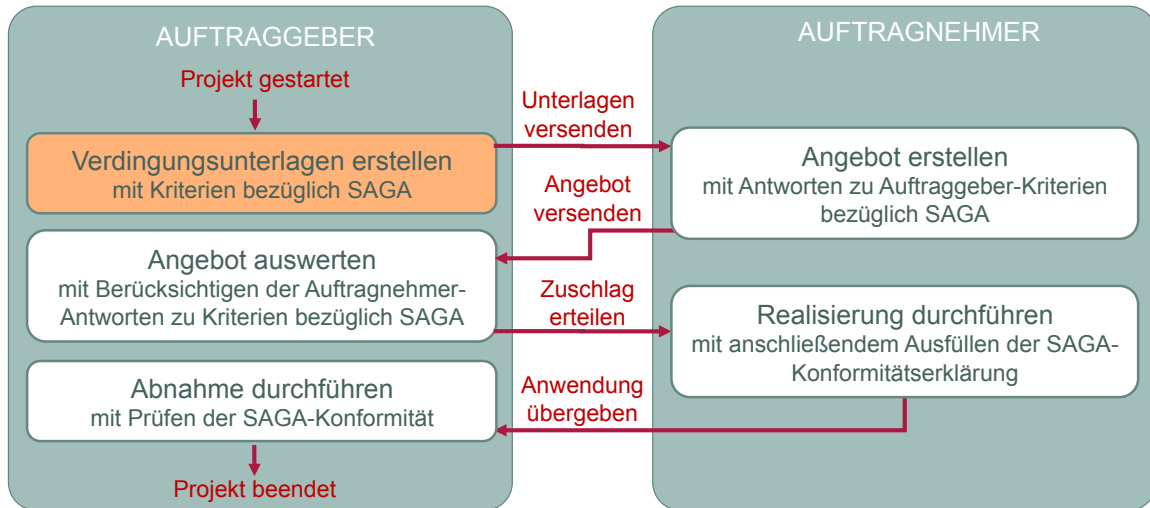
1. Grundlagen der SAGA-Konformität
2. Schwächen der Vergangenheit
3. Stärken der Gegenwart
4. Perspektiven in der Zukunft

8



3. Stärken der Gegenwart

Integration in Ausschreibungs- & Abnahmeprozess (1)

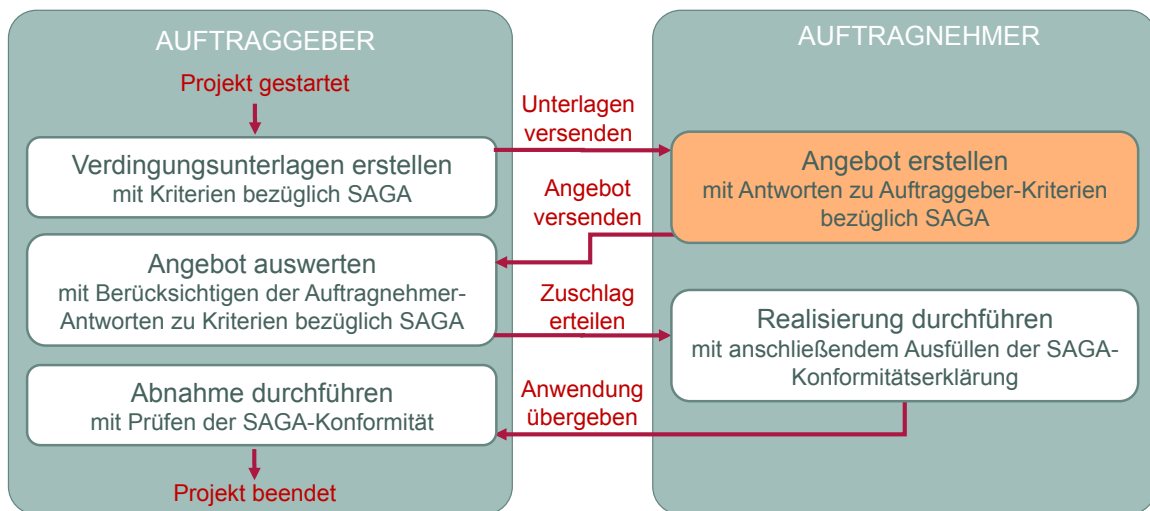


→ Erstellung der Kriteriengruppe „SAGA-Konformität“ als Teil der Verdingungsunterlagen



3. Stärken der Gegenwart

Integration in Ausschreibungs- & Abnahmeprozess (2)

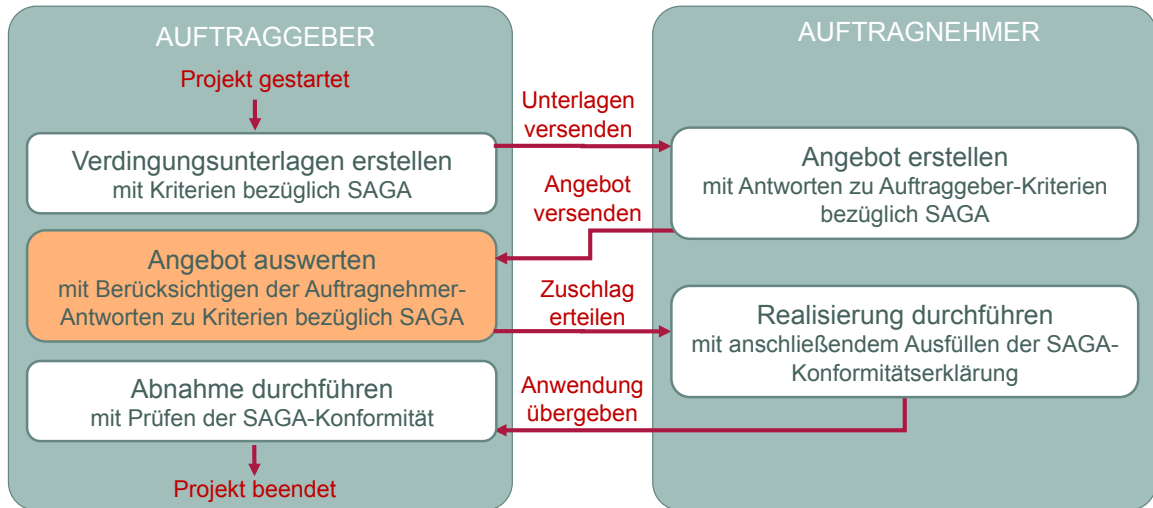


→ Beantwortung der Kriteriengruppe „SAGA-Konformität“ als Bestandteil des Angebots



3. Stärken der Gegenwart

Integration in Ausschreibungs- & Abnahmeprozess (3)

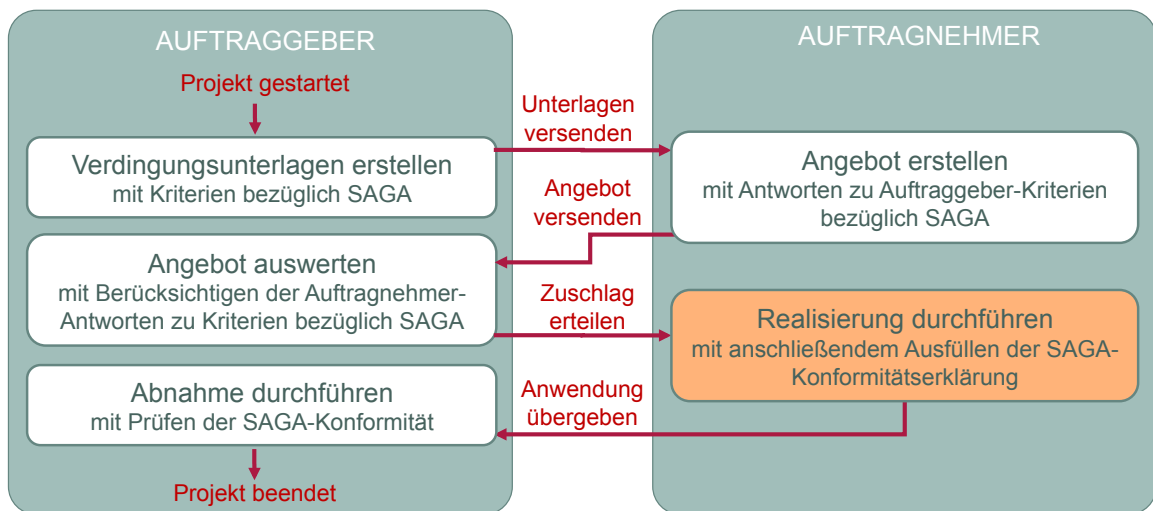


→ Einbeziehung der beantworteten Kriteriengruppe „SAGA-Konformität“ bei der Bewertung der Angebote



3. Stärken der Gegenwart

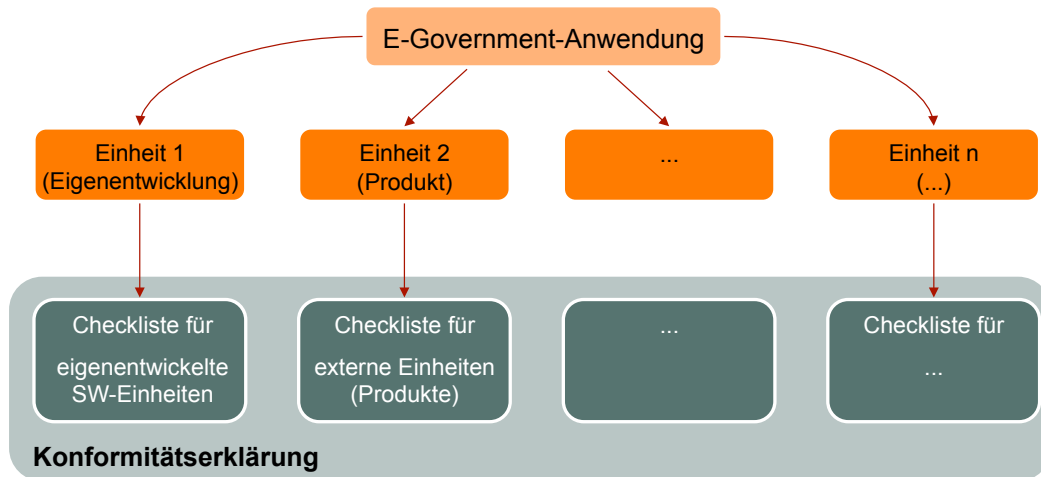
Integration in Ausschreibungs- & Abnahmeprozess (4)



→ Ausfüllen einer SAGA-Konformitätserklärung für jede eigenentwickelte Einheit bzw. Produkte

3. Stärken der Gegenwart

Erstellung einer SAGA-Konformitätserklärung



3. Stärken der Gegenwart

Beispielkonformitätserklärung – Ausschnitt

**Technische Standards und Architekturen**

Applikationsarchitektur (siehe SAGA 4.0, Abschnitt 8.4)

Relevanter Konformitätsaspekt	Standard	SAGA-Konformität gegeben? Ja / Nein
Applikationsarchitektur mit Middleware	Java EE v5	Ja

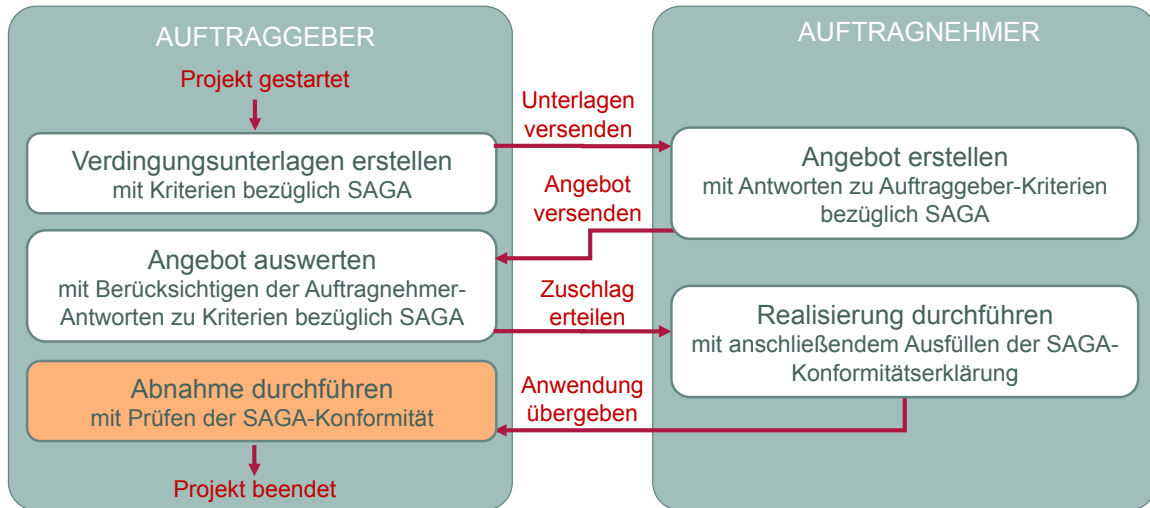
Client (siehe SAGA 4.0, Abschnitt 8.5)

Relevanter Konformitätsaspekt	Anforderungen	Zusicherung	
		ja	nein
Allgemein	Es werden keine Active-X-Controls eingesetzt.	X	
	Bei der Verwendung von aktiven Inhalten existiert ein Parallelangebot ohne aktive Inhalte.	X	



### 3. Stärken der Gegenwart

#### Integration in Ausschreibungs- & Abnahmeprozess (5)



- Einbeziehung der beantworteten Kriteriengruppe „SAGA-Konformität“ sowie der SAGA-Konformitätserklärung

15



### 3. Stärken der Gegenwart

#### Eckpunkte der Konformitätsintegration

- Integration einer Kriteriengruppe „SAGA-Konformität“ in die Ausschreibung
- Detaillierte Angaben über die gewünschten Anforderungen
- Pro
  - Vertragssicherheit
  - Erfüllung überprüfbar
  - Angemessene Angebote
- Contra
  - Höherer Aufwand gegenüber pauschaler Forderung
  - Z.T. technisches Wissen erforderlich

16



### 3. Stärken der Gegenwart

#### Zusammenspiel von V-Modell XT und SAGA

- Hinweis in SAGA auf Verbindlichkeit des V-Modell XT als Vorgehensmodell zur Entwicklung von IT-Systemen in Bundesbehörden
- Hinweis im V-Modell XT Bund auf SAGA als Quelle für technische Anforderungen
- Integration der SAGA-Konformität in den Ausschreibungs- und Abnahmeprozess gemäß V-Modell XT
- Stärkere Anlehnung an die Begriffswelt des V-Modell XT durch SAGA

17




### 3. Stärken der Gegenwart

#### Unterstützung von Projektverantwortlichen

- SAGA-Schulungen mit folgenden Vorteilen
  - Direkte Vermittlung der Ziele und Inhalte von SAGA
  - Geringerer Aufwand bei der Umsetzung der SAGA-Konformität
  - Motivation der Mitarbeiter

18





]init[


SAGA-Konformität: Verbindlichkeit von Standards

### 3. Stärken der Gegenwart

#### Hilfsmittel auf der CIO-Website

- <http://www.cio.bund.de/saga> → Konformität
  - Beispielkriteriengruppe für Auftraggeber
  - Ausgefüllte Beispielkriteriengruppe für Auftragnehmer
  - Beispielkonformitätserklärung für Auftragnehmer
  - Leere Vorlagen für Konformitätserklärungen

19

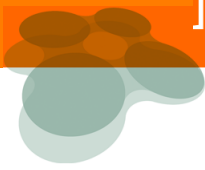


]init[

SAGA-Konformität: Verbindlichkeit von Standards

1. Grundlagen der SAGA-Konformität
2. Schwächen der Vergangenheit
3. Stärken der Gegenwart
4. Perspektiven in der Zukunft

20



]init[

SAGA-Konformität: Verbindlichkeit von Standards

#### 4. Perspektiven in der Zukunft Planungen zu SAGA 5

- Verbindlichkeit für alle Bundesressorts nach Beschluss des IT-Rats
- Neue Klassifikation „Verbindlich“ anstelle von „Obligatorisch“
- Geltungsbereich von E-Government-Anwendungen auf alle Software-Systeme der Bundesverwaltung erweitert
- Daraus folgen mehr Prüfungen der Einhaltung von SAGA-Konformität
  
- Grundsätzlich keine Änderung des aktuell empfohlenen Vorgehens zur Erreichung von SAGA-Konformität

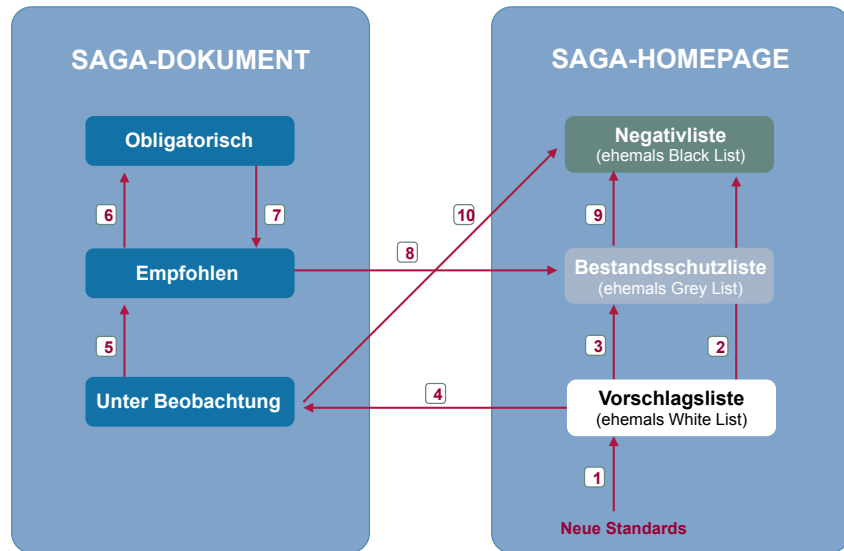
21

]init[  
Digitale Kommunikation

Vielen Dank  
für Ihre Aufmerksamkeit!

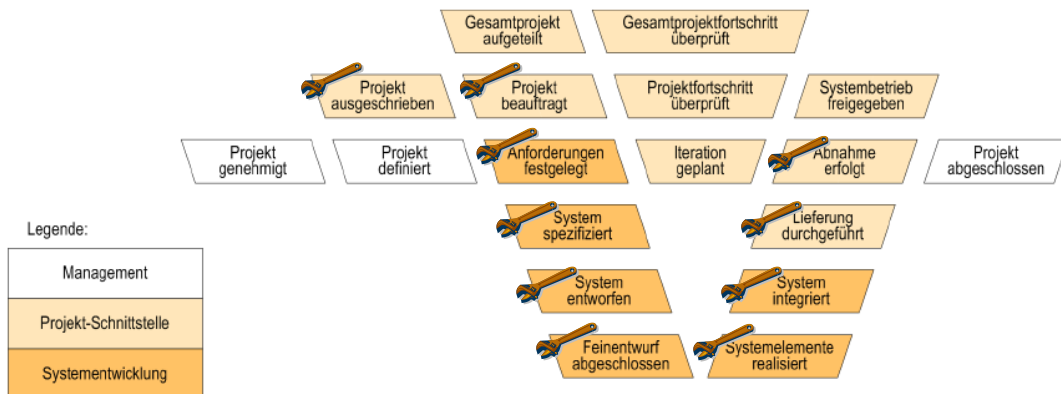
Rico.Apitz@init.de  
Köln, 4. Mai 2010

5. Zugabe  
Lebenszyklen von Standards



23

5. Zugabe  
SAGA-Konformität nach V-Modell XT Bund



24

## **SAGA-konforme Webapplikationen - Herausforderung und Perspektiven**

Dr.-Ing. Thomas Biskup, Dr. Gero presser

QuinScape GmbH  
Wittekindstr. 30  
44139 Dortmund  
thomas.biskup@quinscape.de  
gero.presser@quinscape.de

**Abstract:** Die SAGA-konforme Entwicklung von Webanwendungen bedingt eine Reihe spezieller Herausforderungen, die durch den gezielten und richtigen Einsatz von Open-Source-Lösungen sowohl in Hinblick auf Kosten als auch Risiken effizient behandelt werden können. Dieser Vortrag beschreibt zum Einen die spezifischen Herausforderungen durch Webanwendungen als auch Lösungsansätze und Best Practices für eine nachhaltige Umsetzung webbasierter Fachverfahren.

### **1 Motivation**

SAGA definiert eine Vielzahl von technischen und methodischen Vorgaben für die Entwicklung von eGovernment-Anwendungen. Um Kosten zu reduzieren ist die pauschale Forderung nach SAGA-Konformität nicht gestattet. Gleichzeitig sind die SAGA-Richtlinien an vielen Stellen vage gehalten, um eine wirtschaftlich flexible Umsetzung von Fachverfahren zu unterstützen. In manchen Bereichen (beispielsweise Barrierefreiheit auf Basis von BITV oder der spezifischen Forderung verbindlicher Technologiestandards) gibt es aber auch sehr konkrete, verbindliche und ausschreibungsrelevante Vorgaben. Gerade moderne Webanwendungen („Web 2.0“-basierte Anwendungen auf Basis von AJAX und DHTML) stehen hier vor signifikanten Herausforderungen: Einerseits wird von Fachseite her maximaler Bedienkomfort, hohe Reaktionsfreudigkeit von Anwendungen und eine weitreichende Unterstützung von Gelegenheitsanwendern gewünscht, andererseits müssen solche Anwendungen auch ohne Technologien wie Javascript (oder gar CSS) einsetzbar sein, um Barrierefreiheit sicherzustellen. Insbesondere fordert der SAGA-Standard 4 in Abschnitt 8.5.1.1 b), dass alle wesentlichen Funktionen auch ohne Javascript verfügbar sein sollen. Dies ist mit den meisten modernen Frameworks nur schwer sicherzustellen, da beispielsweise Barrierefreiheit bislang nur eine untergeordnete Priorität in der Entwicklung von Webframeworks genießt.

## 2 Besondere Herausforderungen

Die besonderen sich aus SAGA ergebenden Herausforderungen lassen sich anhand von Anwendungsszenarien in zwei Dimensionen verdeutlichen:

- Webbasierte Angebote können nach der Stärke der Inhaltsstrukturierung unterschieden werden. Zum einen existieren eher schwach strukturierte Inhalte, wie sie typischerweise durch Content Management Systeme verwaltet werden (also „Seiten“, „Abschnitte“, usw., die sich wiederum aus Texten, Formatierungen, Bildern, etc. zusammensetzen) und zum anderen stark strukturierte Inhalte (typische Tabellen und Spalten aus Datenmodellen einer Datenbank).
- Die Komplexität der abgebildeten Prozesse beeinflusst die Wahl der Laufzeitplattform. Eher schwach strukturierte Prozesse finden sich meist in inhaltsbasierten Anwendungen, eher stark strukturierte Prozesse mit vielen Unterschritten und komplexen Arbeitsabläufen sind charakteristisches Merkmal komplexer Fachverfahren und -anwendungen.

Bei dieser Differenzierung wird schnell klar, dass sich für weniger stark strukturierte Inhalte und Prozesse maximal mittlerer Komplexität überwiegend der Einsatz von Content Management Systemen oder Dokumentenmanagementsystemen anbieten wird. Hier kann also auf Produkte zurückgegriffen werden, deren Funktionen und Anforderungen im Rahmen von Ausschreibungen anhand klarer Anforderungslisten abgeprüft und aufgrund der Produkthaftung des Herstellers auch eingefordert werden können.

Werden Prozesse aber äußerst komplex oder sind Daten sehr stark strukturiert, sind solche Systeme eher selten das Mittel der Wahl, da die Komplexität meist mit hohem Customizing-Aufwand einhergeht, der nur dann Sinn macht, wenn das Produkt für die spezielle Fachdomäne besondere Vorteile ausweist. Andernfalls wird eine Individualentwicklung gerade aus Gründen der Flexibilität, Erweiterbarkeit und Zukunftssicherheit das Mittel der Wahl sein, zumal man sich so keinerlei Probleme durch Produkt-Updates und Inkompatibilitäten in den angepassten Produktkomponenten einhandelt.

Daher fokussieren wir unsere weitere Betrachtung auf Laufzeitplattformen für die Individualentwicklung. Hier zeigt sich, dass wesentliche Fallstricke in den folgenden Aspekten liegen:

- *Wirtschaftlichkeit:* Eine Laufzeitplattform für ein individuell entwickeltes System wird sich nur dann wirtschaftlich rechnen, wenn sie für die typischen langen Lebenszeiträume eines Fachverfahrens oder eine Fachanwendung auch hinreichend stabil ist. Betrachtet man gerade aber das Java-Ökosystem – die einzig sinnvolle technische Wahl gemäß den SAGA-Vorgaben für Systeme ab einer mittleren Komplexität, so wird hier schnell klar, dass hier massive Kostenprobleme auftreten können. Denn im Schnitt erneuern sich die meisten Java-Laufzeitplattformen in ihrer Architektur und technischen Ausprägung alle ein bis drei Jahre, so dass ggf. je Generation hohe Migrationskosten anfallen und nach möglicherweise zwei bis drei Generationen einer solchen Plattform gar kein Support oder Know-how für „veraltete“ Versionen verfügbar ist.
- *Sicherheit und Skalierbarkeit:* Hier ist bei einer Individualentwicklung immer darauf zu achten, dass ausreichende Erfahrung in der konkreten Ausprogrammierung der gewünschten Last- und Sicherheitsszenarien mit der gewählten Laufzeitplattform vorliegen. Andernfalls drohen hohe Projektrisiken.
- *Barrierefreiheit:* Barrierefreiheit stellt sich bei genauerer Analyse als überraschend komplexes Problem dar, fordert SAGA 4.0 doch insbesondere in Abschnitt 8.5.1.1 b), dass alle wesentlichen Funktionen einer Anwendung auch bei deaktiviertem Javascript ausführbar sein müssen. Viele Webframeworks vernachlässigen diesen Aspekt aber stark – selbst weit verbreitete Komponentenbibliotheken zu beispielsweise JavaServer Faces (JSF) beinhalten eine Vielzahl von Komponenten, die entweder gar keine AJAX-Unterstützung bieten (und somit zu deutlich reduziertem Bedienkomfort führen) oder aber ohne Javascript nicht mehr fehlerfrei funktionieren.

Zudem zeigt sich, dass die geforderte Interoperabilität voraussetzt, dass Behördenstandards in hinreichendem Maße unterstützt werden. Muss dies bei der Individualentwicklung programmatisch geschehen, bedeutet dies erneute Migrations- und Anpassungsaufwände, sobald sich Standards ändern – selbst wenn es sich nur um kleine Änderungen handelt.

Betrachtet man die heute verbreiteten Frameworks (Java EE, Spring, JSF, GWT, Wicket, usw.) zeigen sich im Detail vielfältige Schwächen bei der Erstellung SAGA-konformer Webanwendungen, so dass der Schluss naheliegt, dass in Analogie zum Government Site Builder für das CMS-Umfeld eine Open-Source-Laufzeitplattform für die Individualentwicklung von SAGA-konformen Anwendungen benötigt wird, die die oben genannten Herausforderungen adressiert und strategisch löst.

Wir empfehlen dazu den Einsatz der OpenSAGA-Plattform (<http://www.opensaga.org>). OpenSAGA ist ein deutsches Open-Source-Projekt, das sich speziell dem Ziel verschrieben hat, die Erstellung von IT-Anwendungen für die öffentliche Verwaltung insbesondere unter Berücksichtigung von SAGA und BITV nachhaltig zu unterstützen.

OpenSAGA setzt auf bewährten Technologien wie Java EE, Spring und JSF auf und bietet ein modellbasiertes Programmiermodell, das die deklarative Beschreibung von Fachverfahren ermöglicht. So können mit erheblich geringerem Einarbeitungs- und Weiterbildungsaufwand als bei klassischen Ansätzen Fachanwendungen entworfen werden. OpenSAGA bietet zudem vielfältige Erweiterungspunkte, die es erlauben, das Standardverhalten des Generators zu überschreiben und so auch hochspezialisierte Algorithmen und Anforderungen umzusetzen. Zudem verankert OpenSAGA die besonderen Anforderungen seitens SAGA und der BITV direkt im Komponentenmodell, so dass Entwicklern z.B. keinerlei Aufwände für die technische Umsetzung von Barrierefreiheit entstehen – denn die Berücksichtigung dieser Aspekte erfolgt durch die OpenSAGA-Plattform. Weiterhin unterstützt OpenSAGA insbesondere die jeweils aktuellen Versionen referenzierter Basisstandards, so dass eine Programmevolution auf neue Versionen durch die technologieneutralen Modelle, die nur in Ausnahmen angepasst werden müssen, deutlich vereinfacht wird.

### 3 Fazit

Die Herausforderungen der Erstellung SAGA-konformer Anwendungen sind mannigfaltig. Für Individualentwicklungen ist insbesondere folgendes zu berücksichtigen:

- Barrierefreiheit und Anforderungen an den Javascript-losen Betrieb einer Anwendung müssen zum frühest möglichen Zeitpunkt geplant werden, da spätere Anpassungen zu signifikanten Aufwänden führen.
- Sicherheit und Skalierbarkeit erfordern je nach Anforderung weitgehende Erfahrungswerte in der jeweiligen Laufzeitplattform – insofern sollten nur gut dokumentierte und hinreichend verbreitete Standards ausgewählt werden.
- Plattformen müssen die individuelle Anpassung enthaltener Standardkomponenten erlauben, um auch hochgradig spezifische Anforderungen umsetzen zu können.
- Werden Standards eingesetzt, die sich kontinuierlich weiterentwickeln, ist zu prüfen, inwieweit dies die Wartungs- und Betriebskosten eines zu erstellenden Systems beeinflusst. Ggf. ist eine Plattform auszuwählen, die besonders wichtige Standards integral unterstützt.

Bei Berücksichtigung dieser Best Practices können auch die vergleichsweise hohen Anforderungen SAGA-konformer Anwendungsentwicklung z.B. durch die Verwendung von Open-Source-Plattformen wie OpenSAGA (<http://www.opensaga.org>) kosteneffizient umgesetzt werden. Mittel- und langfristig wird dadurch insbesondere auch die gesamtwirtschaftliche Kostenbetrachtung derartiger Lösungen vereinfacht, da viele Opportunitätskosten entfallen und einer strategischen Lösungsevolution weichen.

## SAGA-konforme Webapplikationen Herausforderungen und Perspektiven

Dr. Gero Presser, Dr.-Ing. Thomas Biskup  
[www.quinscape.de](http://www.quinscape.de)

### Dr. Gero Presser

- Dr. Gero Presser
  - Geschäftsführer QuinScape GmbH
  - Leiter Marketing & Vertrieb
  - Schwerpunkte:
    - Portale und Webanwendungen
    - Mittelstandsorientierte Portallösungen
    - Kundenbeziehungsmanagement & Vertriebscontrolling





## Dr.-Ing. Thomas Biskup



- Dr.-Ing. Thomas Biskup
  - Geschäftsführer QuinScape GmbH
  - Project Lead OpenSAGA & Enterprise Architect
  - Schwerpunkte:
    - OpenSAGA, Java und Spring
    - Portale und Webanwendungen
    - Modellbasierte Optimierung der Entwicklung von Enterprise-Web-Applikationen
  - Co-Autor des Spring-Praxishandbuchs 1 und 2
  - Autor im Java-Magazin
  - Sprecher auf JAX, W-JAX und wissenschaftlichen Konferenzen
  - Mitglied des Programmkommittees der W-JAX 2009
  - Moderator eGovernment-Day auf der JAX 2010



## Kurzüberblick: QuinScape GmbH



- Unternehmensdaten
  - gegründet Anfang 2001, ca. 50 Mitarbeiter
  - seit Gründung profitabel
  - Stammsitz Dortmund, deutschlandweit tätig
- Schwerpunkte
  - Portale & Webapplikationen, Intranet- und Extranetsysteme
  - Optimierung von Geschäftsprozessen, Informationsflüssen und Unternehmenskommunikation
- Technologien
  - Intrexx (Consulting Partner, führend in Deutschland)
  - OpenSAGA (Project Lead)
  - Spring (SpringSource Premier Systems Integrator)



## Gliederung



- Ziele von SAGA 5.0
- Systematisierung von Webanwendungen
- Betrachtung der Herausforderungen
- Betrachtung der Lösungsansätze
- Empfehlungen
- Fazit

SAGA-konforme Webapplikationen – Herausforderungen und Perspektiven

5

## Ziele von SAGA 5.0



- Technische Ziele
  - Interoperabilität
  - Wiederverwendbarkeit
  - Skalierbarkeit
  - Sicherheit
- Strategische Ziele
  - Wirtschaftlichkeit
  - Agilität
  - Offenheit

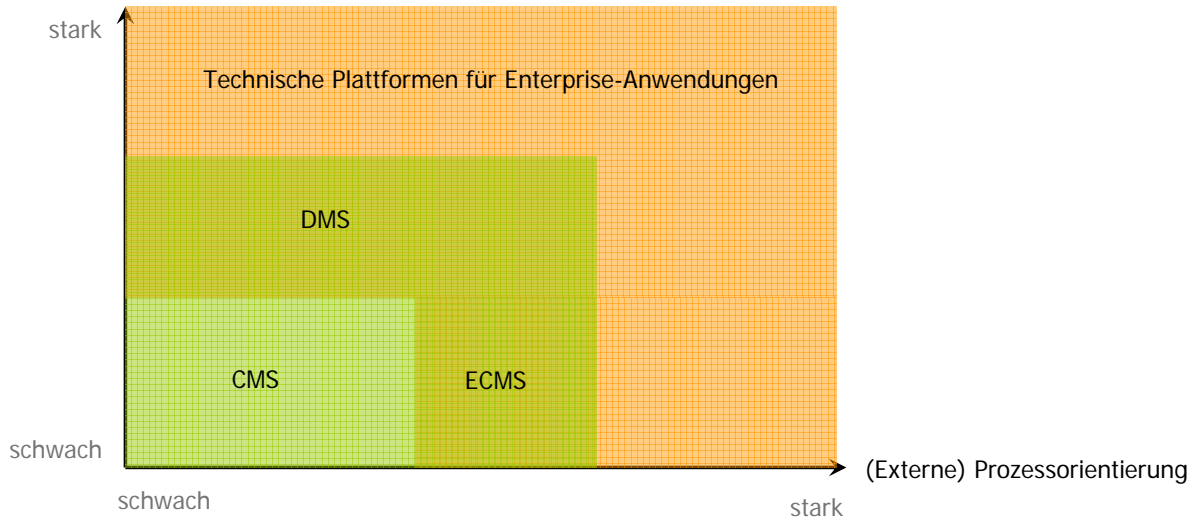
SAGA-konforme Webapplikationen – Herausforderungen und Perspektiven

6

## Dimensionen von Webanwendungen



Strukturierung der Inhalte

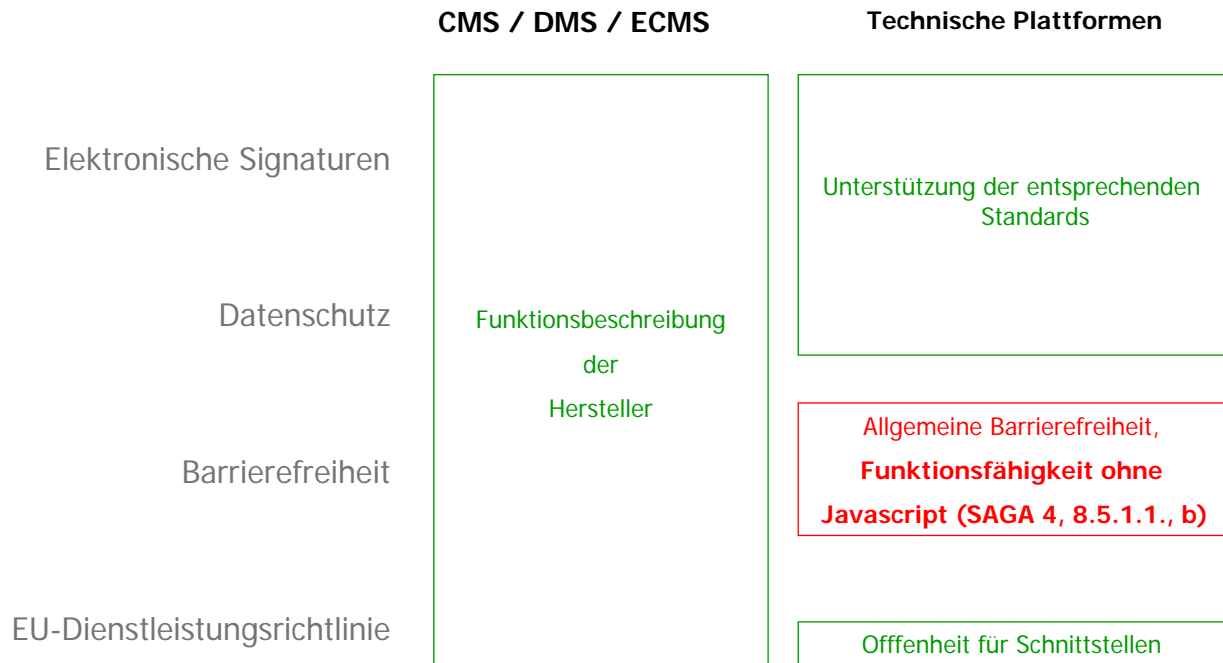


## Herausforderungen durch Ziele



	CMS / DMS / ECMS	Technische Plattformen
Wirtschaftlichkeit	"bei Vendor-Lockin"	bei Auswahl der richtigen langfristigen Strategie
Agilität	Standardisierung, Dokumentation und Verbreitung	
Offenheit	Open Source oder Bundeslizenzen, Schnittstellen, Standards	
Interoperabilität	Schnittstellen, Standards	Schnittstellen, Standards
Wiederverwendbarkeit	Mandantenfähigkeit, Mehrsprachigkeit	Langlebigkeit
Skalierbarkeit	Erfahrungswerte & Referenzen	Erfahrungswerte & Referenzen
Sicherheit	Erfahrungswerte & Referenzen	Erfahrungswerte & Referenzen

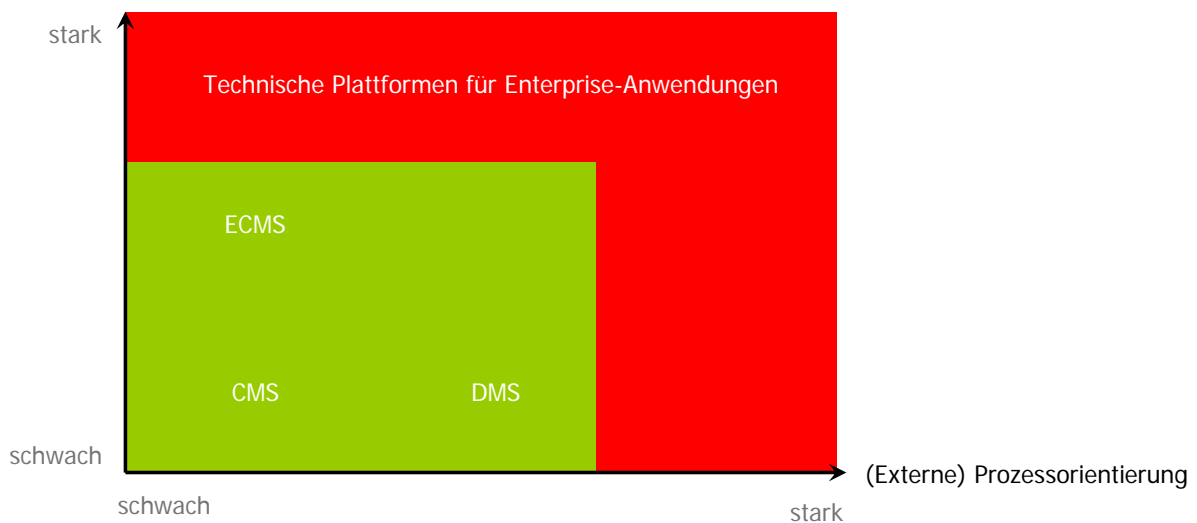
# Herausforderungen durch rechtliche Rahmenbedingungen



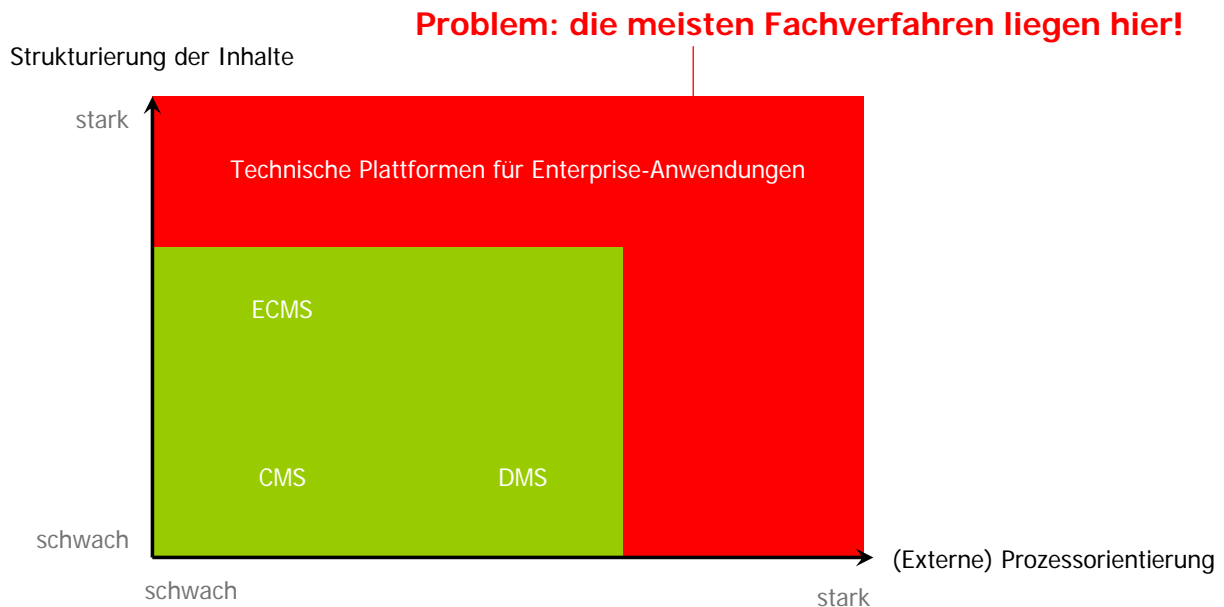
# Herausforderungen



Strukturierung der Inhalte



## Herausforderungen



## Herausforderungen (Zusammenfassung)

- CMS, ECMS, DMS
  - werden gelöst durch Produkthaftung und sinnvolle Formulierung der Ausschreibungsunterlagen
- Technische Plattformen
  - Java obligatorisch für komplexere Anwendungen
  - Ausschreibungsbedingungen können Anforderungen nur temporär behandeln (mangelnder Produktcharakter der meisten Plattformen)
  - **Funktionsfähigkeit ohne Javascript ist schwerwiegendes Problem im Webumfeld**
  - **Plattform entwickelt sich nicht mit neuen Standardversionen weiter** (ggf. im Gegensatz zu Produkten)

## Betrachtung der Lösungsansätze



- nur noch für Enterprise-Web-Plattformen, da hier die zentralen schwer lösbaren Herausforderungen liegen
  - Java EE
  - Spring
  - JSF
  - GWT
  - Wicket
- Fokus auf
  - Skalierbarkeit
  - Sicherheit
  - Barrierefreiheit (insb. ohne Javascript)
  - Wirtschaftlichkeit (Kontinuität)

SAGA-konforme Webapplikationen – Herausforderungen und Perspektiven

13

## Perspektiven



	Skalierbarkeit	Sicherheit	Barrierefreiheit	Wirtschaftlichkeit
Java EE	+1 (bei richtiger Programmierung)	+1 (Vielfalt)	0 (keine Unterstützung, alles manuell)	-1 (Stabilität +, Lernkurve --)
Spring	+1 (bei richtiger Programmierung)	+2 (Vielfalt & Komfort)	-1 (Unterstützung, nur teilweise funktionsfähig)	0 (Plattform ++, JS --)
JSF	0 (bieten keine besondere Unterstützung, behindern aber auch nicht)	0 (erfordern Integration mit Drittsystemen)	-1 (Unterstützung, nur teilweise funktionsfähig)	-1 (benötigt viele Zusätze)
GWT	0 (bieten keine besondere Unterstützung, behindern aber auch nicht)	0 (erfordern Integration mit Drittsystemen)	-10 (funktioniert nicht ohne JS)	-10 (JS --)
Wicket	0 (bieten keine besondere Unterstützung, behindern aber auch nicht)	0 (erfordern Integration mit Drittsystemen)	0 (keine Unterstützung, alles manuell)	-1 (benötigt viele Zusätze)

SAGA-konforme Webapplikationen – Herausforderungen und Perspektiven

14

## Erkenntnis



- Es fehlt eine auf Verwaltungsbedürfnisse optimierte Entwicklungsplattform.
- Funktionsfähigkeit ohne Javascript ist in modernen Anwendungen ein nachhaltiger Kostentreiber.

**Frage:** Geht das nicht besser?

**Antwort:** Ja - OpenSAGA

## Unsere Empfehlung



Modellgetriebene, deklarative und SAGA-konforme Webanwendungen.

<http://www.opensaga.de>

- Open-Source-Plattform
- Technologien: Java EE, JSF und Spring
- Zielsetzung: SAGA-Konformität (& BITV & ...)
- Entwickelt sich mit Verwaltungsstandards weiter (Produkt)

## OpenSAGA im Überblick



- **Entwicklungsmetapher**
  - Deklarativ, intentional ("Was soll die Anwendung leisten?")
  - Modellbasiertes (Strenges MDSD)
  - Strategie-Interfaces für programmatische Erweiterungen
- **Mehrwert**
  - Hochgradige Technologieabstraktion (einfache Migration, Einarbeitung und Weiterbildung)
  - BITV-Konformität "out of the box"
  - SAGA-Konformität als zentrales Designziel
  - Neue Standards werden durch Austausch der Laufzeitversion berücksichtigt – normalerweise kein Migrationsaufwand für die Fachanwendungen

## Fazit



- Eine strategische **Entwicklungsplattform** für Anwendungen der öffentlichen Verwaltung (und insbesondere eGovernment-Anwendungen) unterstützt die nachhaltige Erreichung der SAGA-Ziele!
- Ohne eine strategische Entwicklungsplattform bleibt vor allem Wirtschaftlichkeit ein "moving target".
- OpenSAGA ist bislang die einzige strategische Open-Source-Plattform für Anwendungen der öffentlichen Verwaltung: <http://www.opensaga.org>



### 12.3 SAGA-konforme Webapplikationen – Herausforderung und Perspektiven