

TUM

INSTITUT FÜR INFORMATIK

Proceedings zur SEE 2009

Jan Friedrich, Marco Kuhrmann, Andreas Rausch,
Marc Sihling



TUM-I0922
September 09

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM-INFO-09-I0922-0/1.-FI

Alle Rechte vorbehalten

Nachdruck auch auszugsweise verboten

©2009

Druck: Institut für Informatik der
Technischen Universität München

SEE 2009

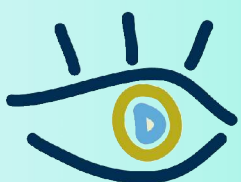
4SOFT



TU Clausthal



25.-27.
Mai '09
Berlin



Software & Systems
Engineering
Essentials

www.see-conf.de

Editoren

Jan Friedrich¹, Dr. Marco Kuhrmann², Prof. Dr. Andreas Rausch³, Dr. Marc Sihling¹

¹4Soft GmbH
Mittererstr. 3
80336 München
{friedrich | sihling}@4soft.de

²Technische Universität München, Institut für Informatik – I4
Boltzmannstr. 3
85748 Garching b. München
kuhrmann@in.tum.de

³Technische Univesität Clausthal, Institut für Informatik – Software Systems Engineering
Albrecht-von-Groddeck-Str. 7
38678 Clausthal-Zellerfeld
andreas.rausch@tu-clausthal.de

Inhaltsverzeichnis

Einführung	iii
Überblick	iii
Zusammenfassung	iii
I. Tag 1	1
1. Session 1: Anpassung und Einführung von Vorgehensmodellen	3
1.1. Die Einführung der Methode HERMES im Unternehmen	4
1.2. V-Modell XT – Ausgangsbasis für professionelle Prozesse	23
1.3. Das Vorgehensmodell Bayern im neuen Gewand	35
2. Session 2: Projektmanagement	51
2.1. Das Sporthemd unterm Anzug V-Modell XT with Scrum inside	52
2.2. Projektmanagement nach PMI im V-Modell XT	57
2.3. Projektplanung im neuen V-Modell XT V1.3: Wie geht das denn genau?	81
3. Session 3: Werkzeuge zur Kollaboration in Projekten	97
3.1. Koordinierte Softwareentwicklung im Team mittels Jazz	98
3.2. Evaluierung von Werkzeugen für Distributed Pair Programming: Eine Fallstudie	111
3.3. Flexible Prozess-Werkzeug Integration	123
4. Session 4: Nutzen und Schwächen von Vorgehensmodellen	135
4.1. Einsatz des V-Modell XT in der Behördenpraxis	136
4.2. ITIL Prozesse in der BWI Systeme GmbH	150
4.3. Kostenseitige Begleitung von Vorgehensmodellen	164
5. Session 5: Agiles Schätzen und Planen	181
5.1. Agiles Schätzen und Planen	182
5.2. Planungsebenen der iterativen Systementwicklung	198
5.3. Pragmatische Aufwandsschätzung	212
6. Session 6: Modellbasierte Systementwicklung	225
6.1. Modelle verbinden: Wie Projektleiter und Entwickler reibungslos zusammenarbeiten	226
6.2. Modellbasierter Ansatz im Multi-Projekt-Umfeld	237
6.3. Effiziente Wege zum Datenaustausch zwischen Behörden	250
II. Tag 2	259
7. Session 7: Anpassung und Einführung von Vorgehensmodellen	261
7.1. flyXT – Das neue Vorgehensmodell der EADS DE	262
7.2. Studie für die Einführung HERMES PowerUser in der PostFinance	276

7.3. Projekt Recovery nach dem War Room Konzept	293
8. Session 8: Unternehmensweites Architektur- und Anforderungsmanagement	313
8.1. IT-Architekturmanagement	314
8.2. From Architecture to Source Code	334
8.3. Telepathie im Automotive-Bereich	345
9. Session 9: Agiles Projektmanagement	353
9.1. Tool-gestütztes Vorgehen nach Scrum mit Projektron BCS	354
9.2. Multiprojektmanagement mit dem Projectserver	362
9.3. Gezähmte Papiertiger: Mit Tailoring eine Methode für das Projekt nützlich machen	382
10. Session 10: Fundamente für Vorgehensmodelle	393
10.1. SPEM und ISO/IEC 24744 im Vergleich	394
10.2. Werkzeuggestützte Konsistenzsicherung von SE-Prozessen	409
10.3. Erstellung und Bereitstellung von Prozessbeschreibungen	423
11. Session 11: Test und Wartung von IT-Systemen	435
11.1. Funktionales Agiles Testen durch einen Keyword-Driven Ansatz	436
11.2. SW quality improvements by using quality KPIs	446
11.3. Effort Reduction and Error Prevention during Software Development and Maintenance	467
12. Session 12: Werkzeuge zur Kollaboration in Projekten	485
12.1. Generierung von Prozess Templates für IBM Jazz auf Basis des V-Modell XT486	
12.2. Unica - a unified CASE Tool	499
12.3. Anforderungen nachweisbar erfüllen	508

Einführung

Prozessverantwortliche und Projektleiter erfahren oft leidvoll, dass sinnvolle Vorgaben und Standards nicht eingehalten und gelebt werden, weil sie zunächst einmal mit Mehrarbeit verbunden sind. Die Folge sind Qualitätsmängel bei den Projektergebnissen und ineffiziente Prozesse. Geeignete Werkzeugumgebungen versprechen einen Ausweg: Indem sie die Projektmitarbeiter bei ihrer Arbeit unterstützen, erleichtern sie die Anwendung von Prozessen und Methoden oder ermöglichen sie sogar erst.

Werkzeuge spielen damit eine Schlüsselrolle bei der Vereinheitlichung und Standardisierung der Projektarbeit. Die SEE 2009 greift speziell diesen Aspekt auf und beantwortet Fragen wie: Welche Werkzeuge sind am Markt verfügbar? Wie erhalte ich eine durchgängige Werkzeugumgebung? Wann setze ich integrierte Herstellersysteme ein, wann heterogene Best-of-Breed Tools und wann Open-Source-Werkzeuge? Welche Werkzeuge passen zu meinem hausinternen Vorgehensmodell? Wie gehe ich bei der Auswahl und der Einführung vor? Und schließlich: Welchen Nutzen bringen die unterschiedlichen Werkzeuge wirklich?

Überblick

Die Konferenz fand vom 25. bis 27. Mai 2009 in Berlin statt. Den Besuchern wurden Keynotes, Workshops und Vorträge, eine Tool Time und ein Tool Shootout sowie eine Ausstellung von Werkzeugherstellern und Beratungsfirmen geboten. Als Keynoter haben Prof. Dr. Ernst Denert, Vorstandsvorsitzender der IVU Traffic Technologies AG und Honorarprofessor an der Technischen Universität München, sowie Dr. Hans Bernhard Beus, Staatssekretär im Bundesministerium des Innern und Beauftragter der Bundesregierung für Informationstechnik, gesprochen. Die 36 Vorträge waren in folgende Themengebiete gegliedert:

- Anpassung und Einführung von Vorgehensmodellen
- Projektmanagement
- Werkzeuge zur Kollaboration in Projekten
- Nutzen und Schwächen von Vorgehensmodellen
- Agiles Schätzen und Planen
- Modellbasierte Systementwicklung
- Unternehmensweites Architektur- und Anforderungsmanagement
- Agiles Projektmanagement
- Fundamente für Vorgehensmodelle
- Test und Wartung von IT-Systemen

Zusammenfassung

Krisenzeiten sind schwierig für Konferenzen. Weiterbildungsbudgets fallen dem Rotstift zum Opfer und Reisen werden seltener genehmigt. Die Probleme bei der Durchführung von Projekten sind dieses Jahr aber nicht minder brisant, und der Informationsaustausch ist wichtiger denn je. In sofern freut es uns besonders, dass wir auch im Krisenjahr 2009 die SEE-Konferenz durchführen konnten und bedanken uns bei allen Beteiligten und Teilnehmern für die Unterstützung.

Programmkomitee

- Prof. Dr. Urs Andelfinger (SEI Europe/Hochschule Darmstadt)
- Prof. Dr. Stefan Biffel (OCG/Technische Universität Wien)
- Hubert Biskup (IBM Rational Software)
- Prof. Dr. Ruth Breu (Universität Linz)
- Nils Cordes (Bundesministerium des Innern, IT-Steuerung Bund)
- Jutta Eckstein (IT Communication)
- Stephan Görlicke (ISQI GmbH)
- Roger Griessen (Informatikstrategieorgan Bund ISB)
- Prof. Dr. Volker Gruhn (Universität Leipzig)
- Dr. Helmut Hummel (IT Beratung)
- Wolfgang Kranz (EADS)
- Dr. Marco Kuhrmann (Technische Universität München)
- Dr. Christian Lange (Bundesstelle für Informationstechnik, BIT)
- Peter Lang (Fa. PETER LANG MANAGEMENT- & IT-CONSULTING, Thun)
- Ludger Meyer (Siemens AG)
- Dr. Jürgen Münch (Fraunhofer IESE)
- Bernd Oestereich (oose Innovative Informatik GmbH)
- Prof. Dr. Andreas Rausch (Technische Universität Clausthal)
- Martin Rother (PRINCE2 Deutschland e.V.)
- Erik Saas (IT Amt der Bundeswehr)
- Dr. André Schnackenburg (Bundesstelle für Informationstechnik, BIT)
- Dr. Marc Sihling (4Soft GmbH)

Workshoporganisation

- Dr. Marco Kuhrmann (Technische Universität München)
- Dr. Marc Sihling (4Soft GmbH)

I.
Tag 1

1. Session 1: Anpassung und Einführung von Vorgehensmodellen

Inhalt

1.1. Die Einführung der Methode HERMES im Unternehmen	4
1.2. V-Modell XT – Ausgangsbasis für professionelle Prozesse	23
1.3. Das Vorgehensmodell Bayern im neuen Gewand	35

1.1. Die Einführung der Methode HERMES im Unternehmen

Hélène Mourgue d'Algue, Informatikstrategieorgan Bund ISB

Motivation and Context

HERMES ist die Projektführungsmethode in der Informations- und Kommunikationstechnik (IKT) der schweizerischen Bundesbehörden. Seit ihrer Einführung im Jahre 1975 wurde die Methode drei grundlegenden Revisionen - mit neuen Ausgaben in den Jahren 1986, 1995 und 2003 - unterzogen, um sie an die aktuellsten methodischen Anforderungen anzupassen. Die Methode wurde also bisher im Rahmen eines rund zehn Jahre dauernden Lebenszyklus gepflegt und weiter entwickelt.

Zwischen diesen Revisionen müssen Benutzer die Methode in ihren eigenen Strukturen und internen Prozessen einführen. Die konsequente Übernahme eines Produktes garantiert auch eine effiziente Nutzung. Die Methode muss sowohl die interne Kultur fördern als auch den Gewohnheiten der Organisation entsprechen. Dafür entstehen immer öfters so genannten Projektoffices, die als Kernaufgabe die neuen Produkte aus dem Markt der Methoden integrieren und an die Bedürfnisse des Hauses anpassen.

Allerdings haben Methodenhersteller die Aufgabe, diese Aspekte bei der Integration zu unterstützen. Dafür steht ihnen einiges zu Verfügung: die Entwicklung von Tools, die Begleitung der Benutzer, die Publikation von Artikeln, die Zielorientierte Ausbildung, usw. Vorgängig soll aber auch eine Vorbereitungsarbeit geleistet werden, indem man die heutigen Organisationsformen berücksichtigt beim Aufbau einer Methode. Die Herausforderung für die Hersteller von Methoden wie HERMES ist es, die aktuellen Trends im IT Projektmanagement zu untersuchen und gleichzeitig die Bedürfnisse von Projektoffices zu verstehen, um die Umsetzung zu erleichtern.

Approach

Nach einer kurzen Einführung zu HERMES wird vertieft auf die konkreten Anforderungen an die Übergabe von Vorgehensmodellen in die Praxis eingegangen. Eine doppelte Perspektive wird dafür gebraucht: die Organisation als Empfängerin und die Organisation als Herstellerin.

Aufgrund bei zahlreichen Benutzern gesammelter Erfahrungen werden die Schwerpunkte einer Einführung detailliert. Organisationsmodelle, IT Umgebung, Prozesslandschaft und die Verknüpfung mit weiteren Methoden bilden den Leitfaden durch diesen Teil des Referates. Darüber hinaus werden die heutigen Trends der Arbeitsorganisation dargestellt, die in diesem Umfeld relevant sind. Anschliessend wird die Sicht der Hersteller analysiert. Basierend auf eigenen Erfahrungen des HERMES Teams, soll sich dieser Teil des Referates vor allem auf Forschungsarbeiten und Studien aus dem IT Umfeld abstützen.

Conclusions and Outlook

Konklusion: Ein Methodenhersteller muss sich vertieft mit der Arbeitsumgebung beschäftigen, für die seine Methode konzipiert wird. Die schlichte Sicht auf die Methode an sich ist ungenügend, um den Erfolg in der Praxis zu garantieren. Arbeitssoziologie, Architekturstudien, Forschungsergebnisse im Bereich Organisation sind einige der nötigen Komponenten, um die Einführung einer Methode in Unternehmen zu unterstützen.

Die Einführung der Methode HERMES im Unternehmen

26.05.2009 – SEE 2009
Hélène Mourgue d'Algue



Plan

1. Die HERMES Methode
2. Einführung von HERMES
3. Die Erstellung der Methode



1. Die HERMES Methode

- Werteumgebung
- Ein Phasenmodell / 2 Projekttypen
- Submodelle
- Drei Sichten auf das Projekt
- Das Tailoring



Informatikstrategieorgan Bund ISB

• Aufgabe ISB:

Das Informatikstrategieorgan Bund ISB ist die Stabstelle des Informatikrates Bund IRB.

Es erarbeitet die Entscheidungsgrundlagen für die strategische Steuerung der Informatik in der Bundesverwaltung.

• Methode

Bundesweit einheitliche Methoden und Werkzeuge stellt das Informatikstrategieorgan Bund ISB zur Verfügung.

- Strategische Informatikplanung SIP
- Informatikcontrolling-Methode ICO
- Projektführungsmethode HERMES
- ...



Offenheit und Flexibilität

Konkret:

- Mehrsprachig: Deutsch – Französisch – Italienisch – Englisch (teilweise)
- Freie Verwendung und Verfügbarkeit der Daten
- Einsetzbar in jedem Unternehmen → Projektorganisation
- Eingesetzt Kanton Bern, Stadt Zürich, Stadt Biel, Postfinance, Fachhochschulen, Kantonspital Waadt, ...
- HERMES Fachgruppe im Verein eCH: die Benutzer beteiligen sich



Vollständigkeit

- Handbücher



Tools



- Web / Newsletter

HERMES Newsletter - Februar 2009

Neuigkeiten zur Projektführungsmethode HERMES

- Ausbildung

- Zertifizierung: von SAQ durchgeführt 2 Stufen



Standardisierung



- Basiert auf internationalem Standard: SPEM
Software Process Engineering Meta-Model

➤ Qualität und Kompatibilität

- Verein eCH: Standard für E-Government Projekte
- Standard in der schweiz. Bundesverwaltung

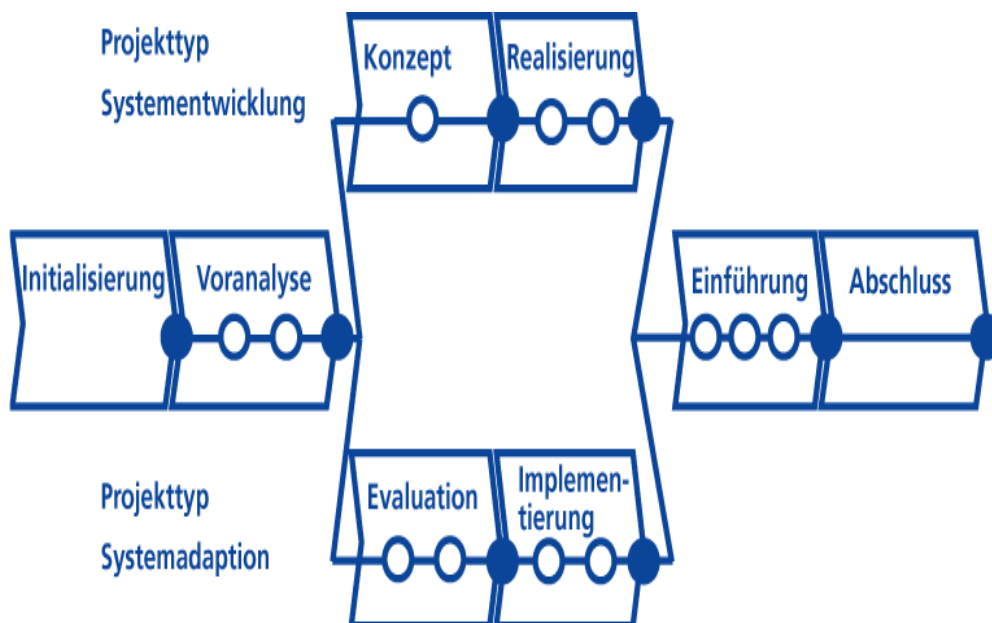


Verständlichkeit

- Gemeinsamer Wortschatz und Vorgehen
- Die Konzepte sind klar
- Nicht nur für IT-Profis
- Konzept ist in wenigen Minuten erklärbar

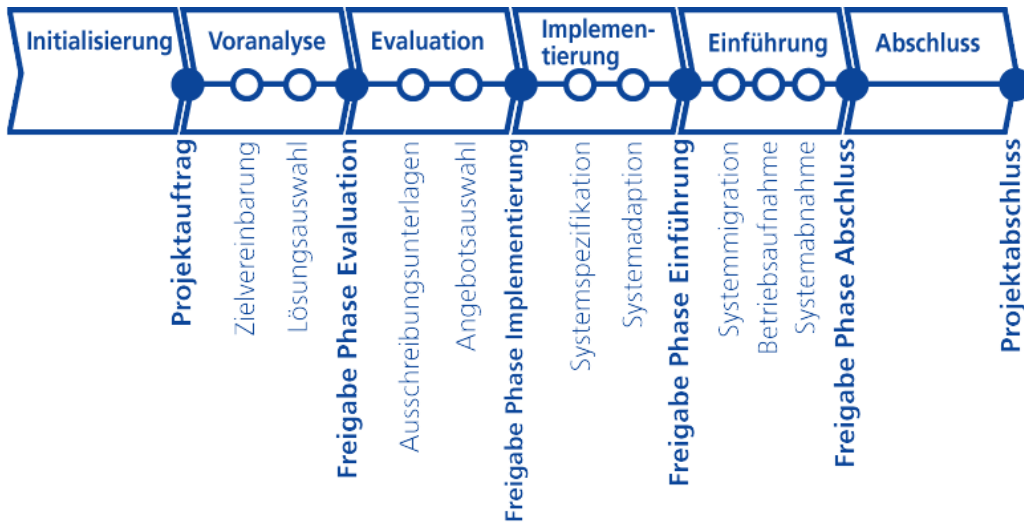


Ein Phasenmodell / 2 Projekttypen

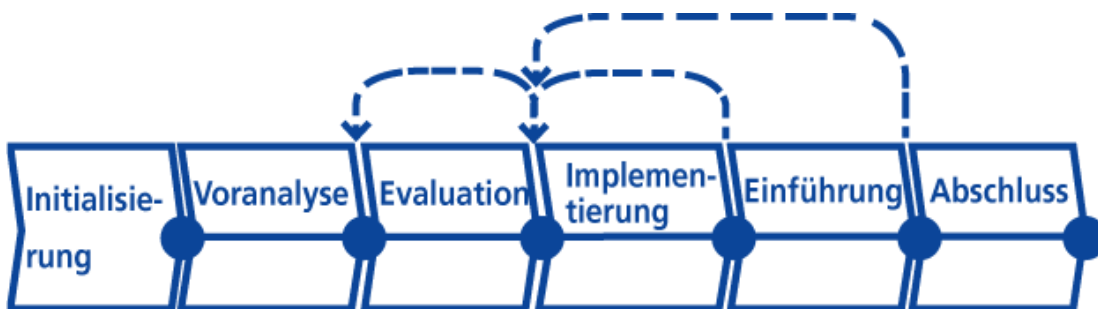




Projekttyp mit Entscheidungspunkten



Phasen-Iteration



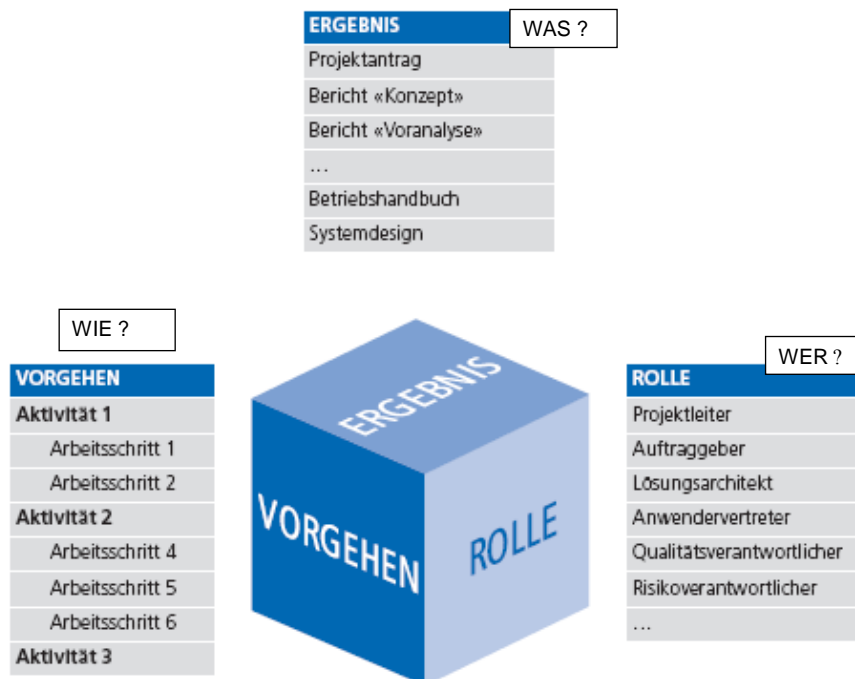


Submodelle für die Querschnittsfunktionen

- Projektmanagement
- Qualitätssicherung
- Risikomanagement
- Konfigurationsmanagement
- Projektmarketing



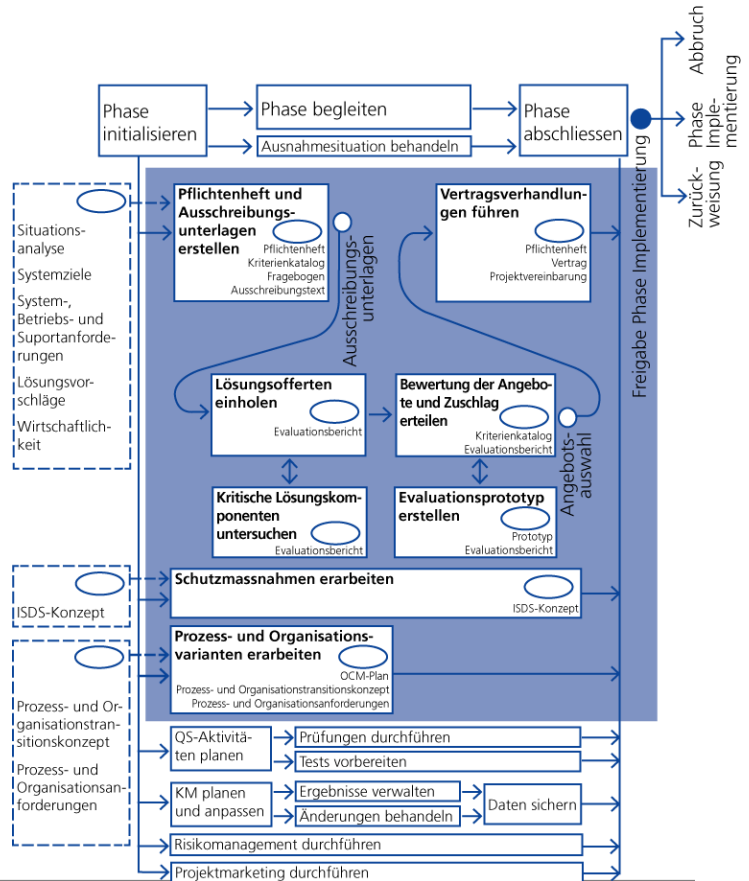
Drei Sichten auf das Projekt



1.1. Die Einführung der Methode HERMES im Unternehmen



Vorgehen: Wie wird gearbeitet?



Eidgenössisches Finanzdepartement EFD
 Informatikstrategieorgan Bund ISB

moh /26.05.2009



15



Ergebnis: Was wird geliefert?

Projektantrag

Titel	Wichtigste Identifikationsmerkmale
Beschreibung	Einzelteil in Prüfung / abgeschlossen
Projektkategorie	HERMES für Organisationsprojekte
Projektschritt	IS-Organisation
Projektnummer	6661
Projektleiter	Hélène Mouque d'Algue
Auftraggeber	ICH - Änderungsausdruck
Autor	Hélène Mouque d'Algue
Initiales	moh
Bearbeitendes	Beauftragte
Prüfendes	Prüfer
Genehmigendes	Genehmigende
Verfasser	Verfasser
Datum	Dokumentationsnummer
Kurzbeschreibung	Kurzbeschreibung

Änderungskontrolle, Prüfung, Genehmigt

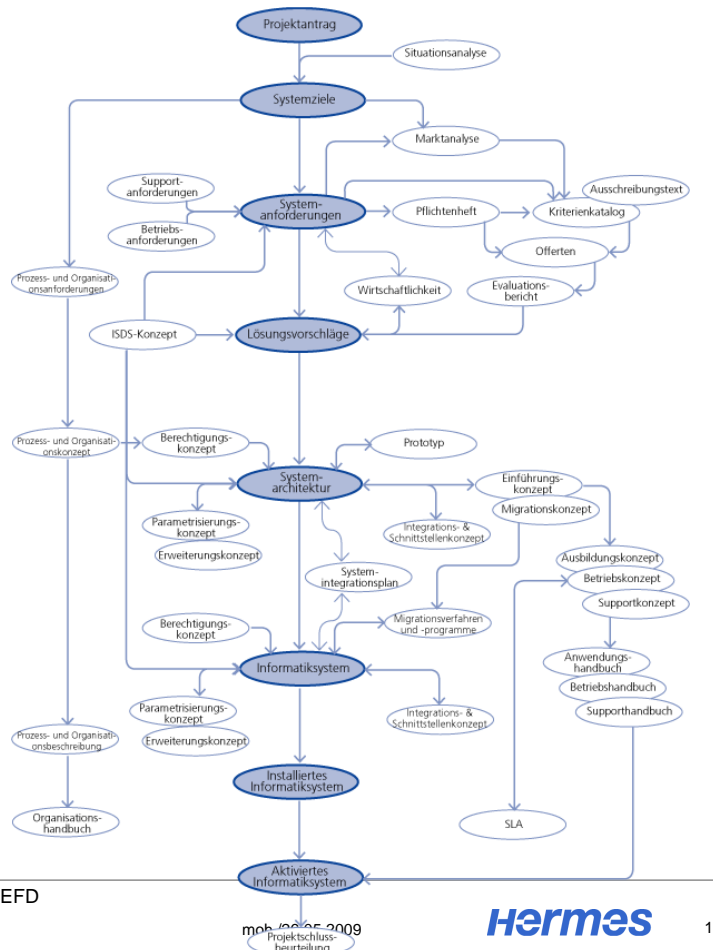
Version	Datum	Beschreibung, Bemerkung	Name oder Rolle

Definieren, Abnormen und Abhandlung

Begriff/Abkürzung	Bedeutung

Referenzen

Bezeichnung	Titel, Quelle



Eidgenössisches Finanzdepartement EFD
 Informatikstrategieorgan Bund ISB

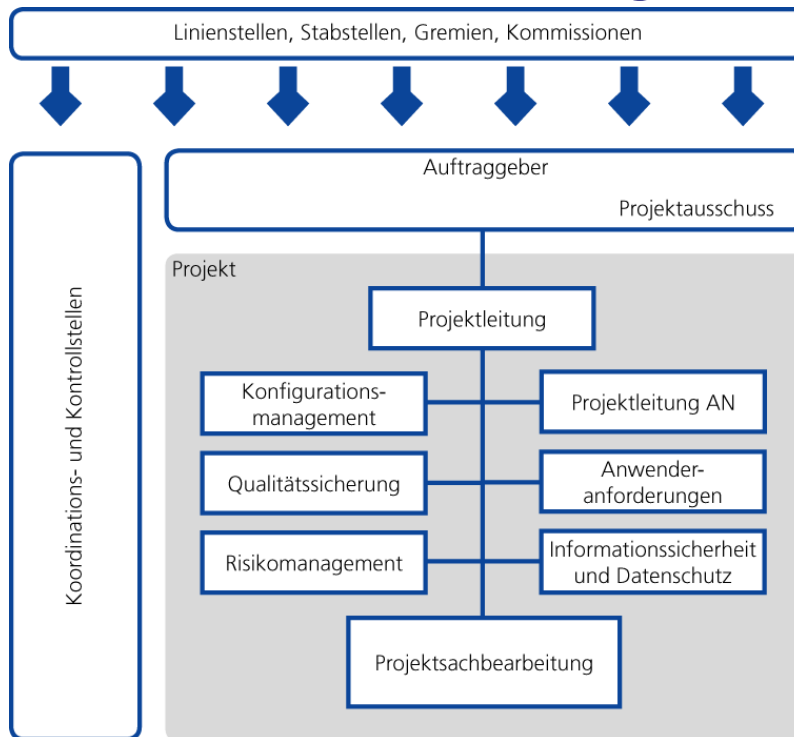
moh /26.05.2009
 Projektschlussbeurteilung



16



Rolle: Wer ist zuständig?



Das Tailoring

Was bedeutet es?

- Anpassung des Vorgehens an die jeweilige Projektsituation

Warum braucht man es?

- Insgesamt beinhaltet HERMES mehr als 100 Aktivitäten und Ergebnisse

Wie funktioniert es?

- Vor dem Projekt -> Projektorganisation am Anfang
- Während dem Projekt -> fortlaufendes Tailoring



2. Einführung von HERMES

- HERMES eine Wunderlösung?
- Integration im Projektmanagement Umfeld
- HERMES: abgespeckt
- HERMES: mittel
- HERMES: umfangreich

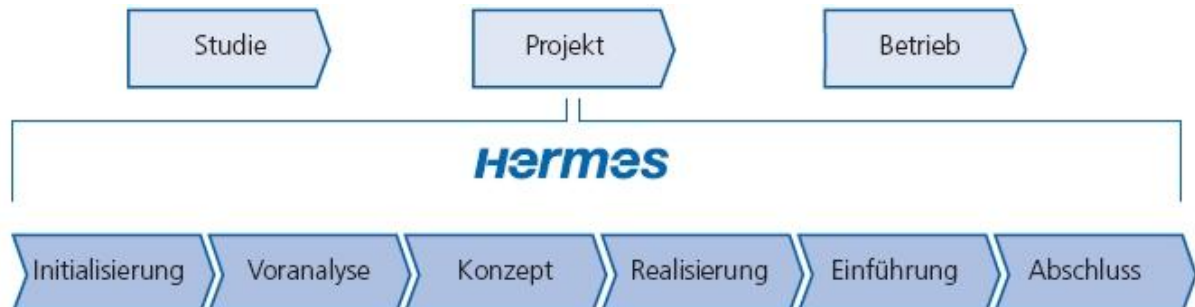


HERMES eine Wunderlösung ?

- Zu berücksichtigen:
 - Organisationsmodell
 - IT-Landschaft
 - Weitere Methoden
 - Kultur
- Konkrete Fragen:
 - Welche Ergebnisse / Aktivitäten / Rollen sind Pflicht?
 - Wie sind die Schnittstellen zu Budgetierung, HR, Betrieb, Sicherheit,...?
Wie sehen die Vorlagen aus?
 - Wo werden die Daten gespeichert?
 - Welches Tool zum Risikomanagement?



Integration im Projektmanagement Umfeld



Genauere Vorgaben zur Einführung von HERMES kann man nicht definieren, weil die Bedingungen unterschiedlich sind. Man kann jedoch den Spielraum aufzeigen.



HERMES: abgespeckt

- Für wen ?**
- 5 Projektleiter
 - Organisation basiert auf unformelle Beziehungen
 - Wenig Ressourcen für die Einführung
 - z.B. Verein, klein KMU
- Wie ?**
- **Methode:** 10 Ergebnisse - Aktivitäten
 - **Werkzeuge:** eingesetzt Kollaborationstool, Risikomanagement
 - **Organisation:** 1 Verantwortlicher für die Methode
 - **Ausbildung:** 1/2 Tag Ausbildung



Beispiel

Phasen SA	Aktivitäten	Ergebnisse	keep
Initialisierung	Projekt initialisieren	Projektantrag	1
	Entscheid über Projektauftrag treffen	Projektentscheide	
Voranalyse	Lösungen suchen	Systemanforderungen Lösungsvorschläge	1
	Phase abschliessen	Bericht <<Voranalyse>>	
Evaluation	Entscheid über Freigabe Phase Evaluation treffen	Projektentscheide	
	Lösungsofferten einholen	Pflichtenheft	1
		Evaluationsbericht	1
		Testplan	1
	Phase abschliessen	Bericht <<Evaluation>>	
	Entscheid über Freigabe Phase Implementierung treffen	Projektentscheide	
Implementierung	Detailkonzept entwickeln	Erweiterungskonzept Parametrisierungskonzept	1
		Systemintegrationsplan	1
	Einführungs- und Betriebskonzepte entwickeln	Einführungskonzept	1
		SLA	1
	Phase abschliessen	Bericht <<Implementierung>>	
	Entscheid über Freigabe Phase Einführung treffen	Projektentscheide	
Einführung	Abnahmetests durchführen	Testbericht	1
	Phase abschliessen	Bericht <<Einführung>>	
	Entscheid über Freigabe Phase Abschluss	Projektentscheide	



HERMES: mittel

- Für wen ?**
- Amt, mittlere KMU
 - 40-50 Projektleiter -> Zahlreiche Benutzer
 - Eine PM-Methode existiert schon, wird aber nicht gebraucht
 - Ressourcen für die Einführung

Wie ?

- **Methode:** 10 Ergebnisse – Aktivitäten + interne Vorschriften
- **Werkzeuge:** eingesetzt Kollaborationstool, Risikomanagement
- **Organisation:** Projekt Office
- **Ausbildung:** 1 Tag Ausbildung (HERMES Pur + HERMES bei uns)



Beispiel: Kantonale Verwaltung

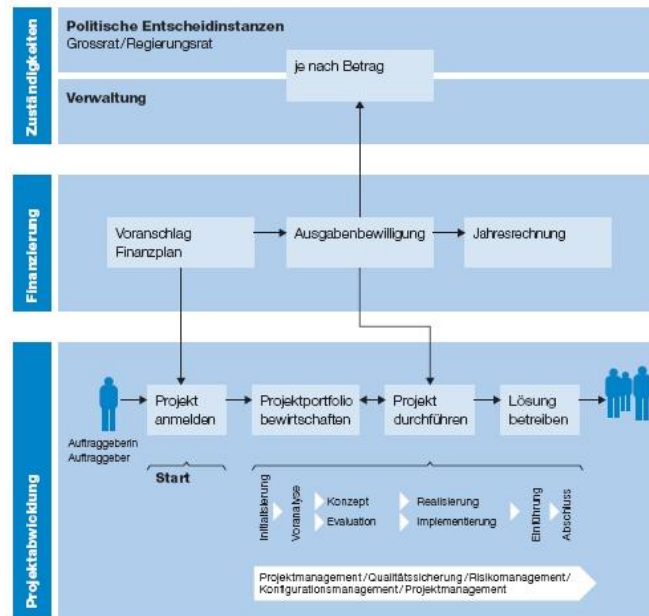


Abbildung 2: Projektlandschaft



HERMES: umfangreich

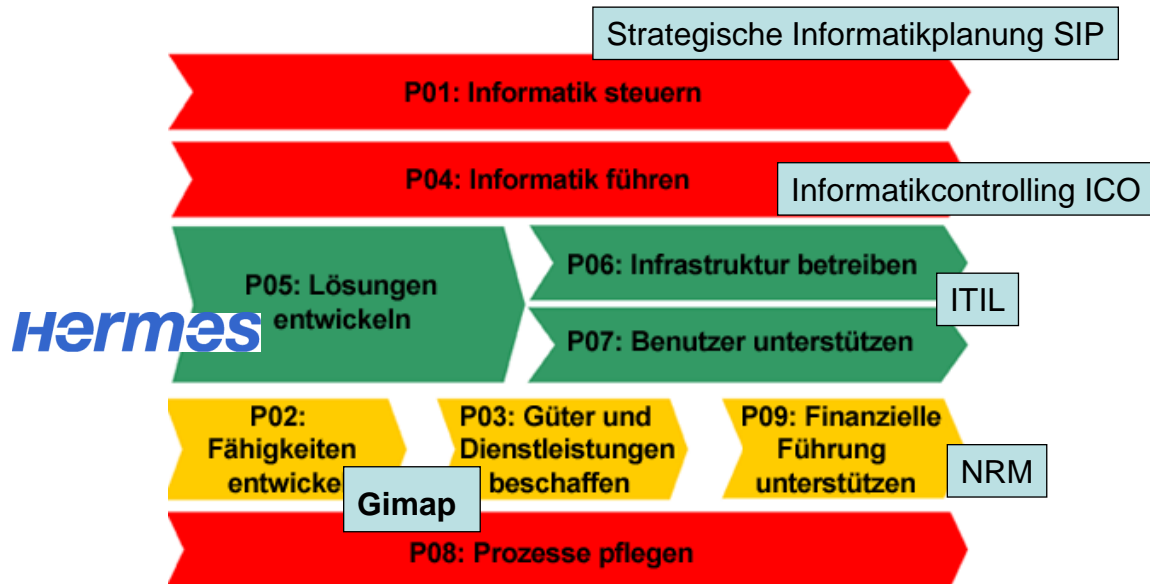
- Für wen ?**
- Multinationale, Grossunternehmen
 - 100 Projektleiter -> Zahlreiche Benützer
 - Organisiert Projektoffice
 - Ressourcen für die Einführung

- Wie ?**
- **Methode:** HERMES im eigenen Prozess integrieren
 - **Werkzeuge:** eingesetzt Kollaborationstool, Riskmanagement
 - **Organisation:** ein Projektleiter für die Einführung
 - **Ausbildung:** 1 Tag Ausbildung (HERMES Pur + HERMES bei uns)



Beispiel

- Prozesslandschaft bei der Bundesverwaltung



3. Die Erstellung der Methode

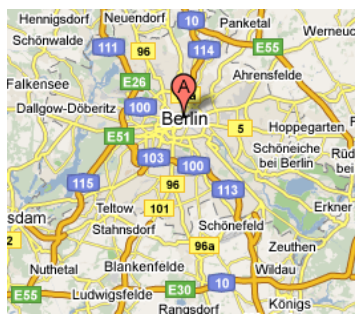
- Komplexität vs Vereinfachung
- Übersichtlichkeit vs Vollständigkeit
- Thematik vs Bereich



Komplexität der Umgebung -> vereinfacht darstellen



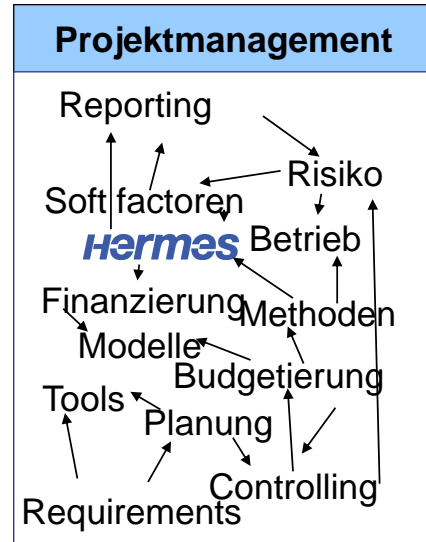
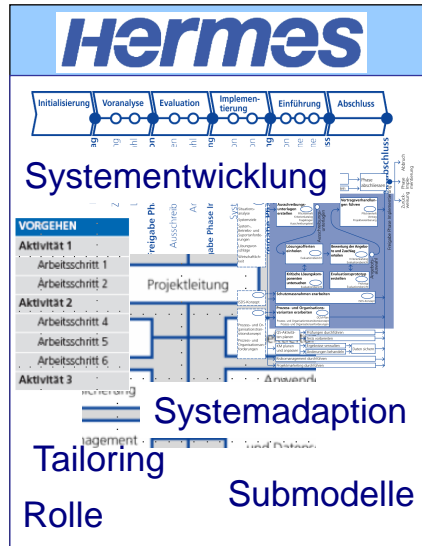
Übersichtliches Modell -> das alles machen kann



1.1. Die Einführung der Methode HERMES im Unternehmen



Die Thematik vollständig darstellen -> In die Umgebung integrieren



HERMES entwickeln: langfristig

Alle 8 bis 10 Jahre wird HERMES neu überarbeitet:

- 1975 – 1986 – 1995 – 2003

Neue Version 2012??

- Zusammen mit Benutzer und Experten
- Wichtige Trends im Projekt-Management erkennen
- Aktuelle Tools anbieten

ZIELE: Wettbewerbsfähig bleiben auf dem Methodik-Markt und die Stabilität garantieren.



HERMES entwickeln: kurzfristig

Jedes Jahr wird HERMES verbessert / ergänzt



Departemente / Bedürfnisse der Ämter

2009

- Integration / Vereinfachung von HERMES PowerUser
- Projektleitfaden mit Schnittstelle zu ICO / Cockpit



Allgemeine Bedürfnisse der Benutzer

HERMES Fachgruppe im Verein eCH -> Änderungsausschuss

2009

- HERMES & ITIL
- HERMES für Organisationsprojekte



Schlusswort



Einführung

Herstellung





Weitere Informationen

www.hermes.admin.ch

Hélène Mourgue d'Algue

Verantwortliche HERMES-Methode

Eidgenössisches Finanzdepartement EFD

Informatikstrategieorgan Bund ISB

Friedheimweg 14, 3003 Bern

Tel. +41 31 323 45 05

Fax +41 31 322 45 66

helene.mourguedalque@isb.admin.ch

1.2. V-Modell XT – ideale Ausgangsbasis für professionelle Prozesse in einer Vielzahl von Unternehmen

Doris Rauh, Dr. Winfried Rußwurm

CT SE 3

Siemens AG

Otto-Hahn-Ring 6

81739 München

{doris.rauh | russwurm}@siemens.com

Abstract

Das V-Modell XT, ein Prozessrahmenwerk, wurde von mehreren Universitäten und Firmen im Auftrag der Bundesministerien des Innern und der Verteidigung entwickelt, um einerseits den Erfolg von Entwicklungsprojekten zu erhöhen und andererseits das prozessorientierte Arbeiten in kleinen, mittleren und großen Unternehmen zu fördern. Das V-Modell XT umfasst dazu einen eigenen Projekttyp zur Einführung und Pflege eines organisationsspezifisch angepassten Prozesses. Zusätzlich steht zur Anpassung des V-Modell XT an verschiedenste Organisationen ein durch das Metamodell unterstütztes Verfahren zur Verfügung. Eine wesentliche Anforderung bei der Entwicklung des V-Modell XT war darüber hinaus die Implementierung des weltweit anerkannten Prozessmodells CMMI. Damit kann das V-Modell XT als schneller und kostengünstiger Startpunkt für die Definition von Prozessen verwendet werden, die potentiell bei entsprechender Anwendung den Reifegrad 3 des CMMI erreichen können.

1 Motivation

Häufig erreichen Entwicklungsprojekte ihre Ziele bzgl. Kosten, Zeit und Qualität nicht. Eine der Hauptursachen dafür ist das Fehlen eines ausgereiften Prozesses. Um diese Situation zu verbessern, wurde das V-Modell XT als Prozessrahmenwerk im Auftrag der Bundesministerien des Inneren und der Verteidigung entwickelt. Mit dem V-Modell XT [1] als Standard-Rahmenwerk sollen Firmen aller Größenordnungen, also auch kleinere und mittlere Unternehmen, in die Lage versetzt werden, sich daraus schnell und kostengünstig eigene reife Prozesse zu erstellen. Damit sind Vorgehensweisen klar definiert und sind über mehrere Unternehmen hinweg leichter vergleichbar. Die Kommunikation und das Verständnis zwischen Partnern (z.B. Auftraggeber und Auftragnehmer) werden deutlich erleichtert, Reibungsverluste werden verringert. Die Anforderungen einer Reihe wichtiger Prozessstandards, wie z.B. CMMI, werden abgedeckt. Damit steigt die Wettbewerbsfähigkeit dieser Unternehmen deutlich.

2 Das CMMI (Capability Maturity Model Integration)

Seit nunmehr fast 2 Jahrzehnten gibt es Reifegradmodelle für die Entwicklung. Daraus entstand vor 7 Jahren das derzeit aktuellste und am weitesten verbreitete Prozessmodell, das CMMI (Capability Maturity Model Integration), derzeit in der Version CMMI-DEV

1.2. V-Modell XT – Ausgangsbasis für professionelle Prozesse

v1.2 [2]. Dieses wird vom SEI (Software Engineering Institute) der Carnegie Mellon Universität in Pittsburgh herausgegeben. Zur Produktfamilie des CMMI gehören auch Bewertungsverfahren, wie z.B. das SCAMPI (Standard CMMI Appraisal Method for Process Improvement) [3]. CMMI als Prozessmodell stellt eine in Prozessgebiete, Ziele und Praktiken strukturierte Sammlung von Themen dar, die sich in Theorie und Praxis als hilfreich und wirksam für eine professionelle Produktentwicklung (HW, SW, System) zur Vermeidung von Fehlern und zur Erreichung von wichtigen Projektzielen erwiesen hat. Das Modell stellt Eigenschaften bzw. Anforderungen an wirksame Prozesse vor, es ist aber kein Prozess „von der Stange“. Weiterhin werden Wege zur Prozessverbesserung aufgezeigt, die je nach den Zielrichtungen des Unternehmens eingeschlagen werden können. Es obliegt den jeweiligen Unternehmen, sich aus dem CMMI einen eigenen dokumentierten Prozess abzuleiten, der die spezifischen Bedürfnisse und Eigenheiten berücksichtigt. In einem SCAMPI Appraisal werden dann die dokumentierten und gelebten Entwicklungspraktiken an den Themen des CMMI gespiegelt und daraus der Fortschritt der Prozessverbesserung und evtl. auch ein sich daraus ergebender Reifegrad ermittelt.

3 Der Zusammenhang zwischen V-Modell XT und CMMI

CMMI hat in der Zwischenzeit einen hohen Stellenwert bei der Bewertung von Entwicklungsprozessen erreicht und es werden weltweit von Auftraggebern zunehmend Reifegradforderungen gestellt. Gleichzeitig sollte das V-Modell XT eine ideale und kostengünstige Ausgangsbasis für professionelle Prozesse sein. Es war daher eine logische Anforderung an das V-Modell XT, sich auch an CMMI zu orientieren. Das V-Modell XT hat deshalb die Praktiken und Ziele des CMMI bis zur Erreichung des Reifegrades 3 berücksichtigt, soweit dies im Rahmen eines Prozessrahmenwerks sinnvoll war. Die Erfüllung der Anforderung wurde mit Hilfe einer „virtuellen“ Prozessbewertung in Form einer Diplomarbeit bewertet. Die sich daraus ergebenden Befunde wurden in Änderungsvorschläge umgesetzt und zur Verbesserung des V-Modell XT verwendet. Damit unterstützt es jetzt in eindeutiger Weise die Definition von Prozessen in Unternehmen, die potenziell in der Lage sind, einen CMMI Reifegrad 3 zu erreichen. Dies ist allerdings kein Automatismus, die jeweilige Organisation ist immer noch dafür zuständig, das V-Modell XT für seine Bedürfnisse anzupassen, zu ergänzen und vor allem in der gesamten Organisation zur Anwendung zu bringen. Ein dokumentierter Prozess alleine kann noch keinen Reifegrad erreichen, es zählt immer der entsprechende Einsatz in der Praxis.

4 Anpassung des V-Modell XT an Organisationen

Die Anpassung des V-Modell XT an organisationsspezifische Gegebenheiten und die Einführung und Pflege dieser Prozesse trägt wesentlich zum Erfolg von Projekten bei. Dies wurde bei der Entwicklung des V-Modell XT frühzeitig erkannt. Das V-Modell XT umfasst deshalb einen eigenen Projekttyp zur Einführung und Pflege eines organisationsspezifisch angepassten Prozesses. Der V-Modell XT Editor (Open Source Tool), der bereits zur Entwicklung des V-Modell XT verwendet wurde, steht für spezifische Anpassungen an Organisationen zur Verfügung. Die Anpassung des V-Modell XT an verschiedenste Organisationen wird seit kurzem durch das Metamodell unterstützt. Dies war notwendig, da sich sehr schnell das Aktualisieren von organisationsspezifischen Anpassungen als mühsam herausgestellt hat.

Spielt für die Organisation CMMI eine wesentliche Rolle, ist dies kein Problem, da sich das V-Modell XT stark auf CMMI bezieht. Es ist daher sehr einfach möglich die Arbeitsprodukte des V-Modell XT den Praktiken des CMMI zuzuordnen. Zur Erleichterung eines späteren SCAMPI Appraisal, können damit ohne zu großen Aufwand, parallel zur Anpassung des V-Modell XT an die Organisation, sogenannte PIIDs (Practice Implementation Indicator Descriptions) erstellt werden. Aufgabe dieser PIIDs ist es den Praktiken des CMMI direkte und indirekte Artefakte zuzuordnen, die die Erfüllung der jeweiligen Praktik belegen. Ein weiterer Vorteil der Erstellung von PIIDs parallel zur Prozessdefinition ist die frühzeitige Erkennung von Schwachstellen bzgl. der betrachteten Prozesse im Hinblick auf das gesteckte Verbesserungsziel.

Das V-Modell XT ist damit eine ideale Ausgangsbasis für Organisationen, um sehr schnell und vor allem auch sehr kostengünstig zu einem reifen Prozess zu kommen.

Literaturverzeichnis

1. V-Modell®XT, <http://www.v-modell-xt.de>, Koordinierungs- und Beratungsstelle der Bundesregierung, 2008
2. CMMI for Development V1.2, Technical Report CMU/SEI-2006-TR-08, Software Engineering Institute, Carnegie Mellon University, 2008, <http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html>
3. Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.2: Method Description Document, Handbook CMU/SEI-2006-HB-002, Software Engineering Institute, Carnegie Mellon University, 2008, <http://www.sei.cmu.edu/publications/documents/06.reports/06hb002.html>

Corporate Technology

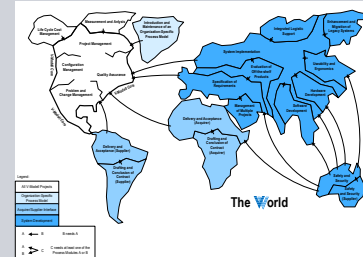


V-Modell® XT

Ideale Ausgangsbasis für professionelle Prozesse in einer Vielzahl von Unternehmen

Doris Rauh
Dr. Winfried Rußwurm
Siemens AG, CT SE 3, München

SEE
Berlin, Mai 2009



Copyright © Siemens AG 2009. All rights reserved.

V-Modell® XT

Ideale Ausgangsbasis für professionelle Prozesse in einer Vielzahl von Unternehmen



V-Modell - Motivation, Historie und Anforderungen
Organisationsspezifische Prozesse auf Basis des V-Modell XT
CMMI®-DEV und V-Modell XT
Anwendungsbeispiel

© CMM, CMMI, Capability Maturity Modeling, CERT, and CERT Coordination Centre are trademarks copyrighted of the Carnegie Mellon University, Pittsburgh, Pennsylvania (USA), registered in the U.S. Patent and Trademark Office

SM CMM Integration, SCAMPI, SCAMPI Lead Assessor, SCAMPI Lead Appraiser, IDEAL, SEI, SEPG, Interim Profile, Personal Software Process, PSP, SCE, Team Software Process, and TSP are service marks of the Carnegie Mellon University

TM Carnegie Mellon Software Engineering Institute design and stylized hexagon are trademarks of the Carnegie Mellon University

V-Modell – Motivation und Historie

- Problem
 - Die Zahl der gescheiterten oder wirtschaftlich nicht erfolgreichen Projekte ist zu hoch. Gründe dafür sind häufig unsystematische Vorgehensweisen
 - Bei unternehmensübergreifenden Projekten (z.B. in Konsortialprojekten) oder bei ständig wechselnden Organisationsstrukturen entstehen Reibungsverluste durch unterschiedliche Prozesse.
- V-Modell
 - Entwickelt im Auftrag der Bundesministerien der Verteidigung und des Inneren um diesen Problemen entgegenzuwirken.
 - Verbindlicher Entwicklungsstandard für IT-Systeme des Bundes (Basis für Ausschreibungen)
 - Erste Versionen: V-Modell 92 und V-Modell 97
 - Aktuelle Version: V-Modell XT
 - Aktuelles Release 1.3 (Feb 2009)
 - Entwicklung: WEIT Project, Weiterentwicklung und Pflege: WEIT e.V.
 - Kostenfrei verfügbar in Deutsch und Englisch
 - Freigegeben unter der Apache Lizenz Version 2.0

Anforderungen an die Weiterentwicklung des V-Modells

- Strukturelle Anforderungen (bessere Anwendbarkeit, Anpassbarkeit, Skalierbarkeit, Änder- und Erweiterbarkeit)
- Inhaltliche Anforderungen
 - Intensivierung der Managementaspekte
 - Systementwicklungsstandard inklusive Hardware und Logistik
 - Abdeckung des gesamten Produktlebenszyklus
 - Organisationsweite Themen, z.B. organisationsspezifische Anpassung
- Weitergehende Anforderungen
 - Anwendungshilfen (Beispieldokumente, Tailoringtool, etc.)
 - Integration von (Quasi-) Standards, Normen und Vorschriften (z.B. CMMI-DEV)
 - Englische Version, Hypertext, Anwender-Plattform, etc.

Organisationsspezifische Prozesse

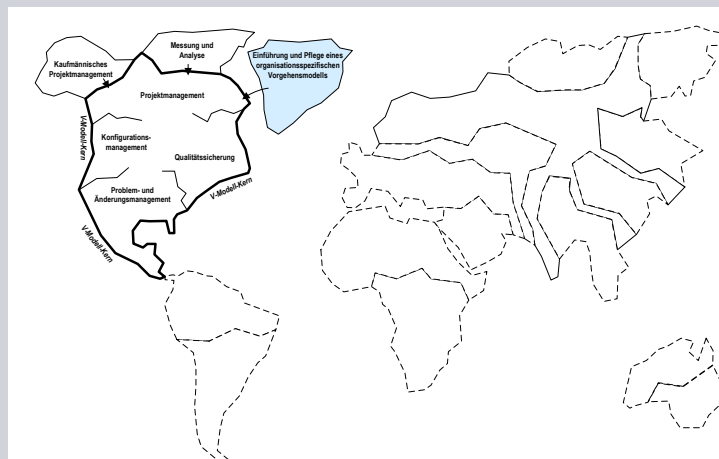
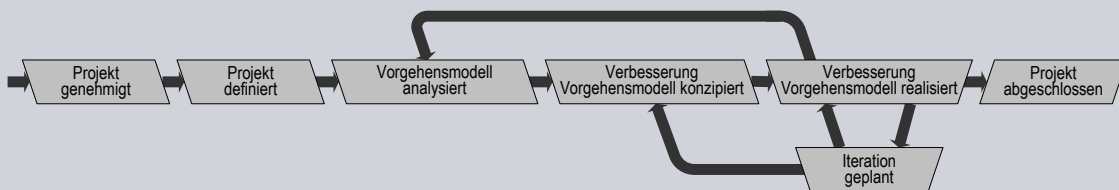
Vorteile von organisationsspezifischen Prozessen

- Gemeinsamkeiten über Projekte hinweg
- Klar geregelte und erprobte Vorgehensweisen, klare Verantwortlichkeiten
- Gemeinsames Verständnis, gemeinsame Sprache
- Breite Erfahrungsbasis, Nutzung organisationsweiter Daten

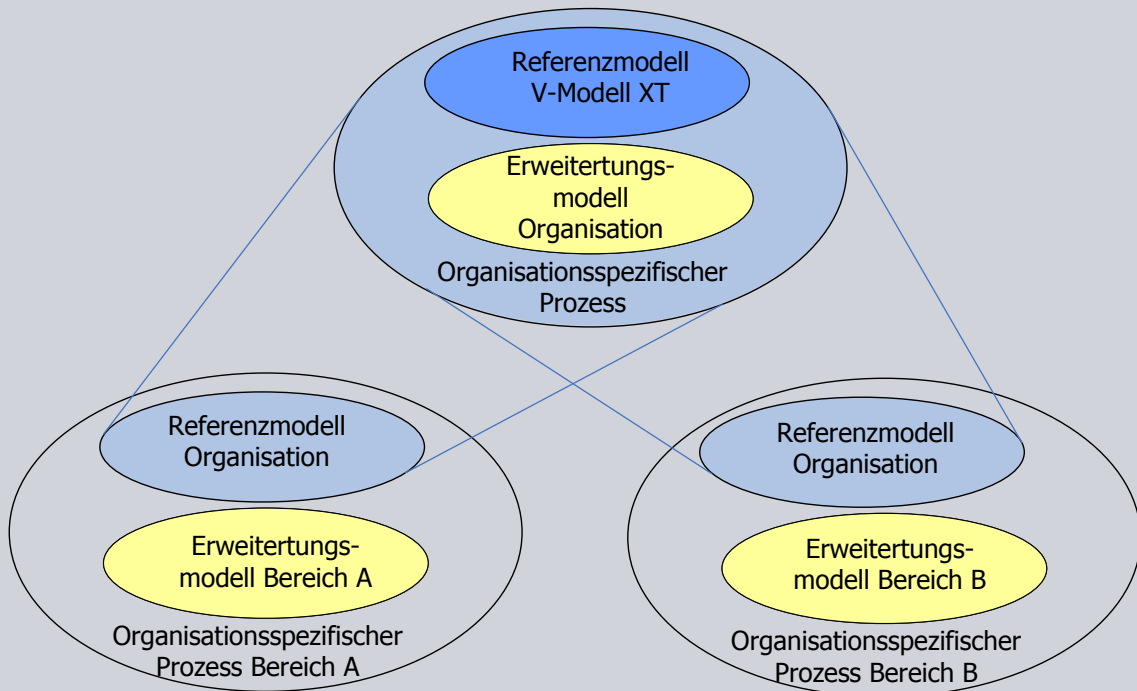
Organisationsspezifische Anpassung und Einführung des V-Modell XT

- Bewertung des Status Quo der Prozesse der Organisation
- Definition der Anforderungen an die Prozessverbesserung
- Prozessdefinition auf Basis des V-Modell XT
 - Organisationsspezifisches Tailoring
 - Organisationsspezifische Änderungen und Erweiterungen
 - Ankopplung an bereits vorhandene Geschäftsprozesse
 - Festlegung von organisationsweiten Vorgaben, Methoden, Tools, ...
 - Berücksichtigung von Standards
- Pilotierung und Breitereinführung

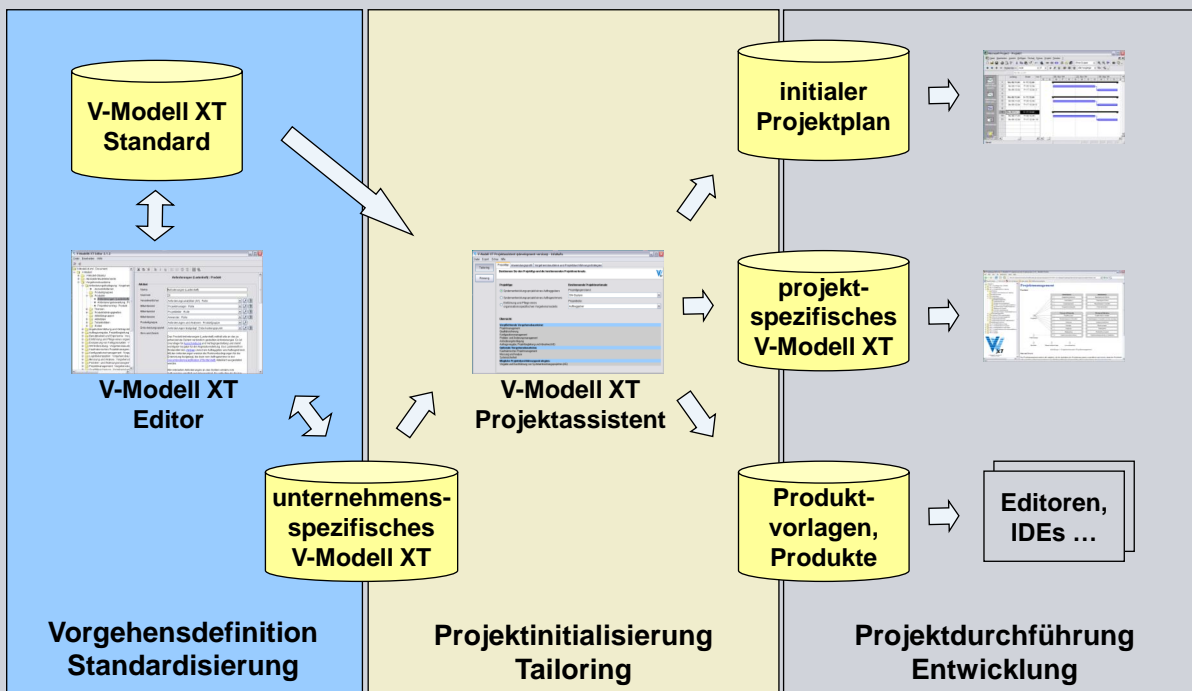
Projekttyp: Einführung und Pflege eines organisationsspezifischen Vorgehensmodells



Metamodellunterstützung für organisationsspezifische Anpassungen



Open-Source-Werkzeuge des V-Modell XT

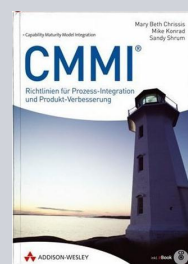
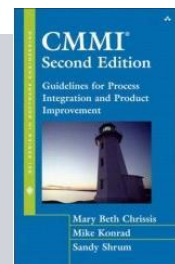


Anforderungen an die Weiterentwicklung des V-Modells

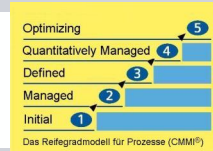
- Strukturelle Anforderungen (bessere Anwendbarkeit, Anpassbarkeit, Skalierbarkeit, Änder- und Erweiterbarkeit)
- Inhaltliche Anforderungen
 - Intensivierung der Managementaspekte
 - Systementwicklungsstandard inklusive Hardware und Logistik
 - Abdeckung des gesamten Produktlebenszyklus
 - Organisationsweite Themen, z.B. organisationsspezifische Anpassung
- Weitergehende Anforderungen
 - Anwendungshilfen (Beispieldokumente, Tailoringtool, etc.)
 - Integration von (Quasi-) Standards, Normen und Vorschriften (z.B. CMMI-DEV)
 - Englische Version, Hypertext, Anwender-Plattform, etc.

Was ist CMMI®?

- CMMI® (Capability Maturity Model Integration) besteht aus mehreren Teilen, die zur Verbesserung der Prozesse bzw. Arbeitsabläufe in einer Organisation bzw. deren Projekte verwendet werden:
 - Prozessmodell
 - Bewertungsverfahren (Appraisal-Verfahren, SCAMPI = Standard CMMI Appraisal Method for Process Improvement)
 - Kurse (z.B. Introduction to CMMI)
- Pflege durch SEI (Software Engineering Institute) der Carnegie Mellon University in Pittsburgh, PA
- Es gibt inzwischen 3 domänenorientierte Prozessmodelle:
 - Entwicklung: CMMI-DEV (aktuelle Version v1.2)
 - Service: CMMI-SVC
 - Beschaffung: CMMI-ACQ
- Die nächste Version CMMI-DEV v1.3 ist für 2010 vorgesehen
- Seit Mai 2009 gibt es CMMI-DEV v1.2 auch in einer vom SEI freigegebenen deutschen Übersetzung



Wozu dient CMMI®?



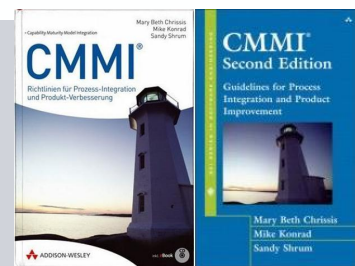
SIEMENS

- Leitfaden/Modell für die Verbesserung der Entwicklungsprozesse in Definition (Beschreibung) und Anwendung (tägliche Praxis)
- Ziele:
 - Bessere Erreichung von Geschäftszielen wie z.B. Kosten, Termine, Qualität, Funktionsumfang durch einen transparenteren Entwicklungsablauf
 - Vermeidung/Reduzierung von Kostenüberschreitungen, Terminverzügen, Panikaktionen, Kundenunzufriedenheit, Nacharbeiten, Qualitätsproblemen, schlechter Mitarbeitermoral etc.
- Vorgehen:
 - Spiegelung der aktuellen Praktiken am Prozessmodell durch Appraiser (Assessoren) mit Hilfe eines dafür geeigneten Verfahrens (z.B. SCAMPI A, Siemens Prozess Assessment).
 - Reifegradmodell: Zuordnung der Prozessgebiete zu Reifegraden
 - Ergebnis eines Appraisals: meist „Reifegrad“ des Prozesses
 - Schrittweise Verbesserung unreifer Arbeitsweisen hin zu reifen Prozessen durch sukzessive Einführung von Prozessgebieten, z.B. Projektplanung, QS, Organisationsweite Aus- und Weiterbildung.

Verbreitung des CMMI® (Stand Dezember 2008)

SIEMENS

- Es wurden bisher ca. 98000 Personen weltweit offiziell in CMMI geschult
- Es gibt pro Jahr weltweit ca. 1000 SCAMPI A Appraisals (in Summe bisher ca. 4100)
- Die meisten Appraisals werden in USA (ca. 1300), China (ca. 750) und Indien (ca. 400) durchgeführt.
- SCAMPI A Appraisals wurden bisher in 67 Ländern durchgeführt.
- Deutschland ist mit 64 SCAMPI A Appraisals bisher noch hinter Frankreich, Spanien, Japan, UK und Korea, aber gleichauf mit Argentinien und etwas vor Ägypten und Malaysia.
- Es gibt weltweit ca. 500 SCAMPI A Lead Appraiser und ca. 430 Instruktoren für „Introduction to CMMI“.



V-Modell XT und CMMI®-DEV v1.2

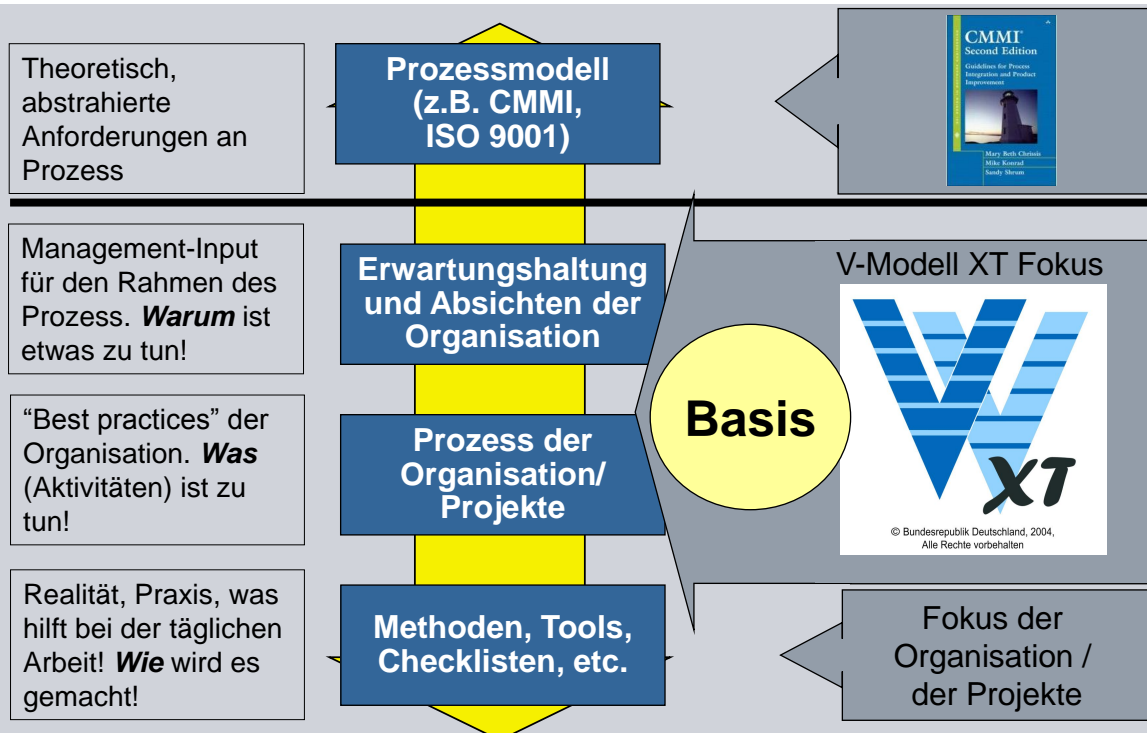
Ist das das gleiche oder ist es konkurrierend?

- CMMI-DEV v1.2: Prozessmodell, das die Eigenschaften reifer Prozesse beschreibt (für Disziplinen SE, SW, HW)
- V-Modell XT: Prozess Rahmenwerk als Ausgangsbasis für eine schnelle Erarbeitung eines organisationsweiten Prozesses
Inhalte: Produkte, Aktivitäten, Rollen, Projektdurchführungsstrategien, Produktvorlagen, ...

Anforderung an V-Modell XT: Berücksichtigung der CMMI-DEV Praktiken bis zum Reifegrad 3

- Beschreibung der Umsetzung dieser Praktiken incl. einiger Einschränkungen in Form einer Konventionsabbildung für CMMI
- “Virtuelle” Prozessbewertung:
 - Das V-Modell XT als “dokumentierter Standardprozess” wurde gegen CMMI-DEV mit Hilfe der Diplomarbeit von Michael Kranz bewertet
 - Beseitigung der erkannten Lücken in weiteren V-Modell XT Versionen

Prozess und Prozessmodell



Wie unterstützt V-Modell XT Appraisals nach CMMI®-DEV v1.2?

SIEMENS

PIID's (Practice Implementation Indicator Descriptions): Eine weit etablierte Praxis zur Nachweisführung bei SCAMPI Appraisals

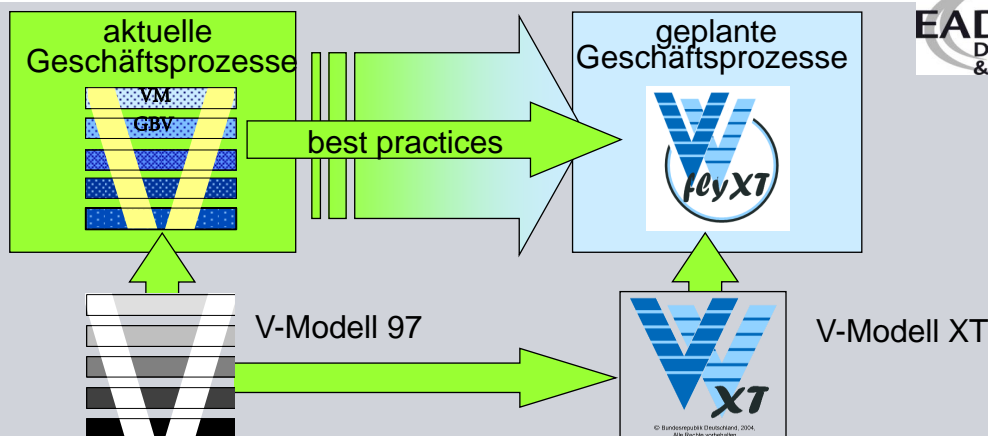
- Abbildung der Nachweise (z.B. direkte und indirekte Artefakte) auf die Praktiken des CMMI nach Projekten
- Prüfung und Bewertung der Nachweise durch die Appraiser (zusammen mit Informationen aus Interviews)
- ≥ 4000 Nachweise bei SCAMPI A mit Ziel Reifegrad 3 und 3 Projekten
- Hoher Aufwand die PIID's mit passenden Artefakten zu füllen

V-Modell XT:

- Gute Unterstützung zur Identifizierung/Abbildung der Nachweise
- Geringerer Aufwand: PIID's können in rationeller Weise (da jedes Projekt Artefakte in gleicher Weise erzeugt) befüllt werden
- Einem Reifegrad 3 bei einem Appraisal steht nichts im Wege. Regelungen, die V-Modell XT nicht bieten kann, wie z.B. Tools und Methoden sind zu ergänzen
- Voraussetzung: vollständige und sorgfältige Durchführung (kein „Papiertiger“).

Beispiel für eine V-Modell XT Anpassung: Die Geschäftsprozesse der EADS DE

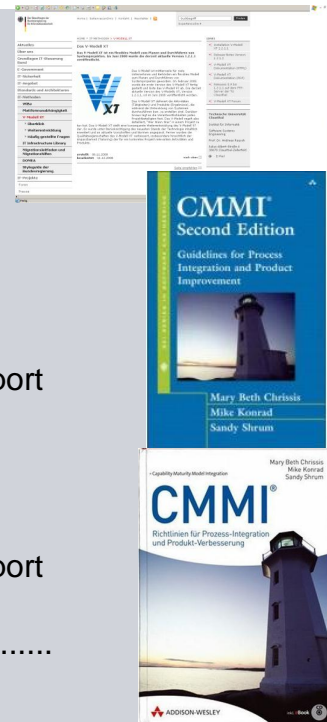
SIEMENS



- flyXT:
 - Basiert auf dem V-Modell XT Metamodell
 - Sukzessive Migration der best practices der aktuellen Geschäftsprozesse der EADS DE und dadurch teilweise Ersetzung des V-Modell XT Inhaltes.
 - V-Modell XT Konformität (Abbildung zwischen V-Modell XT und flyXT)
- Einbeziehung von Prozess Standards:
 - CMMI: Erstellung von PIIDs mit typischen Produkten des flyXT
 - Luftfahrt-Standards: DO-178B, DO-254

Wo finde ich noch mehr Informationen?

- Aktuelle Information zum V-Modell XT, sowie Downloads (V-Modell in Deutsch und Englisch): <http://www.v-modell-xt.de>
- Siemens V-Modell XT Website: <https://www.ct.siemens.com/de/technologies/se/beispiele/vmodell.html>
- CMMI-Einstiegsseite des SEI: <http://www.sei.cmu.edu/cmmi/>
- CMMI-DEV-Website und Downloads des SEI: <http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html>
- SEI-Website für übersetzte Versionen des CMMI: <http://www.sei.cmu.edu/cmmi/translations/>
- Website und Downloads für SCAMPI A Verfahren: <http://www.sei.cmu.edu/publications/documents/06.reports/06hb002.html>
- Oder einfach die Autoren dieses Beitrags kontaktieren.....



Vielen Dank für Ihre Aufmerksamkeit!



Doris Rauh

doris.rauh@siemens.com

Tel: +49-89-636-53393

Siemens AG, CT SE 3

Otto-Hahn-Ring 6

81739 München

Fax +49-89-636-44424



Dr. Winfried Rußwurm

russwurm@siemens.com

Tel: +49-89-636-42627

Siemens AG, CT SE 3

Otto-Hahn-Ring 6

81739 München

Fax +49-89-636-44424

1.3. Das Vorgehensmodell Bayern im neuen Gewand

Christian Schober, Dr. Andreas Mück
Zentrale IuK-Leitstelle
Bayer. Staatsministerium des Innern
Odeonsplatz 3
D-80539 München
{christian.schober | andreas.mueck}@stmi.bayern.de

Abstract

Das V-Modell XT wurde bereits 2005 für die bayer. Staatsverwaltung organisationspezifisch angepasst und das daraus resultierende V-Modell XT Bayern in der Version 1.0 den Staatsbehörden als unverbindliches Hilfsmittel zur richtlinienkonformen Durchführung von Auftraggeberprojekten im Bereich der Softwareentwicklung und -anpassung zur Verfügung gestellt. In die nunmehr aktuelle Version 1.3 floss die Umstellung auf die neue Metadatenstruktur des V-Modell XT ebenso ein, wie weitere organisationspezifische Anpassungen an mittlerweile neu erlassene, verbindliche Richtlinien im Umfeld der Softwareentwicklung.

1 Ausgangslage

Mit der raschen Verbreitung des Internets sind die Erwartungen der Bürgerinnen und Bürger sowie der Wirtschaft an die Online-Fähigkeit der Verwaltung deutlich gestiegen. Der Schlüssel für noch mehr elektronische Dienste der öffentlichen Hand ist eine leistungsfähige Informations- und Kommunikationstechnik (IuK-Technik), die sich zu einem nicht unerheblichen Teil an der Güte der eingesetzten Softwareprodukte messen lassen muss. Der Qualität von Softwareentwicklungsprojekten kommt deswegen für die Realisierung der eGovernment-Ziele der Bayer. Staatsregierung eine ganz besondere Bedeutung zu.

Zur Umsetzung eines Kabinettsbeschlusses zur stärkeren Vereinheitlichung der IuK-Technik wurden seit 2004 für alle bayer. Staatsbehörden verbindliche Richtlinien erlassen, die insbesondere auch das Vorgehen in Softwareentwicklungsprojekten regeln. Dadurch soll die Qualität der Abwicklung von Projekten insbesondere durch die Einhaltung von Mindeststandards und eine geregelte Kommunikation der Projektbeteiligten auf der Grundlage einer einheitlichen Terminologie erhöht werden. Die bislang erlassenen Richtlinien reglementieren im Bereich der Softwareentwicklung die Kernbereiche Wirtschaftlichkeit, Projektmanagement, Qualitätssicherung, Konfigurationsmanagement, Modellierungssprachen sowie Problem- und Änderungsmanagement. Zahlreiche weitere Richtlinien wie zum für Betriebs- und Datenbanksysteme, Arbeitsplatzsysteme oder XML-Standards für den Austausch von Fachdaten wirken sich mittelbar auf die Anforderungen an Softwaresysteme ebenso aus wie diverse IT-Sicherheitsrichtlinien.

Die mit der Entwicklung von Softwaresystemen befassten Behörden stehen somit vor einer Vielzahl von Richtlinien, die es nun für ihre Projekte und zu entwickelnden Systeme einzuhalten gilt.

2 Das V-Modell XT Bayern

Um der damit verbundenen Komplexität Rechnung zu tragen, wurde zusätzlich zur den verbindlichen Richtlinien das V-Modell XT Bayern als organisationspezifische Anpassung des V-Modell XT erarbeitet und der staatlichen Verwaltung als unverbindliches Hilfsmittel zur richtlinienkonformen Durchführung von Softwareentwicklungsprojekten zur Verfügung gestellt. In der Version 1.0 beinhaltet das V-Modell XT Bayern seit 2005 die Projektdurchführungsstrategie für Auftraggeber „Vergabe und Durchführung von IT-Entwicklungsprojekten“, die auf die bis zu diesem Zeitpunkt vorhandenen Richtlinien angepasst war.

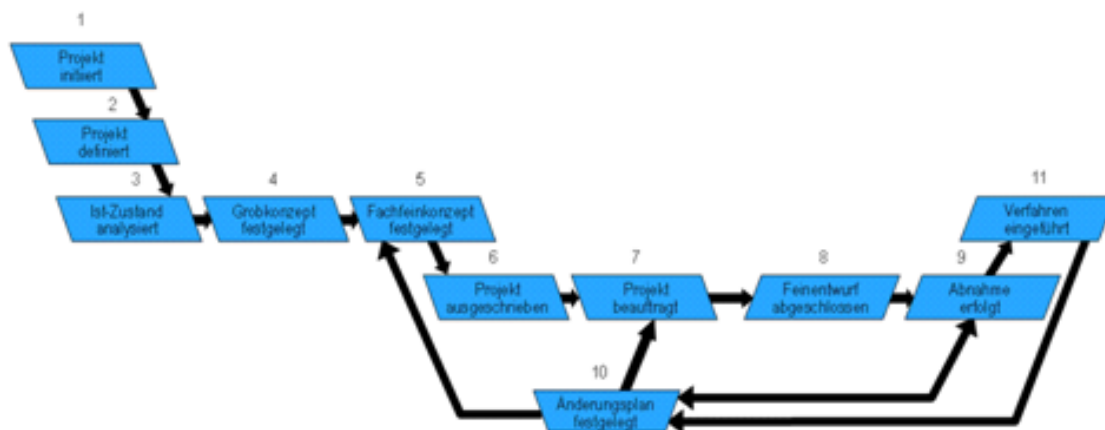


Abbildung 1.1.: Projektdurchführungsstrategie: Vergabe und Durchführung von IT-Entwicklungsprojekten

2.1 Erfahrungen mit der Version 1.0

Obgleich die Behörden das V-Modell XT Bayern als praxisorientierten Leitfaden durch das Dickicht von einschlägigen Richtlinien akzeptierten, wurden - neben der schon nahezu sprichwörtlichen „Dokumentenlastigkeit“ des V-Modells - folgende Punkte kritisiert:

- Die mit dem V-Modell XT in der Version 1.0 ausgelieferten Installationsroutinen erwiesen sich insbesondere für organisationspezifische Anpassungen als wenig benutzerfreundlich.
- Der Projektassistent eignete sich kaum für ein feinkörniges, projektspezifisches Tailoring.
- Die seit 2005 neu erlassenen Richtlinien wurden vom V-Modell XT Bayern in der Version 1.0 nicht abgedeckt.
- Die Einführung des V-Modell XT Bayern wurde nicht durch ein entsprechendes Schulungskonzept unterstützt. Ferner erwies sich die Metadatenstruktur der ersten Version des V-Modell XT als wenig geeignet für die laufende Fortschreibung einer organisationspezifischen Anpassung, insbesondere hinsichtlich einer automatisierten Übernahme von Änderungen im Referenzmodell.

2.2 Die Version 1.3

Die in der Bayer. Staatsverwaltung für strategische IT-Steuerung verantwortliche Zentrale IuK-Leitstelle setzte sich für die Version 1.3 das Ziel, die vorgenannten Defizite des V-Modell XT Bayern zu beseitigen. Um die mittlerweile für das V-Modell XT stark verbesserten Installationsroutinen und Werkzeuge wie z.B. den Projektassistenten auch mit dem V-Modell XT Bayern nutzen zu können, wurde zunächst mit Unterstützung der TU Clausthal (Lehrstuhl Prof. Dr. Rausch) eine Migration des V-Modell XT Bayern auf die neue Metadatenstruktur des V-Modell XT durchgeführt.

Der Umstieg auf die neue Metadatenstruktur erlaubt nun auch eine automatisierte Übernahme von Änderungen am V-Modell XT in das V-Modell XT Bayern. Hierfür wird die an der TU Clausthal für das V-Modell XT bereitgestellte Produktions- und Änderungs-umgebung genutzt. Mittelfristig ist geplant, auch die Nutzer des V-Modell XT Bayern in den Änderungsprozess mit einzubeziehen.

Seit 2005 neu erlassene Richtlinien mit Auswirkung auf das Projektvorgehen wurden als inhaltliche Änderung bzw. Ergänzung und Themenbeschreibungen in das V-Modell XT Bayern eingearbeitet. Richtlinien, die sich auf die Anforderungen des zu entwickelnden Systems auswirken bzw. hier Vorfestlegungen treffen, wurden als Mustertexte implementiert. Die Übernahme dieser Mustertexte in entsprechende Produkte des V-Modell XT Bayern kann flexibel über den Projektassistenten gesteuert werden. Einen Eindruck davon vermittelt der nachstehende Screen Shot.

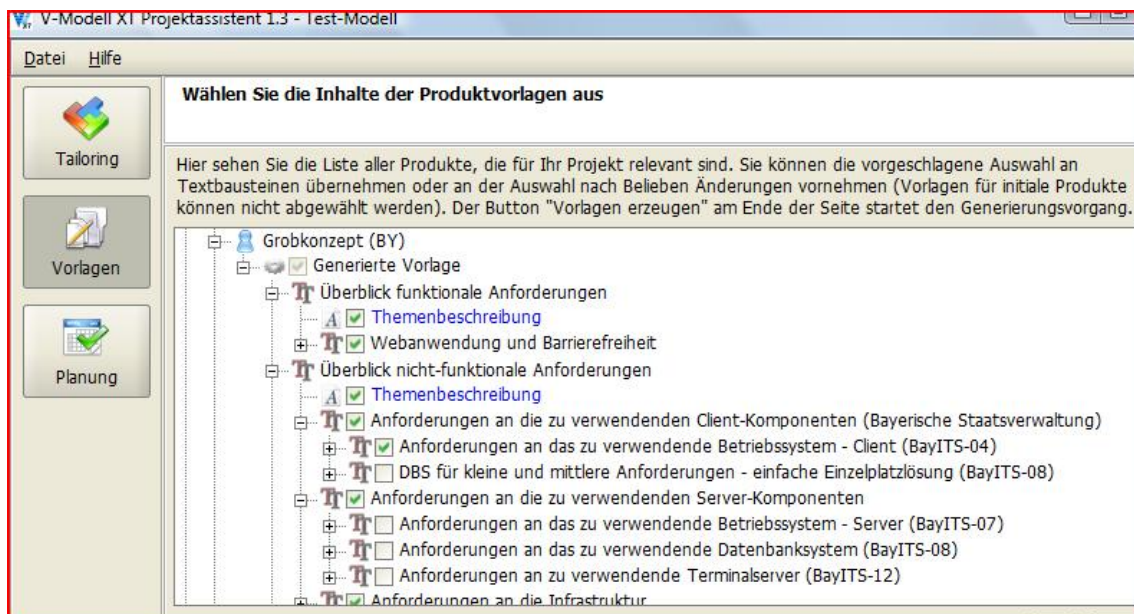


Abbildung 1.2.: Inhaltliche Ausgestaltung

Im Rahmen der Qualifizierungsoffensiven für die Staatsverwaltung wird derzeit mit internen Fortbildungseinrichtungen an einem Schulungskonzept gearbeitet. Entsprechende Schulungsmaßnahmen sollen dort angeboten werden.

3 Ausblick

In der bayer. Staatsverwaltung soll der gesamte IT-Betrieb bis 2012 in zwei staatlichen Rechenzentren konzentriert werden. Dies hat auch Auswirkungen auf die Entwicklung von Softwaresystemen. Einerseits werden mit der Anforderungsanalyse bereits Festlegungen für den Betrieb getroffen (z.B. Performance, Verfügbarkeit), andererseits müssen die Rahmenbedingungen für den Betrieb in einem Rechenzentrum (z.B. strategische RZ-Plattformen) bei der Entwicklung der Systeme berücksichtigt werden. Daraus ergeben sich auch Schnittstellen der Softwareentwicklungsprozesse mit denen des IT-Betriebs. Diese Schnittstellen sollen mittelfristig auch im V-Modell XT Bayern abgebildet werden.



Das V-Modell XT Bayern Version 1.3



Christian Schober, Dr. Andreas Mück
{andreas . mueck, christian . schober} (at) stmi . bayern . de

SEE 2009
Seite 1



Gliederung

- Ausgangssituation
- Warum V-Modell Bayern?
- V-Modell Bayern, Version 1.0
- Defizite der Version 1.0
- V-Modell Bayern, Version 1.3
- Ausblick: Integration Entwicklung/Betrieb

SEE 2009
Seite 2





Ausgangssituation - Heterogene IT-Landschaft -

- ca. 1200 IT-Betriebszentren
- ca. 5000 Server
- ca. 110.000 Arbeitsplatzrechner (PCs)
- über 2000 Fachanwendungen
-



Ausgangssituation - Maßnahmen -

Konsolidierung der IT-Landschaft durch

- Konsolidierung der Betriebsstätten in 2 RZen
- Erarbeitung und Fortschreibung IT-Strategie
- Genehmigung von IT-Vorhaben
- IT-Controlling
- Festlegung von Standards und Richtlinien





Ausgangssituation

- Übersicht Richtlinien-

- **IT-Sicherheitsleitlinie**
- **IT-Sicherheitsorganisation**
- IT-Sicherheitsrahmenrichtlinie für BayKom-Daten
- Koppelung der VPN
- Betrieb eines Übergangs in das Internet
- Einsatz drahtloser Netzwerke
- Betrieb von IP-VPNs
- **Telearbeits- und mobile Arbeitsplätze**
- **Fernwartung**
- Einsatz mobiler Geräte
- **Penetrationstests**
- Extranet-/Dienstleister-VPN
- Wählverbindungen im Bayerischen Behördennetz
- **Nutzung von Anwendungen über das Internet**
- **Anzeige von IT-Vorhaben**
- **Durchführung von IuK-Projekten**
- Planungsrichtlinien für Kommunikationsnetze
- Rahmenrichtlinie für die Betriebsdienstleistungen der staatlichen Rechenzentren
- Richtlinie für die Nutzung von Internet und E-Mail in der bayerischen Staatsverwaltung
- **Softwarekonfigurationsmanagement**
- **Wirtschaftlichkeitsbetrachtungen**
- **Barrierefreiheit (BayBITV)**

SEE 2009
Seite 5



Ausgangssituation

- Überblick Standards-

- **Austausch von Dokumenten**
- **Webanwendungen**
- **Client-Betriebssysteme**
- E-Mail-Clients
- **Webbrowser**
- **Server-Betriebssysteme**
- **Datenbanksysteme**
- **Office-Pakete**
- Austausch von Dokumenten und Vorgängen zwischen Dokumentenmanagementsystemen
- PDF-Konverter
- **PC-Arbeitsplätze**
- **Terminalserver**
- Virens Scanner
- Softwareverteilung
- **Projektmanagementsysteme**
- Aktive Netzwerkkomponenten
- **Werkzeuggestützte Modellierungssprachen**
- **Verzeichnisdienste**

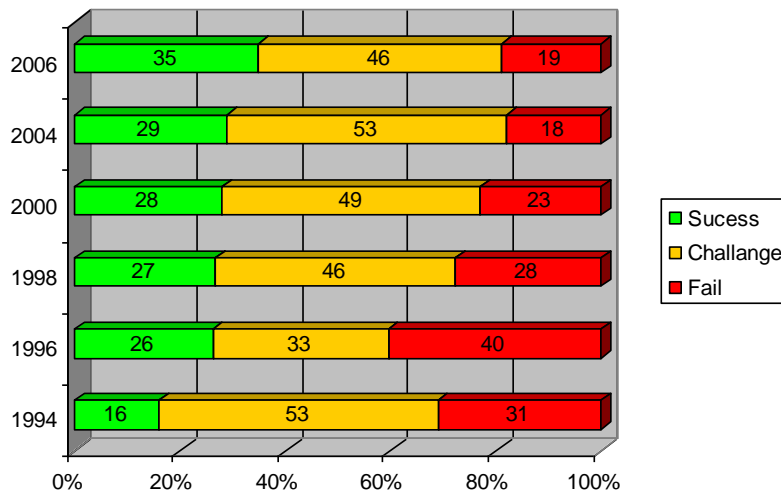
SEE 2009
Seite 6





Ausgangssituation

Chaos Report, Standish Group



SEE 2009
Seite 7



Warum V-Modell Bayern?

- Hilfsmittel zur richtlinienkonformen Projektdurchführung
- Zusätzliche Hilfestellung
 - Komplexität SW-Architekturen (Java, .net)
 - Komplexe IT-Architekturen (SOA)
 - hoher Integrationsgrad von Anwendungen
 - organisatorische Trennung von Entwicklung und Betrieb
- Integration der Fach-, Entwicklungs- und Betriebsaspekte
- V-Modell XT ist Standardmodell des Bundes

SEE 2009
Seite 8





Defizite der Version 1.0

- deckt mit PDS nur Projektrichtlinie ab
- unzureichende Installationsroutinen
- geringer Nutzen des Projektassistenten
 - projektspez. Tailoring kaum möglich
- beruht auf altem Metamodell
- manuelle Anpassung an inhaltliche Fortschreibung des Muttermodells
- keine Schulung



V-Modell Bayern, Version 1.3

- Neuerungen im Überblick -

- Neue Projektmerkmale zu Datenschutz und IT-Vorhaben
- Projekttypvariante mit verfeinerter Anforderungsfestlegung
- Einbindung von Richtlinien und Standards
- Konventionsabbildung für Konfigurationsmgmt.
- Kopiervorlagen für Verfahrensbeschreibung, Grobkonzept und EVB-IT-Verträge
- Mustertexte für Grobkonzept und Projekthandbuch





Zwei neue Projektmerkmale

- Anzeige von IuK-Vorhaben (BY)
 - Im Projektvorschlag zu berücksichtigen
- Datenschutz (BY)
 - Im Grobkonzept zu berücksichtigen
 - Verfahrensbeschreibung

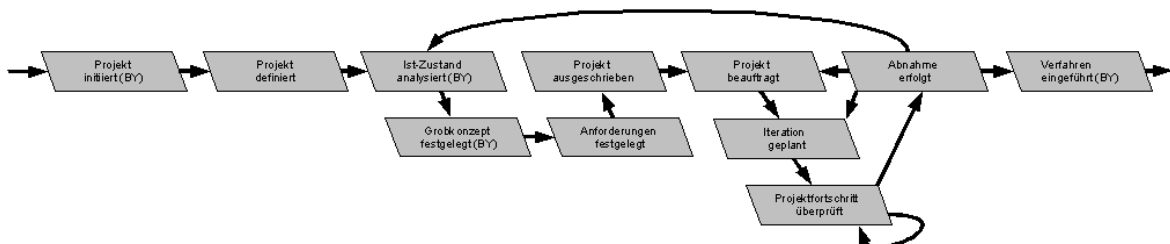
Anzeige von IuK-Vorhaben (BY):	Ja	Staatskanzlei und Staatsministerien sind grundsätzlich verpflichtet, für ihre Geschäftsbereiche alle IuK-Planungen und -Projekte sowie -Beschaffungen möglichst früh der Zentralen IuK-Leitstelle anzuzeigen. Einzelheiten sind in der Richtlinie BayITR-01 festgelegt. Für Projekte gilt: Anzuzeigen sind alle IuK-Projekte von ressortübergreifender oder grundsätzlicher Bedeutung, sowie Projekte zur Erstellung neuer Systeme mit einem Aufwand von über 40 Personentagen.
	Nein	
	Ja	
Datenschutz (BY):	Ja	
Exportverzeichnis:	c:\export	

SEE 2009
Seite 11



Projekttypvariante mit verfeinerter Anforderungsfestlegung

- Ist-Analyse und Grobkonzept zusätzlich zum Fachkonzept
- Neue Projektdurchführungsstrategie + Entscheidungspunkte



SEE 2009
Seite 12





Einbindung von Standards und Richtlinien

- Richtlinien mit maßgebendem Charakter für Projektdurchführung
 - inhaltliche Ergänzungen/Änderungen
 - Themenbeschreibung für Dokumentation und Produkte
 - Konventionsabbildungen
- Anforderungen aus Standards und Richtlinien
 - Mustertexte
 - Konfigurierbar über Projektassistent
- „Externe“ Produkte (z.B. EVB-IT)
 - Externe Kopiervorlagen



Kopiervorlagen

Beispiel: „Automatisierte Übernahme“ von Anforderungen aus den Standards und Richtlinien in ein Grobkonzept:





Mustertexte

Beispiel: Mustertext für ein Grobkonzept

3 ÜBERBLICK NICHT-FUNKTIONALE ANFORDERUNGEN
[Hier Ihren Text einfügen...]

3.1 Anforderungen an die zu verwendenden Client-Komponenten (Bayerische Staatsverwaltung)
[Hier Ihren Text einfügen...]

3.1.1 Anforderungen an das zu verwendende Betriebssystem - Client (BayITS-04)
[Hier Ihren Text einfügen...]
Betriebssystem - Client
Beschreibung:
Ein Betriebssystem ist ein Programm, das dem Benutzer und Anwendungsprogrammen elementare Dienste bereitstellt. Es steuert und überwacht die Abwicklung von Programmen und regelt den Betrieb des Rechnersystems. Hier wird zunächst nur der sog. „Fat Client“ betrachtet.

SEE 2009
Seite 15



Ausblick: Integration Entwicklung/Betrieb¹

„Softwareentwicklung und IT-Betrieb beeinflussen sich gegenseitig!“

- Anforderungsanalyse legt bereits Rahmenbedingungen für den Betrieb fest
- SW-Entwicklung muss die Rahmenbedingungen des Betriebs berücksichtigen
- Betrieb hat Auswirkungen auf SW-Pflege

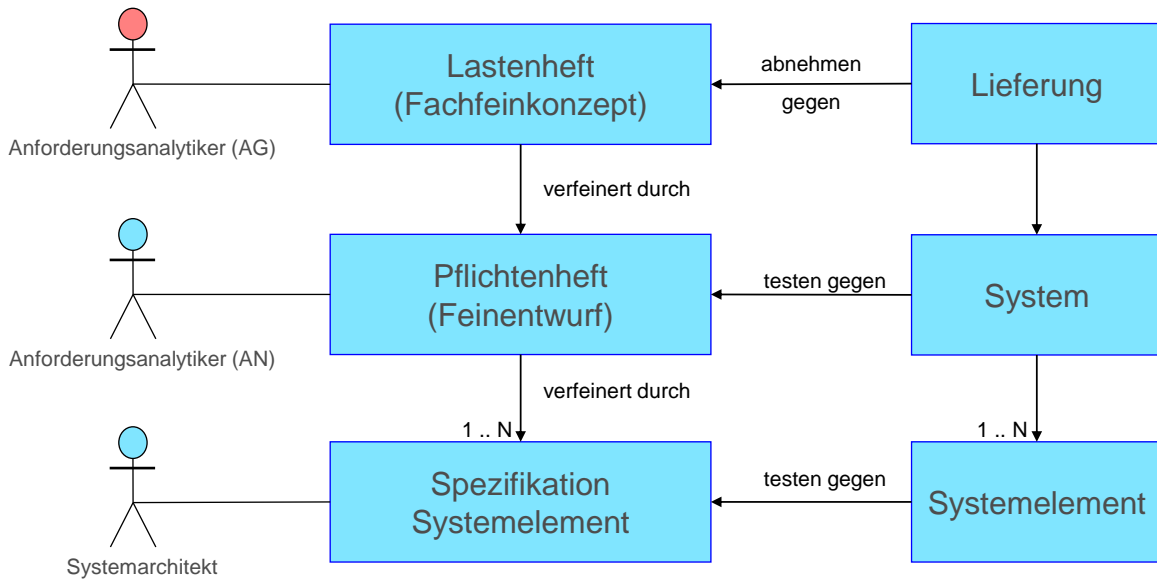
¹: nachstehende Folien vgl. Brenner, Garschhammer, Nickl:
Requirements Engineering und IT Service Management – Ansatzpunkte einer integrierten Sichtweise

SEE 2009
Seite 16





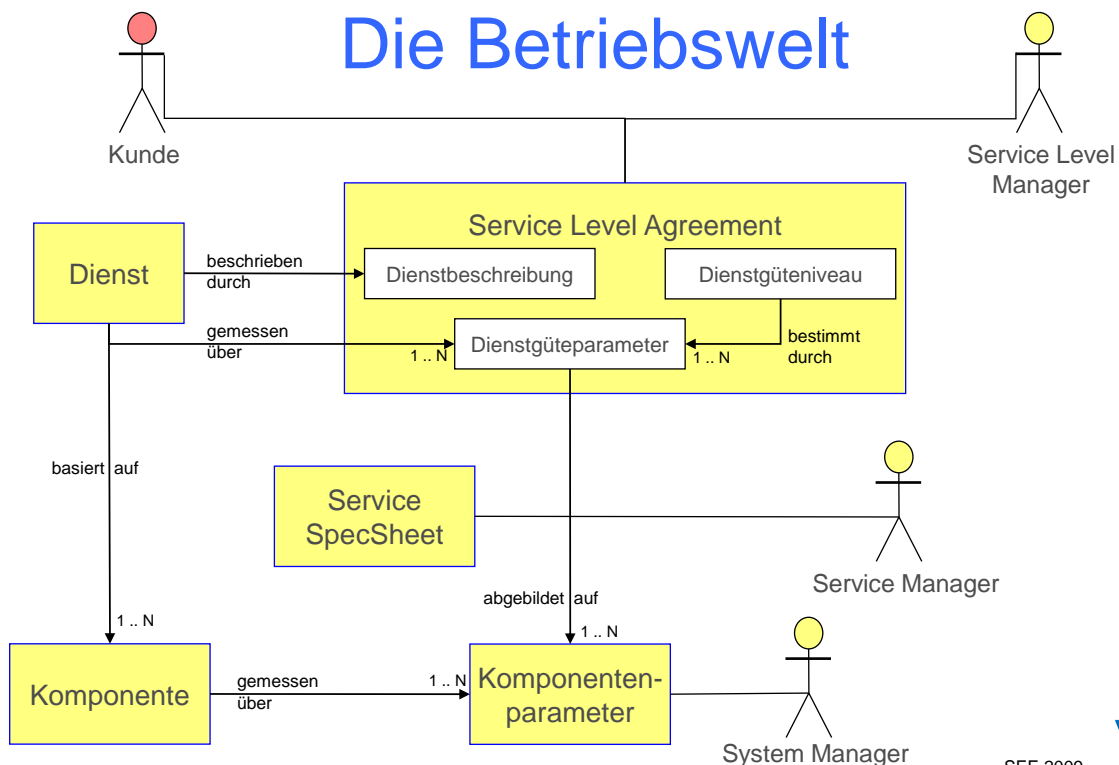
Die Entwicklerwelt



SEE 2009 Seite 17



Die Betriebswelt

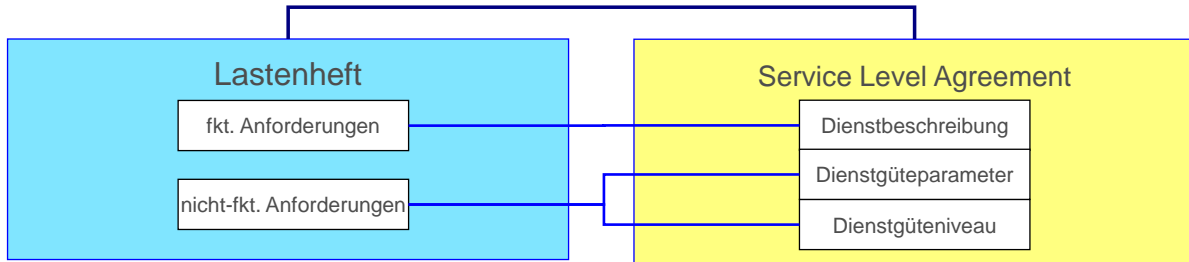


SEE 2009 Seite 18

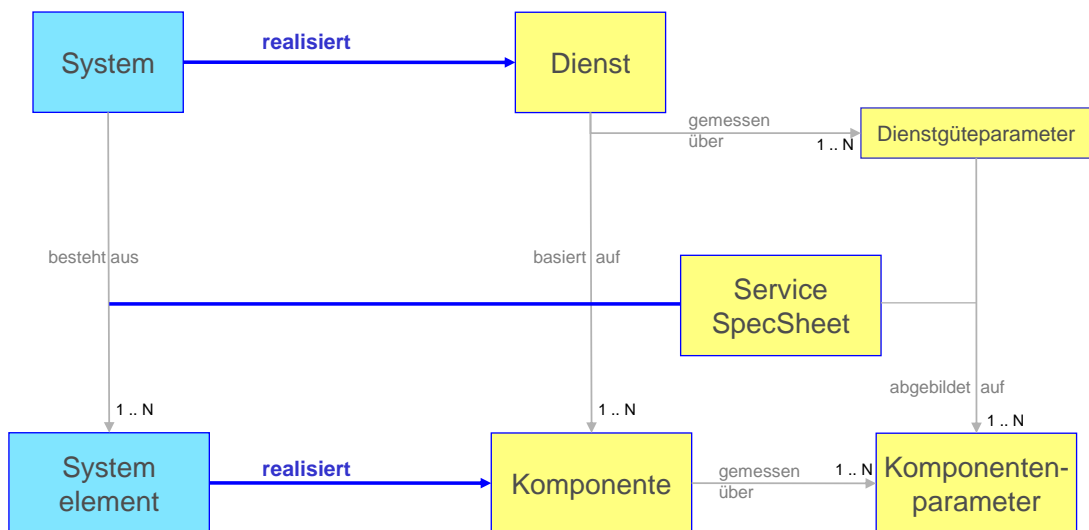




Die integrierte Welt - Lastenheft und SLA -

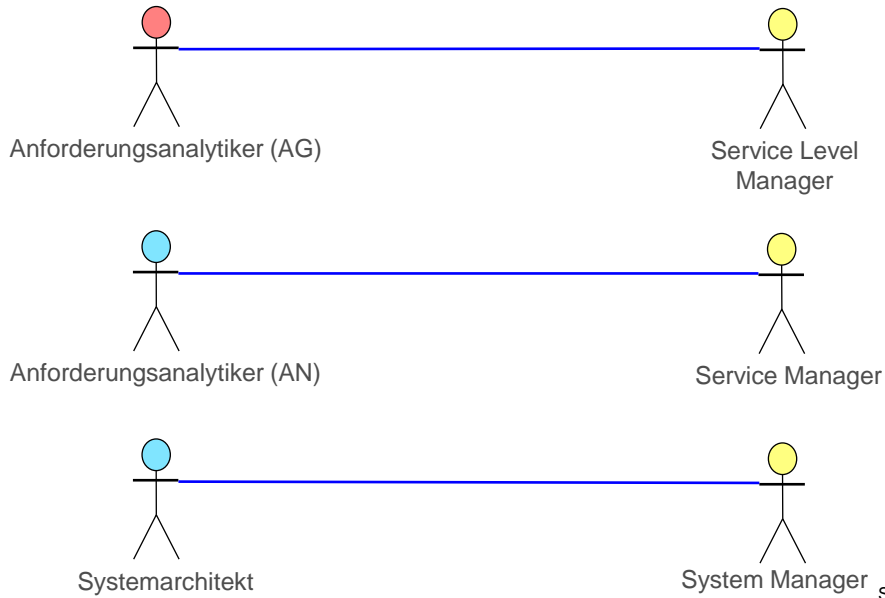


Die integrierte Welt - System und Dienst -





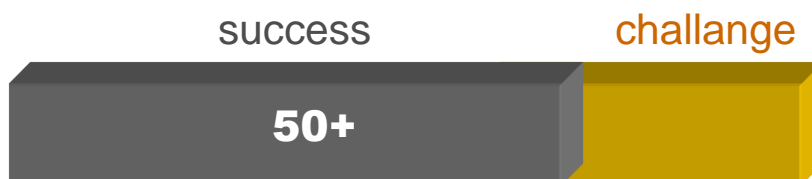
Die integrierte Welt - Rollen -



SEE 2009
Seite 21



Unser Ziel in Bayern



SEE 2009
Seite 22



1.3. Das Vorgehensmodell Bayern im neuen Gewand

2. Session 2: Projektmanagement

Inhalt

2.1. Das Sporthemd unterm Anzug V-Modell XT with Scrum inside	52
2.2. Projektmanagement nach PMI im V-Modell XT	57
2.3. Projektplanung im neuen V-Modell XT V1.3: Wie geht das denn genau?	81

2.1. Das Sporthemd unterm Anzug V-Modell XT with Scrum inside

Olaf Lewitz
microTOOL GmbH
Voltastr. 5
D-13355 Berlin
{Olaf.Lewitz}@microTOOL.de

Abstract

Das V-Modell XT sieht als Planungs- und Steuerungsmittel zur Durchführung eines Projekts eine Reihe von Aktivitäten vor, die im Wesentlichen zur Erzeugung der V-Modell®-Produkte dienen. Die Projektpraxis zeigt, dass die Planung mit V-Modell XT Aktivitäten aus verschiedenen Gründen für die Steuerung des Projektes häufig unpraktisch ist. Da Aufwände in den Projekten nicht im Wesentlichen in die Erstellung von Dokumenten fließen sollten, ist die Fertigstellung dieser Dokumente (oder Systemelemente) als alleiniges Merkmal zur Feststellung des Fertigstellungsgrads ungeeignet.

Motivation

Das V-Modell® XT ist ein bewährter Standard zur Durchführung von Systementwicklungsprojekten, insbesondere für öffentliche Auftraggeber. Hier wird häufig der Einsatz des V-Modells® XT vertraglich vereinbart. Scrum ist der derzeit erfolgreichste Vertreter der agilen Vorgehensweisen zur Projektdurchführung. Agile Methoden haben sich in vielen Softwareentwicklungsprojekten bewährt. Dienstleister mit solch positiven Erfahrungen möchten auch in Projekten, in denen vom öffentlichen Auftraggeber die Verwendung des V-Modells® XT vorgeschrieben ist, diese bewährten Vorgehensweisen einsetzen.

Doch wie kombiniert man diese gegensätzlichen Prozesse? Auf der einen Seite das V-Modell® XT mit seinen gut 30 Rollen und fast hundert Arten von Artefakten, auf der anderen Scrum mit drei Rollen, einer guten Handvoll von Artefakten und ein paar Meetings? Ich möchte hier einen Vorschlag vorstellen, wie man Scrum zur Projektsteuerung auch in V-Modell® XT Projekten anwenden kann. Dazu werde ich zunächst verschiedene Steuerungs-alternativen vorstellen, die ich für sinnvoller halte als die im V-Modell® XT vorgesehene. Ich möchte nicht empfehlen, intern Scrum zu verwenden und nach außen hin wie XT „zu tun“, also muss diskutiert werden, ob die von mir vorgeschlagene XT-Interpretation den Geist des V-Modells® XT erfüllt.

Projektsteuerung im V-Modell® XT

Das V-Modell® XT definiert für die Projektdurchführung Aktivitäten zur Produkterzeugung. Um flexibles Tailoring zu gewährleisten, wurde eine eindeutige Beziehung zwischen Aktivitäten und Produkten so gewählt, dass jede Aktivität genau ein Produkt erzeugt. Das bedeutet umgekehrt, dass jedes erzeugte V-Modell® XT Produkt zu einer Aktivität, also einem Vorgang oder Arbeitspaket im Projektplan, führt. Dadurch wird zum einen die Planungsgranularität des Projektplans festgelegt, zum anderen die Arbeit im

Projekt auf die Erzeugung von Produkten, die im V-Modell® XT ja größtenteils Dokumente sind, reduziert. Diese Reduzierung hat in meinen Augen vor allem den Nachteil, dass überhaupt nicht berücksichtigt wird, in welchem Maße die unterschiedlichen Aktivitäten - Analyse, Design, Entwicklung, Test - zur Wertschöpfung und zum Business Case des Projekts beitragen. Darüber hinaus wird mit dieser Planungstechnik, in Kombination mit den vorgegebenen Rollen, ein für den Projekterfolg ungünstiger Taylorismus nahegelegt. Die vorgegebene Granularität der Planung, für jedes Produkt eine Aktivität mit Aufwand, Ressourcen und Termin zu planen, ist zudem nur selten wirklich sinnvoll. Zur Illustration: Ein Statusbericht, der vielleicht in wenigen Minuten hochgradig automatisiert erzeugt werden kann, wenn eine durch das V-Modell® XT nahegelegte integrierte Projektinfrastruktur genutzt wird, lohnt kaum den Aufwand eines dafür geplanten Arbeitspakets. Die Erstellung eines Lastenhefts für eine große Ausschreibung andererseits ist ein sich über Monate hinziehender Prozess mit vielen beteiligten Personen - auch hier erscheint die Steuerung mit einem Arbeitspaket wenig passend.

Projekte beinhalten kreative Leistungen vieler beteiligter Personen. Auch wenn die zu erstellenden Produkte für eine Systementwicklung sich gut standardisieren lassen (wobei man trefflich darüber streiten kann, ob man deren Erstellung vorschreiben sollte), gilt das für die durchzuführenden Aktivitäten auf keinen Fall. Jedes Projekt ist anders, und jedes Team hat unterschiedliche Bedürfnisse an Art und Detaillierungsgrad der Planung und Steuerung. Daher ist der Ansatz, wie ihn das V-Modell® XT vorgibt, die Aktivitäten eines Projektes zu strukturieren, suboptimal.

Methoden der Projektsteuerung

Steuerung mit Aktivitäten

Klassisch werden Projekte mit Aktivitäten gesteuert, wie sie ja prinzipiell auch im V-Modell® XT verwendet werden. Alternativ werden diese auch Vorgänge oder Arbeitspakete genannt. Geplant wird je Aktivität der Aufwand für die zu leistende Arbeit, ein Start- und Ende-Termin, woraus sich die Dauer ergibt, sowie der Arbeitseinsatz der zugeordneten Mitarbeiter. Aufwand, Dauer und Einsatz bilden einen Dreisatz, so dass z.B. der Aufwand in Stunden multipliziert mit dem Einsatz in Stunden pro Tag die Dauer der Aktivität in Tagen ergibt. Klassischer Vorteil dieser Planungstechnik ist - zumindest in der Theorie - die Möglichkeit der Mitarbeiter-Einsatzplanung sowie die Möglichkeit, exakte Abhängigkeiten zu definieren. In der Praxis werden diese Vorteile selten wirksam: Für die Definition der Abhängigkeiten (Kontrollflüsse) gibt es selten eine wirklich sinnvolle Ebene, auf der man sie im Plan exakt festlegen kann. Sind die Abhängigkeiten so grob, dass sie nicht eingehalten werden, weil das Design-Team schon mit der Arbeit beginnen möchte, bevor die Anforderungsdefinition abgeschlossen ist, sind solche Abhängigkeiten Makulatur. Versucht man hingegen auf der Ebene kleinerer Features, deren einzelne Teilschritte komplett mit Abhängigkeiten zu versehen, wird der Plan leicht sehr unübersichtlich und schwer änderbar. Weitere Nachteile klassischer Aktivitätsplanung hat Mike Cohn in Agile Estimating and Planning analysiert:

- Aufwandschätzungen werden häufig als Terminzusagen interpretiert
- Aktivitäten werden nie zu früh fertig, damit sind Verspätungen vorprogrammiert
- Verspätungen addieren sich durch die Abhängigkeiten über die Laufzeit
- Multitasking ist weniger effizient, als einzelne Aufgaben nacheinander fertig zu stellen

2.1. Das Sporthemd unterm Anzug V-Modell XT with Scrum inside

Gerade der letzte Punkt ist auch ein wesentlicher Grund, warum die mathematische Einsatzplanung der Mitarbeiter mit dieser Technik selten gut funktioniert, neben der Tatsache, dass die Projektmitarbeiter im Allgemeinen keine frei austauschbaren Ressourcen sind. Grundsätzlich funktioniert diese Planungstechnik nur, wenn das Projekt sich auf die Zusagen der Organisation zur Verfügbarkeit der Projektmitarbeiter verlassen kann. Ist eine solche verlässliche Zusage nicht gegeben, ist die Planung mit Aktivitäten nicht sinnvoll möglich.

Häufig wird jedoch von Kunden oder Management ein klassischer Projektplan mit Aktivitäten verlangt. Dann empfehle ich, diese Technik pragmatisch einzusetzen, die Arbeitspakete klein zu halten - nicht länger als eine Woche - und auf Abhängigkeiten soweit wie möglich zu verzichten. Nutzen Sie regelmäßige Meilensteine, um den Projektfortschritt zu konsolidieren und transparent zu machen. Im Kontext des V-Modells® XT empfiehlt es sich, die vorgegebenen Aktivitäten als Ausgangspunkt der Planung zu nehmen, sich an diese Vorgabe aber nicht sklavisch zu halten. Ich habe gute Erfahrungen mit Projekten gemacht, in denen ausschließlich frei geplante Aktivitäten für die Steuerung verwendet und die vom V-Modell® XT geforderten Produkte zu den Meilensteinen aus der Projektdatenbank mit Anforderungen, Testfällen, Änderungen etc. generiert wurden.

Steuerung mit Action Items

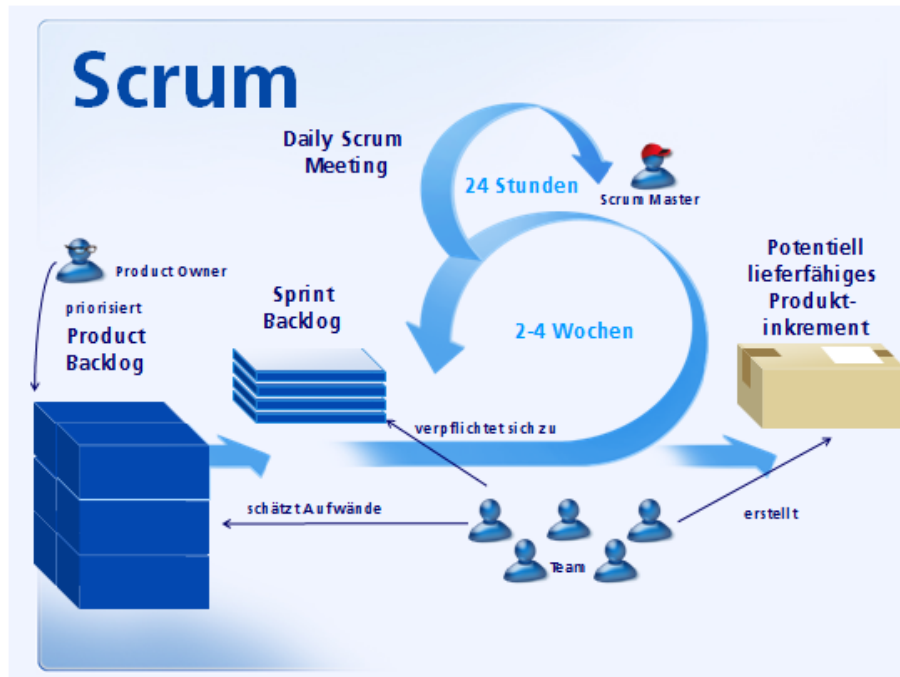
Action Items bieten eine extrem pragmatische Steuerungsmethode, sind jedoch keine Projektplanung im eigentlichen Sinne, weil damit keine vollständige Planung eines nicht trivialen Vorhabens möglich ist. Hierbei wird in regelmäßigen Projektmeetings eine Liste offener Punkte gepflegt, für die jeweils ein Mitarbeiter und ein Zieltermin vereinbart werden. Die Erledigung dieser Punkte wird im jeweils nächsten Meeting überprüft und besprochen, die nächsten Punkte definiert und somit eine rollierende Feinplanung realisiert.

Da das V-Modell® XT mit den Entscheidungspunkten einen guten Grobplan für das Projekt vorgibt, eignet sich diese Feinplanung gut als Ergänzung zu einer sinnvollen Gesamtplanung. Pragmatisch einsetzbar ist diese Technik vor allem dann, wenn die Organisation keinerlei Verfügbarkeitszusagen für die Projektmitarbeiter gibt und der Projektleiter keine Weisungsbefugnis über seine Mitarbeiter hat. In solchen Organisationen ist die mit dieser Steuerungsmethode überhaupt nicht mögliche Einsatzplanung auch kein wirklicher praktischer Nachteil.

Steuerung mit Scrum

Eine vollständige Darstellung von Scrum würde den Rahmen dieses Handouts sprengen, eine hervorragende Informationsquelle sind die Seiten der Scrum Alliance: Sie finden dort z.B. unter <http://www.scrumalliance.org/resources> den Scrum Guide. Dieser bietet eine gute und kurze Übersicht. Einen guten Einstieg bietet auch der Wikipedia-Artikel unter <http://de.wikipedia.org/wiki/Scrum>. Die wesentlichen Elemente von Scrum zeigt die folgende Grafik:

Eine auf einer Product Vision basierende priorisierte Anforderungsliste, das sogenannte Product Backlog, wird von einem cross-funktionalen Team in regelmäßigen Iterationen, sogenannten Sprints, abgearbeitet. Hierbei wird in jedem Sprint ein potentiell lieferfähiges Produktinkrement erstellt. Das Team organisiert seine Arbeit selbst. Der Prozess hat einen geringen Overhead, bietet eine gute Fortschrittskontrolle und ermöglicht das



frühzeitige Erkennen von Risiken. Darüber hinaus sorgt die Priorisierung des Backlogs für eine Optimierung des Geschäftswerts und die enge Zusammenarbeit im Team schafft eine hohe Effizienz bei der Produktentwicklung. Scrum vermeidet die oben genannten Nachteile der anderen Methoden:

- Abhängigkeiten werden durch die Einplanung in unterschiedliche Sprints abgebildet und müssen nicht auf den einzelnen Elementen nachvollziehbar sein
- Ressourcen-Zusagen müssen nur je Sprint gemacht werden, das vereinfacht die projektübergreifende Einsatzplanung der Mitarbeiter
- Planänderungen sind relativ einfach möglich
- Die Selbstorganisation des Teams sorgt typischerweise für eine hohe Motivation der Mitarbeiter

Somit ist Scrum meines Erachtens die erfolgversprechendste der vorgestellten Steuerungstechniken. Ich würde den Einsatz von Scrum immer dann empfehlen, wenn keine kulturellen oder vertraglichen Einschränkungen dagegen sprechen.

Integration von Scrum in das V-Modell® XT

Das V-Modell® XT definiert Rollen, Produkte und Aktivitäten, darüber hinaus Entscheidungspunkte für die Grobplanung. Die organisationsspezifische Anpassung des V-Modells® XT ist explizit empfohlen, somit lassen sich diese Elemente an die Bedürfnisse der Organisation anpassen. Mit einem Tool wie dem V-Modell® XT Editor oder in-Step ist es leicht möglich, die vom V-Modell® XT definierten Aktivitäten zu löschen, die Rollen zu vereinfachen und lediglich die für die Grobplanung erforderlichen Entscheidungspunkte und die Produkte zu belassen.

Action Items zur Planung bringt das V-Modell® XT bereits in Form des Produkts Arbeitsauftrag im Vorgehensbaustein Projektmanagement bereits mit. Die Arbeitsaufträge lassen sich zur rollierenden Feinplanung wie oben dargestellt nutzen.

2.1. Das Sporthemd unterm Anzug V-Modell XT with Scrum inside

Die Produkte und Rollen von Scrum lassen sich in einen neuen Vorgehensbaustein integrieren und mit einem Projektmerkmal verbinden, so dass Scrum als Alternative zur Projektsteuerung beim Tailoring zur Auswahl steht. Für die Grobplanung gibt es zwei Möglichkeiten:

- Sie nutzen die Standard-Durchführungsstrategien, z.B. für inkrementelle Entwicklung, und planen den groben Projektfortschritt mit deren Entscheidungspunkten. Die Anforderung an das System und die Erzeugung der zur Erreichung der Entscheidungspunkte notwendigen V-Modell® XT Produkte planen Sie in Ihr Backlog ein. Scrum Sprints sind Zwischenschritte auf dem Weg zum nächsten Entscheidungspunkt. Dabei können und sollten Sie die Projektabschnitte für Spezifikation, Architektur, Entwicklung und Test parallelisieren. Je nachdem wie viele V-Inkmente Sie durchlaufen, werden dabei vorläufige Zwischenstände von Pflichtenheft, Architektur etc. fertig gestellt oder die Entscheidungspunkte liegen zeitlich alle nah am Projektende, wenn die Software schon weitgehend fertig gestellt ist.
- Sie definieren eine eigene Durchführungsstrategie für Scrum. Diese enthält nur einen Entscheidungspunkt Release, den Sie beliebig oft durchlaufen können. In diesem Fall können Sie kaum noch von einer „Durchführung nach dem V-Modell® XT“ Ihres Projektes sprechen, aber möglicherweise haben Sie ja ein anderes Ziel: Innerhalb der Anpassung des V-Modells® XT an eine Organisation mag ja ein "reines Scrum-Projekt durchaus als ein Typ von Projekt unter mehreren möglichen sein.

Beide Alternativen sind in der Praxis durchführbar, und es sind vermutlich nicht die einzigen. Die erste Alternative ist günstig für einen Auftragnehmer, der ein Projekt nach dem V-Modell® XT durchführen muss, und dessen Auftraggeber sich erstens als Product Owner mit in das Projekt einbringt und zweitens nicht auf einem Standard-(Balken-)Projektplan besteht. Die zweite ist nützlich für Unternehmen, die unterschiedliche Varianten von Prozessen für verschiedene Typen von Projekten brauchen und daher das flexible Metamodell des V Modells® XT verwenden, um einen für die gesamte Organisation zu tailornen Prozess zu entwickeln. In einem solchen Prozess kann Scrum eine Option für ein Projekt sein. Beide Ansätze lassen sich übrigens auch problemlos kombinieren.

2.2. Projektmanagement nach PMI im V-Modell XT

Dr. Karl Kollischan
kobaXX Consultants
Dr. Karl Kollischan und Peter Bakalov GbR
Haag 1
91468 Gutenstetten
{karl.kollischan}@kobaxx.com

Abstract

Das Project Management Instituts (PMI) stellt mit dem PMBOK Guide einen international weit verbreiteten und auf breite Akzeptanz stößenden Standard für Projektmanagement zur Verfügung, der jedoch nicht die speziellen Belange von Software- bzw. Systementwicklungsprojekten abdeckt. Diese Lücke könnte durch auf Systementwicklung spezialisierte Vorgehensmodelle wie dem V-Modell XT geschlossen werden. Auf der anderen Seite wird Kritik am Projektmanagementverständnis des V-Modells XT dahingehend geübt, dass der Kenntnisstand der Projektmanagementlehre nicht adäquat berücksichtigt werde und eine stärkere Anlehnung an etablierte Standards und Begriffe wünschenswert wäre.

Dies legt es nahe, beide Methodiken miteinander zu verbinden, um zum einen das im V-Modell liegenden Expertenwissen für Systementwicklung zugänglich zu machen, und zum anderen sich mehr an etablierte Projektmanagementstandards anzulehnen. In diesem Vortrag wird gezeigt, dass eine Ad-Hoc-Anbindung ohne allzu großen Aufwand möglich ist. Die im PMBOK beschriebene Methodik weist einen stark deskriptiven Charakter auf, der es erlaubt, ein Vorgehen nach V-Modell XT als spezielle Ausprägung dieser Projektmanagementmethodik aufzufassen.

Um dies zu verdeutlichen, wird das Projektmanagementverständnis und die grundlegende Struktur des PMBOK skizziert sowie kurz auf die wesentlichen Projektmanagementaspekte des V-Modell XT eingegangen. Anhand eines fiktiven Projekts, das ein nach V-Modell XT vorgehender Auftraggeber an einen der PMI-Methodik folgenden Auftragnehmer vergeben hat, folgen wir den Überlegungen des designierten Projektleiters, wie sein Projekt auf möglichst pragmatische Art und Weise sowohl dem PMI-Standard seiner Firma als auch den VMXT-Vorgaben des Auftraggebers genügen kann. Nach einer Begriffsklärung und Analyse der grundlegenden Unterschiede und Gemeinsamkeiten wird gezeigt, wie die für unser Projekt vom VMXT-Projektassistenten erzeugten Rollen und Produkte sowie die gewählte Projektdurchführungsstrategie gemäß PMBOK interpretiert werden können bzw. diese Methodik unterstützen.

Als Fazit wird gezogen, dass ein Zusammenspiel der beiden Methodiken jetzt schon ohne großen Aufwand möglich ist, dieses jedoch seitens des V-Modells durch eine Anpassung des Vorgehensbausteins Projektmanagement (bzw. V-Modell-Kerns) harmonischer und damit attraktiver gestaltet werden könnte. Gegenstand des abschließenden Ausblicks ist es u.a., auch für andere Standards (etwa GPM oder PRINCE2) Varianten des Vorgehensbaustein Projektmanagement/ V-Modell-Kerns zur Verfügung zu stellen (bzw. diesen hinreichend allgemein zu formulieren, um mehreren Standards zu genügen).

Projektmanagement nach PMI im V-Modell XT

Dr. Karl Kollischan Projektmanagement nach PMI im V-Modell XT SEE2009 26. Mai 2009

Agenda

- Motivation
- Übersicht PMBOK
- Übersicht V-Modell XT
- VMXT- Projekt nach PMBOK
- Fazit

Dr. Karl Kollischan Projektmanagement nach PMI im V-Modell XT SEE2009 26. Mai 2009

2

Motivation



- Brückenschlag zwischen Projektmanagement und Systementwicklung
- Berührungsängste überwinden
- Weiterentwicklung des VMXT anregen
- Gegenseitiges Expertenwissen zugänglich machen

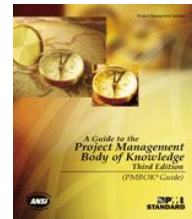
Project Management Institute



- Gründungsjahr 1969.
- Über 100.000 Mitglieder weltweit.
- Führende gemeinnützige Vereinigung für Themen des Projektmanagements.
- Entwickelt Standards für das Projektmanagement.
- Zertifizierung zum PM Professional (PMP®).
- Vision:
Weltweite Professionalität für PM in der Praxis mittels Standards, welche weitgehend anerkannt sind und in dieser Form durchgehend angewandt werden.



Project Management Body of Knowledge (PMBOK)



- ANSI Standard (ANSI 99-001-2008)
- Beschreibt zusammenfassend das allgemein anerkannte Wissen im Projektmanagement
- Stellt ein allgemeines Lexikon für Projektmanagement bereit
- Ist für alle am Prozess des Projektmanagements Beteiligten gedacht
- Befindet sich in einem kontinuierlichen Entwicklungsprozess
- Basis der Zertifizierung zum Project Management Professional (PMP®)

PMBOK: Definition Projekt



- Zeitlich beschränkt: Jedes Projekt hat einen definierten Start- und Endtermin
- Einzigartigkeit
 - eines Produktes,
 - einer Dienstleistung oder
 - eines Ergebnisses
- Fortschreitende Ausarbeitung (Progressive Elaboration)
 - Schrittweises Vorgehen unter sorgfältiger Berücksichtigung der zunehmenden Detaillierung
 - Iteratives Vorgehen

PMBOK: Projektmanagement



- Projektmanagement ist die Anwendung von Wissen, Fertigkeiten, Werkzeugen und Methoden auf Projektvorgänge, um die Projektanforderungen zu erfüllen.
- Projektmanagement wird bewältigt durch die korrekte Anwendung und Integration der 42 Projektmanagementprozesse:
 - geordnet nach 9 Wissensgebieten
 - und 5 Projektprozessgruppen

Projektmanagement umfasst typischerweise...



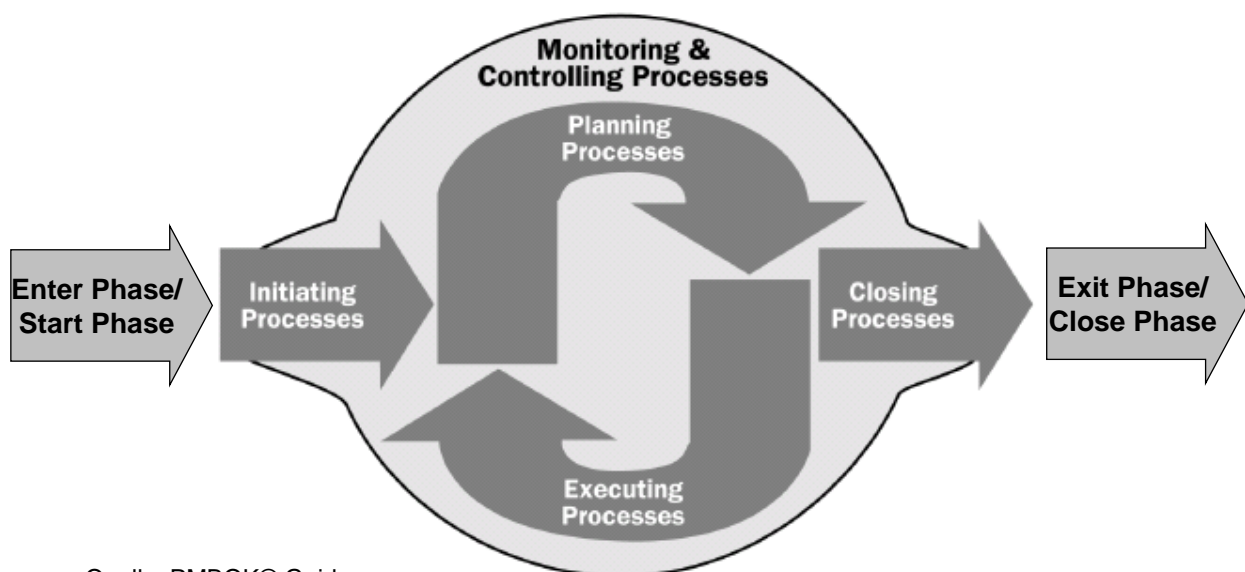
- Identifizieren von Anforderungen
- Den unterschiedlichen Bedarfen, Bedenken und Erwartungen der Stakeholder gerecht werden
- Ausbalancieren konkurrierender Anforderungen und Projekttrandbedingungen bzgl.
 - Inhalt und Umfang
 - Qualität
 - Terminplan
 - Budget
 - Ressourcen
 - Risiken
- Deswegen ist der Projektmanagementplan iterativ und wird während des Projektlebenszyklus immer weiter ausgearbeitet

Projektleiter und Projektponsor



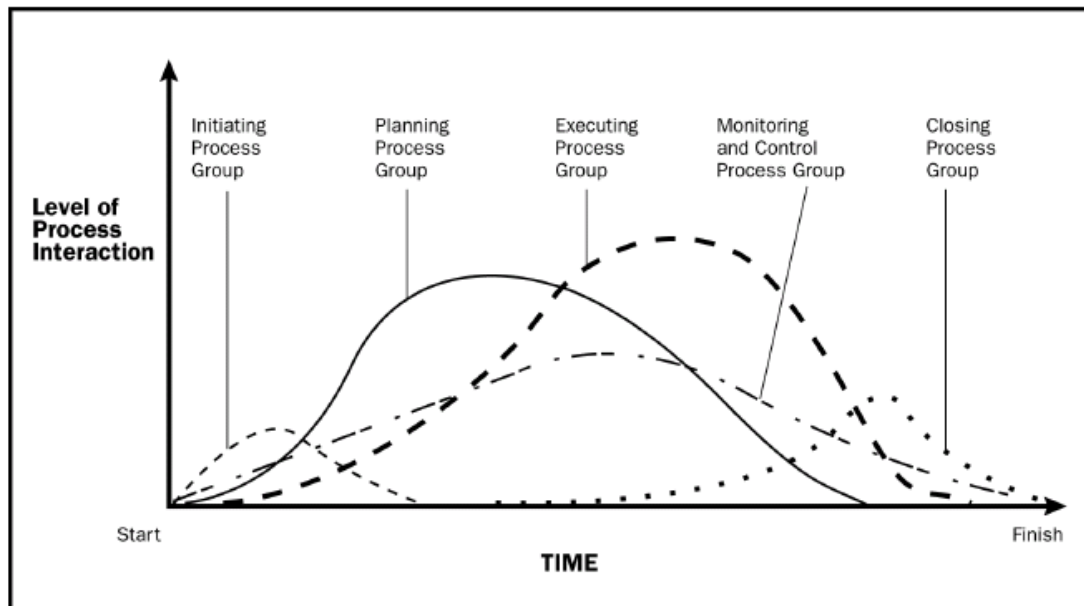
- Der Projektleiter ist die von der Organisation zur Erreichung der Projektziele beauftragte Person
- Der Projektponsor ist die Person oder Gruppe, die
 - die finanziellen Mittel für das Projekt bereitstellt
 - für das Projekt wirbt
 - als Sprecher zu höheren Managementebenen auftritt
 - das Projekt beauftragt
 - als Eskalationspfad dient
 - in andere wichtige Themen (z.B. Reviews, Projektfortschrittsentscheidungen,...) eingebunden sein kann

Projektmanagement Prozessgruppen



Quelle: PMBOK® Guide

Lebenszyklus und Prozessgruppen

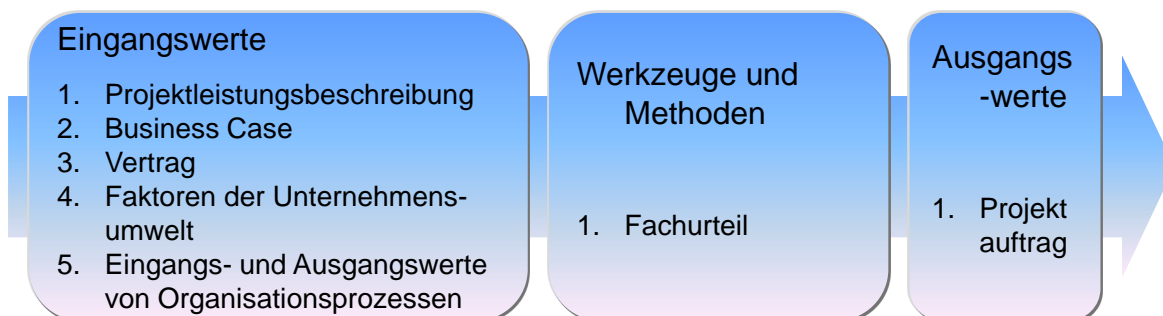


Quelle: PMBOK® Guide

42 Projektmanagementprozesse



- Thematisch strukturiert nach „Wissensgebieten“
- Beschrieben durch:
 - Eingangswerte
 - Werkzeuge und Methodden
 - Ausgangswerte

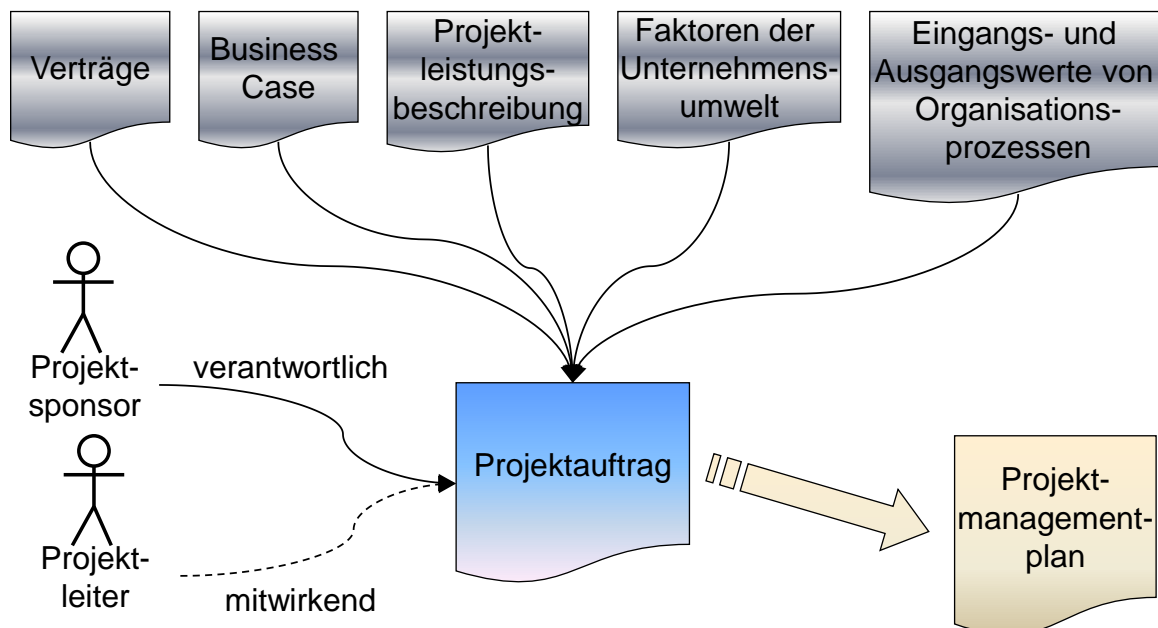


Beispiel: 4.1 Projektauftrag entwickeln

Wissensgebiete



Projektauftrag



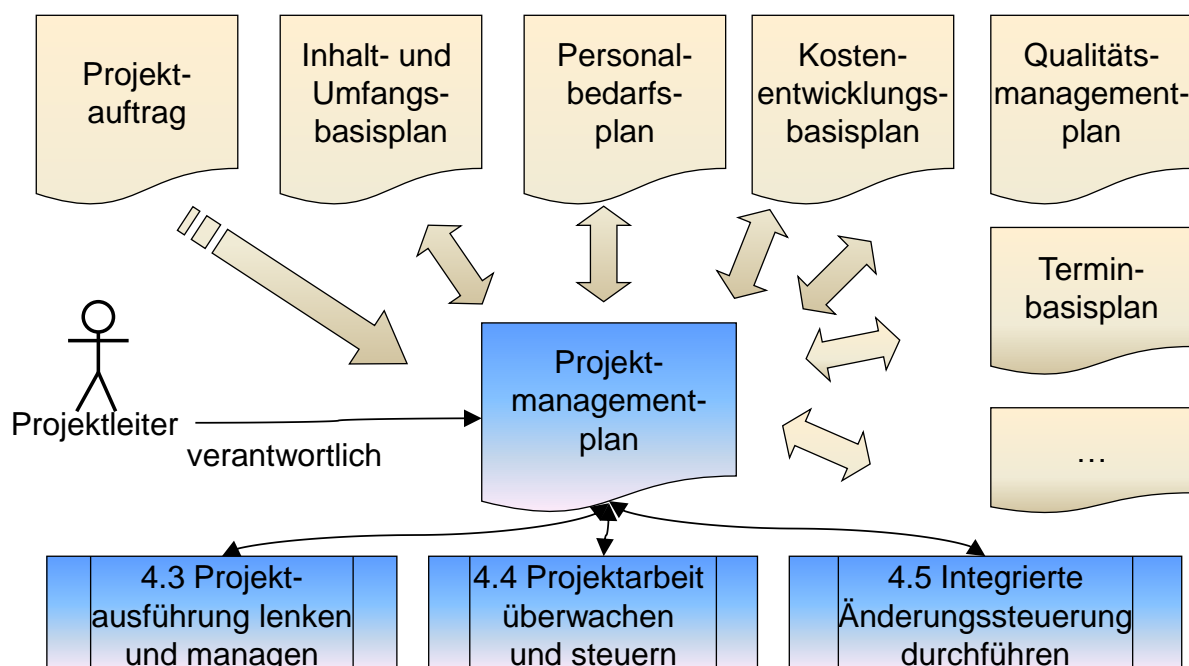
Inhalt Projektauftrag



Das Dokument, mit dem das Projekt formal autorisiert wird:

- Zweck und Begründung des Projekts
- Messbare Projektziele und Erfolgsfaktoren
- Anforderungen auf hoher Ebene
- Projektbeschreibung auf hoher Ebene
- Risiken auf hoher Ebene
- Übersichtsplan der Meilensteine
- Übersicht Budget
- Projekterfolgskriterien
- Projektleiter und Projektsponsor

Projektmanagementplan

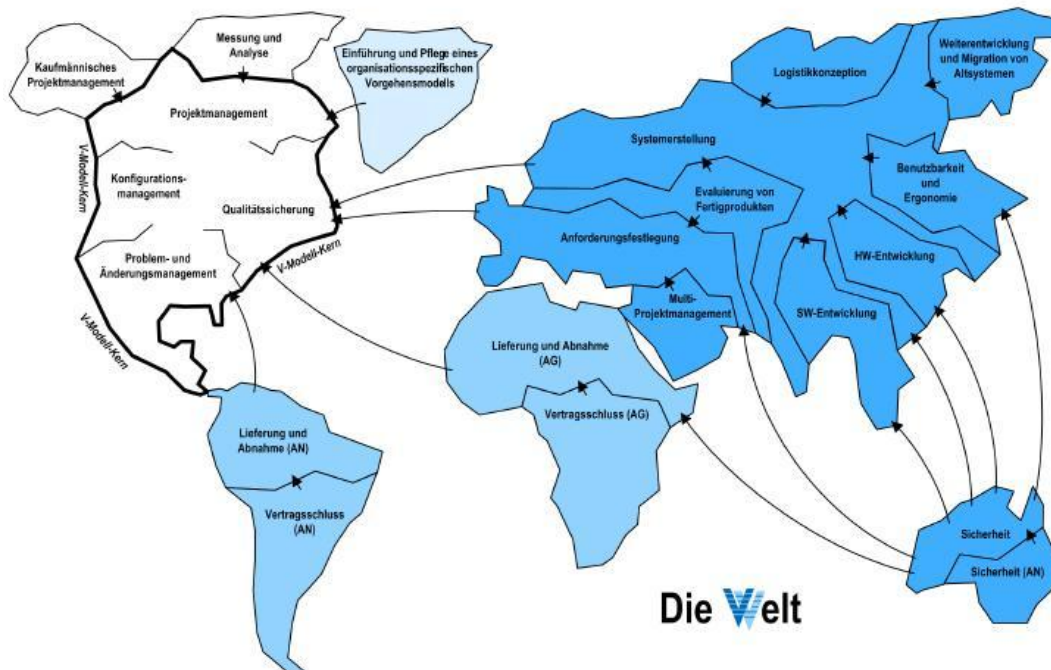


Inhalt Projektmanagementplan

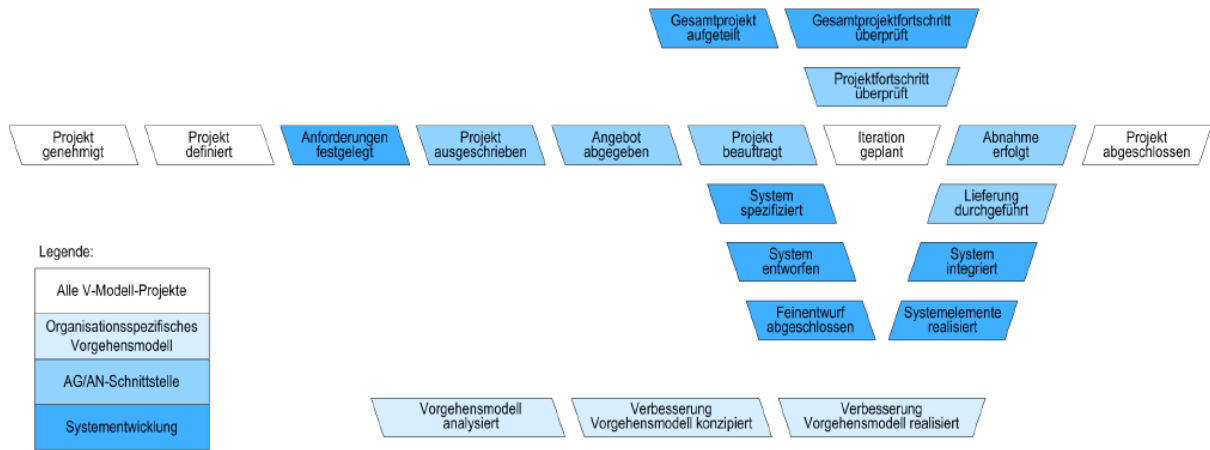
Definiert, wie das Projekt ausgeführt, überwacht, gesteuert und abgeschlossen wird:

- Untergeordnete Pläne bzw. Referenzen darauf
- Lebenszyklus des Projekts
- Tailoringergebnisse
- Wie Arbeiten ausgeführt werden
- Änderungsmanagementplan
- Konfigurationsmanagementplan
- Bedarf und Methoden zur Kommunikation mit Stakeholdern
- etc.

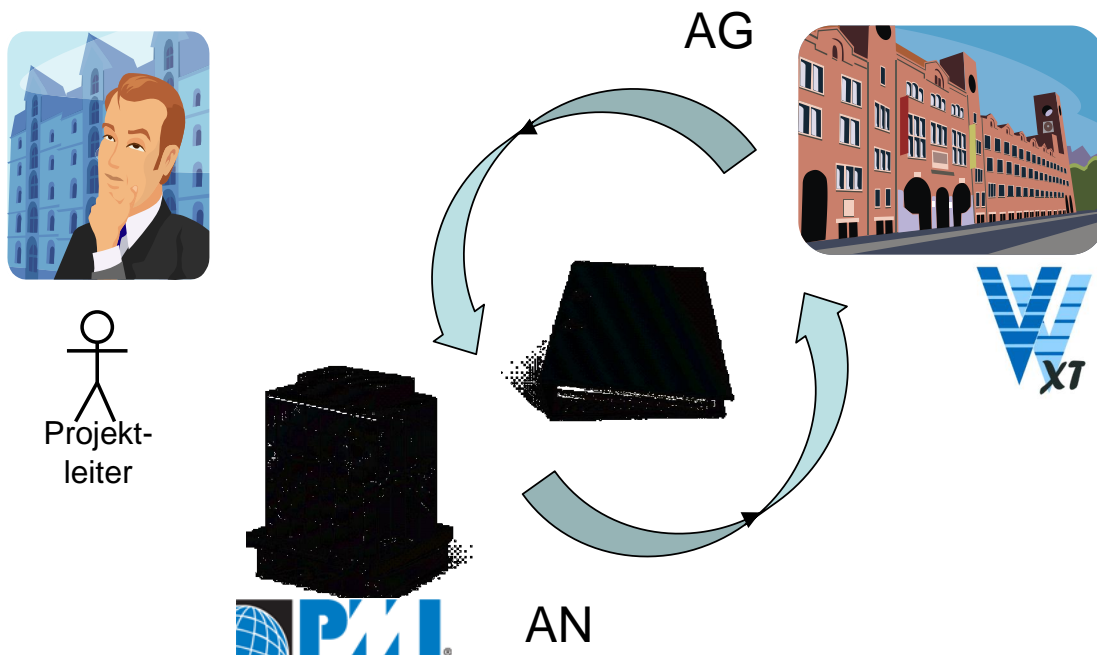
Die V-Modell XT Welt



Projektdurchführungsstrategien



VMXT-Projekt nach PMBOK





Grundlegender Vergleich



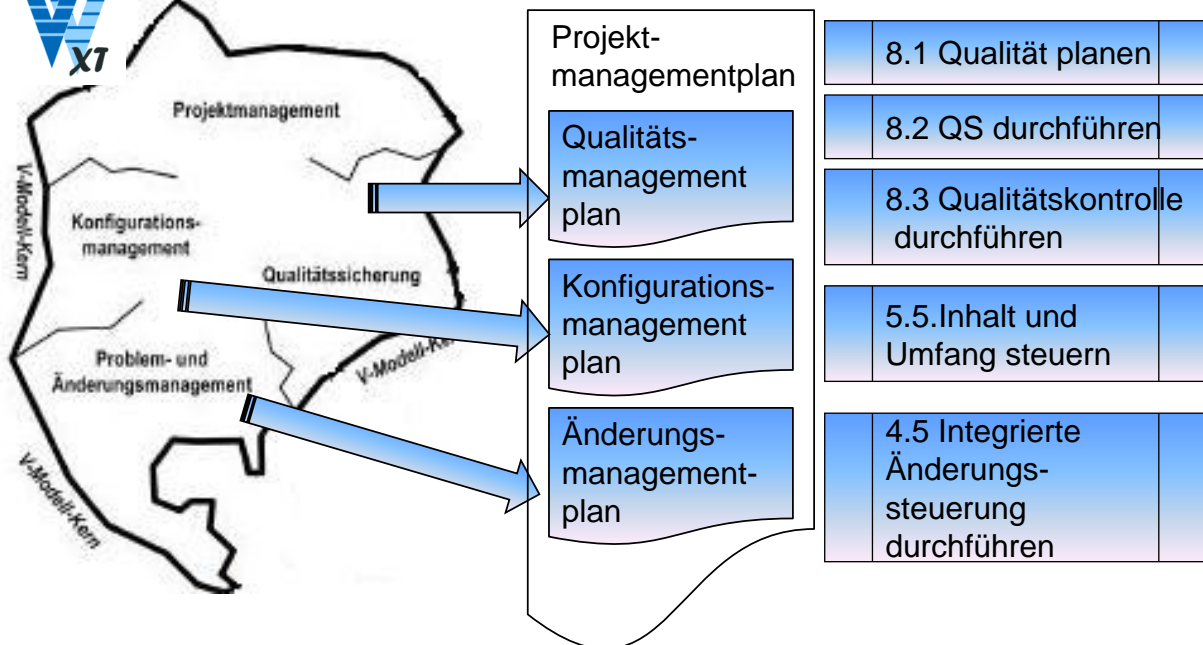
		
Projektmanagement		iterativ
Projektart	Alle Arten von Projekten	Systementwicklungsprojekte
	Nur Projektmanagement	Projektmanagement und Systementwicklung
	Alle Aspekte des Projektmanagements	Einige Aspekte
	deskriptiv	präskriptiv
	aktivitätszentriert	produktzentriert
Phasen	vom Anwendungsbereich abhängig	systementwicklungsspezifisch

Begriffe



	
Projektleiter	Projektleiter
Projektsponsor	(in etwa) Projektmanager
Liefergegenstand	Produkt
Personalressource	(enthält) Rolle
Meilenstein	Entscheidungspunkt
Vorgang	Aktivität
Arbeitspaket	Arbeitspaket

Der V-Modell Kern und PMBOK



Projekthandbuch

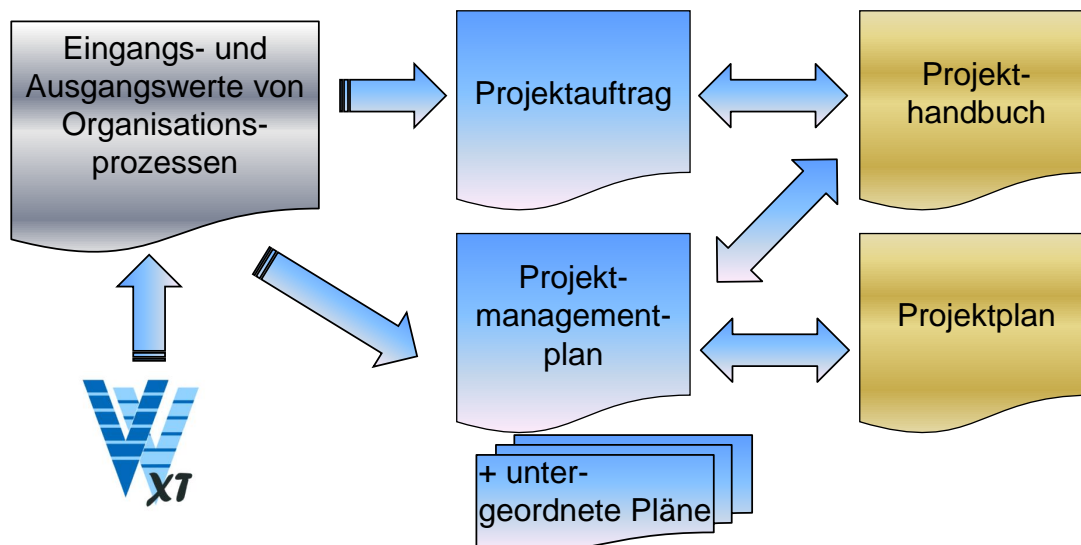


1. Projektüberblick, Projektziele und Erfolgsfaktoren
2. Teilprojekte
3. Projektspezifisches V-Modell
4. Abweichungen vom V-Modell
5. Projektdurchführungsplan
6. Mitwirkung und Beistellungen des Auftraggebers
7. Organisation und Vorgaben zum Projektmanagement
8. Organisation und Vorgaben zum Risikomanagement etc.
9. Vorgaben für das Projekthandbuch der Auftragnehmer
10. Berichtswesen und Kommunikationswege

Projektplan

1. Projektdurchführungsplan
2. Integrierte Planung
3. Prüfplan Dokumente
4. Integrations- und Prüfplan Systemelemente
5. Prüfplan Prozesse
6. Ausbildungsplan

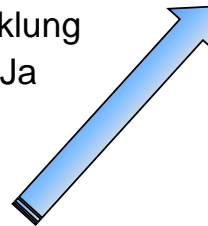
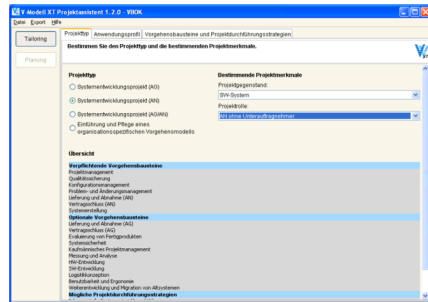
Projekthandbuch und -plan



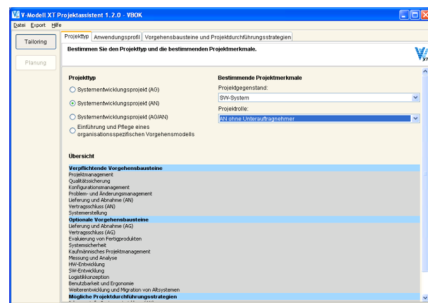
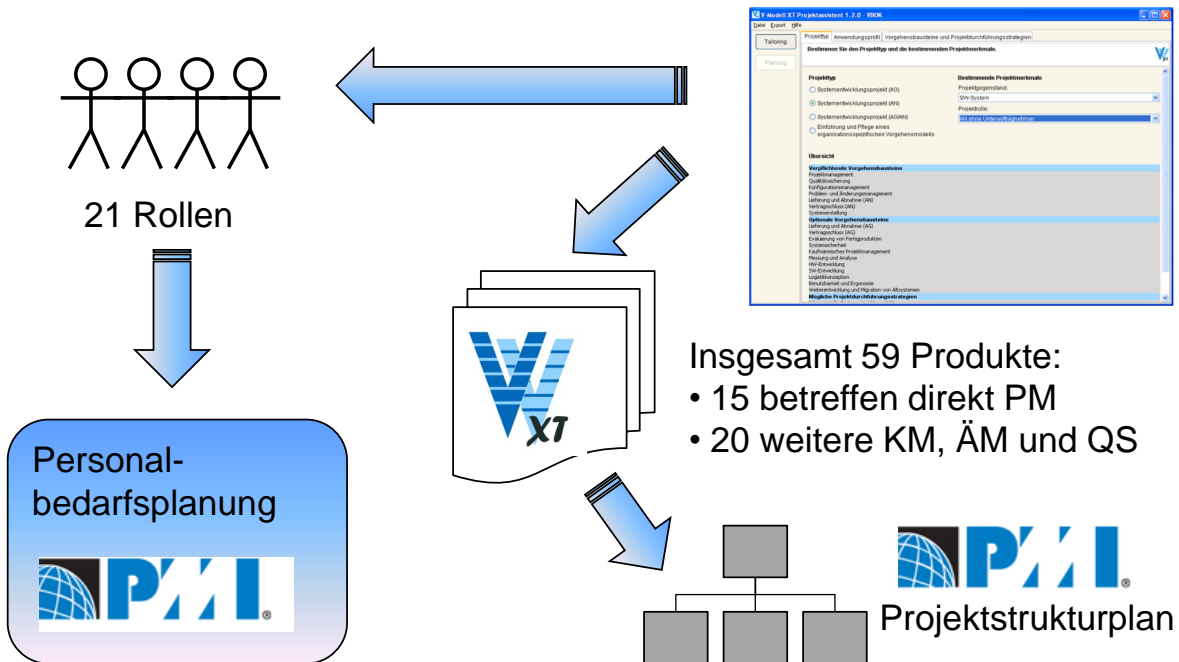
VMXT Projektassistent



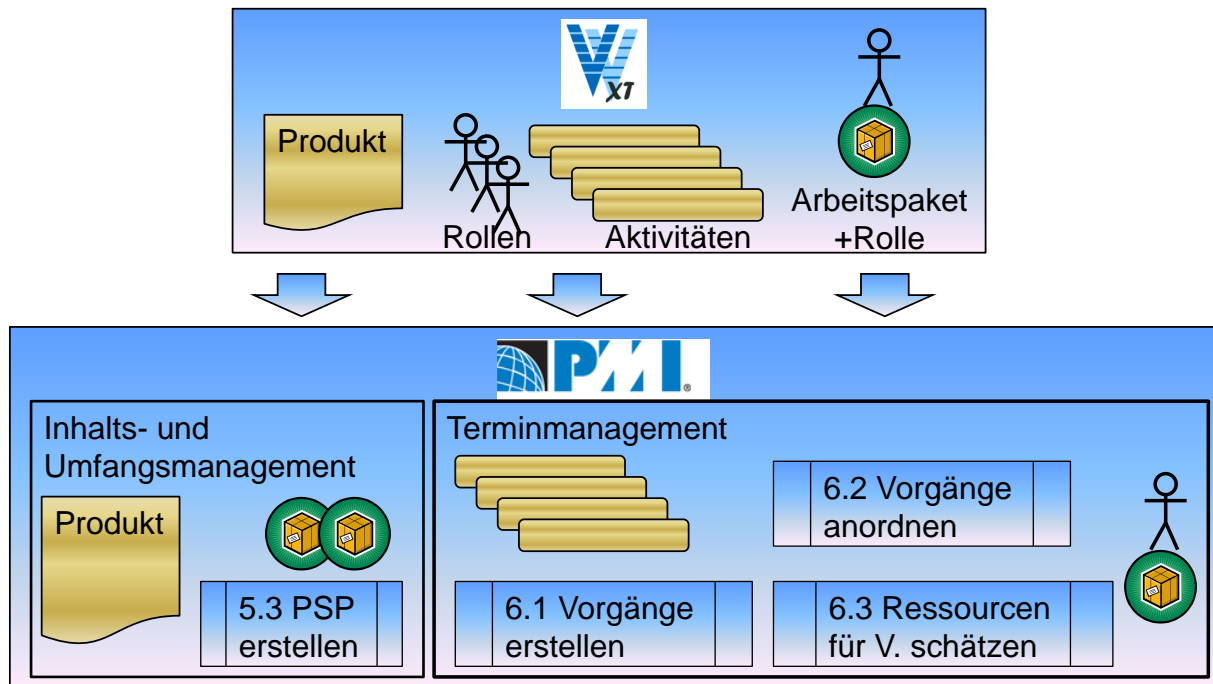
- **Projekttyp**
 - Systementwicklungsprojekt (AN)
 - SW-System
 - AN ohne Unter-AN
- **Anwenderprofil**
 - Projektgegenstand: SW-System
 - Projektrolle: AN ohne Unterauftragnehmer
 - Systemlebenszyklusausschnitt: Entwicklung
 - Kaufmännisches Projektmanagement: Ja
 - Quantitative Projektkennzahlen: Ja
 - Fertigprodukte: Nein
 - Benutzerschnittstelle: Ja
 - Safety and Security: Ja
 - Hohe Realisierungsrisiken: Nein



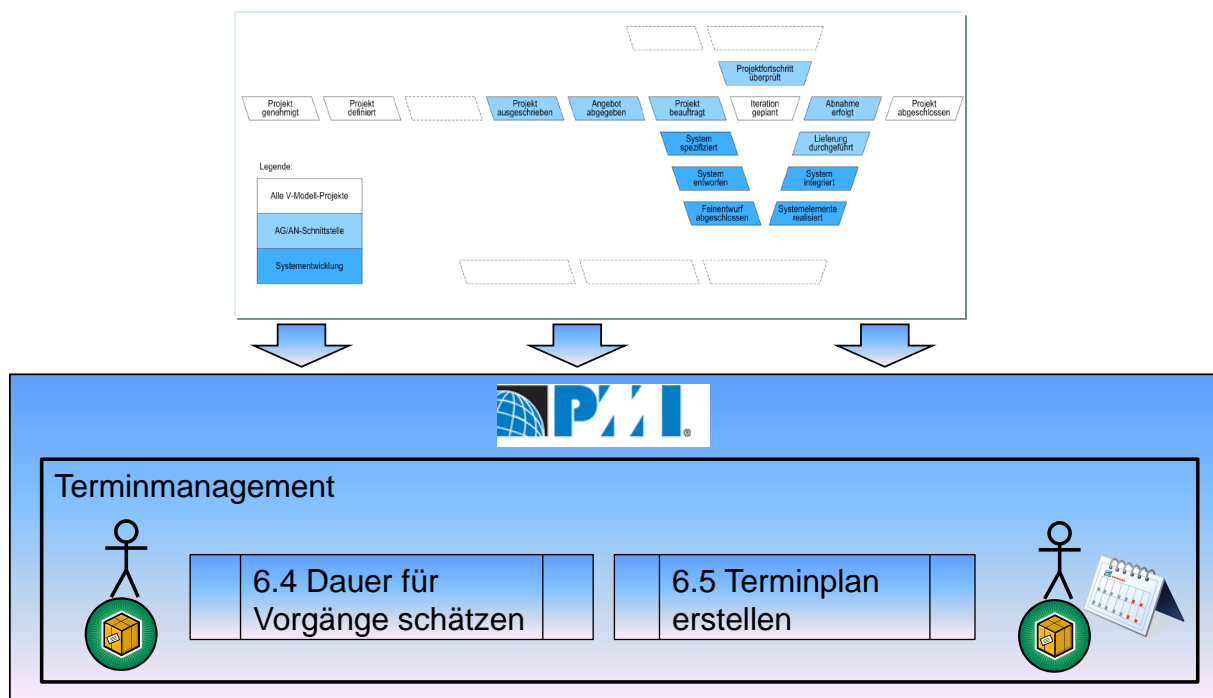
Zuordnung von Rollen und Produkten



Arbeitspakete



Terminplanung



Wie ergänzt das V-Modell XT das PMBOK?



- Mehr Anleitung für das Projektmanagement durch Vorgabe von
 - Rollen
 - Produkten
 - Abhängigkeiten
 - Meilensteinen (Entscheidungspunkten)
- Expertenwissen in Systementwicklung
- Gute Integration von Projektmanagement- und Systementwicklungsprozess

Wie ergänzt das PMBOK das V-Modell XT?



- Weitergehendes Projektmanagementverständnis
 - allgemein anerkannte Management-Methoden
 - Ausführliche Beschreibung und Strukturierung der Aufgaben im Projektmanagement
- Anknüpfung an etablierten Standard

Fazit



- PMBOK und V-Modell XT können sich sinnvoll ergänzen
- Ein Zusammenspiel ist jetzt schon ohne allzu großen Aufwand möglich
- Seitens des V-Modells können Bedingungen geschaffen werden, um eine Anbindung zu erleichtern:
 - Begriffe
 - Produkte und Themen

Ausblick



- Vorgehensbaustein „Einführung und Pflege eines organisationsspezifischen V-Modells“ unter Randbedingung „PMBOK“ weiterentwickeln
- Dieses Vorgehen auf andere Projektmanagement Methodiken ausweiten
- Anbindung an externe Methodik in Tools integrieren:
 - zum Bsp. im PA „PM-Methode“ selektieren – Dokumente werden dann entsprechend erzeugt

Literatur



- PMBOK® Guide – Fourth Edition
- V-Modell XT Version 1.3
- Siegfried Seibert: Das aktuelle Stichwort: V-Modell XT (projektMANAGEMENT aktuell 2/2006)

Vielen Dank für Ihre
Aufmerksamkeit!



Fragen
?

Dr. Karl Kollischan


kobaXX Consultants GbR
Tel. +49 1520 498 92 92
karl.kollischan@kobaxx.com

Anhang: die 9 Wissensgebiete




4 Integrationsmanagement



Initiierungsprozessgruppe	Planungsprozessgruppe	Ausführungsprozessgruppe	Überwachungs- & Steuerungsprozessgruppe	Abschlussprozessgruppe
4.1 Projekt-auftrag entwickeln	4.2 Projekt-management-plan entwickeln	4.3 Projekt-ausführung lenken und managen	4.4 Projekt-arbeit steuern und überwachen 4.5 Integrierte Änderungs-steuerung durchführen	4.6 Projekt oder Phase abschließen 


5 Inhalts- und Umfangsmanagement




Initiierungsprozessgruppe	Planungsprozessgruppe	Ausführungsprozessgruppe	Überwachungs- & Steuerungsprozessgruppe	Abschlussprozessgruppe
	5.1 Anforderungen erfassen 5.2 Inhalt und Umfang planen 5.3 PSP erstellen		5.4 Inhalt und Umfang verifizieren 5.5 Inhalt und Umfang steuern	

6 Terminmanagement




Initiierungsprozessgruppe	Planungsprozessgruppe	Ausführungsprozessgruppe	Überwachungs- & Steuerungsprozessgruppe	Abschlussprozessgruppe
	6.1 Vorgänge definieren 6.2 Vorgangsfolgen festlegen 6.3 Aufwand für Vorgang schätzen 6.4 Dauer des Vorgangs schätzen 6.5 Terminplan entwickeln		6.6 Terminplan steuern	

7 Kostenmanagement


Initiierungsprozessgruppe	Planungsprozessgruppe	Ausführungsprozessgruppe	Überwachungs- & Steuerungsprozessgruppe	Abschlussprozessgruppe
	7.1 Kosten schätzen 7.2 Budget festlegen		7.3 Kosten steuern	

8 Qualitätsmanagement

Initiierungsprozessgruppe	Planungsprozessgruppe	Ausführungsprozessgruppe	Überwachungs- & Steuerungsprozessgruppe	Abschlussprozessgruppe
	8.1 Qualität planen	8.2 Qualitäts-sicherung durchführen	8.3 Qualitäts-kontrolle durchführen	


9 Personalbedarfsmanagement



Initiierungsprozessgruppe	Planungsprozessgruppe	Ausführungsprozessgruppe	Überwachungs- & Steuerungsprozessgruppe	Abschlussprozessgruppe
	9.1 Personalbedarf planen	9.2 Projektteam zusammenstellen 9.3 Projektteam entwickeln 9.4 Projektteam leiten		

10 Kommunikationsmanagement



Initiierungsprozessgruppe	Planungsprozessgruppe	Ausführungsprozessgruppe	Überwachungs- & Steuerungsprozessgruppe	Abschlussprozessgruppe
10.1 Stakeholder identifizieren	10.2 Kommunikation planen	10.3 Information verteilen 10.4 Erwartungen der Stakeholder managen	10.5 Über Projektfortschritt berichten	

11 Risikomanagement

Initiierungsprozessgruppe	Planungsprozessgruppe	Ausführungsprozessgruppe	Überwachungs- & Steuerungsprozessgruppe	Abschlussprozessgruppe
	11.1 Risikomanagement planen 11.2 Risiken identifizieren 11.3 Qualitative Risikoanalyse durchführen 11.4 Quantitative Risikoanalyse durchführen 11.5 Risikobewältigung planen		11.6 Risiken überwachen und steuern	

12 Beschaffungsmanagement

Initiierungsprozessgruppe	Planungsprozessgruppe	Ausführungsprozessgruppe	Überwachungs- & Steuerungsprozessgruppe	Abschlussprozessgruppe
	12.1 Beschaffungen planen	12.2 Beschaffungen durchführen	12.3 Beschaffungen verwalten	12.4 Beschaffungen abschließen

2.3. Projektplanung im neuen V-Modell XT V1.3: Wie geht das denn genau?

Jan Friedrich, Dr. Klaus Bergner

4Soft GmbH

Mittererstr. 3

80336 München

{friedrich | bergner}@4soft.de

Abstract

Mit der Version 1.3 des V-Modell XT verfügen Prozessingenieure über ein komplett überarbeitetes Metamodell, insbesondere auch hinsichtlich der Projektdurchführungsstrategien und Abläufe. Der Vortrag zeigt, welche neuen Möglichkeiten Projektleiter damit in der Praxis gewinnen, um auf Basis von Projektdurchführungsstrategien V-Modell-konforme Projektpläne zu erstellen. In der Folge geht er darauf ein, wie das Metamodell und seine Semantik formal definiert sind. Dieser zweite Teil ist besonders für Prozessingenieure und Werkzeughersteller relevant, die Änderungen am V-Modell vornehmen oder Projektplanungswerkzeuge für das V-Modell entwickeln. Indem der Vortrag die Brücke zwischen der praktischen Anwendung und den theoretischen Grundlagen schlägt, demonstriert er insgesamt den Nutzen formal fundierter Vorgehensmodelle.

Motivation

Die Projektdurchführungsstrategien des V-Modell XT [1] legen fest, in welcher Reihenfolge die einzelnen Meilensteine in einem Projekt erreicht werden sollen. Bis zur V-Modell-Version 1.2.1.1 waren diese Ablaufinformationen im Modell sehr unstrukturiert und für den Prozessingenieur undurchsichtig abgelegt (siehe auch [2]). Auch mangelte es an Ausdrucksmächtigkeit, sodass der Projektleiter bei der Projektplanung mit dem V-Modell XT Projektassistenten teilweise Expertenwissen besitzen musste, um den gewünschten Projektplan zu erhalten.

Im V-Modell XT ab Version 1.3 hat sich die Beschreibung von Projektdurchführungsstrategien grundlegend geändert. Zudem wurde ihre Semantik klar definiert und teilweise erweitert. Dies hat konkrete, in der Praxis relevante Auswirkungen auf unterschiedliche Zielgruppen:

- **Projektleiter** passen ihre Projektpläne in der Praxis häufig an und müssen deshalb wissen, welche Projektpläne von einer Projektdurchführungsstrategie zugelassen sind und welche nicht.
- **Werkzeughersteller** wollen den Projektleiter in seiner Arbeit unterstützen und müssen deshalb Metamodell und Semantik so umsetzen, dass ihr Werkzeug das Vorgehensmodell einhält, aber dem Projektleiter größtmögliche Freiheit lässt.
- **Prozessingenieure**, die das V-Modell XT anpassen oder erweitern, müssen den umgekehrten Weg gehen: Basierend auf den erwünschten Projektabläufen müssen sie eine Projektdurchführungsstrategie entwerfen, die alle gewünschten Möglichkeiten unterstützt, aber keine ungewollten Freiheiten bietet.

2.3. Projektplanung im neuen V-Modell XT V1.3: Wie geht das denn genau?

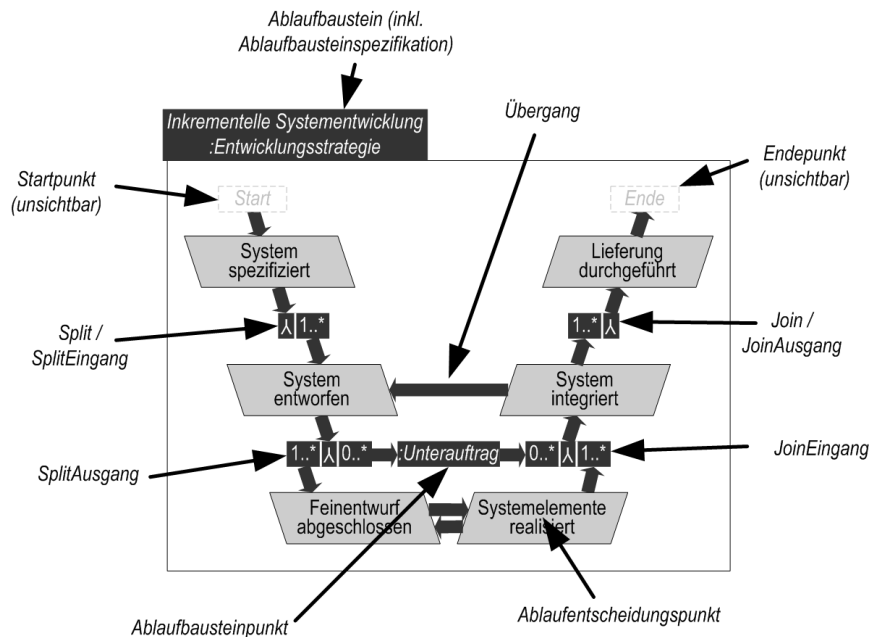


Abbildung 2.1.: Beispiel-Projekt durchführungsstrategie

Ansatz

Das neue Metamodell für Projektdurchführungsstrategien orientiert sich stark am UML-Metamodell für Aktivitätsdiagramme, wie Abbildung 2.2 zeigt. Es ist allerdings keine Teilmenge des UML-Metamodells und unterscheidet sich auch hinsichtlich der Semantik in einigen wesentlichen Punkten. Die Abweichungen begründen sich aus den speziellen Anforderungen bei der Durchführung von Projekten: Im Gegensatz zu „normalen“ Geschäftsprozessen laufen sie nicht jedes Mal auf die gleiche Art und Weise ab, sondern müssen flexibel geplant und umgeplant werden können. Dies betrifft insbesondere die Behandlung von Inkrementen und Iterationen in Kombination mit parallelen Teilprojekten und Unterabläufen.

Die formale Definition der Semantik der V-Modell-Projekt durchführungsstrategien basiert auf graphischen Beschreibungstechniken, die selbst wiederum in Form einer mathematisch-algebraischen Spezifikation definiert sind. Um entscheiden zu können, ob ein Projektplan zu einer Projekt durchführungsstrategie passt, werden die beiden Hilfskonstrukte Petri-Netz und Petri-Spur verwendet, wie Abbildung 2.3 zeigt.

Bewertung

Der Vortrag schlägt die Brücke zwischen der praktischen Anwendung und den theoretischen Grundlagen. Einerseits zeigt er anschaulich die Möglichkeiten des neuen V-Modell XT V1.3 auf, andererseits demonstriert er den Nutzen einer formalen, semantisch präzisen Fundierung von Vorgehensmodellen. Die erarbeitete formale Semantikdefinition bringt Klarheit für Prozessingenieure, Werkzeughersteller und Projektleiter. Insbesondere für letztgenannte sind allerdings mathematische Formeln nicht die angemessene Darstellungsform. Sekundärliteratur wie beispielsweise [3] oder [4] muss hier auf anschauliche Art Hilfe leisten.

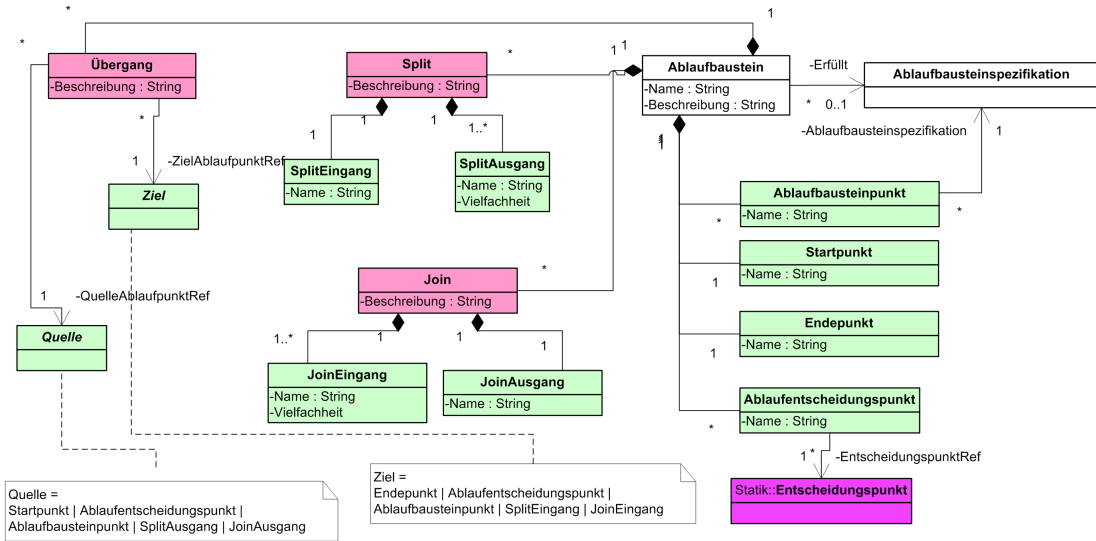


Abbildung 2.2.: Ablauf-Metamodell des V-Modell XT V1.3

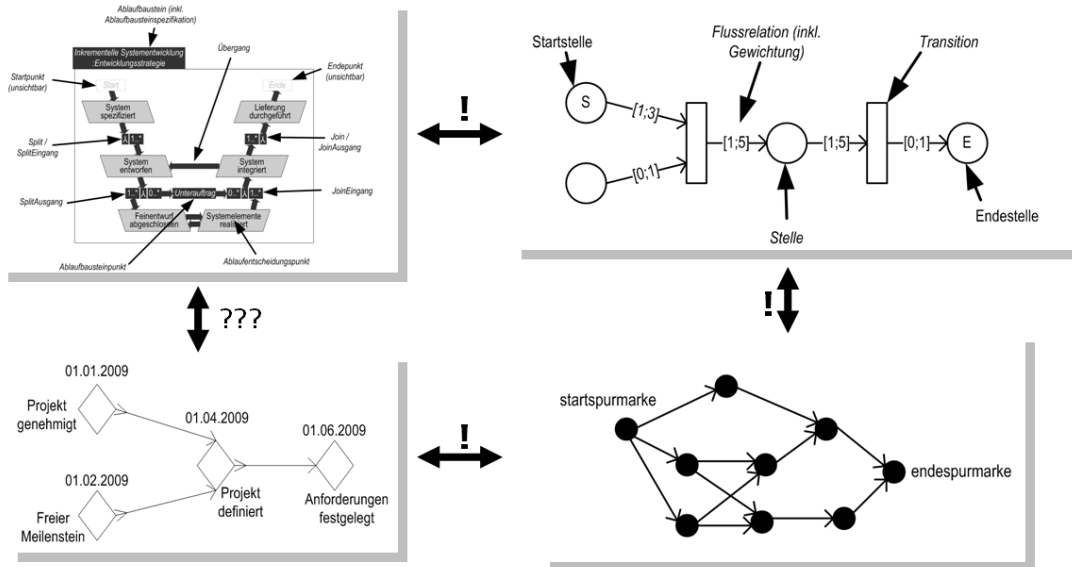


Abbildung 2.3.: Beziehungen zwischen Konstrukten der Semantik-Definition

2.3. Projektplanung im neuen V-Modell XT V1.3: Wie geht das denn genau?

Hier ist es Aufgabe der Werkzeughersteller, die Plan-Erzeugung auf Basis von Projektdurchführungsstrategien möglichst handhabbar und komfortabel umzusetzen. Zum jetzigen Zeitpunkt gibt es allerdings noch kein Werkzeug, das die gegebene Spezifikation in allen Einzelheiten implementiert. Der neue V-Modell XT Projektassistent erzeugt zwar ausschließlich Pläne, die der Spezifikation entsprechen, ist allerdings nicht in der Lage, zu jeder erlaubten Projektdurchführungsstrategie die Gesamtmenge solcher Pläne zu erzeugen.

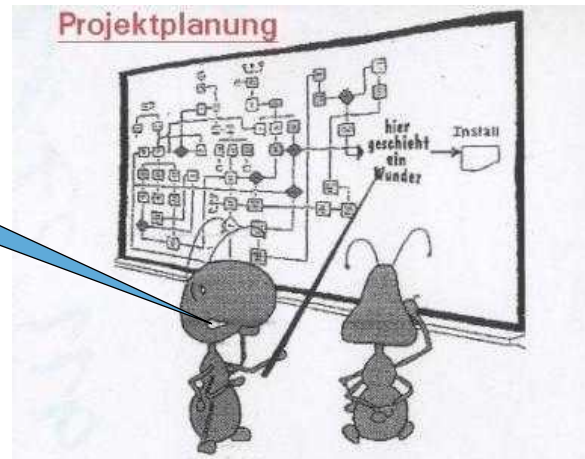
Literaturverzeichnis

1. Bundesrepublik Deutschland: V-Modell XT - Version 1.3: verfügbar unter www.v-modell-xt.de
2. Bergner, K.: Plangenerierung im V-Modell XT 1.2. IN (Hochberger, C.; Liskowsky R.; Hrsg.): Lecture Notes in Informatics, Proceedings Informatik 2006, Gesellschaft für In-formatik, Bonn, 2006
3. Höhn, R.; Höppner S.: Das V-Modell XT, Springer Verlag Berlin Heidelberg New York, 2008
4. Friedrich, J. et.al.: Das V-Modell XT - Für Projektleiter und QS-Verantwortliche kompakt und übersichtlich. Springer Berlin Heidelberg, 2008



Projektplanung mit dem neuen V-Modell XT V1.3

Wie geht das denn genau?

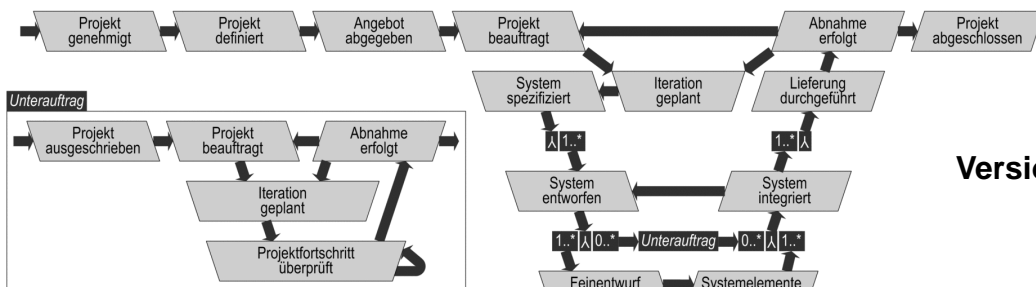
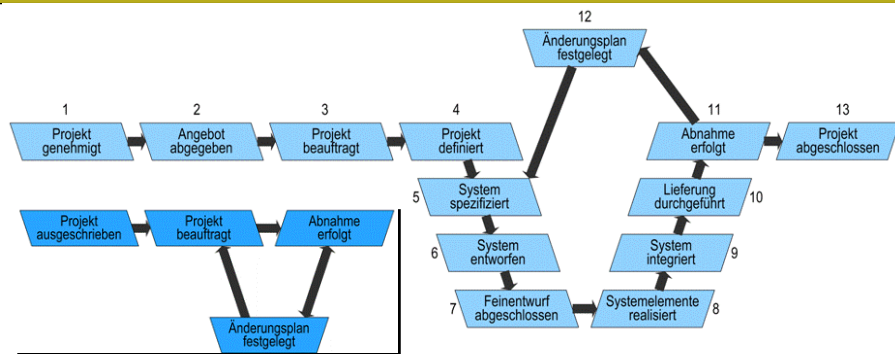


Dr. Klaus Bergner, Jan Friedrich
4Soft GmbH
SEE 2009, 26. Mai 2009

Die inkrementelle Strategie im Wandel der Zeit...



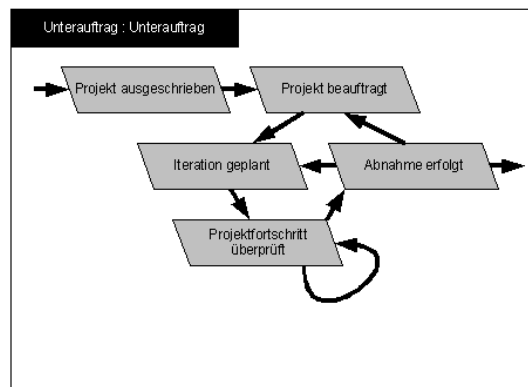
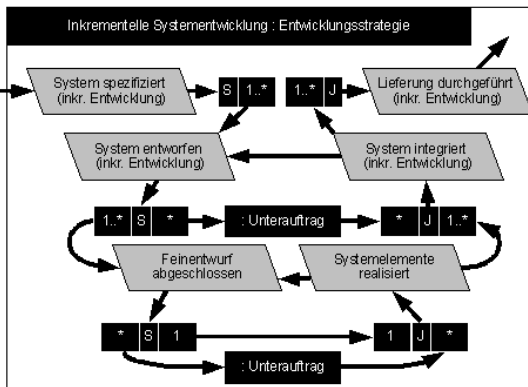
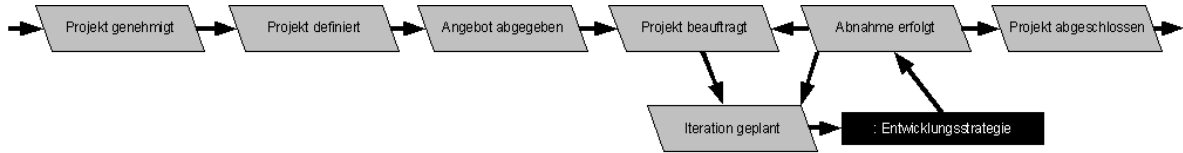
Version 1.0/1.1



Version 1.2

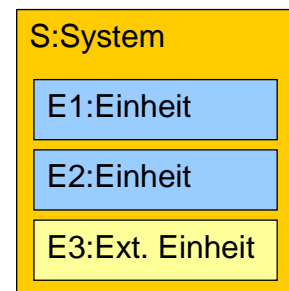
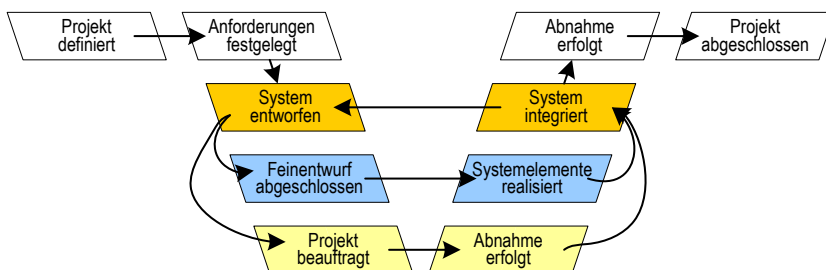
Die inkrementelle Strategie im Wandel der Zeit...

Version 1.3

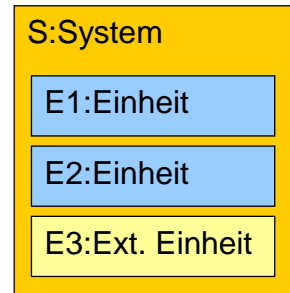
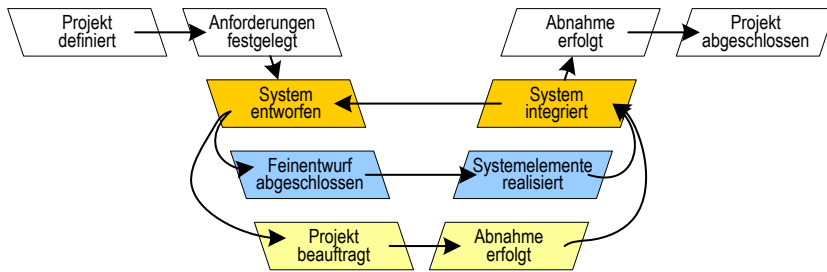


Einführung des Beispiels

- Vereinfachte Projektdurchführungsstrategie
 - Verkürzter Vorlauf
 - Iterationen nur auf Systemebene
 - Vereinfachter Unterauftrag
 - Kein Pflichtenheft, keine Auslieferung
- Einfaches Beispiel-System
 - Ein System
 - Zwei selbst entwickelte Einheiten (E1, E2)
 - Eine im Unterauftrag entwickelte Einheit (E3)

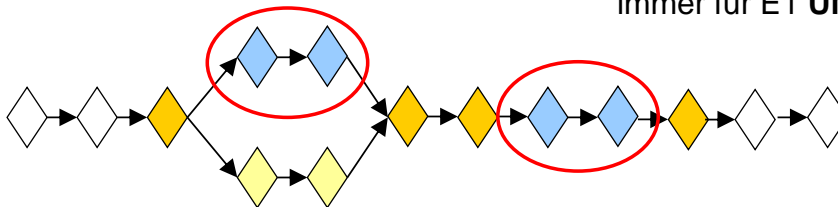


Wie war's im V-Modell XT V 1.0/1.1?

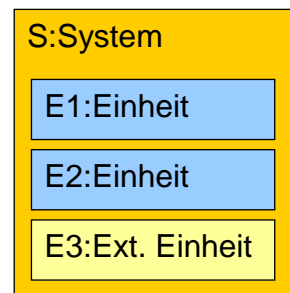
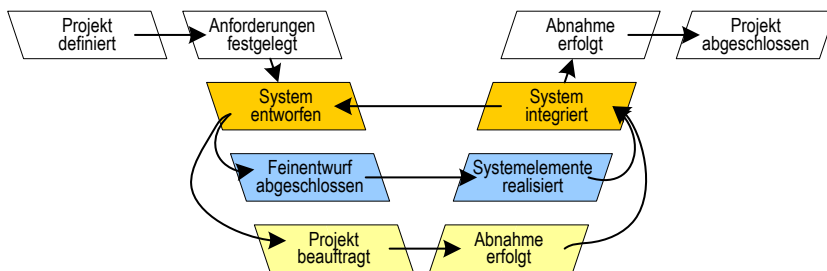


- Möglicher Projektablauf:

Problem: Hellblaue EPs gelten immer für E1 **UND** E2

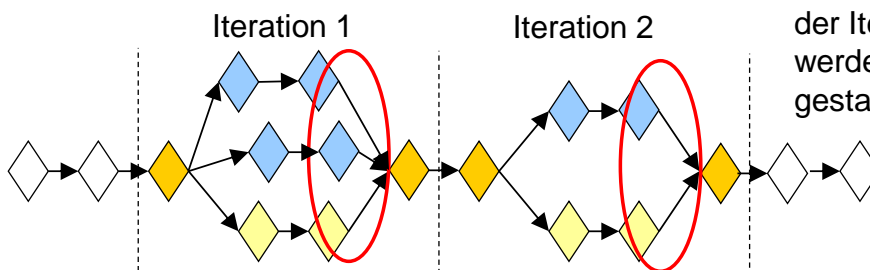


Wie war's im V-Modell XT V 1.2?



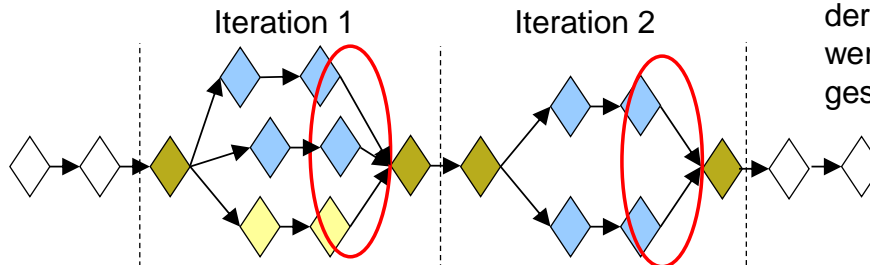
- Möglicher Projektablauf:

Problem: Alle Einheiten müssen in der Iteration fertig werden, in der sie gestartet werden.



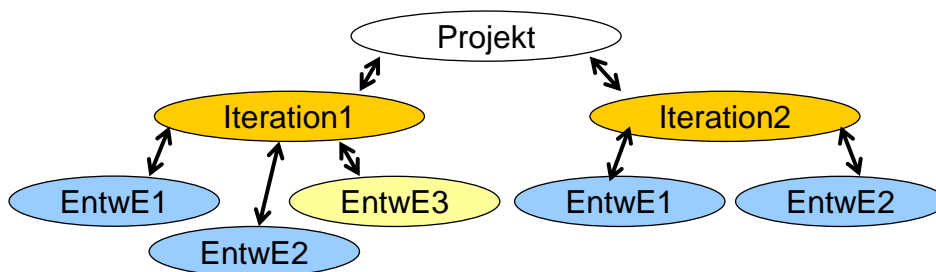
Wie war's im V-Modell XT V 1.2?

- Möglicher Projektablauf:

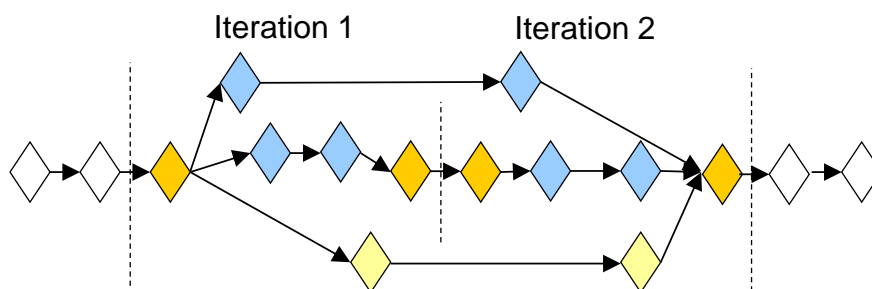


Problem: Alle Einheiten müssen in der Iteration fertig werden, in der sie gestartet werden.

- Ablaufstruktur:

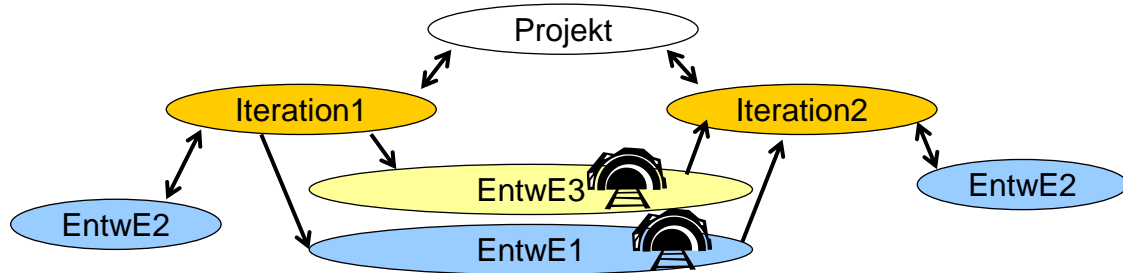


Was wollen wir noch?

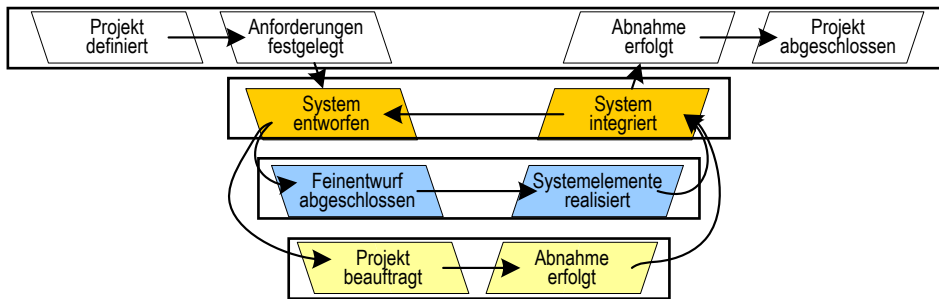


- Anwendungsfall: Nicht alle Einheiten benötigen die gleiche Zeit für die Entwicklung
- Einheiten sollen in einer Iteration gestartet, in einer späteren Iteration integriert werden. Dabei dürfen die beiden Iterationen unterschiedlichen Strategien folgen.
- → **Mit V-Modell XT Version 1.2 nicht möglich!**

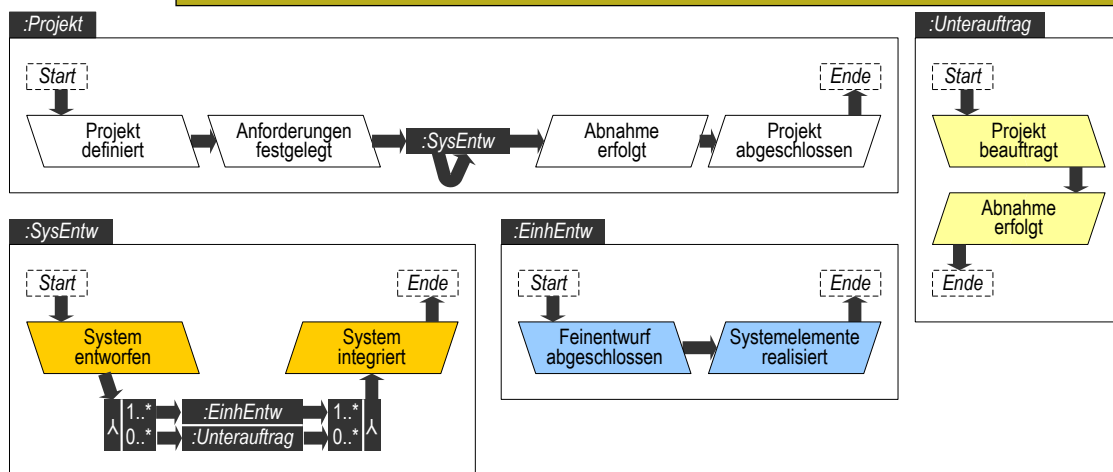
Wie sieht die Ablaufstruktur aus?



- Auf dieser Basis modularisieren wir die Projektdurchführungsstrategie...



Bildung von Ablaufbausteinen



- Analog zu den Vorgehensbausteinen gibt es im V-Modell XT jetzt **Ablaufbausteine**, die aufeinander aufbauen und im Tailoring ausgewählt werden.
- Das Tailoring definiert außerdem genau einen Ablaufbaustein als „**Wurzel**“

Und was läuft unter der Haube ab?



TUM
INSTITUT FÜR INFORMATIK

Modulare Spezifikation von Projektablaufen

Klaus Bergner, Jan Friedrich

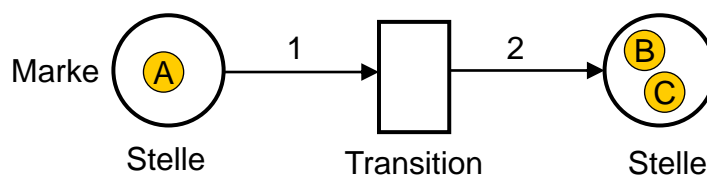


TUM-I0912
April 09

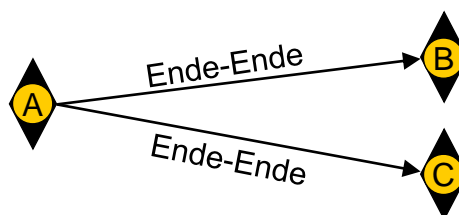
TECHNISCHE UNIVERSITÄT MÜNCHEN

Und wie kommt man jetzt zum Projektplan?

- Formal gesehen liegt hinter den Ablaufbausteinen ein Petri-Netz



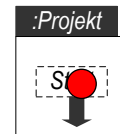
- Wir zeichnen den Weg der Marken durch das Petri-Netz auf und bilden daraus einen Projektplan



Das große Markenverschiebespiel: Die Spielregeln



1. Zu Beginn des Spiels liegt eine Marke auf dem Start des Wurzelbausteins



2. Marken können entlang der Pfeile verschoben werden



3. An Splitpunkten können aus einer Marke mehrere entstehen, entsprechend den Kardinalitäten



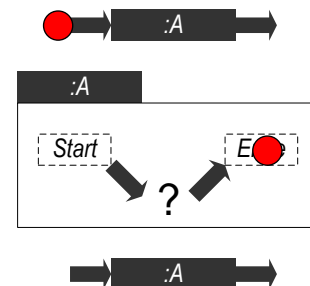
4. An Joinpunkten kann aus mehreren Marken eine einzelne Marke entstehen



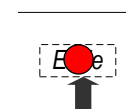
Das große Markenverschiebespiel: Die Spielregeln



5. Beim Betreten eines Ablaufbausteinpunktes wird die Marke auf den Start eines dazugehörigen Ablaufbausteins gestellt. Befindet sich eine Marke auf dem Ende eines Ablaufbausteins, darf sie diesen verlassen und von einem **beliebigen** dazugehörigen Ablaufbausteinpunkt weiterziehen.

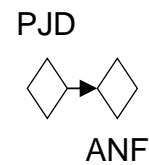
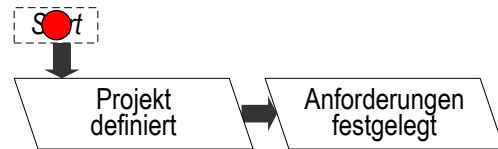


6. Am Ende des Spiels liegt genau eine Marke auf dem Ende des Wurzelbausteins. Es existieren keine weiteren Marken!

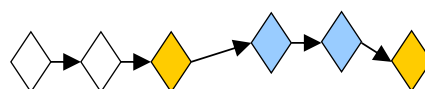
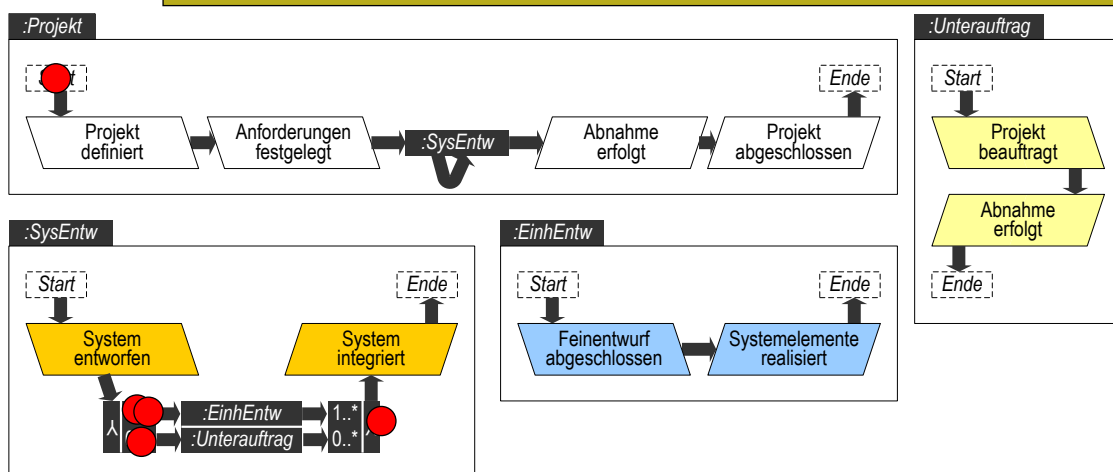


Das große Markenverschiebespiel: Die Spielregeln

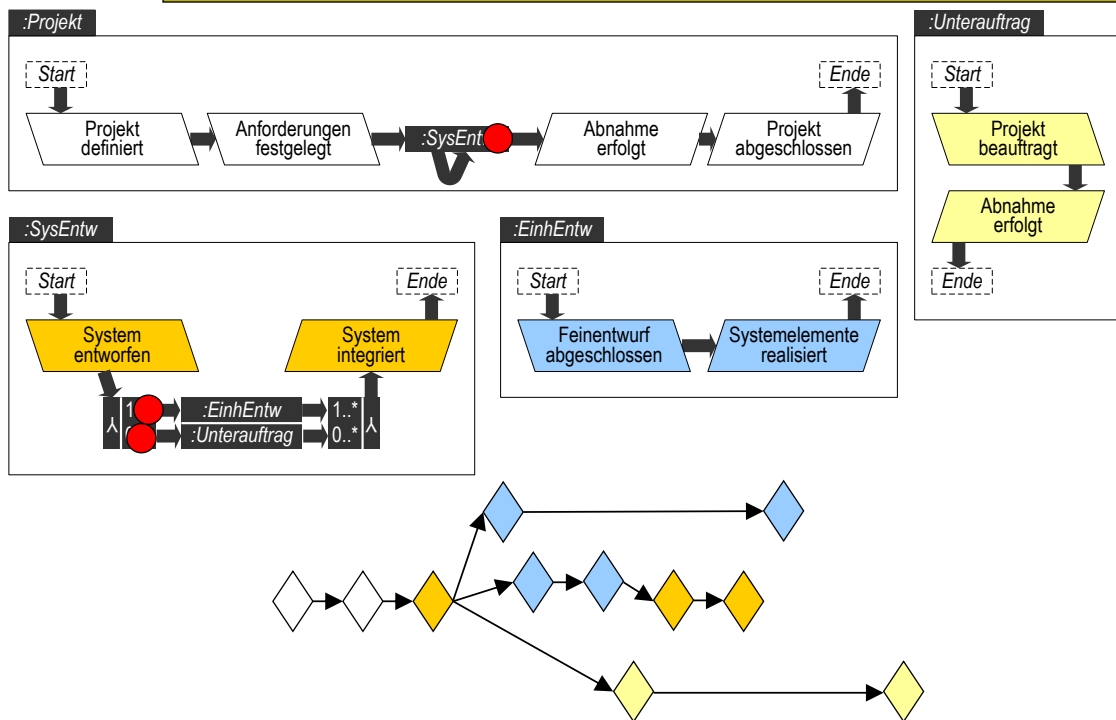
7. Passiert eine Marke einen Entscheidungspunkt, dann wird dafür ein Meilenstein im Projektplan angelegt. Eine Marke „merkt sich“ auf welchen Entscheidungspunkten sie zuvor stand und dementsprechend werden Ende-zu-Ende-Abhängigkeiten zwischen den Meilensteinen angelegt.



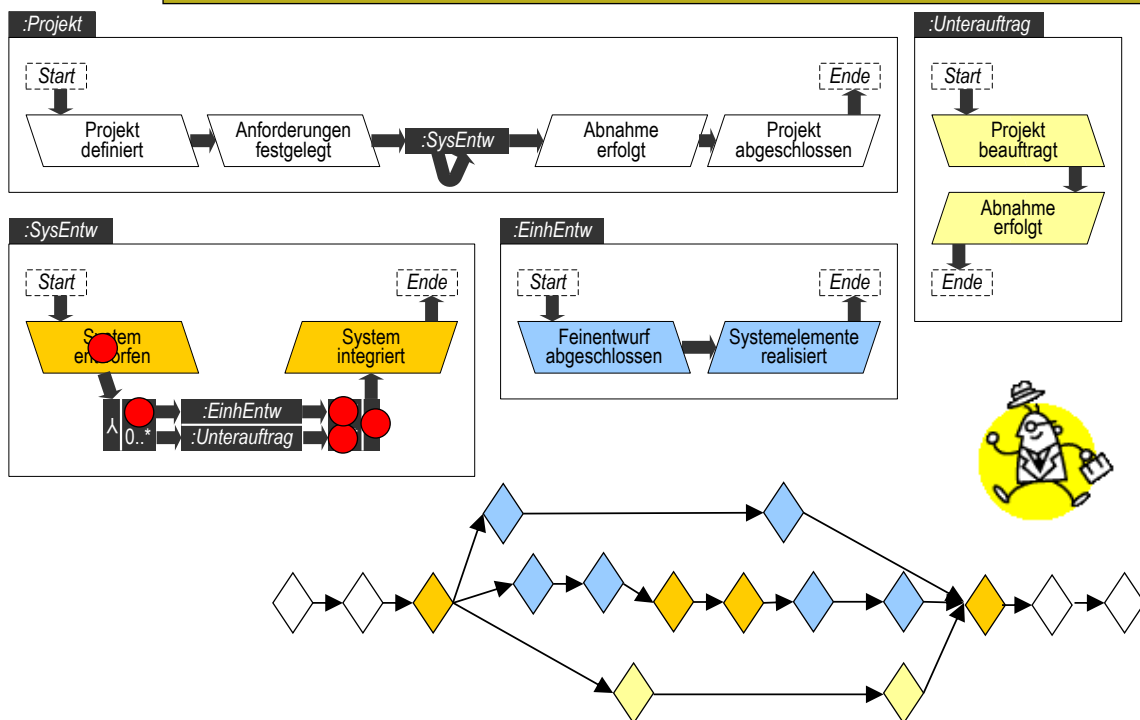
Vorwärtsplanung im Detail (1/3)



Vorwärtsplanung im Detail (2/3)

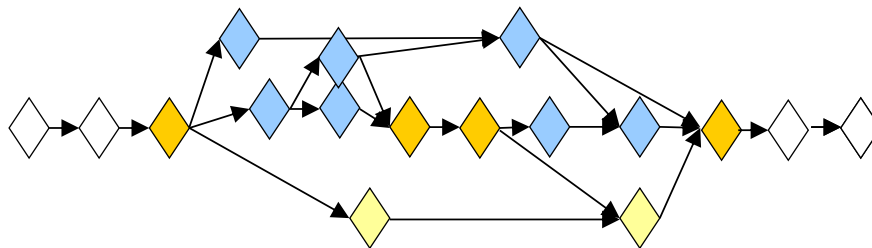


Vorwärtsplanung im Detail (3/3)



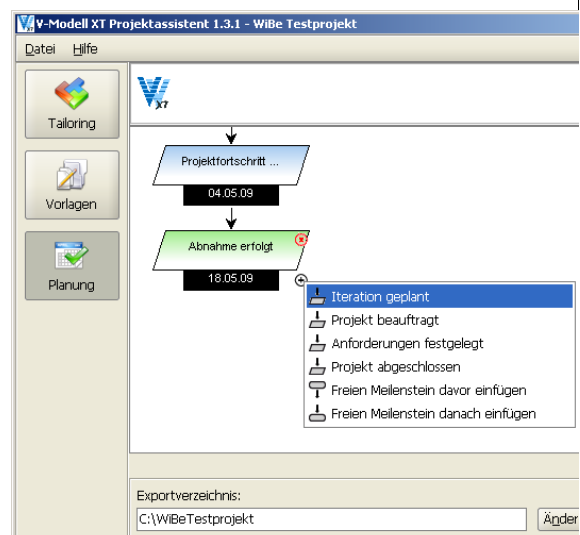
Zusatzregeln für den Projektleiter

- Der Projektleiter darf unabhängige (parallele) Meilensteine zusammenfassen.
- Der Projektleiter darf über das Vorgehensmodell hinaus zusätzliche Restriktionen auf seinem Projektplan einführen. Der Projektplan darf also immer „strenger“ sein als das Vorgehensmodell.
- Der Projektleiter darf zusätzliche „freie“ Meilensteine einfügen.



Werkzeuglandschaft

- V-Modell XT Projektassistent
 - Open-Source-Implementierung
 - Erzeugt nur gültige Projektpläne
 - **Aber:** nicht alle gültigen Projektpläne können damit erzeugt werden
 - Probleme: „Strategie-Sprung“ und Zusatzregeln für Projektleiter.
- Kommerzielle Software
 - microTOOL arbeitet an neuer in-Step-Version: Implementierung der Projektplanung vorgesehen. → Abwarten...
 - Kein anderer Werkzeughersteller bekannt.





2.3. Projektplanung im neuen V-Modell XT V1.3: Wie geht das denn genau?

3. Session 3: Werkzeuge zur Kollaboration in Projekten

Inhalt

3.1. Koordinierte Softwareentwicklung im Team mittels Jazz	98
3.2. Evaluierung von Werkzeugen für Distributed Pair Programming: Eine Fallstudie	111
3.3. Flexible Prozess-Werkzeug Integration	123

3.1. Koordinierte Softwareentwicklung im Team mittels Jazz

Michael Müller¹, Veronika Thurner², Martin Wassermann¹

¹ARS Computer und Consulting GmbH Ridlerstraße 55 80339 München
{fmichael.mueller | martin.wassermann}@ars.de

²Hochschule München Lothstraße 64 80335 München {thurner}@hm.edu

Motivation

Da die Anforderungen an Softwaresysteme immer umfangreicher und komplexer werden, wird Software heute fast ausschließlich in Teams entwickelt. Mit steigender Größe sowohl der Projekte als auch der Entwicklungsteams wird die effektive und reibungslose Koordination der Zusammenarbeit aller Projektbeteiligten dabei zum zentralen Faktor für die erfolgreiche Projektdurchführung.

Integrierte Entwicklungsumgebungen (IDEs) verbessern zwar in hohem Maße die individuelle Produktivität der einzelnen Projektbeteiligten, unterstützen jedoch die Zusammenarbeit im Team nur rudimentär, beispielsweise durch Versionsverwaltung der einzelnen Entwicklungsartefakte. Ergänzend sind Werkzeuge und Mechanismen erforderlich, welche die Zusammenarbeit im Team planen und koordinieren und so die Produktivität des gesamten Teams verbessern, die einzelnen Teammitglieder dabei jedoch so wenig wie möglich einschränken und nicht unnötig mit Verwaltungstätigkeiten belasten.

Jazz als Plattform für koordinierte Softwareentwicklung

Jazz [2] ist eine Plattform, welche die Zusammenarbeit im Team über den gesamten Softwareentwicklungszyklus hinweg auf aufgabenorientierte Weise unterstützt. Grundlage der Jazz-Plattform ist der von IBM entwickelte Jazz Team Server. Dieser stützt sich ab auf eine Datenbank, in der das Jazz-Repository abgelegt wird. Für den Zugriff auf das Jazz-Repository stellt der Jazz Team Server verschiedene Dienste zur Verfügung, die von entsprechenden Jazz Clients aus genutzt werden können.

Die Inhalte des Repositories sind gegliedert in Project Areas, in denen jeweils die zu einem Jazz-Projekt gehörenden Artefakte abgelegt sind. Jedem Projekt ist dabei ein Prozess zugeordnet, der die Abwicklung des Projektes steuert und dessen Rollenmodell, Koordinationsstrukturen und Iterationsplanung über Jazz abgebildet werden.

Die in einem Projekt durchzuführenden Arbeitsaufgaben werden in Form von Work Items beschrieben. Welche Arten von Work Items verfügbar sind wird durch die zugrunde liegende Prozessdefinition bestimmt, ebenso wie die zugehörigen Eigenschaften und Zustandsmodelle. Bei Bedarf können diese Vorgaben angepasst und erweitert werden [1].

Projekterfahrung

Um dieses noch junge Werkzeug frühzeitig in der Praxis zu erproben hat die ARS Computer und Consulting GmbH ein bereits laufendes Projekt zur Entwicklung eines neuen Internet-Auftritts incl. Content und der zugehörigen Webapplikation ab Frühjahr 2008

mittels Jazz abgewickelt. Dabei wurde nicht nur die Zusammenarbeit des Kernentwicklungsteams (bestehend aus Projektleiter, Entwicklern, und Testern) der Webapplikation über Jazz koordiniert, sondern auch die Auftraggeber, die Nutzer der Applikation und die Verantwortlichen für den Content des Webauftritts über Jazz in den Entwicklungsprozess mit eingebunden.

Konfiguration von Jazz für ein konkretes Projekt

Damit die Vorteile von Jazz in einem konkreten Projekt voll zum Tragen kommen muss das Werkzeug zunächst an die Bedürfnisse dieses Projektes angepasst werden. Die Konfigurationseinstellungen sind in Form von XML-Dateien abgelegt, die über geeignete Masken mit dem Jazz Eclipse Client bearbeitet werden können [3]. Ausgehend von einem bereits in Jazz vordefinierten agilen Prozessmodell wurde für dieses Projekt zunächst ein Rollenmodell und deren zugehörige Sichten definiert und mit geeigneten Berechtigungen hinterlegt.

Im nächsten Schritt wurden die vordefinierten Work-Item-Typen des zugrunde liegenden agilen Prozessmodells an die Bedürfnisse des Projektes angepasst und um selbstdefinierte Work-Item-Typen erweitert. Folgende Work-Item-Typen wurden in diesem Projekt verwendet:

- Story bündelt mehrere inhaltlich zusammenhängende Work Items zu einer Einheit und veranschaulicht so den groben Kontext der einzelnen Work Items.
- Concept dokumentiert Entwurfsaktivitäten und die dabei erzielten Entscheidungen.
- Task bezeichnet einen Einzelschritt in einer Neuentwicklung.
- Enhancement fokussiert die Weiterentwicklung eines bestehenden Artefaktes.
- Defect beschreibt eine Tätigkeit zur Fehlerkorrektur.
- Test definiert Aktivitäten zur Qualitätssicherung.

Ergänzend wird für jeden Work-Item-Typ ein Zustandsmodell definiert, das die möglichen Bearbeitungsstände eines konkreten Work Items dieses Typs beschreibt sowie Übergänge zwischen diesen Zuständen festlegt. Des Weiteren werden in einer Auswahlliste so genannte Resolutions gesammelt, d.h. mögliche Gründe für das Eintreten eines Work Items in einen bestimmten Zustand. Geht beispielsweise ein konkreter Work Item in den Zustand „abgeschlossen“ bzw. „closed“ über, so wird über eine solche Resolution ausgedrückt, warum der Work Item nicht weiter verfolgt werden muss.

Soll das Projekt als iterative Entwicklung realisiert werden, lassen sich ferner auch Typen für Iterationen definieren. Für jeden dieser Typen können Regeln angegeben werden, welche die in einer solchen Iteration möglichen Aktionen einschränken. Wurden beispielsweise die Iterationstypen „Entwicklung“ und „Stabilisierung“ definiert, so könnte über die Regeln festgelegt werden, dass einer Stabilisierungsiteration keine neuen Arbeitsaufträge und damit Work Items zugewiesen werden dürfen, ausgenommen Work Items vom Typ „Defect“ zur Fehlerkorrektur. Ebenfalls lässt sich regeln, dass in einer Stabilisierungsiteration Quelltext nur dann ins Projekt-Repository eingepflegt werden darf, wenn er zuvor nach bestimmten Mechanismen erfolgreich qualitätsgesichert wurde. Die definierten Konfigurationseinstellungen können bei Bedarf auch während eines laufenden Projektes modifiziert werden. Darüber hinaus ist es möglich, eine einmal definierte Projektkonfiguration auch auf nachfolgende Projekte zu übertragen.

Produktiver Einsatz von Jazz im laufenden Projekt

Nachdem Jazz für die spezifischen Bedürfnisse eines konkreten Projektes konfiguriert wurde und somit alle Muster und Schemata für Planungs- bzw. Kommunikationsstrukturen und Verwaltungsartefakte von Jazz angepasst wurden, kann Jazz im Projekt produktiv eingesetzt werden.

Da das Projekt bei der Umstellung auf Jazz bereits seit gut einem halben Jahr lief, wurden zunächst laufende bzw. geplante Entwicklungstätigkeiten als Work Items in Jazz angelegt. Relevante zugehörige Dokumente und Kommunikationshistorien, die bisher teilweise in E-Mails abgelegt bzw. als Textdateien in die Source-Code-Verwaltung eingebunden waren, wurden soweit nötig in Jazz übertragen und mit den entsprechenden Work Items verknüpft. Diese initiale Menge von Work Items wurde im weiteren Verlauf des Projektes bei Bedarf um neue Work Items erweitert, beispielsweise um Tests und QS-Maßnahmen für bestehende Entwicklungsartefakte, die Behandlung von identifizierten Fehlern oder die Realisierung neuer Anforderungen.

Um die zeitliche Reihenfolge der Abarbeitung von Work Items systematisch zu planen und zu verfolgen werden zu erledigende Work Items einer Iteration zugewiesen, in der sie zu realisieren sind. Im Testprojekt wurden in einer initialen Planung die wichtigsten laufenden Work Items grob den nächsten Iterationen zugewiesen. Zu Beginn der zweiten Hälfte jeder Iteration wurde dann die unmittelbar nachfolgende Iteration anhand der aktuell vorhandenen Menge von Work Items im Detail und die nachfolgenden Iterationen grob vorgeplant.

Fazit und Ausblick

Die Beteiligten des Testprojektes haben Jazz durchwegs sehr positiv beurteilt, unabhängig von ihrer Rolle im Projekt. Als Hauptnutzen des Einsatzes von Jazz wurde der damit erreichte verbesserte Überblick über den aktuellen Stand des Projektes und die als nächstes anstehenden Tätigkeiten gewertet, ebenso wie eine verbesserte Team Awareness. Darüber hinaus ermöglichen die enge Verzahnung von Planung und Vorgehensweise und die Integration von technischen und planerischen Artefakten in ein einziges Werkzeug eine weit gehend automatische Steuerung entsprechender Workflows, wodurch Zeit- und Aufgabenplanung ebenso wie die Fortschrittskontrolle erheblich einfacher und effektiver werden.

Auch die Kommunikation zu einzelnen Aufgaben im Gesamtprojekt wird durch Jazz signifikant unterstützt. Insbesondere ist es möglich, per Mail, Chat oder über sonstige Dokumente geführte Diskussionen zu einem einzelnen Work Item systematisch mit diesem abzulegen und zu verwalten.

Da sich Jazz bei ARS in einem ersten Praxiseinsatz bewährt hat, werden zukünftig auch weitere Projekte die Jazz-Plattform zur Koordination der Teamarbeit nutzen. Dabei sollen nicht nur die bereits in Jazz vordefinierten Prozess-Templates genutzt werden, sondern je nach Bedarf vermehrt auch weitere Vorgehensmodelle auf Jazz abgebildet werden. Exemplarisch wird derzeit das V-Modell XT auf Jazz abgebildet. Begleitend ist die Entwicklung eines generischen Leitfadens in Arbeit, der eine Vorgehensweise für die systematische Abbildung eines beliebigen Vorgehensmodell auf die technologischen Möglichkeiten von Jazz beschreibt.

Literatur

1. IBM Rational. Get started with Jazz... <https://jazz.net/learn/LearnItem.jsp?href=gettingstarted.html> (Registrierung erforderlich), Dezember 2008
2. IBM Rational. Jazz Community Site. www.jazz.net, Dezember 2008
3. Process Team, Jazz Community Site. Team Process Developer Guide. <https://jazz.net/wiki/bin/view/Main/TeamProcessDeveloperGuide> (Registrierung erforderlich), November 2007

Koordinierte Softwareentwicklung im Team mittels Jazz

SEE 2009 in Berlin

Michael Müller, ARS Computer und Consulting GmbH, www.ars.de

Motivation

Situation in der Softwareentwicklung, insb. im eclipse Umfeld bisher:

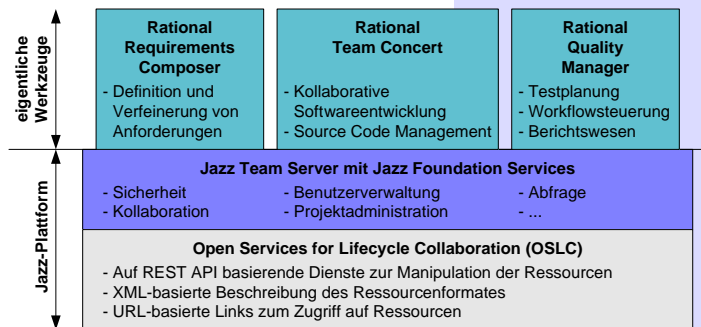
- Fokus auf Produktivität des einzelnen Entwicklers gesetzt
- Team Aspekte abgebildet auf
 - SCM (Team-Provider)
 - Plug-ins für Sicht auf externe Tools
 - Ansätze im Task-Management (Mylyn)
- Kommunikation im Team über
 - Quelltext
 - Commit Kommentare
 - Bug-Tracking Tool
 - Email, Instant Messaging, Telefon, Meetings

Zielsetzung:

- Integration aller obigen Punkte
- Fokus auf die Produktivität des Teams

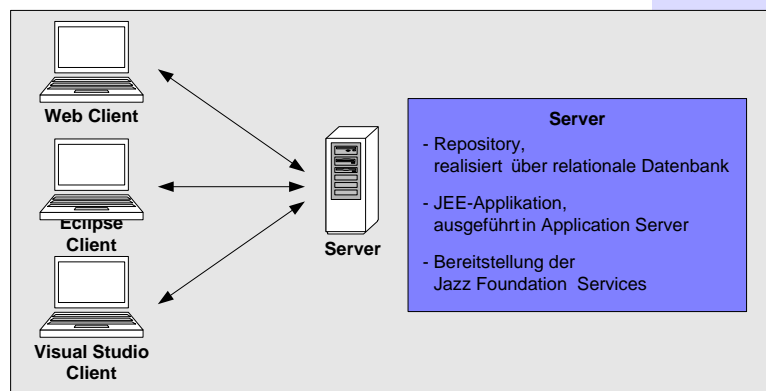
Logische Architektur

- Jazz Foundation
 - Open Services for Lifecycle Collaboration
 - Basisdienste für Werkzeuge zum Austausch von Team Artefakten
 - Gemeinsames Verständnis der Beziehungen zwischen Ressourcen eines Lebenszyklus
 - Klare detaillierte Grundlage über die Informationen einer Ressource, z.B. Use Case
 - Jazz Integrationsarchitektur
 - Jazz Foundation Services
 - Werkzeug Hersteller stellen spezifische Funktionalität zur Verfügung
- Jazz basierte Produkte
 - Rational Team Concert (RTC)
 - Rational Quality Manager (RQM)
 - Rational Requirements Composer (RRC)

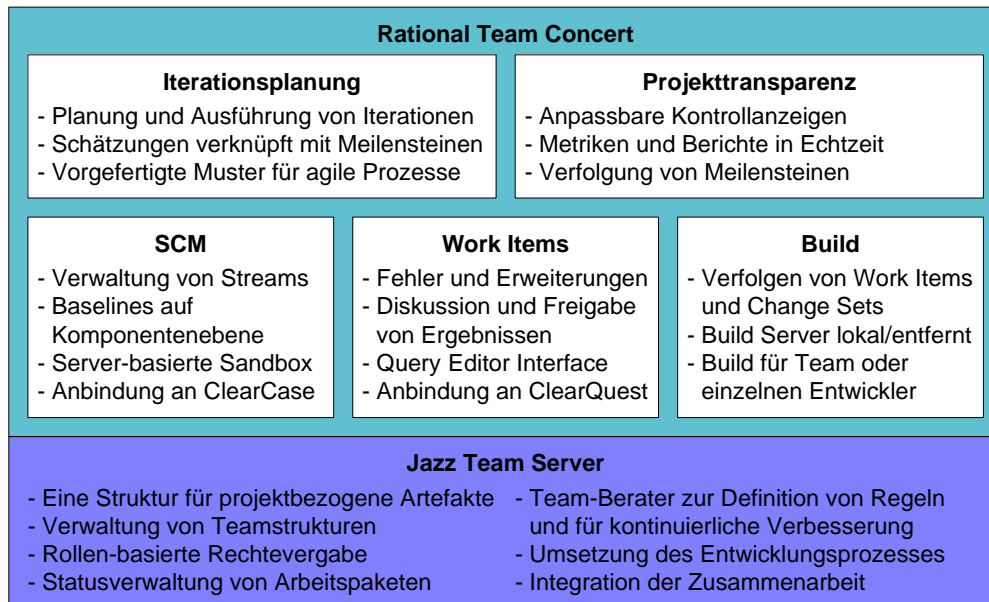


Physikalische Architektur

- Server
 - DB zur Ablage aller Daten, incl. Quelltextverwaltung
 - JavaEE Applikation
 - Services
- Client
 - Web basiert
 - Eclipse
 - Visual Studio



Rational Team Concert - Building Blocks



Methodisches Vorgehen Philosophie der Nutzung

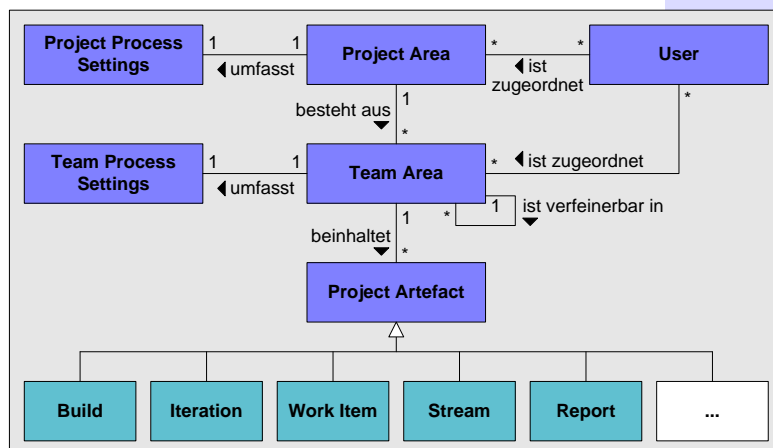
- **Gamma:** *"Jazz helps teams work together more effectively"*
- **Prozessunterstützung**
 - Entwicklungsprozesse und deren Methodiken
 - "so wenig wie möglich, so viel wie nötig"
- **Global Transparency**
 - Projektstand von jedem einsehbar
 - Status immer Zeitaktuell
- **Team Kollaboration im Kontext**
 - Ereignisse und Neuigkeiten im Projekt (auch als RSS Feed)
 - Codeänderungen des Teams
 - Statusänderungen
 - Continuous Integration, Build Health
 - Teammitglieder und deren Onlinestatus

Methodisches Vorgehen Vordefinierte Prozessmuster

- Verwendung mitgelieferter Prozessmuster
 - Agile Process
 - Scrum
 - OpenUP
 - Eclipse Way
- Definition eigener Prozesse
 - Auf bestehende aufsetzen und anpassen
 - Mit Rational Method Composer eigene definieren

Aufsetzen des Projekts Projekt- und Teamstruktur

- Ein Projekt besteht aus einem oder mehreren Teams
 - Hierarchischer Aufbau
 - Teammitglieder können in mehreren Teams mit unterschiedlichen Rollen sein
- Jedes Team kann Prozessregeln anpassen



Aufsetzen des Projekts Iterationsplanung

- Iterationen werden in einer Project Area definiert
- Ein Projekte kann mehrere Development Lines haben
 - Bsp: Development Release 2.0 und Maintenance 1.0.1
- In einer Development Line können Iterationen definiert werden
 - Hierarchischer Aufbau möglich
 - Nur eine Iteration einer Development Line aktiv
- Mehrere Iterationstypen möglich
 - Pro Iterationstyp unterschiedliche Prozessregeln verwendbar
 - Beispielsweise: Development Phase / Stabilization Phase
- Planung der Iteration
 - Zeiträume
 - Inhalte der Iteration durch Zuordnung von Workitems

Dynamische Alltagsarbeit Workitems

- Erstellung von Workitems
 - Beschreibung (Subject und Description)
 - Verknüpfung von Workitems
 - child/parent, depends, blocks, duplicate, related, ...
 - Zuordnen der Workitems
 - Team
 - User
 - Iteration
 - ...
 - Festhalten der geschätzten Aufwände
 - Weitere Attribute (Priority, Severity, ...)
- Workitems unterliegen einem Workflow
 - Startzustand
 - Zustandsübergang
 - Endzustand/Auflösung
 - Abhängig vom Typ des Workitems



Transitions		Deferred	Done	In Progress	Not Done	Ready for Review
Deferred	To	<None>	None	None	Reopen	None
Done	From	None	<None>	None	Reopen	Ready for Review
In Progress	From	Defer	None	<None>	None	Ready for Review
Not Done	From	Defer	None	Start Working	<None>	None
Ready for Review	From	Defer	Done	Reiterate	Reopen	<None>

Dynamische Alltagsarbeit Aufgabenplanung, Termine und Aufwände erfassen 2/2

ARS

SEE 2009
Berlin

- Zuordnung der Aufgaben an Teammitglieder unter Berücksichtigung:
 - Bereits zugeordnete Workitems
 - Termine
 - Arbeitszeiten und blockierter Zeiten der einzelnen User
- Festhalten der tatsächlichen Aufwände
 - Allerdings keine Protokollierung
 - Grundlage für Auswertungen

Task	Assignee	Estimate	Priority	Status
Dynamisches Menü mit CSS oder JS	Christina Zahn	135	High	Open
Einblenden von Logo Grafiken auf Webseiten aus einer Liste in Notes DB - Eigenschaft vererbt / per Notes Dokument ändern	Christina Zahn	95	High	Open
Layout umsetzen	Christina Zahn	53	High	Open
Layout im Schulungsbereich	Christina Zahn	121	Medium	Open
Farbänderung	Christina Zahn	100	Medium	Open
Änderung des Fly-Over und der Hintergrundfarbe im Navigator links und oben	Christina Zahn	96	Medium	Open
Imagemaps aus dem Schulungsbereich werden im IE bei zu geringer Breite nach unten geschoben	Christina Zahn	144	Low	Open
Mauszeiger im IE wird bei Links im Menü nicht korrekt angezeigt	Christina Zahn	120	Low	Open
Martin Wassermann				
Abgeladene News nicht einfach löschen sondern in einer Art Archiv vorhalten	Martin Wassermann	129	High	Open
Build Files erstellen	Martin Wassermann	122	High	Open
Build File für den ShortCutParser erstellen	Martin Wassermann	136	Medium	Open
Zerkerbruch innerhalb eines Paragraphen soll möglich sein.	Martin Wassermann	130	Medium	Open
Fehleranalyse mit automatischer Benachrichtigung	Martin Wassermann	20	Medium	Open
ARSWebLanguage	Martin Wassermann	16	Low	Open

© ARS Computer und Consulting GmbH 2009, www.ars.de

11

Dynamische Alltagsarbeit Abhängigkeiten, Zustände verfolgen

ARS

SEE 2009
Berlin

Workitem Queries und Reports

- Abfragesprache zum Suchen nach Aufgaben
 - Team Queries
 - » Vordefinierte
 - » Individuelle
 - Persönliche
 - Komfortabler Query Builder zum Erstellen der WI Queries
- Reports
 - Darstellung der Projektsituation
 - ggf. auch Prozessabhängige Reports
 - Eigene Reports via BIRT realisierbar

© ARS Computer und Consulting GmbH 2009, www.ars.de

12

Dynamische Alltagsarbeit Flexibilität und Anpassungsmöglichkeiten

ARS

SEE 2009
Berlin

Allgemein

- Anpassungen jederzeit vornehmbar
- Auf verschiedenen Ebenen (Projekt, Team, Iterationstyp, Development Lines, Iteration)
- Individuell für einzelne Teams

Definition eigener Workitem Typen

- Concept, Story, Task, Defect, Enhancement,...
- Erweiterung um eigene Attribute

Workflow Änderungen

- Zustände
- Übergänge
- Abhängig von Bestätigungen (Approvals)

Bedingungen für Prozessaktionen

- Vorbedingungen, Nachgelagerte Aktionen
- Permissions

© ARS Computer und Consulting GmbH 2009, www.ars.de

13

ARS

SEE 2009
Berlin

Bewertung

Positiv

- Einfacher Start
- Sehr gute Eclipse Integration
- Hohe Flexibilität
 - Dynamische Prozessanpassung
 - Workitem Attribute
- "Open Commercial Development", Transparenz für Jazz/RTC selbst
- Integrationsplattform

Negativ

- Repository Management monolithisch
- Prozessregeln "aushebelbar"
- Lesezugriff nicht einschränkbar => RTC 2.0
- Rechtesystem bei Workflows nicht gegeben => RTC 2.0
- Leider kein Open Source :-)

© ARS Computer und Consulting GmbH 2009, www.ars.de

14

Ausblick

More to come...

- Bereits verfügbare Produkte
 - Rational Requirements Composer
 - Rational Quality Manager
- Rational Tool Landschaft zukünftig Jazz basiert
 - Rational ClearCase
 - Rational ClearQuest
 - Rational BuildForge
 - ...
- RTC Enterprise Edition für große Nutzerzahlen (> 250)
- Projektmanagement Werkzeug (Tara)
- Integrationsmöglichkeiten für andere Hersteller
 - bsp. Mainsoft Document Collaboration for Rational Jazz

© ARS Computer und Consulting GmbH 2009, www.ars.de

15

Demo

Demo

© ARS Computer und Consulting GmbH 2009, www.ars.de

16

Kontaktinformation

**Für Rückfragen stehen wir Ihnen
gerne zur Verfügung.**

Kontakt

- ARS Computer und Consulting GmbH
Ridlerstraße 55
80339 München
- <http://www.ars.de>
- Michael Müller
michael.mueller@ars.de
089/32468-170



3.2. Evaluierung von Werkzeugen für Distributed Pair Programming: Eine Fallstudie

Dietmar Winkler, Stefan Biffel

Technische Universität Wien

Institut für Softwaretechnik und Interaktive Systeme

Favoritenstrasse 9/188, A-1040 Wien

{Dietmar.Winkler | Stefan.Biffel}@tuwien.ac.at

Abstract

Ein zentrales Ziel der industriellen Software Entwicklung ist die Herstellung qualitativ hochwertiger Produkte innerhalb definierter Zeit- und Kostengrenzen (möglichst kurze Entwicklungszeiten bei möglichst kostengünstiger Herstellung) sowie eine schnelle Reaktionsfähigkeit auf geänderte Kundenanforderungen. Dieser Trend führt dazu, (a) Vorteile agiler Techniken in der Softwareentwicklung nutzen und (b) die Softwareentwicklung zu „globalisieren“ und „zu verteilen“. Diese zunehmende Globalisierung in der Softwareentwicklung erfordert eine effiziente und flexible Vorgehensweise, eine strukturierte Planung und eine enge Zusammenarbeit aller Mitglieder eines Entwicklungsteams. Agile Prozesse, wie beispielsweise Scrum oder eXtreme Programming (XP), bieten diese Flexibilität, in dem durch engen Kundenkontakt rasch auf geänderte Anforderungen reagiert werden kann. Agile Praktiken, wie beispielsweise Pair Programming (PP), ermöglichen eine effektive und effiziente Entwicklung von qualitativ hochwertigen Softwareprodukten. Ein Pair, bestehend aus einem Driver und einem Observer, arbeitet gleichzeitig am selben Produkt und nutzt dieselbe Entwicklungsumgebung. Während der Driver die Entwicklungsarbeit, z.B. eine Implementierungsaufgabe, ausführt, wird er durch den Observer unterstützt, der z.B. Qualitätssicherungsaufgaben, wie die Durchführung kontinuierlicher Reviews, wahrnimmt. Diese Rollenverteilung kann zyklisch wechseln. Vorteile von PP sind beispielsweise (a) gesteigerte Produktivität, (b) verbesserte Qualität der Produkte und (c) erhöhte Flexibilität durch die Einbindung in einen agilen Prozess. Weiters bietet PP eine effiziente Möglichkeit, neue Teammitglieder in ein Projekt einzubinden (PP als Lernmethodik).

Um diese Vorteile agiler Prozesse und Praktiken in einem globalen und verteilten Umfeld nutzen zu können, sind effektive und effiziente Methoden und Werkzeuge erforderlich, die einerseits die Stärken der Ägilität unterstützen und andererseits die räumliche Trennung der Teammitglieder quasi aufheben. In der Praxis existieren zahlreiche Kollaborations-Werkzeuge, die unterschiedliche (Teil-)Aufgaben, wie z.B. Tools für die Unterstützung der Kommunikation oder Document Sharing, unterstützen können. Die Auswahl geeigneter Tools im jeweiligen Unternehmens- oder Projektkontext erfolgt in der Regel durch Ausprobieren oder aus Erfahrungsberichten. Um passende Werkzeuge zu finden, ist eine systematische Evaluierung der in Frage kommenden Werkzeuge erforderlich.

Der Einsatz geeigneter Tools muss die Bedürfnisse der Anwender - in diesem Fall ein global verteiltes Pair - angepasst sein. Daher ist es notwendig, (a) einen Überblick über die erforderlichen Anforderungen an das Werkzeug im jeweiligen Anwendungskontext zu ermitteln, (b) ein effizientes Evaluierungsframework für die systematische Bewertung der Werkzeuge zu erstellen und dieses (c) auf eine Auswahl möglicher Werkzeuge anzuwenden. Die Anwendung dieses Evaluierungsframeworks ermöglicht (a) eine systema-

3.2. Evaluierung von Werkzeugen für Distributed Pair Programming: Eine Fallstudie

tische Untersuchung über die Eignung ausgewählter Werkzeuge und (b) eine Unterstützung bei der Auswahl eines möglichst passenden Werkzeugs im jeweiligen konkreten Anwendungsfall.

Diese Präsentation richtet sich an (a) Projektleiter zur Entscheidungsunterstützung bei der Auswahl geeigneter Werkzeuge in einem konkreten Anwendungsfall, (b) an Werkzeughersteller zur systematischen Sammlung von Anforderungen an eine Werkzeuglösung in einem konkreten Anwendungsbereich und kann somit als Ausgangsbasis für die Verbesserung und Erweiterung existierender Lösungen dienen. Ziele der Präsentation sind daher:

- Vorstellung einer generischen Vorgehensweise zur systematischen Evaluierung von Werkzeugen in einem definierten Kontext.
- Anwendung einer systematischen Werkzeug-Evaluierung am Fallbeispiel von Distributed Pair Programming (DPP):
 - Sammlung und Klassifikation von Anforderungen an ein Werkzeug zur Unterstützung verteilter Entwicklungen.
 - Erstellung eines konkreten Evaluierungsframeworks.
 - Durchführung der Evaluierung von möglichen Werkzeugen für die Unterstützung von Distributed Pair Programming.

Ergebnisse der Arbeit sind eine grundlegende Vorgehensweise, wie bei der Evaluierung von Werkzeugen vorgegangen werden kann sowie eine mögliche Unterstützung bei der verteilten Entwicklung von Softwareprodukten durch Distributed Pair Programming. Durch die Ergebnisse wird mögliches Verbesserungspotenzial von bestehenden Werkzeugen aufgezeigt, die zur Weiterentwicklung der untersuchten Tools verwendet werden können um Anforderungen der globalen und verteilten Entwicklung adressieren zu können.

Evaluierung von Werkzeugen für Distributed Pair Programming: Eine Fallstudie

Dietmar Winkler Stefan Biffli

Institute of Software Technology and Interactive Systems,
Vienna University of Technology

OCG Arbeitskreis „Software Prozesse“

dietmar.winkler@tuwien.ac.at

<http://qse.ifs.tuwien.ac.at>

.....
Institut für Softwaretechnik und Interaktive Systeme

Motivation

Herausforderungen in der modernen Softwareentwicklung:

- Erstellung von qualitativ hochwertigen Produkten in immer kürzerer Zeit (kurze Iterationen).
- Flexibilität im Hinblick auf rasch und häufig ändernde Anforderungen.
- Aus der Vielzahl von unterschiedlichen verfügbaren Werkzeugen gilt es, das am besten geeignete Werkzeug einzusetzen.

Diese Herausforderungen meistern durch ...

- Zunehmende globale Softwareentwicklung (24h Entwicklung).
- Einsatz von flexiblen und agilen Softwareprozessen (V-Modell XT, Agile Ansätze) zur kontrollierten Abwicklung der Entwicklung.
- Anwendung bewährter agiler Praktiken (z.B. Pair Programming).
- Distributed Pair Programming mit geeigneter Werkzeugunterstützung.

Ziele:

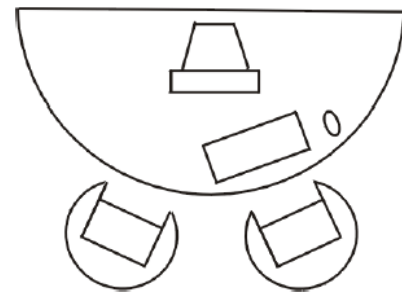
- Welche Anforderungen müssen diese Werkzeuge erfüllen?
- Wie kann eine Werkzeugevaluierung effizient durchgeführt werden?
- Welche Werkzeuge eignen sich speziell für Distributed Pair Programming?

.....
Institut für Softwaretechnik und Interaktive Systeme

Traditional Pair Programming



- Traditionelles Pair Programming ist eine etablierte agile Praxis für die effiziente Entwicklung eines Softwareproduktes.
- Eignung für **Codeentwicklung**, **Design**, **Algorithmen** und **Test** (“Paar-Aktivitäten”).
- Grundidee von Pair Programming:
 - 2 Entwickler (ein Pair) arbeiten **gleichzeitig an einem Softwareartefakt** und **teilen sich eine Arbeitsumgebung** (Bildschirm, Tastatur und Maus).
 - **Klare Rollenverteilung** mit wechselnder Rollenzuordnung.
 - **Driver**: Umsetzung der konkreten Aufgabe.
 - **Observer / Navigator**: Unterstützung des Drivers durch kontinuierliche Reviews, Fehlersuche, strategische Überlegungen für nächste Schritte.
- Nutzen von Pair Programming (aus emp. Studien):
 - Höhere **Qualität**, **Effektivität** und **Produktivität**.
 - **Verbesserte Kommunikation** im Team.
 - **Fokus auf dasselbe Artefakt** (“Pair Pressure”).
 - “**Learning in Pairs**”.
- Anwendbarkeit von Pair Programming in einer verteilten Umgebung?



Institut für Softwaretechnik und Interaktive Systeme

Distributed Pair Programming



- Die Nutzung von traditionellem Pair Programming setzt denselben Standort voraus (gemeinsame Nutzung der Arbeitsumgebung).
- Idee von Distributed Pair Programming (DPP):
 - **Räumliche Trennung** der Pairs bei gleichzeitiger
 - Nutzung der **Vorteile von traditionellem Pair Programming**
 - durch **kontinuierliche Zusammenarbeit**.
- Voraussetzungen:
 - Effiziente **Kommunikationsmittel**.
 - Geeignete Werkzeuge zur **Unterstützung des simultanen Arbeitens** am selben Artefakt in **derselben Arbeitsumgebung**.
- Zentrale Herausforderung ist die Auswahl eines geeigneten Werkzeuges zur Unterstützung von Distributed Pair Programming.

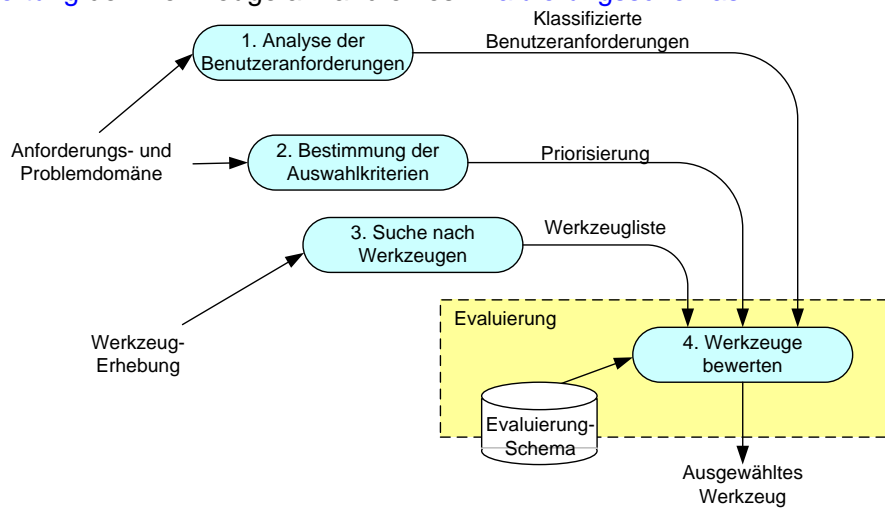


Institut für Softwaretechnik und Interaktive Systeme

Werkzeugauswahl

Vier wesentliche Schritte bei der Auswahl von Werkzeugen [Poston, 1992]:

1. Analyse und Klassifikation der **Benutzeranforderungen** und **erwarteten Werkzeugeigenschaften**.
2. Bestimmung und **Priorisierung** der **Auswahlkriterien**.
3. **Suche und Klassifikation** von Werkzeugen.
4. **Bewertung** der Werkzeuge anhand eines **Evaluierungsschemas**.



[Poston, 1992]

Institut für Softwaretechnik und Interaktive Systeme

Step 1: Anforderungen: Analyse der Benutzeranforderungen

- Identifikation **bestehender Analysen** der Anwendungs- und Problemdomäne (z.B. durch eine systematische Literaturrecherche oder aus Projekterfahrungen).
- **Brainstorming aller relevanten Stakeholder** zur Vervollständigung der individuellen Anforderungen.
- Beispiel: Distributed Pair Programming: Bestehende Analysen der Problemdomäne z.B. durch [Hanks, 2005], [Cox *et al.*, 2000].
- Wichtigste Anforderungen bzw. Anforderungskategorien an ein Werkzeug zur Unterstützung von DPP:
 - **Workspace Control and Awareness**. Definierte Kontrolle über Maus und Keyboard bzw. Sichtbarkeit der beteiligten Personen innerhalb der Arbeitsumgebung.
 - **Unterstützung von Screen Sharing**. Verteilte Verfügbarkeit der Bildschirminhalte.
 - **Floor control**: kontrollierte und nachvollziehbare Änderungen an einem Artefakt durch definierte Rollen.
 - **Gesturing**. Fähigkeit, durch Handbewegung oder Zeigen auf ein dargestelltes Bildelement auf Besonderheiten hinweisen können (z.B. Fehler).
 - Effizienter **Informationsaustausch** durch Kommunikation (Text, Sprache, Video).
 - **Plattformunabhängigkeit, Benutzerfreundlichkeit, Dokumentation**.

Institut für Softwaretechnik und Interaktive Systeme

Step 2: Anforderungen: Klassifizierung und Priorisierung

- Klassifikation und Gewichtung der ermittelten Anforderungen durch die Stakeholder z.B. in einem Requirements Elicitation Workshop (z.B. EWW) [Boehm *et al.*, 2001]

- Auszug aus der Anforderungsanalyse

Allgemeine Anforderungen (2)		Priorisierung
	Support of category supported features (e.g., Screen Sharing, Collaborative Work Support)	Critical (C)
	Support of Workspace Awareness	Critical (C)
Floor Control (7)		
	Support of Floor Control	Critical (C)
	Support of Role Changes	Critical (C)
	Role Assignment information	Medium (M)
Gesturing (4)		
	Second Pointer for the Navigator	Critical (C)
	Support of Highlighting	Low (L)
Kommunikation (5)		
	Voice Channels	High (H)
	Textual Chat	Medium (M)
	Video channel	Low (L)
Plattform (3)		
Usability (10)		
General tool Properties (8)		

- Priorisierung von Anforderungen

Priorisierung	Gewichtung
Critical	10
High importance	5
Medium importance	2,5
Low importance	1

- **Critical:** Basisfunktionen für DPP.
- **High importance:** Voraussetzung für effizienten PP Einsatz.
- **Medium importance:** Unterstützende Eigenschaften zur Steigerung des effizienten Werkzeugeinsatzes.
- **Low importance:** Nice-to-have Eigenschaften.

Step 3: Werkzeuge: Recherche und Klassifizierung

Identifikation von möglichen Werkzeugen:

- **Recherche über verwendbare** Werkzeuge für Computer Supported Collaborative Work:
 - Im Hinblick auf Anforderungskategorien.
 - Werkzeuge zur Unterstützung von verteiltem Arbeiten (unabhängig von DPP).
 - Werkzeuge zur expliziten Unterstützung von DPP.

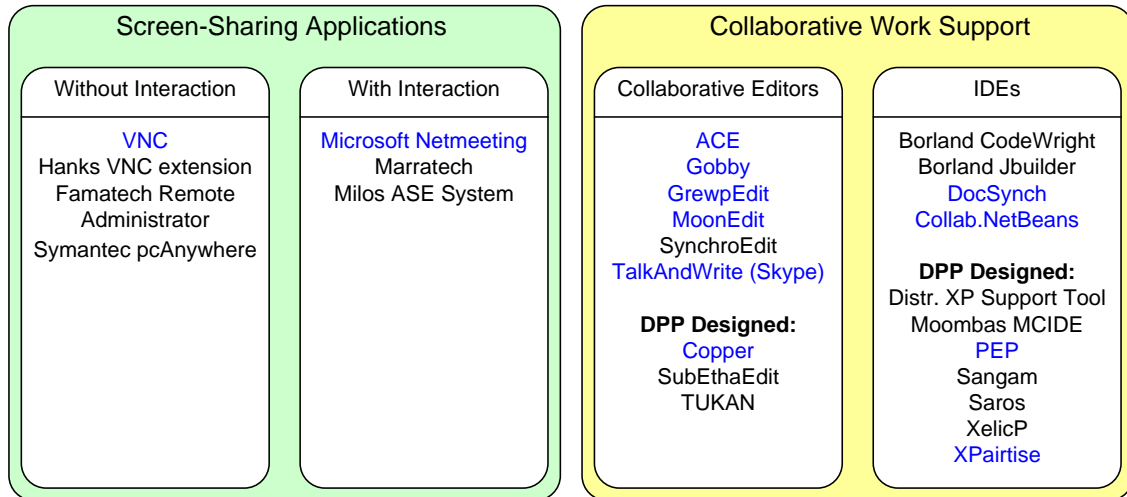
Grundlegende Klassifikation von Werkzeugen.

- **Screen-Sharing Applications.**
 - **Screen-Sharing ohne Interaktion:** Übertragung des Bildschirminhaltes (z.B. VNC). Typische Anwendungen bei Systemadministratoren oder Helpdesks.
 - **Screen-Sharing mit Interaktion:** zusätzliche Features wie Whiteboards, Chat (z.B. MS Netmeeting). Typische Anwendungen für Videokonferenzen.
 - **(Angepasste) Werkzeuge** für DPP Unterstützung.
- **Collaboration-aware applications.**
 - Verteilte Editoren.
 - Integrierte Entwicklungsumgebung (IDE).

Auszug aus der Werkzeugrecherche



- Exemplarische Auswahl aus der Werkzeugrecherche nach Verfügbarkeit.

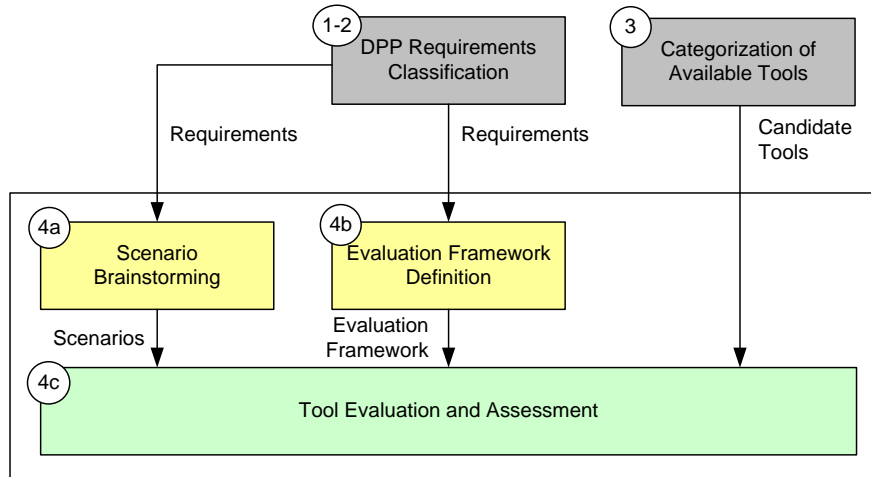


Institut für Softwaretechnik und Interaktive Systeme

Step 4: Bewertung der Werkzeuge: Evaluation Framework



- Evaluierungsschema für die systematische Bewertung von Werkzeug-Kandidaten im Hinblick auf die kategorisierten Anforderungen.
- 4a. Identifikation von **erfolgskritischen Szenarien** zur Evaluierung.
- 4b. Definition des **Evaluierungsschemas**.
- 4c. **Evaluierung und Bewertung** der Tools anhand der Szenarien.



Institut für Softwaretechnik und Interaktive Systeme

Step 4a: Bewertung der Werkzeuge: Identifikation erfolgskritischer Szenarien



OESTERREICHISCHE
COMPUTER GESELLSCHAFT
AUSTRIAN
COMPUTER SOCIETY



Szenarien

- Typische Arbeitsabläufe, abgeleitet aus den Benutzeranforderungen, die **durch das Werkzeug unterstützt** werden müssen.
- **Guidelines** für die Durchführung von konkreten typischen Aufgaben.
- **Szenario-Brainstorming Workshop** für Einsetzbarkeit von DPP.

6 typische Szenarien

- **Allgemeine Szenarien:**
 - Installation und Konfiguration des Werkzeugs.
 - Leistungsfähigkeit des Werkzeugs.
- **Spezielle DPP Szenarien**
 - Aufbau einer DPP Sitzung.
 - Unterstützung des Rollenwechsels (Floor control).
 - Session Management (Store/Restoring sessions).
 - Anwendung des Werkzeugs zur Lösung einer kleinen Implementierungsaufgabe.
- Evaluierung der Werkzeuge durch Anwendung der Szenarien und subjektive Einschätzung der Anwendung (DPP Teamschätzung).

Institut für Softwaretechnik und Interaktive Systeme

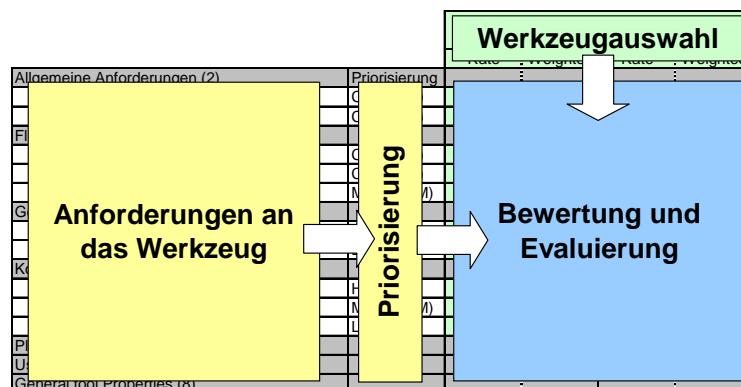
Step 4b: Bewertung der Werkzeuge: Erstellung eines Evaluierungsschemas



OESTERREICHISCHE
COMPUTER GESELLSCHAFT
AUSTRIAN
COMPUTER SOCIETY



- **Kategorisierte Anforderungen und Prioritäten** (y-Achse).
- **Mögliche ausgewählte Werkzeuge** für die Evaluierung (x-Achse).
- **Werkzeugevaluierung** (Evaluierungsmatrix).
 - **Einschätzung** des Grades der Anforderungsunterstützung durch das Werkzeug:
 - **Bewertungsskala** von 0 (nicht vorhanden/ nicht unterstützt) bis 5 einfache Anwendung und effiziente Unterstützung.
 - **Binäre Einschätzung** z.B. Plattformunterstützung (0/1 bzw. ja/nein).
 - **Gewichtung der subjektiven Bewertung** (gemäß Priorität der Anforderung).



Institut für Softwaretechnik und Interaktive Systeme

Step 4c: Bewertung der Werkzeuge: Auszug aus der Evaluierungsmatrix



- Verwendung der Werkzeuge und Anwendung von definierten Szenarien durch reale verteilte Pairs.
- Subjektive Team-Bewertung während/nach der Werkzeuganwendung entsprechend den kategorisierten Eigenschaften und dem Bewertungsschema.

		Screen Sharing				Collaborative Work				
		VNC		Netmeeting		Copper		PEP		
		Rate	Weighted	Rate	Weighted	Rate	Weighted	Rate	Weighted	
Allgemeine Anforderungen (2)		Priorisierung								
	Support of category supported features	Critical (C)	4	40	4	40	4	40	3	30
	Support of Workspace Awareness	Critical (C)	5	25	4	40	4	40	2	20
Floor Control (7)										
	Support of Floor Control	Critical (C)	0	0	3	30	5	50	3	30
	Support of Role Changes	Critical (C)	0	0	2	20	5	50	4	40
	Role Assignment information	Medium (M)	0	0	2	5	5	12,5	4	10
Gesturing (4)										
	Second Pointer for the Navigator	Critical (C)	1	1	1	10	0	0	0	0
	Support of Highlighting	Low (L)	5	25	2	2	2	2	1	1
Kommunikation (5)										
	Voice Channels	High (H)	0	0	5	25	1	5	0	0
	Textual Chat	Medium (M)	0	0	4	10	3	7,5	5	12,5
	Video channel	Low (L)	0	0	5	5	0	0	0	0
Plattform (3)										
Usability (10)										
General tool Properties (8)										

Institut für Softwaretechnik und Interaktive Systeme

Step 4c: Aggregation von Werkzeuganforderungen



- Betrachtung der einzelnen Anforderungskategorien (z.B. Summenbildung).
- Gesamtbetrachtung des Werkzeugs (Total).
- Threshold für eine dreistufige Bewertung ausgehend vom Maximalwert pro Kategorie
 - Eigenschaft / Anforderung wenig unterstützt: 0-33% (C, rot).
 - Durchschnittliche Unterstützung: 33-66% (B, orange).
 - Gute Unterstützung durch das Werkzeug: 66-100% (A, grün).

	Maximal		Screen Sharing				Collaborative Work			
	Rate	Weighted	VNC		Netmeeting		Copper		PEP	
			Rate	Weighted	Rate	Weighted	Rate	Weighted	Rate	Weighted
Allgemeine Anforderungen (2)	10	100	9	90	8	80	8	80	5	50
Floor Control (7)	35	155	7	17,5	11	65	25	137,5	20	102,5
Gesturing (4)	20	92,5	9	38	9	27,5	2	10	1	5
Kommunikation (5)	25	60	0	0	18	50	6	17,5	7	17,5
Plattform (3)	3	6	2	5	1	2,5	3	6	3	6
Usability (10)	50	222,5	35	170	29	154	36	185,5	35	161,5
General Tool Properties (8)	16	16	7	7	11	11	5	5	8	8
Total (39)	159	652	69	327,5	87	390	85	441,5	79	350,5

Institut für Softwaretechnik und Interaktive Systeme

Ergebnisse: Evaluerte Werkzeuge im Überblick



- **Prozentuelle Betrachtung** der evaluierten Werkzeuge im Hinblick auf die gewichtete Bewertung gemäß Anforderungskategorie.

Werkzeug	Kategorie		Anforderungen							Total		
	Screen-Sharing	Collaboration	Basics	Gesturing	Floor Control	Communication	Plattform	Usability	Tool	Rate	Weighted	Weighted [%]
Xpairtise		X	100%	22%	82%	38%	100%	83%	56%	100	471	72%
Copper		X	80%	11%	89%	29%	100%	83%	31%	85	442	68%
TalkAndWrite		X	90%	85%	0%	83%	42%	73%	75%	91	396	61%
Net-Meeting	X		80%	30%	42%	83%	42%	69%	69%	87	390	60%
PEP		X	50%	5%	66%	29%	100%	73%	50%	79	351	54%
DocSync		X	80%	0%	63%	24%	100%	60%	56%	72	342	52%
VNC	X		90%	41%	11%	0%	83%	76%	44%	69	328	50%
Gobby		X	90%	5%	0%	21%	100%	82%	50%	61	304	47%
GrewpEdit		X	70%	16%	0%	25%	100%	78%	25%	56	284	44%
MoonEdit		X	60%	16%	0%	0%	100%	86%	56%	58	282	43%
NetBeans Coll.		X	60%	0%	8%	29%	100%	63%	81%	68	249	38%
ACE		X	50%	22%	8%	8%	83%	71%	75%	57	250	38%
Maximal erreichbare Bewertung:									159	652		

Institut für Softwaretechnik und Interaktive Systeme

Ergebnisse: Gewonnene Erkenntnisse



Gute Unterstützung für DPP:

- **Screen-Sharing** bzw. **Collaboration** (Basic-Anforderungen an ein DPP-Werkzeug)
→ Grundvoraussetzung und primäres Auswahlkriterium für DPP Werkzeuge.
- **Usability** (Benutzerfreundlichkeit) und **Plattformunabhängigkeit** (Werkzeuflösungen für unterschiedliche Plattformen).

Partielle Unterstützung:

- **Gesturing**, **Floor Control** und **Kommunikation** werden teilweise unterstützt; meist Schwerpunkt auf nur **eine** Anforderungskategorie.
- **Speziell für DPP gestaltete Werkzeuge**, wie XPairtise, Copper und PEP unterstützen DPP tendenziell zwar besser, bieten aber noch zahlreiche Verbesserungsmöglichkeiten.
- **Keines** der untersuchten Werkzeuge erfüllt **alle notwendigen und wünschenswerten Anforderungen** zur effizienten Unterstützung von DPP.

Institut für Softwaretechnik und Interaktive Systeme

Conclusion & Further Work



- Der Einsatz von **Distributed Pair Programming** kann im Rahmen eines **flexiblen Prozesses** helfen, **qualitativ hochwertige Produkte** in kurzen Iterationen zu erstellen.
- Grundvoraussetzung dabei ist die **Verfügbarkeit von geeigneten Werkzeugen**.
- Die **Auswahl eines geeigneten Werkzeugs** erfordert ein **systematisches Vorgehen** im jeweiligen Projektkontext.
- Klassifizierte **Anforderungen**, **kategorisierte Werkzeuge** und **Szenarien** können als Ausgangsbasis für die Evaluierung dienen.
- Der vorgestellte **Evaluierungsprozess** unterstützt
 - **Projektleiter und Entwickler** bei der Auswahl passender Werkzeuge.
 - **Werkzeughersteller** bei der Identifikation von Verbesserungspotenzial zur individuellen Weiterentwicklung der Werkzeuge.
- Die durchgeführte Werkzeugevaluierung zeigte sowohl **Stärken** als auch **Schwächen** und kann als **Ausgangsbasis** für die weitere Entwicklung von Werkzeugen zum effizienten Einsatz von Distributed Pair Programming dienen.
- **Nächste Schritte:**
 - Verfeinerung des Evaluierungsprozesses und Verifikation der Ergebnisse.
 - Berücksichtigung weiterer Werkzeuge →

Institut für Softwaretechnik und Interaktive Systeme

**Vielen Dank für Ihre
Aufmerksamkeit!**



Evaluierung von Werkzeugen für Distributed Pair Programming: Eine Fallstudie

**Dietmar Winkler
Stefan Biffi**

Technische Universität Wien
Institut für Softwaretechnik und Interaktive Systeme

<http://qse.ifs.tuwien.ac.at>
{Dietmar.Winkler, Stefan.Biffi}@tuwien.ac.at

Institut für Softwaretechnik und Interaktive Systeme

Referenzen



- S. Atsuta anG. Canfora, A. Cimitile and C. A. Visaggio: "Lessons learned about distributed pair programming: what are the knowledge needs to address?", In Proc 12th IEEE Int. Wsh. on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE03), pages 314–319, 2003.
- B. Boehm, P. Grünbacher, R. Briggs: „EasyWinWin: A Groupware-Supported Methodology for Requirements Negotiation“, Int. Conf. on Software Engineering, 2001.
- A. Cockburn and L. A. Williams: "The costs and benefits of pair programming", in Extreme Programming Examined, G. Succi, M. Marchesi (editors), pages 223-248, Addison Wesley, 2001.
- D. Cox and S. Greenberg: "Supporting collaborative interpretation in distributed groupware", in CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work, pages 289–298, 2000.
- B. F. Hanks: "Empirical Studies of Distributed Pair Programming", Dissertation, University of California Santa Cruz, 2005.
- B. Kitchenham, S. Linkman and D. Law: "DESMET: A methodology for evaluating software engineering methods and tools", in Computing and Control Engineering J, pages 120-126, 1997.
- R.M.Poston, M.P. Sexton: "Evaluating and selecting testing tools", IEEE Software (9/3), 1992.
- T. and J. Schummer: "Support for distributed teams in extreme programming", in G. Succi and M. Marchesi, editors, Extreme Programming Examined, pages 355-378, Addison-Wesley, 2001.
- L. A. Williams and R. Kessler: "All I Really Need to Know about Pair Programming I Learned in Kindergarten", in Communication of the ACM May 2000/Vol 43 No. 5, pages 108-114, May 2000.
- L. A. Williams, R. Kessler, W. Cunningham and R. Jeffries: "Strengthening the case for pair-programming", in IEEE Software, 17:4, pages 19-25, July/August 2000.

3.3. Flexible Prozess-Werkzeug Integration

Georg Kalus, Marco Kuhrmann, Manuel Then
Technische Universität München, Institut für Informatik
Boltzmannstr. 3
85748 Garching bei München
{kalus | kuhrmann | then}@in.tum.de

Abstract

Als erstes V-Modell unterstützt das V-Modell XT seine Anwender bereits auf der Ebene des Standards durch Werkzeuge, die das Aufsetzen eines Projekts unterstützen. Die Kette der Referenzwerkzeuge endet jedoch nach dem Tailoring mit dem Projektassistenten. In dieser Präsentation stellen wir ein Konzept vor, das die flexible Integration verschiedener Prozesse in unterschiedliche Werkzeuge erlaubt. Ziel ist es dabei, umfangreiche Prozesse zu filtern und die für die adressierten Werkzeuge relevanten Informationen so aufzubereiten, dass ein Maximum an Anwenderunterstützung erreicht werden kann. Die Ergebnisse werden anhand des aktuellen V-Modell XT gezeigt.

Motivation

In Software Entwicklungsprojekten sind Anwender üblicherweise darauf angewiesen, entweder auf integrierte, kommerzielle Produkte zurückzugreifen oder zu improvisieren, wenn es darum geht, Prozesse zu automatisieren. Oftmals werden die gewünschten Prozesse aber auch von kommerziellen Werkzeugen nicht ausreichend unterstützt, sodass die pragmatische Kopplung verschiedener, nicht zwangsweise aufeinander abgestimmter Werkzeuge erforderlich ist. Die Durchgängigkeit der Umsetzung und Unterstützung des gewählten Entwicklungsprozesses ist hierbei nicht zwangsweise gegeben. Dies tritt insbesondere dann auf, wenn Werkzeuge nur Teilaspekte der Projektaktivitäten abdecken. Zu beantworten sind hier die Fragen: „Wie können Prozesse durchgängig in gegebene Werkzeugumgebungen integriert werden?“ und „Wie können Entwickler durch werkzeugunterstützte Prozesse bei ihrer täglichen Arbeit unterstützt werden?“

Ansatz

Zur Beantwortung dieser Fragen wurde – aufbauend auf den im Projekt CollabXT gesammelten Erfahrungen (vorgestellt auf der SEE2008) – ein flexibles Generatorframework (Abbildung 3.1) entwickelt, das als Eingabe eine Metamodell-basierte Beschreibung eines Entwicklungsprozesses verarbeitet und auf das gewünschte Werkzeug abbildet. Grundlage ist ein abstraktes Zwischenmodell zur Beschreibung von Prozessen. Der unterstützende Prozess wird zunächst auf dieses Modell abgebildet und dann in das entsprechende Format des Zielwerkzeugs transformiert. Durch diese Entkopplung von Eingabeprozess und Ausgabewerkzeug ist es möglich, flexibel verschiedene Prozesse zu unterstützen. Genau so ist es möglich, weitere Werkzeuge anzubinden. Auch können für einen Eingabeprozess eine Reihe von Zielwerkzeugen angegeben werden. Damit kann jede Projektrolle mit dem für sie am besten geeigneten Werkzeug arbeiten, ohne das verbindende Gerüst des gemeinsamen Entwicklungsprozesses aufzugeben.

3.3. Flexible Prozess-Werkzeug Integration

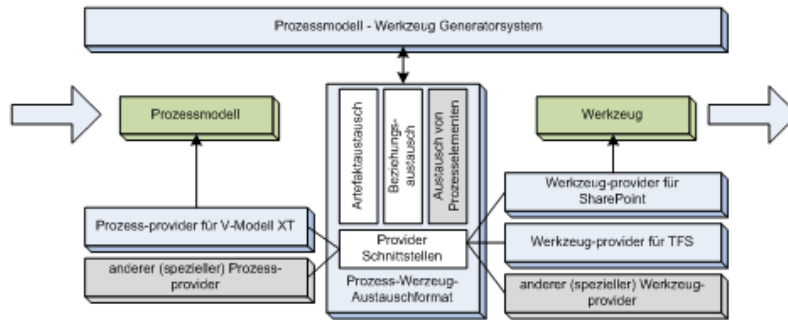


Abbildung 3.1.: Basisarchitektur des Frameworks

Um einen Entwicklungsprozess – z.B. RUP oder allg. SPEM-basierte Prozesse – in die Generatorinfrastruktur einzubinden, ist nur die Umwandlung der Prozessbeschreibung in das Zwischenformat (im Sinne der Abbildung ein Prozess-Provider) bereitzustellen. Analog kann ein neues Werkzeug durch Abbildung vom Zwischenformat auf selbiges unterstützt werden (mithilfe eines Werkzeug-Providers). Bisher gibt es folgende, beispielhafte Implementierungen: Auf Prozessseite wird das V-Modell XT 1.2.x und 1.3 unterstützt. Auf Werkzeugseite wird Microsoft SharePoint, Microsoft Team Foundation Server und Microsoft Office unterstützt. Abhängig vom adressierten Werkzeug wird ein Werkzeug-provider meist nur einen Teil des gesamten Prozessmodells berücksichtigen. Sollen z.B. nur Produktvorlagen für Microsoft Office erzeugt werden, können große Teile des Abhängigkeitsgeflechts im Prozessmodell ignoriert werden - usw.

Das Zwischenmodell

Das Zwischenformat (Abbildung 3.2) ist durch das V-Modell XT Metamodell inspiriert. Es enthält eine minimale Menge von Prozesselementen, die für eine Werkzeugunterstützung notwendig sind. In der aktuellen Fassung sind dies:

- Produkte (Artefakte, Arbeitsergebnisse)
- Aktivitäten
- Rollen (Verantwortlichkeiten)
- Meilensteine
- Disziplinen

Dieses Zwischenmodell bildet die wesentlichen Prozessartefakte in einer generischen Form ab. Eine Erweiterung bzw. Spezialisierung dieses Zwischenformats ist dabei immer noch möglich, sollte der Abdeckungsgrad nicht genügen. Das Zwischenformat selbst macht keine Annahmen hinsichtlich der Semantik der Prozesse auf der einen und der Werkzeuge auf der anderen Seite. Das ist die Grundlage der freien Kombinierbarkeit.

Neben diesen Grundbausteinen eines Entwicklungsprozesses definiert das Zwischenmodell eine Reihe von Abhängigkeitsklassen (Abbildung 3.3) zur Darstellung von Beziehungen zwischen den Prozesselementen. Diese sind in ihrer Struktur so allgemein gehalten, dass prinzipiell beliebige Beziehungen zwischen Modellelementen abgebildet werden können. Die Implementierung im Zwischenmodell beschränkt sich in seinem Umfang auf solche Beziehungen, die von den vorhandenen Werkzeug Providern benötigt wird. Würde ein Werkzeugprovider z.B. auf die Beziehungen zwischen Produkten zurückgrei-

3. Session 3: Werkzeuge zur Kollaboration in Projekten

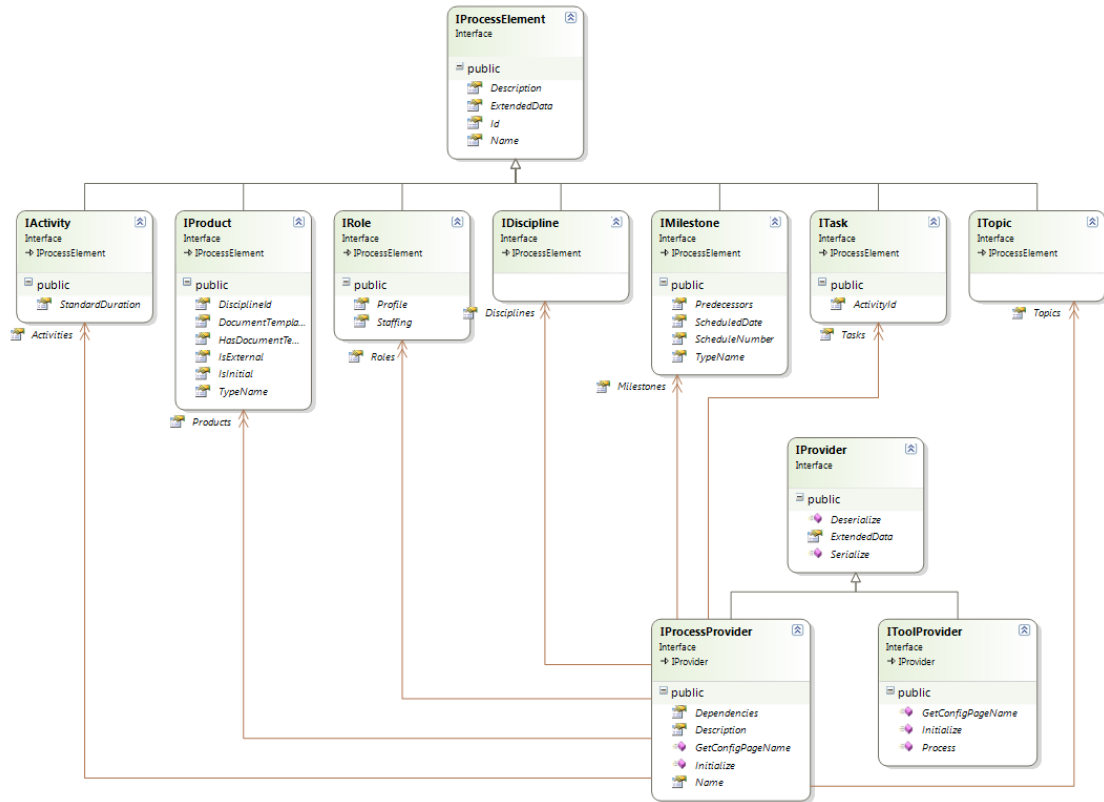


Abbildung 3.2.: Auszug der abbildbaren Prozesselemente

fen, so wäre das Zwischenmodell durch Einführung eines neuen Beziehungstyps zu erweitern.

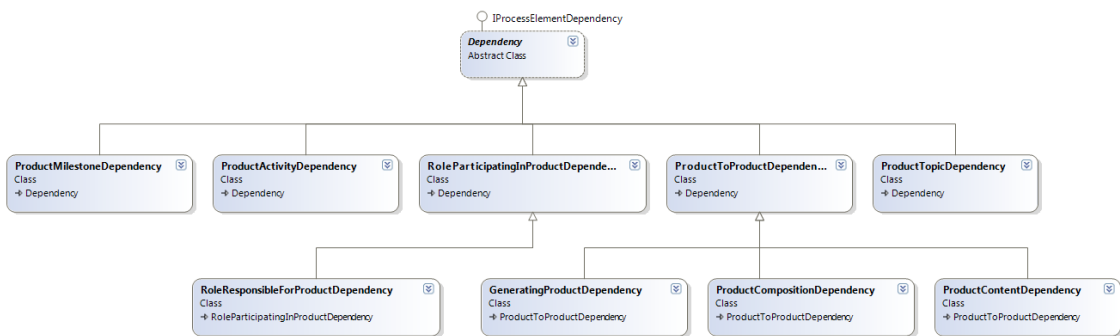


Abbildung 3.3.: Abhängigkeiten im Zwischenmodell

Die beiden zurzeit implementierten Werkzeugprovider stützen sich ausschließlich auf das Zwischenformat ab. Allerdings wird es auch Werkzeugprovider geben, die speziell auf einen bestimmten Prozess zugeschnitten sind. Beispielsweise ist ein Werkzeugprovider zur Generierung von V-Modell XT Produktvorlagen im Microsoft Word 2007 docx-Format in Arbeit. Dieser benötigt spezielle und komposite Inhalte aus dem V-Modell XT, die über die im Zwischenmodell abgebildeten Inhalte hinausgehen. Daher wird im Prozessprovider optional auch das Originalmodell vorgehalten. Es ist naheliegend, dass derart spezialisierte Werkzeugprovider nicht von der Prozessmodell-agnostischen Flexi-

3.3. Flexible Prozess-Werkzeug Integration

bilität des Zwischenmodells profitieren.

Ein Beispiel: Microsoft Team Foundation Server

Der Werkzeugprovider für den Microsoft Team Foundation Server (TFS) ist ein interessantes Fallbeispiel, da der TFS nicht die Ausdrucksmächtigkeit hat, das Prozessmodell vollständig und in allen Aspekten darzustellen: Der TFS ist aktivitätsorientiert konzipiert und kennt im Wesentlichen nur die sogenannten „Work Items“. Inhalte des Prozesses müssen vom Werkzeugprovider also sinnvoll auf dieses Strukturierungselement des TFS abgebildet werden. Teilprozesse und Workflows lassen sich über das Zustandsmodell von Work Items gut abbilden. Zur Darstellung von V-Modell XT Produkten im Werkzeug bleibt aufgrund der aktivitätsorientierten Philosophie des TFS nur die Möglichkeit, diese ebenfalls durch Work Items im Werkzeug zu „spiegeln“. Struktur- und Abhängigkeitsinformationen werden wiederum durch „Links“ zu jedem Work Item dargestellt. Zur Abbildung der V-Modell XT Meilensteine existiert im TFS ebenfalls kein unmittelbar verwendbares Konzept. Hier kommen ebenfalls wieder Stellvertreter-Work-Items zum Einsatz: Die Spiegel-Work-Items der zu einem Meilenstein fertigzustellender Produkte, Aktivitäten und Arbeitsaufträge werden mit den Meilenstein-Work-Items verlinkt, sodass die vom Prozess geforderten Verknüpfungen abgebildet werden kann. Das Beispiel TFS zeigt die Herausforderungen, die bei der automatischen Prozessunterstützung und Transformation auftreten: Konzepte müssen abgeglichen werden und sofern keine unmittelbare Unterstützung verfügbar ist, mithilfe einer Abbildungsvorschrift angemessen umgesetzt werden. Die Umrechnung und Interpretation der Inhalte und Strukturen übernimmt hier vollständig der Werkzeugprovider. Damit wird jedoch keinesfalls die Position eines fähigen Projektleiters überflüssig. Vielmehr kann ein Teil der komplexen Prozessinterpretation in den Werkzeugprovider ausgelagert werden. Der beim Projektleiter gesparte Aufwand wird von Prozessingenieuren und Entwicklern bei der Konzeption und Umsetzung des Werkzeugproviders getragen.

Bewertung

Mit der vorliegenden Lösung wurden die Ergebnisse aus CollabXT auf neue Architektur und damit auf eine breitere und flexiblere Basis gestellt. Die Architektur u.a. mit dem Zwischenformat und dem Plug-In-Konzept für die verschiedenen Provider erlaubt eine vereinfachte Anbindung sowohl neuer Prozesse als auch neuer Werkzeuge. Das Generatorframework soll in Zukunft derart weiterentwickelt werden, dass es auch die Einsteuerung von benutzerdefinierten Methodenbausteinen erlaubt. Dazu sollen geeignete Andockpunkte bei Prozessen identifiziert und bei den Werkzeugen geeignete Repräsentationen der Methoden gefunden werden.

Flexible Prozess-Werkzeug Integration

Georg Kalus, Dr. Marco Kuhrmann, Manuel Then

Institut für Informatik
Software & Systems Engineering

Agenda

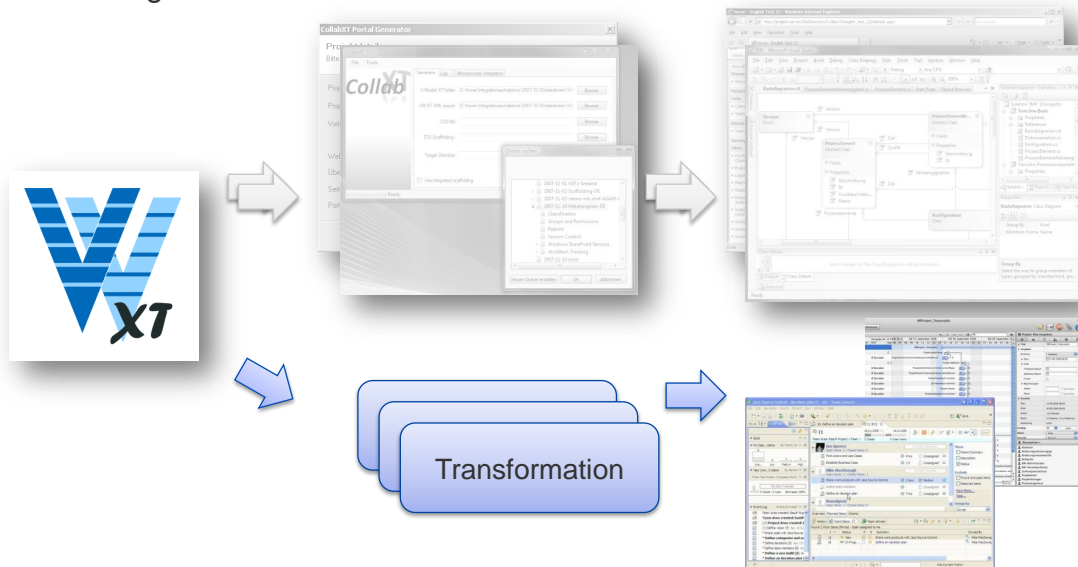
- Intro
 - Rückblick: SEE 2008
 - Ist-Stand und Bedarf

- Das Framework
 - Architektur und Zwischenmodell
 - Beispiel
 - Diskussion

- Ausblick

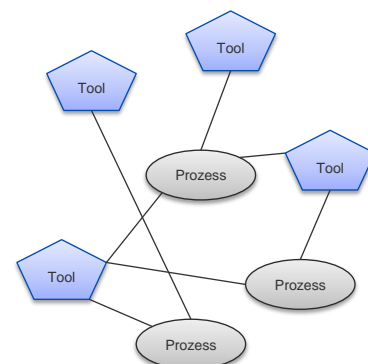
Und in Zukunft?

- Anbindung weiterer Tools?



In Zukunft:

- Prozesse entwickeln sich weiter
 - Wartung und Pflege
 - Neue Modellversionen
- Werkzeuge entwickeln sich weiter
 - Neue Werkzeugversionen
 - Komplette neue Werkzeuge
 - Neue Fähigkeiten der Werkzeuge
- Problematik
 - Neue und Alte Prozesse in
 - Neuen und Alten Werkzeugen



n:m

3.3. Flexible Prozess-Werkzeug Integration

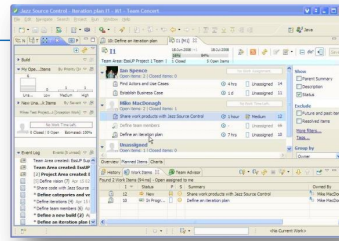
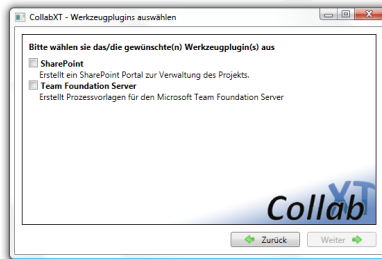
Ziel

- Eine integrierte Plattform

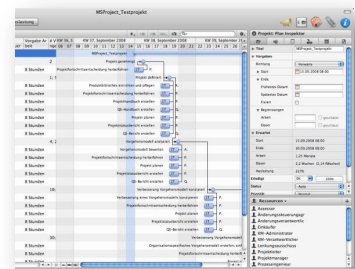
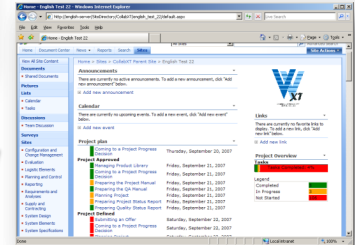


SPEM

XP

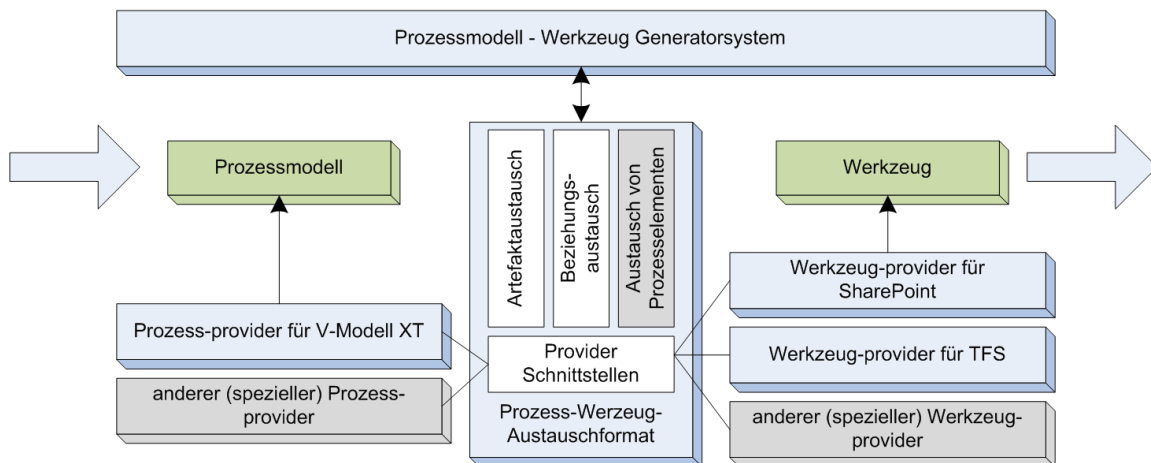


?



Architektur - Überblick

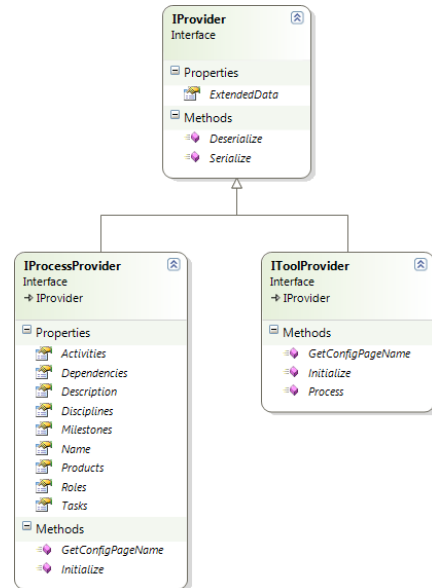
- Framework für flexible Prozess-Werkzeug Kopplung



Architektur - Pluginkonzept

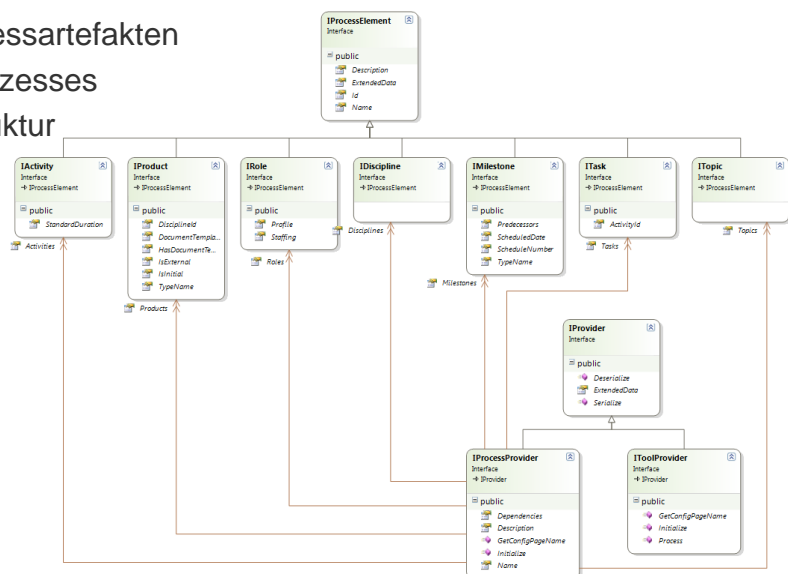
- Generisches Framework mit Plugins
 - Prozess-Provider
(analysieren und interpretieren eine Prozessbeschreibung und überführen diese ins Zwischenmodell)
 - Tool-Provider
(interpretieren das Zwischenmodell und bilden dieses auf das Zielwerkzeug ab)

- Aktuell verfügbare Plugins
 - Prozess: V-Modell XT
 - Tools: Sharepoint, TFS, Word



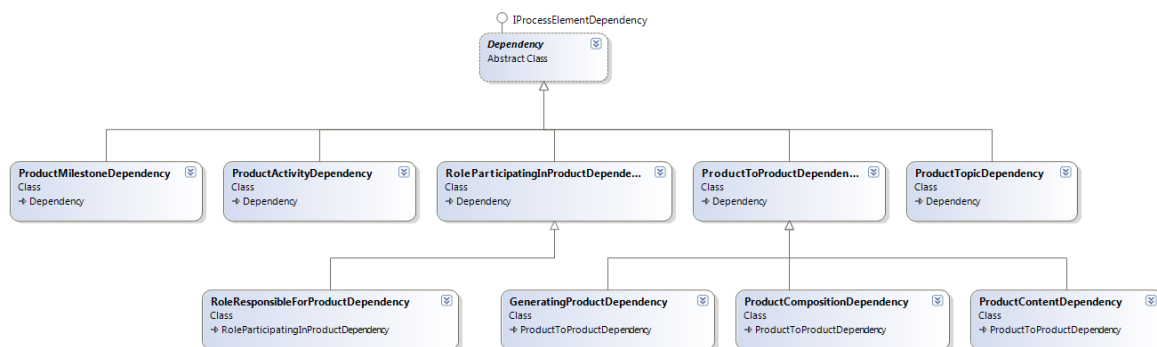
Das Zwischenmodell

- Inspiriert vom V-Modell XT Metamodell
 - Definition von Prozessartefakten
 - Abstraktion des Prozesses
 - Abstraktion der Struktur



Das Zwischenmodell

- Inspiriert vom V-Modell XT Metamodell
 - Beziehungstypen generisch und ausdrucksmächtig
 - Separat von Artefakte definiert

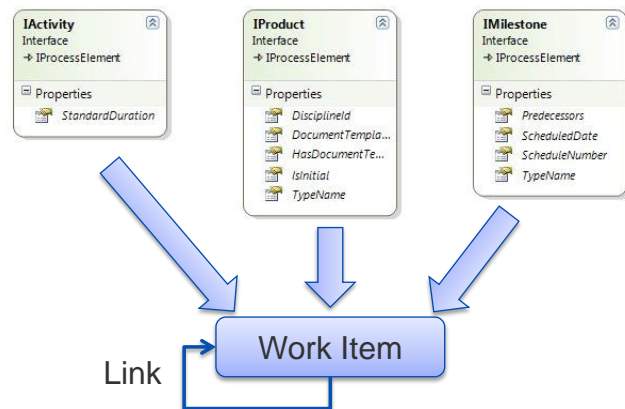


Tool-Provider

- Abstrahieren Plattform-Spezifika und:
 - Zwischenmodell als Eingabe
 - Greifen nur auf „benötigte“ Teilmenge des Zwischenmodells zu (z.B. Projektplan in Project vs. Projektplan im TFS, Produktvorlagen)
 - Enthalten alle Intelligenz zur optimalen Ansteuerung der Werkzeuge
- Tool-Provider können auf Original-Prozessmodell zugreifen
 - Zwischenmodell wird umgangen
 - Verlust der Flexibilität
 - Nötig für speziell auf das Modell zugeschnittene Tool-Provider
 - Beispiel: V-Modell XT Office Produktvorlagen Provider

Tool-Provider – Beispiel TFS

- Spezifika des adressierten Werkzeugs erfordern spezielle Abbildung/Interpretation
 - Team Foundation Server kennt nur „Work Items“
 - Abhängigkeiten lassen sich nur über „Links“ zwischen Work Items darstellen
 - Prozessdokumentation wird separat gehandhabt
 - Vorlagen werden separat gehandhabt
 - Projektplan wird aus TFS heraus generiert
 - ...



Diskussion

- Sichten-basierte Arbeit möglich (ein Modell, viele Werkzeuge)
- Gegebene Tool-Provider funktionieren auch mit neuen Prozessmodellen
 - Solange das Zwischenmodell korrekt befüllt wird
 - Neue Tool-Provider möglich
- Herausforderung: Abbildung weiterer Prozessmodelle auf das Zwischenmodell
 - Zwischenmodell ist erweiterbar und flexibel
 - Neue Prozess-Provider möglich
- Architektur ist flexibel
 - Mächtigkeit nur abhängig von der Intelligenz der Provider

Ausblick

- „Pipelining“ von Tool-Providern erstrebenswert
 - Tool-Provider für V-Modell XT Produktvorlagen im OpenXML Format in Arbeit

- Pilotierung
 - z.B. bei der Bundesnetzagentur

- Open-Source Re-Launch auf CodePlex
 - Existierende Projekte bleiben bestehen
 - Neues Projekt in Kürze online



4. Session 4: Nutzen und Schwächen von Vorgehensmodellen

Inhalt

4.1. Einsatz des V-Modell XT in der Behördenpraxis	136
4.2. ITIL Prozesse in der BWI Systeme GmbH	150
4.3. Kostenseitige Begleitung von Vorgehensmodellen	164

4.1. Einsatz des V-Modell XT in der Behördenpraxis – Analyse, Auswertung, Verbesserung

Christian Lange¹, Marco Kuhrmann², André Schnackenburg¹

¹Bundesstelle für Informationstechnik, BIT 7
Bundesverwaltungsamt
50728 Köln

{christian.lange | andre.schnackenburg}@bva.bund.de

²Technische Universität München, Institut für Informatik
Boltzmannstr. 3
85748 Garching bei München
kuhrmann@in.tum.de

Abstract

Das V-Modell XT ist verbindlich für IT-Projekte in Deutschen Behörden. Durch seine Anpassbarkeit bietet es dem Anwender viele Möglichkeiten, den Einsatz zu gestalten. Fünf Jahre nach der Einführung des V-Modell XT ist allerdings nicht bekannt, in welcher Form, Tiefe und Konsequenz seine Anwendung erfolgt. Die präsentierte Studie analysiert bisher durchgeführte IT-Projekte der Bundesverwaltung und beantwortet die offenen Fragen zu Verbreitung, Einsatz, Problemen und Erfolg der V-Modell XT Projekte. Die Ergebnisse werden der Verbesserung der Anwendbarkeit dienen, indem Empfehlungen, Leitfäden und weitere Maßnahmen abgeleitet werden.

Motivation

Das V-Modell XT ist in Deutschland seit 2004 durch den IMKA¹ zur Anwendung in allen neuen IT-Projekten des Bundes empfohlen. Seit 2004 starten Projekte, deren Durchführung und Organisation den Vorgaben des V-Modell XT folgt. Es bietet umfangreiche Möglichkeiten zur projekt- und organisationsspezifischen Anpassung. Außerdem gibt es unterstützende Werkzeuge, Schulungen und Literatur. Das V-Modell XT dient dazu, die Produkte, Rollen und Aktivitäten in einem IT-Projekt zu strukturieren; aber es lässt dem Anwender auch Freiheiten zur konkreten Ausgestaltung seines Projektes. Bisher ist nicht ausreichend bekannt, in welcher Form, Tiefe und Konsequenz die Anwendung erfolgt. Auch ist der Rücklauf an Erfahrungen aus der V-Modell XT Anwendung bislang nicht etabliert, so dass die Aufnahme, Aufbereitung und Weiterleitung der Erfahrungen in die Weiterentwicklung nur eingeschränkt gewährleistet ist.

Fünf Jahre nach der Einführung des V-Modell XT soll die hier präsentierte Studie den Status Quo der V-Modell XT Anwendung auswerten. Ziel der Studie ist es, die Verbreitung, Art der Anwendung des V-Modell XT und den Erfolg der nach ihm durchgeführten IT-Projekte zu erfassen und darauf basierend Empfehlungen zur weiteren Verbesserung zu geben.

¹ Interministerieller Koordinierungsausschuss für die Informationstechnik in der Bundesverwaltung

Methode

Die hier präsentierte Studie soll die offenen Fragen zum Einsatz des V-Modell XT in der Behördenpraxis beantworten. Die angewandte Forschungsmethode für diese Studie ist eine Fallstudie, bei der mehrere Projekte mit V-Modell XT Einsatz analysiert werden („multi case study“). Die Studie ist in drei Phasen gegliedert: i) Telefonumfrage in der Bundesverwaltung um relevante Projekte zu identifizieren; ii) Fragebogen mit 58 Fragen zur standardisierten Datenerfassung bei den identifizierten Projekten; iii) semistrukturierte Interviews, um detaillierte Sachverhalte, die sich aus dem Datenmaterial des Fragebogens ergeben, zu analysieren. Fragebogen und Interviews bilden den Kern der Studien. Die folgenden beiden zentralen Forschungsfragen werden in der Studie unter Anwendung des Goal-Question-Metric (GQM) Paradigma beantwortet:

1. Wie wird das V-Modell XT in der Behördenpraxis angewendet?
2. Welche Auswirkungen haben die Kontext-Faktoren der Anwendung auf Projekterfolg?

Ergebnisse

Derzeit befindet sich die Studie in der dritten, vertiefenden Phase. Die bisherigen Ergebnisse zeigen, dass es Potential zur Förderung der Akzeptanz, der Anwendbarkeit und zum Know-how zur korrekten Anwendung in der Behördenpraxis gibt. Die Ergebnisse sollen zu Maßnahmen führen, um dieses Potential auszuschöpfen. Insbesondere werden die Ergebnisse der Studie zu folgenden Maßnahmen beitragen:

- Verfahrensanweisungen und Empfehlungen zur Anwendung, insbesondere im Bereich der „Best Practices“, etablierte/standardisierte Methoden.
- Bereitstellung kompakter Leitfäden, insbesondere im Bereich der effizienten, ggf. auch pragmatischen Nutzung und Umsetzung des V-Modell XT.
- Steigerung des Bekanntheitsgrads und der Akzeptanz des V-Modell XT durch Empfehlungen für die Schulung, Hilfe zur Eignungseinstufung, Aufklärung etc.

Die Folge ist die Ableitung weiterer Schritte unter Berücksichtigung der oben genannten Ergebnistypen. Weiterhin soll die Studie dazu beitragen, den durch das Standard V-Modell XT nicht abgedeckten Bedarf der Behörden zu ermitteln.

Darauf aufbauend ist jedoch auch eine weitergehende Tiefenanalyse der Anwendung des V-Modell XT eine Zukunftsoption. Diese kann z.B. durch die rückblickende Analyse der Behördenprojekte („Projektarchäologie“) durchgeführt werden, indem Produkte und Projektverläufe bereits durchgeführter Projekte detailliert analysiert und bewertet werden.

Einsatz des V-Modell XT in der Behördenpraxis

- Analyse, Auswertung, Verbesserung -

Christian Lange, BIT
Marco Kuhrmann, TU München
André Schnackenburg, BIT

SEE 2009 – Berlin

1

Bundesstelle für IT

- Zentraler IT-Dienstleister der Bundesverwaltung
 - BIT2 – IT-Beratung
 - BIT7 – Standards und Methoden, Kompetenzzentrum OSS
- Förderung des Einsatzes des VMXT im Bund
 - IMKA-Beschluss
 - Beratung, Coaching
 - Pflege und Weiterentwicklung

SEE 2009 – Berlin

2

Agenda

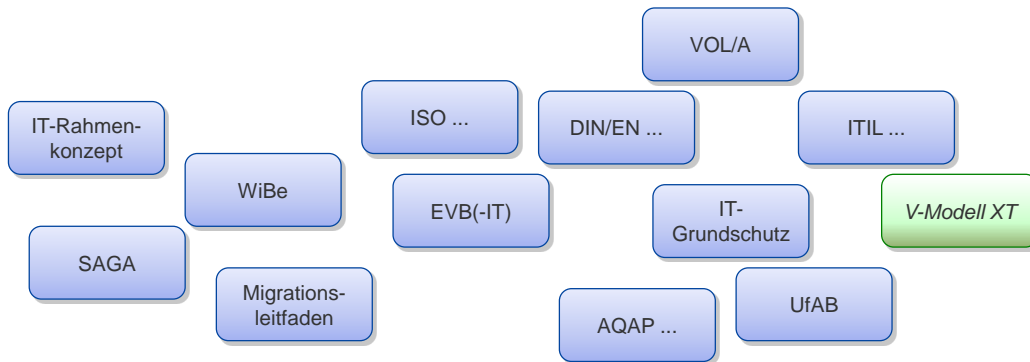
1. Das V-Modell XT in der Öffentlichen Verwaltung
2. Ziel & Aufbau der Studie
3. Erste Ergebnisse und Feststellungen
4. Ausblick

Besondere Anforderungen in der ÖV

- Das V-Modell XT trifft im öffentlichen Bereich auf besondere Situationen
 - Projekte meistens **Auftraggeberprojekte** (Einkauf, Beschaffung)
 - Eigenentwicklungen (**Auftraggeber-/Auftragnehmerprojekte**) nur in bestimmten Bereichen
 - Regelungsdichte durch **Vorgaben, Normen** etc.
 - Ausschreibungs- und Vergaberecht
 - Sicherheitsrichtlinien
 - Barrierefreiheit etc.
- Konsequenzen für Projekte/Prozesse
 - Langfristig (Vergabe, Begleitung, Kontrolle, Abnahme)
vs. schnelllebig (Entwicklung)
 - Zahlreiche Projektbeteiligte (Beteiligungsverfahren)
 - Auftraggeberaufgaben werden zu den Auftragnehmern verlagert etc.

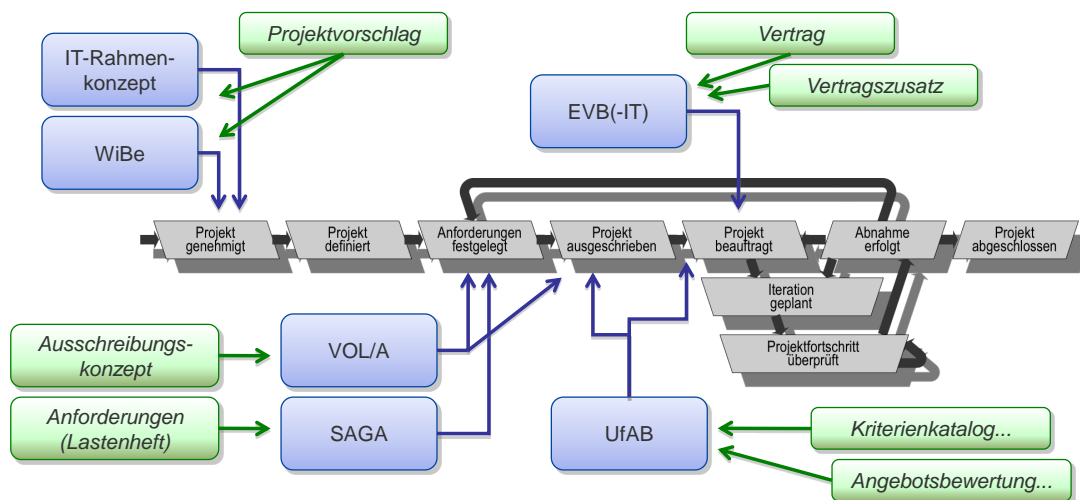
Standards in der IT

- Beispiele von IT-Standards in der öffentlichen Verwaltung...
 - Sind relevant für verschiedene Phasen eines Projekts
 - Sind verpflichtend/optional für bestimmte Rahmenbedingungen



Standards und das VMXT

- V-Modell XT ist kein Widerspruch → Ergänzung/Rahmen



Ziel und Vorgehen der Studie

SEE 2009 – Berlin

7

Ziel der Studie

5 Jahre nach seiner Einführung in der
Bundesverwaltung

Status Quo der V-Modell XT Anwendung ?

Verbreitung?

Erfolge?

Art der Anwendung?

Probleme?

Verbesserung der Anwendung

SEE 2009 – Berlin

8

Übersicht über die Studie

- Fallstudie
 - Auswertung von VMXT-Projekten im Bund
 - Praxisrelevanz der Ergebnisse
 - drei Schritte



3 Schritte



- Erfassung von relevanten Projekten, Teilnehmern
- Verbreitung des VMXT

- umfangreiche, standardisierte Datenerfassung

- detaillierte, teilweise individuelle Datenerfassung
- Einzelfallanalyse

Heransgehensweise

- Pilotierung des Fragebogens
 - Verständlichkeit der Fragen
 - Ausräumung von Fehlern
 - 6 Teilnehmer (nicht Teilnehmer der eigentlichen Studie)
 - 3 mit VMXT-Wissen, 3 ohne
 - Durchschnittliche Dauer: 20 Minuten
 - Ziel: Fragebogen unter 30 Minuten
- Anonymisierung der Daten
- Keine „Überwachungsstudie“

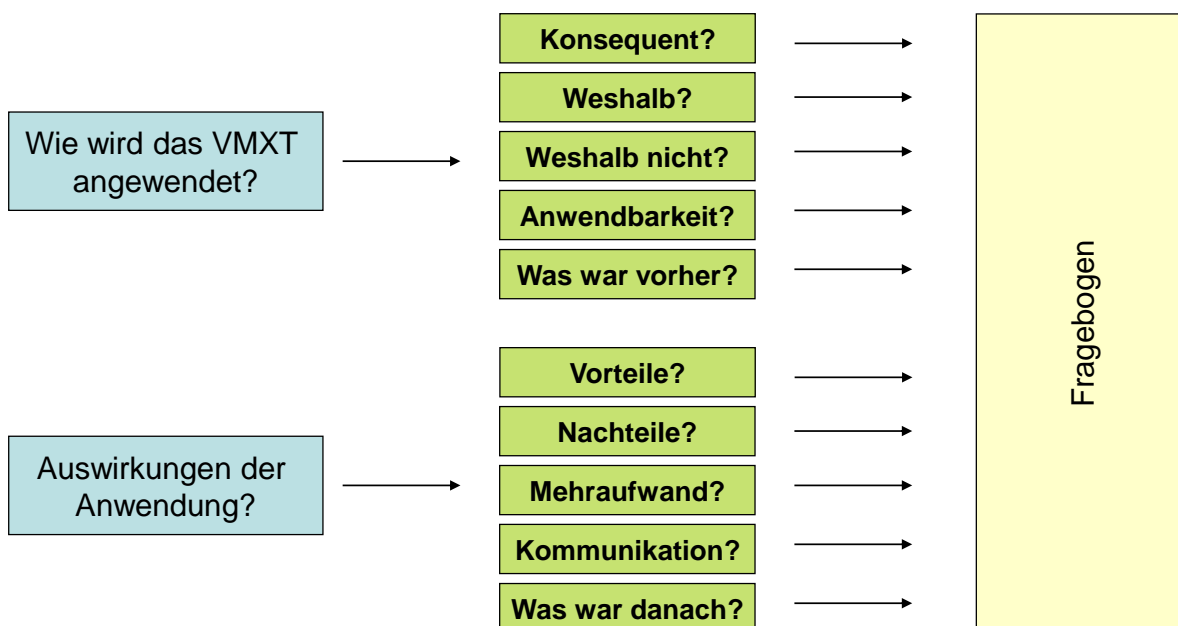
Telefoninterviews

- **Ziel:** Auswahl von potentiellen Teilnehmern für Schritt 2 (Fragebogen)
 - Erläuterung der Studie
 - V-Modell XT Erfahrung?
 - Bereitschaft zur Teilnahme?
- Auswahl der Telefoninterviewpartner
 - Zertifizierung
 - IT-Ansprechpartner in Behörden mit (vermutlichem) IT-Entwicklungsbedarf
 - In der Regel Projektleiter
 - „VMXT-Netzwerk“

Fragebogen- Struktur

- 58 Fragen
- Multiple-Choice
- Hintergrund, Anwendung, Know-how, Erfahrungen, Einschätzungen, Kontext
- Web-basiert
 - Alternativ: papier-basiert bzw. Interview
- Dauer: < 30 Minuten

Fragebogen- Methode



Erste Ergebnisse der Studie

SEE 2009 – Berlin

15

Statistik

- **30** Telefoninterviews mit Projektleitern aus Behörden (davon 28 Bund, 2 Land)
↓
- **16** Projekte mit VMXT-Anwendung
↓
- **6** ausgefüllte Fragebögen erhalten
+
- **2** Pilotprojekte in der BIT ausgewertet

SEE 2009 – Berlin

16

Ergebnis Telefoninterviews

- 16 Behörden setzen VMXT ein
 - ▶ davon 2 behördenspezifische Tailorings
- 3 Behörden zeigen Interesse
 - ▶ Alle haben bereits Schulungen durchgeführt
- 11 Behörden setzen VMXT nicht ein
 - ▶ davon haben 2 eigenes Vorgehensmodell
 - ▶ davon haben 5 eine ablehnenden Haltung zum VMXT

Fragebogen: Projektkontext

- Größe:
 - Personen: 4 bis 10 (sowohl auf AG- als AN-Seite)
 - Laufzeit: 12 bis 60 Monate
- Tailoring wurde in allen Projekten durchgeführt
- „Pionierprojekte“ (keine oder kaum vorherige VMXT-Erfahrung)
 - Allerdings 4 von 6 Projekten mit VMXT-Wissen
- 2 von 6 Projekte standen unter besonders hohem Zeitdruck

Wie wurde das VMXT angewendet?

- „1:1 nach Lehrbuch“
 - Bewusste Abweichungen: 1 Projekt
- **Einhaltung** des VMXT wird kaum kontrolliert
 - Reviews/Checklisten in nur 1 Projekt
- **Verantwortung:** VMXT-Verantwortlicher in 2 Behörden
- Einsatz des VMXT erfolgte mehrheitlich **freiwillig**
 - Gegenüber: hohe Eigenmotivation der Mitarbeiter
- **Werkzeugunterstützung**
 - 2 Projekte ohne PM-Werkzeuge → beide verzögert

Auswirkung des VMXT-Einsatzes

- Produktqualität und Leistungsumfang:
 - eher positiv
- Wissenstransfer AN → AG:
 - positiv
- Vergleich zum Nicht-Einsatz:
 - Flexibilität: neutral
 - Mehraufwand: neutral – leicht erhöht (nicht wg. Dokumentation)
 - Kommunikation: besser
- **Regelungsdichte** wird mehrheitlich als zu hoch empfunden
- Alle Projekte wurden **erfolgreich** abgeschlossen (oder erwartet)

Weitere Ergebnisse der Fragebögen

- **Managementbeachtung:**
 - Mehrheitlich **wenig oder keine** Beachtung des VMXT durch das Management einer Behörde
- **Verbindlichkeit** der Anwendung? → **nicht durchgängig** (bekannt)
- **Schulungsangebote** der BAKöV **werden genutzt**
- **Erster Einsatz**
 - Mehrfach **methodische Fehler** beim ersten Einsatz
- **Weiterer Einsatz** des VMXT
 - Einheitliche Bereitschaft für **Folgeprojekte** vorhanden

Feststellungen

- Einsatz des V-Modell XT in Bundesbehörden noch im Frühstadium
- Schulungen für selbständigen Einsatz nicht ausreichend
 - ▶ Erster Einsatz ohne externes Coaching führt zu methodischen Fehlern und erhöhten Arbeitsaufwänden
- Vertieftes Wissen über Anwendung des VMXT in Behörden soll erreicht werden
- Anwendung des VMXT soll verbreitert werden



Ausblick



- **Datenbasis vergrößern**
 - Erste Ergebnisse sind nicht statistisch signifikant
 - Übereinstimmungen zwischen analysierten Projekten vorhanden
 - Kurzversion Fragebogen (< 10 Minuten)

- **Schritt 3: vertiefende Interviews**
 - Individuelle Problematik in Projekten analysieren
 - Zusammenhang zwischen Anwendung und Projekterfolg

- **V-Modell BUND**
 - Organisationspezifische Anpassung für Bundesbehörden
 - Spezifika: Vergabe, Terminologie, Beteiligungsverfahren...
 - Anwendbarkeit & Akzeptanz erhöhen



Fragen



4.2. ITIL Prozesse in der BWI Systeme GmbH

4.2. ITIL Prozesse in der BWI Systeme GmbH

Wolfgang Treder

SDIT BKZ IT Service Management Prozessmanager Release Management

ITIL Prozesse in der BWI Systeme GmbH

Einführung und Praxiserfahrungen am Beispiel der Prozesse Change Management und Release Management

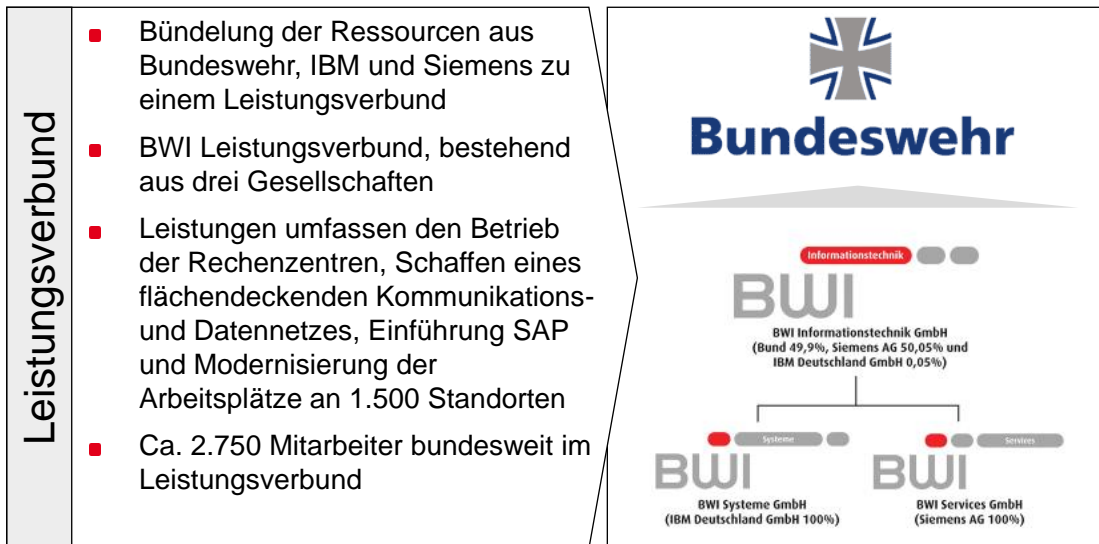
Wolfgang Treder SDIT BKZ IT Service Management
Prozessmanager Release Management
26.05.2009



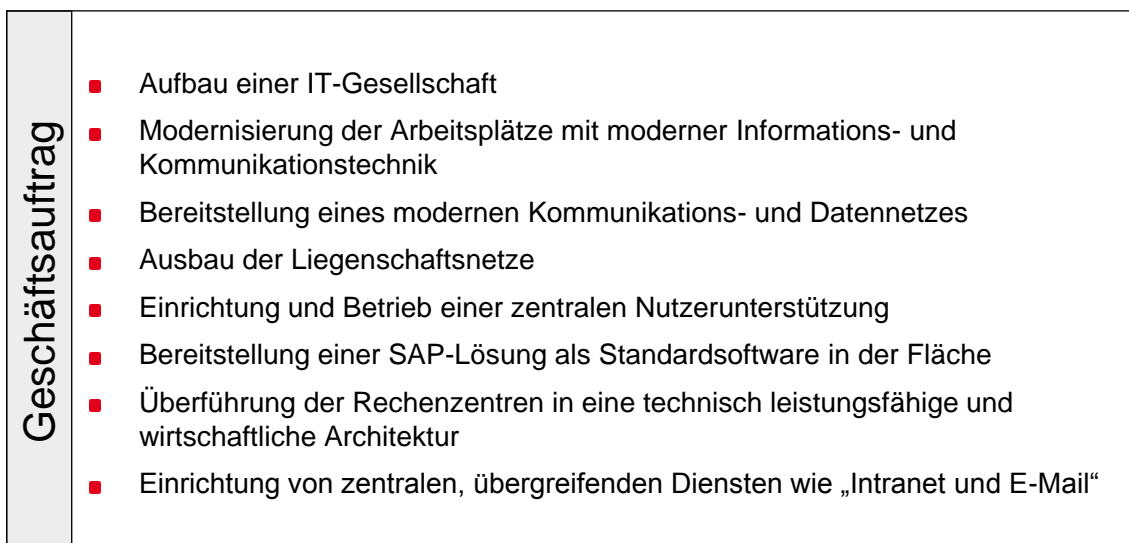
Agenda

Inhalte	■ Rahmenbedingung
	■ Kunde
	■ Projekt HERKULES
	■ SD IT Service Management
	■ Prozesslandkarte HERKULES
	■ ITIL Prozesse
	■ Design Phase
	■ Prozesse Change und Release Management
	■ Einführung der Prozesse
	■ Key Performance Indikatoren
■ Applikationseinsatz	
■ Reflektion	
■ Praxiserfahrung	
■ Lessons Learned	

Der Leistungsverbund BWI unterstützt die Informations- und Kommunikationstechnologie der Bundeswehr



Die Anforderungen der Bundeswehr stehen im Vordergrund unseres Handelns



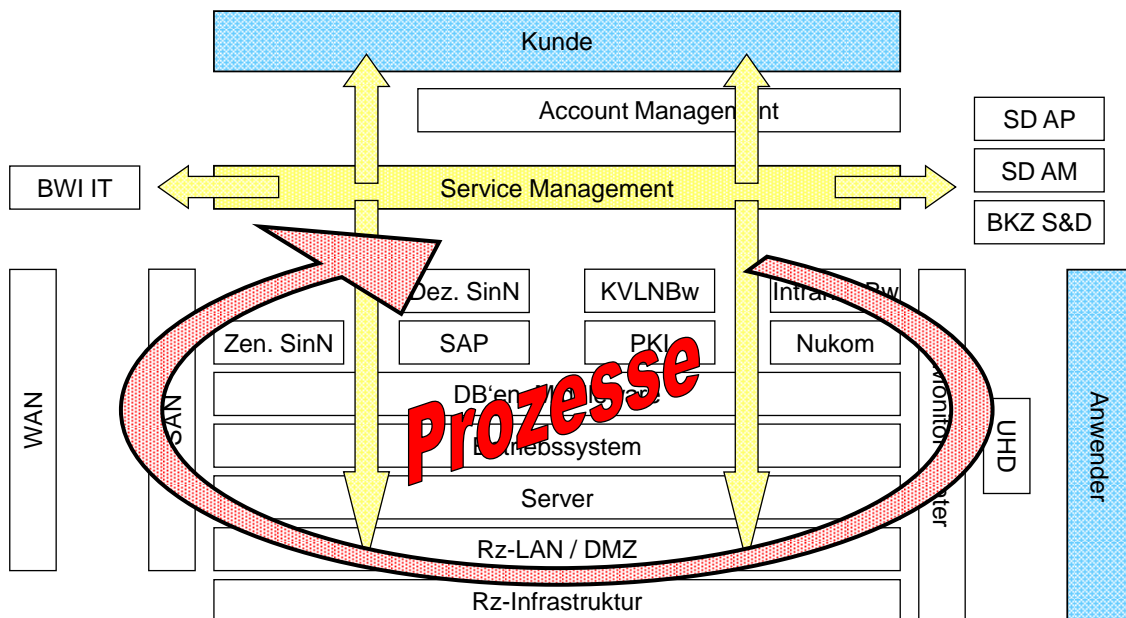
Das Projekt hat eine Laufzeit von 10 Jahren

Projektumfang	<ul style="list-style-type: none"> ■ Vollständige Erneuerung der bestehenden IT Infrastruktur der Bundeswehr innerhalb der ersten 4 Vertragsjahre (Integrationsphase) ■ Im Rahmen der Ersterneuerung werden insbesondere 140.000 PC, ca. 4.500 verteilte Rechner, die zentralen Systeme in den Rechenzentren, sowie die Netzwerkinfrastruktur (WAN & LAN) erneuert ■ Ausrollen von SAP für 45.000 Hauptnutzer ■ BWI betreut alle Standorte in Deutschland ■ Dazu gehört der Betrieb der Rechenzentren und Applikationen (zentral) und die Erbringung von Desktop Services & Netze Services (dezentral, ca. 1.500 Standorte)
---------------	--

Der IT Service wird zentral gesteuert und vor Ort erbracht

Standortkonzept	<ul style="list-style-type: none"> ■ Zwei Standorte (Bonn/Rheinbach und München) für die Betriebskompetenzzentren (BKZ) für IT- und Kommunikationsdienste ■ Vor-Ort-Betreuung ist nach 5 Regionen organisiert, mit insgesamt 22 Servicecenter-Standorten und 3 zusätzlichen Techniker-Standorten ■ User Help Desk wird auf 4 Standorte (Meckenheim, München, Berlin und Hannover) verteilt ■ Rechenzentren an 3 Standorten (Strausberg, Wilhelmshaven und Bonn, plus Außenstelle Köln-Wahn) ■ BWI-Verwaltungssitz in Meckenheim auf über 7.500 m² 	
-----------------	---	--

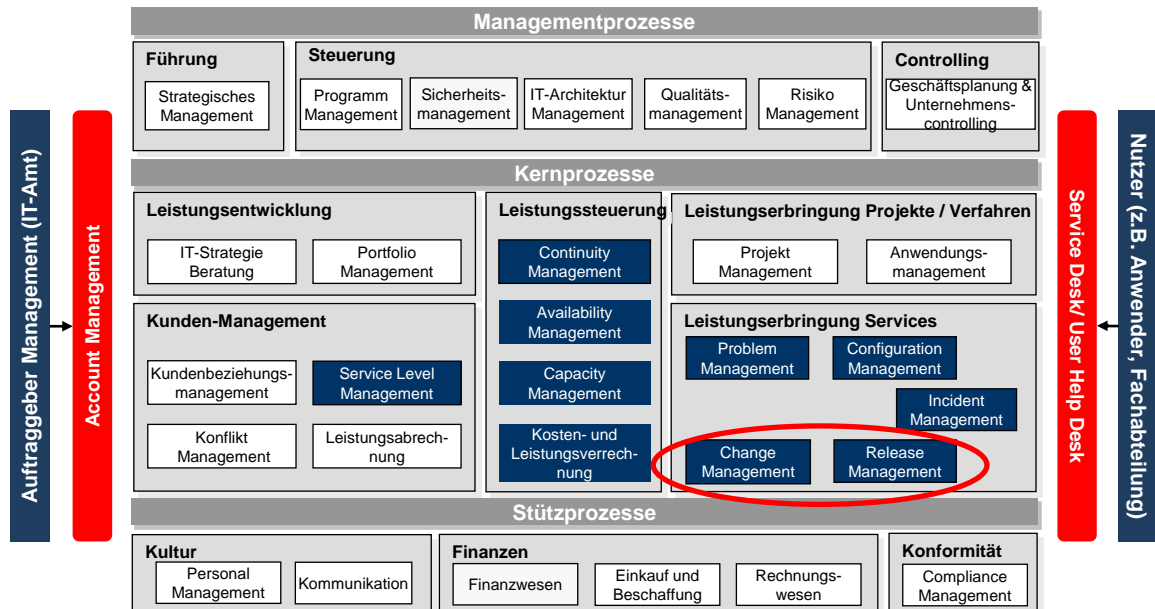
Kommunikationsbeziehungen in der Organisation



Warum Prozesse?

Nutzen	<ul style="list-style-type: none"> ■ Aufbau eines IT-Dienstleisters in kürzester Zeit ■ Mitarbeiter aus unterschiedlichen beruflichen „Kulturen“ ■ klare Vorstellung der wechselseitigen Anforderungen und Verantwortlichkeiten ■ Vermeidung von unklaren Zuständigkeiten ■ gezielte und gesteuerte Kundenkommunikation ■ klare und einheitliche Mess- und Bewertungskriterien für Prozesse ■ rechtzeitiges Erkennen von eventuellen Schwachstellen ■ Verbesserung der IT-Service-Qualität ■ Kostenoptimierung ■ Steigerung der Reaktionsgeschwindigkeit
--------	--

Das Geschäftsprozessmodell der BWI basiert auf ITIL



Design Phase

Rahmenbedingungen	<ul style="list-style-type: none"> ■ schlank modelliert ■ leicht verständlich ■ ITIL konform – Vertragsbestandteil ■ hohe Skalierbarkeit ■ konsortial einsetzbar ■ Intergration von bewährten Prozessteilen ■ frühzeitiges Einbeziehen der Betriebe ■ Modellierungstool: ARIS Business Designer
--------------------------	---

Warum ITIL

Entscheidungskriterien	<ul style="list-style-type: none"> ■ Best Practice <ul style="list-style-type: none"> ■ alle relevanten Prozesse sind in ITIL beschrieben ■ Schnittstellen zu Prince2 ■ Methode für große Organisationen ■ hoher Bekanntheits- und Wissensgrad ■ Vorteil bei der Personalauswahl ■ einheitliche Sprache und klar definierte Begriffe ■ Vertragsbestandteil 	
------------------------	--	--

Ziele

Change/Release Management	<ul style="list-style-type: none"> ■ Änderungen an der IT-Infrastruktur kontrolliert durchführen ■ Änderungen effizient durchführen – ohne den Betrieb zu hindern ■ Anzahl von Änderungen vermindern ■ Risiken von Änderungen minimieren ■ Ausfallzeiten im operativen Betrieb reduzieren ■ effiziente und kostengünstige Implementierung autorisierter Changes mit minimalem Risiko für die bestehende und neue IT-Infrastruktur ■ Servicequalität verbessern 	
---------------------------	---	--

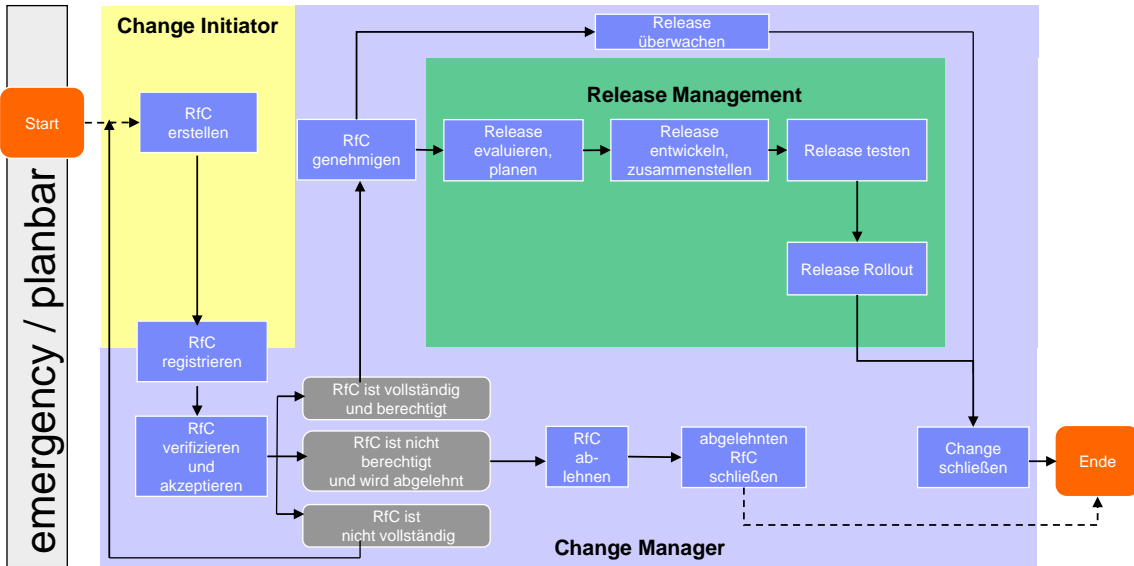
Aufgaben

Change Management	<ul style="list-style-type: none">■ Verantwortung für die kontrollierte Umsetzung von Änderungen<ul style="list-style-type: none">■ registrieren, klassifizieren■ autorisieren■ koordinieren und überwachen■ prüfen■ evaluieren■ Vorgabe des Rahmens für die kontrollierte Umsetzung von Änderungen durch das Release Management■ Durchführung CAB/ECAB
-------------------	--

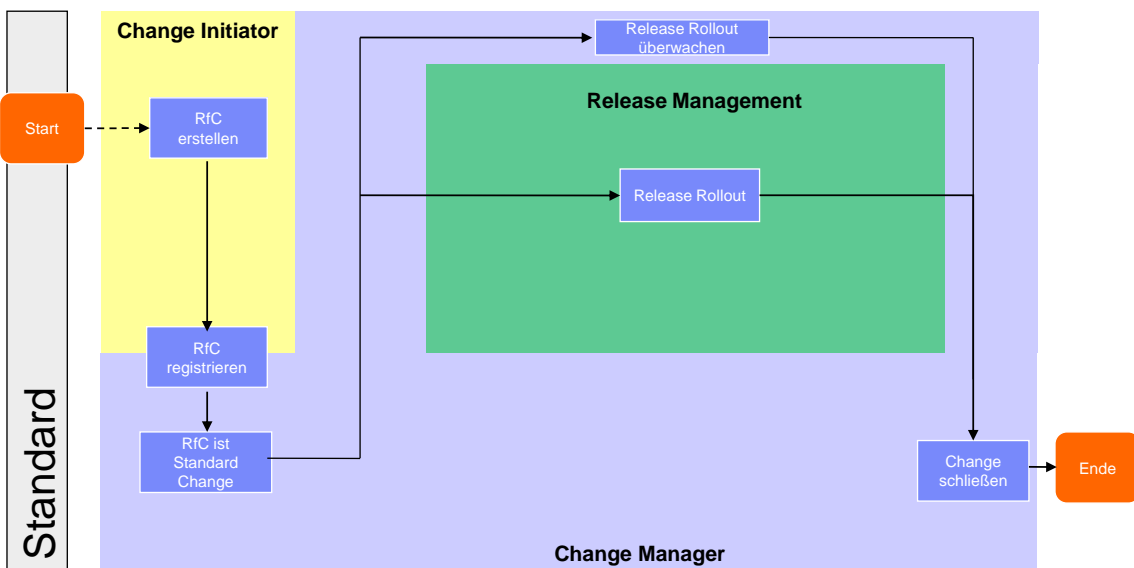
Aufgaben

Release Management	<ul style="list-style-type: none">■ Definition, Planung und Umsetzung des Releases<ul style="list-style-type: none">■ Release Design, Build, Tests (inclusive Back-out Plan) und Abnahme■ lagern der freigegebenen Hardware im Definitive Hardware Store (DHS)■ speichern der freigegebenen Software in der Definitive Software Library (DSL)■ Planung und Durchführung des Rollouts<ul style="list-style-type: none">■ information und ggf. Schulung vor der Auslieferung■ installieren neuer oder geänderter Hardware■ verteilen und installieren der Software■ Produktionsfreigabe
--------------------	---

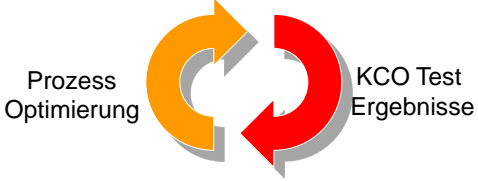
Prozesse Change und Release Management



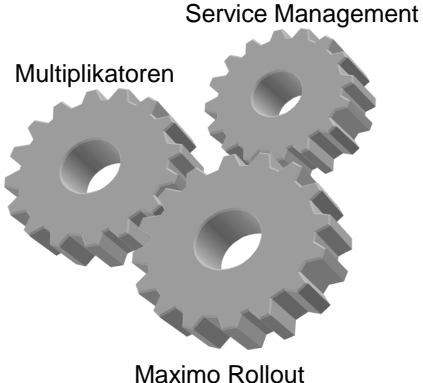
Prozesse Change und Release Management



Key Performance Indikatoren

Leistungsindikatoren	<ul style="list-style-type: none">■ Anzahl Changes<ul style="list-style-type: none">■ registriert■ abgelehnt■ emergency, planbar, Standard■ genehmigt / nicht genehmigt■ abgebrochen■ Anzahl Releases<ul style="list-style-type: none">■ zeitgerecht erstellt und verteilt■ bei denen Backout notwendig war	 <p>Prozess Optimierung</p> <p>KCO Test Ergebnisse</p>
-----------------------------	---	--

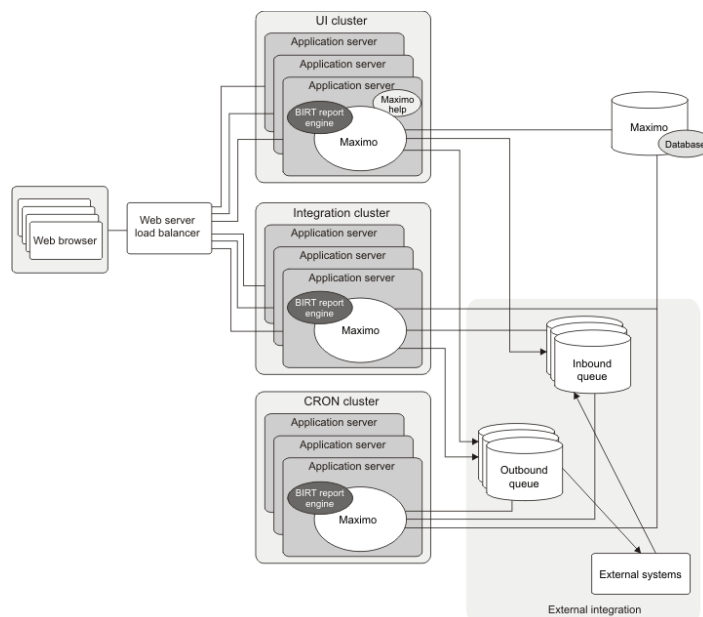
Einführung der Prozesse

Prozess Rollout	<ul style="list-style-type: none">■ Informationsveranstaltungen an den Rz-Standorten<ul style="list-style-type: none">■ Prozesspräsentationen■ Fragen aus der Praxis■ Key User-Ausbildung■ Coaching vor Ort■ Multiplikatoren identifizieren■ Operatives Service Management■ Change Office■ abgestimmt mit dem Maximo Rollout	 <p>Service Management</p> <p>Multiplikatoren</p> <p>Maximo Rollout</p>
------------------------	--	---


Applikationen in IBM Tivoli Asset Management for IT (TAMIT)




Erweitertes System Setup Tivoli Asset Management for IT



Praxiserfahrungen

Erfahrungen	<ul style="list-style-type: none">■ bessere Akzeptanz, wenn der Prozess vollständig in der Applikation abgebildet ist■ hohe Teilnehmerzahl an den Informationsveranstaltungen■ transparenter Prozess – hohe Verständlichkeit■ Prozessmodellierung ohne viele Ausnahmen■ hoher Abdeckungsgrad der Anwendungsfälle■ anfängliche Ablehnung■ spätere Akzeptanz■ ARIS Modellierung nicht allgemein verständlich	 <p>Auswirkung auf das Tagesgeschäft ???</p>
--------------------	---	---

Nutzen von Vorgehensmodellen

ITIL Change/Release Prozess	<ul style="list-style-type: none">■ standardisiertes Vorgehen■ Wissensbasis vorhanden und nutzbar■ schnelle Einarbeitungszeiten■ Flexibilität■ Begriffsdefinitionen vorhanden■ Schnittstellen zu anderen Prozessen geregelt■ hohe Akzeptanz durch Best Practice-Ansatz■ Übergang von der Softwareentwicklung zum Betrieb	 <p>Definitionen Wissensbasis Kommunikation</p>
------------------------------------	---	--

Schwächen von Vorgehensmodellen

ITIL Change/Release Prozess	<ul style="list-style-type: none"> ■ Nutzung von belegten Begriffen (Release, DSL, Configuration ...) ■ Realität hat viele Facetten ■ Methoden- und Modellmüdigkeit ■ pauschale Ablehnung ■ Trennung von Change- und Release-Prozess ■ Vermittelbarkeit ■ hohe Komplexität des Gesamtprozess-Modells 	
-----------------------------	---	--

Lessons Learned

Do's	<ul style="list-style-type: none"> ■ frühzeitiges Einbeziehen der Betriebe ■ keep it simple ■ Praxisszenario zur Prozessverprobung ■ praktische Umsetzungshilfe 	
Dont's	<ul style="list-style-type: none"> ■ Orientierung an Modell- und Meta-Ebene ■ viele Rollen im Prozess ■ absolute Ausnahmen modellieren ■ striktes Orientieren an Modellierungsrichtlinien 	

Vielen Dank für Ihre Aufmerksamkeit!



4.3. Kostenseitige Begleitung von Vorgehensmodellen


Joachim Schöffner
4cost GmbH



SEE 2009 Berlin - 26. Mai 2009

Kostenseitige Begleitung von Vorgehensmodellen. Von der parametrischen Kostenschätzung bis hin zur begleitenden Detailkalkulation

Joachim Schöffner
Geschäftsführender Gesellschafter 4cost GmbH

Powered by 

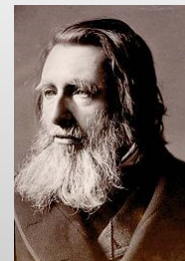
Weisheiten



Es gibt kaum etwas auf dieser Welt, das nicht irgend jemand ein wenig schlechter machen und etwas billiger verkaufen könnte, und die Menschen, die sich nur am Preis orientieren, werden die gerechte Beute solcher Machenschaften.

Es ist unklug, zu viel zu bezahlen, aber es ist noch schlechter, zu wenig zu bezahlen. Wenn Sie zu viel bezahlen, verlieren Sie etwas Geld, das ist alles. Wenn Sie dagegen zu wenig bezahlen, verlieren Sie manchmal alles, da der gekaufte Gegenstand die ihm zugedachte Aufgabe nicht erfüllen kann.

Das Gesetz der Wirtschaft verbietet es, für wenig Geld viel Wert zu erhalten. Nehmen Sie das niedrigste Angebot an, müssen Sie für das Risiko, das Sie eingehen, etwas hinzurechnen. Und wenn Sie das tun, dann haben Sie auch genug Geld, um für etwas Besseres zu bezahlen.



John Ruskin, engl. Sozialreformer (1819 - 1900)

Powered by 



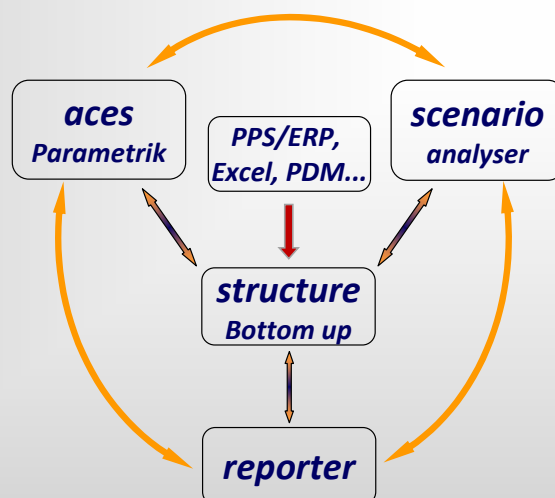
4cost Methodik?

Kalkulation und Kostenschätzungen zu unterschiedlichen Zeitpunkten unter Nutzung verschiedener Methoden...

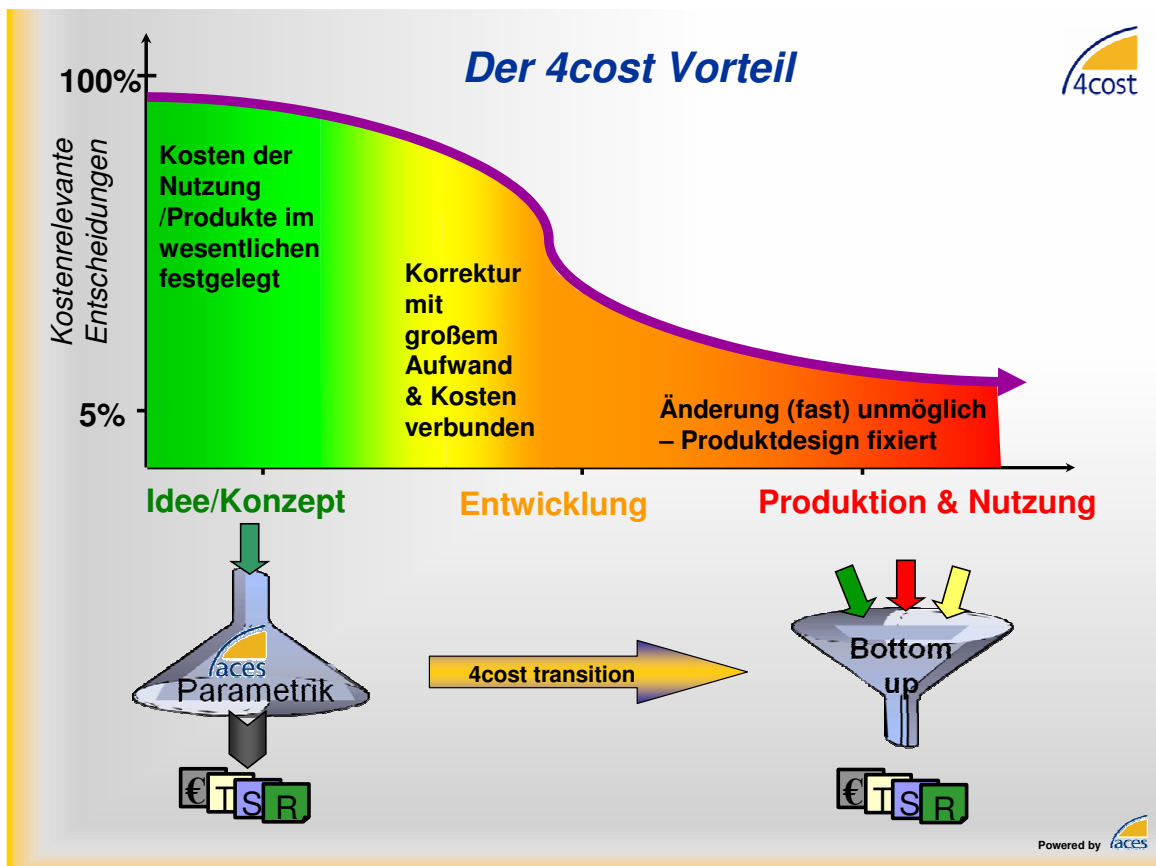
...als aktive und integrierte Management Methode!

Powered by faces

**Aktive und integrierte
Management Methode 4cost
mittels Parametrik und Detailkostenanalyse**



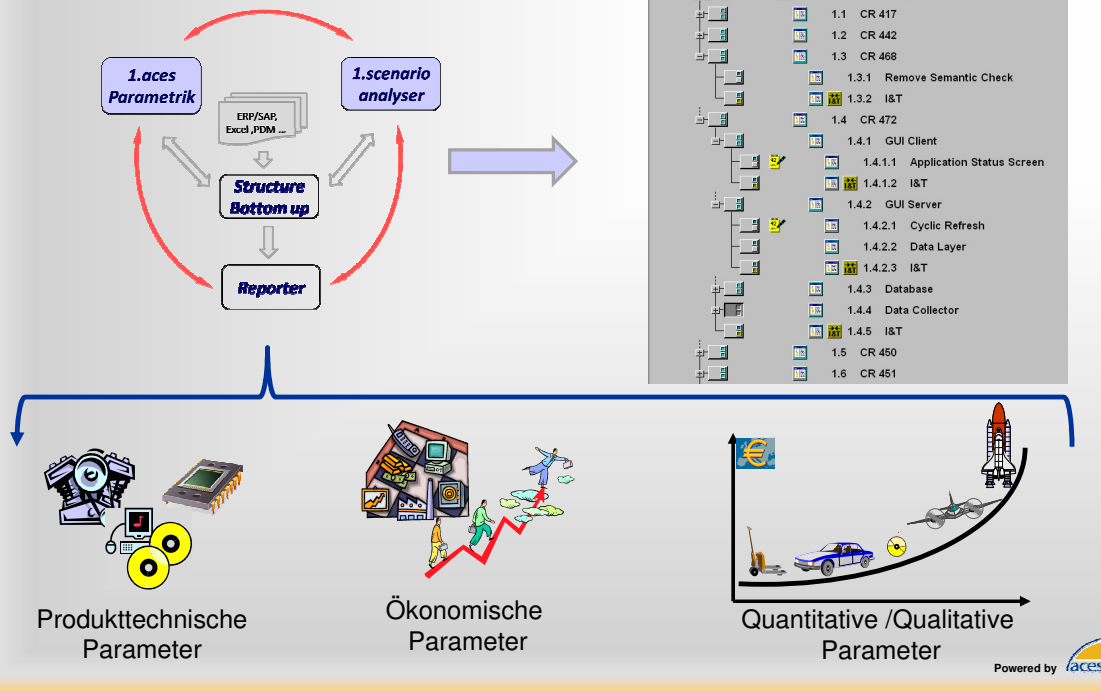
Powered by faces



4.3. Kostenseitige Begleitung von Vorgehensmodellen

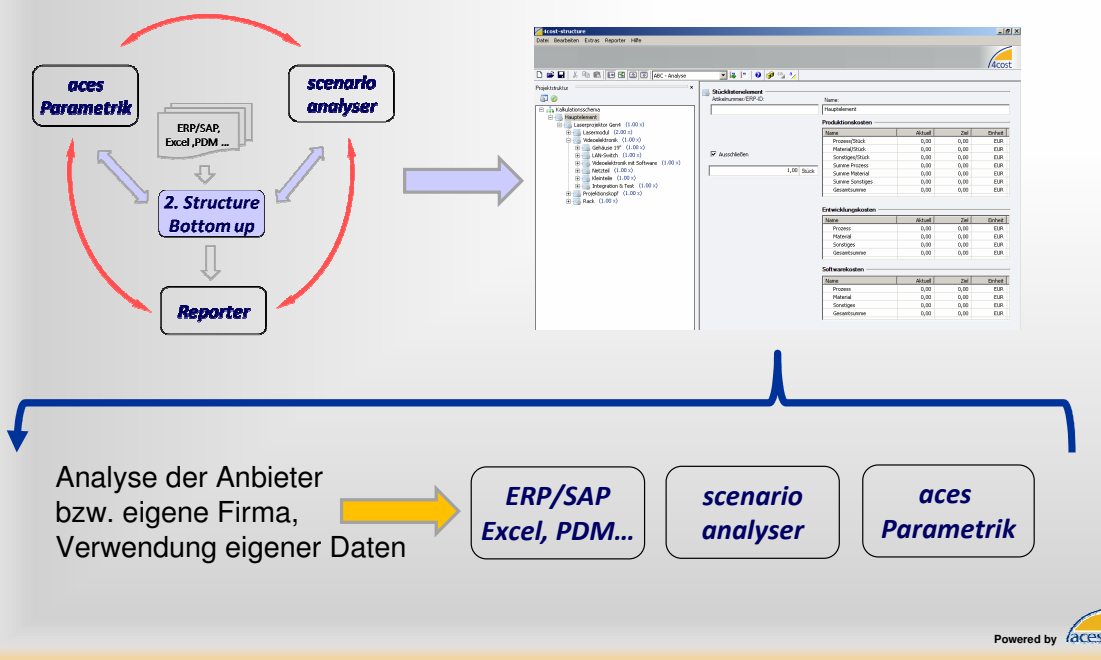
Die Methode 4cost

1. Parametrik - „Top-Down“ Prozess



Die Methode 4cost

2. Detail Kalkulation - „Bottom-Up“



Modul 4cost-structure – komfortable Strukturabbildung...



Projektstruktur

	Aktuell/Stock	Ziel/Stock	Aktuell/1.00 Stück	Ziel/1.00 Stück
	4.900,00	5.000,00	4.900,00	5.000,00
	0,00	0,00	0,00	0,00
	4.900,00	5.000,00	4.900,00	5.000,00
	0,00	0,00	0,00	0,00
IR				
			Enemkosten	Ziel
			1.200,00	2.000,00
			0,00	0,00
			1.200,00	2.000,00
			0,00	0,00
ER				
			Enemkosten	Ziel
			88.935,00	86.000,00
			88.935,00	86.000,00
			0,00	0,00
			0,00	0,00

Powered by

... mit detaillierter Kalkulationsmöglichkeit...



Prozess

Name:

Prozess:

Zuschläge

Gemeinkosten	Intern [%]	Extern [%]
Material	0,00	0,00
Lohn	5,00	0,00
Vertrieb	10,00	0,00
Entwicklung	5,00	0,00
Verwaltungs	8,00	0,00
Sonder	1,00	0,00

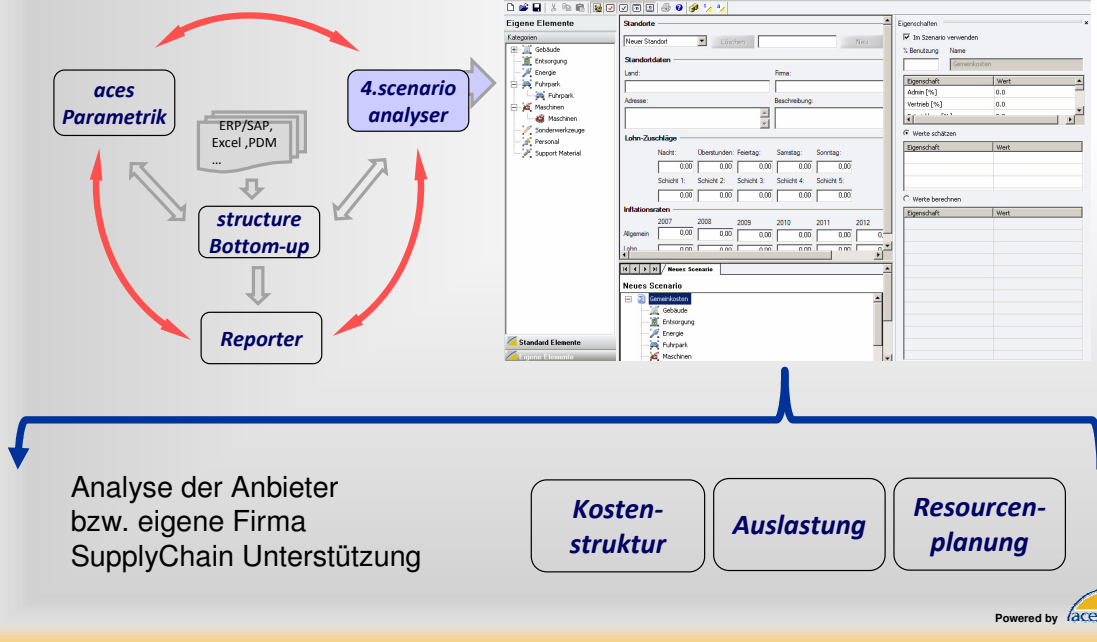
Name	Basis/EUR	Ziel/EUR	Aktuell/EUR	Gemeinkoste...	Menge/Ein...	Einheit	EUR/Einheit	Modifikator	Bezugsgröß
Produktion	0,00	0,00	0,00	Interne	0,00		0,00	0,00	0,0
Entwicklung	0,00	0,00	0,00	Interne	0,00		0,00	0,00	0,0
Software	84.700,00	86.000,00	88.935,00	Interne	0,00		2.630,00	0,00	0,0
Predesign	6.000,00	6.000,00	6.300,00	Interne	50,00	h	120,00	1,00	1,0
Designing	8.400,00	80.000,00	8.820,00	Interne	70,00	h	120,00	1,00	1,0
Coding	39.000,00	0,00	40.950,00	Interne	0,00		1.950,00	0,00	0,0
Dev 01	13.000,00	0,00	13.650,00	Interne	20,00	MT	650,00	1,00	1,0
Dev 02	13.000,00	0,00	13.650,00	Interne	20,00	MT	650,00	1,00	1,0
Dev03	13.000,00	0,00	13.650,00	Interne	20,00	MT	650,00	1,00	1,0
Reviewing	5.000,00	0,00	5.250,00	Interne	50,00	h	100,00	1,00	1,0
Testing	18.000,00	0,00	18.900,00	Interne	150,00	h	120,00	1,00	1,0
Validation	4.800,00	0,00	5.040,00	Interne	40,00	h	120,00	1,00	1,0
Enhancement	3.500,00	0,00	3.675,00	Interne	35,00	h	100,00	1,00	1,0

Powered by

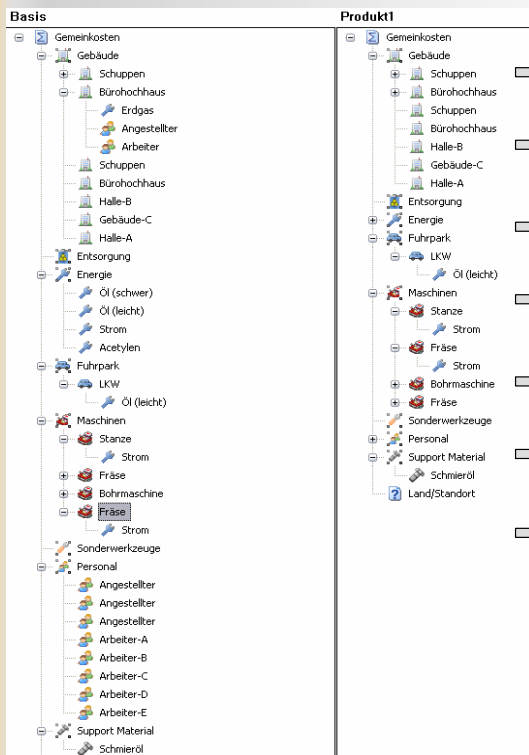


Die Methode 4cost

4. Die Szenarioanalyse



4cost-scenarioanalyser



- Overheads – Lieferantenanalyse
- Szenario auf das Produkt abbilden
- Datenbank –universal/kundenspez.
- Auslastungsanalysen
- Produktbezogene Kostensätze
- Variables Produkt Szenario
- Akkurate / realistische Kosten

Die Methode 4cost

3. Das Reporting

```

    graph TD
      A[acs Parametrik] --> C[Structure Bottom up]
      B[scenario analyser] --> C
      C --> D[3. Reporter]
      E[ERP/SAP, Excel, PDM...] --> C
      
```

Standardisierte Reports und Darstellung der Ergebnisse

Name	Einheit	Standard	Preis	Werte	Einheit	Werte	Einheit	Werte	Einheit	Werte
...
...
...

- Gehäuse komplett - Amuc : 6.14%
- Laserquelle Kanal Grün - Amuc : 34.51%
- ... - Amuc : 17.39%
- ... - Amuc : 19.82%
- ... - Amuc : 9.25%
- ... - Amuc : 12.98%

Automatisierte Erstellung eigener Reports

ERP/SAP
Excel, PDM...

scenario
analyser

acs
Parametrik

structure
Bottom-Up

Powered by

Die Methode 4cost

Ziel sind angemessene Kosten

Überteuert „monopolistisch“

Zu günstig, „Kampfpreis“, Folgekosten!

Dauerhaft realistische Kosten und Preise

Einführungsphase der Methode | Regelmäßige Anwendung | Etablierte Methode

Powered by

Parametrik

$Cost = f(K, M, T, R)$

Parametrik ?

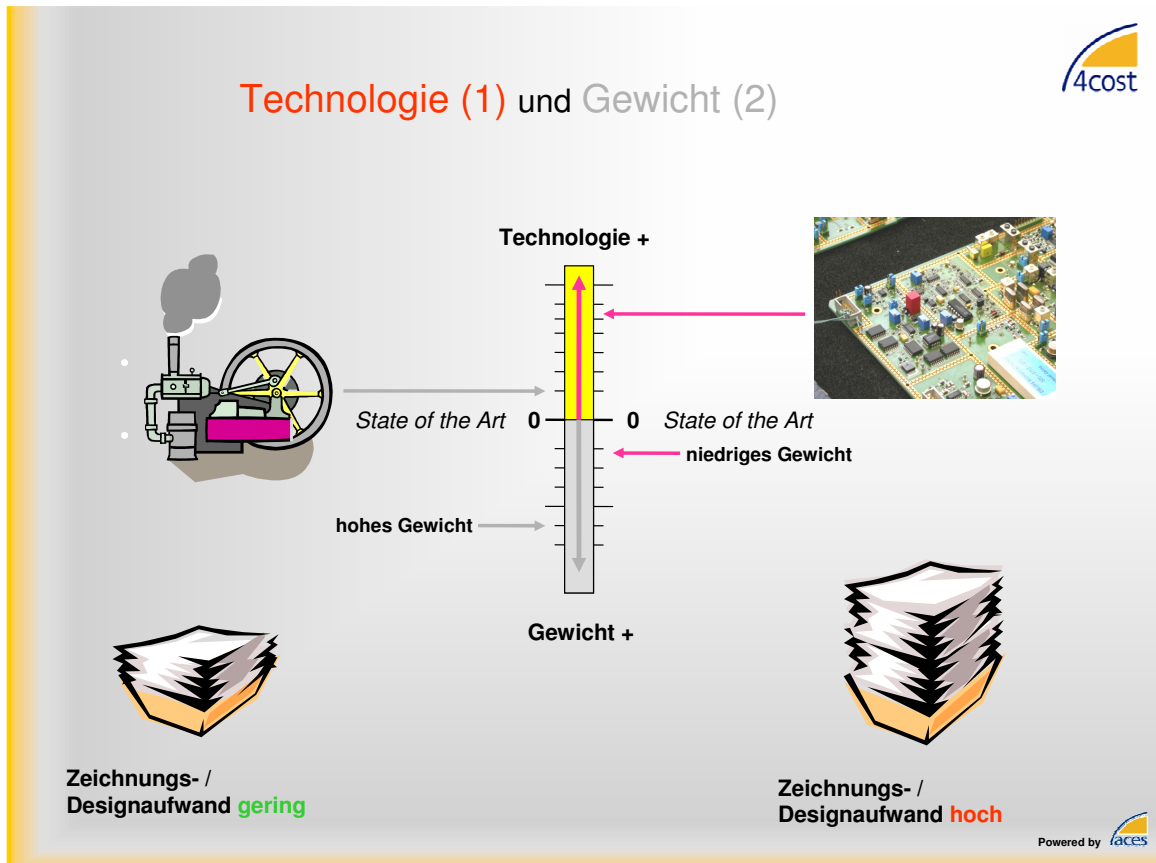
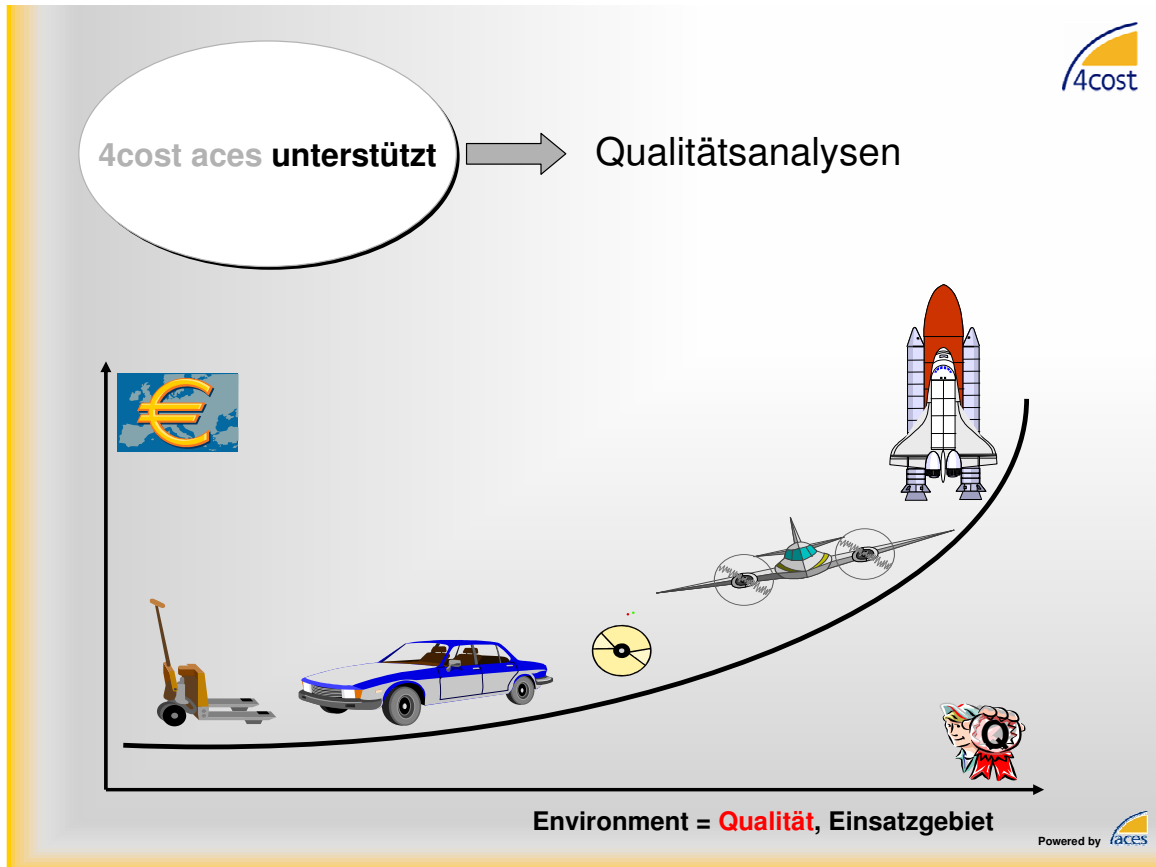
4cost

Powered by faces

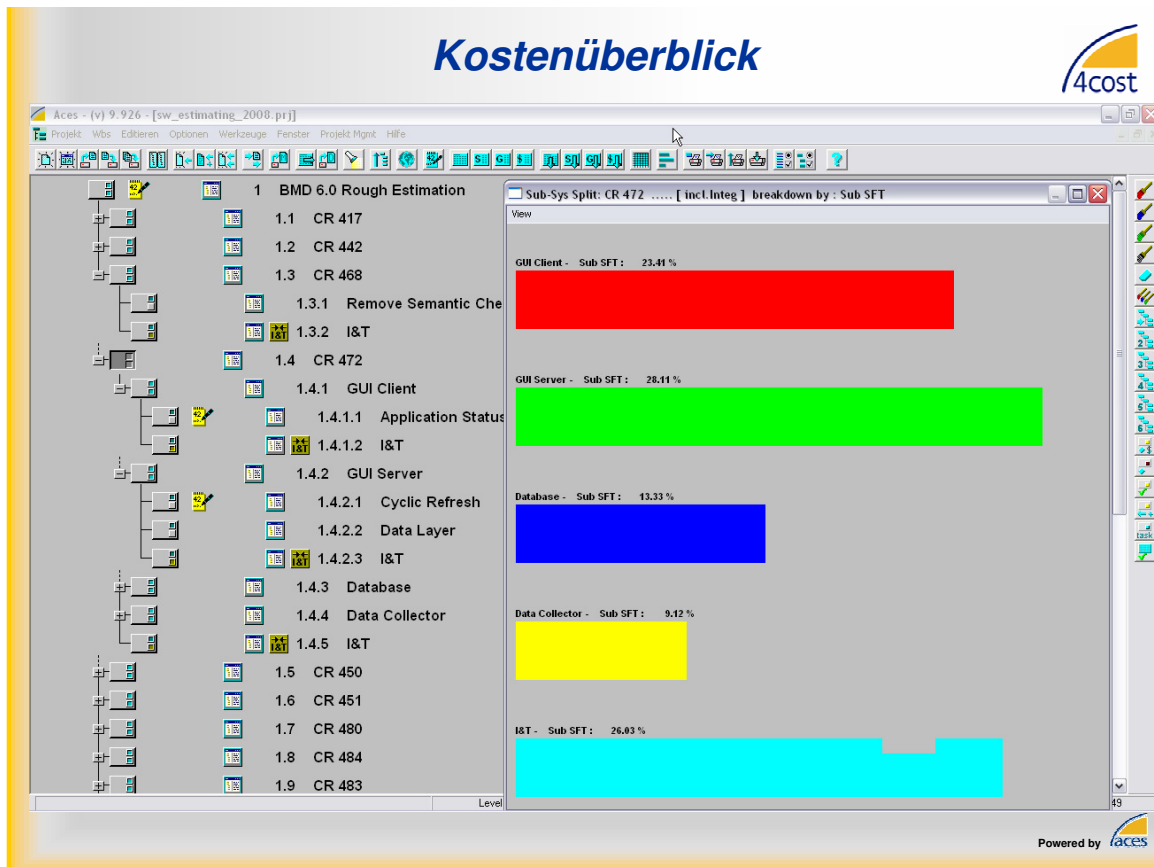
4cost-aces (Parametrik) Top - down - Ansatz

4cost

Powered by faces



4.3. Kostenseitige Begleitung von Vorgehensmodellen



4cost aces unterstützt

➔

Software Schätzmethode

The screenshot displays the 4cost software interface for a project titled 'Project: Software Entwicklung'. The hierarchical tree structure includes the following components:

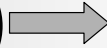
- 1 Project: Software Entwicklung
 - 1.1 Combat Execution
 - 1.2 Combat System Management
 - 1.2.1 Plans and Doctrine Management
 - 1.2.2 Situation Management
 - 1.2.3 Propagation and Detection Prediction
 - 1.2.4 I&T
 - 1.3 Navigation
 - 1.4 Resource Management
 - 1.5 Simulation & Training
 - 1.6 Subsystem Interface
 - 1.7 Infrastruktur
 - 1.8 I&T

- Objekt orientiert Methode
- Function Point Methode
- Analogien
- Personalbedarfs Abschätzung
- Budget abhängige Schätzung
- Source lines of Code (zur Kalibrierung)

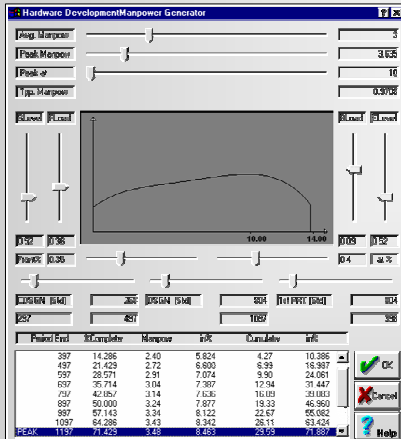
Powered by



4cost aces unterstützt



die Entwicklung HW/SW

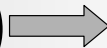


- strukturierte Darstellung (WBS)
- Design-Alternativen
- Personalverteilung
- Prototypkosten (HW)
- Entwicklungsdauer (Zeitachse)
- Penalties
- Team / Task
- Technologie

Powered by faces



4cost aces unterstützt



LCC Simulation

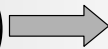


- Standorte weltweit
- Logistik-Szenarien
- Erstbevorratung (Eingabe)
- Erstbevorratung (Simulation)
- Wartungskonzepte
- Einsatzzeit bis zu 50 Jahren
- MTBF berechnet/ Eingabe
- MTTR berechnet/ Eingabe
- Grafische Darstellungen

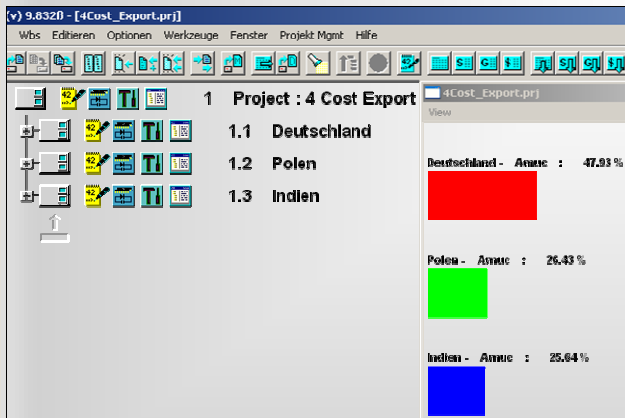
Powered by faces



4cost aces unterstützt



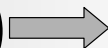
Länderspezifische Analysen



Länderspezifische Module oder Baugruppen können in einer Struktur abgebildet werden (Repräsentation in „neutraler“ Währung)
z. B. €



4cost aces unterstützt



die Fertigung / Produktion

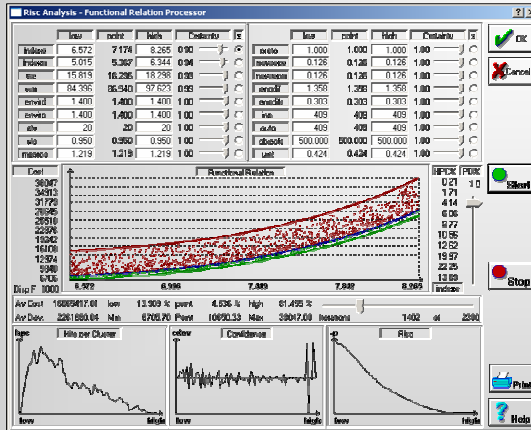


- Strukturierte Darstellung (WBS)
- Produktionsmethoden
- Materialkosten
- Wertschöpfung
- Produktionsdauer
- Penalties
- Losproduktion
- Technologie
- Lernkurven



4cost aces unterstützt

Sensitivitäts- und Risikoanalysen



- Technologie
- relevante Eingabeparameter
- Zeitplan
- Software
- Funktionale Zusammenhänge

Powered by faces

Projektorganisation mit integrierter Datenbank



Mit Drag und Drop aus der Datensammlung ein neues Projekt aufbauen

Number	Title	Date
0001 000	STRUKTUR ALLGEMEIN	24-8-2006
0002 100	SCHALE KOMPLETT	24-8-2006
0003 101	MONT. EINBAU, ZUSB. REP	24-8-2006
0004 110	BEPL. BESCHLAG REP	24-8-2006
0005 111	MONT. EINBAU, ZUSB. REP	24-8-2006
0007 113	DOPPLER REP	24-8-2006
0009 116	BEPL. BESCHLAG REP	24-8-2006
0010 118	KALOTTENBLECH PART	24-8-2006
0011 119	SONST. STRUKT. BEPL. REP	24-8-2006
0012 120	STRINGER ASSY	24-8-2006
0013 121	MONT. EINBAU, ZUSB. STRINGER REP	24-8-2006
0014 122	STRINGER REP	24-8-2006

Powered by faces

Selektierter Export *aller* Modelldaten möglich

Daten Export

Daten Vor-Selekt:

- Alle
- Input Param
 - Hardware
 - Software
- Kosten/Ausga.
 - Hardware
 - Software
- System Param.
- Vor-Definiertes Exp.
 - Granada
 - 4Cost

#	BoxType	Labour Dev Ttl	Mater. C
1	1000	366855	3760
2	1000	366746	3760
3	1000	74531	752
4	10200	0	752
5	10100	50973	752
6	14000	0	752
7	11000	13365	44
8	1000	74531	752
9	10200	0	752
10	10100	50973	752
11	14000	0	752
12	11000	13365	44
13	1000	74531	752
14	10200	0	752
15	10100	50973	752
16	14000	0	752
17	11000	13365	44

Template Datei: C:\Programme\aces\Templates etc\LEADS Standard Prd Lab_Mat_2005_11_03_12h00.tpl
Export Datei:

Separator: Tabs Semikolon Blankis Komma Punkt

Generiere Zeile für: Projekt Sub-Systeme Elemente Keine Integrierten

Zellen Prefix: Wdh. Nummer Strukturiert Ebene Integrierte Kennung Rot Grün Blau Schwarz

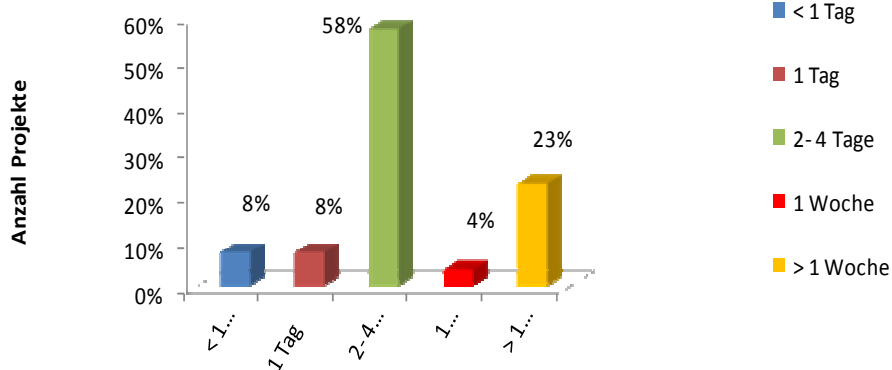
Optionen: Sende Daten zum Clipboard Erstelle Datei <PjName>.tbl Komplexer Param. Datum: 104 -> 01-01-2004

Buttons: Cancel, Help, Do Table, Send Data, OK

4cost-aces unterstützt

**Schnelle
und zuverlässige
Schätzungen**

Typische Schätzdauer von Projekten mit dem parametrischen Modell 4cost-aces



4. Session 4: Nutzen und Schwächen von Vorgehensmodellen

4cost

Detaillierte Kostendarstellung:

- HW Produktion
- HW Entwicklung
- SW Entwicklung
- HW & SW Integration
- Zusammenfassung der Eingabeparameter
- Zeitachsendarstellung

Powered by **faces**

4cost

Powered by **faces**



Kunden und Anwender

..... 

 **Branchen:**

- Raumfahrt
- Luftfahrt
- Automobil
- Konsumgüter
- Defence
- Telekommunikation
- Software Systeme

 Aircatel-Lucent

















Powered by 



Danke für Ihre Aufmerksamkeit!

Powered by 

5. Session 5: Agiles Schätzen und Planen

Inhalt

5.1. Agiles Schätzen und Planen	182
5.2. Planungsebenen der iterativen Systementwicklung	198
5.3. Pragmatische Aufwandsschätzung	212

5.1. Agiles Schätzen und Planen

Jens Clodewey
Coldewey Consulting
Toni-Schmid-Str. 10 b
81825 München
jens_coldewey@acm.org

Abstract

Agile Projekte planen anders: Statt umfangreicher Gantt-Diagramme tauchen plötzlich Karteikarten auf, Insider werfen mit Begriffen wie Schätzpoker und Story Points um sich und Controller bangen um ihr mühsam aufgebautes Berichtswesen.

Dieser Beitrag gibt einen Überblick über die Grundideen agilen Planens, stellt Praktiken vor und blickt dann über den Tellerrand von Iteration und Sprints: Welche Vor- und Nachteile birgt ein solches Planungsverfahren? Was ist bei organisationsweitem Einsatz zu beachten? Und wie können die dabei „abfallenden“ Informationen genutzt werden, um neue Geschäftsmodelle zu finden.

Ziele agiler Planung

Wenn agile Planung richtig durchgeführt wird, leistet sie einen wichtigen Beitrag, das Projekt zum Erfolg zu führen. Dabei ist Erfolg nicht als Einhaltung von Zeit, Umfang, Budget und Qualität definiert, sondern als Beitrag zur Maximierung des Geschäftsnutzens der Software. Das bedeutet, dass agile Planung sowohl die Effektivität der Softwareentwicklung fördern sollte, als auch deren Effizienz - und zwar in dieser Reihenfolge. Zur Förderung der Effektivität werden mehrere Teilziele verfolgt:

- Das Team soll sich darauf fokussieren durch die Anwendung den geschäftlichen Mehrwert des Systems zu optimieren
- Das Planungsverfahren soll Änderungen unterstützen und nicht erschweren
- Risiken sollen minimiert werden

Weitere Teilziele sollen die Effizienz der Entwicklungsarbeit optimieren:

- Voneinander abhängige Aktivitäten sollen koordiniert werden
- Die Planung soll realistische Voraussagen ermöglichen, damit sich andere Projektbeteiligte rechtzeitig darauf einstellen können
- Die Planung sollte eine Basis für realistische Statusbestimmungen liefern

Interessant sind aber auch Ziele, die oft mit Plänen verbunden werden, die aber von agilen Planungsverfahren explizit nicht verfolgt werden:

- Es soll keine korrekte Vorhersage des Projektablaufs erstellt werden, weil das nicht möglich ist. Erfolg ist der geschaffene Mehrwert, nicht das Einhalten eines Plans
- Die Mitarbeiter- und Ressourcenbelegung soll nicht vorausgeplant werden, weil diese Pläne in einem nicht vorhersehbaren Umfeld sehr aufwändig und fragil sind und keinen Nutzen stiften
- Agile Planung ist kein Vehikel, um über „ambitionierte Pläne“ Druck auf das Team auszuüben

- Agile Planung ist kein Vehikel, um Verantwortung weiter zu schieben, indem man unrealistische Zulieferungstermine aufstellt und dann den Lieferverzug verantwortllich macht für eigene Verzögerungen
- Agile Planung erzählt Managern nicht, was sie gerne hören wollen, sondern bietet einen realistischen Blick auf das, was möglich ist

Agile Planung ist damit ein wichtiges Instrument für das Team und das Management, die Wertschöpfung eines Projekts nicht so sehr trotz, sondern mit Hilfe vieler Änderungen zu optimieren.

Die Grundlagen

Agile Teams planen zu Beginn jeder Iteration. Übliche Rhythmen sind wöchentlich, 14-tägig oder monatlich. Zusätzlich können auch noch Quartals- oder Releaseplanungen hinzukommen. Am Planungsmeeting nimmt das gesamte Team teil. Die Struktur eines Planungsmeetings kann man mit „Priorisieren und Abschöpfen“ zusammen fassen:

1. Feststellen des aktuellen Status, oft in Form einer Demonstration und verbunden mit einer Abnahme
2. Ermitteln der aktuellen Velocity, also des geschätzten Aufwandes, der in der letzten Iteration abgeschlossen wurde
3. Einplanen der nächsten Aufgaben, indem für jede Aufgabe die folgenden Schritte durchgeführt werden:
 - Vorstellung der nächsten Aufgabe in der Reihenfolge ihrer Priorität. Dabei sind die Aufgaben vollständig priorisiert, es existiert also eine eindeutige Reihenfolge, die sich aus Geschäftsnutzen, Risiken und Abhängigkeiten ergibt. Die Aufgaben werden im sogenannten „Backlog“ gesammelt und sortiert.
 - Diskussion der Aufgabe, bis sie von allen Beteiligten in ausreichender Tiefe verstanden wurden, um geschätzt werden zu können.
 - Schätzung der Aufgabe. Dabei wird „Schätzpoker“ gespielt, wobei alle mitschätzen, die an der Umsetzung der Aufgabe beteiligt sind. Das entspricht einer Expertenschätzung nach der „Breitband-Delphi-Methode“. Fachlich Verantwortliche und Projektleiter nehmen keinen Einfluss auf die Schätzungen.
4. Es werden so lange Aufgaben verplant, bis die Iteration „gefüllt“ ist, die Summe der Aufwände also die aktuelle Velocity erreicht hat. Zur Sicherheit werden eine bestimmte Anzahl weiterer Aufgaben als Nachrücker definiert.
5. Produktverantwortliche und Team überprüfen die Planung noch mal und legen sie dann als verbindliche Grundlage der Iteration fest. Das Team verpflichtet sich, die Planung diszipliniert von oben nach unten abzuarbeiten. Es verpflichtet sich jedoch nicht, alle Aufgaben bis zur Wasserlinie abzuschließen.

Basis für die Planung ist eine im Team vereinbarte Definition wann eine Aufgabe „fertig“ ist. Diese Definition umfasst immer, dass der zugehörige Code und entsprechende automatisierte Tests auf einem Buildserver eingecheckt sind und alle Tests laufen. Häufig findet man zusätzliche Qualitätssicherungsschritte, wie manuelle Tests, Codereviews und ähnliches.

5.1. Agiles Schätzen und Planen

Feature	Schätzung	Szenarien		
		Optimistisch 0%	Realistisch -15%	Pessimistisch -30%
Feature 1	1	R 1.0	R 1.0	R 1.0
Feature 2	5	R 1.0	R 1.0	R 1.0
Feature 3	3	R 1.0	R 1.0	R 1.0
Feature 4	8	R 1.0	R 1.0	R 1.0
Feature 5	5	R 1.1	R 1.1	R 1.1
Feature 6	13	R 1.1	R 1.1	R 1.2
Feature 7	5	R 1.2	R 1.2	R 1.2
Feature 8	3	R 1.2	R 1.2	R 1.2
Feature 9	1	R 1.2	R 1.2	R 1.3
Feature 10	8	R 1.2	R 1.3	R 2.0
Feature 11	5	R 1.2	R 1.3	R 2.1
Feature 12	3	R 1.3	R 1.3	unklar
Feature 13	3	R 1.3	R 2.0	unklar
Feature 14	8	R 1.3	R 2.0	unklar
Feature 15	8	R 1.3	R 2.1	unklar
Feature 16	13	R 2.0	unklar	unklar
Feature 17	5	R 2.0	unklar	unklar
Feature 18	1	R 2.0	unklar	unklar
Feature 19	3	R 2.1	unklar	unklar
Feature 20	5	R 2.1	unklar	unklar

Abbildung 5.1.: Beispiel für drei Szenarien für die langfristige Planung eines Backlogs

Langfristige Pläne

Da agile Projekte als wesentliche Eigenschaft auf unvorhersehbare Ereignisse reagieren können, macht eine langfristige Planung keinen Sinn. Stattdessen werden mehrere Szenarien definiert, die mögliche zukünftige Verläufe darstellen (siehe Abbildung 5.1).

Zentraler Parameter sind dabei Annahmen über die Veränderung der Velocity. Da mit umso mehr derzeit ungeplante Aufgaben und Aktivitäten gerechnet werden muss, je weiter man in die Zukunft plant, geht ein realistisches Szenario immer davon aus, dass die Velocity sinkt. Als optimistischste Annahme kann man davon ausgehen, dass keine ungeplanten Aufgaben mehr entstehen, die Velocity sich also nicht verändert. Soll das Teams vergrößert werden, kann sich auch das auf die langfristige Velocity auswirken, allerdings nicht durch linearen Zuwachs. Management, umliegende Projekte oder Verkauf können nun ein für die Entscheidung passendes Szenario wählen: Ist ein hohes Terminrisiko akzeptabel, so kann das optimistische Szenario zugrunde gelegt werden, bei geringer Risikotoleranz sollte auf dem pessimistischen Szenario aufgebaut werden.

Geschäftsmodelle

Ist die Velocity eines Teams bekannt und sind die Kosten bekannt, so können auf Basis dieser Daten weitere Geschäftsmodelle entwickelt werden, die über die üblichen zeit-schätzungsbasierten Festpreis- und Aufwandsmodelle hinaus gehen. So können Ange-

bote auf der Basis von Punkten erstellt werden, Punktebudgets verkauft werden oder Reaktionszeiten als Service Level verkauft werden. Durch statistische Auswertungen vergangener Iterationen lassen sich die jeweiligen betriebswirtschaftlichen Kosten für die jeweiligen Modelle bestimmen. Diese Kosten bilden dann gemeinsam mit den gewünschten Margen die Basis für die Preisfindung.

Literaturverzeichnis

1. Beck, K.: Planning Extreme Programming, Addison-Wesley, Reading, MA, 2001
2. Cockburn, A.: Crystal Clear - A Human-Powered Methodology for Small Teams, Addison-Wesley, Reading, MA, 2005
3. Cohn, M.: Agile Estimating and Planning, Prentice Hall, Boston, MA, 200
4. Fink, A.; Schalke, O; Siebe, A.: Erfolg durch Szenario- Management, Campus Verlag GmbH, Frankfurt/Main, 2002



Agiles Schätzen und Planen

Jens Coldewey (BDU)
Coldewey Consulting
Toni-Schmid-Str. 10 b
D-81825 München
Germany
Tel: +49-700-COLDEWEY
Tel: +49-700-26533939
Fax: +49-89-74995703
jens.coldewey@coldewey.com
<http://www.coldewey.com>

SEE 2009,
Berlin, 27. Mai 2009



- Ziele und Technik
- Adaptives Schätzen
- Langfristige Pläne
- Geschäftsmodell
- Diskussion



☞ Ziele und Technik

- Adaptives Schätzen
- Langfristige Pläne
- Geschäftsmodell
- Diskussion



Folie 3; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Ziele agiler Planung

Effektivität erhöhen:

- Fokussieren auf den Mehrwert
- Änderungen unterstützen
- Risiken minimieren

Effizienz erhöhen:

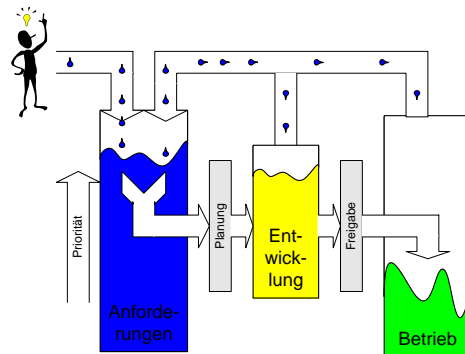
- Koordination von Aktivitäten
- Basis für realistische Zusagen nach außen
- Basis für realistische Statusbestimmung

- Nicht: Korrekte Vorhersage
- Nicht: Ressourcen-Belegungs-Plan
- Nicht: Druck ausüben!
- Nicht: Verantwortung weiter schieben!
- Nicht: Erzählen, was die Stakeholder hören wollen und dann „Chicken Race“ spielen!

Folie 4; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

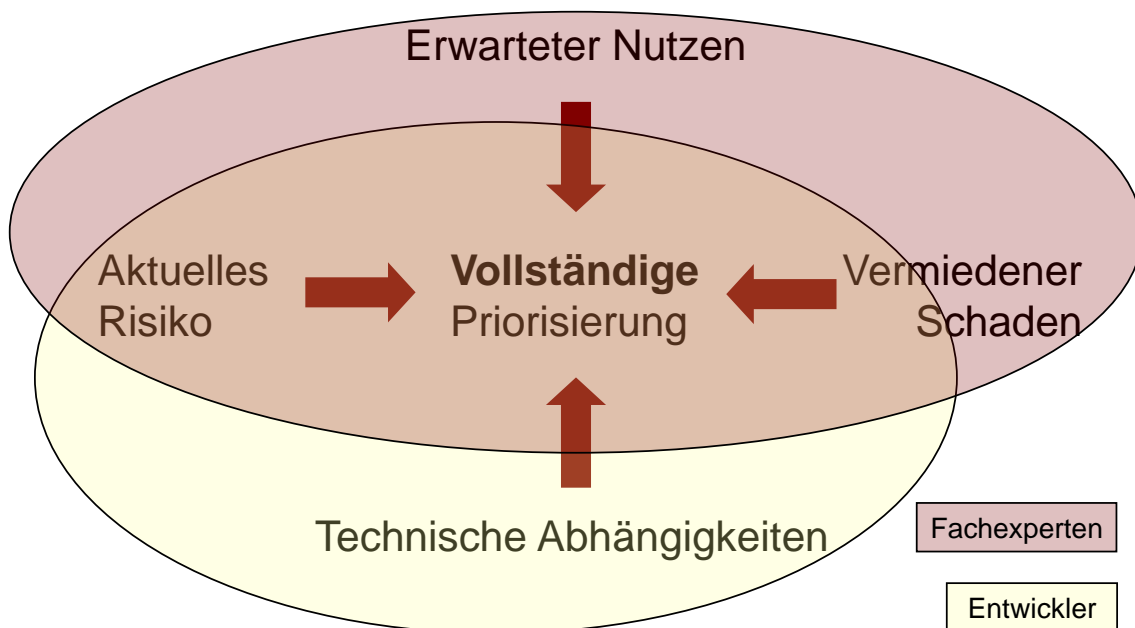
Die Basis: Iterative Planung

- Time-Boxing
- Regelmäßige Neuplanung
- Gemeinsame Planung
 - Priorisieren (Nutzen, Risiko, Abhängigkeiten)
 - Schätzen
 - Auswählen (Die Wasserlinie)



Folie 5; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Priorisieren



Folie 6; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Schätzen

- Die Entwickler schätzen
 - das Ergebnis gilt
- Fachexperten geben Auskunft
- Verfahren:
 - Expertenschätzung (Schätzpoker)
 - Etablierte Schätzverfahren
- Zu große Aufgaben werden zerschlagen



Folie 7; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Auswählen

- Der Umfang einer Iteration ist bekannt
 - > Wasserlinie („Yesterday's Weather“)
- Es werden so lange von oben Aufgaben genommen, bis die Wasserlinie erreicht ist
 - > Umfang dieser Iteration
- Unter der Wasserlinie werden auch noch Karten genommen
 - > Puffer
- ☞ **Wahrscheinlichkeit für Planerreichung: 50%!**



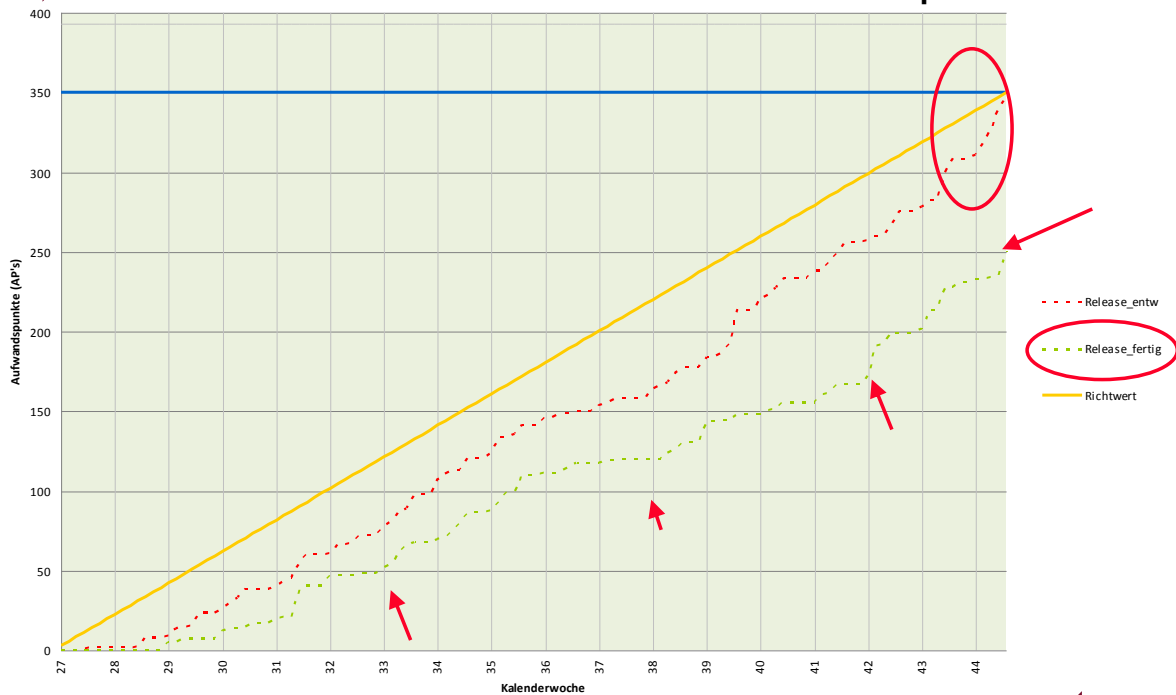
Folie 8; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Fortschrittsanzeige an der Pinwand



Folie 9; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Das Burnup Chart



Folie 10; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

- Ziele und Technik
- **Adaptives Schätzen**
- Langfristige Pläne
- Geschäftsmodell
- Diskussion



Folie 11; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Oft wird (zu) viel Aufwand in die „Kalibrierung“
der Schätzung gesteckt

Schätzung

Realität

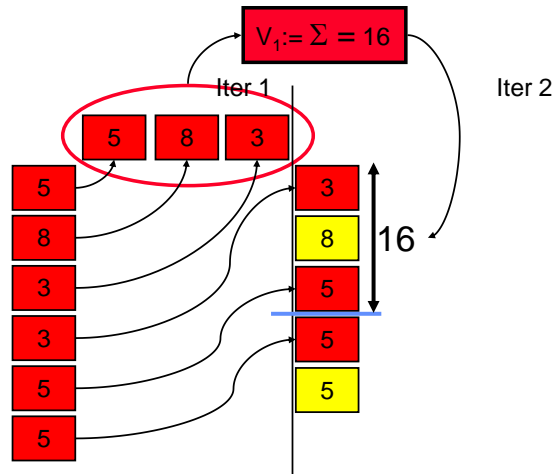
2 Tage

uninteressant

Two sticky notes are shown. The top note is yellow and has the following text: '3P050', '01/180:', 'Error in Commission after invoice', and '1039'. A red circle is drawn around the number '1' at the bottom right. A red arrow points from this '1' to the text '2 Tage'. The bottom note is also yellow and has the following text: '3P050', 'FORDC WEN INBETRIEB VON LUKKX', 'PMSI 20100', and 'MS'. A red circle is drawn around the number '1' at the bottom right, which is crossed out with a red 'X'. A red arrow points from this '1' to the text 'uninteressant'.

Folie 12; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

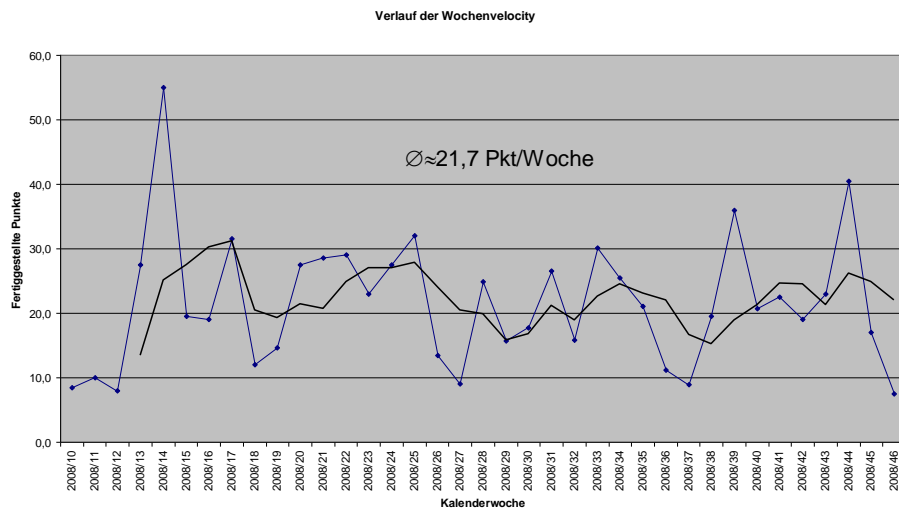
Die Velocity dient als Planungsgrundlage für die nächste Iteration



Folie 13; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

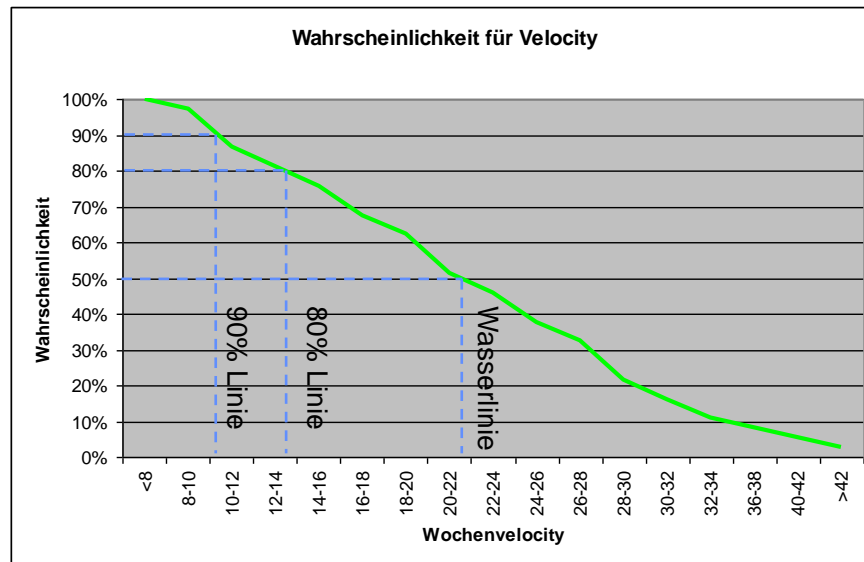
Interessant ist die Velocity der letzten Perioden

Velocity = abgeschlossene Punkte/Periode



Folie 14; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Ein wenig Statistik erlaubt differenzierte Aussagen



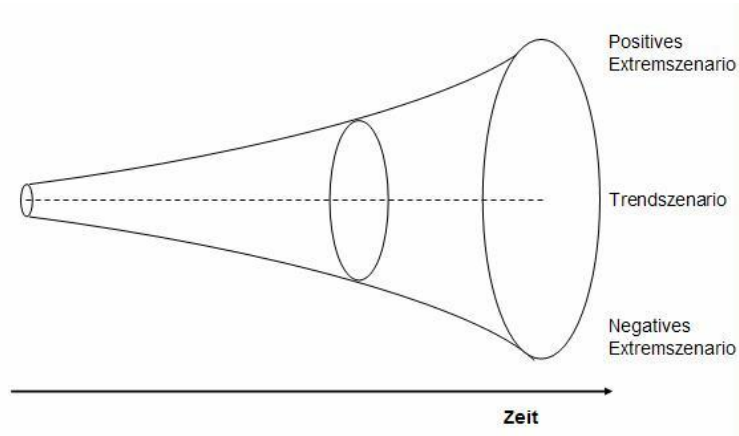
Folie 15; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

- Ziele und Technik
- Adaptives Schätzen
- ☞ **Langfristige Pläne**
- Geschäftsmodell
- Diskussion



Folie 16; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

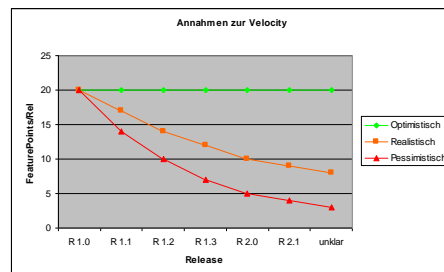
Langfristplanung: Je weiter, um so unsicherer



Folie 17; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Übertragen auf das Backlog ergeben sich Lieferszenarien

Feature	Schätzung	Optimistisch	Realistisch	Pessimistisch
		0%	-15%	-30%
Feature 1	1 R 1.0	R 1.0	R 1.0	R 1.0
Feature 2	5 R 1.0	R 1.0	R 1.0	R 1.0
Feature 3	3 R 1.0	R 1.0	R 1.0	R 1.0
Feature 4	8 R 1.0	R 1.0	R 1.0	R 1.0
Feature 5	5 R 1.1	R 1.1	R 1.1	R 1.1
Feature 6	13 R 1.1	R 1.1	R 1.2	R 1.2
Feature 7	5 R 1.2	R 1.2	R 1.2	R 1.2
Feature 8	3 R 1.2	R 1.2	R 1.2	R 1.2
Feature 9	1 R 1.2	R 1.2	R 1.3	R 1.3
Feature 10	8 R 1.2	R 1.3	R 2.0	R 2.0
Feature 11	5 R 1.2	R 1.3	R 2.1	R 2.1
Feature 12	3 R 1.3	R 1.3	unklar	unklar
Feature 13	3 R 1.3	R 2.0	unklar	unklar
Feature 14	8 R 1.3	R 2.0	unklar	unklar
Feature 15	8 R 1.3	R 2.1	unklar	unklar
Feature 16	13 R 2.0	unklar	unklar	unklar
Feature 17	5 R 2.0	unklar	unklar	unklar
Feature 18	1 R 2.0	unklar	unklar	unklar
Feature 19	3 R 2.1	unklar	unklar	unklar
Feature 20	5 R 2.1	unklar	unklar	unklar



Folie 18; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

- Ziele und Technik
- Adaptives Schätzen
- Langfristige Pläne
- ☞ **Geschäftsmodell**
- Diskussion



Folie 19; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Von der Planung zum Geschäftsmodell

Ansatzpunkte:

- Punkte verkaufen statt Personentagen
⇒ Festpreis mit spezifiziertem Risiko
- Garantierte Reaktionszeiten auf Basis der Wochenvelocity
⇒ Servicelevel nach „Buchungsklasse“
- Punktbudgets verkaufen

- Voraussetzungen:
- Eingeschwungenes Schätzverfahren
- Brauchbare statistische Grundlage
- Planungsdisziplin
- „Fertig“ steht für guten Qualitätslevel
- Vertrauen des Kunden in das Schätzverfahren
- Mittel- bis langfristiges Management

Folie 20; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Danke an



Bewotec, Rösrath
Aladdin, Gräfelfing
Johannes Link
Bernd Schiffer
Michael Becher
Michael Hüttermann
Jim Highsmith

Folie 21; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

- Ziele und Technik
- Adaptives Schätzen
- Langfristige Pläne
- Geschäftsmodell
- 👉 Diskussion

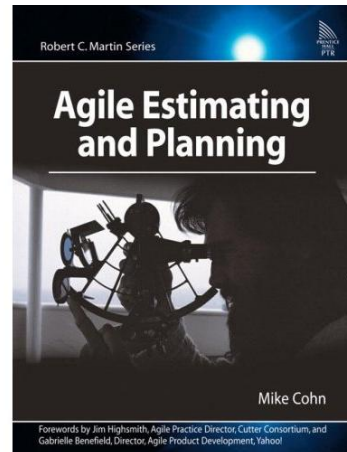


© mills21 - Fotolia.com

Folie 22; © Copyright Jens Coldewey, Coldewey Consulting, All Rights Reserved

Literatur

Mike Cohn
Agile Estimating and
Planning
Prentice Hall 2006
ISBN 0-13-147941-5



5.2. Planungsebenen der iterativen Systementwicklung

Michael Schulze-Ruhfus
oose Innovative Informatik GmbH
Tower Falkenried-Piazza, Straßenbahnring 7
D-20251 Hamburg
michael.schulze-ruhfus@oose.de

Abstract

Ein iteratives Vorgehen wird immer häufiger für komplexe Systementwicklungsprojekte gefordert. Dabei gilt weiterhin, dass jede an einer Systementwicklung beteiligte Disziplin wie Hardware, Software oder Mechanik ihre eigenen Rahmenbedingungen und Entwicklungstechniken hat. Für ein Systementwicklungsprojekt besteht die Herausforderung, die Eigenständigkeit der Disziplinen zuzulassen und dennoch ein stimmiges Gesamtbild zu schaffen. Hierfür sind geeignete Synchronisationspunkte, Schnittstellen und Abhängigkeiten zwischen den Disziplinen notwendig. Gerade diese Aspekte werden durch ein iteratives Vorgehen in den Vordergrund gerückt. In diesem Vortrag wird der iterative Ansatz für die Systementwicklung aus dem Blickwinkel der dazu erforderlichen Planungsebenen betrachtet.

Fragestellungen

Wie wird in der Planung mit verschiedenen Abstraktionsebenen und Planungshorizonten gearbeitet? Welche Planungsebenen werden zu welchen Zeitpunkten von wem erarbeitet? Welche Arten von Features als Planungseinheiten sind sinnvoll? Welche Varianten gibt es, diese inhaltlich zu schneiden? Welche Auswirkungen hat die Projektorganisation auf den Schnitt der Features? Wie wird die Integration der verschiedenen Segmente und Einheiten in der Planung berücksichtigt? Welche Rollen in einem Systementwicklungsprojekt sind für die Planung verantwortlich, welche Abstimmungsprozesse sind sinnvoll? Was hat ein iteratives Vorgehen mit Risikomanagement zu tun? In wie weit ist Netzplantechnik ein geeignetes Mittel zur Planung einer iterativen Systementwicklung? Was bedeutet ein solches Vorgehen für frühe Meilensteine, die als Liefergegenstand Spezifikationen fordern? Wie unterscheiden sich die Planungsebenen der iterativen Systementwicklung von denen der agilen Softwareentwicklung in Großprojekten?

Lösungsansatz

Alle Versuche, Entwicklungsprojekte zu Beginn vollständig durchzuplanen, haben sich als wenig erfolgreich erwiesen. Projekte mit unklaren Anforderungen, Anforderungsänderungen und zunehmendem Erkenntnisgewinn während der Projektlaufzeit sind mit einem iterativen Vorgehen deutlich erfolgreicher abzuwickeln. Dies bedeutet u.a., dass risikobehaftete Aufgaben möglichst früh angegangen werden, und man sich bei der Projektplanung von Beginn an - und nicht erst kurz vor einem dann nicht mehr erreichbaren Meilenstein - fragt, welche fachlichen Funktionalitäten der Auftraggeber unbedingt braucht.

Häufig wird einer agilen Vorgehensweise unterstellt, dass nur kurzfristig geplant werde und keine mittel- und langfristigen Prognosen über die Projektergebnisse abgegeben werden. Dies ist jedoch nicht korrekt. Sowohl bei agiler Softwareentwicklung als auch bei iterativer (agiler) Systementwicklung sollten neben der kurzfristigen Planung der direkt folgenden Iteration auch auf höheren Abstraktionsebenen Pläne erstellt und gepflegt werden, die dann je nach Abstraktionsebene auch entsprechend weit in die Zukunft reichen. Es wird allerdings vermieden, weit in der Zukunft liegende Aktivitäten detailliert auszuplanen, da bei der Kombination „ferne Zukunft“ und „Detailplanung“ die Wahrscheinlichkeit, dass dieser Plan Bestand haben wird, äußerst gering ist. Abbildung 1 stellt die Planungsebenen der iterativen Systementwicklung für ein Projekt mit System- und Einheitenebene schematisch dar. In der Abbildung sind neben den Planungsebenen auch die Wege dargestellt, die zu Planänderungen auf den jeweiligen Ebenen führen.

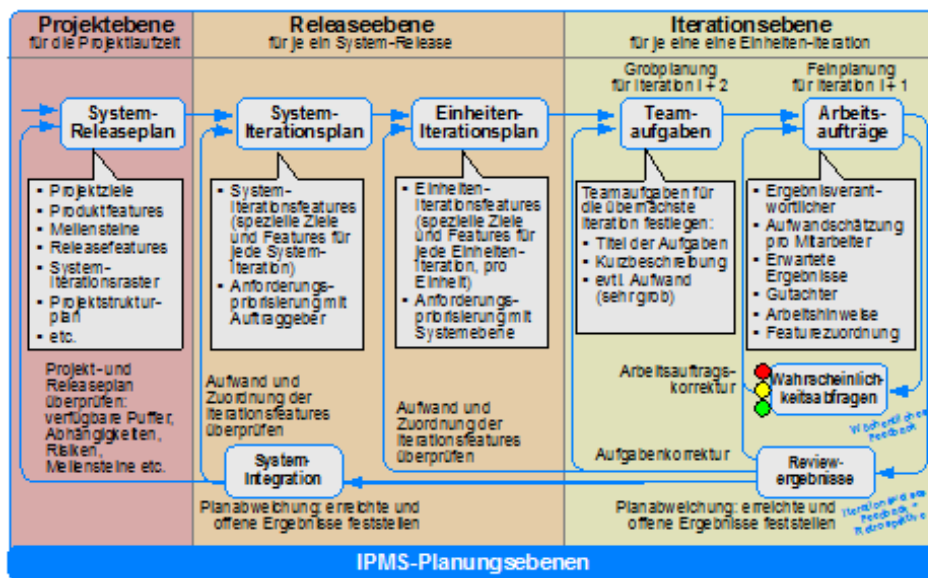


Abbildung 5.2.: Planungsebenen-Übersicht für ein Systementwicklungsprojekt ohne Segmente

Im iterativen Vorgehen wird sehr intensiv und stark am Inhalt orientiert geplant. Wichtig ist dabei, die verschiedenen Abstraktionsebenen auseinander zu halten. Aufgrund der regelmäßigen Messpunkte für den Projektfortschritt, den Iterationsenden, wird dem Projektmanagement ermöglicht, Probleme frühzeitig zu erkennen und entsprechend zu reagieren. Projektstillstand kann praktisch nicht verheimlicht werden, da vorgesehen ist, regelmäßig lauffähige Arbeitsergebnisse vorzuführen. Geplant wird in Form von Features und Arbeitsaufträgen. Netzplantechnik ist auf den unteren Planungsebenen nicht sinnvoll, da Planänderungen hier vor allem die Inhalte der Aktivitäten betreffen und die Termine und Dauern über Timeboxen gehandhabt werden. Für die Ausplanung der Systemintegration kann Netzplantechnik angebracht sein, wenn viele Abhängigkeiten von externen Zulieferleistungen vorhanden sind.

Zusammenfassung

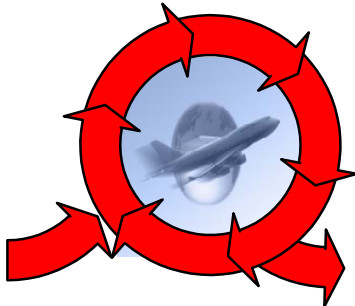
Dynamische Projekte mit unklaren Anforderungen, Anforderungsänderungen und zunehmendem Erkenntnisgewinn während der Projektlaufzeit sind mit einem iterativen Vorgehen deutlich erfolgreicher abzuwickeln, das heißt kostengünstiger, schneller und

5.2. Planungsebenen der iterativen Systementwicklung

mit besserer Qualität als mit einem klassischen, wasserfallartigen Entwicklungsvorgehen. Hierbei muss jedoch beachtet werden, dass eine Planung nur für die direkt folgende Iteration nicht ausreicht, sondern mit geringerer Detailtiefe, aber längeren Planungshorizonten weitere Planungsebenen erforderlich sind. Diese orientieren sich an der Systemstruktur. Jede Planungsebene steht dabei für eine weitere Stufe der Systemintegration. Die oberste Planungsebene bildet die mit dem Auftraggeber des Systementwicklungsprojektes vereinbarten Lieferzeitpunkte der Ausbaustufen des Systems ab.

Literaturverzeichnis

1. Oestereich, Bernd; Weiss, Christian: APM - Agiles Projektmanagement. dpunkt.verlag, Heidelberg, 2008
2. Schulze-Ruhfus, Michael; Weilkiens, Tim: Iteratives Projektmanagement für Systeme. In (GfSE Hrsg.): Proc. Tag des Systems Engineering 2008



Planungsebenen der iterativen Systementwicklung

Software & Systems Engineering Essentials
25.-27. Mai 2009, Berlin

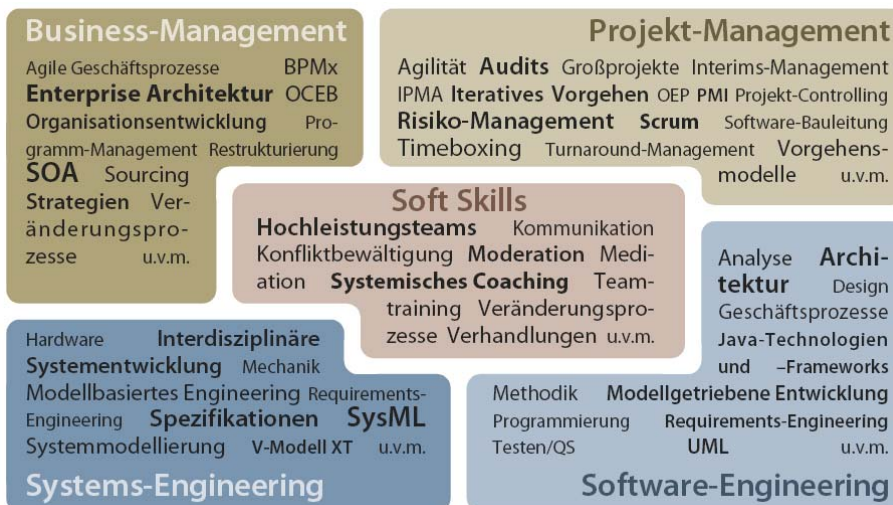
Michael Schulze-Ruhfus

(Bereichsleiter Projektmanagement)
michael.schulze-ruhfus@oose.de



© 2009 by oose GmbH

oose Innovative Informatik GmbH – Beratung. Seminare. Projektarbeit.



© 2009 by oose GmbH

Agenda

- I. **Motivation und Einführung**
- II. Iteratives Vorgehen in der Systementwicklung
- III. Planungsebenen der iterativen Systementwicklung
- IV. Zusammenfassung und Ausblick

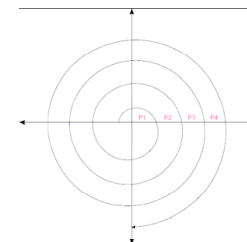


© 2009 by oose GmbH

Alle reden von iterativer Systementwicklung...

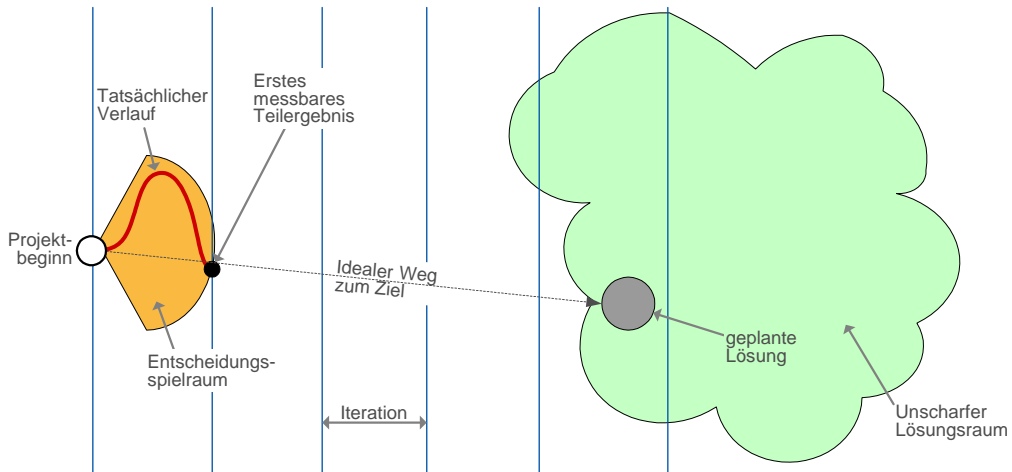
- *Systems engineering is an iterative process of top-down synthesis, development, and operation of a real-world system... (H. Eisner)*
- *The systems engineering process has an iterative nature that supports learning and continuous improvement. (Systems Engineering Handbook v3.1)*
- *The highly related nature of the products necessitate that they be developed in an iterative manner as greater understanding is achieved via the work process. It is not the intent of this section to suggest that a given product is developed and finalized before the next product is addressed. (DoDAF)*

Was bedeutet eigentlich „iterative Systementwicklung“?
Was ist eine Iteration in der Systementwicklung?
Werden alle Systemkomponenten im gleichen Iterationstakt entwickelt?
Wie wird eine iterative Systementwicklung geplant?



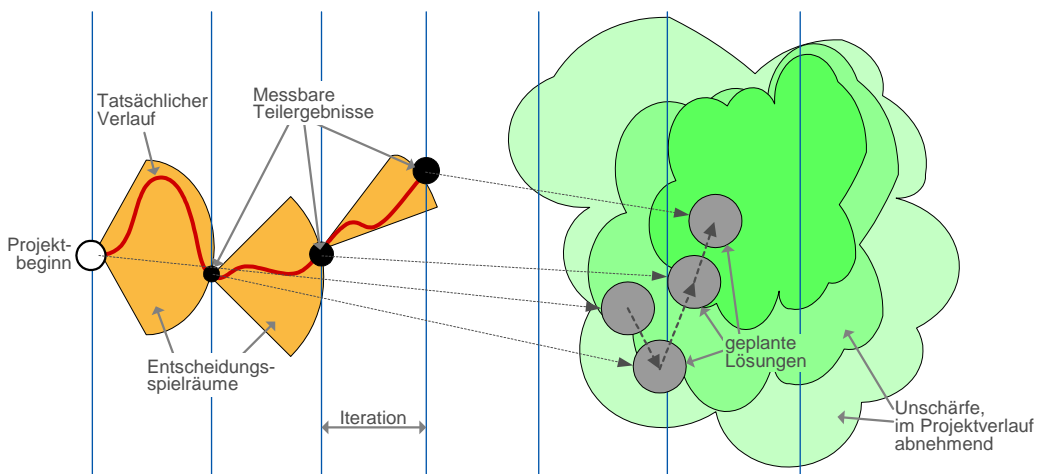
© 2009 by oose GmbH

Agilität durch iteratives Vorgehen



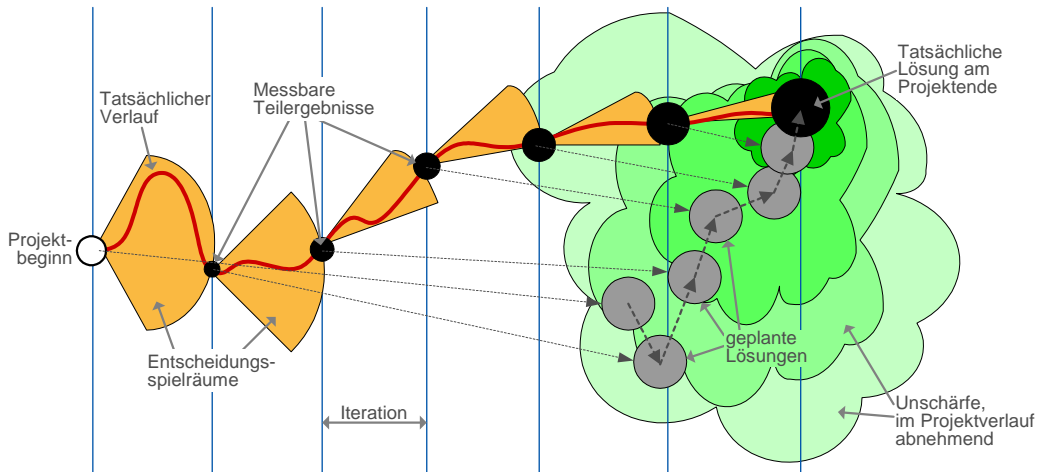
© 2009 by oose GmbH

Agilität durch iteratives Vorgehen



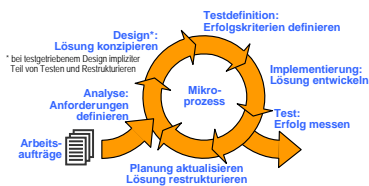
© 2009 by oose GmbH

Agilität durch iteratives Vorgehen



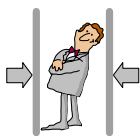
© 2009 by oose GmbH

Grundbegriffe



Eine **Iteration** ist ein in ähnlicher Weise wiederholter Zeitabschnitt innerhalb eines Entwicklungsprozesses. Der Gesamtprozess wird in eine Vielzahl kürzerer Zeitabschnitte untergliedert, in denen prinzipiell die gleichen elementaren Entwicklungsaktivitäten stattfinden.

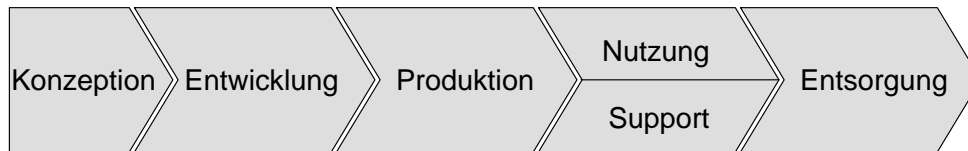
Ein **Meilenstein** definiert einen Termin, zu dem eine Menge von Ergebnissen nachprüfbar vorliegen muß. Liegen die Ergebnisse zum geplanten Termin nicht vor, wird der Meilenstermin verschoben.



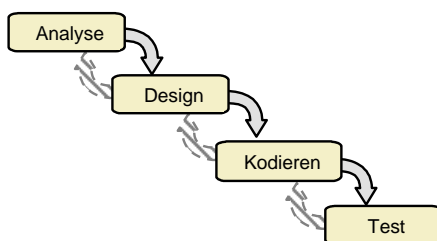
Eine **Timebox** definiert einen unverrückbaren Zeitrahmen, an dessen Ende eine Menge von Ergebnissen nachprüfbar vorliegen soll. Liegen die Ergebnisse nicht wie geplant vor, werden die offenen Teile in eine nachfolgende Timebox verschoben.

© 2009 by oose GmbH

Missverständnisse vermeiden – Das sind KEINE Iterationen



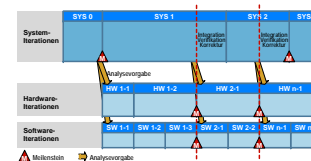
System-Lbensphasen



Wasserfallmodell der Softwareentwicklung

© 2009 by oose GmbH

Vorteile iterativer Entwicklung (Auswahl)



- Sicherheit bzgl. tatsächlichem Projektstatus und inhaltlicher Zielerreichung
 - Regelmäßige Messpunkte
 - Sichtbarer Projektfortschritt

- Flexibilität für Anforderungsänderungen und Reaktion auf Erkenntnisgewinn
 - Regelmäßige Entscheidungs- und Priorisierungspunkte
 - Abgestufte Planungsebenen

- Risikominderung durch frühe und häufige Systemintegrationen

© 2009 by oose GmbH

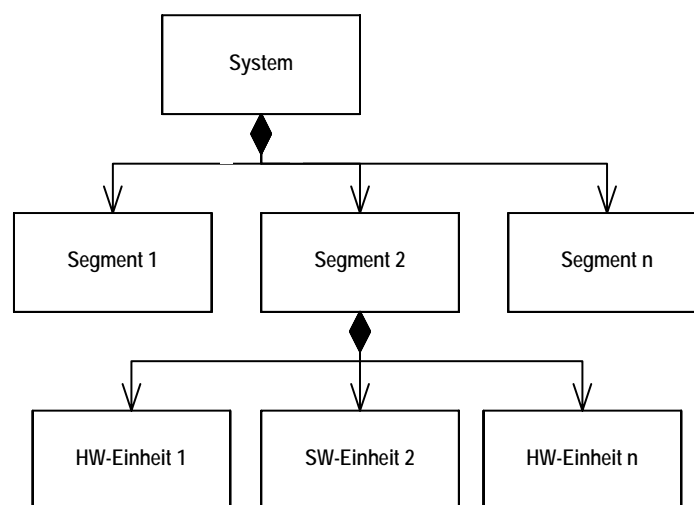
Agenda

- I. Motivation und Einführung
- II. **Iteratives Vorgehen in der Systementwicklung**
- III. Planungsebenen der iterativen Systementwicklung
- IV. Zusammenfassung und Ausblick



© 2009 by oose GmbH

Definitionen Systemkomponenten

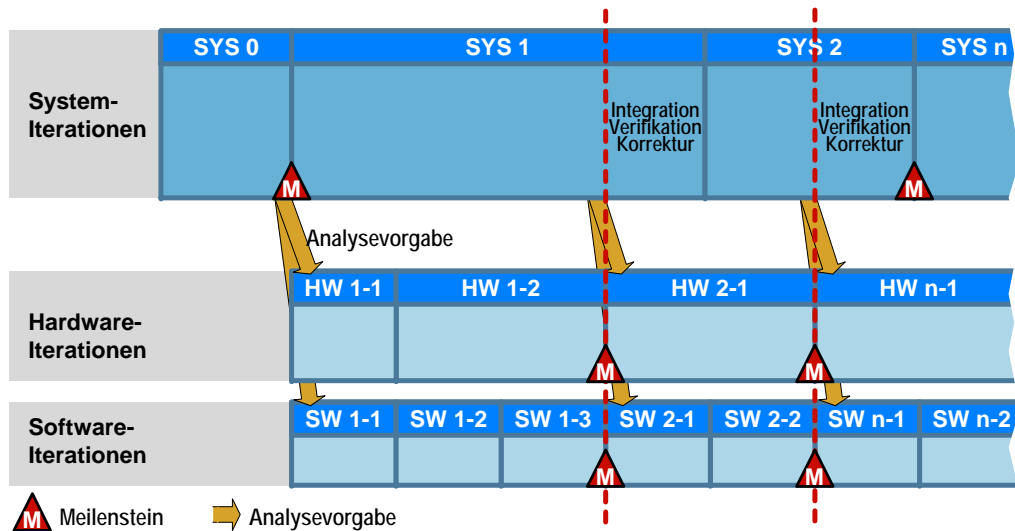


Quelle: V-Modell[®] XT, Version 1.2.1

© 2009 by oose GmbH

Planungsebenen der iterativen Systementwicklung

Beispiel-Zeitstrahl für ein System mit einer HW- und einer SW-Einheit



© 2009 by oose GmbH

Planungsebenen der iterativen Systementwicklung

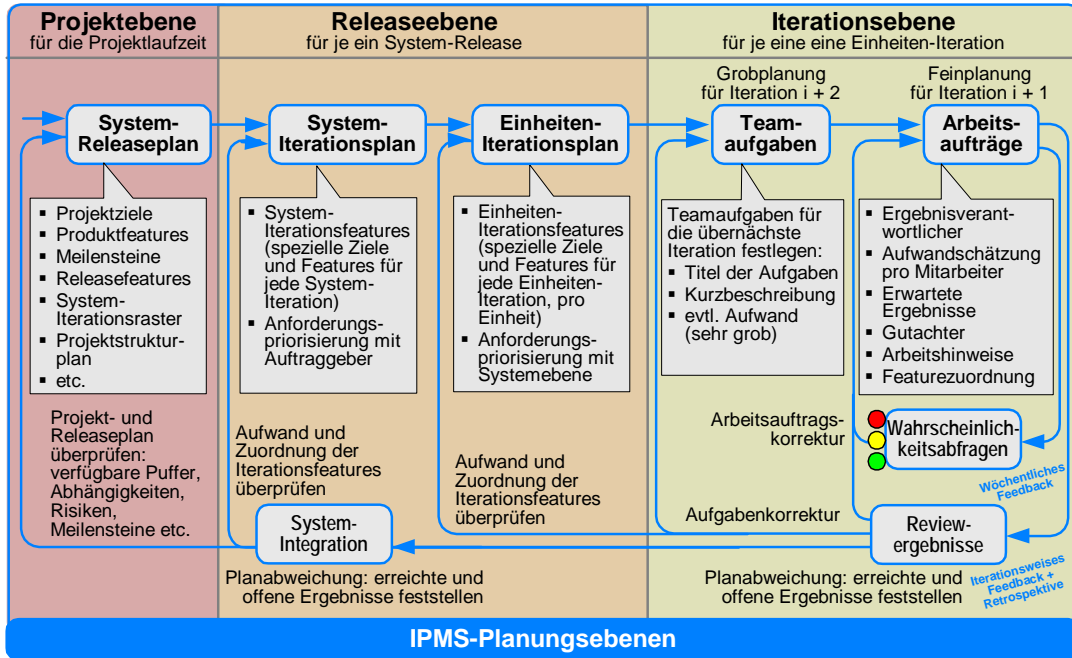
Agenda

- I. Motivation und Einführung
- II. Iteratives Vorgehen in der Systementwicklung
- III. **Planungsebenen der iterativen Systementwicklung**
- IV. Zusammenfassung und Ausblick



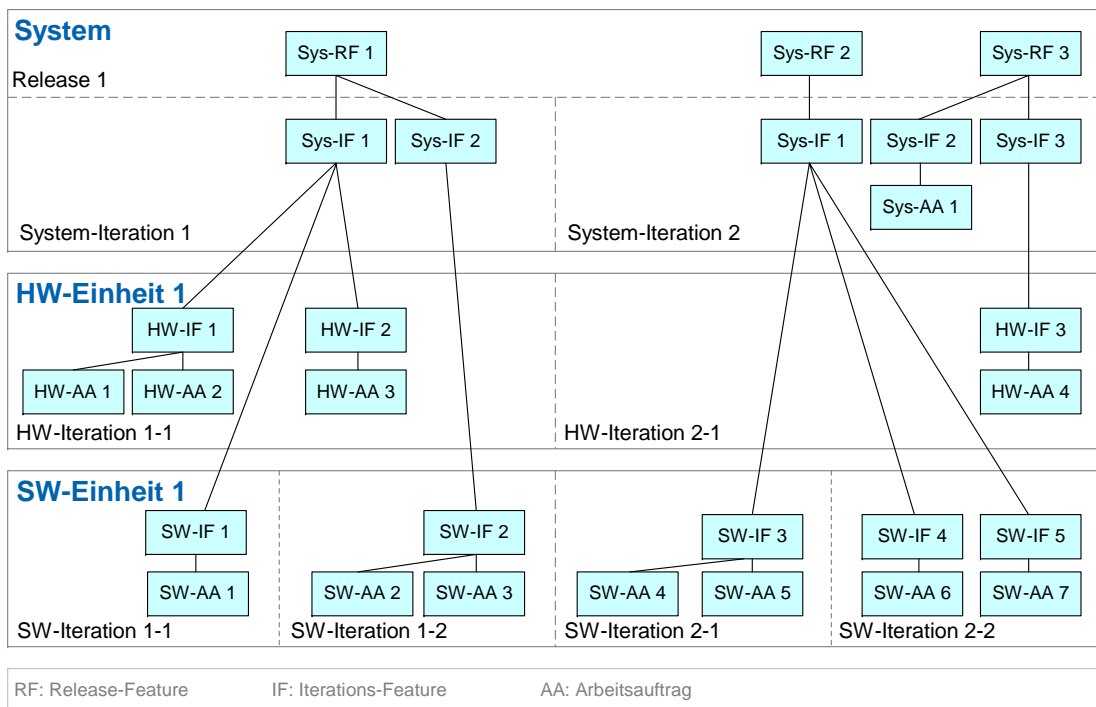
© 2009 by oose GmbH

Planungs-, Steuerungs- und Controlling-Dokumente für die Makro- und Mikroebene



© 2009 by oose GmbH

Beispiel für eine Featurestruktur



© 2009 by oose GmbH

Iterative Systementwicklung und klassisches Projektmanagement

- In wie weit ist Netzplantechnik ein geeignetes Mittel zur Planung einer iterativen Systementwicklung?
- Was bedeutet ein solches Vorgehen für frühe Meilensteine, die als Liefergegenstand Spezifikationen fordern?
- Was hat ein iteratives Vorgehen mit Risikomanagement zu tun?
- Welche Rollen in einem Systementwicklungsprojekt sind für die Planung verantwortlich, welche Abstimmungsprozesse sind sinnvoll?

© 2009 by oose GmbH

Agenda

- I. Motivation und Einführung
- II. Iteratives Vorgehen in der Systementwicklung
- III. Planungsebenen der iterativen Systementwicklung
- IV. **Zusammenfassung und Ausblick**



© 2009 by oose GmbH

Zusammenfassung und Ausblick

- Erste praktische Umsetzungen iterativer Systementwicklung zeigen die gewünschten Effekte.
 - Die Systemstruktur bestimmt die erforderlichen Planungsebenen.
 - Diese haben unterschiedliche Planungshorizonte und eine entsprechend unterschiedliche Detailtiefe.
 - Jede Planungsebene steht für eine Stufe der Systemintegration.
-
- Die Anwendung des „agilen Mindsets“ und ausgewählter agiler Praktiken bringt weitere Vorteile.
 - Wir erwarten, dass sich iterative Systementwicklung weiter durchsetzen wird
 - aufgrund der zunehmenden Komplexität der Systeme,
 - aufgrund zunehmendem Softwareanteil in Systemen und
 - unterstützt durch MBSE (Model Based Systems Engineering).

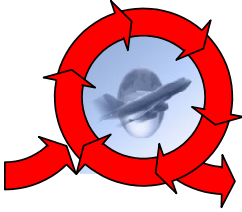
© 2009 by oose GmbH

Fragen und Diskussion

- Entwickeln Sie iterativ Systeme?
- Auf welche Art planen Sie Ihre iterative Systementwicklung?
- Glauben Sie, dass iterative Systementwicklung Herausforderungen wie zunehmende Komplexität, kurze Time-To-Market-Zeiten, globale Entwicklung lösen wird?



© 2009 by oose GmbH



Herzlichen Dank!

oose.

Wissen aus erster Hand.

Michael Schulze-Ruhfus

(Bereichsleiter Projektmanagement)
michael.schulze-ruhfus@oose.de



5.3. Pragmatische Aufwandsschätzung

Holger Bohlmann, Andreas Havenstein, Henning Wolf

akquinet it-agile GmbH

Paul-Stritter-Weg 5

22297 Hamburg

{holger.bohlmann | andreas.havenstein | henning.wolf}@akquinet.de

Abstract

Eine realistische Aufwandsschätzung ist eine wichtige Grundvoraussetzung für ein Projekt. In diesem Bereich sind heute Expertenschätzungen üblich. Sie sind aber leider oft sehr ungenau und führen dann zu unrealistischen Planungen. In diesem Vortrag zeigen wir, welche Erfahrungen wir mit einer einfachen Herangehensweise mit abstrakten Schätzmaßen, der hierarchischen Dekomposition und Featurelisten in agilen Projekten gesammelt haben. Wir zeigen, wie mit alternativen Mitteln das gesamte Team bei der Schätzung eingebunden werden kann. Zusätzlich gibt es einen knappen Abriss über klassische Schätzverfahren wie Cocomo und Function Points, die zumindest eine Liste von relevanten Faktoren für die Entwicklungsgeschwindigkeit beisteuern können.

Ausgangssituation

Zu jedem Softwareprojekt gehört auch eine Aufwandsschätzung, die vor dem Start des Projektes erfolgen muss. Sie ist die Grundlage für jede Zeit- und Preisberechnung. Für diese Schätzung werden die Anforderungen betrachtet. Je genauer diese bekannt sind und je genauer man weiß, wie man diese realisieren kann, desto genauer wird die Schätzung [1]. Die Aufwandsschätzung wird sehr häufig als sogenannte „Expertenschätzung“ durchgeführt. Anforderungen werden mehr oder minder genau zerlegt, wobei es oft dazu kommen kann, dass Details übersehen werden. Für diese Vorarbeit wird unterschiedlich viel Zeit investiert. Die Güte der Schätzung unterliegt der Geschicklichkeit des Fachexperten. Das spätere Entwicklungsteam, zu dem der Fachexperte gehören kann oder nicht, wird für die Realisierung durchaus unterschiedlich viel Zeit brauchen. Dieses „Delta“ ist leider im negativen Sinne oft sehr hoch.

Möglichkeiten der Aufwandsschätzung

Um die Komplexität von Anforderungen bestimmen zu können gibt es die Möglichkeit einer „Function-Point-Analyse“. Auch wenn es damit gelingt, einen Wert von Komplexität zu ermitteln, muss dieser weiter in Personentage und damit in Kosten umgerechnet werden. Eine Lösung bietet dazu das CoCoMo II-Verfahren, welches eine Umrechnungsvorschrift bietet. Hierzu werden zahlreiche Größen aus dem Projektumfeld (Teamzusammensetzung, Fachwissen) ermittelt, die in der Berechnung Beachtung finden. Beide Verfahren haben sich in der Praxis nicht gegen Expertenschätzungen durchsetzen können.

Eine agile Herangehensweise an die Aufwandsschätzung ist es, den eigentlichen Realisierungs-Fachexperten schlechthin einzusetzen: Das Entwicklungsteam. Durch eine systematische Aufspaltung der Anforderungen in zu entwickelnde Features erlangt das Team eine genaue Einschätzung der Komplexität. Eine feste Teamzusammensetzung und

ein Rückblick auf vergangene Projekte gibt dem Team eine gute Basis, um den Realisierungsaufwand von Features abschätzen zu können. Insgesamt ist damit das Abschätzen des Aufwandes zwar zeitintensiver, jedoch wird eine bessere Güte erreicht. Die Technik der hierarchischen Dekomposition ermöglicht es, im Schätzprozess früh einen groben Wert zu ermitteln. Da das Entwicklungsteam selbst den Aufwand benennt, kann es ein verbindliches Commitment abgeben.

Literaturverzeichnis

1. Boehm, Barry: Software Engineering Economics, 1981, Prentice Hall
2. Cohn, Mike: Agile Estimating an Planning, 2006, Prentice Hall
3. Poensgen, Benjamin: Die Function-Point-Analyse: Ein Praxishandbuch



Pragmatische Aufwandsschätzung

Dipl.-Informatiker Holger Bohlmann
Dipl.-Informatiker Andreas Havenstein
Dipl.-Informatiker Henning Wolf



Agenda



-
- Einleitung
 - Story 1: Hochrechnung
 - Story 2: Teamschätzung
 - Story 3: Unklare Anforderungen
 - Story 4: Relativ schätzen
 - Story 5: Abstrakte Schätzmaße
 - Story 6: Schätzfreundliche Anforderungen
 - Abschluss



Einleitung



Winston Churchill:

„Prognosen sind schwierig, insbesondere wenn sie die Zukunft betreffen.“

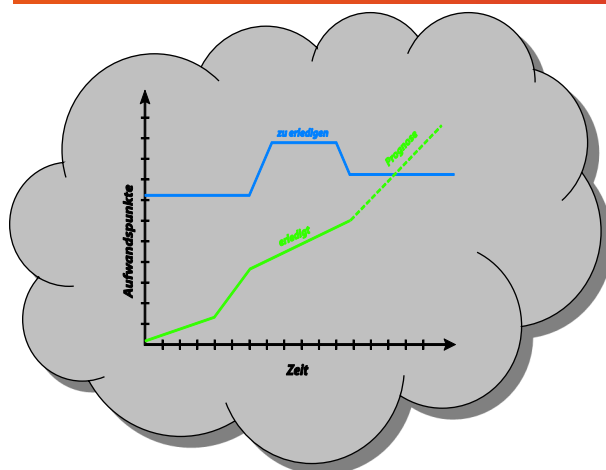


Das stimmt, wir müssen trotzdem Prognosen/Schätzungen machen:

- Wir können aber lernen, besser rückzukoppeln
- Wir können besser erkennen, welche Faktoren falsch oder richtig waren



1. Story: Die Hochrechnung



Geschwindigkeit unbekannt ...

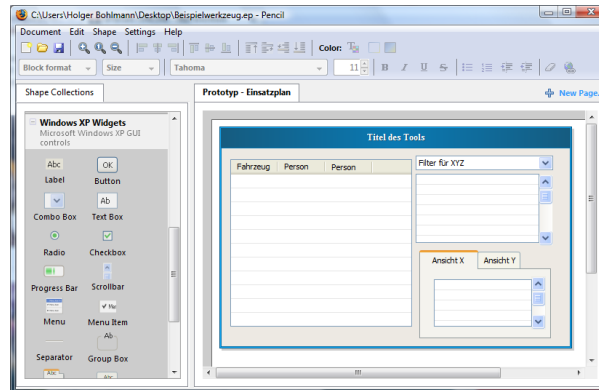
... einfach mal anfangen und messen!



1. Story: Hochrechnung mit Prototyp?



Sehr hilfreich zur Klärung von Anforderungen, aber ...



- Prototyp \neq Fertiges System
- Entwicklungsgeschwindigkeit ist unterschiedlich zur späteren Entwicklung



2. Story: Team-Schätzung



Binsenweisheit: Je mehr Leute schätzen desto genauer wird das Ergebnis, Beispiel:

- 40 Personen schätzen Anzahl Gummibären
- Ergebnis: 300 – 3000 wird geschätzt
- Mittelwert entspricht der Anzahl $\pm 10\%$

Investition in die Schätzung lohnt sich!



2. Story: Team-Schätzung



- Die es entwickeln, sollen es auch schätzen
- Commitment vom Entwicklungsteam
- Der Entwickler ist der ‚Experte‘, er/sie muss es realisieren
- Schätz-Expertise sollen Entwickler haben, kann man auch lernen. Bekannte agile Technik:



... oder bekannter ...



3. Story: Unklare Anforderung



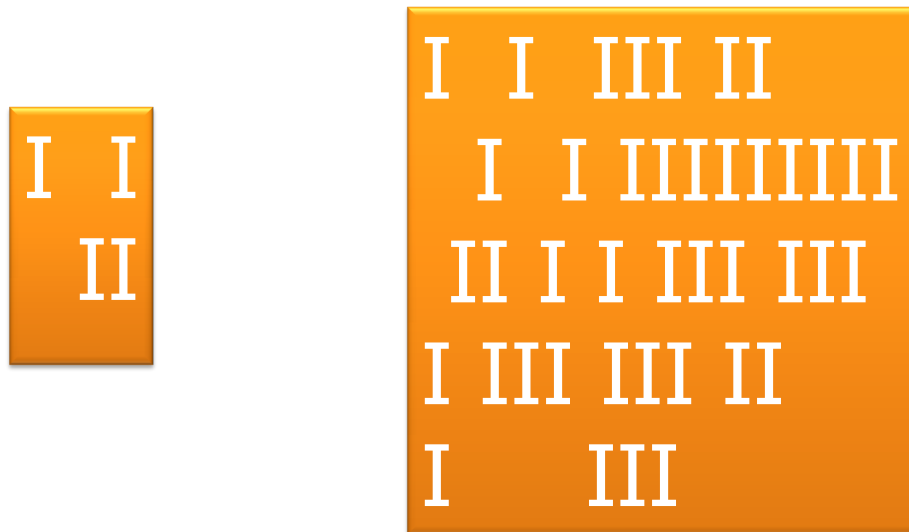
- Unklare Anforderungen ergeben unklare Schätzungen!
- Deshalb: Nicht würfeln, sondern klären.



- Schätzfreundliche Anforderungen können dabei helfen (siehe Story 6)
- Leider schwer zu bekommen:
Der fachliche Ansprechpartner



4. Story: Relativ-Schätzen



4. Story: Relativ-Schätzen Grenzen Absoluter Zahlenbestimmung



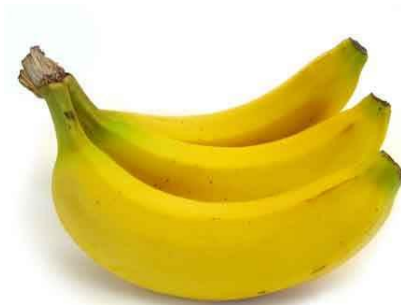
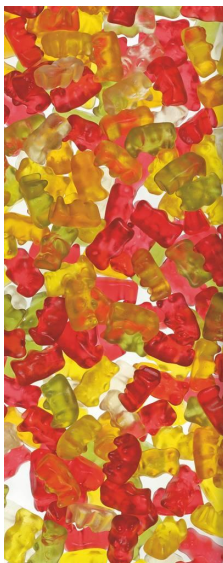
- Frühe römische Antike:
I, II, III, IIII, V, VI, VII, VIII
- Maya in Mittelamerika:
*, **, ***, ****, I, *I, **I, ***I.
- Menschen sind schlecht im Schätzen konkreter Größenangaben.



4. Story: Relativ-Schätzen Ma Dalton will einen Pullover stricken...



5. Story: Abstrakte Schätzmaße



S = 72cm
M = 74cm
L = 76cm
XL = 78cm
XXL = 80cm



S = 46cm
M = 51cm
L = 56cm
XL = 61cm
XXL = 66cm

ADULT - STRAIGHT



5. Story: Abstrakte Schätzmaße - Functions Points



- Größenmaß zur Beschreibung des funktionalen Umfangs
- Perspektivenwechsel: Anwendersicht
- Geschätzt werden:
 - Elementarprozesse
 - Eingabe
 - Ausgabe
 - Abfrage
 - Daten
 - Interne Datenbestände
 - Externe Datenbestände
- Funktionaler Gesamtumfang:

$$\text{uFP} = \sum \text{Prozesspunkte} + \sum \text{Datenbestandspunkte}$$



5. Story: Functions Points - Elementarprozesse



Eingabe		Anzahl Datenelemente		
		1-4	5-15	>= 16
Anzahl Datenbestände	0-1	3	3	3
	2	3	4	6
	>=3	4	6	6

Ausgabe		Anzahl Datenelemente			Abfrage		Anzahl Datenelemente		
		1-5	6-19	>= 20			1-5	6-19	>= 20
Anzahl Datenbestände	0-1	4	4	5	Anzahl Datenbestände	0-1	3	3	4
	2-3	4	5	7		2-3	3	4	6
	>= 4	5	7	7		>= 4	4	6	6



5. Story: Functions Points - Datenbestände



Interner Datenbestand		Anzahl Datenelemente			Externer Datenbestand		Anzahl Datenelemente		
		1-19	20-50	>= 51			1-19	20-50	>= 51
Anzahl Feldgruppen	1	7	7	10	Anzahl Feldgruppen	1	5	5	7
	2-5	7	10	15		2-5	5	7	10
	>=6	10	15	15		>=6	7	10	10



5. Story: Functions Points



Eingabe		Anzahl Datenelemente		
		1-4	5-15	>= 16
Anzahl Datenbestände	0-1	3	3	3
	2	3	4	6
	>=3	4	6	6

Ausgabe		Anzahl Datenelemente		
		1-5	6-19	>= 20
Anzahl Datenbestände	0-1	4	4	5
	2-3	4	5	7
	>=4	5	7	7

Abfrage		Anzahl Datenelemente		
		1-5	6-19	>= 20
Anzahl Datenbestände	0-1	3	3	4
	2-3	3	4	6
	>= 4	4	6	6

Interner Datenbestand		Anzahl Datenelemente		
		1-19	20-50	>= 51
Anzahl Feldgruppen	1	7	7	10
	2-5	7	10	15
	>=6	10	15	15

Externer Datenbestand		Anzahl Datenelemente		
		1-19	20-50	>= 51
Anzahl Feldgruppen	1	5	5	7
	2-5	5	7	10
	>= 6	7	10	10



5. Story: Functions Points



Eingabe		Anzahl Datenelemente		
		1-4	5-15	>= 16
Anzahl Datenbestände	0-1	3	3	3
	2	3	4	6
	>=3	4	6	6

Ausgabe		Anzahl Datenelemente		
		1-5	6-19	>= 20
Anzahl Datenbestände	0-1	4	4	5
	2-3	4	5	7
	>=4	5	7	7

Abfrage		Anzahl Datenelemente		
		1-5	6-19	>= 20
Anzahl Datenbestände	0-1	3	3	4
	2-3	3	4	6
	>= 4	4	6	6

Interner Datenbestand		Anzahl Datenelemente		
		1-19	20-50	>= 51
Anzahl Feldgruppen	1	7	7	10
	2-5	7	10	15
	>=6	10	15	15

Externer Datenbestand		Anzahl Datenelemente		
		1-19	20-50	>= 51
Anzahl Feldgruppen	1	5	5	7
	2-5	5	7	10
	>= 6	7	10	10

} 5-10%



5. Story: Functions Points – Vereinfachte Prognose



Eingabe		Anzahl Datenelemente		
		1-4	5-15	>= 16
Anzahl Datenbestände	0-1	3	3	3
	2	3	4	6
	>=3	4	6	6

Ausgabe		Anzahl Datenelemente		
		1-5	6-19	>= 20
Anzahl Datenbestände	0-1	4	4	5
	2-3	4	5	7
	>=4	5	7	7

Abfrage		Anzahl Datenelemente		
		1-5	6-19	>= 20
Anzahl Datenbestände	0-1	3	3	4
	2-3	3	4	6
	>= 4	4	6	6

Fehler des Gesamtergebnisses: 5% (Poensgen / Bock, *Function Point Analyse*)



5. Story: CoCoMo



Von der Größenbestimmung zur Schätzung:

$$\text{Aufwand} = K * (\text{Größe}) ^ S$$

- K = 17 Kostentreiberfaktoren
 - Reliability
 - Database size
 - ...
- S = 5 Skalenfaktoren
 - Precedentedness (Vorerfahrung)
 - Development Flexibility (Flexibilität des Entwicklungsprozesses)
 - Risk Resolution (Wird Risikomanagement betrieben?)
 - Team Cohesion (Wie ist die Zusammenarbeit im Team?)
 - Process Maturity (Prozessreife: Welcher CMM-Level?)



5. Story: Abstrakte Schätzmaße – Velocity



$$\text{Dauer} = \text{Größe} / \text{Velocity}$$



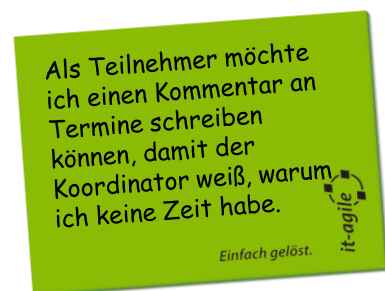
- Schätzungen sind selbstkorrigierend durch Geschwindigkeitsfaktor
- Geschwindigkeitssteigerung messbar
- Größen-Schätzungen altern nicht
- Objektivität: Von den Individuen getrennt. „*Meine Stunden sind nicht deine Stunden*“
- Vergleichende/ abstrakte Schätzungen sind schneller



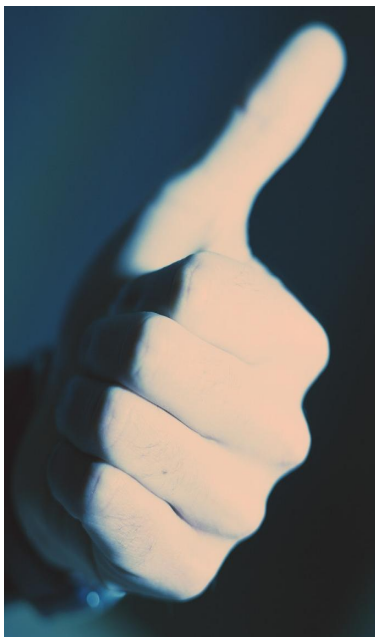
6. Story: Schätzfreundliche Anforderungen



- Anforderungen feingranularer aufschreiben
 - Z.B. Epics, MMF (Minimal Marketable Features)
 - Darunter User-Stories
- Fachliche Ausbaustufen einführen
 - Umfassen dann eine Menge an Epics/MMFs
- Vorteile:
 - Erleichtert die Schätzung (je kleiner das Teil, um so leichter zu schätzen)
 - Teile können untereinander gewichtet werden
 - Simple Preisberechnung, wenn Teile herausgenommen werden
 - Geschwindigkeit durch Rückkopplung: Schaffe ich diesen Teil in X PT?



Abschluss



- Schätzen bleibt eine Herausforderung.
- Es gibt Mittel, um Schätzungen zu vereinfachen und zu verbessern.
- Agile Sichtweise: Wir schätzen nur kleinere Häppchen genauer.
- Agile Sichtweise: Schätzen von Festpreisen kostet Geld.
 - ‚einfach‘ loslegen und schrittweise Anforderungen realisieren könnte für den Kunden günstiger sein
 - insbesondere wenn unterwegs Anforderungen getauscht werden können oder entfallen dürfen: „Money for nothing, change for free.“



6. Session 6: Modellbasierte Systementwicklung

Inhalt

6.1. Modelle verbinden: Wie Projektleiter und Entwickler reibungslos zusammenarbeiten	226
6.2. Modellbasierter Ansatz im Multi-Projekt-Umfeld	237
6.3. Effiziente Wege zum Datenaustausch zwischen Behörden	250

6.1. Modelle verbinden: Wie Projektleiter und Entwickler reibungslos zusammenarbeiten

Johannes Becker, Klaus Bergner, Vinko Novak

4Soft GmbH

Mittererstraße 3

80336 München

{becker | bergner | novak}@4soft.de

Abstract

Leichtgewichtige Entwicklungsansätze verfolgen das Ideal, die Anforderungen möglichst direkt und nachvollziehbar in Quellcode umzusetzen. Dazu werden die Anforderungen Arbeitsaufträgen zugeordnet, welche dann die Entwickler bearbeiten. Der Vortrag zeigt, wie sich eine nachvollziehbare und gesteuerte Entwicklung auch im Rahmen eines modellgetriebenen, architekturzentrierten Ansatzes erreichen lässt. Der Schlüssel dazu ist die Automatisierung von Entwicklungs- und Dokumentationsaktivitäten durch spezielle Code- und Reportgeneratoren auf Basis einer definierten Modellstruktur.

Motivation

Zentrale Aufgabe der Software-Entwicklung ist es, Anforderungen möglichst effizient und bruchlos in Quellcode umzusetzen. Werden die Zusammenhänge überdies nachvollziehbar dokumentiert, so lässt sich in der Folge abschätzen, wie sich Änderungen von Anforderungen auf den Quellcode auswirken und was sie kosten.

Moderne aufgabenorientierte Entwicklungsumgebungen nutzen die Beziehung zwischen Anforderungen und Quellcode unmittelbar zur Steuerung von Projekten. Die Anforderungen werden zunächst gruppiert und dienen so als Grundlage für die Formulierung von Arbeitsaufträgen. Die Entwickler nehmen diese Aufträge an, bearbeiten sie und schließen sie ab. Über die fertiggestellten Arbeitsaufträge gewinnen Projektleiter ein klares Bild vom Entwicklungsstand und können zukünftige Arbeiten schätzen und planen.

Dieses idealisierte Bild hat allerdings einen Haken: Bei komplexen Systemen ist es nicht möglich, in einem Schritt aus einer Anforderung Quellcode zu „zaubern“: Hier steht zunächst einmal die Analyse und Modellierung der Fachdomäne an, auf deren Basis dann die fachliche und technische Architektur entworfen werden.

In vielen Projekten sind diese Modelle und Entwürfe leider nur ein Nebenprodukt - sie entstehen zwar notgedrungen zu Anfang der Entwicklung, sind aber nur unzureichend mit Anforderungen und Quellcode verknüpft. Folglich werden sie bei Änderungen oft übergangen und veralten schnell.

Modellbasierte Ansätze schaffen durch die Automatisierung von Arbeitsschritten einen bruchlosen Übergang von Modellen zum Code. Allerdings haben sie oft Schwierigkeiten mit der Steuerung der Entwickler und der Nachvollziehbarkeit: Im schlechtesten Fall steht im Zentrum des Projekts ein „mysteriöser“ Generator, der „irgendwie“ aus einem Modell Quellcode erzeugt, den dann die Entwickler weiter ausarbeiten müssen, ohne genau zu wissen, wie die Querbezüge zu Anforderungen und Spezifikation sind.

Lösungsansatz

Die modellbasierte Entwicklung bietet ein wesentlich größeres, bisher ungenutztes Potenzial: Durch eine geeignete Strukturierung des Gesamt-Modells lässt sich ein bruchloser, nachvollziehbarer Übergang von den Anforderungen über Analysemodelle hin zu Entwurfsmodellen und schließlich zum Quellcode schaffen und in Form von Reports dokumentieren. Damit ist sichergestellt, dass tatsächlich alle Anforderungen umgesetzt werden, und die Auswirkungen von Änderungen werden klar.

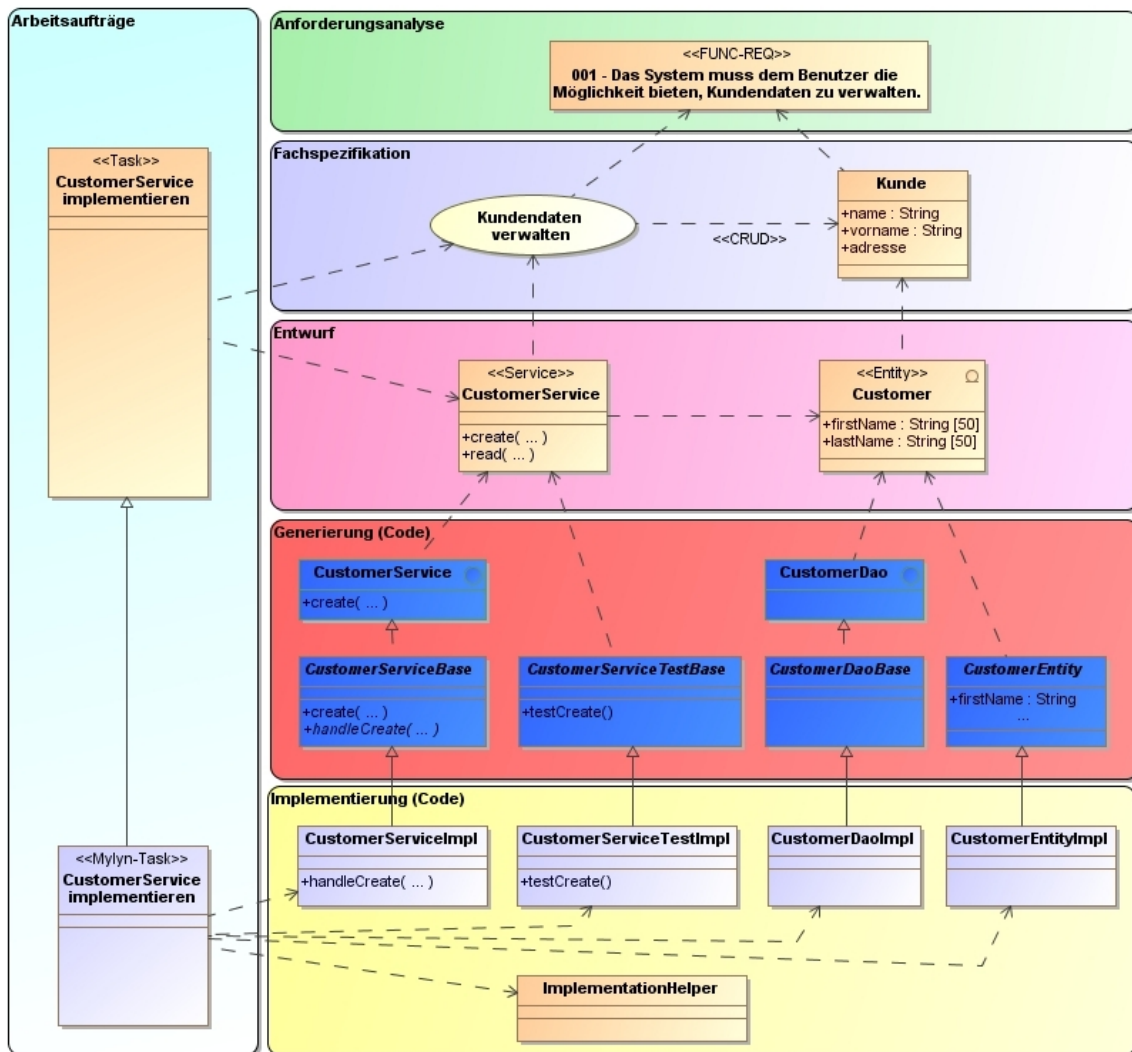


Abbildung 6.1.: ModellIntegra-Modellstruktur bei einem service-orientierten System

Nimmt man zusätzlich die Arbeitsaufträge in das Modell auf, so eröffnen sich weitere Möglichkeiten: Arbeitsaufträge enthalten dann nicht nur einen Verweis auf die „rohen“ Anforderungen, sondern referenzieren zusätzlich den relevanten Modellkontext und die generierten Quellcode-Rahmen. Entwickler benötigen damit weniger Zeit, sich im Modell und im Code zurechtzufinden und können unmittelbar nach der Annahme eines Arbeitsauftrags produktiv arbeiten.

Abbildung 6.1 zeigt einen Überblick über eine entsprechende Modellstruktur am Beispiel der modellbasierten Entwicklung eines service-orientierten Systems mit der werkzeug-

6.1. Modelle verbinden: Wie Projektleiter und Entwickler reibungslos zusammenarbeiten

gestützten ModellIntegra-Methode [3]. Die Entwicklung startet mit der Analyse textueller Anforderungen. Sie werden in einem externen Tool (im einfachsten Fall mit Hilfe von Excel) erstellt und werkzeuggestützt in ein Modellierungswerkzeug wie beispielsweise MagicDraw [2] importiert. Auf dieser Basis erarbeiten Systemanalysten eine fachliche Spezifikation des Systems in Form von Anwendungsfällen und Fachklassen, die sie mit den jeweiligen Anforderungen verknüpfen.

Im nächsten Schritt leitet der Architekt daraus ein Entwurfsmodell ab, das die Services und die Datenobjekte des zu entwickelnden Systems beschreibt. Es ist so strukturiert und mit technischen Informationen angereichert, dass es in einen modellgetriebenen Generator (beispielsweise das Werkzeug AndroMDA) eingespeist werden kann.

Der Generator legt die technische Zielarchitektur fest (dunkelblau dargestellte Klassen) und erzeugt Implementierungsklassen, die in der Folge von den Entwicklern ausgearbeitet werden können (hellblau dargestellte Klassen). Zusätzlich stellt ModellIntegra im Modell die Beziehung zum technischen Entwurf her und sorgt damit dafür, dass die Kette der Nachvollziehbarkeit nicht unterbrochen wird.

Die Methode sieht zusätzlich die Modellierung von Arbeitsaufträgen im Modell vor. Der Architekt erstellt sie als Klassen mit speziellen Stereotypen und setzt sie zu den Entwurfs-Entitäten (und damit auch zu den zugehörigen Anwendungsfällen, Fachklassen und Anforderungen sowie den generierten Code-Rahmen) in Beziehung. Eine spezielle ModellIntegra-Generatorfunktionalität erzeugt dann aus den modellierten Arbeitsaufträgen Tasks für das Mylyn-Plugin [4] der Entwicklungsumgebung Eclipse [1].

Nehmen die Entwickler derartige Tasks an, so präsentiert ihnen Eclipse die jeweils relevanten Quellcode-Klassen und (Referenzen zu den) generierten Modellen und Spezifikationen. Mehr noch: Zusätzliche Implementierungsklassen, die die Entwickler während ihrer Arbeit am jeweiligen Arbeitsauftrag erstellen, lassen sich grundsätzlich automatisch in die Kette der Nachvollziehbarkeit einbinden. Aktuell unterstützen die ModellIntegra-Werkzeuge diesen Schritt allerdings noch nicht.

Bewertung

Durch den konsequenten Einsatz der ModellIntegra-Methodik und der zugehörigen Werkzeuge werden Modelle zum Dreh- und Angelpunkt der Entwicklung. Sie schlagen die Brücke zwischen Anforderungen und dem Quellcode und ermöglichen so eine nachvollziehbare, geregelte und automatisierte Software-Entwicklung.

Literaturverzeichnis

1. Website der Open-Source-Entwicklungsumgebung Eclipse, www.eclipse.org, Eclipse Foundation.
2. Website des Modellierungswerkzeugs NoMagic MagicDraw, www.magicdraw.com, NoMagic, Inc.
3. Website des Open-Source-Projekts ModellIntegra, www.modelintegra.com, 4Soft GmbH.
4. Website des Mylyn-Eclipse-Plugins, www.eclipse.org/mylyn/, Eclipse Foundation.



Modelle verbinden: Wie Projektleiter und Entwickler reibungslos zusammenarbeiten

SEE 2009

Johannes Becker
Dr. Klaus Bergner
Vinko Novak

26. Mai 2009

©2009 4Soft GmbH

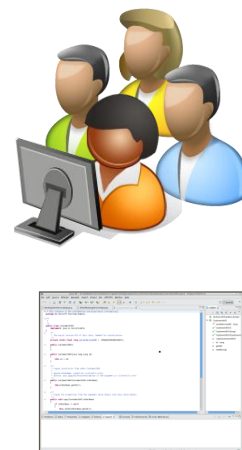


Anwender

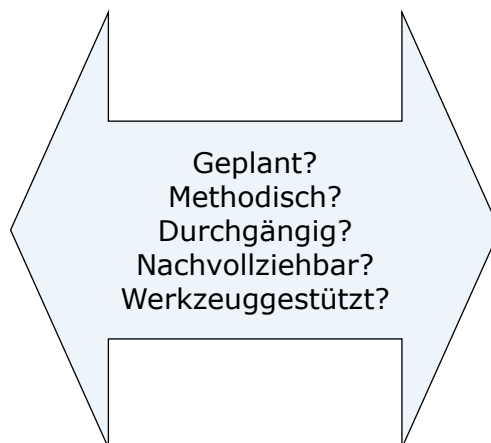


Anforderungen

Entwickler

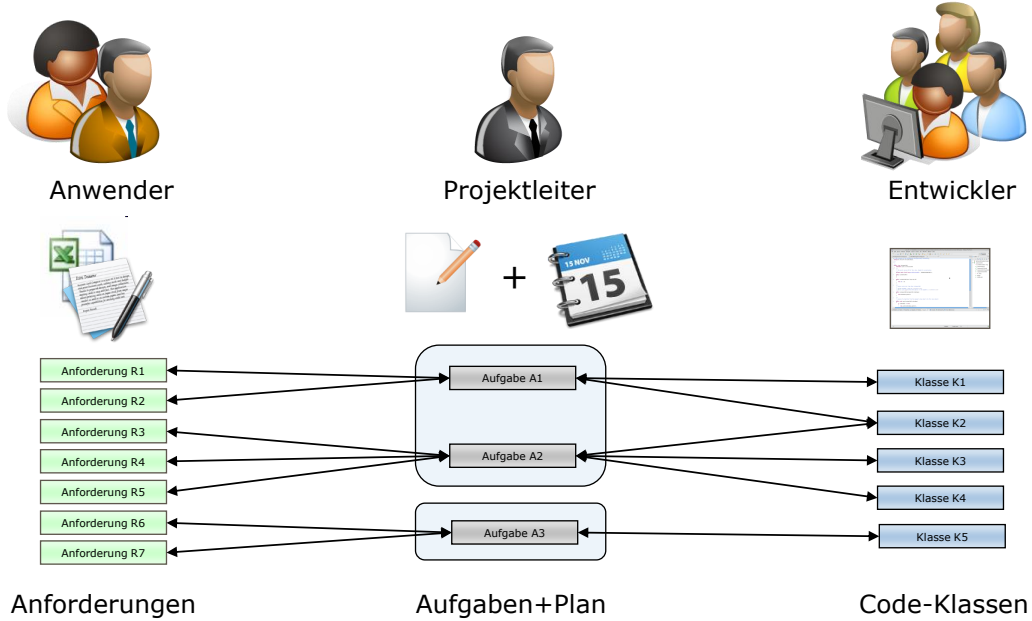


Quellcode

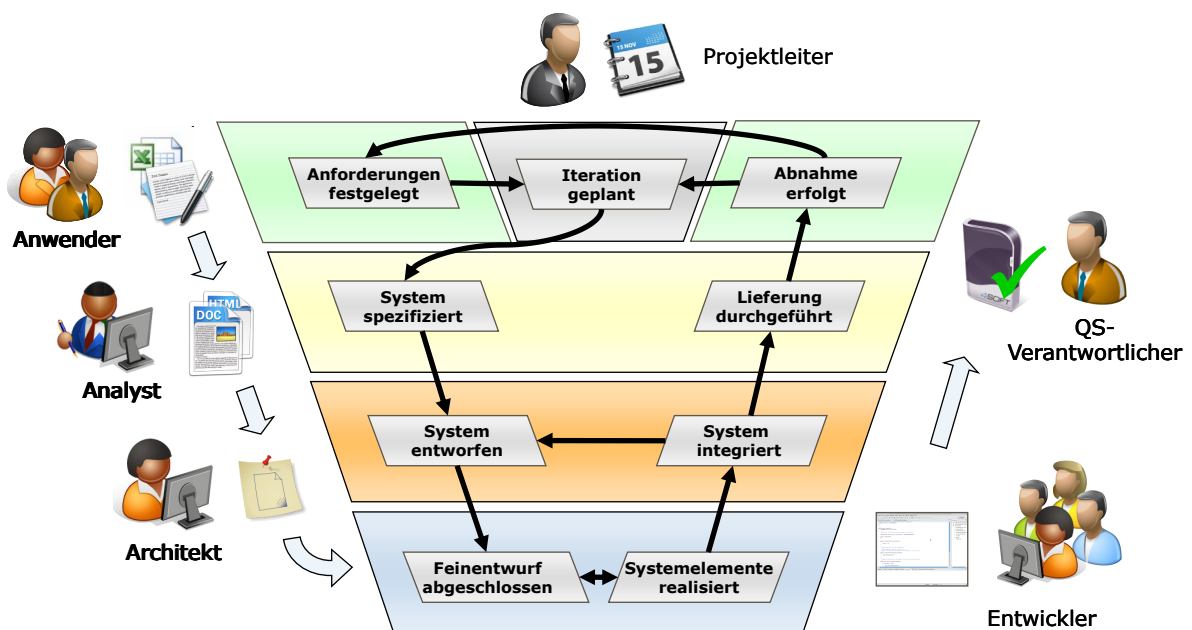




Aufgaben-Orientierte Entwicklung

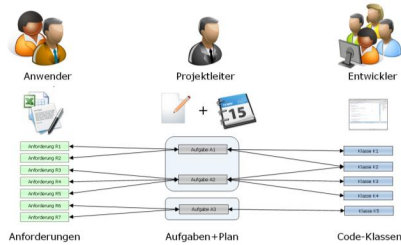


Dokument-orientierte Entwicklung



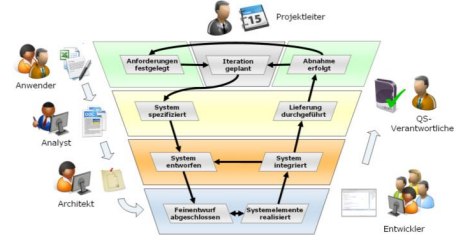


Aufgabenorientierung



- ✓ Gute Planbarkeit
- ✓ Hohe Durchgängigkeit und Nachvollziehbarkeit
- ✓ Gute Werkzeugunterstützung
- ✗ ... aber wo sind Spezifikation und Architektur geblieben?

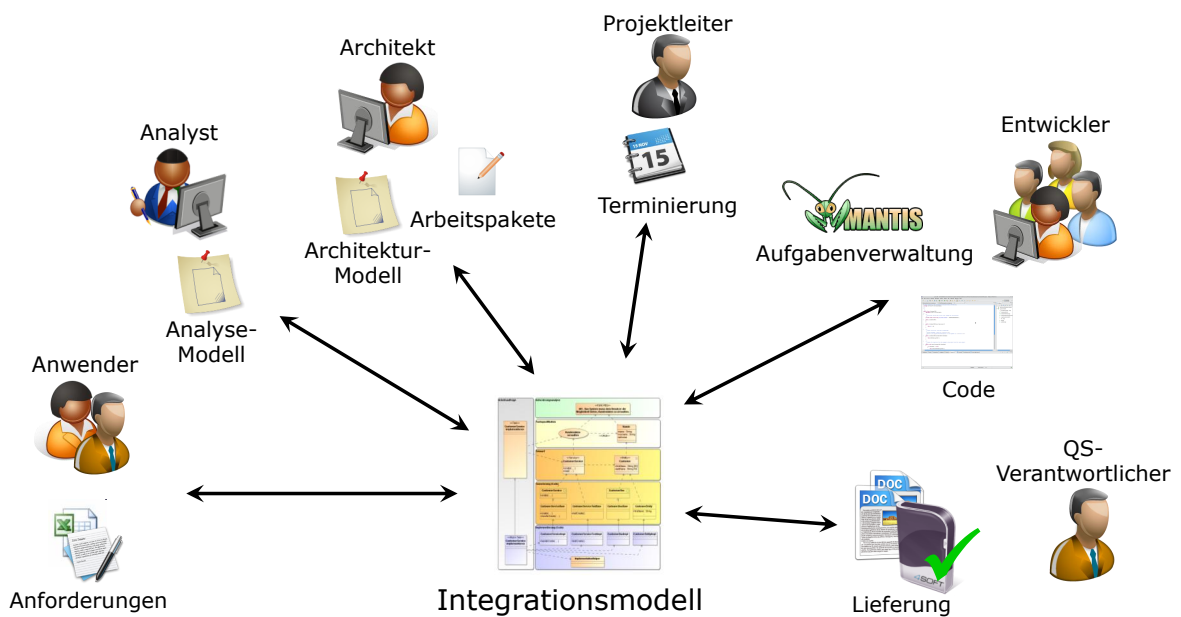
Dokumentorientierung



- ✓ Arbeitsteilung
- ✓ Explizite Ergebnisse für Spezifikation und Architektur
- ✓ Langlebige Dokumentation
- ✗ ... aber Durchgängigkeit und Planbarkeit sind kritisch!

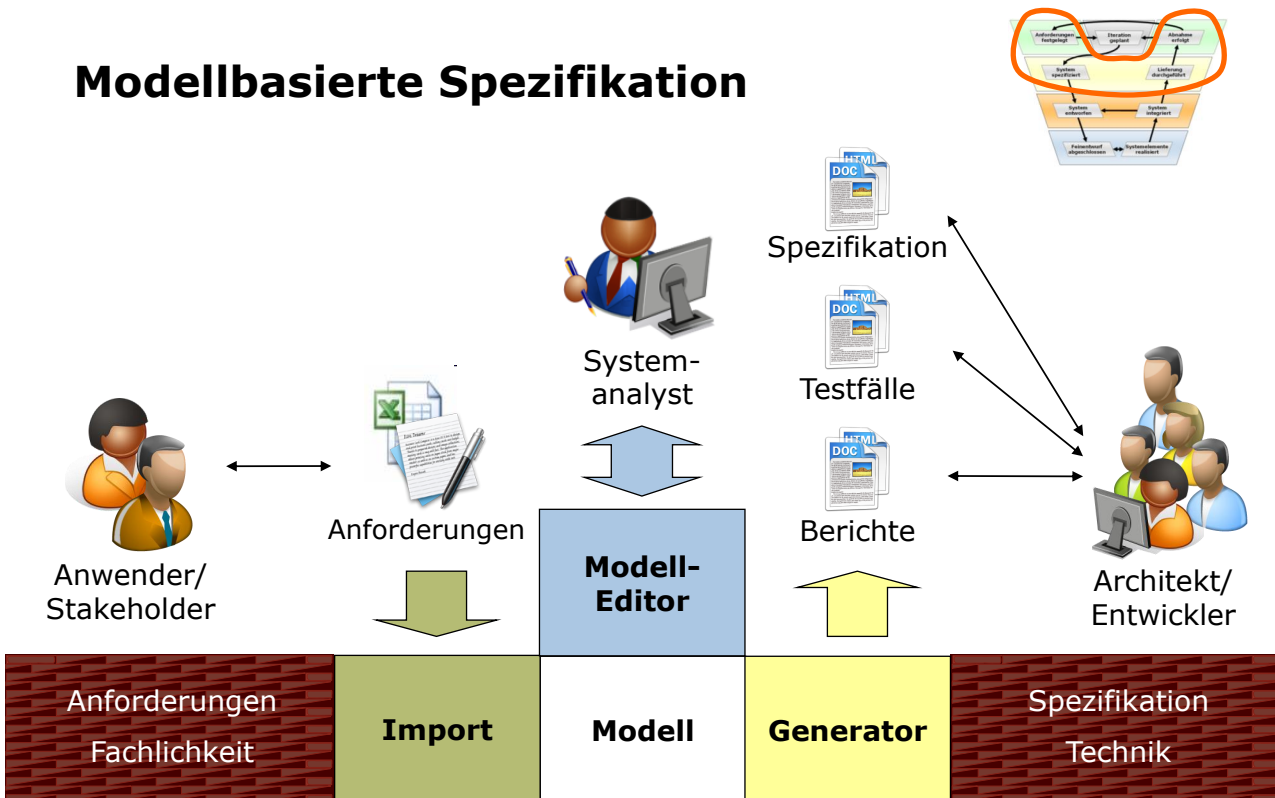


Modellbasierung: Ein Modell als Verbindung

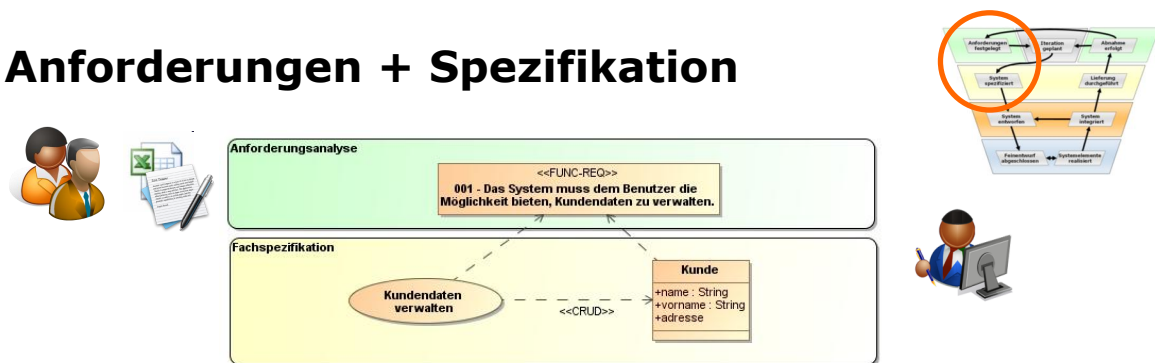




Modellbasierte Spezifikation



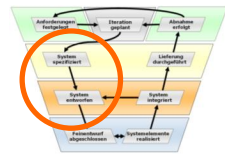
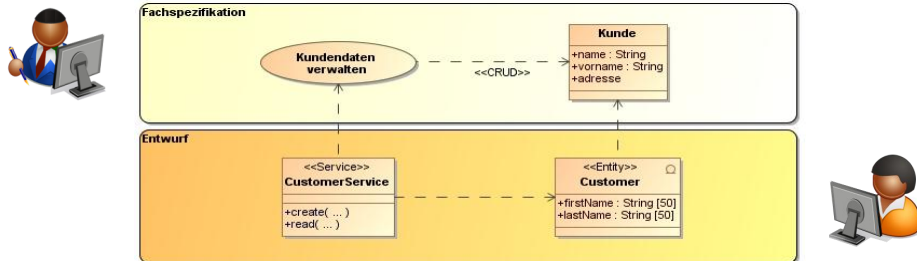
Anforderungen + Spezifikation



- Spezifikation gemäß ModelIntegra-Methode mit UML-Tool
 - Anwendungsfälle für Funktionalität
 - Aktivitätsdiagramme für Interaktion
 - Fachklassen für Daten
- Verlinkung mit eindeutig identifizierten Anforderungen
 - Import und Synchronisation u.a. von Excel-Listen
 - Darstellung als stereotypisierte Klassen



Spezifikation + Architektur



- Architekturentwurf mit UML-Werkzeug
 - Ableitung von Services für Anwendungsfälle
 - Ableitung von Entities und DTOs für Daten
- Verlinkung der Architekturklassen
 - Abhängigkeitsbeziehungen zu Spezifikationskonzepten
 - Aufbereitung und Anreicherung für MDA-Ansatz

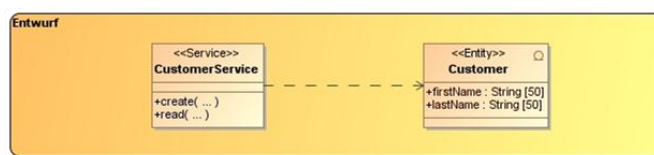
© 4Soft GmbH

9

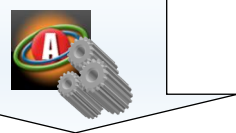


MDA: Architektur → Implementierung

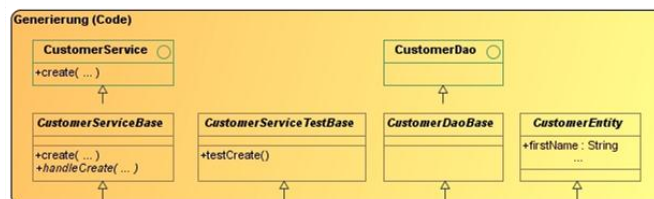
Entwurfsklassen



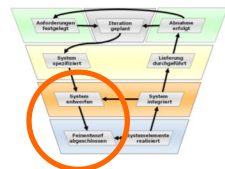
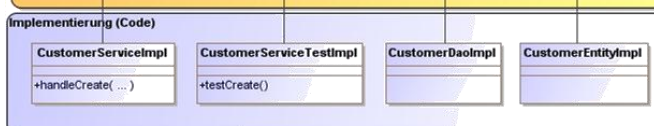
Generator



Generiertes Framework



Rahmenklassen



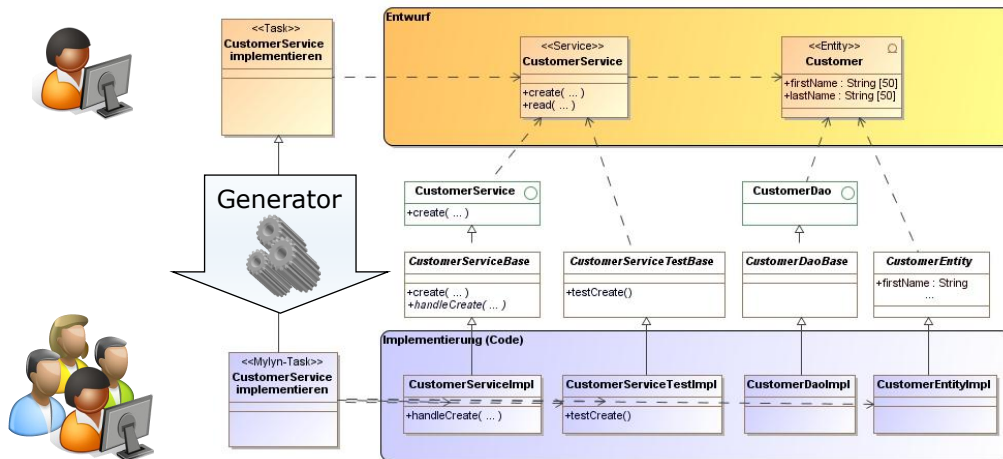
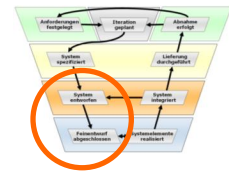
© 4Soft GmbH

10



Modellierung von Aufgaben

- Architekt modelliert Aufgaben in UML
- Architekt verknüpft Aufgaben mit Kontext



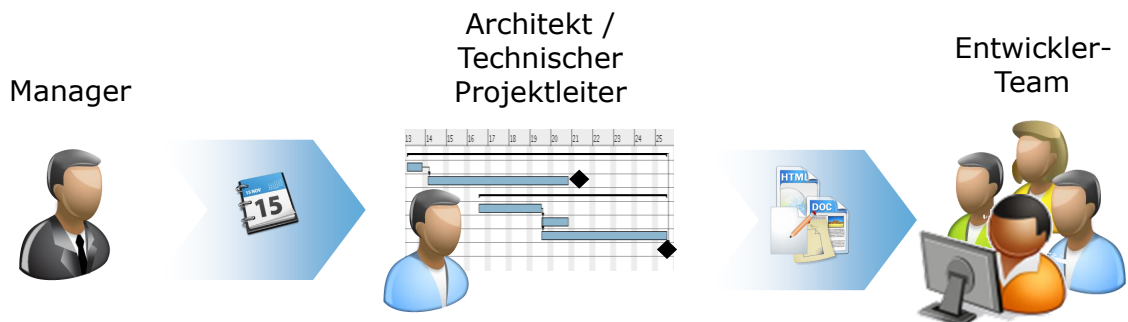
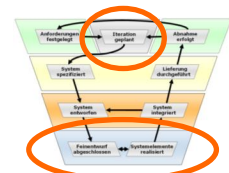
- Entwickler erhalten Mylyn-Kontext: Klassen und Dokumentation



Projektleiter vs. Entwickler

Wie die Umsetzung steuern?

- Meilensteine und Abnahmen planen
- Arbeitsaufträge vergeben und überwachen
- Zeit, Kosten und Qualität überwachen und steuern





Sichten auf die Projektplanung

Projektleiter / Manager

System bauen und integrieren	3/24/08	63
AndroMDA aufsetzen	3/24/08	5
Zeiterfassung [Server] bauen	3/31/08	35
Zeiterfassung [Client] bauen	4/17/08	45
Serverzugriffs-Schicht implementieren	4/17/08	15
Buchungsraum abschließen implementieren	5/8/08	7
Zeiterfassungstabelle bauen	5/8/08	30
System gebaut und integriert	6/19/08	2

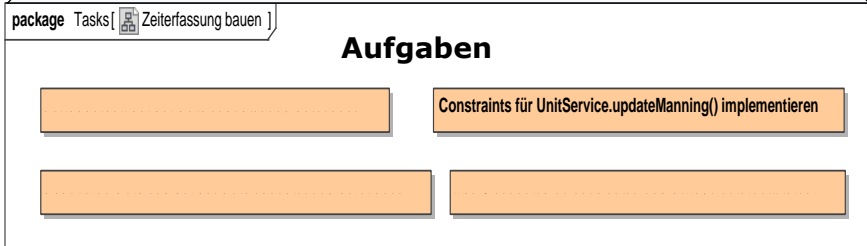
Arbeitspakete und Meilensteine

Technischer PL / Architekt

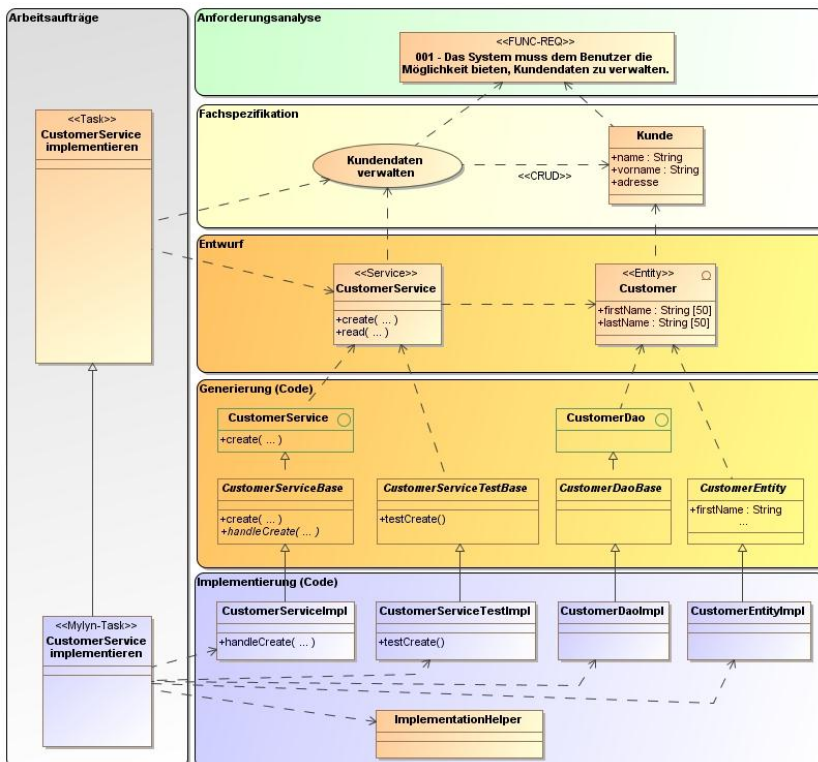


Architektur

Entwickler-Team



Projektleiter



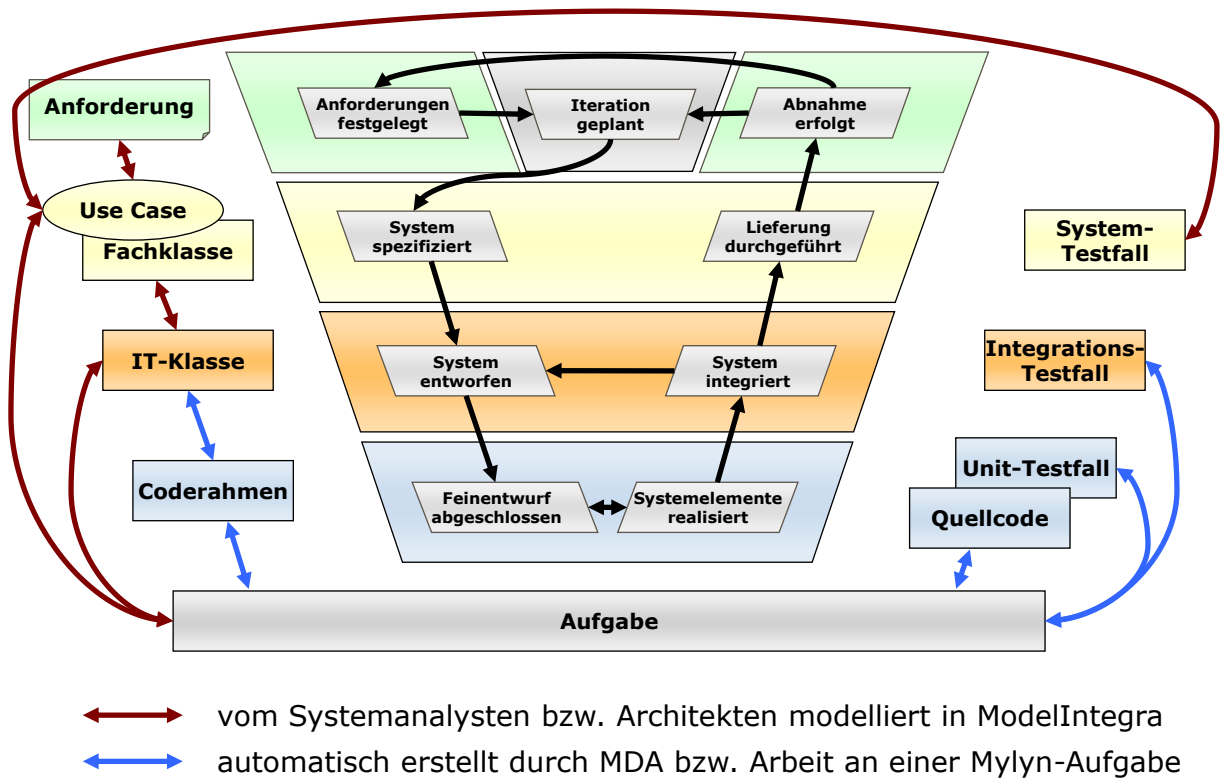
Anwender
Analyst

Architekt

QS-Verantwortlicher

Entwickler

Durchgehende Traceability



Modellbasierung und Aufgabenorientierung



- Aufgaben erleichtern Planung und Controlling
 - Einfache Zuweisung an (andere) Entwickler
 - Reports zum Status automatisch generierbar
 - Aber: Pflege des Aufgabenkontexts erforderlich



- Durchgängigkeit und Traceability
 - Traceability von der Anforderung bis zum Quellcode
 - Generierung von Dokumenten und Reports
 - Hohe Transparenz für alle Beteiligten



- Erleichterungen für Entwickler
 - Alle Aufgaben sind auf einen Blick in der IDE sichtbar
 - Kontext erleichtert Übernahme von Aufgaben
 - Commit-Beschreibung automatisch generierbar, z.B. "RESOLVED - Aufgabe 4410: Implementiere Login/out"

6.2. Modellbasierter Ansatz im Multi-Projekt-Umfeld

Marc Kurzmann, Tilman Seifert

iteratec GmbH Inselkammerstr. 4 82008 München-Unterhaching
{tilman.seifert | marc.kurzmann}@iteratec.de

Abstract

Dieser Erfahrungsbericht stellt das Projekt-Umfeld vor, in dem eine Reihe gleichartiger Anwendungen auf Basis einer gemeinsamen Plattform entwickelt und gepflegt werden. Er zeigt den gewählten modellbasierten Ansatz und diskutiert die Auswirkungen auf den Entwicklungsprozess und die Architektur. Der Bericht schließt mit Beobachtungen hinsichtlich Qualität und Evolution der Anwendungen und der Plattform.

Motivation

Ziel des hier vorgestellten Gesamtvorhabens ist es, eine große, über lange Jahre gewachsene Altanwendung durch eine moderne Lösung abzulösen. Das Altsystem wird von seinem Hersteller zum Teil mit erheblichen individuellen Anpassungen an seine Kunden verkauft. Die neue Lösung soll folgenden Anforderungen gerecht werden:

1. Modularer Aufbau: Eine einheitliche Plattform soll technische Basisaufgaben wie Datenbankzugriff, Authentisierung und Autorisierung, Standard-GUI-Darstellungen, Monitoring etc. zur Verfügung stellen.
2. Die fachliche Funktionalität soll in mehrere Anwendungen aufgeteilt werden, so dass die einzelnen Anwendungen über klar definierte Schnittstellen miteinander kommunizieren. Auf diese Weise soll die Weiterentwickelbarkeit der fachlichen Anwendungen langfristig sichergestellt werden.
3. Für die Entwicklung und Pflege der fachlichen Anwendungen soll eine Werkzeugkette bereitgestellt werden, die die Anwendungsentwickler von technischen Aspekten weitgehend entlastet.

Zu den wichtigsten Herausforderungen in diesem Vorhaben zählt dabei neben der Architektur des Gesamtsystems die Auswahl des Entwicklungsansatzes für die fachlichen Anwendungen. Dieser Aspekt beeinflusst die Möglichkeiten des Herstellers zur Reaktion auf neue oder geänderte fachliche Anforderungen und die Dauer der Umsetzung maßgeblich. Diese Flexibilität ist bestimmend für die Marktposition des Herstellers.

Modellbasierter Ansatz

Für das System wird eine Architektur gewählt, die weitgehend auf Standard-Komponenten wie Java JEE5 und Spring aufbaut. Für besondere Anforderungen des Systems werden spezifische Komponenten ergänzt, beispielsweise Erweiterungen der GUI-Seite für einheitliches Look-and-Feel der Anwendungen.

Für die Entwicklung der fachlichen Anwendungen wird ein modellbasierter Ansatz gewählt. Fachliche Entitäten und Anwendungsfälle werden in einer spezifisch entwickelten

6.2. Modellbasierter Ansatz im Multi-Projekt-Umfeld

DSL (Domain Specific Language) modelliert. Daraus werden Klassen zum Datenbankzugriff, Entitätsklassen, Gerüste für die Fachlogik und für Tests, GUI-Dialoge sowie Dokumentation generiert. Die DSL ist textuell; das eingesetzte Werkzeug ist openArchitectureWare (oAW, <http://www.openarchitectureware.org>).

Technische Details werden auf diese Weise weitgehend in den Generierungstemplates versteckt und sind für den Anwendungsentwickler transparent. Dies hat einerseits den Vorteil, dass Änderungen der Plattform oft keine Anpassungen der Anwendungen notwendig machen; lediglich eine Neugenerierung ist erforderlich. Andererseits ist die Architektur und ihre konkrete Umsetzung im Code in den Generierungstemplates manifestiert und kann innerhalb der Anwendungen nicht verletzt werden. Dies sichert die Aufgabenteilung zwischen den Anwendungsentwicklern und dem Architekturteam, das auch die Plattformkomponenten entwickelt.

Die DSL ist passend für dieses Gesamtvorhaben definiert. So kann frei definiert werden, welche Aspekte modelliert werden. Die Sprache selbst kann im Laufe der Zeit angepasst und erweitert werden, wenn neue Anforderungen an die Modellierung entstehen.

Erfahrungen

Aktuell sind die wesentlichen Plattformkomponenten fertig gestellt; die DSL ist definiert und stabil. Die ersten Anwendungen sind entwickelt. Es liegen Erfahrungen vor, die es erlauben, den gewählten Weg kritisch zu bewerten.

Der Entwicklungsprozess und die Projektdurchführung sind wesentlich durch den modellbasierten Ansatz beeinflusst. Einerseits entsteht ein Zusatzaufwand durch die Definition der DSL und die Erstellung der Generierungstemplates. Dem steht jedoch ein Effizienzgewinn durch die schnelle Erstellung neuer Anwendungen gegenüber, der schon sehr früh, teilweise schon während der ersten Anwendung entsteht. Nach unserer Erfahrung ist die wichtigste Ursache für diesen sehr frühen „return on invest“, dass früh klare Architekturentscheidungen getroffen werden, die durch den Generierungsprozess klar und unmissverständlich formuliert werden; dies beschleunigt die Kommunikation zwischen allen Projektbeteiligten.

Die Modellbasierung trägt so unmittelbar dazu bei, eine einheitliche Architektur konsequent durchzusetzen und einzuhalten - und gleichzeitig noch Feinjustierungen und Änderungen zu erlauben, die die Zusammenarbeit zwischen verteilt arbeitenden Teams nicht beeinträchtigt.

Alle Anwendungen liegen in einer einheitlichen Struktur vor; dadurch sind QS-Maßnahmen wie z. B. Reviews effizient durchführbar, weil die Einarbeitungszeit der Reviewer in fremden Code recht gering ist, da sie die wesentlichen Strukturen bereits aus anderen Anwendungsprojekten kennen.

Der zentrale Punkt des modellbasierten Ansatzes sind die Modelle und das zugrundeliegende Meta-Modell. Im vorgestellten Ansatz kann das Meta-Modell erweitert werden, da es projektspezifisch definiert ist. Dies ermöglicht es, auch die Plattform und ihre Schnittstellen überarbeiten, ohne jedes Mal alle Anwendungen anpassen zu müssen. So wird es möglich, bei der Evolution der Plattform neue Funktionen einzubauen und sich von „Altlasten“ zu trennen - ein wesentlicher Faktor für den Erhalt von Qualität und langfristiger Weiterentwickelbarkeit.



Modellbasierter Ansatz im Multi-Projekt-Umfeld

Marc Kurzmann, Tilman Seifert

Software Engineering Essentials, 26. Mai 2009, Berlin

Agenda

- Herausforderung Multi-Projekt-Umgebungen
- Modellbasierter Ansatz
 - Architektur
 - Tools und Entwicklungsprozess
 - Zusammenarbeit mit Fachabteilung bzw. Kunde
- Erfahrungsbericht

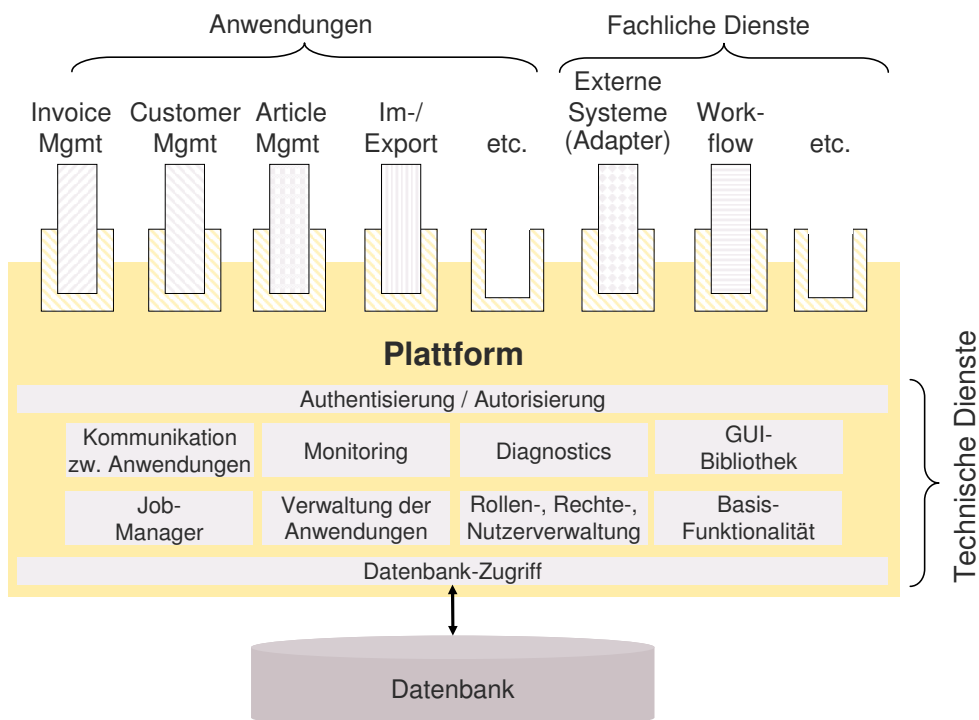
Multi-Projekt-Umfeld

- Eigenschaften
 - Mehrere Projekte, gleiche Plattform
 - Abhängigkeiten zwischen den Projekten
 - Ein Anwendungssystem, bestehend aus Teilprojekten
- Ansätze zur Effizienzsteigerung
 - Synergien
 - Ähnlichkeit der Teilprojekte, Architektur-Konformität
 - Technisches Know-How in anderen / neuen Projekten nutzbar
 - Architektur des Gesamtsystems
 - Klare Strukturierung der Fachlogik
 - Trennung technische / fachliche Anteile
 - Klare Aufgabenverteilung
- Umsetzung: mit modellbasiertem Ansatz (MDSD)

3

© 2009 Iteratec GmbH

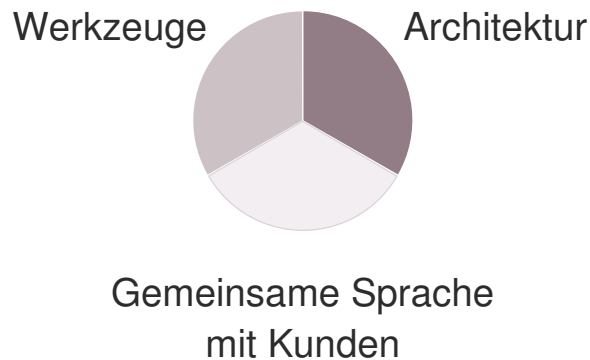
Architekturvision: Multi-Projekt-Umfeld



4

© 2009 Iteratec GmbH

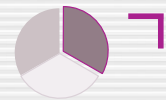
Lösungsansatz: Beruht auf drei Säulen



5

© 2009 Iteratec GmbH

Erfolgsfaktoren: Architektur

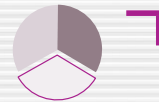


- Technische Basis
 - Funktionalität, Stabilität, Skalierbarkeit etc.
 - Kombination verfügbarer Komponenten / Frameworks
 - z.B. JEE5, Spring, Hibernate, JSF, Swing, JMX, JMS etc.
 - Leistung des Architekten
- Architektur für fachlichen Aufbau: Modularisierung
 - Projektspezifische Definition des Komponenten-Begriffs
 - Muss zur **fachlichen** Problemstellung passen
 - In unserem Beispiel: „Fachliche Anwendung“
 - Technische Standards hierfür nicht anwendbar
 - Spring-/JEE5-Bean: keine fachliche Komponente, zu klein!

6

© 2009 Iteratec GmbH

Erfolgsfaktoren: Fachlich

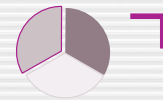


- Zusammenarbeit mit Kunde
 - Zügig fachliche Aspekte verstehen
 - Präzise, eindeutige Ergebnisse und Absprachen
 - Ergebnisse direkt im Entwicklungsprozess nutzbar
- Nutzung fachlicher Modelle
 - Passend zum fachlichen Problem, **ggf. spezifisch entwickelt**
 - Sprache muss für „**Fach**-Leute“ verständlich sein
 - Aber auch für Informatiker!
- Herausforderung
 - Verständliche Darstellung mit formal basierten Anteilen finden
 - Gerücht: „Je formaler, desto unverständlicher“
 - Sonst tauchen Verständnisprobleme bloß nicht auf

7

© 2009 Iteratec GmbH

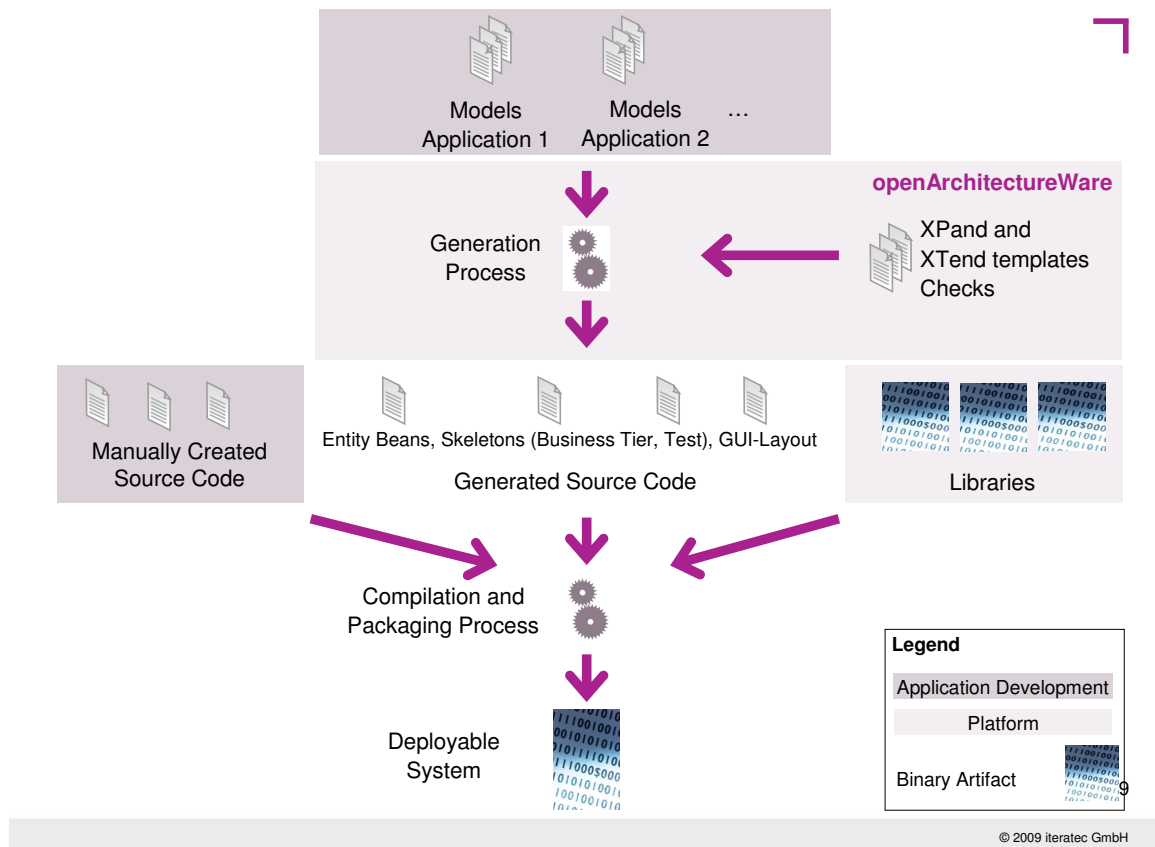
Erfolgsfaktoren: Artefakte, Werkzeuge, Prozesse



- Artefakte
 - Direkte Nutzung der Ergebnisse aller Projektphasen
 - Richtige Abstraktionsebene: Fachliche Begriffe + formale Modelle
 - Fachliche Modelle im Entwicklungsprozess: Definition, Akzeptanz
- Werkzeugkette
 - Generierung gleichartiger Artefakte
 - Auswahl, Konfiguration, Anpassung der Werkzeugkette
 - Wichtiges Projektergebnis
- Prozesse
 - Ausrichtung der Prozesse auf Zusammenarbeit mit Kunde

8

© 2009 Iteratec GmbH



Effizienz durch modellbasierte Entwicklung (MDSD)

- Nutzen fachlicher Modelle / Artefakte im Projekt
 - ➔ Single-Source-Prinzip
 - ➔ Unterschiedliche Artefakte generierbar
 - Code, Konfigurationen, GUI-Layout, Test-Rahmen etc.
 - Dokumentation: UML, tabellarische Darstellung
 - ➔ Fachlogik: ausprogrammiert (Java-Code)
- Architektur
 - ➔ In Generator-Templates manifestiert, Code-Rahmen vorgegeben
 - ➔ Vorteil: Formal definiert
 - ➔ Hoheit über Architektur: Langfristige Weiterentwickelbarkeit
- Werkzeuge: Generatoren und Editoren

Erfahrungsbericht



- Randbedingungen, Anforderungen
 - Architekturtreiber
 - Prozesstreiber
- Gewählte Architektur
- Gewählte Werkzeuge
- Fachliche Modelle im Einsatz
- Erkenntnisse, Erfahrungen

11

© 2009 iteratec GmbH

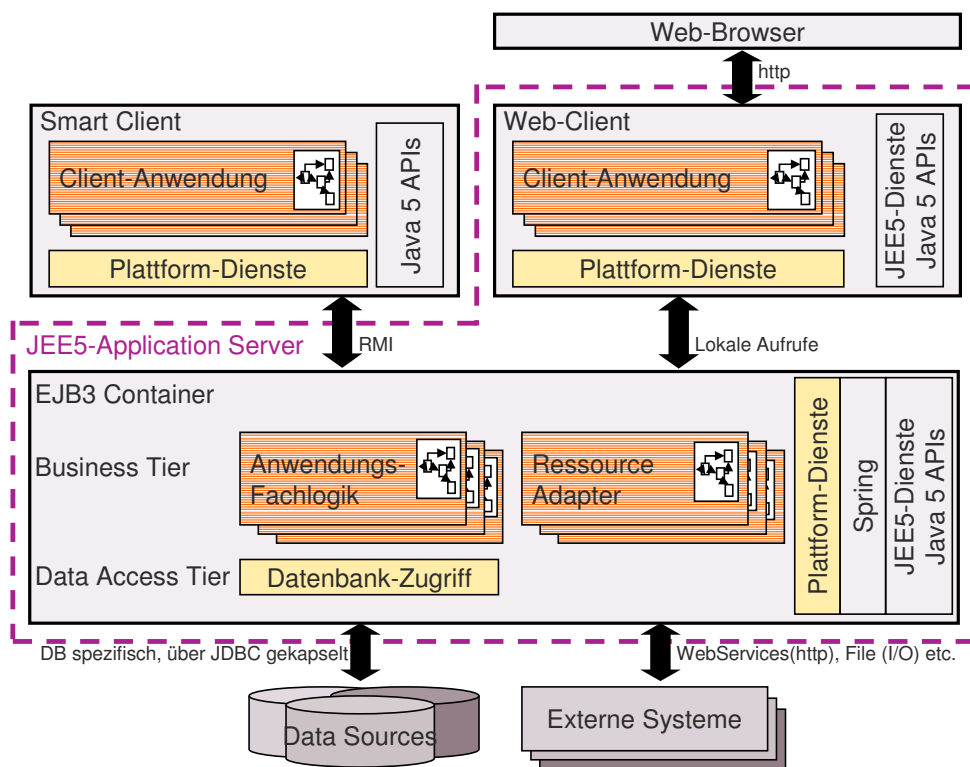
Erfahrungsbericht: Architektur-Treiber



- Flexibilität
 - Horizontale Skalierbarkeit: Modularisierung, Konfigurierbarkeit
 - Vertikale Skalierbarkeit: Unterschiedliche Installationsgröße
- Kostengünstige, schnelle Entwicklung fachlicher Anwendungen
 - Einsatz von Standards
 - Generierung
 - Wiederverwendung von Plattform-Funktionen
- Produktlinien / Varianten
 - Variantenbildung im Geschäft des Kunden unumgänglich
 - Strikte Modularisierung ~ Aufwand bei Branch/Merge minimieren
- Separate Weiterentwicklung von Plattform und Anwendungen
 - Klare Trennung durch definierte Schnittstellen:
 - iteratec: Plattform, Tool-Kette
 - Kunde: Fachliche Anwendungen
 - Getrennte Entwicklung der Anwendungen in unterschiedlichen Teams

12

© 2009 iteratec GmbH



13

© 2009 Iteratec GmbH

DSL: Beispiel (1/2)

```

application invoice;
// ...
entity Article {
  description { "Defines an article." }
  attributes {
    I18NString name {
      legacy „INVOICE.ARTICLE_NAME“;
      description { 'Name, e.g. „ScrewDriver XL“' }
      constraints {
        length 1:50;
        notNull;
        regExp "[A-Za-z0-9 -]*"; }
    }
  }
// ...
}
    
```

DSL: Textuell

- Schnelle Entwicklung
- Editor
- Leichtes Diff/Merge
- UML: generierbar
- openArchitectureWare



14

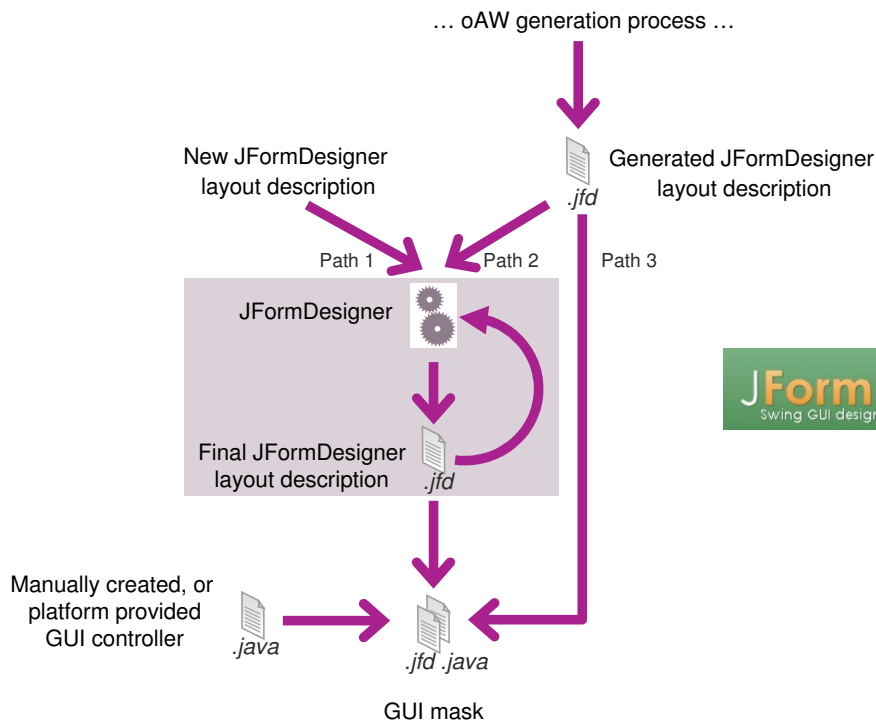
© 2009 Iteratec GmbH

DSL: Beispiel (2/2)

```
// ...
businessInterface {
  persistence { /* ... */ }
  activity ArticleMgmt {
    basicCRUD {
      Article: c r u d;
    }
    methods {
      int checkAvailability ();
    }
  }
}
```

15

© 2009 Iteratec GmbH



16

© 2009 Iteratec GmbH

Erkenntnisse / Erfahrungen (I)



- Zusammenarbeit und Arbeitsteilung
 - Fachliche Expertise durch Kunden
 - Technische Expertise durch iteratec
 - Hat sich gut bewährt!
- MDSD: Modellierung des fachlichen Datenmodells
 - Formal
 - Fachlich, nicht technisch (nicht DB-Schema): Konzentration auf fachliche Zusammenhänge
 - Saubere Spezifikation
 - Direkte Nutzung der Ergebnisse aus „frühen Phasen“
- Architektur
 - Generierte Anwendung per Konstruktion architekturkonform

17

© 2009 iteratec GmbH

Erkenntnisse / Erfahrungen (II)



- Projektspezifische, textuelle DSL
 - Akzeptanz der Entwickler hoch
 - Merge + Diff einfach
 - Für Diskussionen / Iterationen: UML + Doku generierbar
 - Dynamik der Sprache möglich
- Tipps:
 - Vom Wunsch ausgehen: „Wie würde ich es gerne ausdrücken“
 - Vorsicht: nicht versuchen, alles abzudecken
 - Übersichtlichkeit wahren
 - Grenze DSL/Java bewusst definieren
 - DSL: soll keine neue 4GL werden
 - Zunächst einfach halten, dann erweitern

18

© 2009 iteratec GmbH

Erkenntnisse / Erfahrungen (III)

- Pflege der Modelle + Fachlogik
 - Modelle sind neuer Artefakttyp → Projekt-Struktur aufwändiger
 - Refactorings ggf. komplexer (DSL, Templates, Java-Code)
- Evolution der Sprache
 - Editoren zwar gut, aber noch nicht so weit wie bei Java-Editoren
 - Refactoring der Templates umständlich
 - Steigerung der generierten Anteile möglich
- Evolution der Architektur
 - Architektur-Änderungen effizient möglich
 - Große Chance des MDSD-Ansatzes

19

© 2009 Iteratec GmbH

Zusammenfassung

- Effiziente Entwicklung ruht auf drei Säulen
 - Architektur: Auswahl der richtigen Komponenten
 - Werkzeuge: Auswahl und Feinschliff; lohnendes Investment
 - Modelle: Direkte Nutzung der Ergebnisse aus „frühen Phasen“
- Effizienzsteigerung mit modellbasiertem Ansatz
 - Trennung Fachlichkeit / technische Komplexität
 - Einhaltung einer passenden Architektur
- Umsetzung in der Praxis
 - Modellbasierter Ansatz – mit Augenmaß
 - Formaler Ansatz und pragmatischer Zugang
 - Dynamik im Projekt erlauben
 - Weiterentwicklung der DSL
 - Generierte Anteile schrittweise steigern

20

© 2009 Iteratec GmbH



Ihre Fragen?

Marc Kurzmann
kurzmann@infomax-it.de

Tilman Seifert
Tilman.Seifert@iteratec.de

6.3. Effiziente Wege zum Datenaustausch zwischen Behörden

Alexander Bösl, Klaus Weber
MID GmbH
Eibacher Hauptstrasse 141
90451 Nürnberg
{A.Boesl | K.Weber}@mid.de

Motivation

Die Behörden in der Bundesrepublik Deutschland müssen per Gesetz Informationen untereinander austauschen. Ohne eine gemeinsame Begrifflichkeit führt dies häufig zu Fehlinterpretationen und Kommunikationsschwierigkeiten. Ein Mehraufwand an teuren und zeitraubenden korrektiven Maßnahmen ist dabei die Folge.

Um die Interpretationsmöglichkeiten von Begriffen und Datentypen zu verringern, wurde von der OSCI-Leitstelle¹ ein Glossar erarbeitet (XÖV²). Kommunikationspartner werden durch diesen Standard in die Lage versetzt in kurzer Zeit einen eindeutigen Datenaustausch zu vereinbaren.

Auch Unternehmen haben gesetzliche Meldepflichten zu erfüllen³ oder kommunizieren untereinander⁴ zum Wohle ihrer Kunden beziehungsweise des Bürgers. Für die Telekommunikationsbranche gibt es zum Beispiel mit dem TM-Forum eine der OSCI-Leitstelle entsprechende zentrale Instanz zur Definition von Standards. Schwerpunkt in unseren Betrachtungen ist die XÖV-Thematik. Im Rahmen des öffentlichen Datenaustauschs versuchen die Behörden die Leistungen dem Bürger zeitnah und wirtschaftlich zur Verfügung zu stellen. Der rote Faden in Bezug auf die Umsetzung des Datenaustauschs sieht wie folgt aus:

- Auf Basis des XÖV-Standards lässt sich der Informationsbedarf und das Informationsangebot zunächst allgemeinverständlich vereinbaren. Danach kann der Dienstnutzer und der Dienstanbieter unabhängig voneinander die Übersetzung in seine eigene Begrifflichkeit vornehmen und dabei den Ausschnitt der relevanten Information aus seinem internen Datenmodell definieren.
- Bei der Definition der Übersetzungsabbildung aus und in die allgemein verstandene Austauschform kann es sein, dass man die zu übermittelnden Daten in XÖV nicht findet, oder mit anderen Typen versehen findet, was eine weitere Abstimmnotwendigkeit mit dem Kommunikationspartner oder der OSCI-Leitstelle bedeutet. Gerade diesen Mehraufwand gilt es aber zu verhindern bzw. schon von vorneherein zu minimieren.

Ansatz

In unserem im folgenden dargestellten Projekt ForumSTAR der bayerischen Justiz geht es darum Datenaustausch zwischen einer Vielzahl von Beteiligten zuzusichern u.a. zwi-

¹ Online Services Computer Interface . Siehe: <http://www1.osci.de/sixcms/detail.php?id=1181>

² XÖV steht für XML in der öffentlichen Verwaltung

³ z.B. Lohnsummen, Sozialversicherungsbeiträge

⁴ Z.B. Roaming-Informationen im GSM-Netz

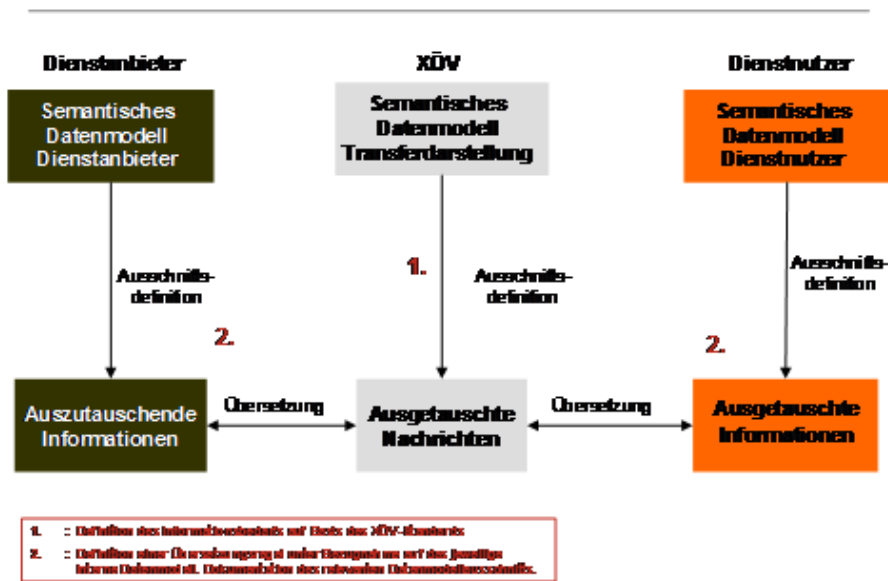


Abbildung 6.2.: Sichten als Mittel der Wahl zur Dokumentation des Informationsbedarfs

schen Gerichten und Staatsanwaltschaften einen elektronischen Rechtsverkehr aufzubauen. Über eGovernment-Ansätze können dabei auch Externe z.B. Rechtsanwälte über genormte Schnittstellen daran teilhaben.

Dabei befindet man sich entweder in der Rolle des Dienstnutzers, oder der eines Dienstanbieters. Das heißt, man muss nur die Hälfte der Problemstellung lösen: Als Dienstanbieter einen Teil seiner internen Daten in die Transferdarstellung zu bringen und als Dienstnutzer die empfangene Nachricht aus der Transferdarstellung in die eigene Datenwelt zu integrieren.

Der Lösungsansatz in ForumSTAR beinhaltet eine modellbasierte Arbeitsweise, um die für den Datenaustausch nötigen Abstimmungen zielgerichtet durchzuführen. Ein zentrales Repository dient als Informationsquelle für alle Beteiligten, um den aktuellen Diskussionsstand zu bewerten und voranzutreiben. Folgende Unterstützung ist entlang der Konzeption eines Datenaustauschvorhabens zwischen unterschiedlichen Behörden (Anwendungen) mit unterschiedlichen internen Datenmodellen wünschenswert:

- Lesender und schreibender Zugriff aller Beteiligten auf gemeinsame Arbeitsergebnisse, um die Abstimmungszyklen zu verkürzen und jedem stets den aktuellen vereinbarten Stand zugänglich zu machen.
- Die Definition der auszutauschenden Informationen als Sichten in die interne Datenhaltung (Ausschnittsbildung) sollte durch bequeme Auswahl aus dem Bestand passieren und nicht durch Abschreiben der Inhalte.
- Die Abbildung von internen Datenstrukturen auf die XÖV-Nachrichten-Strukturen sollte durch paarweises Verknüpfen von Datenbestandteilen stattfinden, ohne die beiden Seiten manuell in Word-Tabellen abzuschreiben, um sie gegenüberstellen zu können.
- Die qualitätssichernde Prüfung auf XÖV-Konformität sollte automatisiert erfolgen, um mühsame und wiederholt stutzufindende manuelle Arbeit zu vermeiden.
- Sobald Änderungen am XÖV-Standard vorgenommen wurden, sind die Auswirkungen auf die eigenen Nachrichten zu bewerten. Hierzu sollten die Nachrichten

6.3. Effiziente Wege zum Datenaustausch zwischen Behörden

sich nicht automatisch an den neuen Standard anpassen, sondern als Vergleichsgrundlage eine lose Verbindung zu den Standard-Elementen besitzen (Trace).

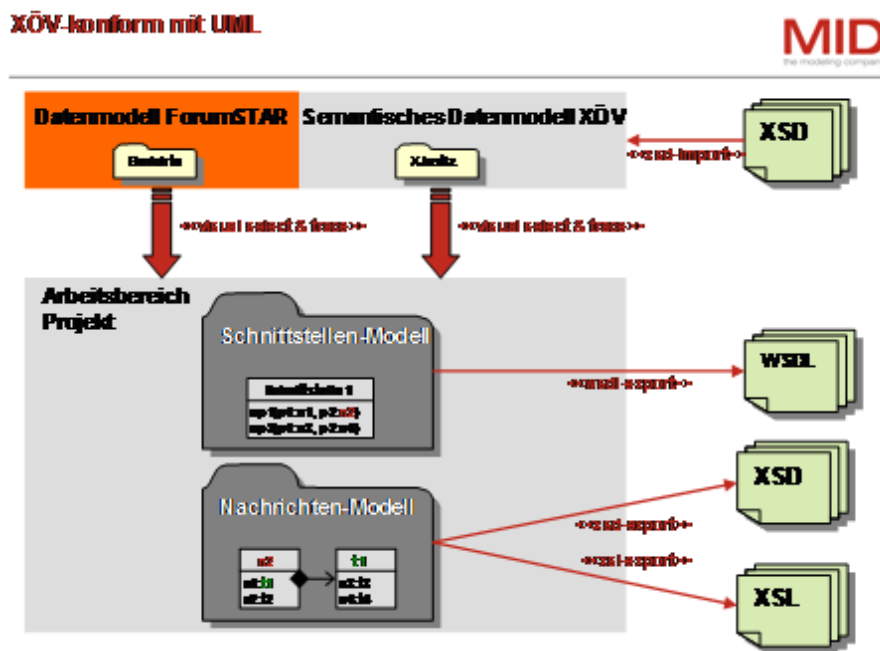


Abbildung 6.3.: Zentrales Repository, Traceability und Visual Select Schnittstellendesign

Die Herangehensweise an die Aufgabe sieht den Einsatz von grafischen Arbeitsmitteln vor: die UML als grafische Syntax zur Visualisierung und grafische Eingabemöglichkeiten, die die Sichtenbildung und Abbildungsregeldefinition unterstützen.

Aus einem so dokumentierten Modell können die Definitionen und die Tracebeziehungen dafür benutzt werden, um WSDL, XSD und XSL-Dateien als Einstieg in die Realisierung zu generieren.

Bewertung

Im Zusammenhang mit einem Datenaustausch, über Domänengrenzen hinweg, ist das Vorhandensein von Standards notwendige Voraussetzung, um gemeinsames Wissen und Verständnis aufzubauen. Effizienzsteigernd hat sich dabei der Einsatz von modellbasierter Vorgehensweise bewährt. Die grafische Visualisierung der Konzepte liefert deutlich schneller belastbare Ergebnisse und minimiert den Abstimmungsaufwand unter den Beteiligten.

Die fachliche Spezifikation und Abstimmung der Austausch Inhalte werden damit optimal unterstützt. Im Rückblick gab es bei dem Vorgehen auch positives Feedback von unerwarteter Seite.

Entwickler sehen in den Generatoren den großen Vorteil, fachliche Änderungen in ihrer Welt identifizieren zu können. Die in Textform gelieferten Spezifikationen (XSD, WSDL, XSLT) lassen sich sehr gut mit Vorgängerständen vergleichen und somit vermeidet man den Vergleich eines UML-Modells mit der Realisierung, was oft manuell durchgeführt werden muss.

Um den oben beschriebenen Anforderungen noch besser gerecht zu werden, sind im Werkzeug Innovator⁵ noch folgende Erweiterungen umzusetzen:

- Eine visuelle Eingabehilfe für die Spezifikation von Übersetzungsregeln aus der Transfersprache in die interne Sprache eines IT-Systems.
- Ein XSLT-Generator, der die Übersetzungsregeln in einen Entwurf für eine XSL-Transformation auf Nachrichten-Telegrammen umsetzt.

Der Vortrag führt durch die beschriebene Thematik eines verbesserten gemeinsamen Datenaustausches via Nutzung von UML-Modellen und endet mit einer Kurzpräsentation des Durchstichs, ausgehend von drei Domänenmodellen hin zu den generierten XSD-Dateien der ausgetauschten Nachrichten.

⁵ Siehe: <http://innovator.mid.de>



Effiziente Wege zum Datenaustausch zwischen Behörden

XÖV-Konformität mit UML

Berlin, SEE 2009, 26.05.09

INHALT

1. **Motivation für Datenaustausch**
2. **Modellbasierter Lösungsansatz**
3. **Demo eines UML-Modells**
4. **Zusammenfassung**



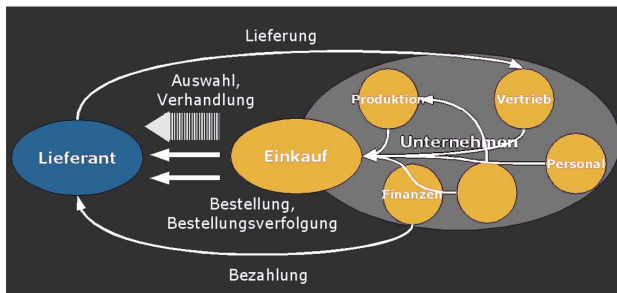
XÖV-Konformität mit UML

Motivation für Datenaustausch

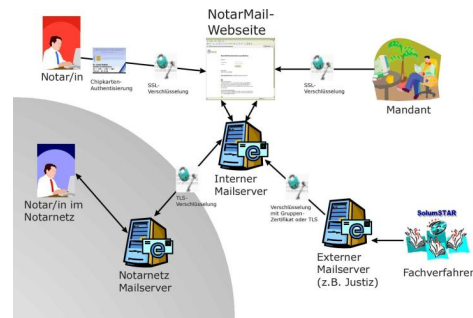


Datenaustausch ist im Zeitalter des eGovernments und eBusiness ein alltäglicher Vorgang

- Bedarf für Informationsaustausch ist vorhanden (Gesetzgeber, Unternehmenskooperationen)
- Informationsaustausch erfordert eine gemeinsame Sprache
- Standards helfen die Anzahl der Übersetzungsvorgänge zu minimieren und Fälle von Fehlinterpretation zu reduzieren



eProcurement



eJustiz

© 2009 MID GmbH

3

XÖV-Konformität mit UML

Motivation für modellbasierte Spezifikation des Datenaustauschs



Eine modellbasierte Arbeitsweise beschleunigt die Spezifikation eines geregelten Datenaustauschs, indem sie die Abstimmzyklen verkürzt

- Ein Modell schafft Transparenz, indem der aktuelle Diskussionsstand stets allen Beteiligten zugänglich ist.
- Eine zentrale Modellinstanz erübrigt die Konsolidierung von dezentral vorbereiteten Zwischenständen.
- Durch eine Aufbereitung der Anforderungen, als Modell, lassen sich formale Prüfungen automatisieren.
- Differenzen zwischen unterschiedlichen Versionen eines Modells können schnell nachvollzogen werden (automatische Differenzenbildung).

© 2009 MID GmbH

4

XÖV-Konformität mit UML

Ausgangssituation für XÖV-Einsatz

Etablierte Begrifflichkeiten bei beiden Kommunikationspartnern führen zu Verständnisschwierigkeiten

- Kommunikationspartner haben unterschiedliche Glossare.
 - Standards bieten Brückenschlagsfunktion
 - XML in der Öffentlichen Verwaltung (XÖV) ist ein Standard für den Datenaustausch zwischen Behörden. Der Einsatz von XÖV ist gesetzlich vorgeschrieben
- Bestehende IT-Lösungen auf beiden Seiten müssen ‚kommunikationsfähig‘ gemacht werden.

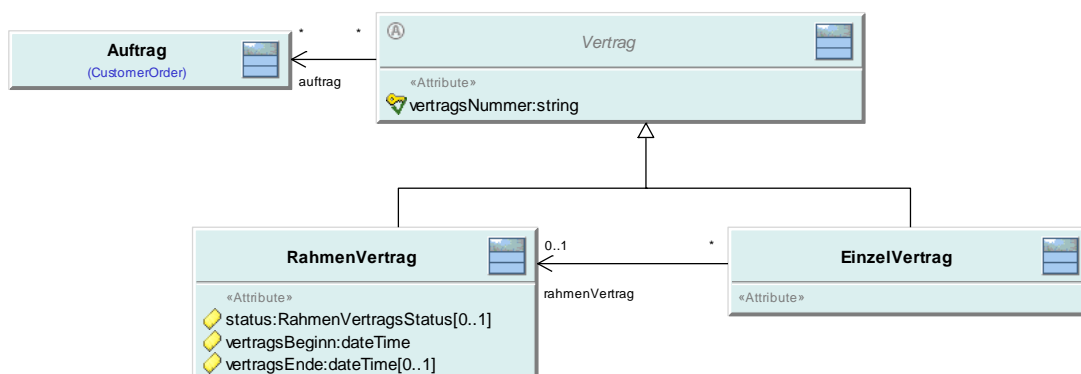


XÖV-Konformität mit UML

Lösungsansatz zur modellbasierten Arbeit

Glossare lassen sich als UML-Klassenmodelle gut visualisieren.

- Meist übernimmt im elektronischen Datenaustausch ein Kommunikationspartner die führende Rolle und bietet Information zum Abruf an: der Dienstanbieter.
- Der Informationsbedarf und das Informationsangebot lassen sich via XÖV allgemein verständlich vereinbaren.
- Aus Gründen des Datenschutz, Kommunikations-Volumen-Begrenzung und zur Minimierung der Erklärungsbedürftigkeit von Nachrichteninhalten werden Ausschnitte definiert: die Sichten.



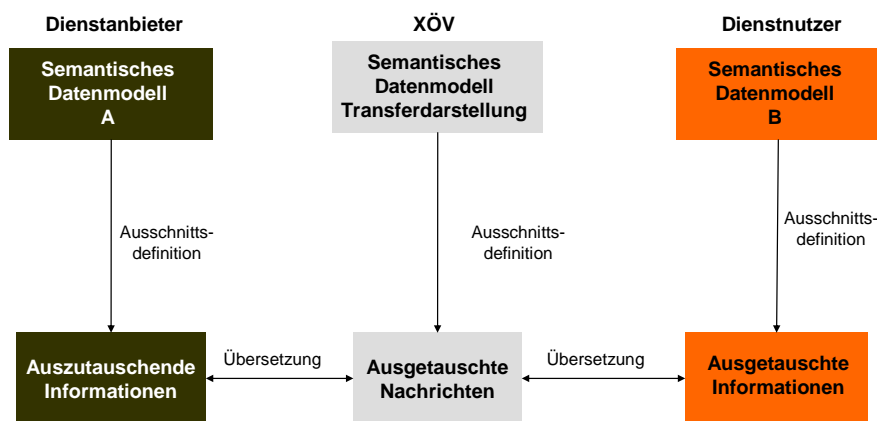
XÖV-Konformität mit UML

Sichten in Bezug zu Glossaren



Nachrichten lassen sich mit vorgegebenen Glossaren per Drag'n'Drop gestalten.

- Die Kommunikationspartner vereinbaren gemeinsam auf Basis des XÖV-Datenmodells die relevanten Kommunikationsinhalte in Form von Sichten.
- Jeder Kommunikationspartner definiert daraufhin
 - die entsprechende Sicht in sein Datenmodell
 - Die Übersetzungsregeln von der XÖV-Sicht in seine Glossar-Sicht.



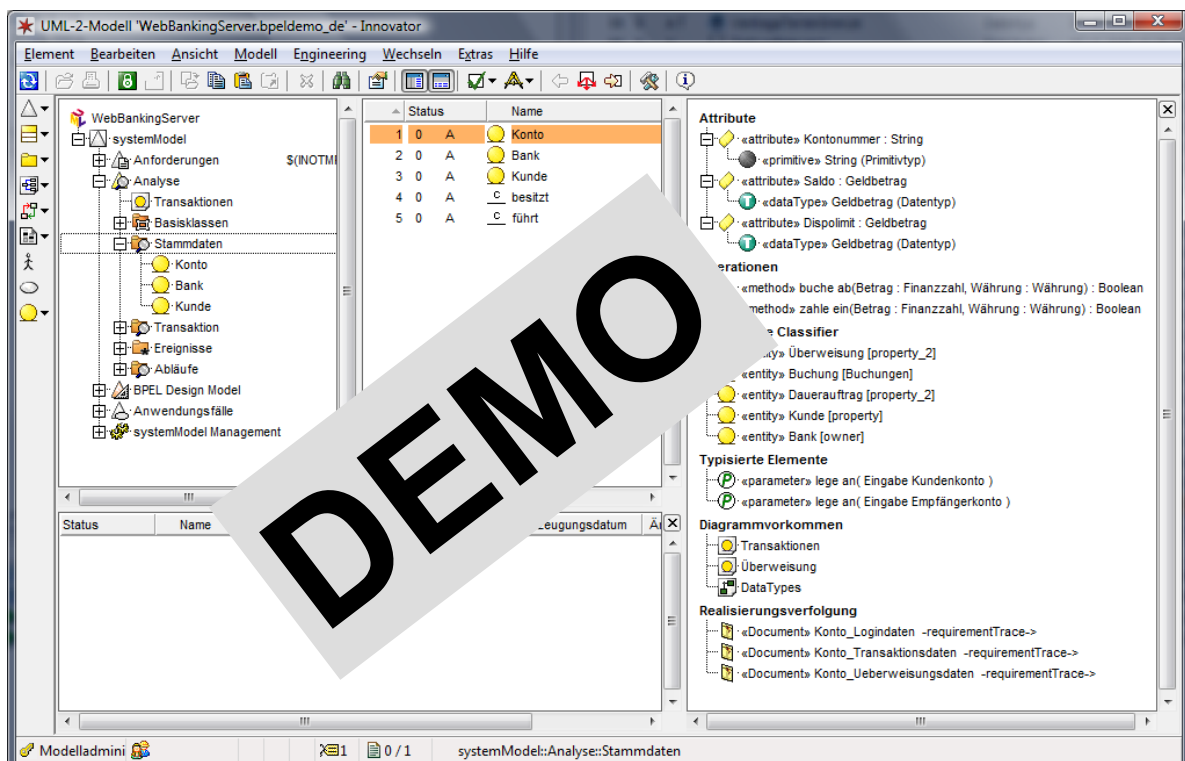
Sichten als Mittel der Wahl zur Dokumentation des Informationsbedarfs

© 2009 MID GmbH

7

XÖV-Konformität mit UML

Demo eines UML-Modells



© 2009 MID GmbH

8

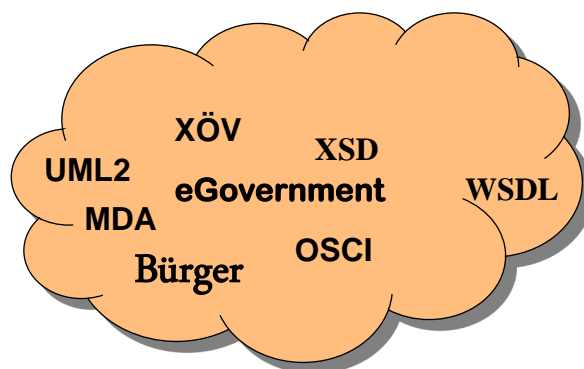
XÖV-Konformität mit UML

Zusammenfassung



Die grafische Visualisierung der Konzepte liefert deutlich schneller belastbare Ergebnisse

- Transparent, zentral und grafisch publizierte Glossare als UML-Klassenmodelle beschleunigen die Abstimmprozesse
- Eine modellbasierte Konzeption der Nachrichten per Drag'n'Drop beschleunigt die fachliche Artikulation
- Eine modellbasierte Konzeption der Übersetzungsregeln für ein- und ausgehende Nachrichten sichert die Konsistenz mit den unterschiedlichen Glossarversionen



© 2009 MID GmbH

9



Alexander Bösl

a.boesl@mid.de

Klaus Weber

k.weber@mid.de

Hauptverwaltung Nürnberg
Eibacher Hauptstraße 141
90451 Nürnberg
Telefon: +49 (0)911 - 968 36-0
Telefax: +49 (0)911 - 968 36-10

[**www.mid.de**](http://www.mid.de)

II.
Tag 2

7. Session 7: Anpassung und Einführung von Vorgehensmodellen

Inhalt

7.1. flyXT – Das neue Vorgehensmodell der EADS DE	262
7.2. Studie für die Einführung HERMES PowerUser in der PostFinance . .	276
7.3. Projekt Recovery nach dem War Room Konzept	293

7.1. flyXT – Das neue Vorgehensmodell der EADS DE

Doris Rauh¹, Wolfgang Kranz²

¹Siemens AG
Corporate Technology (CT)
Otto-Hahn-Ring 6
81739 München
doris.rauh@siemens.com

²EADS Deutschland GmbH
Defence Electronics (DE)
Landshuterstr. 26
85716 Unterschleißheim
wolfgang.kranz@eads.com

Abstract

EADS DE wendet schon heute in den Entwicklungsprojekten ein integriertes Systementwicklungsmodell (bezeichnet als Vorgehensmodell im Geschäftsbereich Verteidigung VM-GBV) im Rahmen seines Geschäftssystems an. Im Projekt flyXT wird das VM-GBV nun durch den aus dem Standard V-Modell XT abgeleiteten Prozess flyXT ersetzt. Dieser ergänzt das V-Modell XT um best practices von EADS DE sowie neue Anforderungen des Marktes. Der Vortrag beschreibt das Projekt flyXT, seine Ziele und das Vorgehen zur Implementierung des neuen Vorgehensmodells. Aus der Implementierungsphase werden erste Schlüsse für die Lessons learnt gezogen.

Motivation

EADS Defence Electronics (EADS DE) ist ein Unternehmensteil der EADS, welcher Radar-, EW- und Avionik-Systeme entwickelt und produziert. Diese Produkte bestehen im wesentlichen aus einer Kombination von Software, Hardware und logistischen Elementen. Sie setzen einen entsprechend integrierten Systementwicklungsprozess voraus. Deswegen wendet EADS DE schon heute in den Entwicklungsprojekten ein integriertes Systementwicklungsmodell (VM-GBV) im Rahmen seines Geschäftssystems an. Als Basis für das VM-GBV wurde das V-Modell 97 verwendet (Abbildung 1).

Nachdem mit dem V-Modell@XT [1] wesentliche Verbesserungen gegenüber dem V-Modell 97 erfolgten, erschien auch eine Überarbeitung des Hausstandards VM-GBV zweckmäßig. Zudem hatte sich aus der Anwendung des VM-GBV und dem kontinuierlichen Verbesserungsprozess bei EADS DE weiterer Handlungsbedarf ergeben. Deshalb wurde das Projekt flyXT aufgesetzt, in welchem das bestehende VM-GBV durch den V-Modell XT basierten Prozess flyXT ersetzt werden soll. Im folgenden werden das Projekt und erste Projektergebnisse vorgestellt.

Projekt & Projektziele

Am Anfang des Projektes flyXT stand zunächst eine Analyse des Status quo und der Anforderungen an einen modernisierten Entwicklungsstandard innerhalb der EADS DE.

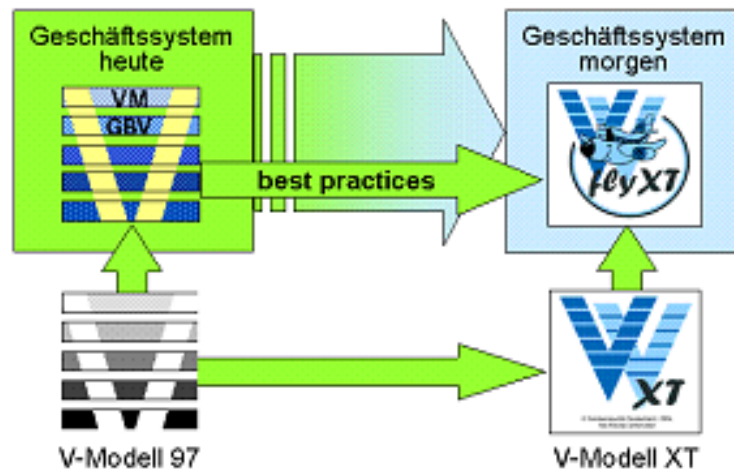


Abbildung 7.1.: Geschäftssystem der EADS DE

Zum einen waren aufgrund des veränderten Umfelds Anforderungen von Luftfahrtstandards wie Airbus- ABD0100 und ABD0200 oder DO-178B und DO-254 zu beachten. Daraus folgten neue Vorgaben für die Verifikation und Validierung sowie die Zertifizierung der Entwicklungsergebnisse. Diese Ziel-Vorgaben ließen sich ohne Überarbeitung des Requirements- und Systems-Engineering nicht erreichen. Zum anderen stand der Wunsch nach einfacherer Anwendbarkeit des Entwicklungsstandards im Vordergrund. Das Ziel war hier insbesondere das vereinfachte Tailoring mit dem Werkzeug „V-Modell XT Projektassistent“ zur Anpassung von flyXT an die spezifischen Erfordernisse der Projekte.

Vorgehen

Für die Überarbeitung des Entwicklungsprozesses gab es feste Randparameter:

- Übernahme der bewährten Vorgehensweisen und Begrifflichkeiten des bestehenden Geschäftssystems (vgl. Abbildung 1),
- Klare Schnittstellen zu anderen Prozessen, wie z.B. dem Akquisitionsprozess,
- Die Festlegungen des Geschäftssystems sollten der Master sein.

Bei der Modellierung des Entwicklungsprozesses wurde auf der Version 1.2.1.1 des V-Modell®XT und der zugehörigen Werkzeuge aufgesetzt. Die Erfahrungen bei der Entwicklung des V-Modells haben gezeigt, dass ein stufenweises Vorgehen bei der Definition eines neuen Prozesses große Vorteile bietet. Analog zu dem Vorgehen im Projekt WEIT wurde daher ein Stufenmodell festgelegt.

- Stufe 1: Festlegung des Produktmodells (z.B. Dokumente, SW oder HW)
- Stufe 2: Definition der Themenstruktur (Kapitelstruktur der Dokumente),
- Stufe 3: Definition der Beschreibungstexte.

Parallel zur Definition des Produktmodells wurde ein Mapping der EADS DE Rollen auf die Rollen des V-Modells durchgeführt. Die Projekttypen des V-Modells wurden alle, bis auf den Projekttyp „Systementwicklungsprojekt (AG)“ übernommen. Die Vorgehensbausteinstrukturen wurden im wesentlichen beibehalten und durch drei neue Vorge-

7.1. flyXT – Das neue Vorgehensmodell der EADS DE

hensbausteine ergänzt. Das Meilensteinmodell der EADS DE konnte mit den Entscheidungspunkten des V-Modells harmonisiert werden.

Die Einhaltung der Konformität zum V-Modell XT war eine der wesentlichen Anforderungen an flyXT. Durch eine stufenweise Ersetzung der Inhalte des V-Modells mit den Inhalten der EADS DE Prozesse und die Abbildung der jeweiligen Inhalte aufeinander soll die Konformität von flyXT zum V-Modell erreicht werden.

Lessons learnt

Durch das Aufsetzen auf der aktuellen Version des V-Modells wurde die Definition des neuen Prozesses flyXT wesentlich vereinfacht. Die Strukturen und die Vernetzung der verschiedenen Objekte konnten in vielen Fällen übernommen werden. Ein großes Plus des V-Modells ist die Möglichkeit von internen Links. Auf diese Weise werden Namensänderungen über das ganze Modell konsistent durchgeführt. Die Möglichkeiten der Verlinkung mit externen Quellen vereinfachte die Kopplung mit den bereits bestehenden Prozessen im Geschäftssystem. Das V-Modell ermöglicht eine klare Definition von Schnittstellen zwischen verschiedenen Prozessen. Dadurch wird eine bessere Einbeziehung von Disziplinen wie z.B. Logistik, Zertifizierung oder Safety in den Systementwurf gewährleistet.

Ausblick

Letztlich wird das V-Modell XT, egal ob in seinem Original oder als modifizierter Hausstandard flyXT, nur erfolgreich sein, wenn die Integration in die IT-Landschaft der anwendenden Unternehmen gelingt. Die Modellierungstechnik des V-Modells bietet hier eine Menge Raum für Innovationen. Zum Beispiel lässt sich durch konsequente Anbindung an die Projektmanagement-Werkzeuge eine geschlossene Werkzeugunterstützung von der Angebotsphase bis zur Projektplanung nach Auftragseingang erreichen. Nach der hier beschriebenen Prozessdefinitionsphase folgt die für den Erfolg ganz entscheidende Einführungsphase, mit Veröffentlichung im Intranet, Pilotierung und Coaching in neuen Projekten sowie intensiver Schulung der betroffenen Mitarbeiter.

Literaturverzeichnis

1. V-Modell XT, <http://www.v-modell-xt.de>, Koordinierungs- und Beratungsstelle der Bundesregierung, 2008

Defence Electronics Operations



flyXT - das neue Vorgehensmodell der EADS/DE

Software & Systems Engineering Essentials - Berlin
27.05.2009

Doris Rauh
Siemens, CT SE 3
doris.rauh@siemens.com
Tel.: 089 636 53393

Wolfgang Kranz
EADS DE
wolfgang.kranz@eads.com
Tel.: 089 3179-2786

© EADS

Page 1 flyXT - das neue Vorgehensmodell der EADS/DE, SEE 27.05.2009

Defence Electronics Operations

EADS and Defence & Security Systems Division

Revenues in € bn: 39.4
Employees: 116,805

Divisions

 Airbus	 Military Transport Aircraft	 Eurocopter	 EADS Astrium	 Defence & Security
--	---	--	--	---

Stefan Zoller

Revenues in € bn: 5.9
Employees: 23,000

Business Units

 Military Aircraft B. Gerwert	 Defence and Communications Systems H. Guillou	 Defence Electronics/ Test & Services B. Wenzler	 MBDA A. Bouvier	 Eurofighter GmbH A. Rauen
---	--	--	--	---

EADS Defence Electronics designs sensors, avionics and electronic warfare equipment

all figures as of end 2006

© EADS

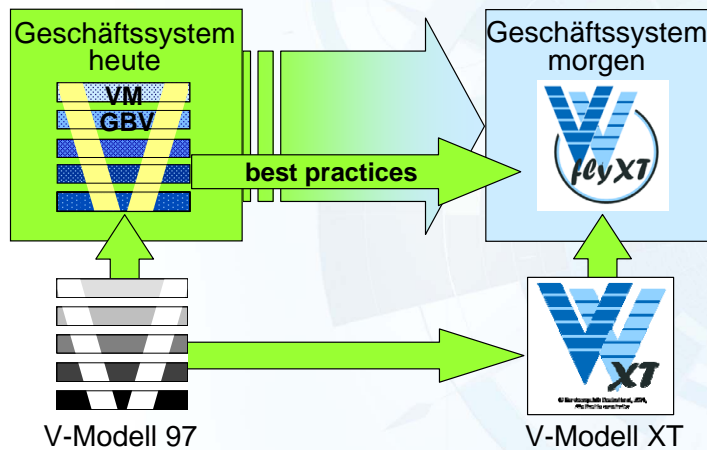
Page 2 flyXT - das neue Vorgehensmodell der EADS/DE, SEE 27.05.2009

Agenda

- ❑ Motivation
- ❑ Projekt & Projektziele
- ❑ Vorgehen
- ❑ Lessons learnt
- ❑ Ausblick

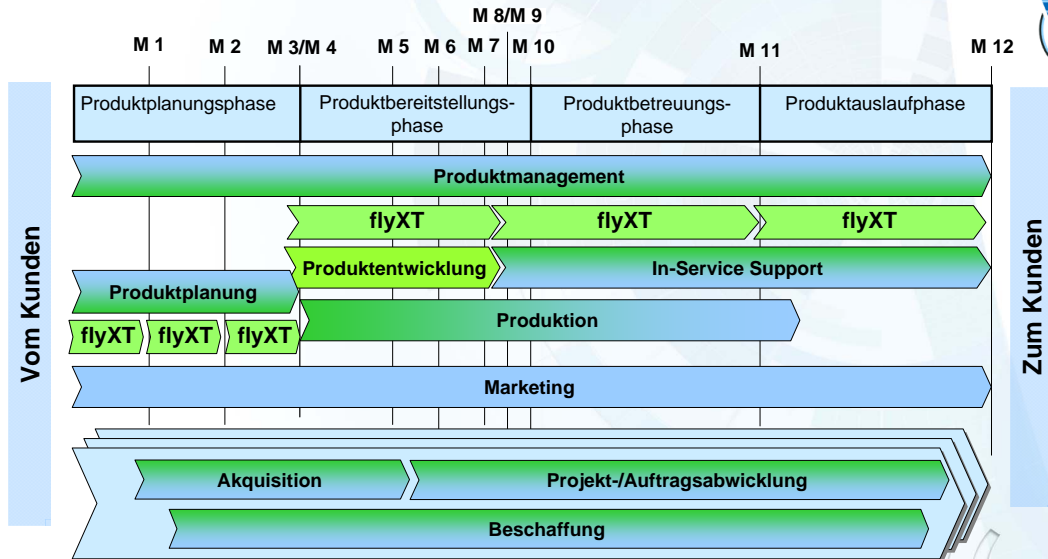


Motivation



- ❑ **Bestehendes Geschäftssystem enthält Entwicklungsprozess auf der Basis des V-Modells 97 (Vorgehensmodell GBV, VM-GBV)**
- ❑ **VM-GBV ist nicht mehr auf dem Stand der Technik**

Ausgangsbasis Geschäftsprozesse bei EADS/DE



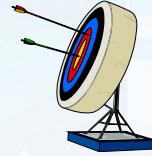
Agenda

- Motivation
- Projekt & Projektziele
- Vorgehen
- Lessons learnt
- Ausblick



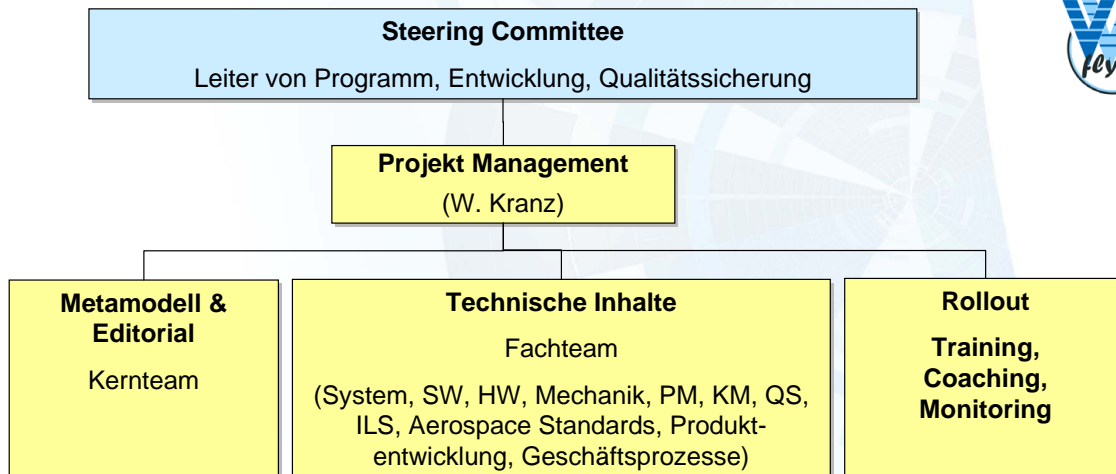
Projektziele flyXT:

- ❑ Ersatz des VM-GBV durch das V-Modell XT
- ❑ Übernahme der best practices aus VM-GBV
- ❑ Einarbeitung neuer Anforderungen
 - ◆ Luftfahrtstandards (ABD100/200, DO-178B, DO-254)
 - ◆ Certification, Verifikation & Validierung
 - ◆ Überarbeitung des Requirements- und Systems-Engineering
- ❑ Breite Anwendung des Haus-Entwicklungsstandards durch
 - ◆ Vereinfachung der Anwendbarkeit in den Projekten
 - ◆ Bessere Anbindung an interne Projektmanagement-Werkzeuge
 - ◆ Schulung, Coaching und Monitoring
- ❑ Unterstützung der Angebots-, Projektstart- und Projektplanungsphase
- ❑ Möglichst frühe Anwendbarkeit in Entwicklungsprojekten



Verbesserung der Projekte & der Projekt-Initialisierungsphase

Organisation & Ressourcen



Unterstützung durch oberes Management im Steering Committee

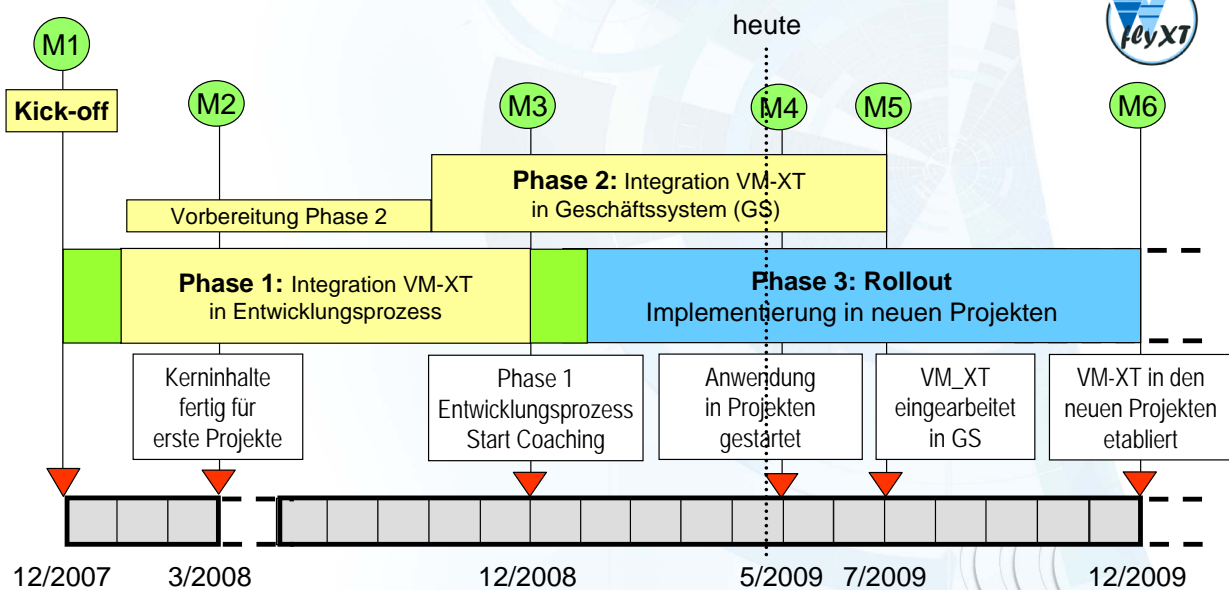
Projektstrategie flyXT



- Durchführung als Standard-Projekt
- Phase 1: Implementierung im Produktentwicklungsprozess
 - ◆ Ersatz von VM-GBV durch VM-XT (übertrage best practices des VM-GBV auf V-Modell XT)
 - ◆ Adaption der Entwicklungsprozess-Dokumentation
 - ◆ Integration der Luftfahrtregelungen
- Phase 2: Integration in das EADS/DE Geschäftssystem
 - Harmonisierung des flyXT mit den Geschäftsprozessen
- Phase 3: Rollout
 - ◆ Training und Coaching in neuen Projekten
 - ◆ Monitoring der Anwendung

Einbeziehung der Experten aller Disziplinen

flyXT Roadmap – Meilensteine



7.1. flyXT – Das neue Vorgehensmodell der EADS DE

Defence Electronics Operations

Agenda

- Motivation
- Projekt & Projektziele
- Vorgehen
- Lessons learnt
- Ausblick



© EADS

Page 11 flyXT - das neue Vorgehensmodell der EADS/DE, SEE 27.05.2009

Defence Electronics Operations

Stufenweises Vorgehen

Ausgangsbasis: Geschäftssystem und V-Modell XT Version 1.2.1.1

1. Grobstruktur



Produkte
Aktivitäten



Rollen



Entscheidungspunkte
und Projektdurch-
führungsstrategien



Projekttypen
Vorgehensbausteine

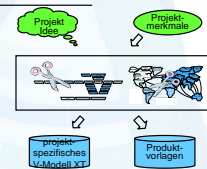


2. Feinstruktur

Gliederg.

1- blablablabilia.....2
2. nochmalblabla.....5
3. und wieder bla.....10
4. das letzte Mal bla.....15

Themen (Kap 1)
Unterthemen (Kap. 1.1)
U/U-Themen (Kap. 1.1.1)
Teilaktivitäten



Projektmerkmale

3. Inhalte

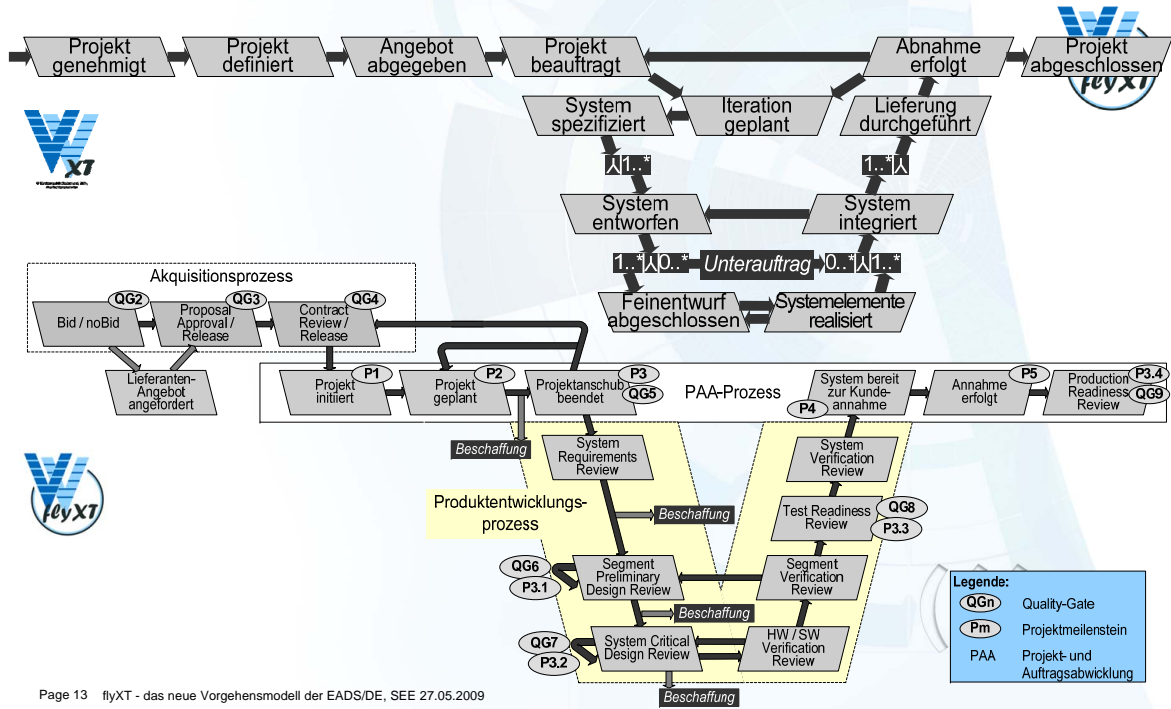
Text

e layout of the PrintPC 6011...
The outline of the Pg to...
The maximum allowing to
Areas where no plating is...
The placement of tiding to...
The top or bottom id PCB...
The board surface on it...
- Manufacturer (Echland)
- Part No. (TKZ)
The board surface on it...
- Manufacturer (Echland)

© EADS

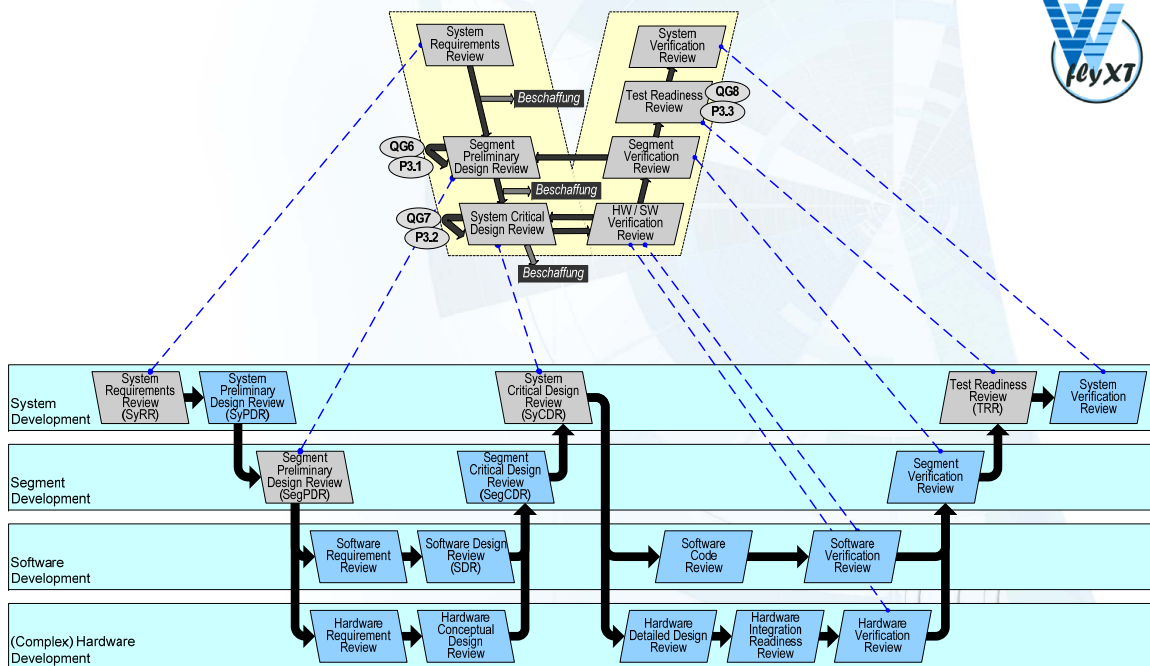
Page 12 flyXT - das neue Vorgehensmodell der EADS/DE, SEE 27.05.2009

Inkrementelle Entwicklung V-Modell XT - flyXT



© EADS

Entwicklungsprozess im Detail



© EADS

Unterschiede zum V-Modell XT Standard



Version 1.3

□ Detailliertere Beschreibungen

- ◆ Einführung von Unterthemen (6 Ebenen)
- ◆ Tabellen und Bilder in Produktbeschreibungen



□ Geschäftssystem als Master bedingt

- ◆ Anpassung der Begriffe
- ◆ Unterschiede in den Produkten (Templates)
- ◆ Verwendung des EADS/DE-spezifischen Rollenmodells
- ◆ Erweiterte Projektdurchführungsstrategien

□ Branchenspezifische Anforderungen (z.B. aus Avionik) bedingen Veränderungen im Tailoring

Realisierung durch neue Vorgehensbausteine, z.B.

- ◆ Luftfahrtzulassung
- ◆ HDL-Entwicklung: für die Entwicklung komplexer HW (FPGAs und ASICs)
- ◆ SW-Entwicklung mit DO-178B: Unterschiedliche Meilensteine & Templates

HDL - HW Design Language

Konformität zu Standards



□ V-Modell XT

Mapping der Inhalte des V-Modell XT auf die Inhalte von flyXT parallel zur Definition der neuen Strukturen und stufenweises Ersetzen der Inhalte des Standards durch die neuen Inhalte von flyXT

□ CMMI

- ◆ Erstellung von PIIDs (Process Implementation Indicator Descriptions)
 - ✦ die die typischen Produkte des flyXT enthalten
 - ✦ Strukturierung in direkte und indirekte Artefakte

□ Luftfahrtstandards (DO-178B, DO-254)

- ◆ Einhaltung der Vorgaben für Verifikation und Validierung
- ◆ Berücksichtigung notwendiger Zertifizierungen

➡ **Frühzeitiges Erkennen von Lücken**

Defence Electronics Operations

Agenda

- Motivation
- Projekt & Projektziele
- Vorgehen
- Lessons learnt
- Ausblick



© EADS

Page 17 flyXT - das neue Vorgehensmodell der EADS/DE, SEE 27.05.2009

Defence Electronics Operations

Coaching

- Start in mehreren neuen Projekten**
 - ◆ mit originalem V-Modell XT
 - ◆ mit flyXT
- Vorgehen**
 - ◆ Tailoring-Workshop in der Angebots-/Projektstartphase
 - ✦ „Grundkurs“ V-Modell XT
 - ✦ Durchführung des Tailorings im Projektteam
 - ◆ Workshop-Teilnehmer:
Bid-Manager, Projektleiter, Systemingenieur, QM-Verantwortlicher, KM-Verantwortlicher, ILS-Manager, SW/HW-Entwickler
 - ◆ Workshop-Ergebnisse innerhalb eines Tages:
 - ✦ Entwicklung einer Produktstruktur nach V-Modell XT
 - ✦ Export einer Grundstruktur für die Projektplanung
 - ✦ Generierung von projektspezifischen Templates mit einheitlichem Layout
 - ◆ Nacharbeit mit Definition der Entwicklungsdokumentation



© EADS

Page 18 flyXT - das neue Vorgehensmodell der EADS/DE, SEE 27.05.2009

Lessons learnt

❑ Modellierung des V-Modell XT

- ☺ ermöglicht projektspezifische Anpassbarkeit an verschiedene Standards
- ☺ ermöglicht geschlossene Toolkette in Angebots-/Projektstartphase
- ☺ unterstützt die Anwender durch präzisere Definition und Abstimmung der Prozesse
- ☹ führt zu erhöhtem Abstimmungsaufwand bei der Prozessentwicklung
- ☹ erhöht die Seitenzahl der Prozessbeschreibung

❑ Elektronische Dokumentation unterstützt die

- ☺ Verwendung von internen und externen Links (z.B. auf vorhandene Dokumente)
- ☺ einfache Kopplung mit den Geschäftsprozessen (keine redundanten Beschreibungen)

❑ Feedback aus dem Coaching

- ☹ Original V-Modell XT sehr generisch
- ☺ Anpassung mit flyXT unterstützt Anwender besser (Begriffe, Praktiken, ..)
- ☺ Schnelle Erarbeitung von Strukturen für das Projekt
- ☹ Umfang der Prozessdokumentation (>800 Seiten) erschreckt Anwender



Agenda

- ❑ Motivation
- ❑ Projekt & Projektziele
- ❑ Vorgehen
- ❑ Lessons learnt
- ❑ Ausblick





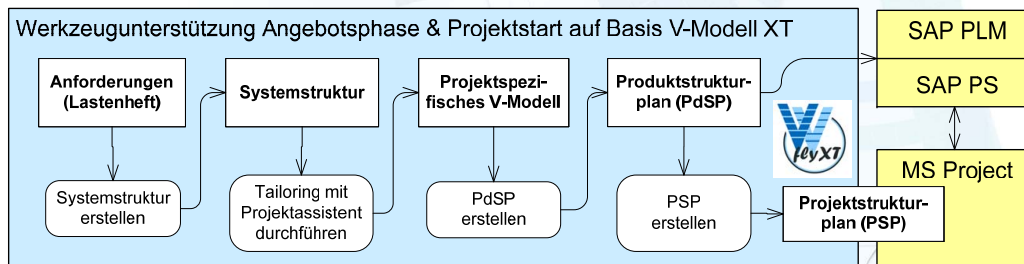
Ausblick: Workflow der Projektstrukturierung

□ Bisher:

- ♦ intensives Wissen über Regelungsteil und seine Bestandteile nötig
- ♦ aufwendiges Tailoring-Verfahren im Projekthandbuch

□ Neu im VM-XT:

- ♦ Tailoring-Wissen des Prozessingenieurs wurde ins V-Modell verlagert
- ♦ Bausteinbasiertes Tailoring anhand von vorgefertigten Bauplänen
- ♦ Modellierung (xml) des V-Modell XT macht Toolunterstützung möglich



PdSP = Produktstrukturplan, PSP = Projektstrukturplan



Danke
für Ihre
Aufmerksamkeit

7.2. Studie für die Einführung HERMES PowerUser in der PostFinance

Hans Peter Schaufelberger, Martin Wenger

Die Schweizerische Post
PostFinance
Nordring 4a
3030 Bern

Hanspeter.Schaufelberger@PostFinance.ch
Martin.Wenger@APP.ch

Abstract

Die PostFinance der Schweizerischen Post setzt HERMES als Projektführungsmethode in angepasster Form erfolgreich ein. Die Pflege und Weiterentwicklung der, auf das Unternehmen tailorierten, Methode sowie der Einsatz in den Projekten basiert auf Standard-Werkzeugen. In der Studie wird untersucht, ob HERMES PowerUser, das offizielle Werkzeug der Schweizerischen Bundesverwaltung, als Methodenrepository in der Postfinance eingesetzt werden kann. Das Studienergebnis hat gezeigt, dass HERMES PowerUser für den Methodenbau und die Methodenpflege auf jeden Fall nutzenstiftend ist und der Einsatz von HERMES PowerUser angestrebt wird.

Ausgangslage

HERMES wird in der PostFinance in angepasster Form dem sog. Hermes-PF (Unternehmenstailoring) seit 1997 angewendet und basiert auf dem HERMES Standard [2].

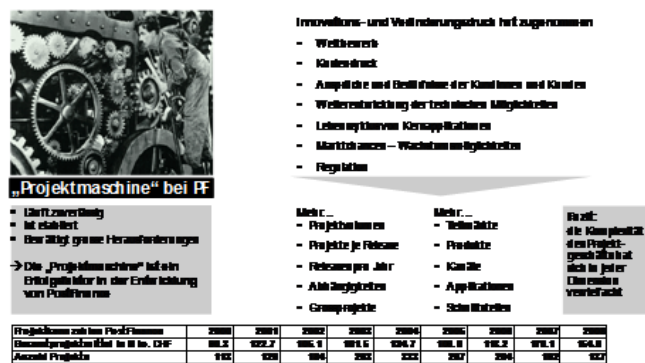


Abbildung 7.2.: Fakten zu Hermes-PF

Mit der Projektführungsmethode Hermes-PF werden 137 Projekte mit einem Budget von rund 155.– Mio. CHF pro Jahr (Stand 2008) durchgeführt (siehe Abbildung 7.2). Hermes-PF wurde mit der Ausgabe 2008 aktualisiert. Mit dem PowerUser [3] wird ein Werkzeug zur Verfügung gestellt, das sowohl für die Projektmitarbeitenden als auch für Methodiker genutzt werden kann (siehe Abbildung 7.3). Das Metamodell des PowerUser beruht auf SPEM [4] und die Architektur basiert auf der Eclipse-Plattform ergänzt mit Eclipse-Process-Framework [1] (siehe Abbildung 7.4).

7. Session 7: Anpassung und Einführung von Vorgehensmodellen



Abbildung 7.3.: Funktionsumfang HERMES PowerUser



Abbildung 7.4.: Architektur HERMES PowerUser

Fragestellung und Ziele

In der Studie ging es um das Schaffen von Entscheidungsgrundlagen für den Einsatz des PowerUsers in der PostFinance. Es war aufzuzeigen, welcher Nutzen durch den Einsatz des HERMES PowerUser für die Projektbeteiligten (i. B. Projektleiter) und für die Methodiker (Fachführung Hermes PF) entsteht.

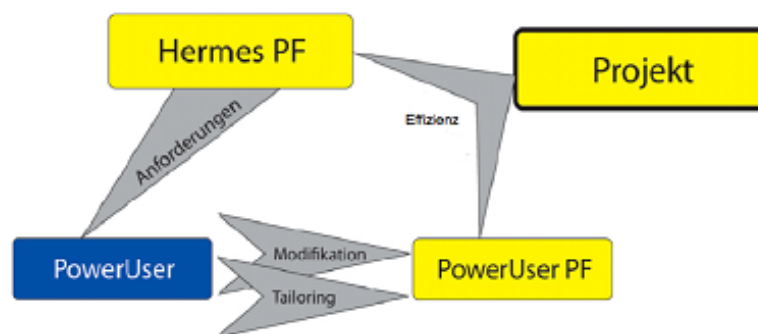


Abbildung 7.5.: Einsatz PowerUser

Die Studie wurde in enger Zusammenarbeit mit der Berner Fachhochschule und dem Methodikerteam der PostFinance durchgeführt. Die Studie zeigte zudem die Planung des weiteren Vorgehens für eine allfällige Einführung und den Einsatz des HERMES PowerUser in der PostFinance auf.

Studiendesign

In der Studie ging es darum, den Einsatz von HERMES PowerUser als Methodenrepository in der PostFinance zu untersuchen. Einerseits sollten die Grundlagen von HERMES Bund (PowerUser und Methode) für PostFinance nutzbar sein und andererseits die Ausgabe Hermes-PF nutzbringend abgebildet und eingesetzt werden.

Lösungsansatz

Die Variantenbildung erfolgte anhand eines morphologischen Kastens in drei Dimensionen mit je zwei bis drei Ausprägungen. Nicht alle Kombinationen sind sinnvoll.

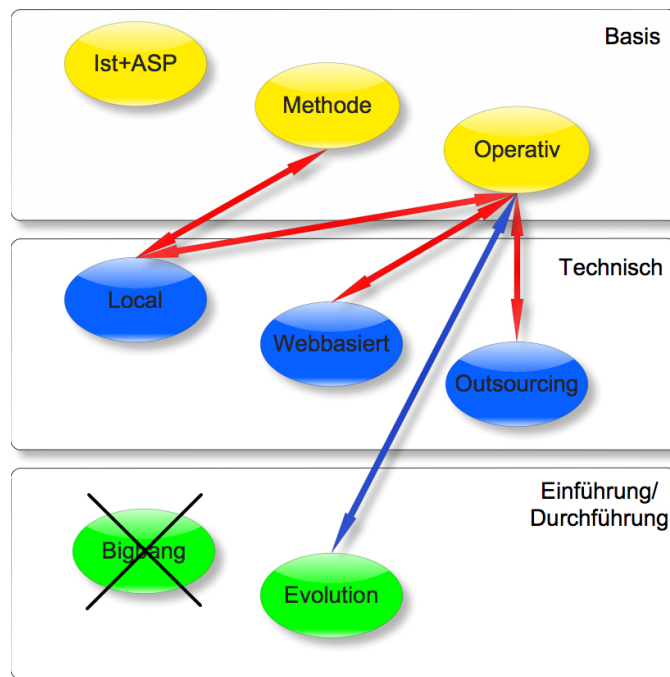


Abbildung 7.6.: Variantenbildung

Untersucht wurden schliesslich folgende Varianten: Varianten Zusammengesetzt aus:

Varianten	Zusammengesetzt aus:		
	Basis	Technisch	Einführung
Variante 0	Ist + ASP	Local	-
Variante 1	Methode	Local	-
Variante 2	Operativ	Local	Evolution
Variante 3	Operativ	Webbasiert	Evolution
Variante 4	Operativ	Outsourcing	Evolution

Abbildung 7.7.: Lösungsvarianten

In der Dimension „Basis“ bedeutet die Ausprägung „Methode“ den Einsatz des PowerUsers nur für den Methodenbau, ein Einsatz für Projektleiter ist hier nicht vorgesehen,

7. Session 7: Anpassung und Einführung von Vorgehensmodellen

ganz im Gegensatz zu „Operativ“, die genau dies vorsieht. Die technischen Ausprägungen sind eine lokale Installation des PowerUsers („Local“), einen Zugriff via Web - allenfalls auch als Extranet für Partner von PostFinance - („Webbasiert“), oder ein Betrieb durch Dritte („Outsourcing“). Da eine Big-Bang-Einführung aufgrund unternehmensinterner Gegebenheiten nicht in Frage kommt, bleibt es bei den operativen Varianten bei der evolutiven Einführung.

Kriterien

Die Lösung hat folgende Anforderungen zu erfüllen:

<i>NFID</i>	Nicht funktionale Anforderungen	<i>FID</i>	Funktionale Anforderungen
1	Projektleitung leicht automatisiert anwendbar	1	Projektplan integrierbar
2	leicht zugänglich	2	PF Instrument leicht anbindbar
3	leicht bedienbar	3	in Produktiv-Server integrierbar
4	intuitiv	4	Template Dokumente aktualisierbar
5	Zeitsparend durch automatisierte Abläufe	5	Windows & Linux tauglich
		6	Vista kompatibel
		7	erweiterbar
		8	updatebar
		9	läuft stabil

Abbildung 7.8.: Anforderungen an die Lösung

Die Wirtschaftlichkeit der Varianten wurde anhand einer Barwertbetrachtung untersucht.

Studienergebnis

Die Studie hat ergeben, dass für PostFinance schon die Variante 1 mit der Beschränkung auf den Einsatz im Methodenbau nutzenstiftend ist und sich rechnet. Eine wirkliche Hebelwirkung entsteht allerdings erst, wenn PowerUser auch operativ eingesetzt wird. Ein Outsourcing kommt nicht in Betracht. Welche der Varianten, 2 - Local oder 3 - Webbasiert, schliesslich das Rennen macht, ist noch Gegenstand interner Diskussionen und hängt wesentlich davon ab, ob HERMES PowerUser nur dem relativ kleinen Kreis von Projekten hoher Komplexität mit entsprechendem Bedarf an individueller Methodentailorierung zur Verfügung gestellt wird, oder auch dem grossen Rest der Projekte, die mit dem hochstandardisierten PostFinance-Projekttablauf optimal bedient sind und deshalb kaum Bedarf an Tailorierung haben.

Entscheidungsgrundlagen

Das Ergebnis der Studie diente der Entscheidungsfindung und basierte auf zwei Bewertungsgrundlagen:

7.2. Studie für die Einführung HERMES PowerUser in der PostFinance

- Kriterienkatalog, aus den Anforderungen abgeleitet
- Wirtschaftlichkeitsrechnung

Im Weiteren wurden die Machbarkeit und die Anwendbarkeit des HERMES PowerUsers mittels eines Piloten für die Methodiker untersucht und beurteilt.

Fazit

HERMES PowerUser ist, verglichen mit der bei PostFinance bisher in Projekten verwendeten Office-Umgebung, ein recht komplexes Instrument, das seine volle Wirkung dann entfaltet, wenn es auch in entsprechend komplexen Projekten eingesetzt wird. Für den Methodenbau und die Methodenpflege ist es auf jeden Fall nutzenstiftend.

Weiteres Vorgehen

Aufgrund der Studie wurde für den sofortigen Einsatz von HERMES PowerUser für den Methodenbau und die Methodenpflege entschieden. Um den flächendeckenden Einsatz des HERMES PowerUsers für die Projektleiter in der PostFinance weiter voranzutreiben, wurde ein Pilotprojekt (als Prototyp) in einem Projekt mittlerer Komplexität in Auftrag gegeben.

Literaturverzeichnis

1. Eclipse Process Framework Project, Eclipse.org, 2008
2. HERMES Systemadaption, ISB, Ausgabe April 2005
3. HERMES PowerUser, ISB, Release 2, 2008
4. Software Process Engineering Meta-Model, Object Management Group, Version 1.1

PostFinance



PowerUser Hermes PF

SEE 2009

Hans Peter Schaufelberger, PostFinance
Martin Wenger, APP Unternehmensberatung AG
X01.05, 27.05.2009

tailoriert

Slide 1

Agenda SEE 2009 Berlin

1. Einleitung

- Die Schweizerische Post – PostFinance

2. HERMES & PowerUser

- blaue Welt
- PowerUser – SPEM

3. Vorstellung Hermes PostFinance

- Projekte bei PostFinance: Fakten & Zahlen
- Hermes PF

4. PowerUser @ PostFinance (PU@PF)

- Wirtschaftlichkeit – Argumentarium
- Anwendung der Projektleiter
- Tailoring & Einführungskonzept

5. Fazit

Da stehen wir heute



Neben dem Zahlungsverkehr seit 1996
Ausbau zum Retail-Finanzinstitut
mit Produkten zu:

- Zahlen
- Sparen
- Anlegen
- Vorsorgen
- Finanzieren

Fakten und Zahlen 2008



- Bilanzsumme: 67,16 Mrd. CHF
- Gewinn (EBT): 235 Mio. CHF
- Anzahl Konten: 3,646 Mio. CHF
- Bestand Kundengelder: 49,265 Mrd. CHF
- E-Finance-Benutzer: 984'600 Personen
- Personalbestand: 2'889 Personen

Wege zu den Kundinnen und Kunden



- Regionale Verkaufsorganisationen für Geschäftskunden
- Geschäftskundenmanagement für Grosskunden/Banken
- PostFinance-Filialen für Privatkundinnen und -kunden
- Mobile Beraterinnen und Berater
- Kundendienst 7 x 24 Stunden
- Poststellen
- Telemarketing
- Internet www.postfinance.ch

Agenda SEE 2009 Berlin

1. Einleitung

- Die Schweizerische Post – PostFinance

2. HERMES & PowerUser

- blaue Welt
- PowerUser – SPEM

3. Vorstellung Hermes PostFinance

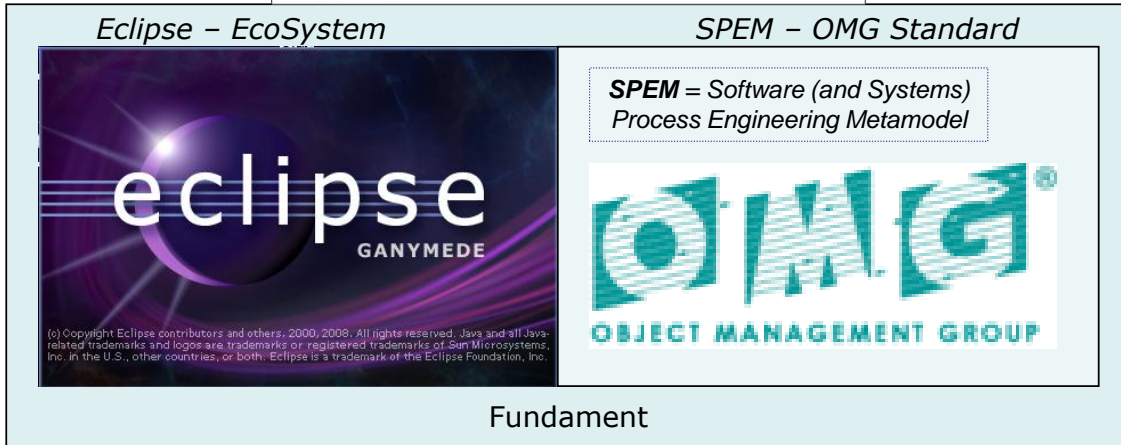
- Projekte bei PostFinance: Fakten & Zahlen
- Hermes PF

4. PowerUser @ PostFinance (PU@PF)

- Wirtschaftlichkeit – Argumentarium
- Anwendung der Projektleiter
- Tailoring & Einführungskonzept

5. Fazit

Fundament von HERMES PowerUser



Rollenspezifische Benutzer



Rollenspezifisches Benutzen von HERMES PowerUser

HERMES PowerUser Release 2 bietet den Benutzerinnen und Benutzern des Werkzeugs einen rollenzentrierten Einstieg: Abhängig von der jeweiligen Rolle bei der Arbeit mit HERMES stellt PowerUser spezifische Sichten auf die Methode oder auf ein Projekt zur Verfügung.

Im Arbeitsbereich – der **Workbench** – gibt es deshalb für jede Rolle eine spezifische **Perspektive**. Perspektiven dienen der übersichtlichen Darstellung wichtiger Funktionen und Informationen in **Sichten** – das sind die einzelnen Fenster auf der Workbench.

Klicken Sie auf ein Icon und Sie gelangen zur Perspektive für Ihre Rolle:



SPEM

- **SPEM:** Software (and Systems) Process Engineering Metamodel
- OMG Standard und Metamodell HERMES
- Add on eclipse process framework (EPF)

Agenda SEE 2009 Berlin

1. Einleitung

- Die Schweizerische Post – PostFinance

2. HERMES & PowerUser

- blaue Welt
- PowerUser – SPEM

3. Vorstellung Hermes PostFinance

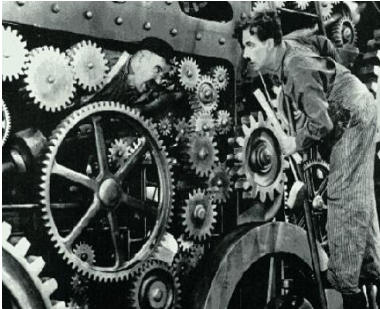
- Projekte bei PostFinance: Fakten & Zahlen
- Hermes PF

4. PowerUser @ PostFinance (PU@PF)

- Wirtschaftlichkeit – Argumentarium
- Anwendung der Projektleiter
- Tailoring & Einführungskonzept

5. Fazit

Die Projektmaschine bei PF ist herausgefordert...



„Projektmaschine“ bei PF

- Läuft zuverlässig
- Ist etabliert
- Bewältigt grosse Herausforderungen

→ Die „Projektmaschine“ ist ein Erfolgsfaktor in der Entwicklung von PostFinance

Innovations- und Veränderungsdruck hat zugenommen

- Wettbewerb
- Kostendruck
- Ansprüche und Bedürfnisse der Kundinnen und Kunden
- Weiterentwicklung der technischen Möglichkeiten
- Lebenszyklus von Kernapplikationen
- Marktchancen – Wachstumsmöglichkeiten
- Regulation

Konsequenz:

Einführung eines neuen Release von Hermes PF per Mitte letzten Jahres: **Hermes PF 2008**

Prinzipien für das Unternehmenstailoring am Beispiel PostFinance

Übergeordnetes Prinzip

- Verwendung von Standards
- Daraus folgt: Hermes nicht neu erfinden
- Daraus folgt: Hermes PF 2008 ist eine Standard-Tailorierung von Hermes 2003/2005

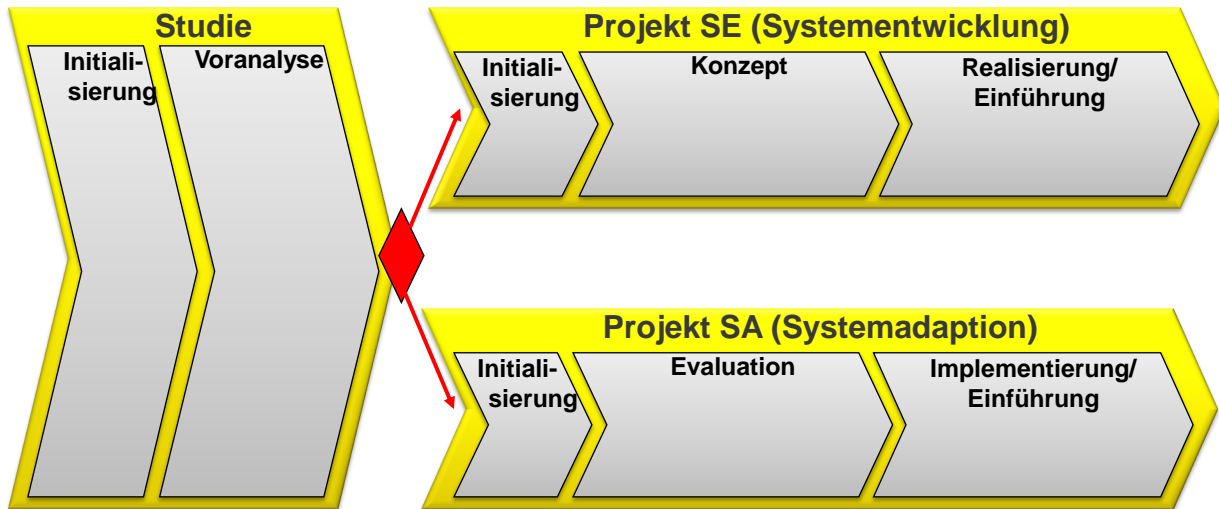
Drei Arten der Tailorierung

1. Streichung nicht relevanter Aufgaben, Rollen, Ergebnisse
2. Anpassung der verbleibenden Aufgaben, Rollen, Ergebnisse
3. Definition zusätzlicher Aufgaben, Rollen, Ergebnisse

Tailorierungsprinzipien bei PostFinance

- Streichung ist Vorgehensweise der Wahl
- Anpassung konservativ einsetzen, wenn PostFinance-Gepflogenheiten es erfordern
- Zusätzliches Definieren nur in absoluten Ausnahmefällen

Phasenmodell Hermes PF2008: auf PostFinance tailoriert

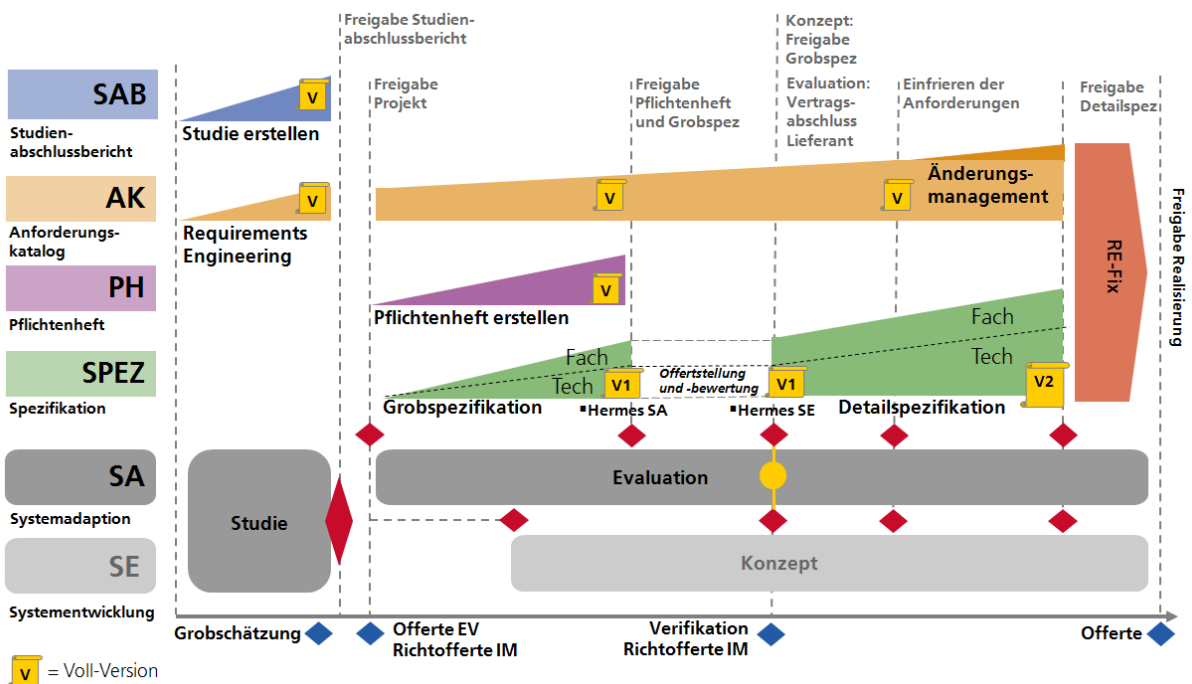


Version: x01.05 Datum 27.05.2009

PostFinance / 13

Die Schweizerische Post / PostFinance

Hermes PF 2008: Studie und Konzept



Version: x01.05 Datum 27.05.2009

PostFinance / 14

Die Schweizerische Post / PostFinance

Agenda SEE 2009 Berlin

1. Einleitung

- Die Schweizerische Post – PostFinance

2. HERMES & PowerUser

- blaue Welt
- PowerUser – SPEM

3. Vorstellung Hermes PostFinance

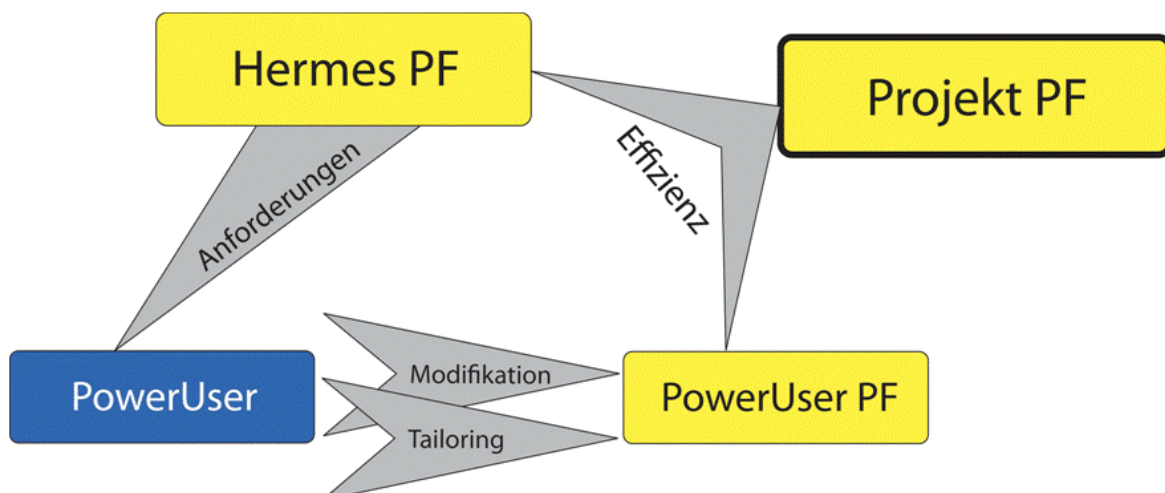
- Projekte bei PostFinance: Fakten & Zahlen
- Hermes PF

4. PowerUser @ PostFinance (PU@PF)

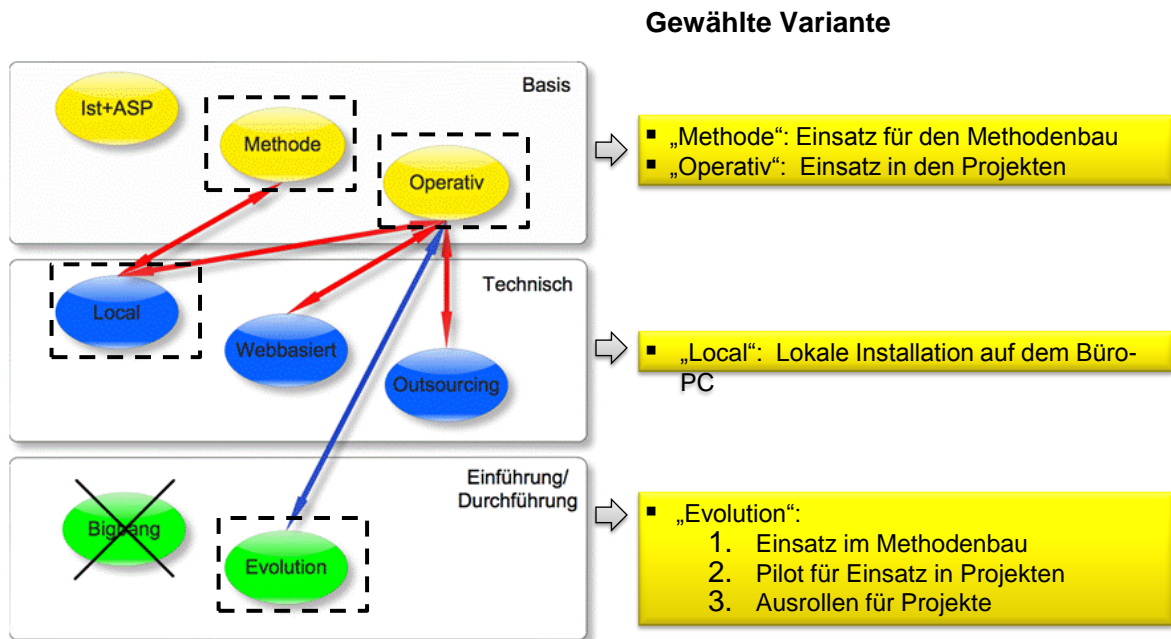
- Wirtschaftlichkeit – Argumentarium
- Anwendung der Projektleiter
- Tailoring & Einführungskonzept

5. Fazit

Studienauftrag



Variantenbildung anhand eines morphologischen Kastens



Version: x01.05 Datum 27.05.2009

PostFinance / 17

Die Schweizerische Post / PostFinance

Anforderungen

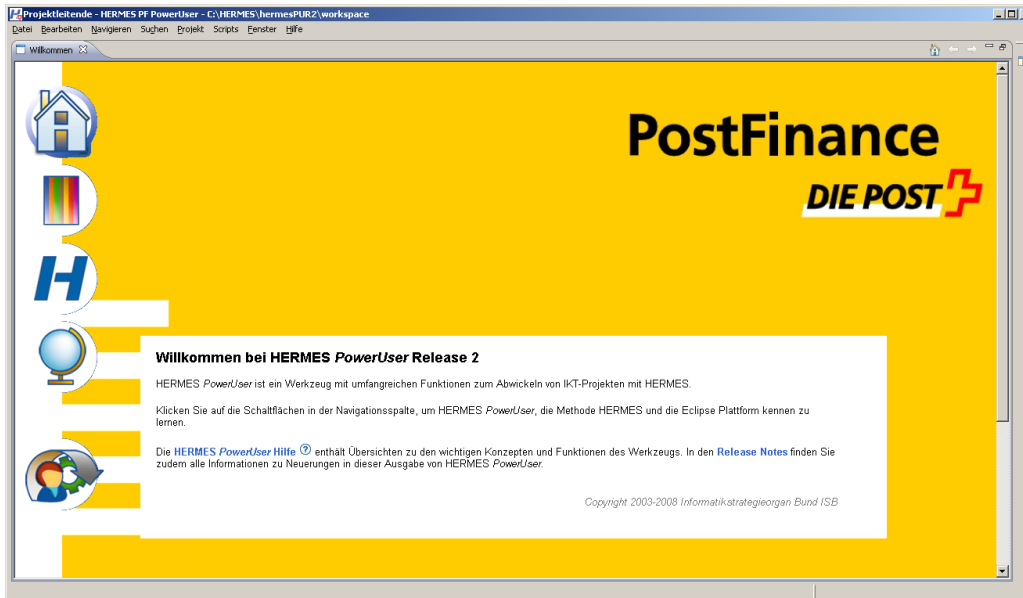
Nicht-funktionale Anforderungen	Funktionale Anforderungen
Leicht automatisiert anwendbar für Projektleitung	Projektplan integrierbar
Leicht zugänglich	PostFinance-Tools leicht anbindbar
Leicht bedienbar	In Produktiv-Umgebung integrierbar
Intuitiv	Template-Dokumente aktualisierbar
Zeitsparend durch automatisierte Abläufe	Windows und Linux tauglich
Stabilität im Betrieb	Vista kompatibel
	Update möglich

Version: x01.05 Datum 27.05.2009

PostFinance / 18

Die Schweizerische Post / PostFinance

PowerUser auf PostFinance tailoriert



Argumente pro & kontra Einsatz PowerUser@PostFinance

Pro

- + OpenSource, frei verfügbar
- + Nationaler Bundesstandard
- + SPEM-konformes Repository
- + Rückt PF näher an HERMES Bund

Contra

- Komplexität des Tools
- Hohe Ausbildungskosten
- Integration in Toollandschaft schwierig und aufwändig
- Durchgängigkeit zur MS-Office-Welt

Agenda SEE 2009 Berlin

1. Einleitung

- Die Schweizerische Post – PostFinance

2. HERMES & PowerUser

- blaue Welt
- PowerUser – SPEM

3. Vorstellung Hermes PostFinance

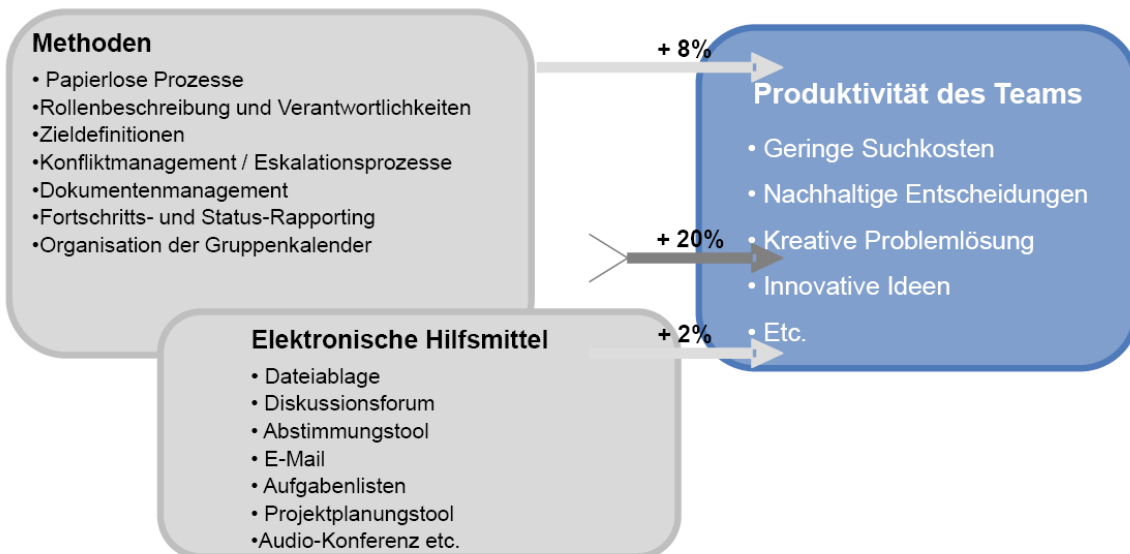
- Projekte bei PostFinance: Fakten & Zahlen
- Hermes PF

4. PowerUser @ PostFinance (PU@PF)

- Wirtschaftlichkeit – Argumentarium
- Anwendung der Projektleiter
- Tailoring & Einführungskonzept

5. Fazit

These: Leverage der Wirkung – Produktivität der Teams



Fazit

Nutzen:

- Im Methodenbau nutzenstiftend
- Hebelwirkung erst bei operativem Einsatz (PL & Team)

Aber: Komplexes Instrument

Keine Einführung im 2009, frühestens ab 2010:

- Die Transformation zu Hermes PF 2008 ist noch nicht abgeschlossen, die Anwendung von Hermes PF 2008 ist noch nicht genügend verankert
- Die Toollandschaft bei PostFinance ist im Umbruch begriffen und nicht stabil genug für eine Integration des PowerUsers

Erfahrungen im Einsatz mit PowerUser:

- Bund durchzogen
- PostFinance Pilot noch nicht messbar (Messung/Beurteilung noch zu früh)

Kontakt



Hans Peter Schaufelberger

Ingenieur ETH

Die Schweizerische Post, PostFinance

hanspeter.schaufelberger@postfinance.ch / +41 58 338 01 82

Nordring 4a

CH-3030 Bern



Martin Wenger

EI. Ingenieur CMC

APP Unternehmensberatung AG

martin.wenger@app.ch / +41 31 380 59 59

Monbijoustrasse 10

Postfach 5109

CH-3001 Bern

7.3. Projekt Recovery nach dem War Room Konzept

Joachim Bauchrowitz

Senior Consultant Development Processes, Certified CMMI Lead Appraiser

ESG Elektroniksystem- und Logistik-GmbH

Geschäftsbereich Automotive

Frankfurter Ring 211

D-80807 München

Joachim.Bauchrowitz@esg.de

Mobil: +49 . (0)174 / 30 59 51 4

Abstract

Wie kann ein Projekt gerettet werden, das über eine Auftraggeber/Auftragnehmer-Kette von 4 Firmen an 6 europäischen Standorten entwickelt wird und offensichtlich sein Ziel nicht erreichen kann, ein SIL 3 Steuergerät (fristgerecht) zu entwickeln?

An Hand eines (anonymisierten) Beispiels wird das Konzept des War Rooms beschrieben und die pragmatisch eingeschlagenen Vorgehensstrategien vorgestellt. Das Projekt stammt nicht aus meinem bisherigen Kundenkreis und beschreibt das Dilemma, in welches Unternehmen geraten, die die Anforderungen einer sicherheitskritischen Serien-Systementwicklung (mit Mechanik, Elektrik, Elektronik und Software) unterschätzen und versuchen, die Situation mit ad-hoc- und Try-and-Error-Methoden zu meistern.

Es beschreibt aber auch, wie die Zulieferer am Ende der Kette durch sukzessives Einführen von Entwicklungsprozessen aus der Lähmung herauskommen und bei den Unternehmen in der Lieferkette davor wieder an Reputation gewinnen können. Es zeigt, dass Projektführungen von Unternehmen in der Auftragskette eine (historisch verständliche) Blockadehaltung nicht durchhalten können, wenn am Ende der Lieferkette vernünftige Prozesse pragmatisch eingeführt, sukzessive aufgebaut und gelebt werden.

In diesem Beispiel ist es nicht wichtig, welches Prozessmodell als Orientierung für die Verbesserung genutzt wird: In diesem konkreten Falle lieferten die Ziele von CMMI ML3 die Vision und die Automotive SPICE Prozesse die konkrete Vorlage für die Leitlinien der Prozessimplementierung.

„Wir brauchen keinen Qualitätssicherer, bei uns erzeugen die Mitarbeiter Qualität in ihrer täglichen Arbeit“ 



QUALITY
ASSURANCE
AND THE
SINKING OF
THE
LARGEST
OFFSHORE
OIL
PLATFORM
March 2001

1

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH



SEE 2009



Projekt Recovery nach dem
WarRoom Konzept

Joachim Bauchrowitz

ESG Elektroniksystem- und Logistik-GmbH

27. Mai 2009



Joachim Bauchrowitz

CMM Lead Assessor

Zertifizierter CMMI Lead Appraiser

ESG Elektroniksystem- und Logistik-GmbH

Regionalbüro München Nord

D-80807 München

Frankfurter Ring 211

+49 . (0)89 / 9216 – 2669

+49 . (0)174 / 30 59 51 4

Joachim.Bauchrowitz@esg.de

www.esg.de

2

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Agenda



- Informationen zu meinem Unternehmer (kurz, siehe Proceedings)
- Informationen zu mir (kurz, siehe Proceedings)
- Vorbemerkungen
- Das Projekt, die Ausgangssituation und unser Auftrag
- Ansatz
- WarRoom Konzept
- Iteratives Projekt Recovery
- Iterativer Prozessaufbau
- Ergebnisse im Projekt
- Ergebnisse bei den Prozessen
- Was haben CMMI, Automotive SPICE und V-Modell gebracht
- Lerneffekte
- Assessment-Killer bei diesem Vorgehen

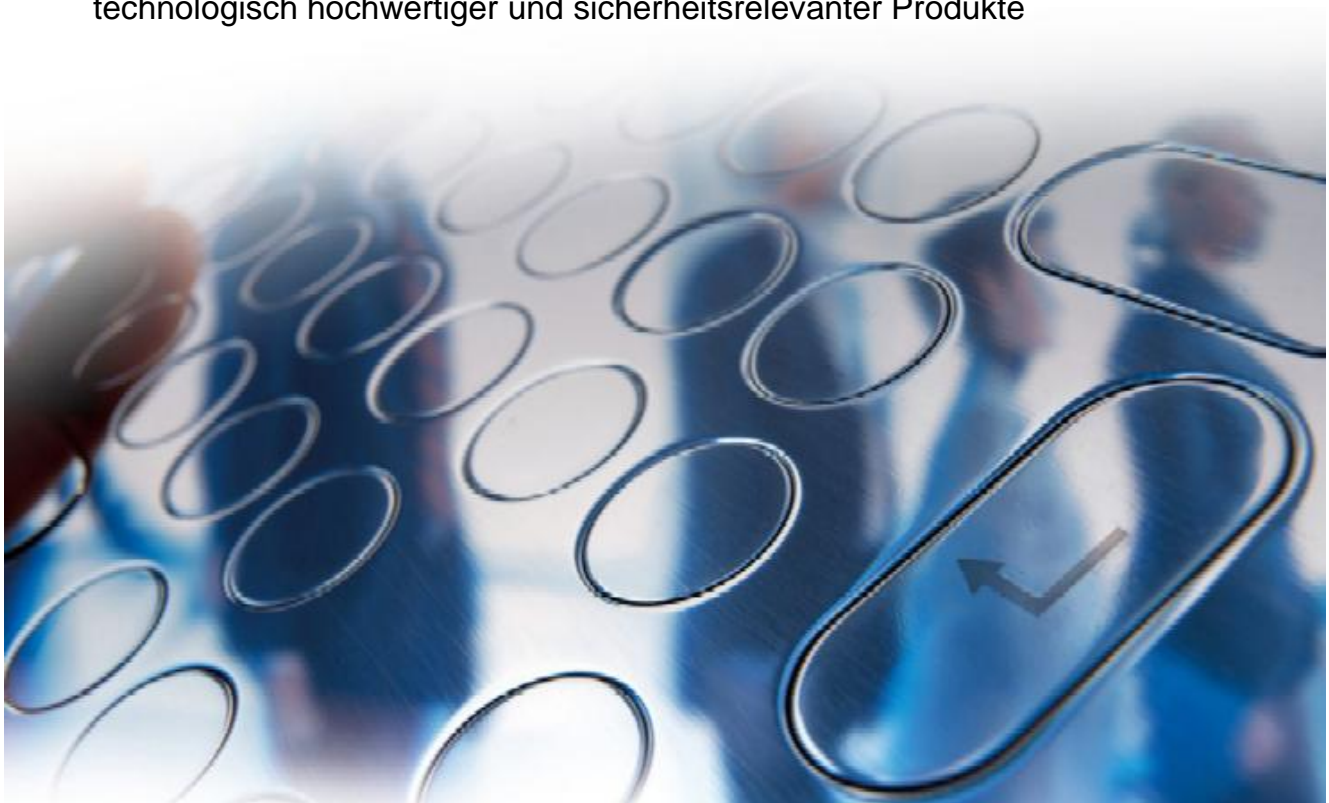
3

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Das System- und Softwarehaus ...



... für Entwicklungs- und Serviceprozesse softwareintensiver, komplexer, technologisch hochwertiger und sicherheitsrelevanter Produkte

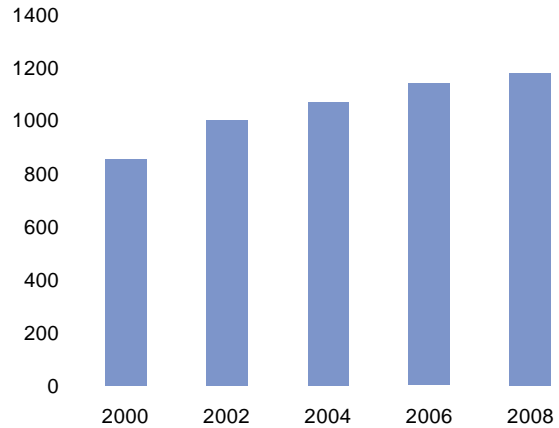
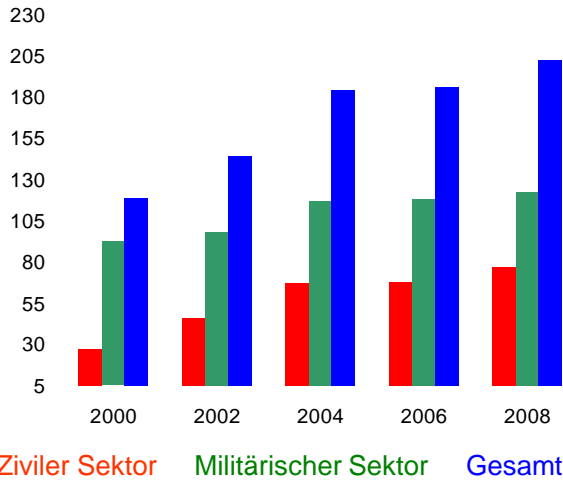


Stetes Wachstum



Umsatz (Mio €)

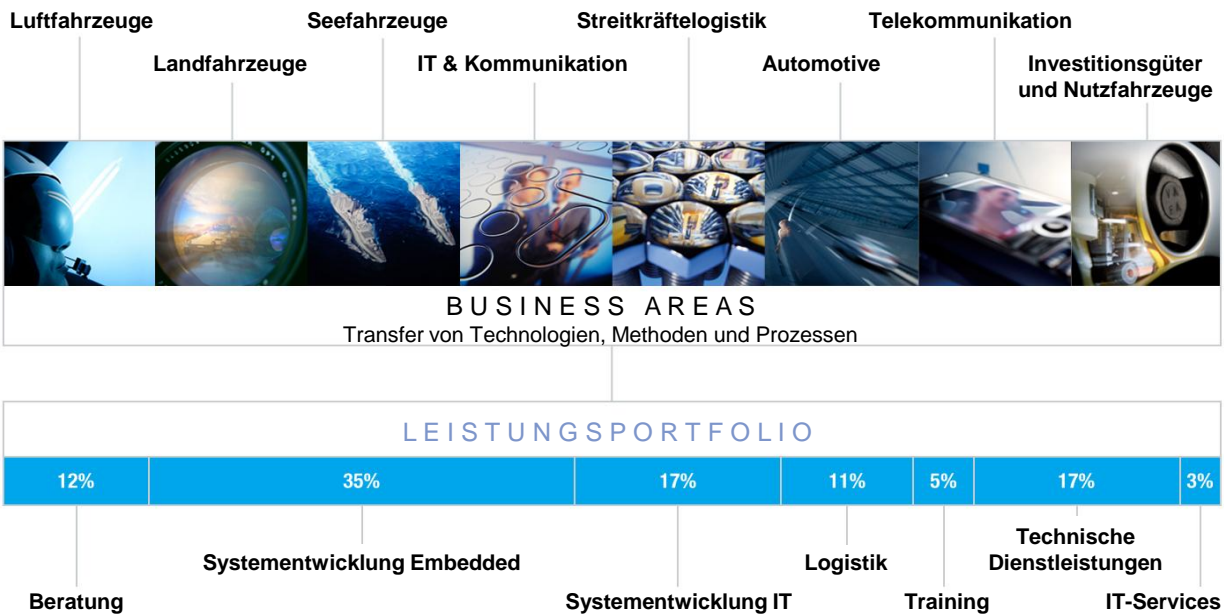
Mitarbeiter



7

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Bereichsübergreifender Technologietransfer






8

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Standorte der ESG-Gruppe



-  **Zentrale**
Fürstenfeldbruck
-  **Niederlassungen**
Berlin
Bonn
Hamburg
Ingolstadt
Koblenz
Köln
Marseille
München
Paris
Raunheim
Stuttgart
Wilhelmshaven
Wolfsburg
-  **Beteiligungen**
Bonn
Fürstenfeldbruck
Hamburg
Langenargen
Paris
Detroit

9

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Kundenliste ESG (Auszug)



10

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Joachim Bauchrowitz

Position: Senior Consultant

Spezialgebiet:

Prozessanalyse und -verbesserung
System- und Software- Engineering

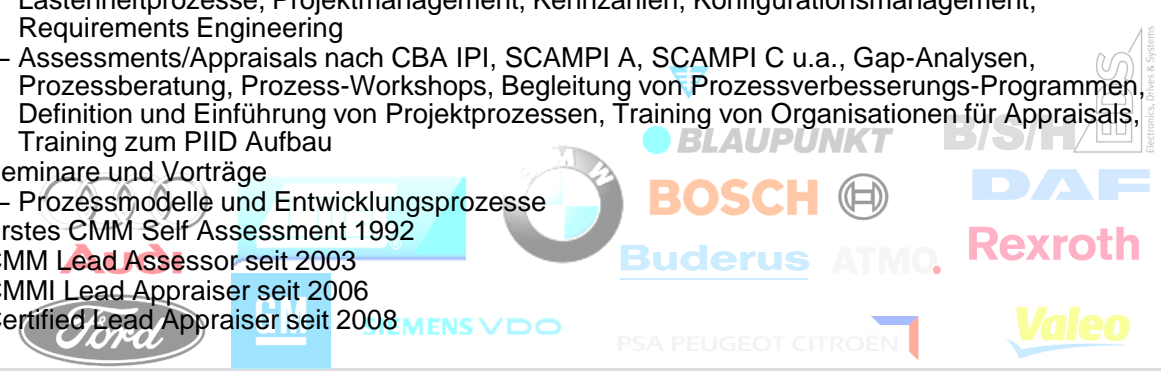


Projekterfahrung (Auswahl)

Kunde	Projekt	Aufgabenfokus
ESG BWB/BMVG	Qualitätsmanagement und Engineering	Qualitätsbeauftragter und Engineering-Beauftragter für einen Geschäftsbereich Konzeption, Aufbau und Betrieb unternehmensweite Standard-Entwicklungsumgebung, sowie interne und externe Methodenberatung und Consulting Mitglied des Industrieboards bei der Entwicklung des deutschen V-Modells '92
BWB/BMVG	SAMOC	Projekt- und Entwicklungsleitung für die Entwicklung des Planungssystems für die Raketen-Flugabwehrsysteme der deutschen Luftwaffe
Innenministerium Mecklenburg- Vorpommern	LAPIS	Konzeption und Entwicklungsbegleitung für das „Landesweite Polizeiinformationssystem Mecklenburg-Vorpommern“
Automobil- Industrie (OEM und Zulieferer)	Prozessverbesserungs- maßnahmen	Durchführung CMMI Appraisals und CMM Assessments Definition und Umsetzung von Prozessen für Projektmanagement, Kennzahlen, Prüfstrategien, Qualität, Zulieferer, Anforderungen Erarbeitung von Konzernweiten Freigabeprozessen Analyse und Verbesserung von Entwicklungsprozessen in Entwicklung und Produktion Coaching von Verbesserungsprojekten und -teams

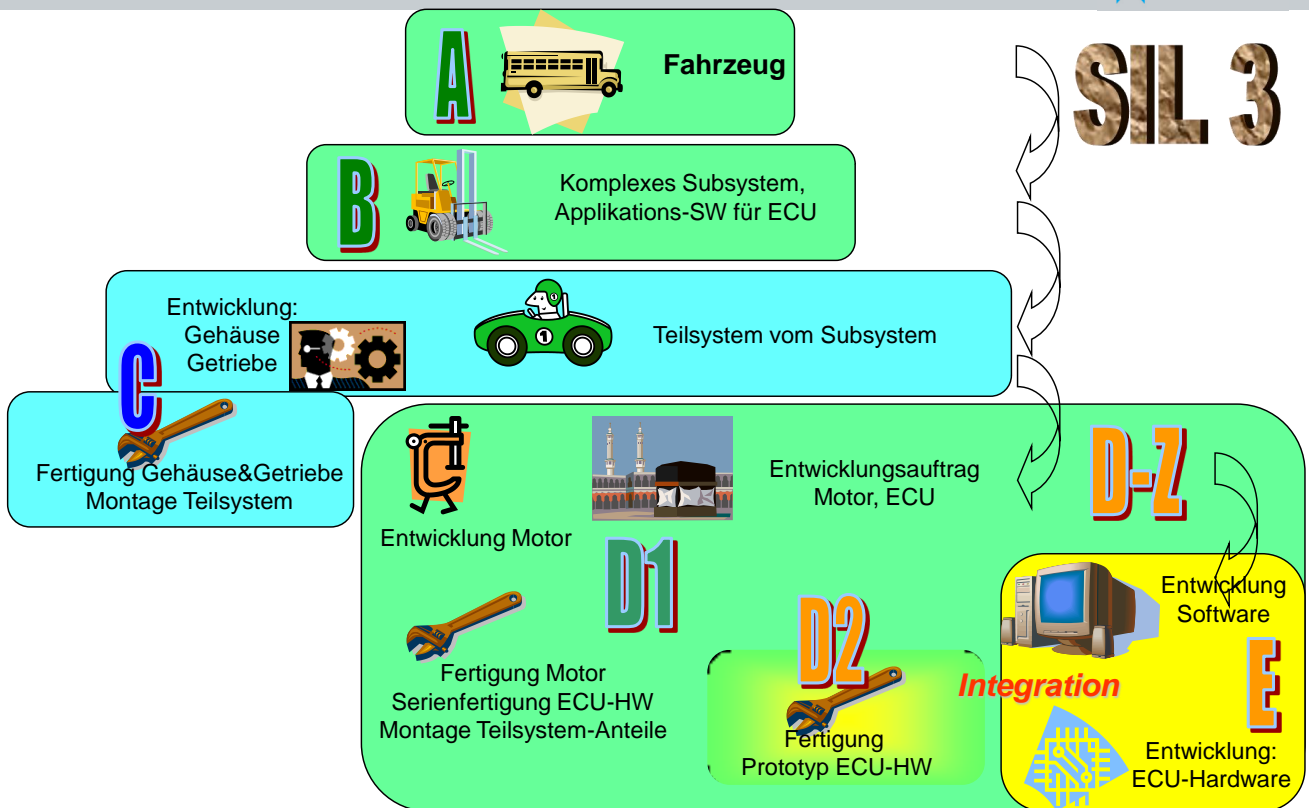
Erfahrungen und Kunden Joachim Bauchrowitz

- Informatiker
 - Compilerbau, Programm-Korrektheit
- Realtime-Entwicklung
 - Assembler, Pearl, C
- Software Engineering und Consulting
 - Seit über 25 Jahren
 - Entwicklung und Betrieb Unternehmensweite Standard-Entwicklungsumgebung
 - Methodenberatung, Consulting und Support intern und extern
- Mitglied des Industrieboards
 - Bei der Entwicklung des deutschen V-Modells '92
- Projektleitung SW-Großprojekt
 - Führungsinformationssystem der deutschen Luftwaffe
- Automotive-Consulting Entwicklungsprozesse seit 1999 (Software, Hardware, Systeme, ...)
 - KMM/SIM, Flashprozesse, Freigabeprozesse, Prüfprozesse, Entwicklungsprozesse, Lastenheftprozesse, Projektmanagement, Kennzahlen, Konfigurationsmanagement, Requirements Engineering
 - Assessments/Appraisals nach CBA IPI, SCAMPI A, SCAMPI C u.a., Gap-Analysen, Prozessberatung, Prozess-Workshops, Begleitung von Prozessverbesserungs-Programmen, Definition und Einführung von Projektprozessen, Training von Organisationen für Appraisals, Training zum PIID Aufbau
- Seminare und Vorträge
 - Prozessmodelle und Entwicklungsprozesse
- Erstes CMM Self Assessment 1992
- CMM Lead Assessor seit 2003
- CMMI Lead Appraiser seit 2006
- Certified Lead Appraiser seit 2008



- Erster Einsatz für diesen Kunden
- Der Kunde steht nicht auf der Referenzliste
- Es handelt sich um einen europäischen, aber keinen deutschen OEM

Die Beauftragungskette



Ausgangsbasis



- Management by Telekom
- Alle 2-3 Monate Wechsel des Projektleiters (geschlechtsneutral)
- Kein Entwicklungsplan
- Keine Qualitätsorganisation „Qualität entsteht bei der Entwicklung“
- Sicherheitskonzept fehlt
- SIL-Aktivitäten beschränken sich auf FMEA/FDEMA-Erstellung
- Lieferanten B, C, D, E ohne SIL3-Erfahrung, angeblich SIL2-Erfahrung
- Lieferanten B, D ohne SPICE Erfahrung
- Lieferant E ohne Erfahrung in Serienentwicklung und Leistungselektronik
- Keine Prozessdefinitionen bei Lieferanten D, E
- Ausschließlich „ad-hoc“ Prozesse
- Anforderungen waren nicht fixiert und analysiert
- Alle Lieferungen waren mangelbehaftet und verzögert
- Massive Reklamationen aller Mitglieder der Kunden- und Lieferantenkette
- Projekt läuft seit 18 Monaten, 6 Monate Laufzeit verbleiben

15

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Auftrag an ESG



- Klärung der Frage im Auftrag von „D-Z“: Was läuft schief?
 - ⇒ Erst nach mehreren Wochen erfahren wir, dass das Projekt in der Business Line von D1 hängt
 - ⇒ Das Werk D1 ist erst seit kurzer Zeit Teil des Konzerns
 - ⇒ Das Werk D2 ist ebenfalls erst seit kurzer Zeit im Konzern
- Retten Sie das Projekt
- Produzieren Sie das gewünschte Produkt
- Stellen Sie die Kunden zufrieden
- Bestehen Sie ein SPICE 2 Assessment
- Bestehen Sie ein SIL 3 Audit
- Sie haben 6 Monate Zeit

16

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Ansatz



- Bündelung der Management-Kräfte: Komplett neue Führungsschicht
- Etablierung
 - Projektleiter
 - Prozessverantwortlichen und Sub-Projektleiter
 - Anforderungsmanager
 - Change Manager
 - Test Manager
 - Qualitäts Manager
 - Hardware Entwicklungsleiter
 - SIL-Manager
- Blockieren aller vorhandenen Kommunikationskanäle mit den div. Kunden
- Schnelle Skizzierung von fundamentalen Basis-Prozessen (Flowcharts)
 - SUP1 (Qualitätssicherung), SUP10 (Change Mgt), SPL2 (Release Mgt)
- Aufstellung eines WarRooms (Konzept kommt gleich)

17

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

War Room



- Bündelung (Kasernierung) der Entwicklungs-Kräfte in einem Raum
- Jede Wand senkt die Effizienz des WarRooms
- Verfügbarkeit kleinerer Besprechungsräume für Kundenkonferenzen und vertrauliche Besprechungen
- Kurzfristige Verfügbarkeit mehrerer Besprechungsräume mit Beamer
- Projekt-Leitungsmeetings werden im Stehen abgehalten und Online dokumentiert
- Etablierung einer WarRoom Kommunikations-Charta
- Aufbau einer internen Projekt-Kommunikationsstruktur
- Anpassung der externen Projekt-Kommunikationsstruktur, externe Kommunikation ausschließlich über den Projektleiter
- Leitwolf als Projektleiter mit „Hot-standby“-Ersatz und Assistenz
- Täglicher Informationsabgleich der Teamleiter mit strikter Moderation
- Klare Teamstruktur mit eindeutiger Klärung von Verantwortlichkeit, Befugnis und Team-Zuordnung
- Positionierung aller Teamleiter bei ihren Teams

18

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Täglicher Jour Fixe



Tägliche Besprechung des Projektleiters mit den Teamleitern

20 Minuten am Beginn jedes Arbeitstages

Im Stehen

Assistent hat Bistro-Tisch mit Netzanschluss

19

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Regularien Jour Fixe



Jeder Teilnehmer ist pünktlich da

Moderator weist die Sprechzeiten jedem Teamleiter zu (3 Minuten)

Moderator stoppt Kommentierungen und Diskussionen

Moderator bestimmt Folge-Meetings und Teilnehmer

Äußerungen werden nicht kommentiert

Fragen zur Klärung von Punkten sind gestattet

Sofortige Diskussion der Punkte ist nicht gestattet

Jeder Team Leiter ist auf die Agenda (folgt) vorbereitet

Es werden ausschließlich Fakten berichtet (sonst unterbricht der Moderator sofort)

Assistent dokumentiert Anwesenheit von Personen

Assistent schreibt relevante Informationen sofort in die LoP (List of open Points)

Assistent markiert die Punkte, die Entscheidungen sind

Assistent managt die LoP (Terminverfolgung)

Ergebnis ist für jedes Team Mitglied auf dem File-Server sichtbar

20

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH



Beispiele

Date	Participants	Par	Pa	Pa	Pa	Pa	Par	Pa	Pa	Pa	Par	Pa	Pa	Pa	Par	Pa	Pa	Pa	Par	Pa	Pa	Pa	Par	Pa	Pa	Pa
		sci	ric	ric	ric	ric	sci	ric	ric	ric	sci	ric	ric	ric	sci	ric	ric	ric	sci	ric	ric	ric	sci	ric	ric	ric
13.09.2006	participant present = y participant not present = n participant not invited this time = i participants sick/on holiday = h	y	y	y	y	y	h	h	h																	
20.09.2006		y	y	y	y	y	y	y	y																	
27.09.2006	Remark: Participants who are not invited, who are sick or on holiday will not decrease the attendance.	y	y	h	y	h	y	y	n																	
04.10.2006		y	y	y	y	y	y	y	n		n	n														
11.10.2006		i	y	y	i	n	y	y	y																	
18.10.2006		y	y	y	y	y	y	y	y		h	y														
25.10.2006																										
01.11.2006																										
08.11.2006																										
15.11.2006																										
22.11.2006																										
29.11.2006																										
06.12.2006																										
13.12.2006																										
20.12.2006																										
27.12.2006																										

ID	Item	Prozesskategorie	Bereich	Verantwortlicher	Termin	Wichtigkeit	Status	Benennung	Region
1	89	Struktur	HR/HR		17.09.06	40%	Yellow		
2		HR	HR/HR		17.09.06	40%	Green		
3	84	HR	HR/HR		18.09.06	40%	Red		
4		HR	HR/HR		18.09.06	40%	Yellow		
5		HR	HR/HR		17.09.06	40%	Red		
6		HR	HR/HR		18.09.06	40%	Red		
7		HR	HR/HR		18.09.06	40%	Green		
8		HR	HR/HR		17.09.06	40%	Green		
9		HR	HR/HR		18.09.06	40%	Green		
10		HR	HR/HR		18.09.06	40%	Green		
11		HR	HR/HR		18.09.06	40%	Green		



Teilnehmer Jour Fixe

Teilnehmer sind Teamleiter (Fett: Report)

- NN Moderator
- NN Projektleiter**
- NN Teamleiter ECU Hardware**
- NN Teamleiter ECU Software**
- NN Teamleiter Requirements Management & Change Management**
- NN Teamleiter Test Management**
- NN Teamleiter Qualität**
- NN Teamleiter Prozesse & SPICE
- NN Teamleiter SIL
- NN Teamleiter Prototyp Entwicklung D2
- NN Teamleiter Motor & Mechanik Entwicklung, Produktion D1
- NN Assistent LoP

- Für jeden Teamleiter nach Aufruf (jeder 3 Minuten)
 - Status und Fortschritt des Teams (im/nach/vor Zeitplan)
 - Wichtige Aktionen für den Tag
 - Risiken, Road Blocker und bekannte Probleme
 - Notwendige Entscheidungen (Team- oder Projektleiter)
- Weitere Entscheidungen (2 Minuten)

- Management aller Informationen durch Projektleiter mit Assistent
- Kaskadierung der Informationen aus Jour Fixe in Teambesprechungen
- Jede Entwicklungstätigkeit erfolgt zentral
- Ermittlung / Diskussion Leistungsfortschritt durch Projektleiter erfolgt offline
- Durchführung aller Kundenbesprechungen durch Projektleiter
- Aufhebung der Lieferantenschnittstelle Dx / E durch PL und Joint Team

- Definition Projekthandbuch, Kommunikationsdiagramm, Projekt-OrgChart
- Planung von Baselines und Releases (sukzessiv: was ist wann zu liefern?)
- Definition/Verabschiedung eines internen Absicherungsprozesses
- Lieferungen ausschließlich nach Freigabe durch Qualität oder Sonderfreigabe durch Leitung (D1 / D-Z)
 - ⇒ Erste Lieferung 2 Wochen nach Projektstart
 - ⇒ Erdbeben für die Organisation
 - ⇒ Lieferungen im 2-4 Wochenrhythmus

Vorgehen Projekt , Prozessbezogen



- Dokumentation aller Informationen und Entscheidungen in einer LoP
- Sukzessive Erweiterung des Planungshorizonts von Tagen über Wochen auf Monate und Milestones
- Einführung einer Planverfolgungsmethodik
- Tägliche Abstimmung der Teamleiter

- Institutionalisierung SUP10, SPL2 und SUP1 (Basis)
- Skizzierung von Rollen und Befugnissen
- Skizzierung weiterer Prozesse (MAN3, MAN5, ENGx, SUP8, SUP9)
- Einführung eines Problem- und ChangeRequest-Management-Tools
- Einführung eines KM-Tools
- Erfassung und Abstimmung der Requirements im Entwicklungsteam
- Erarbeitung eines umfassenden Testkonzepts mit EMV und SIL3
- Erweiterung der Aktivitäten und Konzepte um SIL-Notwendigkeiten
- Aufbau eines Design Validation Plans

25

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Vorgehen bei Prozessen



- Definition eines Prozessplans für das Projekt
 - ⇒ Change Request Management, Release Management
 - ⇒ Automotive SPICE HIS Scope
 - ⇒ Erweiterung Scope um weitere Prozessbereiche gemäß Vorgabe AG
- Analyse der verfügbaren und nutzbaren Unternehmensprozesse
- Definition der Prozesse **durch die Arbeitsebene**
- Definition, Review und Freigabe eines Review Prozesses
- Review jedes Prozesses durch Review Teams, Freigabe durch QS
- Ableitung von Audit-Fragebögen aus den Prozessen
- Aufbau eines Audit-Prozesses (in Vertretung für eine Unternehmens-QM)
- Audits der praktischen Arbeitsabläufe im Projekt
- Monatliche Capability Analyse der Projektprozesse (Automotive SPICE)
- Information des Kunden über die aktuellen Prozess-Capabilities
- Abstimmung des Assessment Scopes mit den Kunden

26

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

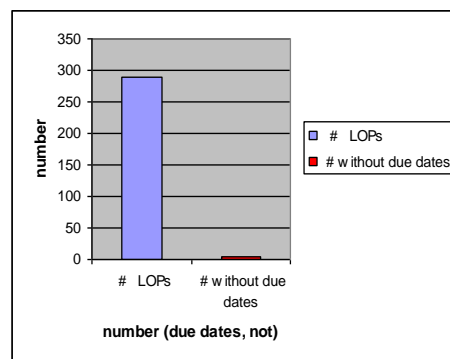
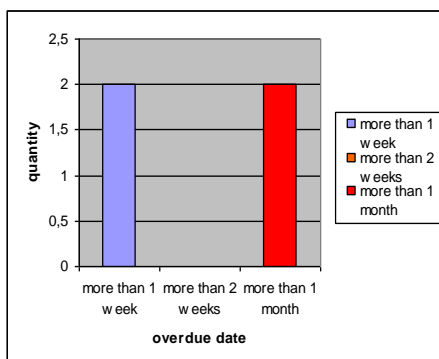
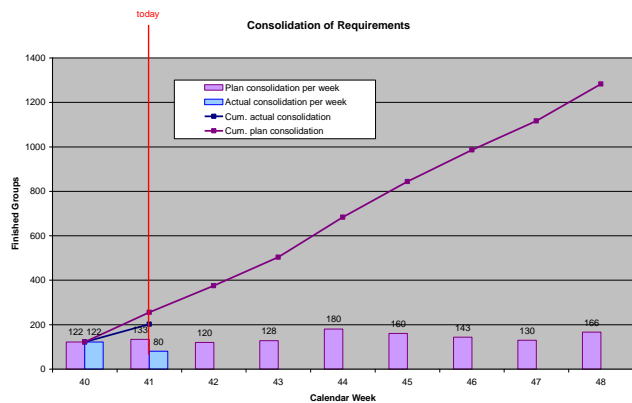
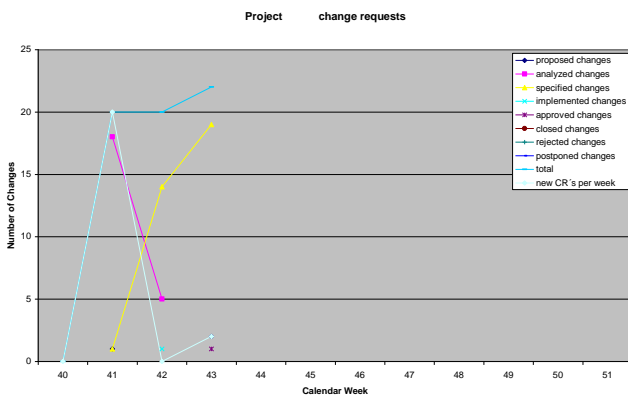
Prozessbezogene Leistungsstufen

- Jan 09
 - ⇒ Grundprozesse beschrieben
 - ⇒ Start der Reviews der Prozesse
- April 09 (geplant, voraussichtlich erreichbar)
 - ⇒ Traceability über alle ENG-Bereiche
 - ⇒ Verfügbarkeit ECU-HW-, Mechanik- und Motor- Prozesse analog SW
 - ⇒ Automotive SPICE: Nachweisfähigkeit der Basis Praktiken
 - Abwicklung Prozesse gemäß Prozessbeschreibung (19 PA)
 - ⇒ Bereitschaft zum Assessment Level 1
 - ⇒ Einführung eines unternehmensbezogenen Qualitätsmanagements
- Juli 09 (geplant, voraussichtlich erreichbar)
 - ⇒ Bereitschaft zum Assessment Level2
 - Nachweis von Audits gemäß Strategie
 - Führung des Projekts über qualifizierte Nutzung von Kennzahlen (LoP, Requirements, CR's, MISRA, Defects, Reviews, Cx-Cov, ...)

29

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Beispiele für Kennzahlen



30

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Was haben CMMI, Automotive SPICE und V-Modell gebracht?



- CMMI
 - ⇒ Übergeordnete Ziele (Planbarkeit, ...)
 - ⇒ Best Practices
 - ⇒ Beispiele aus Beratung
 - Templates, Prozesse, Kennzahlen, Erfahrungen, ...
- Automotive SPICE
 - ⇒ Zielorientierung externe Prüfung
 - ⇒ Gap-Analysen zur Ermittlung der Prozess-Defizite
 - ⇒ Inhalte von Dokumenten
 - ⇒ Practices als Checklisten für Prozess-Inhalte
- V-Modell
 - ⇒ Produktmuster
 - ⇒ Konkrete Abläufe

31

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Was lernen wir daraus?



- Ein total vermurkstes Projekt kann auch ohne rigorosen Stopp wieder auf die Beine kommen, ein WarRoom hilft dabei
- Eine Kasernierung der Mitarbeiter ist zielführend, aber extrem belastend
- Der Aufbau eines WarRooms darf nicht in Tages-Reisenähe der Mitarbeiter erfolgen
- Wenn Teile der Leistungserbringung im WarRoom fehlen, sackt die Erfolgswahrscheinlichkeit dramatisch ab (Motor-Entwicklung, Produktion)
- Der Projektleiter muss absolute Weisungsbefugnis haben
- Wir haben ständig mit 2 Projektleitern parallel agiert (Firefighter+Strategie)
- Leistungserbringung kann trotz Urlaub, Wechsel, Kündigung, Krankheit, Erziehungsurlaube, Firmenfeiern kontrolliert vorangetrieben werden
- Frühe Konzeptentscheidungen konnten nie wieder entscheidend korrigiert werden
 - ⇒ Architekturdesign ohne Sicherheitskonzept, Leistungselektronik-Erfahrung und EMV-Betrachtung
 - ⇒ Fehlende Serien-Erfahrung beim Entwurf
 - ⇒ Stückkosten waren nicht überwacht und außer Kontrolle

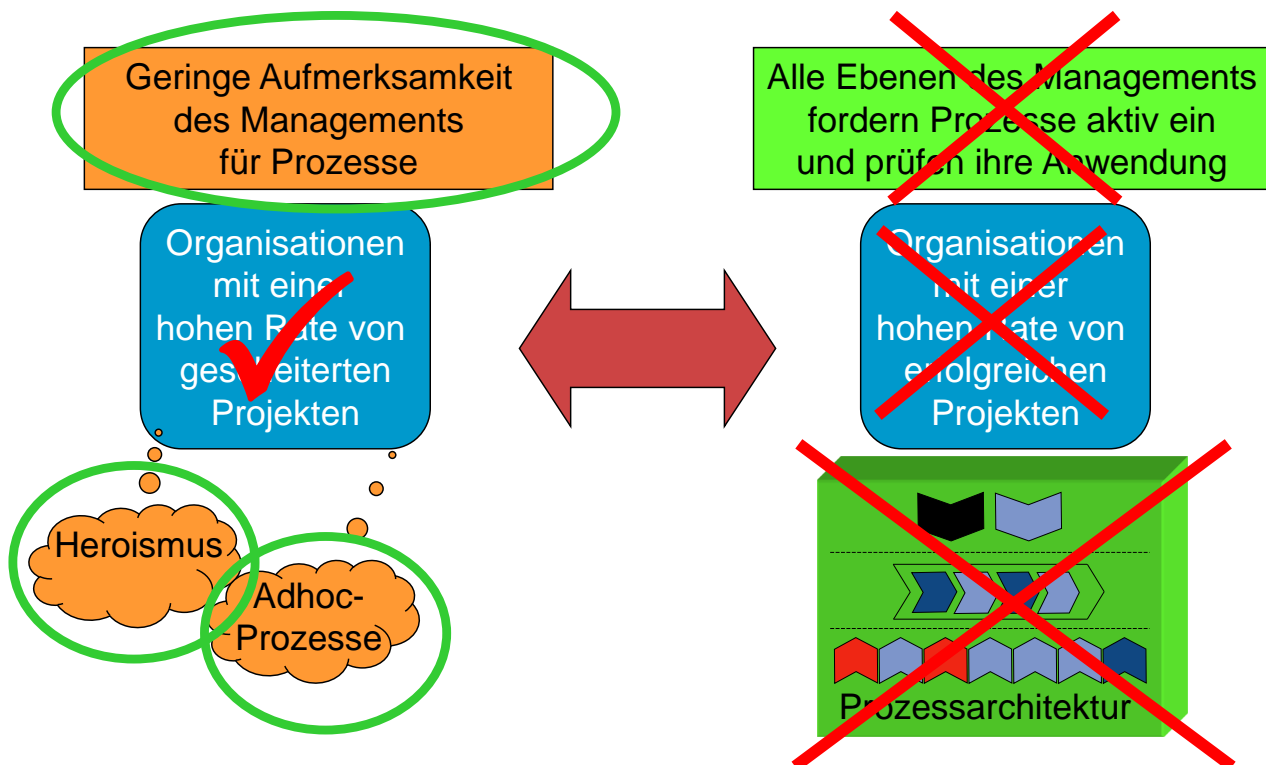
32

ESG-Bauchrowitz | 27.05.2009 | © Alle Rechte bei ESG Elektroniksystem- und Logistik-GmbH

Was wären die Assessment-Killer gewesen?

- Automotive SPICE
 - ⇒ Schätzungen von Arbeitspaketen
 - ⇒ Nachhaltigkeit der Verankerung der Qualität im Unternehmen
 - ⇒ Auswahl und Steuerung von Zulieferern
 - ⇒ Nachverfolgbarkeit der Anforderungen
- SIL
 - ⇒ Entscheidungsgrundlage des Managements
 - ⇒ Hazard-Basis und Feldausfall-Verfolgung
 - ⇒ Nachhaltigkeit der Verankerung der Qualität im Unternehmen
 - ⇒ Formalistische Prozesse contra praktikable Prozesse
 - ⇒ Querschüsse durch verspätetes Sicherheitskonzept
 - ⇒ Produktkonzept
- Sonstiges
 - ⇒ Mangelndes Interesse der Organisation (am Ende waren nur noch 2 der 37 Mitarbeiter im WarRoom beim AG angestellt)
 - ⇒ Patentrecht
 - ⇒ Musterzulassung
 - ⇒ Produktkosten

Heuristische Bestätigung





7.3. Projekt Recovery nach dem War Room Konzept

8. Session 8: Unternehmensweites Architektur- und Anforderungsmanagement

Inhalt

8.1. IT-Architekturmanagement	314
8.2. From Architecture to Source Code	334
8.3. Telepathie im Automotive-Bereich	345

8.1. IT-Architekturmanagement

Jürgen Fehn

Landesamt für Zentrale Polizeiliche Dienste NRW

Abteilungsleiter 1

Schifferstr. 10

47059 Duisburg

klaus.schenk@polizei.nrw.de

Abstract

Beim Landesamt für Zentrale Polizeiliche Dienste (LZPD) werden eine Vielzahl von Anwendungen betrieben. Diese Anwendungen sind in vielen Bereichen voneinander abhängig und nutzen beispielsweise gleiche Datenbank- und/oder Betriebssysteme. Viele Anwendungen sind über Schnittstellen miteinander verbunden. Die Darstellung der Abhängigkeiten der Anwendungen untereinander wurde mit dem ARIS-Architekturmanagement realisiert.

Motivation

Beim LZPD werden ca. 150 IT-Anwendungen betrieben, dazu kommen noch ca. 290 dezentrale Anwendungen. Viele der Anwendungen sind über Schnittstellen miteinander verbunden. Die Basistechnologien, die hinter diesen Anwendungen stehen (z.B. Datenbank ORACLE oder Architekturelemente wie Java) sowie die unterstützenden Prozesse sind in vielen Bereichen gleich.

Ausgangspunkt für das IT-Architekturmanagement ist die fehlende Möglichkeit, die Zusammenhänge der gesamten Anwendungslandschaft in Form eines Wissensmanagement vorzuhalten. Wird beispielsweise eine Version von Java geändert, musste jedes Mal aufwendig geprüft werden, wer von diesen Änderungen betroffen ist. Erste Ansätze, diese Zusammenhänge in Form einer Sharepoint-Liste („IT-Masterliste“) zu erfassen, sind aufgrund der mangelnden Datenqualität gescheitert.

Die Erwartungen an ein Architekturmanagement war die Lieferung zielgruppengerechter Informationen über die existierenden IT-Anwendungen und deren Abhängigkeiten.

Ansatz

Bei der Polizei NRW werden die Geschäftsprozessanalysen mit dem Werkzeug ARIS der Fa. IDS-Scheer durchgeführt. Aufgrund des inhaltlich engen Zusammenhangs mit der IT-Architektur wurde diese Firma beauftragt, ein Werkzeug für die Darstellung der Zusammenhänge der Anwendungslandschaft (ARIS-Business Designer) zu entwickeln. Das IT-Architekturmanagement soll einen Überblick über die Anwendungslandschaft geben, insbesondere zu folgenden Fragestellungen: Welche Anwendungen gibt es? Wie interagieren die Anwendungen untereinander (Schnittstellen)? Welche Abhängigkeiten (Funktionen, Daten, Software) bestehen? Welche Geschäftsprozesse unterstützen sie? Welche Systemressourcen benötigen sie?

Ziel war es weiterhin, dass jeder IT-Verfahrensverantwortliche (IT-VV) in der Lage ist, ohne aufwendige Schulung die erforderlichen Daten selbstständig in das System einzutragen und zu pflegen. Das IT-Architekturmanagement soll die strategische Planung unterstützen, indem es Begriffe und Bezeichnungen normiert, die Standardisierung (Komponenten) und Konsolidierung (Datenobjekte, Kataloge) unterstützt sowie die Planung von Lebenszyklen und notwendigen Finanzmitteln und Ressourcen verbessert.

Bewertung

Das IT-Architekturmanagement befindet sich zur Zeit im Aufbau, es hat sich aber gezeigt, dass bereits wertvolle Unterstützung bei der jährlich zu erstellenden IT-Rahmenplanung für das Folgejahr geleistet wurde. Wichtig beim Aufbau einer solchen Architektur ist die Klärung, welche Informationen bereit gestellt werden, weil jede Information für einen Bedarfsträger eine Erfassung durch den IT-VV (und damit auch Aufwand) bedeutet. Im Weiteren werden alle Verfahren erfasst und konsolidiert. Danach werden die Verfahren auf technologische Standards und auf die Verwendung eines unternehmensweiten Datenmodells (ZIR) hin überprüft. Auf dieser Basis kann dann eine Modularisierung der Verfahren und möglicherweise der Einsatz von wieder verwendbaren Services erfolgen.

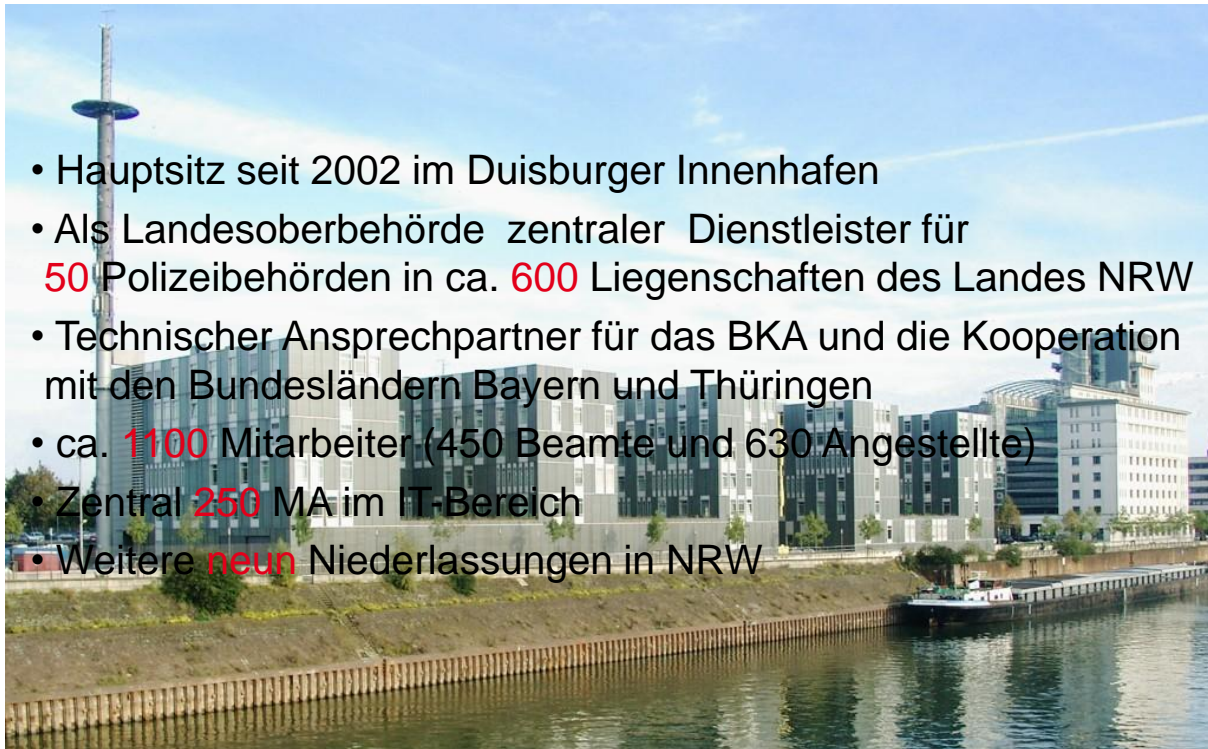


IT-Architekturmanagement



AGENDA

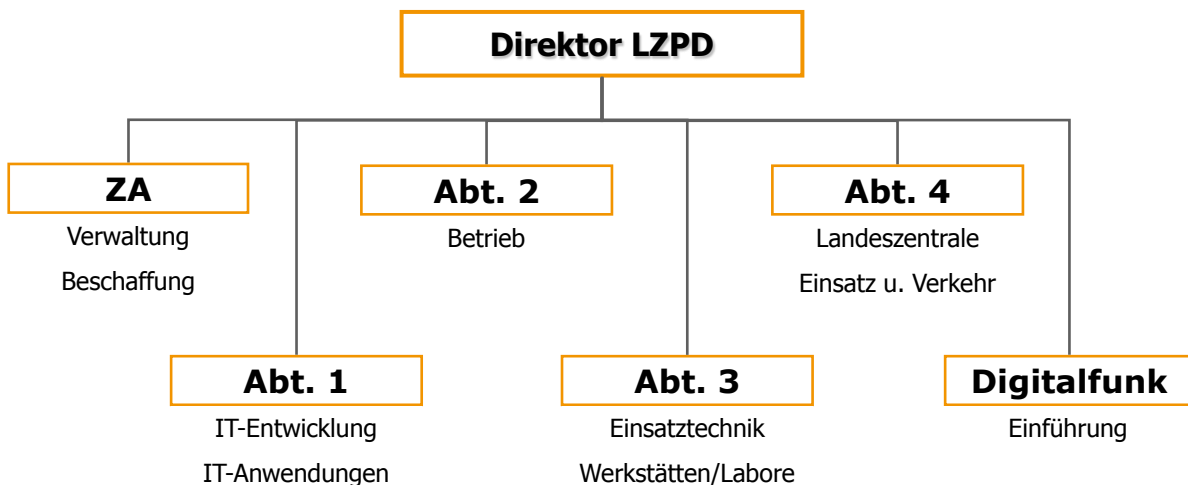
1. Das Landesamt für Zentrale Polizeiliche Dienste
2. Komplexität der polizeilichen IT-Verfahren in NRW
3. Präsentation der EAM-Lösung



- Hauptsitz seit 2002 im Duisburger Innenhafen
- Als Landesoberbehörde zentraler Dienstleister für **50** Polizeibehörden in ca. **600** Liegenschaften des Landes NRW
- Technischer Ansprechpartner für das BKA und die Kooperation mit den Bundesländern Bayern und Thüringen
- ca. **1100** Mitarbeiter (450 Beamte und 630 Angestellte)
- Zentral **250** MA im IT-Bereich
- Weitere **neun** Niederlassungen in NRW



Organigramm



8.1. IT-Architekturmanagement



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Verwaltung & Landesweite Beschaffung

Zentralabteilung



5 IT-Architekturmanagement

2006 Berlin, 27.06.2009 PD NRW



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Informationstechnologie

Abteilung 1



6 IT-Architekturmanagement

Berlin, 27.05.2009

Betrieb der IT-Technik

Abteilung 2

IT-Leitstelle



Systemüberwachung



CN-Pol



Rechenzentrum



**Operative Einsatztechnik
Werkstätten, Speziallabore
Logistik-Center**

Abteilung 3



Abteilung 4



9 IT-Architekturmanagement

2009 Berlin, 27.06.2009 PD NRW

Einführung digitaler Kommunikationstechnik

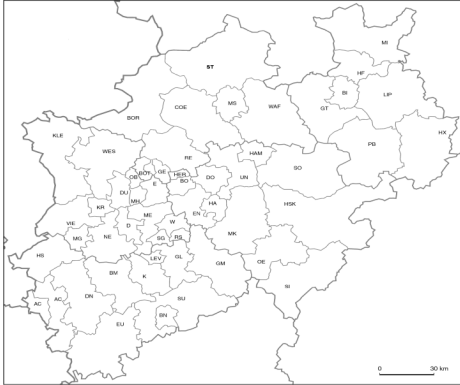
Digitalfunk



10 IT-Architekturmanagement

Berlin, 27.05.2009

Polizei Nordrhein-Westfalen



- ca. 50.000 Beschäftigte
- ca. 3 Milliarden € jährliches Haushaltsvolumen
- über 30.000 vernetzte Arbeitsplatz-PC + Sonderarbeitsplätze
- Garant für Innere Sicherheit für 8 Millionen Bürger auf einer Fläche von 34.000 km²

Kernaufgaben der Polizei:

Einsatzbewältigung
Gefahrenabwehr

Kriminalitäts-
bekämpfung

polizeiliche Verkehrs-
sicherheitsarbeit

AGENDA

1. Das Landesamt für Zentrale Polizeiliche Dienste
2. Komplexität der polizeilichen IT-Verfahren in NRW
3. Präsentation der EAM-Lösung

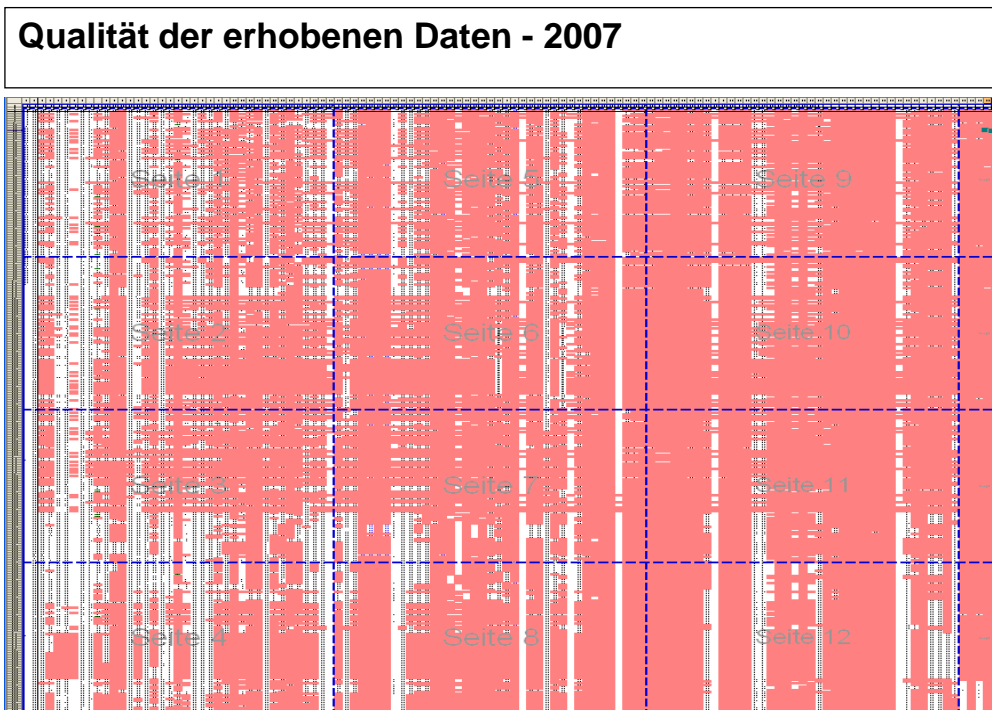
Informationssammlung erste Generation: Die „Masterliste“

Homepage | Dokumente und Listen | Erstellen | Websiteinstellungen | Hilfe Nach oben zu ZPD NRW

IT-Anwendungslandschaft
IT-Masterliste: ARIS Business Designer

|
 |
 |

Bezeichnung *	<input type="text" value="ARIS Business Designer"/> <small>Name des CI (bei Software Produktbezeichnung /Modulname, nicht des Verfahrens z.B. SP-Expert, - nicht DSM - oder PVP, -nicht IGVP-</small>
CI (Eintragsgruppe) *	<div style="border: 1px solid black; padding: 2px;"> Software ▼ Hardware Infrastruktur Nachlaufscript Software Format/Sprache/Protokoll Schnittstelle in Bearbeitung aus der Liste entfernt noch nicht eingeordnet </div> <input)="" <="" td="" type="text" value="Item"/>
Version	<input type="text" value=""/> <small>er (z.B. 7.1) ohne Servicepack oder Patch</small>
Version mit Release	<input type="text" value=""/> <small>er Releasenummer aufzuführen</small>
Hersteller / Gremium	<input type="text" value=""/> <small>auch bei Freeware Eigenentwicklung = ZPD</small>
Verfahren / Projekt	<input type="text" value="ARIS"/> <small>kann Wiederholung der Bezeichnung sein</small>
Enthaltene SP	<input type="text" value=""/> <small>hier sind !alle! erforderlichen Servicepacks einzutragen</small>
Patches	<input type="text" value=""/> <small>hier sind !alle! erforderlichen Patches einzutragen</small>
Anwendungsbereich	<input type="text" value="3.3 Sonstige unterstütz. AW"/> ▼ <small>bei Unklarheit Rückruf bei ITKO ZPD</small>
kurze Beschreibung (Anwendungsbereich)	<input type="text" value="Umfangsreduzierte Variante welches im Web-Design
dargestellt wird. Kann nur im Netz mit Zugang zum ARIS-
Server betrieben werden. Daher keine Einzelplatzlösung!!!!
Werkzeug für die Darstellung, Speicherung"/>





Erwartungen an das Architekturmanagement I

Lieferant zielgruppengerechter Informationen über die existierenden IT-Anwendungen und deren Abhängigkeiten, speziell

Überblick über die Anwendungslandschaft

- Welche Anwendungen gibt es ?
- Wie interagieren die Anwendungen untereinander (Schnittstellen)?
- Welche Abhängigkeiten (Funktionen, Daten, Software) bestehen ?
- Welche Geschäftsprozesse unterstützen sie?
- Welche Systemressourcen werden benötigt?



Erwartungen an das Architekturmanagement II

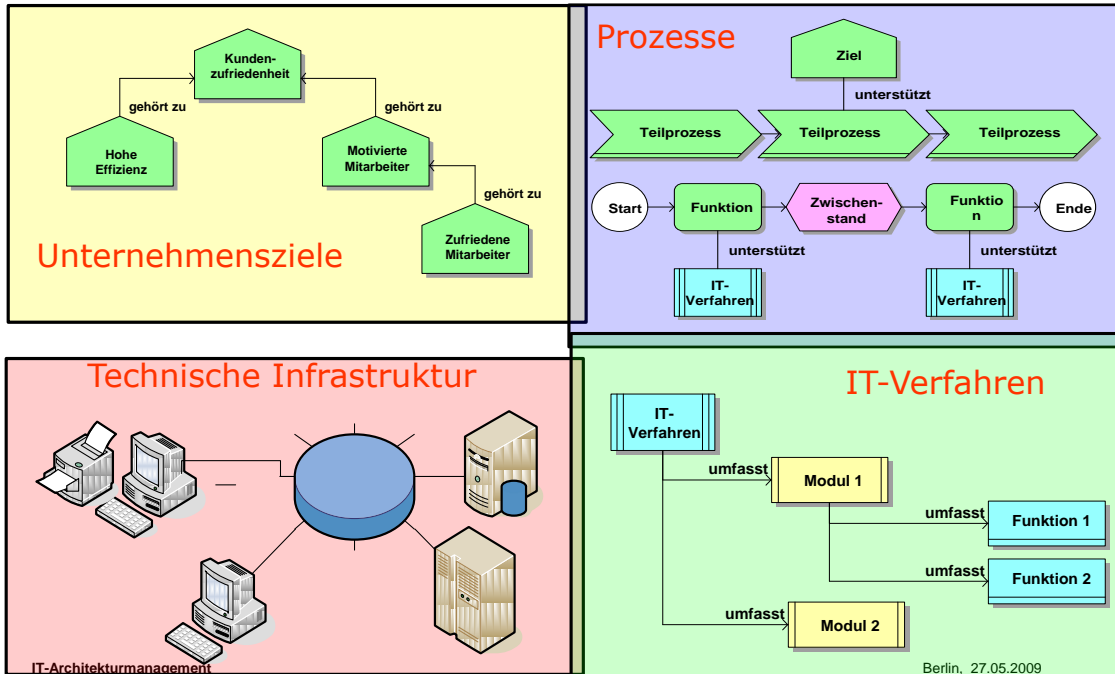
Bebauungsplan (strategische Planung)

- Normierung von Begriffen und Bezeichnungen
- Unterstützung von Standardisierung (Komponenten) und Konsolidierung (Datenobjekte, Kataloge)
- Bessere Planung von Lebenszyklen und notwendigen Finanzmittel und Ressourcen

Wissensmanagement

- Allgemeine Verfügbarkeit von „gewachsenem“ Wissen um Verfahren und Projekten

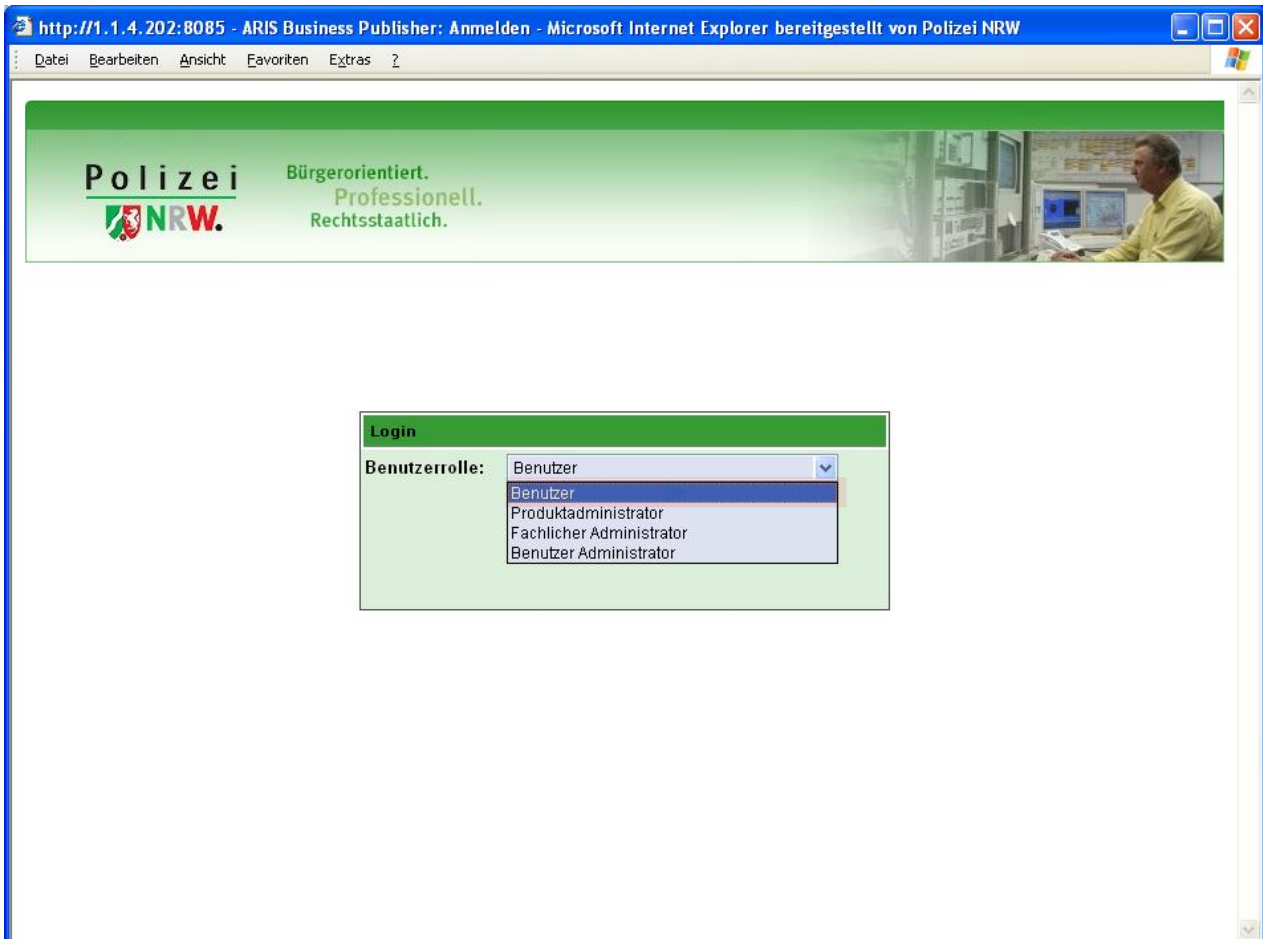
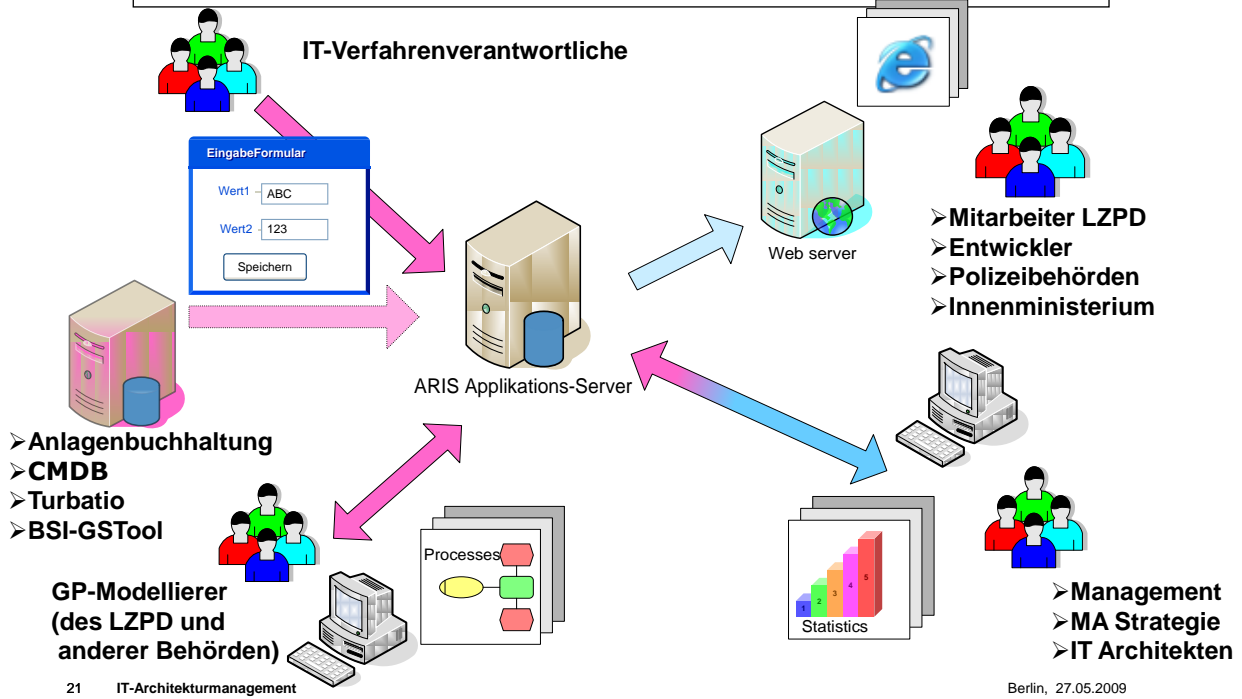
Von der Prozesslandkarte ...zu einer Anwendungsarchitektur



AGENDA

1. Das Landesamt für Zentrale Polizeiliche Dienste
2. Komplexität der polizeilichen IT-Verfahren in NRW
3. Präsentation der EAM-Lösung

Die verschiedenen Nutzer des IT-Architekturmanagements



8. Session 8: Unternehmensweites Architektur- und Anforderungsmanagement

http://1.1.4.202:8085 - LZPD EAM - Microsoft Internet Explorer bereitgestellt von Polizei NRW

Info | Hilfe | Kontakt | Abmelden

Polizei NRW. Bürgerorientiert. Professionell. Rechtsstaatlich.

IT-Systemdaten | Architektur Management | Sie sind angemeldet als: nw012959 (Benutzer / Intern)

IT-Systemübersicht

- Alle IT-Systeme (lesen)
- Meine IT-Systeme (be...

Alle IT-Systeme (lesend):

5	FINDUS	produktiv	LZPD NRW	Bernd Krusenba
52	FISPol NRW	in Entwicklung	LZPD NRW	Stefan Meiners
69	Formularmenü (Rückfallebene IGVP)	produktiv	LZPD NRW	Hilmar Pracht
66	GEMINI	in Entwicklung	LZPD NRW	Gernot Jochem
68	GESA2006	produktiv	LZPD NRW	Jürgen02 Hennir
23	GSL.net	produktiv	LZPD NRW	Frank Reimann
58	HP-UX 11.11	in Entwicklung	LZPD NRW	Thomas Höpfner
72	Ibase Version 4	in Entwicklung	Duisburg	Carsten Plitt
40	IGS-fire	produktiv	LZPD NRW	Ulrich Wallbaum
6	IGVP	produktiv	LZPD NRW	Reiner Gerard
74	Intranet 2.0	in Entwicklung	LZPD NRW	Rainer Kaliske
84	ipc - Modul Beurteilung	produktiv	LAFP	Sabine Sauer
32	IT- Verfahren im Ausland	produktiv	LZPD NRW	Harald Vetter
24	IT- Verfahren Staatsanwaltschaften	produktiv	LZPD NRW	Harald Vetter

Neues IT-System anlegen

http://1.1.4.202:8085 - LZPD EAM - Microsoft Internet Explorer bereitgestellt von Polizei NRW

Info | Hilfe | Kontakt | Abmelden

Polizei NRW. Bürgerorientiert. Professionell. Rechtsstaatlich.

IT-Systemdaten | Architektur Management | Sie sind angemeldet als: nw012959 (Systemverantwortliche(r) / Intern)

IT-Systemdetails

- Allgemein
- Phase Lebenszyklus
- Systemklassifikation
- Stammdaten
- Ansprechpartner
- Anwendungskomponenten
- Schnittstellen
- Betriebsrahmen
- Daten
- Dokumente
- Historie
- Systemdetails verlassen

Kerndaten: ARIS Business Architect, 1

IT-SystemNr. *

Bezeichnung *

Kurzbeschreibung

Systemverantwortung

Status der Standardisierung

Status des IT Systems *

Klassifikation IT Standard *

Version *

Hersteller / Lieferant

Speichern **Abbrechen** **Löschung beantragen** **Systemdaten sind aktuell**

8.1. IT-Architekturmanagement

http://1.1.4.202:8085 - LZPD EAM - Microsoft Internet Explorer bereitgestellt von Polizei NRW

Info | Hilfe | Kontakt | Abmelden

Polizei NRW Bürgerorientiert. Professionell. Rechtsstaatlich.

IT-Systemdaten | **Architektur Management** | Sie sind angemeldet als: nw012959 (Systemverantwortliche(r) / Intern)

Systemklassifikation: ARIS Business Architect, 1

Anwendungsbereich * Zielgruppenorient. AVV

Betriebssystemklasse / -typ Windows Windows XP

Datenbanksystem(e) Oracle Oracle 10g

weitere Architekturelemente Architekturframework JRE 1.5.0_12

Unterstützte Prozesse * IT-System entwickeln, Informationen sammeln und bereitstellen

Speichern Abbrechen

LZPD EAM - Microsoft Internet Explorer bereitgestellt von Polizei NRW

Info | Hilfe | Kontakt | Abmelden

Polizei NRW Bürgerorientiert. Professionell. Rechtsstaatlich.

IT-Systemdaten | **Architektur Management** | Sie sind angemeldet als: nw012959 (Benutzer / Intern)

GeDat | GEMINI | GESA2006 | GSL.net | HP DL380 | HP DL585 | HP ProLian | HP-UX 11.1 | IBase Vers | IGS-fire | IGVP

Ausstellungsbezirk RAKK fahndung

Insgesamt 318 Kataloge und 170 Formulare

unterstützt

IGVP

Programmierersprache Java

Programmierersprache C++

Objektinfo

Java (Unbenannt)

Beziehungen

ist Entwicklungssprache von	ARIS Business Architect
ist Entwicklungssprache von	ARIS Business Designer
ist Entwicklungssprache von	Intranet 2.0
ist Entwicklungssprache von	EPOST810
ist Entwicklungssprache von	APOLLO
ist Entwicklungssprache von	POLAS NRW
ist Entwicklungssprache von	IGVP
ist Entwicklungssprache von	Neues IT-System
gehört zu	Programming Languages

8. Session 8: Unternehmensweites Architektur- und Anforderungsmanagement

LZPD EAM - Microsoft Internet Explorer bereitgestellt von Polizei NRW

Info | Hilfe | Kontakt | Abmelden

Polizei NRW Bürgerorientiert. Professionell. Rechtsstaatlich.

IT-Systemdaten | **Architektur Management** | Sie sind angemeldet als: nw012959 (Benutzer / Intern)

Güterverkehr | Ausstellungsbezirk RAKK | fahndung

BAG | AUS | AZP

Insgesamt 318 Kataloge und 170

externes Meldewesen betreiben (Geschäftsprozess)

IGV

Programmiersprache: Java

Programmiersprache: C++

Objektinfo

externes Meldewesen betreiben (Unbenannt)

- wird unterstützt durch [IT-Verfahren Verbände u. freie Wirtschaft](#)
- wird unterstützt durch [Adobe Acrobat 8 Professional](#)
- wird unterstützt durch [FISPol NRW](#)
- wird unterstützt durch [Verkehrswarndienst](#)
- wird unterstützt durch [EPOST810](#)
- wird unterstützt durch [IT-Verfahren im Ausland](#)
- wird unterstützt durch [Adobe Acrobat 6 Professional](#)
- wird unterstützt durch [Adobe Acrobat 7 Standard](#)
- wird unterstützt durch [Livescan](#)
- wird unterstützt durch [Adobe Acrobat 7 Professional](#)
- wird unterstützt durch [FINDUS](#)

http://1.1.4.202:8085 - LZPD EAM - Microsoft Internet Explorer bereitgestellt von Polizei NRW

Info | Hilfe | Kontakt | Abmelden

Polizei NRW Bürgerorientiert. Professionell. Rechtsstaatlich.

IT-Systemdaten | **Architektur Management** | Sie sind angemeldet als: nw012959 (Benutzer / Intern)

GeDat | GEMINI | GESA2006 | GSL.net | GSL.net#IT-Verfahren BI | HP DL380 G4 | HP DL585 G2 | HP ProLiant ML 370 | HP-UX 11.11 | IBase Version 4 | IGS-fire | IGVP

IGVP [IT-Systemkon | IGVP [Systemkonto: | IGVP_Betriebssysteme | IGVP_Datenbanken

IGVP#FINDUS | IGVP#Neues IT-System | IGVP#PKS | IGVP#PolGIS | IGVP#SERVUS/ VUD-direkt | IGWEB | Importtabellen(neue Anl | Importtabellen(Standort | Intranet 2.0 | ipc - Modul Beurteilung | IT-Verfahren BKA | IT-Verfahren BKA#Test_ | IT-Verfahren Bund (auss | IT-Verfahren Bund (auss

Neues IT-System

IGVP

PKS | SERVUS/ VUD-direkt | FINDUS | PolGIS

8.1. IT-Architekturmanagement

http://1.1.4.202:8085 - LZPD EAM - Microsoft Internet Explorer bereitgestellt von Polizei NRW

Info | Hilfe | Kontakt | Abmelden

Polizei NRW Bürgerorientiert. Professionell. Rechtsstaatlich.

IT-Systemdaten | **Architektur Management** | Sie sind angemeldet als: nw012959 (Benutzer / Intern)

The diagram illustrates the data flow and structure within the IGVP system. It shows a central 'IGVP' component connected to an 'Open FT' component, which in turn connects to 'PolGIS'. Below the 'IGVP' component, there are several data entities represented by red boxes: 'Adresse', 'Sache', 'sonstige', 'Objekt', 'Melde-Ereignis', and 'Beziehungen'. The 'Open FT' component is connected to 'PolGIS', which is also connected to the 'Adresse' entity.

http://1.1.4.202:8085 - LZPD EAM - Microsoft Internet Explorer bereitgestellt von Polizei NRW

Info | Hilfe | Kontakt | Abmelden

Polizei NRW Bürgerorientiert. Professionell. Rechtsstaatlich.

IT-Systemdaten | **Architektur Management** | Sie sind angemeldet als: nw012959 (Benutzer / Intern)

The 'Objektinfo' window displays the following data:

Adresse (Unbenannt)	
ist Output von	Exportnach Excel
ist Output von	PolGIS
ist Output von	IGVP
ist Output von	KDVZ
ist Output von	KDVZ
ist Output von	VUD
ist Output von	Echtzeitschnittstelle e-CEBIUS
ist Output von	VOWI
ist Output von	MESTA
ist Output von	PolGIS
ist Output von	INFO
ist Output von	PKS
ist Output von	KP31 b über BKA an andere BuLä
ist Output von	Database-Link statt Schnittstelle
ist Output von	KUNO
ist Output von	ZEUS
ist Output von	KP31 b aus den BuLä über das BKA
ist Output von	PolGIS
ist Teil von	Ort

8. Session 8: Unternehmensweites Architektur- und Anforderungsmanagement

http://1.1.4.202:8085 - LZPD EAM - Microsoft Internet Explorer bereitgestellt von Polizei NRW

Info | Hilfe | Kontakt | Abmelden

Polizei NRW. Bürgerorientiert. Professionell. Rechtsstaatlich.

IT-Systemdaten | **Architektur Management** | Sie sind angemeldet als: nw012959 (Benutzer / Intern)

Objektinfo

Open FT (Unbenannt)

Beziehungen

wird verwendet von	PolGIS
wird verwendet von	GSL.net
wird verwendet von	IT-Verfahren BKA
wird verwendet von	SafeDat
wird verwendet von	IGVP
gehört zu	Basic Services and Protocols



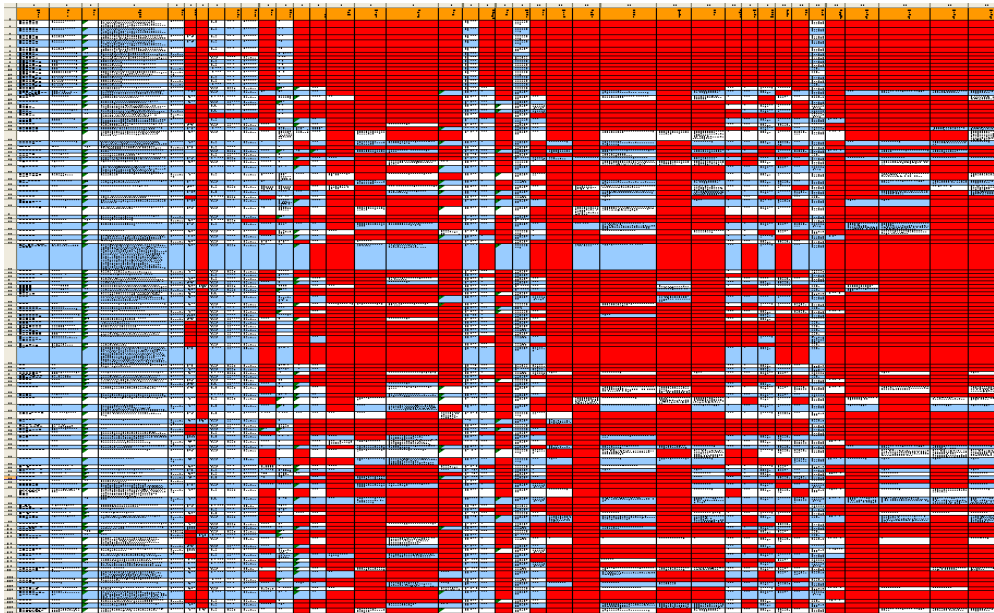
Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Qualität der erhobenen Daten - 2007



Qualität der erhobenen Daten - 2009



Ziel

Planungsgrundlage

für neue integrierte
Vorgangsbearbeitung Auswertung

für Reduzierung
Anwendungsvielfalt Betriebskosten



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Vielen Dank für die Aufmerksamkeit

Jürgen Fehn
Landesamt für zentrale polizeiliche Dienst
Tel.: 0203 41751000
Email: Juergen.Fehn@polizei.nrw.de

8.2. From Architecture to Source Code - How to Ensure Architecture Compliance in the Implemented System

Jens Knodel, Dirk Muthig
Fraunhofer Institut für Experimentelles Software Engineering (IESE)
Fraunhofer-Platz 1
67663 Kaiserslautern, Germany
{jens.knodel | dirk.muthig}@iese.fraunhofer.de

Abstract

Software architecture is the key factor for efficient communication, planning, development, maintenance, and hence, the overall success of the development project. Architecting is an upfront investment made by development organizations to assure that the resulting system(s) will meet the required quality criteria in time and effort. Among others, the architecture captures the envisioned structure of the system at development (i.e., the decomposition of the system in manageable units like components). Verifying this planned decomposition late in the lifecycle of the software system reveals - too often - that the implemented system is not compliant to specified structure. Consequently, efforts spent for architecting were made in vain because the decision and assumption made are no longer reliable and useful. To pro-actively prevent this structural decay, we propose constructive architecture compliance checking, which constantly monitors the modifications made by several (teams of) developers from day one of the implementation phase. Whenever structural violation is detected, the particular developer receives live feedback on the violation allowing its prompt removal of the violations, ensures compliance of the implemented system with the architecture, and hence, sustains the investment made into architecting¹.

Extended Abstract

Software architecture is the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution [1]. Software architectures are typically documented by a set of architectural views whereby the most popular views are the conceptual view (i.e., the most abstract view capturing the fundamental concepts), the behavioral view (describing the system at run time), structural view (describing the system at development time) or named alike. The structural view describes the static decomposition of a system in terms of layers, subsystems, components, etc., and the relationships between the various elements. The structural view is the input for a coding process performed by one or several (teams of) developers. The coding process results in source code elements like folders, packages, compilation units, routines and dependencies between them, in other words, the source code elements represent the architecture on a lower level of abstraction. Practical experiences show that the realization - the implemented system - is typically not compliant to the specification - the planned structure (e.g., see [2]). More often than not, there are unwanted or unnecessary dependencies among the source code elements

¹ This work was performed as part of the project ArQuE (Architecture-Centric Quality Engineering), which is partially funded by the German Ministry of Education and Research (BMBF) under grant number 01 IS F14.

that violate the decision made on the abstract level of the architecture. Such structural violations contribute constitute a threat to successful system development: one the one hand, structural violations contribute to long-term maintenance problems. The architecture as communication and decision vehicle is no longer reliable and useful, and the further he evolution becomes time-consuming and effort-intensive because the abstractions provided by the architecture are no longer valid. On the other hand, the removal of structural violations causes a short-term overhead effort. Because they are detected late in the development process, they require additional, unplanned effort for workshops and meetings, refactorings, re-understanding of the source code, and regression testing to avoid unwanted side effects.

We propose structural compliance checking as a constructive quality engineering activity that pro-actively prevents the introduction of structural violations (opposed to analytical compliance checking that detects them late after their introduction). Constructive compliance checking is executed constantly and continuously for every single modification made to source code. Developers receive live feedback while they are implementing the system and violations are revealed as soon as they are introduced allowing their prompt removal. The high frequency (i.e., all the time) of compliance checking turns the analytical technique into a constructive technique. The constructive architecture compliance checking is automated as a variant of the SAVE tool (SAVE stands for Software Architecture Visualization and Evaluation). SAVE is a tool for analyzing and optimizing the architecture of implemented software systems. The new variant (called SAVE LiFe where LiFe stands for Live Feedback) is fully integrated into the Eclipse development environment (IDE) and enables the live feedback for multiple developers. The developers receive compliance checking feedback in real-time as soon as a central server performed the computation of the results on the modifications they just made. First experiences in experiment and case studies provided evidence regarding the benefits of constructive compliance checking. The results showed a significant reduction of the number of violations for the supported development teams.

References

1. IEEE-Std.1471. ANSI/IEEE Std 1471-2000 - Recommended Practice for Architectural Description of Software-Intensive Systems. IEEE, New York, 2000
2. J. Knodel, M. Lindvall, D. Muthig, and M. Naab. "Static Evaluation of Software Architectures". 10th European Conference on Software Maintenance and Reengineering (CSMR), Bari, Italy, 2006

From Architecture to Source Code - How to Ensure Architecture Compliance in the Implemented System

SEE 2009, Berlin, Germany

Jens Knodel
jens.knodel@iese.fraunhofer.de

Fraunhofer IESE, Kaiserslautern, Germany

Copyright © Fraunhofer IESE 2009

May 2009



From Architecture to Source Code - How to Ensure Architecture Compliance in the Implemented System

Motivation

- **Compliance** is the state of having accomplished required or requested demands
 - Compliance is always measured relative to two distinct aspects
 - Specification, intention, or, plan vs. realization, facts, or, actual
- **Architecture compliance** means that the implemented system adheres to the abstract, specified architecture: $A_{spec} = A_{impl}$
 - Traceability between architecture and source code is ensured
 - Architectures enable efficient development and maintenance and hence, **development project success**

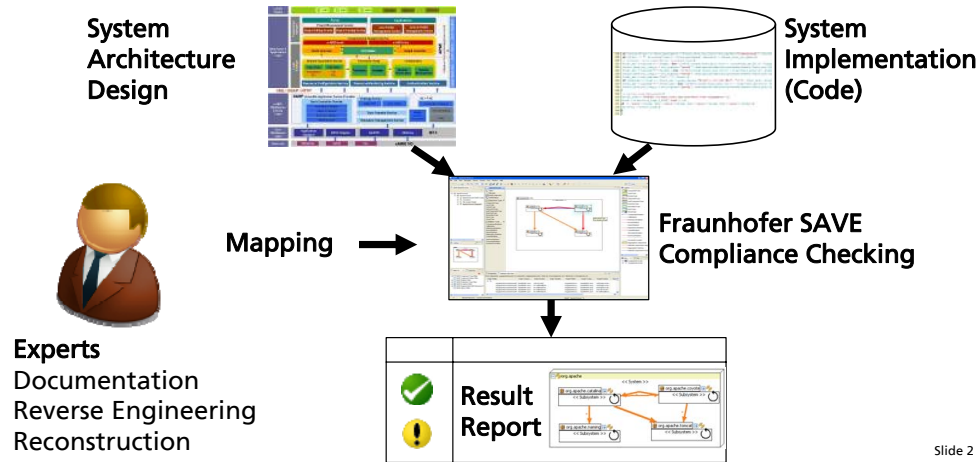
Slide 1

Copyright © Fraunhofer IESE 2009

May 2009



Compliance Checking



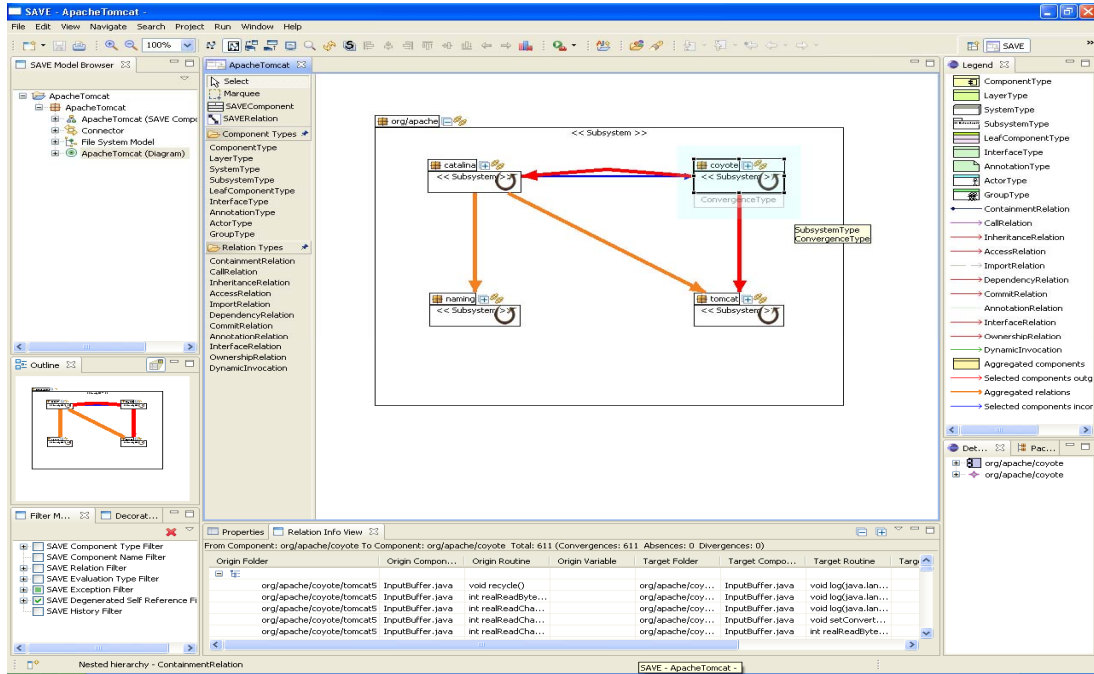
Fraunhofer SAVE



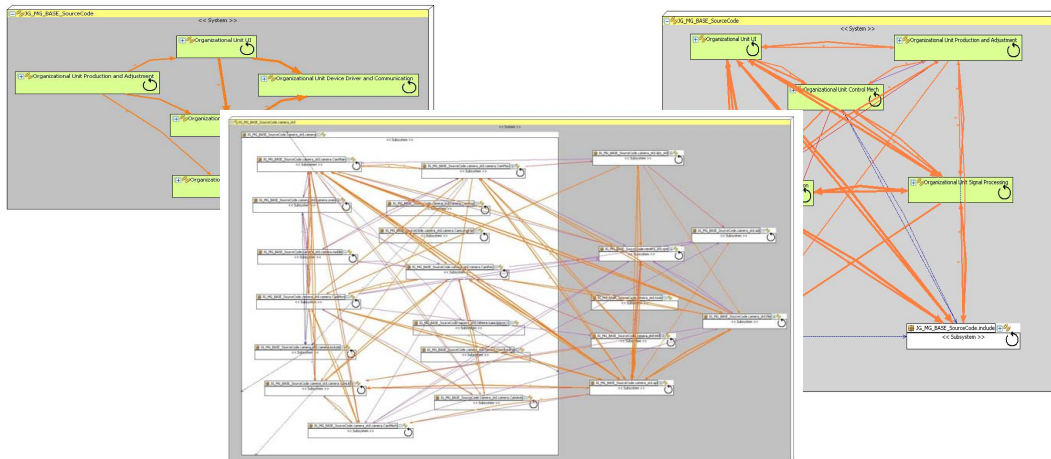
Awarded Innovation Price
Rhineland-Palatinate,
Germany 2005

- **Fraunhofer SAVE** is an Eclipse plug-in for goal-oriented analysis, compliance checking, and optimization of software architectures.
 - SAVE stands for **S**oftware **A**rchitecture **V**isualization and **E**valuation
 - SAVE is the architects' tool to visualize system artifacts in diagrams and to conduct architecture analyses
 - SAVE analyzes of existing systems implemented in Java, C/C++, Delphi, Simulink supported
- SAVE is a joint development of
 - Fraunhofer IESE (Institute for Experimental Software Engineering) in Kaiserslautern, Germany
 - Fraunhofer Center Maryland (Center for Experimental Software Engineering) in College Park, Maryland, USA

SAVE Visualization – The Eclipse Perspective



SAVE Visualization – Views on Existing Architectures



Slide 5

Motivation (Revisited)

- Practical experiences show that the implemented system is **typically not compliant** with the architecture
- Lack of compliance causes an **overhead effort** for correction
 - Workshops and meetings to discuss compliance checking results
 - Re-understanding and potential refactorings of the source code
- Lack of compliance increases the overall **maintenance effort**
 - The architecture as communication and decision vehicle is no longer reliable and useful
 - The evolution becomes time-consuming and effort-intensive
- **Prevention is easier than correction!**
 - No overhead effort and no rework required
 - Continuous and prompt feedback aims at ensuring compliance to prevent implementations to degenerate

Slide 6

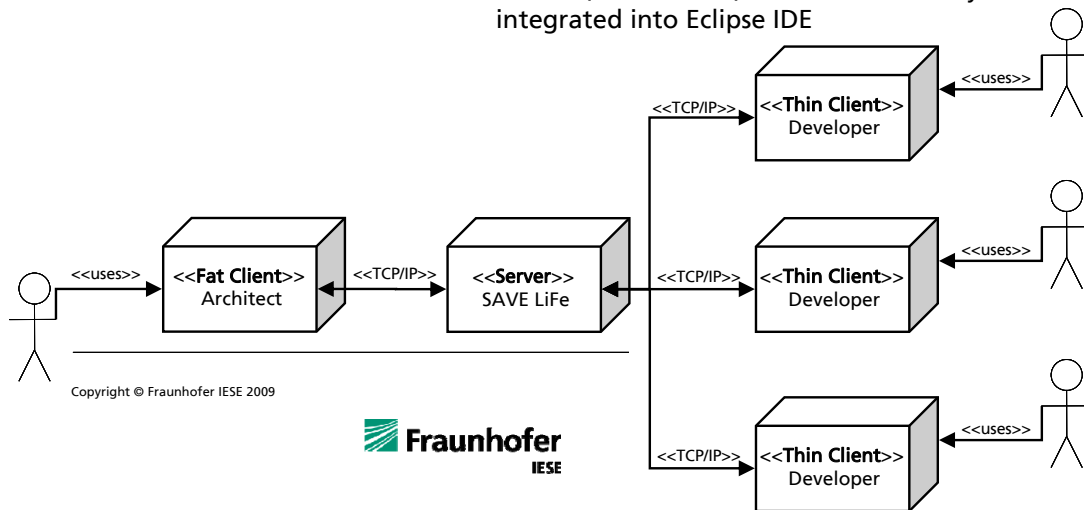
Constructive Reverse Engineering

- Exploiting reversing engineering over the whole lifetime of software the software system
 - Reverse engineering starts **on the first day** of development
 - Modifications are **instantly analyzed** and feedback is given **immediately** to developers
 - Execute the "check" phase concurrently to the "do" phase, **continuously and constantly**
 - **Live feedback** enables prompt counteractive actions in case deviations from the plan are detected
- Key idea is the **pro-active prevention of compliance decay**

Slide 7

Fraunhofer SAVE LiFe (Live Feedback)

- Client-server SAVE variant enabling constructive reverse engineering for distributed teams
 - Clients (fat and thin) and server are fully integrated into Eclipse IDE

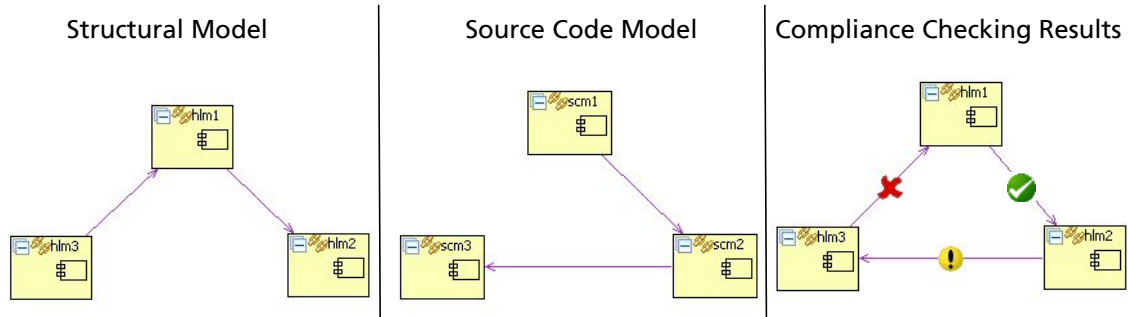


SAVE LiFe: Server

- One central server performs analyses
 - Bi-directional communication with clients (architects and developers)
- SAVE LiFe supports an extensible set of live architecture analyses
 - **Structural compliance checking** (based on reflexion models)
 - Conflict prevention agent
 - Neighborhood analysis
- Architecture analyses can be customized and extended on demand
- Analyses are executed only on the deltas submitted by the clients

Slide 9

Structural Compliance Checking based on Reflexion Models



Slide 10

Copyright © Fraunhofer IESE 2009

May 2009



SAVE LiFe: Fat Client for Architects

- Full functionality of regular Fraunhofer SAVE
 - Extended by monitor management and user administration
- Architects' client allows the definition of the structural view
- Architects can modify the structural architectural view at any time
 - Changes are immediately visible to the whole team
 - Compliance always checked against latest state of structural view
- Architects can always access the latest state of development

Slide 11

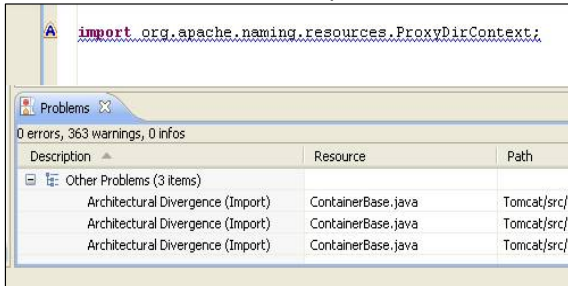
Copyright © Fraunhofer IESE 2009

May 2009



SAVE LiFe: Thin Client for Developers

Screenshot: Developers' Client



- Deltas (i.e., modifications just made) are submitted to server
 - Multiple teams of developers are supported
 - Analyses are executed on the server
- The developers receive live feedback on the compliance checking results (online and in real-time)
 - Violations are highlighted in code editor
 - Tabular list in problems
- Automated compliance checking
 - Automatically on elements in the developer's focus (i.e., elements just modified)
- Compliance checking on demand
 - On specified elements, developers manually starts analyses "by hand"

Slide 12

Copyright © Fraunhofer IESE 2009

May 2009



Experiment on Constructive Compliance Checking (1)

- Practical course, University of Kaiserslautern, 2007
 - 19 Participants in 6 component development teams (each developing one component)
 - 3 teams supported by constructive compliance checking
 - 3 teams not supported, regular development
- Students executed the complete development process
 - Architecture has been defined by the students, but no explicit role of an architect
 - Implementation, test, and integration with SAVE LiFe support (a period of 35 days)
- Approximately, the same effort was spent in both groups (by each developer)

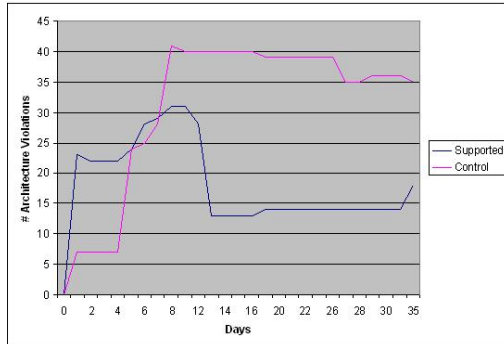
Slide 13

Knodel, Muthig, Rost: "Constructive Architecture Compliance Checking – An Experiment on Support by Live Feedback", International Conference on Software Maintenance (ICSM), Beijing, China, 2008.

May 2009



Experiment on Constructive Compliance Checking (2)



- Task results supported group
 - After an initial peak, the supported group had constantly about **60% less architecture violations**
 - The results indicate that constructive compliance checking had a positive effect
- Effort savings
 - Less workshops or meetings to communicate results required
 - 60% less items for (potential) refactoring of violations

Slide 14

Knodel, Muthig, Rost: "Constructive Architecture Compliance Checking – An Experiment on Support by Live Feedback", International Conference on Software Maintenance (ICSM), Beijing, China, 2008.

May 2009



Summary

- Constructive reverse engineering
 - Constantly and continuously executed with live feedback for developers
 - Structural compliance checking IESE's first instance of constructive reverse engineering
- Client-server platform SAVE LiFe (Live Feedback)
 - Immediate visibility of architects' decisions and support for distributed teams of developers
 - Extensible analyses platform allowing integration of customized analyses
 - Non-intrusive for developers due to smooth Eclipse integration
- Empirical results for constructive structural compliance checking are promising
 - 60% less architectural violations in controlled experiment
 - Positive feedback in ongoing industrial pilot studies

Slide 15

Copyright © Fraunhofer IESE 2009

May 2009



Thank you for your attention!

**Discussion,
Questions,
Feedback, ...**



Contact Information

Jens Knodel: jens.knodel@iese.fraunhofer.de
+49 (631) 6800-2168
Fraunhofer IESE: <http://www.iese.fraunhofer.de>

Slide 16

8.3. Telepathie im Automotive-Bereich

Hajo Hoffmann, Matthias Strößner
SOPHIST GmbH
Vordere Cramergasse 11-13
90478 Nürnberg
{hajo.hoffmann | matthias.stroessner}@sophist.de

Abstract

Die Wünsche der Kunden zu erfüllen ist auch im automobilen Umfeld das Ziel der Systementwicklung. Wissen Sie, wie Sie am besten an die bewussten, unbewussten und unterbewussten Anforderungen herankommen? Und wissen Sie wie sie für 1000de an Autofahrern das passende System schneiden? Requirements Engineering Methoden bieten Ihnen zahlreiche Alternativen und unterstützen Sie bei dem Vorhaben, die Kundenwünsche unter einen Hut zu bringen und gleichzeitig realisierbare Schritte zu gehen. Gewonnenes Wissen muss dokumentiert werden, sodass es lesbar, ausschreibbar, realisierbar und wart bar ist. An Beispielen aus unseren Projekten zeigt der Vortrag Notationstechniken und deren Anwendung im Projekt.

Die Projektrealität

Warum scheitern namhafte Projekte wie TollCollect oder was steckt hinter dem Scheitern der A-Klasse beim Elchtest. Hin und wieder bekommt man die Meldung mit, dass ein Automobilhersteller eine Rückrufaktion durchführt. Wir zeigen auf, was die Hintergründe sind, die zu solch einer Aktion führen können. Auf die wichtigsten Faktoren aus RE-Sicht gehen wir dabei ein und zeigen Lösungsansätze wie diese Riffe umschifft werden können.

Die Prinzipien im Hintergrund

Die Rahmenbedingungen der einzelnen Projekte ist unterschiedlich. Die Faktoren, von denen der Erfolg abhängt ist jedoch oft ähnlich. Oder negativ formuliert, es gibt eine Reihe von Faktoren, die zu einem Scheitern von IT Projekten führen können, wenn sie nicht beachtet werden. Diese Faktoren, wie dass das Wissen nicht auffindbar ist oder dass die Systeme immer komplexer werden sind nur zwei Beispiele die das Scheitern oder die gravierenden Verzögerungen in Projekten zu erklären.

Vergleichen sie ein Autoradio vor 15 Jahren mit einem Autoradio von heute. Wobei der Begriff Autoradio von heute schon als abwertend gilt, denn es kann weit mehr als nur Radio hören. Diese Funktionalitäten des neuen Radios müssen dokumentiert und verwaltet werden. Hier ist die Herausforderung ein gutes Requirementsmanagement aufzubauen, da es nicht trivial ist 20000 Anforderungen zu verwalten. Neben dem Verwalten müssen die Anforderungen auch gepflegt werden, d.h. es muss einen Änderungsprozess geben, sowie die Möglichkeit dies Änderung in die Spezifikation einfließen zu lassen.

Ein anderer Aspekt ist das Wissen. Dieses ist gerade bei großen Projekten sehr verteilt und oftmals nicht direkt abrufbar. Viele der Stakeholder kennen die Leistungsfaktoren oder die gerade laufenden Neuentwicklungen besser, als die klassischen und überaus

8.3. Telepathie im Automotive-Bereich

wichtigen Basisfaktoren. Vielleicht erinnert sie dies an das Kano Modell anhand dem man diese Abgrenzung erklären und skizzieren kann.


Die Beziehung des OEMs zum Kunden und Lieferanten

Ein elementarer Faktor ist die Beziehung zwischen dem OEM und dem Kunden und die, zwischen dem OEM und dem Lieferanten. Gerade die Beziehung zum Kunden gestaltet sich bei den OEMs auf dem ersten Blick als schwierig. Denn wenn sie sich überlegen, dass jeder Autobesitzer oder jeder Autofahrer seine Wünsche als Stakeholder bei einem OEM abgibt, erkennen sie schnell die Fülle an sinnvollen oder weniger sinnvollen Anforderungen, die den OEM zu kämen. Dafür gibt es einfache Methoden, wie Studien etc., um die Stakeholdermeinungen zu erfassen und zu konsolidieren. Schwieriger und vielschichtiger ist die Beziehung zum Lieferanten. Hier geht es um Vertragsmodelle, Geschäftsmodelle usw. Auch das Auslagern von Entwicklungsumfängen ist für einen OEM eine nicht zu unterschätzende Herausforderung. Probleme gerade in der Zusammenarbeit werden oftmals zu spät wahrgenommen oder nicht für wichtig genug erachtet. Frei nach dem Motto: "Wir schmeißen was über den Zaun, dann macht er das schon richtigist nicht zu empfehlen. Gerade wenn sie komplexe Komponenten ins fremdsprachige Ausland zu einem neuen Lieferanten vergeben, brauchen sie zwingend einen sehr guten Abstimmungsprozess, damit der Lieferant eine Chance auf Rückfragen hat und sie brauchen ein gutes Lastenheft, damit der Lieferant weiss was er entwickeln oder umsetzen soll.

Lösungsansätze

Nach der Schilderung dieser häufig anzutreffenden Projektsituationen, stellen wir ihnen verschiedene Lösungsansätze vor, damit sie das Dilemma vermeiden können. Dabei sind einfache Dinge oft schon ausreichend. Welchen Aggregatzustand haben die Anforderungen? Sind die Anforderungen abgestimmt und auf einer Detailierungsebene oder mischen sich noch Vision mit konkreten Realisierungsanforderungen. Haben sie die richtige Sprache gewählt oder sind die Anforderungen mehrdeutig. Diese Dinge gilt es zu beachten, wenn sie ein gutes Lastenheft schreiben wollen. Weiterhin ist es oft unerlässlich, komplexe Sachverhalte und Beziehungen in formalisierten Modellen zu beschreiben. Scheuen sie sich nicht davor auch Konzepte oder Ideen zu spezifizieren, die noch nicht komplett ausgereift vorliegen. Wenn das bewusst und strukturiert passiert und der Zulieferer einen Rückkanal für Fragestellungen bekommt, ist das eine sehr gute Möglichkeit, Wissen zu ermitteln und zu transportieren.

Der Einsatzzeitpunkt für die verschiedenen Lösungsansätze ist sehr früh in einem Projekt. Er beginnt in der Anforderungsermittlung und der Anforderungsdokumentation. Viele der Probleme hätten vermieden werden können oder wären nur abgeschwächt aufgetreten, wenn frühzeitig auf durchdachtes RE eingegangen wäre. Gerade dort legen sie den Grundstein für ihr Projekt. Wenn der wackelt oder schief ist, wird eine Korrektur, um so teurer, je später sie den Fehler bemerken.

open minded...
SOPHIST GROUP 

Telepathie im Automotive-Bereich

Ein Erfahrungsbericht ...

SOPHIST GROUP
Vordere Cramergasse 13
90478 Nürnberg

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

www.sophist.de
Hajo.Hoffmann@sophist.de
Matthias.Stroessner@sophist.de

SOPHIST GROUP | Telepathie | SEE - 2009

Wer sind wir?



- > Trainer & Berater
- > Seit Winter 2005 bei den SOPHISTen
- > Überwiegend im Automotive Umfeld tätig
- > RE & RM einführen und optimieren
- > Schulungs- und Workshop-Konzepte für RE/RM

- > Trainer & Berater
- > Seit Herbst 2007 bei den SOPHISTen
- > Überwiegend im Automotive Umfeld tätig
- > RE (Coaching, Methodenentwicklung)
- > RM (Prozessentwicklung, Strukturkonzeption)



Was macht SOPHIST?



REQUIREMENTS ENGINEERING & MANAGEMENT	UML 2 GLASKLAR	AGILE SOFTWARE-ENTWICKLUNG	BASISWISSEN REQUIREMENTS ENGINEERING <i>Neu!</i>	SYSTEMANALYSE KOMPAKT	AGILITY KOMPAKT <i>Neuaufgabe</i>
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
4. Auflage seit Oktober 2006 im Handel	3. Auflage seit August 2007 im Handel	seit Februar 2002 im Handel	ab März 2009 im Handel	2. Auflage seit März 2008 im Handel	2. Auflage ab April 2009 im Handel
ISBN: 978-344-640-5097	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

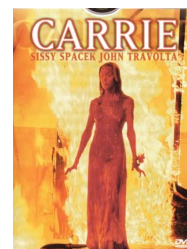


SOPHIST GROUP

Telepathie

SEE - 2009

Um was es nicht geht



SOPHIST GROUP

Telepathie

SEE - 2009

Die grundsätzlichen Probleme ...



1. Komplexes System
2. Chaotische Spezifikation
3. Viele Anforderungen



Die grundsätzlichen Probleme ...



1. Komplexes System
2. Chaotische Spezifikation
3. Viele Anforderungen

1. Sinnvolle Schneidung
2. Abstraktionsebenen nutzen
3. Workshops
4. Abstimmung von Details



Das Menschenn ...



1. Internationale Teams
2. Verschiedene Ausdrucksfähigkeit
3. Vielfältiger Hintergrund



Die grundsätzlichen Probleme ...



1. Internationale Teams
2. Verschiedene Ausdrucksfähigkeit
3. Vielfältiger Hintergrund

1. Templates
2. Glossar
3. Ermittlungstechniken



Und das RM ...



1. Keine Neuentwicklungen
2. Hohe Wiederverwendung, Redundanzgefahr
3. Lange Produkt- und Projektlaufzeiten



Und das RM ...



1. Keine Neuentwicklungen
2. Hohe Wiederverwendung, Redundanzgefahr
3. Lange Produkt- und Projektlaufzeiten

1. Common-Modell
2. Traceability
3. Change Management



Zu guter Letzt...



- Es
- Kon
- kon
- Exis
- anw

Die drei wichtigsten Maßnahmen:

1. Eine zentrale RE-Methoden Abteilung etablieren
2. Intensiver Kontakt zwischen RE und Entwicklung
3. Knowledge Management aufbauen



REQUIREMENTS DAYS

20-22 Oktober 2009, München

Die führende Konferenz zu Requirements Engineering im deutschsprachigen Raum.
Themenschwerpunkte 2009: "Useability" & "Requirements Tools"

Meet & greet & listen to:

- Keynote: Professor Larry Constantine - University of Madeira
- Dörthe Kretschmer - Capgemini sd&m AG
- Dr. Jean Hani - Roche Diagnostics GmbH
- Bernd Körner - Deutsche Telekom AG
- Dr. Stefan Walburg - Bundesagentur für Arbeit
- Martin Künzle & Tobias Stenzel - Siemens AG Healthcare
- Chris Rupp - SOPHIST GmbH
- Peter Hruschka - The Atlantic Systems Guild

Sichern Sie sich den Frühbucher-Rabatt von 1.150,00 € bis 04.09.2009

Anmeldung unter: www.redays.de

9. Session 9: Agiles Projektmanagement

Inhalt

9.1. Tool-gestütztes Vorgehen nach Scrum mit Projektron BCS	354
9.2. Multiprojektmanagement mit dem Projectserver	362
9.3. Gezähmte Papiertiger: Mit Tailoring eine Methode für das Projekt nützlich machen	382

9.1. Tool-gestütztes Vorgehen nach Scrum mit Projektron BCS

Roland Petrasch¹, Robert Kalweit², Florian Fieber³

¹FB VI Informatik und Medien

TFH Berlin

Luxemburger Str. 10

13353 Berlin

petrasch@tfh-berlin.de

²Game Duell GmbH

Ackerstr. 14-15

10115 Berlin

robert.kalweit@gameduell.de

³qme Software

Gustav-Meyer-Allee 25

13355 Berlin

florian.fieber@qme-software.de

Einleitung

Die zunehmende Bedeutung agiler Prozesse für Software-Entwicklungsvorhaben im Allgemeinen und für das Projektmanagement (PM) bei Scrum im Speziellen 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 Burndown Chart (Restaufwand) kaum manuell erstellen können oder wollen. Dabei muss sich ein Projektmanagement-Tool quasi "minimalinvasiv" in die durch die Beteiligten geprägten agilen Projekte und Methoden einbringen. Ein simples Überstülpen eines klassischen Projektmanagementansatzes über ein agiles Vorhaben würde wenig zielführend sein. Die Akzeptanz im Team und damit der Gesamterfolg wären gefährdet (vgl. [8]).

Motiviert durch die Tatsache, dass viele Teams in der Praxis agil vorgehen, wurde anhand von Scrum [12] untersucht, inwiefern die Projektmanagement-Software Projektron BCS einsetzbar ist und welche Anpassungen notwendig sind, um einen erfolgreichen Einsatz zu ermöglichen. Dabei sollte neben einem Konzept auch ein konkreter Einsatz mit einer bereits angepassten Version von Projektron BCS stattfinden, damit frühzeitig noch offene Punkte geklärt werden können.

Da bei agilen Vorhaben das Team stark in das Projektmanagement eingebunden ist, z.B. beim Thema Aufwandsschätzung, beziehen sich die Betrachtungen nicht nur auf die Rolle des Projektleiters, sondern auf das gesamte agile Team. Daher sind die hier vorgestellten Erkenntnisse für alle Teammitglieder eines Projektes relevant - getreu der agilen Idee vom „whole Team“.

Scrum und Tooleinsatz

Scrum-Prozess

Scrum kam als weit verbreitetes agiles Verfahren beim zu analysierenden Praxispartner seit einigen Jahren zum Einsatz, wobei jedoch individuelle Anpassungen vorgenommen wurden. Um die Abläufe besser verstehen zu können, erfolgt u.a. eine prozessorientierte Analyse. Als Beispiel dient hier der Ausschnitt aus einem Prozessmodell mittels UML-Activity-Diagramm (s. Abbildung 9.1).

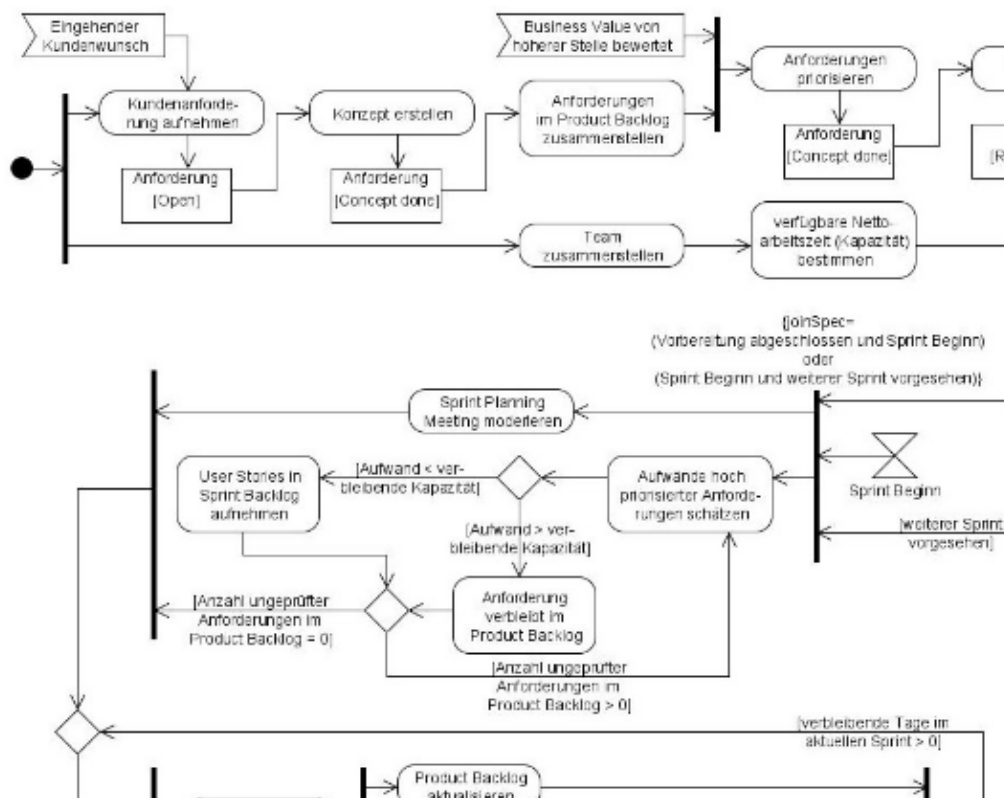


Abbildung 9.1.: Scrum-Prozess als UML-Activity-Diagramm (Ausschnitt)

Die Prozesse bildeten dann als Scrum-Vorgehensmodell, auf dessen Basis sich Verbesserungen definieren und konkrete Projekte instanziiieren lässt.

Scrum - eine kurze Einführung

Anforderungserhebung und -management werden in Scrum mit dem Product Backlog realisiert. Alle bekannten funktionalen und nicht funktionalen Anforderungen von Kunden und Benutzern, Management, und Entwicklern sind hier aufgelistet. Es empfiehlt sich die Verwendung von User Stories [2], da diese das gemeinsame Verständnis der Anforderungen aller Projektbeteiligten forcieren. Das Product Backlog gehört dem Product Owner, einer der drei Rollen in Scrum. Der Product Owner vertritt die Interessen sämtlicher Stakeholder des Projekts. Er priorisiert die Anforderungen im Product Backlog und stellt auf diese Weise sicher, dass die richtigen und wichtigsten Anforderungen umgesetzt werden (vgl. [8]).

Die zweite Rolle in Scrum ist der ScrumMaster. Aufgabe des ScrumMasters ist es, für die Einhaltung und den möglichst reibungslosen Ablauf des Scrum-Prozesses zu sorgen. Er

9.1. Tool-gestütztes Vorgehen nach Scrum mit Projektron BCS

steht daher neben dem Product Owner und dem Team und unterstützt sie bei der Erfüllung Ihrer Verantwortlichkeiten, hilft ihnen also sich selbst zu managen. Das Team ist in Scrum die dritte und letzte Rolle. Ein Scrum Team ist interdisziplinär besetzt, weswegen keine weitere Unterscheidung erfolgt. Obwohl etliche Rollen, wie z.B. Architekten und Entwickler, im Team vertreten sind, fordert Scrum, sich über ein kompromissloses Verständnis dieser Rollen hinwegzusetzen. Was zählt, ist die enge Zusammenarbeit und das Erreichen des Projektziels.

Das Product Backlog stellt die Grundlage für die Release Planung dar. Die Priorisierung sowie grobe Schätzungen der einzelnen Anforderungen helfen bei einer sinnvollen Gruppierung in Releases. Zur Aufwandsschätzung gibt es zahlreiche Methoden wie z.B. Planning Poker, die Verwendung abstrakter Aufwandsschätzungen mit Story Points, oder das Zurückgreifen auf historische Daten (vgl. [8]).

Die Umsetzung der Anforderungen, die in einem Release enthalten sind, erfolgt iterativ in sogenannten Sprints. Dabei handelt es sich um kurze (meist auf 5-30 Tage begrenzte) Zeiträume, in denen das Team potenziell lieferbare Software produziert. Dies ist einer der essenziellen Bestandteile von Scrum: Am Ende eines jeden Sprints sollte ein Teil der Anforderungen komplett fertig gestellt und vollständig getestet sein. Die für ein Release vorgesehenen Anforderungen werden im Sprint Planning Meeting für einen bestimmten Sprint eingeplant. Das auf diese Weise entstandene Sprint Backlog wird innerhalb des Sprints nicht mehr verändert.

Die Summe der Aufwände aller User Stories entspricht der im Sprint zu leistenden Arbeit. Diese Arbeit wird innerhalb des Teams regelmäßig synchronisiert. Beim Daily Scrum, einem täglichen, viertelstündlichen Meeting, schildert jedes Teammitglied schildert, was es am seit dem letzten Daily Scrum erreicht hat, woran es plant, bis zum nächsten Daily Scrum zu arbeiten und ob es Störungen im Arbeitsablauf gibt, derer sich der Scrum Master annehmen muss.

Das Team aktualisiert täglich die verbleibende Arbeit an einer User Story. Der Restaufwand sollte kontinuierlich sinken. Ist das nicht der Fall, stimmt etwas im Projekt nicht - ein Indikator dafür ist das Burndown Chart [8]. Es bietet eine täglich aktuelle Sicht auf die verbleibende Arbeit im Sprint oder Release. Trotz Schwankungen (die Restaufwände können auch steigen) sollte das Burndown Chart einen klaren Abwärtstrend aufweisen. Die Trendlinie stellt die Entwicklungsgeschwindigkeit dar und ermöglicht die Vorhersage des ungefähren Datums der Fertigstellung der Anforderungen im Sprint oder Release.

Bugs begegnet Scrum mit zwei Regeln: Werden innerhalb eines Sprints an der Umsetzung einer User Story Bugs festgestellt, sollten diese sofort beseitigt werden, da eine auch nur minimal fehlerhafte Umsetzung als nicht abgeschlossen gilt. Die Abnahme einer User Story erfolgt im Sprint Review Meeting. Wird erst danach in einem abgenommenen Feature ein Bug gefunden, sollte dieser als neue User Story ins Product Backlog aufgenommen und im folgenden Sprint priorisiert bearbeitet werden (vgl. [8]).

Neben den Planning- und Review Meetings und dem Daily Scrum stellt die Sprint Retrospektive das vierte Meeting innerhalb des Scrum-Prozesses dar. Die Retrospektive, moderiert vom ScrumMaster, dient der regelmäßigen Prozess-verbesserung. Sowohl Product Owner als auch das gesamte Team sind eingeladen, ihre Sicht der Dinge zu schildern und Vorschläge einzubringen.

Tool-Unterstützung für Scrum-Projekte

Sowohl Sprint- und Releaseplanung, als auch Sprint-Durchführung und die Abnahme der Anforderungen können durch Projektmanagement-Tools unterstützt werden. Beim Daily Scrum können je nach Arbeitsumgebung Burndown Chart und eine Übersicht der offenen User Stories visualisiert werden. Ein Tool-Einsatz ist hier noch denkbar. Bei der Retrospektive dient das Projektmanagement-Tool als Informationslieferant, der der Historie eines Projektes aufzeigt.

Mittlerweile sind zahlreiche Tool für Scrum-Projekte verfügbar, z.B. Acunote, Banana Scrum, Taskboard oder tinyPM (eine Liste findet sich unter [13]). Eine Evaluation ergab allerdings, dass bei den meisten Tools mehr oder weniger viele Funktionen für das Projektmanagement nach Scrum fehlten. So war teilweise die Release-Planung eingeschränkt, keine Multiprojektmanagement umsetzbar oder die Einbeziehung von externen Beteiligten mit entsprechenden Rechten nicht möglich.

Im Ergebnis fiel die Wahl auf Projektron BCS, weil hier ein leistungsfähiges web-basiertes Projektmanagement die Basis liefert und Erweiterungen für Scrum zur Verfügung stehen. Im Folgenden sei die Scrum-Unterstützung kurz beschrieben.

Tool-Beispiel Projektron BCS

Projektron BCS ermöglicht Anforderungsmanagement in Product Backlogs. Hier werden, parallel zu Projekten, Anforderungen gepflegt und können einem Sprint in einem Projekt zugeordnet werden. Ein solcher Sprint wird durch einen neuen Subtyp einer Aufgabe repräsentiert, der Anforderungen enthalten kann und Zeiterfassung ermöglicht. Darüber hinaus können Teammitglieder im Aufwandsplan für einen Sprint zu einem Teil ihrer Arbeitszeit verplant werden (s. Abbildung 9.2). Werden dabei die Grundlast oder weitere Aufgaben eines Teammitglieds nicht berücksichtigt, führt dies zu Konflikten in der Ressourcenauslastung. Auf diese Weise kann man in Projektron BCS mit einem Sprint nicht nur sicherstellen, dass Mitarbeiter nicht überlastet werden, sondern ermöglicht auch Teilzeit-Scrum-Projekte.

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

Abbildung 9.2.: Aufwandsplan eines Scrum-Projekts

Anforderungen stellen einen neuen Objekttyp in Projektron BCS dar. Dieser ordnet sich

9.1. Tool-gestütztes Vorgehen nach Scrum mit Projektron BCS

inhaltlich zwischen Aufgaben und den bereits existierenden Tickets ein. Während Anforderungen anfänglich hierarchisch in verschiedenen Product Backlogs verwaltet werden, stellt die Gesamtheit aller bereits einem Sprint zugeordneten Anforderungen das jeweilige Sprint Backlog dar. Dabei werden nicht nur die initiale Hierarchie, sondern auch eventuell bereits festgehaltene abstrakte Aufwandsschätzungen [2] übernommen. Es wird zwischen vier verschiedenen Anforderungstypen unterschieden: Backlog Items, Epics, User Stories und Akzeptanztests. Während User Stories die eigentlichen Anforderungen beschreiben, stellen Akzeptanztests Abnahmekriterien in Form von Testszenarien dar, die erfolgreich ablaufen müssen. Ein Epic ist eine noch zu spezifizierende Anforderung, die User Stories enthalten kann, während ein Backlog Item, wie in Scrum, eine Gruppierung von Anforderungen repräsentiert (s. Abbildung 9.3). Da verschiedene Anforderungen diversen Prozessschritten dienen, ist es möglich, jeweils unterschiedliche Lebenszyklen zu definieren.

Sind Anforderungen einem Sprint zugeordnet worden, kann auf diese auch Zeiterfassung erfolgen. Das ermöglicht nicht nur Auswertungen mit höherer Granularität, sondern auch die Erfassung von bspw. Konzeptions- oder Managementaufwänden für ein Backlog Item.

Strukturplan Voreinstellung ▾ Anpassen ...

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

Abbildung 9.3.: Strukturplan eines Scrum-Projekts

Die Summe aller abstrakten Aufwände der Anforderungen innerhalb eines Sprints stellt die Grundlage für das Burndown Chart in Projektron BCS dar (s. Abbildung 9.4). Dabei werden nur offene Anforderungen, die noch nicht ihren finalen Status erreicht haben,

berücksichtigt.

Nach Abschluss eines Sprints wird die Summe der Aufwände sämtlicher als abgeschlossen geltender Anforderungen erfasst. Diese Zahl wird anschließend als Entwicklungsgeschwindigkeit gespeichert und sollte die Grundlage für die Planung der zukünftigen Sprints darstellen.

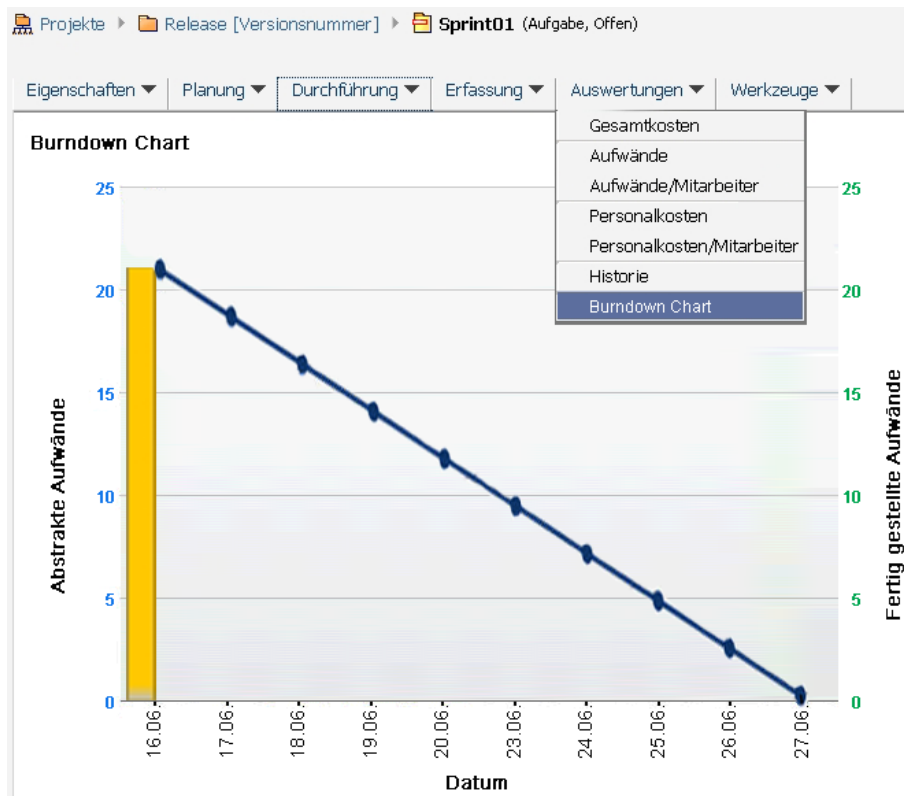


Abbildung 9.4.: Burndown Chart in Projektron BCS

Um die Teamplanung Scrum-konform zu ermöglichen (s. Abbildung 9.5), wird das Rechtesystem von Projektron BCS angepasst werden. Die Rollen Teammember (TM), Product Owner (PO) und ScrumMaster (SC) werden eine freie Konfiguration der Benutzerrechte der Scrum-Rollen gewährleisten, ohne dass die Durchführung von Nicht-Scrum-Projekten beeinträchtigt wird.

Auch weitere Aspekte beim Software-Entwicklungsprozess konnte das Beispielprojekt untersuchen, auf die allerdings nicht weiter eingegangen wird; z.B. fanden im Rahmen der Qualitätssicherung Retrospektiven statt [3] und beim Risikomanagement wurden Risiken im Vorfeld beachtet und bei der Planung entsprechend berücksichtigt [5].

Bewertung und Ausblick

Die Nutzung eines Projektmanagement-Tools macht auch bei agilen Vorhaben Sinn. Besonders bei umfangreichen Projekten ist eine rein manuelle Erstellung und Pflege der Informationen für das Projektmanagement nicht zu rechtfertigen. Auch bei verteilten Teams erscheint ein Werkzeug unabdingbar. Im Rahmen eines Forschungsprojektes, an dem u.a. die Firma Projektron GmbH und die Beuth Hochschule Berlin beteiligt waren, konnte anhand eines konkreten Scrum-Projektes die Einsatzfähigkeit von Projektron BCS

9.1. Tool-gestütztes Vorgehen nach Scrum mit Projektron BCS

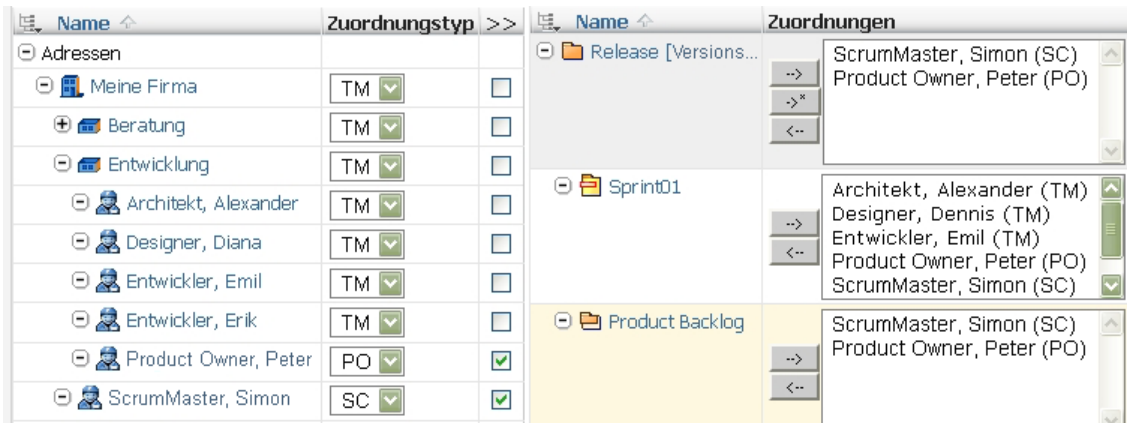


Abbildung 9.5.: Teamplanung mit Scrum-Rollen

für Scrum nachgewiesen werden. Entsprechende Anforderungen ließen sich einerseits durch Customizing der bestehenden Anwendung und andererseits durch Programmierung von Erweiterungen umsetzen.

Dies führte beispielsweise dazu, dass Tickets zu Aufgaben „konvertiert“ werden können, was besonders bei Scrum im Falle der Planung von Sprints wichtig ist: Der Sprint-Backlog besteht aus einzelnen Anforderungen, die sich als Tickets abbilden lassen und dann im Team als Arbeitsaufgaben bearbeitet werden.

Wichtige Erkenntnisse gab es auch bei der Akzeptanz des Tooleinsatzes: Der Wiedererkennungseffekt, bei dem die bekannten Techniken und Darstellungen auch in der Software abgebildet bzw. unterstützt werden, fördert die nahtlose Einführung, z.B. in Form des Burndown-Charts.

Trotz aller „Lichtblicke“ im Bereich Agilität und Projektmanagement sind noch weitere Anstrengungen notwendig, um die Professionalisierung des Software-Engineerings voranzutreiben. Bspw. erfordert die modellbasierte (oder modellgetriebene) Software-Entwicklung neue und geänderte Aufgaben, welche in den verschiedenen Vorgehensweisen bisher kaum Beachtung finden [10]. So setzt sich ein typisches modellbasiertes Projekt aus zwei getrennten Teilprojekten zusammen - dem eigentlichen Kundenprojekt und einem Generator- oder „Infrastrukturprojekt“ (Entwicklung des Generators) [9]. Modellbasierte Entwicklung erfordert aus unserer Sicht ein spezielles agiles Projektmanagement, um effektiv eingesetzt werden zu können. Bspw. geht die Entwicklung von Generatoren mit fortlaufenden Tests und notwendigen Anpassungen am Metamodell bzw. den Transformationen einher und wird von zahlreichen Rückkopplungen der Iterationen untereinander begleitet. Hier ist ein entsprechend modifizierter Scrum-basierter Ansatz zu entwickeln.

Ebenfalls zu untersuchen ist, wie Scrum in großen Projekten mit mehreren, verteilten Teams zum Einsatz kommt. Multiprojektmanagement und insbesondere die Ressourcenplanung und -auslastung sind dabei von Interesse. Wichtig auch hier ist, dass die Teams den entscheidenden Vorteil von Scrum bei ihrer Arbeit erhalten: Die Agilität.

Literaturverzeichnis

1. Angermeier, Georg: Projektmanagement-Lexikon. Projekt Magazin, 2005 <http://www.projektmagazin.de/glossar/index.html>. - ISBN 3000181148

2. Cohn, Mike: User Stories Applied: For Agile Software Development. Addison-Wesley, 2004. - ISBN 0321205685
3. Derby, Esther ; Larsen, Diana: Agile Retrospectives: Making Good Teams Great. Pragmatic Programmers, 2006. - ISBN 0977616649
4. DIN: DIN 69901-69905: Projektwirtschaft. Deutsches Institut für Normung, 1987-2000
5. Gaulke, Markus: Risikomanagement in IT-Projekten. Oldenbourg Wissenschaftsverlag GmbH, 2004. - ISBN 3486275992
6. Greenleaf, Robert K.: Servant Leadership: A Journey into the Nature of Legitimate Power and Greatness. 25th anniversary ed. New York : Paulist Press, 2002. - ISBN 0809105543
7. Oestereich, Bernd ; Weiss, Christian: APM - Agiles Projektmanagement. Erfolgreiches Timeboxing für IT-Projekte. Dpunkt Verlag GmbH, 2008. - ISBN 3898643867
8. Pichler, Roman: Scrum - Agiles Projektmanagement erfolgreich einsetzen. Dpunkt. Verlag GmbH, 2007. - ISBN 3898644782
9. Petrasch, Roland; Fieber, Florian; Bednarz, Maciej: Agile modellbasierte Softwareentwicklung in der Praxis. In: Objekt-Spektrum 2/2008. SIGS-DATACOM Verlag, 2008
10. Petrasch, Roland; Meimberg, Oliver: Model-Driven Architecture - eine praxisorientierte Einführung. Dpunkt Verlag 2006
11. PMBoK: A Guide to the Project Management Body of Knowledge: Dritte Ausgabe. Project Management Institute (PMI), 2004. - ISBN 1930699727
12. Schwaber, Ken: Agile Project Management with Scrum. Microsoft Press, 2004. - ISBN 073561993X
13. Gloger, Boris: Blog <http://scrum4you.wordpress.com/2009/03/01/scrum-tools-list>

9.2. Multiprojektmanagement mit dem Projectserver

Klaus Schenk

Landesamt für Zentrale Polizeiliche Dienste NRW

Abteilung 1, Sachgebiet Multiprojektmanagement / Zentrales Projektbüro

Schifferstr. 10

47059 Duisburg

klaus.schenk@polizei.nrw.de

Abstract

Beim Landesamt für Zentrale Polizeiliche Dienste (LZPD) werden eine Vielzahl von Projekten durchgeführt. Um eine Einheitlichkeit bei der Projektdurchführung zu erhalten, wurde im Jahr 2003 ein Zentrales Projektbüro, später dann auch ein Multiprojektmanagement eingerichtet. Als technische Lösung zur Unterstützung dieser Arbeit wurde MS-Projectserver ausgewählt.

Motivation

In allen Projekten gibt es Projektgeschäftsführer, die u.a. für die Aufgabenüberwachung innerhalb des Projektes sowie die Dokumentenablage verantwortlich sind. Diese Aufgaben wurden in einem Zentralen Projektbüro (ZPB) gebündelt, um Synergieeffekte zu erzielen.

Ausgangspunkt für das Multiprojektmanagement (MPM) war das Projekt ErWin (Erneuerung Windows). Ziel dieses Projektes war es, aus ca. 60 einzelnen Domänen bei der Polizei NRW unter Windows NT eine einheitliche Domäne polizei.nrw.de zu schaffen. Hierzu wurde ein zentraler Projektplan erstellt, wobei die Umsetzung in jeder Behörde wiederum jeweils ein Teilprojekt mit eigenem Projektplan war. Zur Koordinierung dieses Projektes (Gesamtprojektplan und 60 Teilprojektpläne) wurde MS-Projectserver 2003 beschafft. Der Einsatz von MS-Project war so überzeugend, dass auch alle anderen Landes- und Strategieprojekte ihre Projektpläne auf dem Projectserver anlegen und pflegen.

Diese beiden Aufgaben wurden in einem Sachgebiet innerhalb der Abteilung 1 (IT-Anwendungen) des LZPD gebündelt, da in dieser Abteilung die Mehrzahl aller Projekte durchgeführt wird.

Ansatz

Als technische Lösung für das MPM wurde MS-Projectserver 2003 ausgewählt. Die Projektpläne werden auf dem zentralen Server angelegt und gepflegt. Es gibt damit tatsächlich nur noch einen Plan, auf den die ProjektmitarbeiterInnen über Project-Webaccess lesenden Zugriff haben. Ein Versand eines Projektplans ist damit nicht erforderlich, die Lizenzkosten halten sich in Grenzen.

Neben Projectserver wurden auch die Sharepointservices 2.0 eingeführt. Zu jedem Projekt wird automatisch eine Projekt-Homepage erzeugt, auf die nur die Projektmitglieder Zugriff erhalten. Hier wurde eine Dokumentenbibliothek -Struktur angelegt, die für alle Projekte einheitlich ist. Außerdem wird damit sichergestellt, dass ein Zugriff auch behördenübergreifend möglich ist (kein lokales Fileshare mehr).

Für Projekte wird eine Aufgabenüberwachung durchgeführt. Da nicht alle Aufgaben in einem Projekt in dem Projektplan geführt werden, wird hierzu eine Sharepointliste angelegt. Die Terminüberwachung erfolgt durch das ZPB. Die Statusberichte der einzelnen Projekte werden monatlich abgefragt und in Form eines Multiprojektstatusberichtes dem Behördenleiter, quartalsweise dann auch dem Innenministerium vorgelegt.

Bewertung

Die bisherigen Erfahrungen machen deutlich, dass durch die Einrichtung dieser Arbeitsrate beim LZPD Synergieeffekte erzielt werden konnten. Dies betrifft zum einen eine Personaleinsparung, da die „Geschäftsführertätigkeiten“ mehrerer Projekte von einem Mitarbeiter des ZPB übernommen werden können. Durch die erlangte Handlungssicherheit sind keine Schulungsmaßnahmen erforderlich. Die MA des ZPB beraten und unterstützen den Projektleiter bei der Einrichtung von Projekten.

Auf das vollständige und projektübergreifende Ressourcenmanagement wurde zunächst verzichtet. Das Ressourcenmanagement wird aber mit der Einführung von Projectserver2007 als weiterer Schritt eingeführt. Mit Veröffentlichung der Projektmanagementrichtlinie für die Polizei NRW im Juni 2008 sind alle IT-Projekte auf dem Projectserver zu führen, alle anderen Projekte sind, ab einer gewissen Bedeutung, dem Multiprojektmanagement bekannt zu geben.

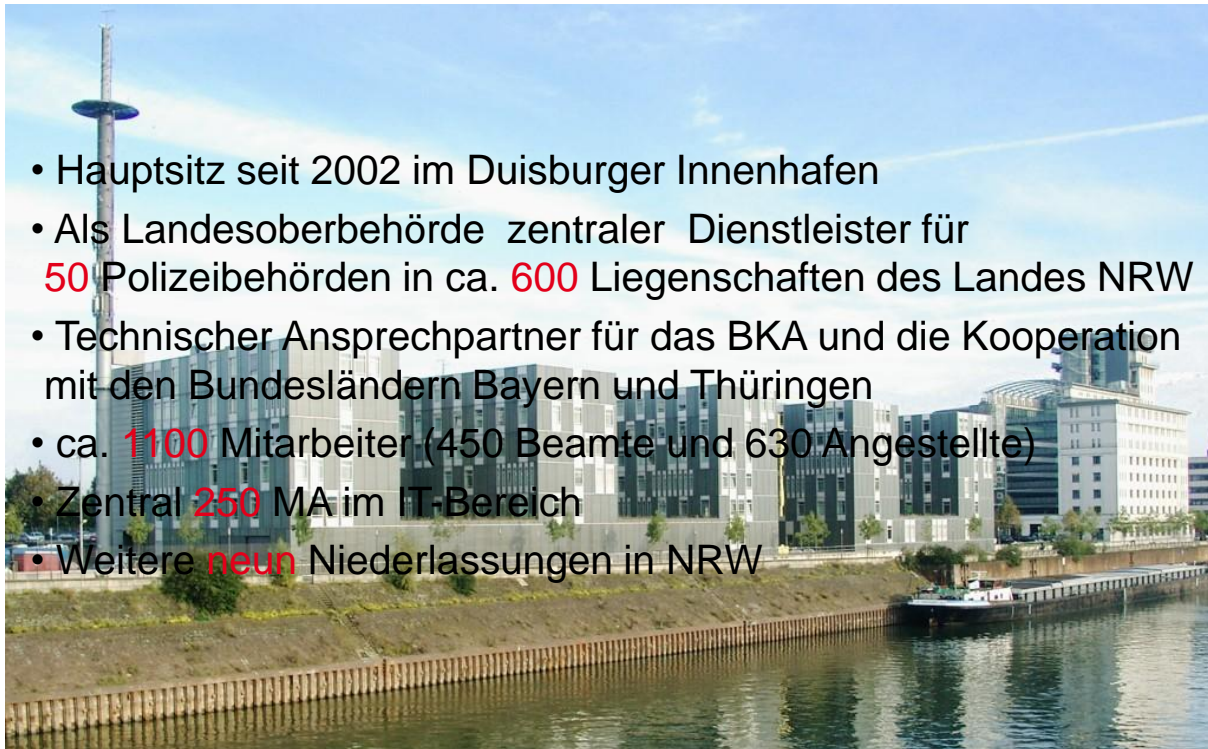


Multiprojektmanagement mit dem Projectserver



AGENDA

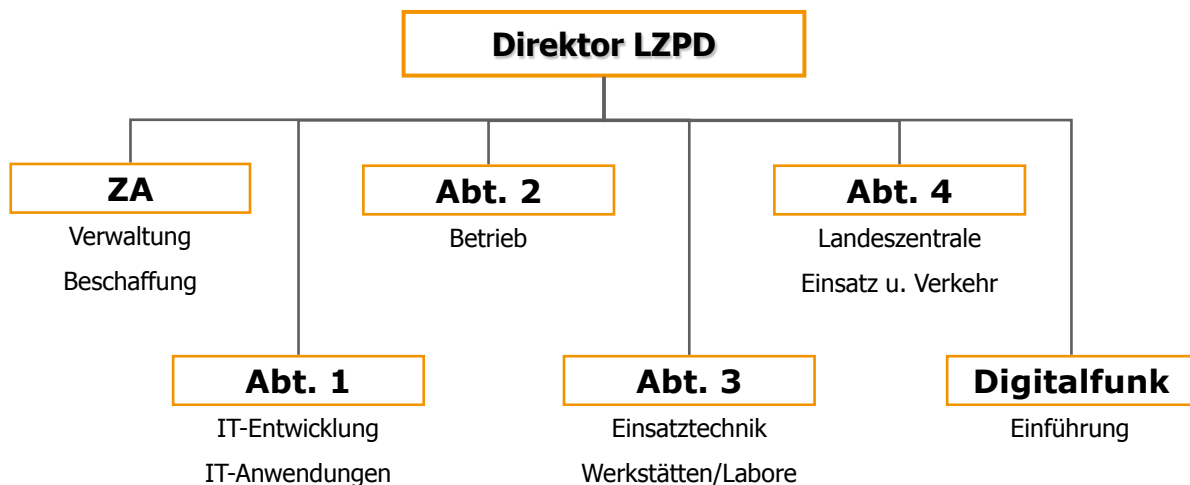
1. Das Landesamt für Zentrale Polizeiliche Dienste
2. Ausgangssituation und Zielsetzung
3. Präsentation des Vorgehens bei der Polizei NRW
4. Fazit



- Hauptsitz seit 2002 im Duisburger Innenhafen
- Als Landesoberbehörde zentraler Dienstleister für **50** Polizeibehörden in ca. **600** Liegenschaften des Landes NRW
- Technischer Ansprechpartner für das BKA und die Kooperation mit den Bundesländern Bayern und Thüringen
- ca. **1100** Mitarbeiter (450 Beamte und 630 Angestellte)
- Zentral **250** MA im IT-Bereich
- Weitere **neun** Niederlassungen in NRW



Organigramm



9.2. Multiprojektmanagement mit dem Projectserver



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Verwaltung & Landesweite Beschaffung

Zentralabteilung



5 Multiprojektmanagement

Berlin, 27.05.2009 - Pressestelle ZPD NRW



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Informationstechnologie

Abteilung 1



6 Multiprojektmanagement

Berlin, 27.05.2009



Betrieb der IT-Technik

Abteilung 2

IT-Leitstelle



Systemüberwachung



CN-Pol



Rechenzentrum



**Operative Einsatztechnik
Werkstätten, Speziallabore
Logistik-Center**

Abteilung 3





Abteilung 4



Zentrale Informationsstelle
Sporteinsätze



Kräftekoordinierung



Fliegerstaffel



Landesleitstelle

9 Multiprojektmanagement

Berlin, 27.05.2009 Pressestelle ZPD NRW



Einführung digitaler Kommunikationstechnik

Digitalfunk



10 Multiprojektmanagement



Berlin, 27.05.2009



Polizei Nordrhein-Westfalen



- ca. 50.000 Beschäftigte
- ca. 3 Milliarden € jährliches Haushaltsvolumen
- über 30.000 vernetzte Arbeitsplätze-PC + Sonderarbeitsplätze
- Garant für Innere Sicherheit für 8 Millionen Bürger auf einer Fläche von 34.000 km²

Kernaufgaben der Polizei:

Einsatzbewältigung
Gefahrenabwehr

Kriminalitäts-
bekämpfung

polizeiliche Verkehrs-
sicherheitsarbeit



AGENDA

1. Das Landesamt für Zentrale Polizeiliche Dienste
2. Ausgangssituation und Zielsetzung
3. Präsentation des Vorgehens bei der Polizei NRW
4. Fazit



Ausgangssituation

- > Durchführung einer Vielzahl von (IT-) Projekten
- > jedes Projekt hat eigene Geschäftsführung
- > keine einheitliche IT-Unterstützung für das Projektmanagement

- > Besondere Herausforderung
Projekt ErWin (**E**rneuerung **W**indows)
 - Auflösung von 60 Einzeldomänen
 - Errichtung einer Domäne *polizei.nrw.de*
 - Installation von ca. 30.000 APC mit Windows XP
 - Behördenübergreifendes Projekt



Zielsetzung

Einrichtung eines Zentralen Projektbüros

- > Vereinheitlichung der Durchführung von Projekten
- > Übernahme der „Geschäftsführertätigkeit“
- > Synergieeffekte

Für das Projekt „ErWin“

- > Überblick über ein Gesamtprojekt mit 60 Teilprojekten
- > Erprobung einer Projektmanagement-Software



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



AGENDA

1. Das Landesamt für Zentrale Polizeiliche Dienste
2. Ausgangssituation und Zielsetzung
- 3. Präsentation des Vorgehens bei der Polizei NRW**
4. Fazit

15 Multiprojektmanagement

Berlin, 27.05.2009



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Projekt „ErWin“

- > Unterstützung des Projektes mit dem Projectserver 2003
- > pro Behörde (60) einen Projektplan
- > Übersicht über alle Projekte bei der Gesamtprojektleitung

- > Vorgefertigte Projektpläne in jeder Behörde

- > Definierte Ampelansichten in Project-Webaccess
 - für Projekt-Übersicht
 - für jedes Einzelprojekt

- > im Laufe des Projektes Verzicht auf Statusberichte

16 Multiprojektmanagement

Berlin, 27.05.2009

9.2. Multiprojektmanagement mit dem Projectserver



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Projekt „ErWin“

Ampelübersicht über alle Projekte

Ampel / Proje	Projektname	APC Soll	APC Ist	▲ Anfang	Ende	Zust
●	Ampel / Projekt <u>luckung</u>	0	0	08.12.2004	18.08.2005	
●	Rollout_DSO123_Münster	0	0	08.12.2004	20.04.2006	
●	02.Rollout.Höxter	0	0	28.01.2005	25.02.2005	
●	01.Rollout.Euskirchen	0	0	01.03.2005	01.03.2005	
●	03.Rollout.Essen	2450	0	01.03.2005	28.07.2005	
●	ErWinRollout_PP_A-Stadt	670	230	21.03.2005	26.04.2005	
●	05.Rollout.Siegen	930	0	01.04.2005	01.04.2005	
●	SIS_Testprojekt_01	0	0	01.04.2005	07.04.2005	
●	04.Rollout.Aachen	1900	0	01.04.2005	01.04.2005	
●	SIS_Testversion_01	0	0	01.05.2005	01.05.2005	
●	02.Rollout.Steinfurt	810	0	02.05.2005	02.05.2005	
●	08.Rollout.Dortmund	2750	0	02.05.2005	02.05.2005	
●	Schulungesplan	0	0	04.07.2005	15.12.2005	
●	ErWin_PP_Köln	0	0	11.07.2005	21.02.2006	
●	ErWin_ZPD	0	0	18.07.2005	22.09.2005	

17 Multiprojektmanagement

Berlin, 27.05.2009



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Projekt „ErWin“

Ampelansicht Einzelprojekt

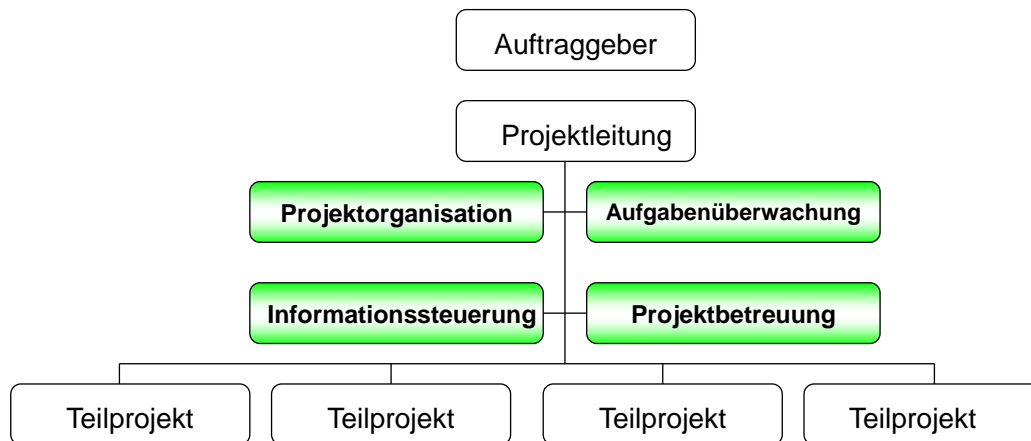
▲ Nr.	Ampel / Vorgi	Vorgangname	Arbei	Geplante Arbeit	Abweichung Arbeit	Aktuelle
0		ErWinRollout_PP_A-Stadt:Veröffentlicht	10Std.	10Std.	0Std.	
1	●	PP Teststadt	10Std.	10Std.	0Std.	
2	●	Server	1Std.	1Std.	0Std.	
3	●	Netzwerk	1Std.	1Std.	0Std.	
4	●	PI 1	2Std.	2Std.	0Std.	
5	●	Wache 1	1Std.	1Std.	0Std.	
6	●	Wache 2	1Std.	1Std.	0Std.	
7	●	PI 2	2Std.	2Std.	0Std.	
8	●	Wache 1	1Std.	1Std.	0Std.	
9	●	Wache 2	1Std.	1Std.	0Std.	
10	●	PI 3	2Std.	2Std.	0Std.	
11	●	Wache 1	1Std.	1Std.	0Std.	
12	●	Wache 2	1Std.	1Std.	0Std.	
13	●	PI 4	2Std.	2Std.	0Std.	
14	●	Wache 1	1Std.	1Std.	0Std.	
15	●	Wache 2	1Std.	1Std.	0Std.	

18 Multiprojektmanagement

Berlin, 27.05.2009

Zentrales Projektbüro

- > Besetzung mit projekterfahrenen Mitarbeitern
- > Beschreibung eines Aufgaben-/Leistungskataloges



Zwischen-Ergebnis

Projekt „ErWin“

- > anfängliche Akzeptanzschwierigkeiten
- > Wichtig: Schulung und Betreuung
- > Arbeitserleichterungen wurden aber erkannt
- > der Microsoft- Projectserver hat sich bewährt

Zentrales Projektbüro

- > erwartete Synergieeffekte sind eingetreten
- > Gegenseitige Vertretung innerhalb des Zentralen Projektbüros
- > auch hier: Einsatz des Microsoft- Projectservers positiv

- > Standard zur Projektdurchführung hat sich etabliert



Querschnittsaufgaben im Dezernat 12:

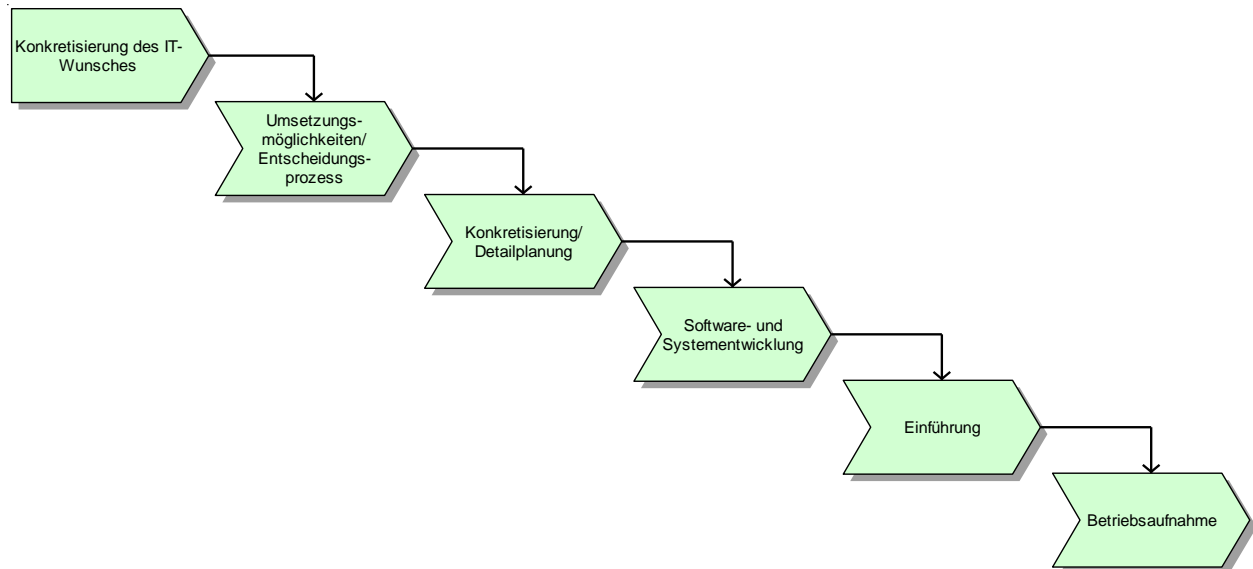
- > Mitarbeit bei der Festlegung der IT-Strategie
- > Mitarbeit bei Fortschreibung der Projektrichtlinie
- > Organisationspezifische Anpassung des V-Modell XT



Aufgabenverteilung im Dezernat 12

- SG 12.1 (Anforderungs- und Changemanagement)
- SG 12.2 (Zentrales Projektbüro, Multiprojektmanagement)
- SG 12.3 (Geschäftsprozessanalyse)

Prozesskette „Vom Wunsch zur Wirklichkeit“



Technische Unterstützung

Einsatz des Projectserver 2007, Project-Webaccess und den Sharepointservices 3.0

- > Nutzung von Projekt-Vorlagen
- > Einheitliche Ablagestruktur für Projekte
- > Führung von Aufgabenlisten in Sharepoint
- > Führung von Risikolisten in Sharepoint
- > Erstellung von Führungsansichten und Projektportfolios
- > Erstellung von Multiprojekt-Statusberichten

9.2. Multiprojektmanagement mit dem Projectserver



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Projekt-Vorlage

Vorgang	Vorgangname	Dauer
1	Projektmanagementaufgaben	1 t?
2	Vorhabensantrag	1 t?
6	Lastenheft	1 t?
10	Wirtschaftlichkeitsbetrachtung1	1 t?
13	Projektauftrag	1 t?
17	Detaillierete Projektplanung	1 t?
21	Prozessanalyse	1 t?
27	Pflichtenheft	1 t?
35	Wirtschaftlichkeitsbetrachtung2	1 t?
38	Risikomanagement	1 t?
44	Schulungskonzept	1 t?
48	Betriebskonzept	1 t?
54	Sicherheitskonzept	1 t?
58	Verfahrensverzeichnis	1 t?
62	Projektstatusberichte	1 t?
67	Projektbesprechungen	1 t?
72	Projektdurchführung	1 t?
73	Arbeitspaket1	1 t?
75	Arbeitspaket2	1 t?
77	Arbeitspaket3	1 t?

25 Multiprojektmanagement

Berlin, 27.05.2009



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Einheitliche Dokumentenablage

Dokumente
■ Projektakte
■ Projektleitung
■ Projektbüro
■ Informationen
■ Schriftverkehr

01 Projektakte	
Offizielle Projektdokumente	
Typ	Name
📁	01_Projektantrag
📁	02_Projektauftrag
📁	03_Projekteinrichtungsverfügung
📁	04_Projekthandbuch
📁	05_Projektplan
📁	06_Projektabschluss
📁	07_Verfahrensverzeichnis
📁	08_Betriebskonzept
📁	09_Wirtschaftlichkeitsbetrachtung
📁	10_Statusberichte
📁	11_Verträge
📁	12_Haushalt
📁	13_Lastenheft
📁	14_Pflichtenheft
📁	15_IT-Sicherheitskonzept

26 Multiprojektmanagement

Berlin, 27.05.2009



Auftragsliste

Projekt Intranet > Auftragsliste

Auftragsliste

Liste für Arbeitsaufträge im Projekt (Sortiert nach Offen, Erledigt, Wartend und Überschreibung) -->>

Nr.	Jahr	Ordner	Zu erledigende Aufgaben	Status	Verantwortl.	Termin	Bereich	Auftraggeber	Erfassung
Jahr : 2007 (30)									
Jahr : 2008 (74)									
Jahr : 2009 (40)									
001	2009	PL	Nachfrage ZA 2 zur Haushaltsabwicklung in 2008	erledigt	Kathrin	15.01.2009	Leitung	Kathrin	14.01.2009
002	2009	PL	Bericht an IM zum Sachstand	erledigt	Kathrin	14.01.2009	Leitung	Kathrin	14.01.2009
003	2009	PL	Problembeschreibung A 2	erledigt	Kathrin	15.01.2009	Leitung	Kathrin	14.01.2009
004	2009	Link	Bearbeitung Stellungnahme Fallo	erledigt	Kathrin	19.02.2009	Leitung	Kathrin	28.01.2009
005	2009	Link	Dankanschriften für TPJL an deren Behörden	erledigt	Kathrin	15.02.2009	Leitung	Kathrin	02.02.2009
006	2009	Link	Interview-Anfrage	erledigt	Kathrin	15.02.2009	Öffentlichkeitsarbeit	Kathrin	03.02.2009
007	2009	Link	Antrag Freschaltung USB u. Laufwerke	erledigt	-	13.02.2009	Sonstiges	Kathrin	03.02.2009
008	2009	Link	Konzept "Foren" erstellen	offen	Maat	16.04.2009	Fachlichkeit	Kathrin	05.02.2009
009	2009	Link	Benutzungshinweise "Foren" erstellen	offen	Maat	16.04.2009	Fachlichkeit	Kathrin	05.02.2009
010	2009	Link	Nutzungsregelungen "Teamsite" erarbeiten	offen	Maat	16.04.2009	Fachlichkeit	Kathrin	05.02.2009
011	2009	Link	Entwurf eines Redaktionsstatus erarbeiten	erledigt	Maat	27.03.2009	Fachlichkeit	Kathrin	05.02.2009
012	2009	Link	Fachliches Administrationskonzept erarbeiten	offen	Maat	16.04.2009	Realisierung	Kathrin	05.02.2009
013	2009	Link	Architekturkonzept erstellen	offen	Maat	16.04.2009	Realisierung	Kathrin	05.02.2009
014	2009	Link	Definition Standardentwicklungswerkzeuge	erledigt	Maat	19.03.2009	Realisierung	Kathrin	06.02.2009
015	2009	Link	Definition Standards Styleguide	offen	Maat	16.04.2009	Realisierung	Kathrin	06.02.2009
016	2009	Link	Konzept BAO-Portal	offen	Maat	16.04.2009	Fachlichkeit	Kathrin	09.02.2009
017	2009	Link	Klärung der Handlungszuständigkeit für Schulung	erledigt	Kathrin		Leitung	Kathrin	09.02.2009
018	2009	Link	Trinationale Intranetplattform in der	erledigt	Kathrin	06.03.2009	Fachlichkeit	Kathrin	13.02.2009
019	2009	Link	vertrags- und haushaltsrechtliche Prüfung der Rahmenvereinbarung	erledigt	Kathrin	27.02.2009	Sonstiges	Kathrin	18.02.2009
020	2009	Link	Entscheidung über Layout-Entwurf	wartend	Kathrin	16.04.2009	Fachlichkeit	Kathrin	18.02.2009
021	2009	Link	Übersendung NCP-Client-Lizenzen an ZA	erledigt	Kathrin	06.03.2009	Assistenz	Kathrin	20.02.2009
022	2009	Link	Berichterstellung wegen Probleme bei der Übergabe an den Betrieb	erledigt	Kathrin	06.03.2009	Leitung	Kathrin	20.02.2009
023	2009	Link	Genehmigung Layout barrierefrei	wartend	Kathrin	16.04.2009	Leitung	Kathrin	24.02.2009
024	2009	Link	Vorlage Lastenheft Version 1.0 an den Auftraggeber	erledigt	Kathrin		Leitung	Kathrin	02.03.2009
025	2009	Link	Beschaffung Netzwerkkomponenten zur Anbindung an das CNPol	wartend	Kathrin	16.04.2009	Assistenz	Kathrin	02.03.2009
026	2009	Link	Aufgabenwahrnehmung innerhalb des LZPD	offen	Kathrin	16.04.2009	Leitung	Kathrin	06.03.2009

27 Multiprojektmanagement

Berlin, 27.05.2009



Risikomanagement

Überblick > Risiken

Risiken

Verwenden Sie die Risikoliste, um eine Gruppe von Risiken in diesem Projekt zu verwalten. Sie können Risiken zuordnen, Prioritäten festlegen und den Fortschritt der Risiken von Anfang bis Ende nachverfolgen.

ID	Titel	Zugeordnet zu	Status	Gefahrenpotenzial	Kategorie	Fällig am	Teilprojekt
Anzahl = 15							
Teilprojekt : MVL (10)							
Anzahl = 10							
4	Geeignetes Netz steht nicht zur Verfügung		(1) Aktiv	10	Technik		MVL
14	Verzögerung durch Vergabebeschwerden		(1) Aktiv	8	Technik		MVL
3	Abfluss der Haushaltsmittel erfolgt nicht im Zeitraster der VE		(1) Aktiv	7,2	Finanzen		MVL
7	unerwartete Kosten durch Anforderungen		(1) Aktiv	6,4	Finanzen		MVL
15	Verzögerungen durch Entwicklungs- und Realisierungsaufwand bei AN		(1) Aktiv	6,4	Personal		MVL
8	Ressourcenengpässe		(1) Aktiv	5,6	Personal		MVL
5	Testabnahme führt zu negativem Ergebnis		(1) Aktiv	2,7	Technik		MVL
13	kein geeignetes Angebot		(1) Aktiv	2,7	Technik		MVL
6	Monopolstellung eines Anbieters		(1) Aktiv	2,4	Finanzen		MVL
9	Auftragnehmer erhält keine Zertifizierung		(1) Aktiv	1	Leistung		MVL
Teilprojekt : Schulung (5)							
Anzahl = 5							
17	Schulungen beim können nicht oder nicht im geplanten Umfang gestartet werden		(1) Aktiv	10			Schulung
12	Behördenschulung nicht oder nicht im geplantem Umfang		(1) Aktiv	6,4	Technik		Schulung
11	Kein ausreichend qualifiziertes Schulungspersonal		(1) Aktiv	1,4	Personal		Schulung
10	Kein Schulungspersonal		(1) Aktiv	1	Personal		Schulung
16	Schulungsinhalte können nicht zielgerichtet und zeitgerecht erarbeitet werden		(1) Aktiv	1	Technik		Schulung

28 Multiprojektmanagement

Berlin, 27.05.2009

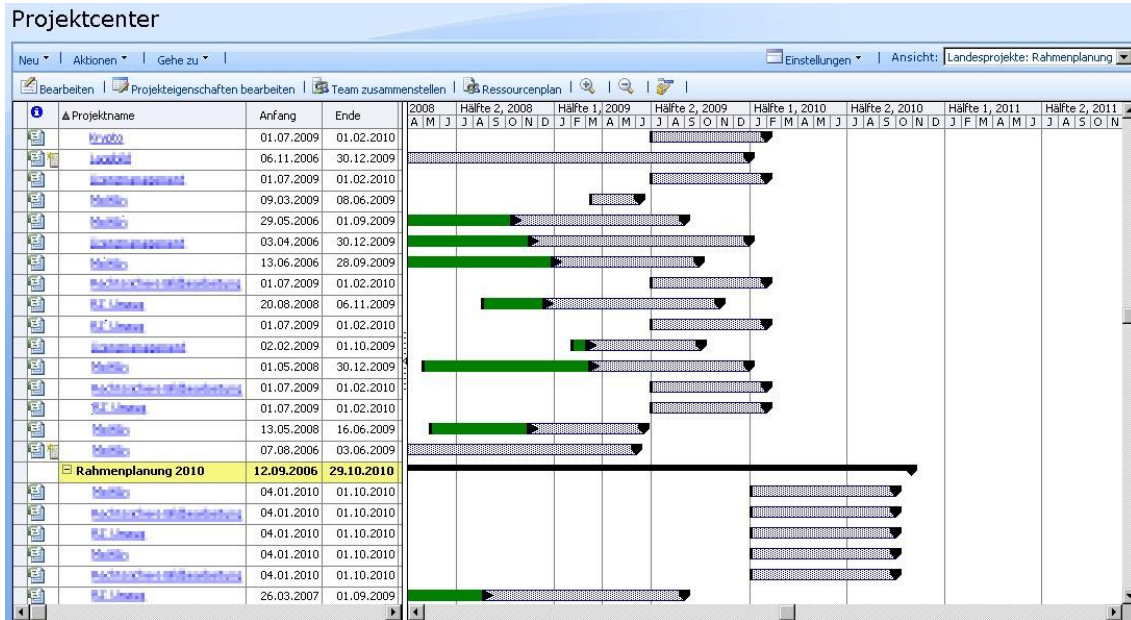
9.2. Multiprojektmanagement mit dem Projectserver



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Projektübersicht



29 Multiprojektmanagement

Berlin, 27.05.2009



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Multiprojektstatusbericht

Projekte > Projekte

Neu | Aktionen | Einstellungen | Ansicht: LZPD_Monatsberichte

Projektname	Startdatum	Enddatum	Projektphase	PAUF Status	PAUF	Auftragnehmer	Projektleiter	Ampel/M	Monatsbericht	Projektart
Ampel/M: (6)										
...	30.01.2009	31.07.2009	Umsetzung	Genehmigt	090129	LZPD	...	●	Februar 2009	Landesprojekt
...	05.05.2008	31.07.2009	Umsetzung	Genehmigt	080415	LZPD	...	●	Februar 2009	Strategie
...	01.08.2006	30.04.2009	Umsetzung	Genehmigt	65925	LZPD	...	●	Februar 2009	Strategie
...	01.05.2008	01.01.2009	Umsetzung	Genehmigt	80417	LZPD	...	●	Februar 2009	Landesprojekt
...	13.05.2008	31.05.2009	Umsetzung	Genehmigt	80319	LZPD	...	●	Februar 2009	Landesprojekt
...	12.03.2007	31.05.2009	Umsetzung	Genehmigt	70306	LZPD	...	●	Februar 2009	Strategie
Ampel/M: (6)										
...	02.01.2008	31.12.2009	Umsetzung	Genehmigt	080414	LZPD	...	☹	Februar 2009	Strategie
...	20.10.2008	29.05.2009	Umsetzung	Genehmigt	81017	LZPD	...	☹	Februar 2009	Strategie
...	26.10.2007	30.06.2009	Umsetzung	Genehmigt	71025	LZPD	...	☹	Februar 2009	Strategie
...	05.02.2008	01.03.2010	Umsetzung	Genehmigt	80205	LZPD	...	☹	Februar 2009	Strategie
...	28.01.2004	31.03.2009	Umsetzung	Genehmigt	40128	LZPD	...	☹	Februar 2009	Landesprojekt
...	01.04.2007	30.09.2009	Umsetzung	Genehmigt	70305	LZPD	...	☹	Februar 2009	Strategie
Ampel/M: (5)										
...	14.03.2007	31.12.2009	Umsetzung	Genehmigt	70313	LZPD	...	●	Februar 2009	Landesprojekt
...	12.09.2006	30.09.2010	Umsetzung	Genehmigt	60912	LZPD	...	●	Februar 2009	Strategie
...	15.09.2007		Umsetzung	Genehmigt	70814	LZPD	...	●	Februar 2009	Strategie
...	12.06.2006		Umsetzung	Genehmigt	60612	LZPD	...	●	Februar 2009	Strategie
...	23.08.2006	31.03.2009	Umsetzung	Genehmigt	60823	LZPD	...	●	Februar 2009	Strategie
Ampel/M: (9)										
...			Vorbereitung	Vorgelegt	080710	LZPD	...			Landesprojekt
...	20.01.2006	30.06.2011	Umsetzung	Genehmigt	61013	LZPD	...			Landesprojekt
...	02.10.2007	31.05.2008	Umsetzung	Genehmigt		LZPD	...			Behördenprojekt
...	12.08.2004	01.04.2008	Umsetzung	Genehmigt	40812	LZPD	...			Strategie
...	01.07.2007	01.07.2008	Umsetzung	Genehmigt	70410	LZPD	...			Landesprojekt

30 Multiprojektmanagement

Berlin, 27.05.2009



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



... einige Zahlen

- > zur Zeit 73 Projekte auf dem Projectserver
- > davon 32 laufende Landesprojekte und 5 Behördenprojekte

... und wie geht es weiter ?

- > Einführung des Ressourcenmanagement
- > Verbesserung des Projektportfoliomanagement



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



AGENDA

1. Das Landesamt für Zentrale Polizeiliche Dienste
2. Ausgangssituation und Zielsetzung
3. Präsentation des Vorgehens bei der Polizei NRW
4. Fazit



Fazit Teil -1-

- > Personaleinsparung durch Bündelung der „Geschäftsführertätigkeit“
- > große Handlungssicherheit der beteiligten Mitarbeiter
- > kein zusätzlicher Schulungsaufwand erforderlich
- > Vereinheitlichung der Projektdurchführung

- > Nutzung von MS-Project funktioniert nur mit entsprechendem Schulungsaufwand
- > laufende Betreuung notwendig
- > Einführung nur bei „neuen“ Projekten
- > keine Überfrachtung der Funktionalitäten



Fazit Teil -2-

-> keine Einführung einer Softwareunterstützung ohne Definition von Standards

aber

-> auch keine Einführung von Standards ohne Softwareunterstützung

Wichtig: Externe Unterstützung



Landesamt für Zentrale Polizeiliche Dienste Nordrhein-Westfalen



Vielen Dank für die Aufmerksamkeit

Klaus Schenk
Landesamt für zentrale polizeiliche Dienst
Tel.: 0203 41751223
Email: Klaus.Schenk@polizei.nrw.de

9.3. Gezähmte Papiertiger: Mit Tailoring eine Methode für das Projekt nützlich machen

Thorsten Gau
Methoden und Werkzeuge
IBM Deutschland
Beim Strohause 17
20097 Hamburg
tgau@de.ibm.com

Abstract

Der Vortrag zeigt, wie generische Methoden (z.B. SCRUM oder RUP) in kurzer Zeit mit Hilfe von Werkzeugen wie dem Eclipse Process Framework Composer (EPFC) oder dem Rational Method Composer (RMC) so angepasst werden können, dass Projektteams sie gerne nutzen. Praxisbeispiele zeigen, wie eine zentrale Projektablage auf Basis der Methode zum Projekterfolg beiträgt und dass konkrete Hilfestellungen die Akzeptanz deutlich erhöhen.

Motivation

An agilen und reichhaltigen Vorgehensmodellen herrscht kein Mangel. Einige Methoden kommen als knappe Ansammlung von Prinzipien daher, andere in Form von langen und ausführlichen Texten. Projektteams stehen häufig vor dem Problem, dass es an konkreten Handlungsanweisungen fehlt. Viele im Prinzip gute Methoden enden als zahnlose Papiertier im Regal oder verstauben virtuell im Intranet, da es häufig nicht gelingt, Projektteams von deren Nützlichkeit zu überzeugen.

Ansatz

Generische Vorgehensmodelle und Methoden müssen an die konkrete Projektsituation angepasst werden. Hierzu ist es notwendig alle nicht essentiellen Inhalte wie Ergebnistypen, Aufgaben und Rollendefinitionen zu entfernen und gegebenenfalls Elemente aus anderen Methoden hinzuzufügen. In mehreren Projekten haben wir die Erfahrung gemacht, dass konkrete Hilfestellungen wie Dokumentvorlagen, Beispiele und Checklisten den größten Nutzen für das Projektteam bringen.

Seit mehreren Jahren führt man zu Beginn eines Projektes einen Methoden-Adoptions-Workshop (MAW) durch. Ein Projekt-Kernteam passt die gewählte(n) Methode(n) an die erwartete Projektsituation an. Während dieses Tailorings adoptiert das Projekt die Methode, d.h. sie wird als „eigenes Baby“ betrachtet und nicht als von außen vorgegeben.

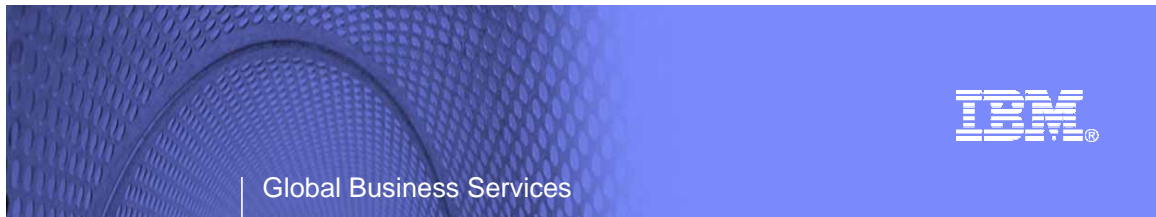
Seit der Veröffentlichung der Software & Systems Process Engineering Meta-Model Specification (SPEM) 2.0 durch die Object Management Group (OMG) steht ein flexibles Meta-Modell zur Beschreibung von Methoden zur Verfügung. Inzwischen stehen sowohl agile als auch reichhaltige Vorgehensmodelle als SPEM-konforme Beschreibung zur Verfügung. Mit Hilfe des freien Eclipse Process Framework Composers (EPFC) oder dem

kommerziellen Rational Method Composer (RMC) können Methoden-Tailorings schnell und flexibel durchgeführt werden.

Der letzte und wichtigste Schritt zum Erfolg ist die Verankerung der Methode im Projekt. Die adoptierte Methode muss Teil des Projektalltags werden, d.h. sie muss jedem Projektbeteiligten mit minimalem Aufwand zur Verfügung stehen und von den verwendeten Werkzeugen angemessen unterstützt werden. Auch dies ist möglich und der Vortrag zeigt, wie wir dies im Projektalltag umsetzen.

Bewertung

Erprobte Werkzeuge unterstützen das Tailoring einer Methode und die Verankerung im Projekt. Je besser Methode und Projekt zusammenpassen und je konkreter die Hilfestellung für die Projektbeteiligten desto höher ist die Bereitschaft sich mit der Methode zu beschäftigen. Je mehr Vorgehensmodelle auf Basis von SPEM beschrieben werden, desto leichter lassen sich deren Best Practices ins hauseigene Vorgehensmodell integrieren.



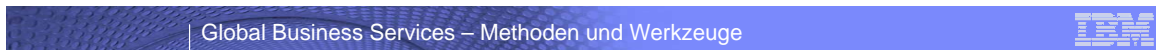
Gezähmte Papiertiger Mit Tailoring eine Methode für das Projekt nützlich machen

Thorsten Gau
Executive IT Architect
IBM Deutschland



SEE 2009 | Berlin | 27.05.09

© 2006 IBM Corporation



Thorsten Gau




- **Tätig bei IBM seit 1994**
 - Begonnen als Spezialist im Document Imaging Competence Center in Hamburg.
Unterstützt diverse namhafte Unternehmen bei der Einführung von Enterprise **Content Management** Lösungen.
 - Executive IT **Architekt**
Viele erfolgreiche **globale** Systemintegrations- und Anwendungsentwicklungsprojekte.
 - Zu seinem Repertoire gehört der Einsatz von Objekt-, Webservice- und Web 2.0 Technologien; häufig im Rahmen Service Orientierter Architekturen (SOA).
- Open Group 
- Bei IBM Global Business Services europaweit für die Ausbildung zum Thema **Methoden**, Tools & Architektur verantwortlich



https://www.xing.com/profile/Thorsten_Gau
<https://apps.lotuslive.com/contacts/profiles/view/981>



Global Business Services – Methoden und Werkzeuge 

Agenda

► Gründe für Tailoring


Methodik = Inhalt + Prozess

(Frei) verfügbare Werkzeuge

Verankerung als Erfolgsgrundlage

Wer von Ihnen arbeitet vollständig nach dem durch das Vorgehensmodell vorgegebenen Arbeitsstrukturbaum (WBS)?

3 | Gezähmte Papiertieger | Thorsten Gau | SEE 2009 | © 2009 IBM Deutschland

Global Business Services – Methoden und Werkzeuge 

Generische Vorgehensmodelle und Methoden werden vom Projektteam oft als *zu groß*, *zu klein* oder *unpassend* abgelehnt

- V-Modell XT
- Rational Unified Process (RUP)
- Open Unified Process (OpenUP)
- Scrum
- ...

4 | Gezähmte Papiertieger | Thorsten Gau | SEE 2009 | © 2009 IBM Deutschland

Generische Vorgehensmodelle und Methoden müssen an die konkrete Projektsituation angepasst werden: *Tailoring*

- V-Modell XT
- Rational Unified Process (RUP)
- Open Unified Process (OpenUP)
- Scrum
- ...



Anzug
"von der Stange"



Der Schneider
passt Anzug an



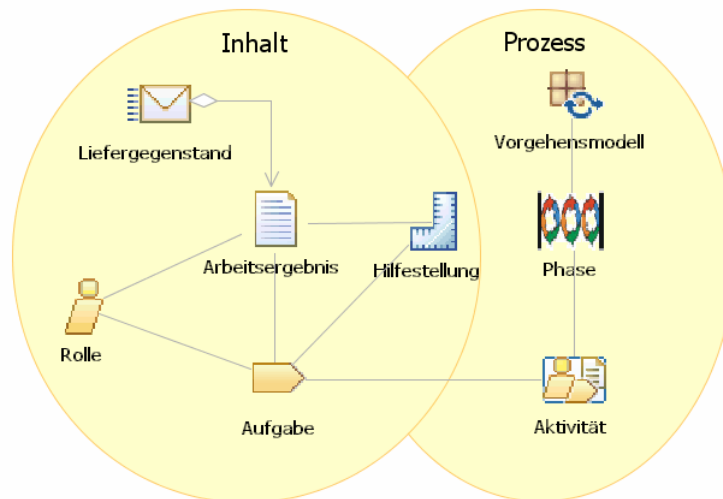
Der Anzug sitzt
perfekt

Agenda

Gründe für Tailoring















- ▶ Methodik = Inhalt + Prozess
- (Frei) verfügbare Werkzeuge
- Verankerung als Erfolgsgrundlage

Eine Methodik besteht aus Inhalts-Elementen und einem Prozess, der die Reihenfolge der Aufgaben festlegt



Object Management Group (OMG)
 Software & Systems Process Engineering Meta-Model Specification (SPEM) 2.0

Durch projektspezifische Hilfestellungen wird die Methode für die Teammitglieder wertvoll – weil konkret.

-  **Checklist:** provides a list of items that need to be completed or verified, often used in reviews or for validation
-  **Concepts:** outlines key ideas or underlying principles that serve as foundation for more explicit guidance
-  **Estimation Considerations:** provides guidance on the amount of effort to produce a work product or perform a task including any influencing factors
-  **Examples:** used to include typical samples of the items to be produced, may often only be a partial sample that is intended as further guidance rather than something to be reused
-  **Guideline:** general type of guidance used to provide additional detail on to perform some set of actions or additional rules or recommendations related to work products and their properties
-  **Practice:** represents a proven way or strategy of doing something, they may also represent standards
-  **Report:** is a predefined template that represents the output of an automated tool that may represent a combination of information from other work products
-  **Reusable Asset:** guidance linking method content to intellectual capital that can be utilized to perform some task or leveraged as a starting point for the creation of a solution
-  **Roadmap:** is process specific guidance that represents a linear walkthrough of those items from a particular perspective
-  **Supporting Material:** is a catchall used to represent any other type of guidance that does not have a specific type
-  **Template:** a form or empty instance of a work product that can be used as starting point for the creation of a new one
-  **Term Definition:** used to define items that make up the glossary
-  **Tool Mentor:** specialize form of guidance on how to apply a specific tool to accomplish some piece of work or develop a particular work product
-  **Whitepaper:** specialized version of concept that has been externally reviewed or published

9.3. Gezähmte Papiertiger: Mit Tailoring eine Methode für das Projekt nützlich machen

Global Business Services – Methoden und Werkzeuge

Agenda

Gründe für Tailoring

- Methodik = Inhalt + Prozess
- (Frei) verfügbare Werkzeuge
- Verankerung als Erfolgsgrundlage

9 | Gezähmte Papiertieger | Thorsten Gau | SEE 2009 | © 2009 IBM Deutschland

Ziel des Eclipse Process Framework Projektes ist ein agiler und anpassbarer SE-Prozess. Der *Composer* ist das passende Werkzeug.

The screenshot shows the Eclipse Process Framework Composer interface. The main window displays a project structure for 'OpenUP Lifecycle' with a table of tasks and their dependencies.

Presentation Name	In...	Predecessors	Model Info	Type	Plann...	Repe...
OpenUP Lifecycle	0			Delivery ...	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Inception Phase	1			Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Inception Iteration [1..n]	2			Iteration	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Initiate Project	3		extends 'inception_p...	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Plan and Manage Iteration	6		extends 'initiate_projec...	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Identify and Refine Requirements	10	3	extends 'identify_and_r...	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Identify and Outline Requirements	11			Task Des...	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Analyst			Primary Performer	Role Des...	<input type="checkbox"/>	<input type="checkbox"/>
Architect			Additional Performer	Role Des...	<input type="checkbox"/>	<input type="checkbox"/>
Developer			Additional Performer	Role Des...	<input type="checkbox"/>	<input type="checkbox"/>
Stakeholder			Additional Performer	Role Des...	<input type="checkbox"/>	<input type="checkbox"/>
Tester			Additional Performer	Role Des...	<input type="checkbox"/>	<input type="checkbox"/>
[Technical Specification]			Optional Input	Artifact De...	<input type="checkbox"/>	<input type="checkbox"/>
Glossary			Output	Artifact De...	<input type="checkbox"/>	<input type="checkbox"/>
System-Wide Requirements			Output	Artifact De...	<input type="checkbox"/>	<input type="checkbox"/>
Use Case			Output	Artifact De...	<input type="checkbox"/>	<input type="checkbox"/>
Use-Case Model			Output	Artifact De...	<input type="checkbox"/>	<input type="checkbox"/>
Detail Use-Case Scenarios	12			Task Des...	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Detail System-Wide Requirements	13			Task Des...	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Create Test Cases	14			Task Des...	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Agree on Technical Approach	15	3	extends 'agree_techni...	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Lifecycle Objectives Milestone	17	2		Milestone	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Elaboration Phase	18	17,1		Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Elaboration Iteration [1..n]	19		extends 'elaboration...	Iteration	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Lifecycle Architecture Milestone	48	19		Milestone	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Construction Phase	49	48,18		Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Construction Iteration [1..n]	50		extends 'construction...	Iteration	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Der EPF-Composer unterstützt verschiedene Wege des Tailorings.

- Projekt-spezifische Hilfestellungen hinzufügen:
z.B. Vorlagen, Beispiele, (Qualitäts-)Checklisten
- Nicht relevante Elemente ausblenden:
z.B. Ergebnistypen, Rollen, Aufgaben sowie Aktivitäten
- Ergebnistypen zusammenfassen:
Liefergegenstände enthalten Artefakte “als Kapitel”
- Erstellung eines Arbeitsstrukturbaumes (WBS) “von Null auf” per drag’n’drop aus der Methodenbibliothek

➔ Das Ergebnis kann als interaktive Website oder MS Project Plan ausgegeben werden

Der Rational Method Composer (RMC) bietet erweiterte Tailoring-Möglichkeiten: Automatisches Ausblenden von WBS-Elementen

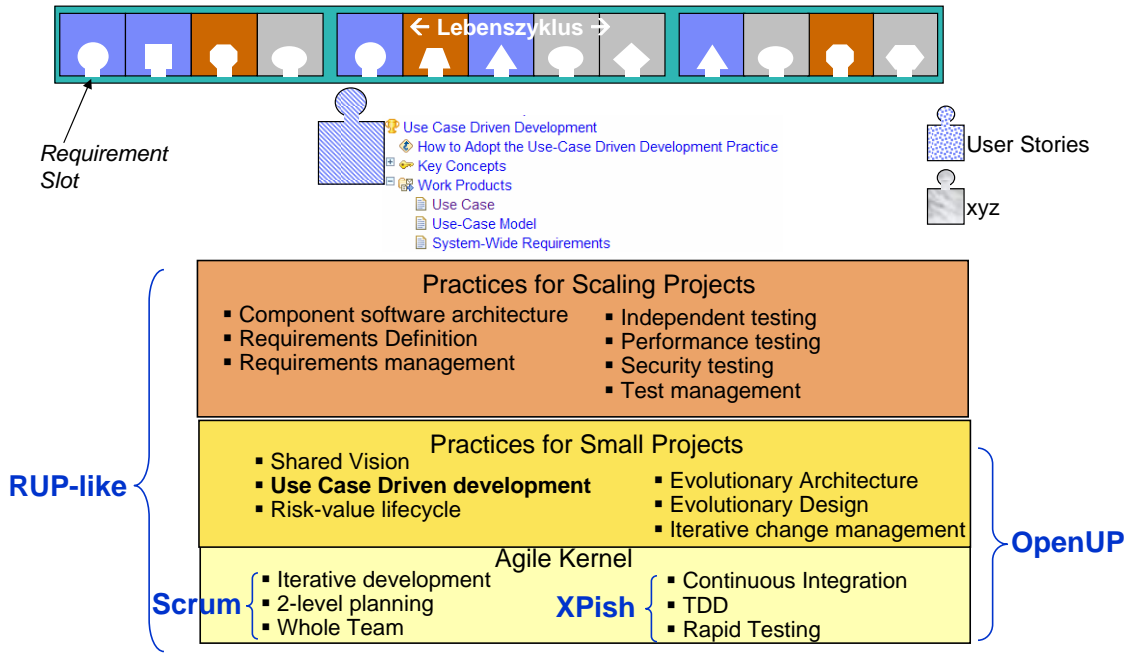
The screenshot shows the IBM Rational Method Composer interface. On the left is a tree view of 'Verfügbare Elemente' (Available Elements) under 'MyTailoring 1'. The main area displays a table of project elements:

Präsentationsname	In...	V...	Modellinfo	Typ	Gep...	Wied...	Mehr...
myRUP	0			Bereitstel	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Konzeption	1			Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Konzeptionsiteration [n]	2			Aktivität	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Neues Projekt konzipieren	3			Aktivität	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Projektumgebungen vorbereiten	8			Aktivität	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Umgebung für eine Iteration vorbereiten	13	3,8		Aktivität	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Projektpläne definieren	15	13		Aktivität	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Projekt überwachen und steuern	23	13		Aktivität	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Systemumfang verwalten	28			Aktivität	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vision entwickeln	29			Aufgabe...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Systemanalytiker			Primärer Ausführender	Rollende	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Iterationsplan			Verbindliche Eingabe	Artefaktid.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kosten-Nutzen-Analyse			Verbindliche Eingabe	Artefaktid.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Stakeholder-Anfragen			Verbindliche Eingabe	Artefaktid.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Geschäftsanalysemodell			Optionale Eingabe	Artefaktid.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Geschäftsanwendungsfallmodell			Optionale Eingabe	Artefaktid.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Geschäftsdesignmodell			Optionale Eingabe	Artefaktid.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Geschäftsregel			Optionale Eingabe	Artefaktid.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vision			Optionale Eingabe	Artefaktid.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Anforderungsattribute			Ausgabe	Artefaktid.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vision			Ausgabe	Artefaktid.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prioritäten für Anwendungsfälle vergeben	30			Aufgabe...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
System definieren	31			Aktivität	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Below the table, the 'Arbeitsergebnisdeskriptor: Vision' is shown, with a description: 'Dieses Artefakt definiert die Stakeholder-Sicht des zu entwickelnden Produkts anhand der wichtigsten Anforderungen und Features des Stakeholders. Es enthält eine Übersicht über die vorgesehenen...'

9.3. Gezähmte Papiertiger: Mit Tailoring eine Methode für das Projekt nützlich machen

Neben dem “Tailoring durch Ausblenden” können mit dem *Process Builder* Practices zu Prozessen *zusammengesteckt* werden



Agenda

Gründe für Tailoring

Methodik = Inhalt + Prozess

(Frei) verfügbare Werkzeuge

▶ Verankerung als Erfolgsgrundlage

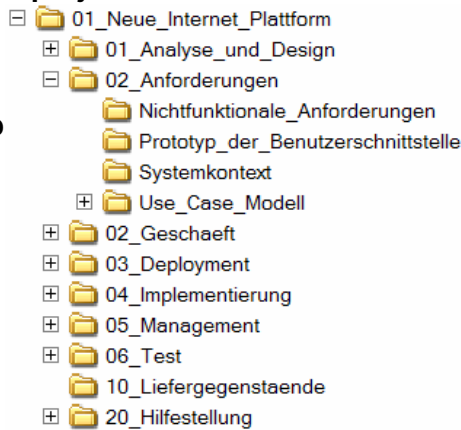
Die einmal adoptierte minimale Methodik muss bei der täglichen Arbeit stets erkennbar sein: *Projektablage*

- Jedes Arbeitsergebnis ist einer Domäne* zugeordnet.
- Domäne und Ergebnistyp ergeben eine natürliche Ablagehierarchie

1. Ebene: **Name des Teilprojektes**

2. Ebene: **Domäne**

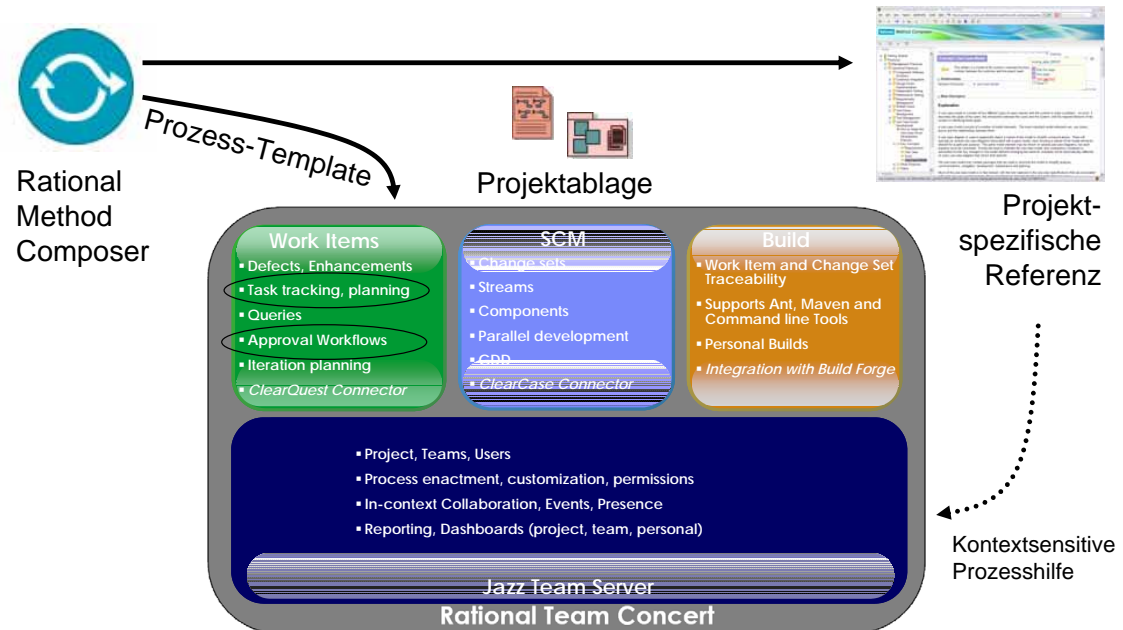
3. Ebene: **Ergebnistyp**



unter Versionskontrolle (SCM)

*manchmal auch Disziplin genannt

Die einmal adoptierte minimale Methodik muss bei der täglichen Arbeit stets erkennbar sein: *Rational Team Concert*





Zusammenfassung

- Generische Vorgehensmodelle und Methoden müssen an die konkrete Projektsituation angepasst werden: *Tailoring*
- Eine Methodik besteht aus Inhalts-Elementen und einem Prozess. Die OMG hat mit SPEM 2.0 einen internationalen Standard zur Beschreibung von Methodiken etabliert.
- Vorlagen, Beispiele, Checklisten und weitere Hilfestellungen machen eine Methodik konkret und nützlich für das Team.
- Mit dem Eclipse Process Framework Composer und dem Rational Method Composer stehen leistungsfähige Tailoring-Werkzeuge zur Verfügung.
- Die einmal adoptierte minimale Methodik muss bei der täglichen Arbeit stets erkennbar und im direkten Zugriff sein.



Weiterführende Quellen

- Eclipse Process Framework (EPF)
<http://www.eclipse.org/epf>
 - Composer
 - OpenUP / Agile Practices
 - Scrum
- Rational Method Composer (RMC)
http://www-142.ibm.com/software/dre/ecatalog/detail.wss?locale=de_DE&synkey=L271254L87028R67
- Jazz + Rational Team Concert (RTC)
<http://jazz.net>
http://www-142.ibm.com/software/dre/ecatalog/detail.wss?locale=de_DE&synkey=N556894X56284V28
- Software & Systems Process Engineering Metamodel specification (SPEM) Version 2.0
<http://www.omg.org/spec/SPEM/2.0/>

10. Session 10: Fundamente für Vorgehensmodelle

Inhalt

10.1. SPEM und ISO/IEC 24744 im Vergleich	394
10.2. Werkzeuggestützte Konsistenzsicherung von SE-Prozessen	409
10.3. Erstellung und Bereitstellung von Prozessbeschreibungen	423

10.1. Metamodell-basierte Modellierung eines Software-Entwicklungsprozesses in der Automobilindustrie: SPEM und ISO/IEC 24744 im Vergleich

Jens Palluch¹, Cristina Romcea²

¹Method Park Software AG
Wetterkreuz 19a, 91058 Erlangen
Jens.Palluch@methodpark.de

²Continental AG
Sieboldstraße 19, 90411 Nürnberg
Cristina.Romcea@continental-corporation.com

Abstract

Zur Modellierung von komplexen Software-Entwicklungsprozessen, wie sie z.B. in der Automobilbranche existieren, entstanden in den letzten Jahren zwei Metamodelle: das „Software & Systems Engineering Meta-Model“ (SPEM) und das „Software Engineering Meta-Model for Development Methodologies“ [4]. Im Rahmen eines industriellen Forschungsprojektes wurden beide Metamodelle hinsichtlich ihrer Eignung untersucht, indem ein Ausschnitt aus dem existierenden Software-Entwicklungsprozess von Continental Automotive Systems mit beiden modelliert wurde.

Extended Abstract

Bei der Continental AG erfolgt die Software-Entwicklung im Bereich Automotive Systems nach komplexen Entwicklungsprozessen, da die Entwicklung von Software stark verzahnt ist mit der Entwicklung der Elektronik, der Hardware, usw. Hinzu kommt, dass die Entwicklung an verschiedenen Standorten stattfindet, was umfangreiche Abstimmungsmaßnahmen erfordert. Außerdem müssen bei der Software-Entwicklung verschiedene Normen und Standards wie Automotive SPICE, ISO/TS 16949 und ISO CD 26262 berücksichtigt werden. Zur Minimierung der Projektrisiken und zur Verbesserung der Qualität wurde daher ein standortübergreifender Software-Entwicklungsprozess definiert. Zur Unterstützung der Definition von Prozessen gibt es eine Vielfalt an Prozess-Modellierungssprachen, z.B. ereignisgesteuerte Prozessketten (EPK), Business Process Execution Language (BPEL), Business Process Modeling Notation (BPMN), XML Process Definition Language (XPDL). Die meisten sind jedoch für die Modellierung von Abläufen, wie sie typischerweise in Geschäftsprozessen vorkommen, gedacht und eignen sich daher nicht besonders gut für die Modellierung von Entwicklungsprozessen. Vor diesem Hintergrund wurden zwei Metamodelle entwickelt, um den speziellen Bedürfnissen von Entwicklungsprozessen gerecht zu werden: das „Software & Systems Engineering Meta-Model“ (SPEM) von der Object Management Group (OMG) und das „Software Engineering Meta-Model for Development Methodologies“ [4] der ISO.

SPEM wurde im Jahr 2002 als Version 1.0 freigegeben. Nach einigen kleinen Änderungen daran wurde im Jahre 2005 Version 1.1 veröffentlicht. Die 1.x-Versionen fanden aber keine große Verbreitung in der Industrie. Hauptkritikpunkte waren z.B. die mangelnde Unterstützung zur Prozessausführung (Process Enactment), die in den Spezifikationen

sogar explizit ausgeschlossen war sowie die Mehrdeutigkeit bestimmter Konzepte [1]. Um diese Mängel zu beheben startete in 2004 die Entwicklung von SPEM 2.0. Sie wurde nach ungefähr dreieinhalb Jahren im April 2008 mit der Freigabe der endgültigen Version abgeschlossen.

In verschiedenen Veröffentlichungen wurde Kritik an SPEM geäußert [2, 3]. Die Kritiker von SPEM begannen ab 2003 mit der Definition eines eigenen Metamodells, SEMDM, das als Internationaler Standard ISO/IEC 24744 im Februar 2007 veröffentlicht wurde [4].

Im Rahmen eines industriellen Forschungsprojektes wurden beide Metamodelle näher untersucht, indem ein Teil des existierenden Software-Entwicklungsprozesses von Continental Automotive Systems mit SPEM und SEMDM modelliert wurde. Im Falle von SPEM wurde auf das in der SPEM-Spezifikation definierte UML-Profil zurückgegriffen und die Modellierung mit einem UML-Werkzeug durchgeführt. Zur Modellierung mit SEMDM wurde ein eigens entwickelter Prototyp verwendet.

Beide Metamodelle eignen sich zwar grundsätzlich zur Modellierung von Software-Entwicklungsprozessen, haben aber einige Nachteile. Das Metamodell von SPEM 2.0 und das UML-Profil sind in der Spezifikation von SPEM an einigen Stellen unterschiedlich definiert, obwohl das nicht so sein sollte. In der Version 2.0 von SPEM gibt es noch einige weitere Fehler und Widersprüche. Außerdem gibt es noch immer keine wirkliche Unterstützung für die Prozessausführung, obwohl das eine Zielsetzung für diese Version war. SEMDM erlaubt zwar Erweiterungen des Metamodells, diese sind aber nur eingeschränkt möglich, weil existierende Elemente weder verändert noch ersetzt werden dürfen

In diesem Vortrag sollen beide Metamodelle vorgestellt, miteinander verglichen und die Ergebnisse der Evaluierung präsentiert werden.

Literaturverzeichnis

1. OMG: Software & Systems Process Engineering Metamodel Specification v2.0, April 2008
2. Henderson-Sellers, Brian und Cesar Gonzalez-Perez: A Comparison of Four Process Metamodels and the Creation of a New Generic Standard. *Information and Software Technology*, 47(1). 2005
3. Henderson-Sellers, Brian und Cesar Gonzalez-Perez: The Rationale of Powertype-Based Metamodelling. In: *Australian Computer Science Communications* vol. 7, number 6. Australian Computer Society. 2005
4. ISO/IEC. Software Engineering - Metamodel for Development Methodologies. ISO/IEC 24744: 2007



Willkommen



Metamodell-basierte Modellierung
eines Software-Entwicklungsprozesses
in der Automobilindustrie:

SPEM und ISO/IEC 24744 im Vergleich

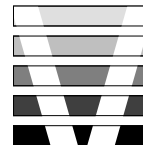
Jens Palluch, Method Park Software AG
Cristina Romcea, Continental AG

Entwicklungsprozess bei Continental



Basis

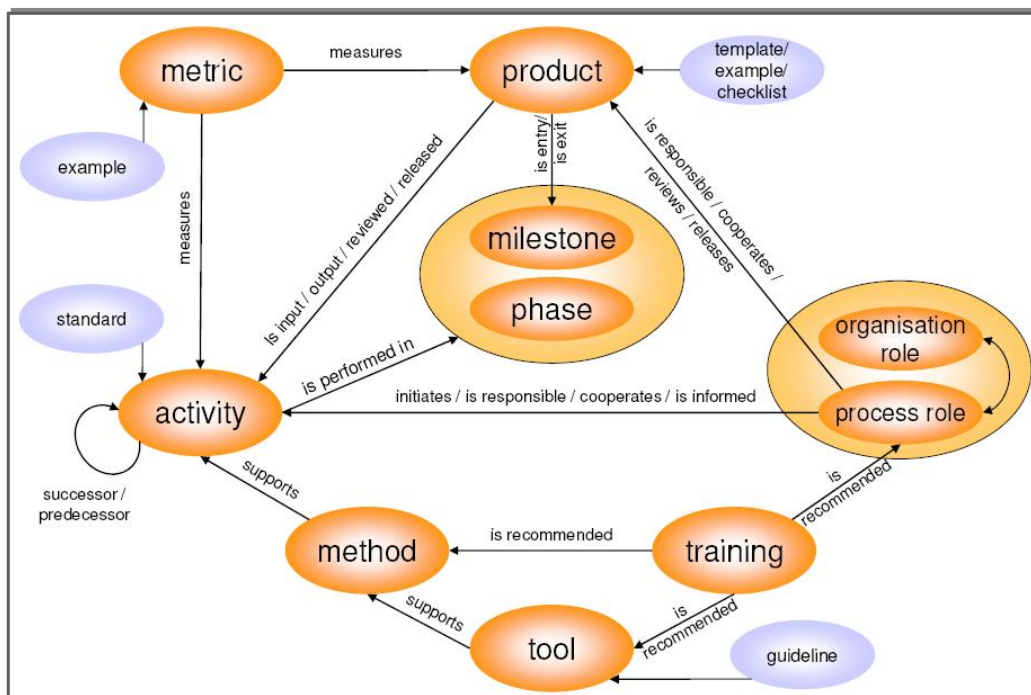
- V-Modell 97
- Automotive SPICE® Base Practices und Arbeitsprodukte
- Anforderungen an die funktionale Sicherheit (ISO CD 26262)
- spezifische Prozessanweisungen und Richtlinien der Geschäftsbereiche



Ergebnis

- Continental Metamodell
- P&C Entwicklungsprozess

Metamodell bei Continental



stages

Prozessmodellierungstool ist „Stages“ von Method Park SW AG



Activity: Identify Software Requirements Released

Activity description

The objective of the activity is to define the software requirements in such a way that they form a base for the design of the software subsystems, their acceptance and their alteration. Also in this activity the prerequisites are laid down that ensure the software requirements can be traced back over the total lifecycle of the software subsystems. Needs from other subsystems (hardware, mechanics) are taken into account by means of [System Requirements](#) or [System Architecture](#).

Software requirements are continuously refined and improved in an iterative process until a suitable quality and level of detail for the software design has been reached. [Predefined states](#) are assigned to requirements during their lifecycle.

[\[Change\]](#)

Predecessor	Successor
[Change] [New]	<ul style="list-style-type: none"> ■ Analyse Software Requirements [Change] [New]

Products		
<table style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> Input: <ul style="list-style-type: none"> ■ System Requirements ■ Traceability Information ■ System Architecture <p style="text-align: right;">[Change] [New]</p> </td> <td style="width: 50%; vertical-align: top;"> Output: <ul style="list-style-type: none"> ■ Software Requirements ■ Traceability Information ■ Safety Requirements Specification Subsystem (SRS-Subsystem) <p style="text-align: right;">[Change] [New]</p> </td> </tr> </table>	Input: <ul style="list-style-type: none"> ■ System Requirements ■ Traceability Information ■ System Architecture <p style="text-align: right;">[Change] [New]</p>	Output: <ul style="list-style-type: none"> ■ Software Requirements ■ Traceability Information ■ Safety Requirements Specification Subsystem (SRS-Subsystem) <p style="text-align: right;">[Change] [New]</p>
Input: <ul style="list-style-type: none"> ■ System Requirements ■ Traceability Information ■ System Architecture <p style="text-align: right;">[Change] [New]</p>	Output: <ul style="list-style-type: none"> ■ Software Requirements ■ Traceability Information ■ Safety Requirements Specification Subsystem (SRS-Subsystem) <p style="text-align: right;">[Change] [New]</p>	

Prozessmodellierung – Aktivitätensicht



Roles	
Initiator: Software Project Manager [Change] [New]	Responsible role: Software Requirements Engineer [Change] [New]
Cooperating roles: ▪ Customer [Change] [New]	Roles to be informed: [Change] [New]
Performance in phase	
▪ Software Requirements Analysis [Change] [New]	
Applied methods	Applied metrics
▪ Requirements Analysis and Specification [Change] [New]	[Change] [New]
Standard	
[Change] [New]	
Reference mappings to Automotive SPICE 2.2	
▪ ENG: Engineering Process Group > ENG 4: Software requirements analysis > ENG 4.BP1: Identifv software requirements [Outcome 1]	

Prozessmodellierung – Produktsicht



Product: Software Requirements Tailor away ● Released

Product instances (1)	Resources (5)								
<table border="1"> <thead> <tr> <th>File</th> <th>Version</th> <th>Date</th> <th>State</th> </tr> </thead> <tbody> <tr> <td> Software_Requirements_PROJECTNAME.pdf</td> <td></td> <td></td> <td>No versions</td> </tr> </tbody> </table>		File	Version	Date	State	Software_Requirements_PROJECTNAME.pdf			No versions
File	Version	Date	State						
Software_Requirements_PROJECTNAME.pdf			No versions						
Product description									
Roles									
Responsible: ▪ Software Requirements Engineer	Cooperating: ▪ Customer ▪ Core Team ▪ Project Manager ▪ Software Project Manager								
Review Responsible: ▪ Software Quality Engineer	Release Responsible: ▪ Software Quality Engineer								
Applied metrics									
▪ Requirement Status [Change] [New]									
Reference mappings to Automotive SPICE 2.2									
▪ 08-00 > 08-16: Release plan ▪ 17-00 > 17-11: Software requirements specification									
Reference mappings to ISO WD 26262									
▪ Concept phase (Part 3) > 3-4: Item definition > 3-4.4: Provisions (normative) > 3-4.4.1 ▪ Product development, Software level (Part 6) > 6-5: Specification of software safety requirements > 6-5.4: Requirements (normative) > 6-5.4.1: The software safety requirements shall be specified according to the overall management of safety requirements (WD 26262-9, chapter 5) using the methods and measures listed in table 5.1.									
Reference mappings to ISO CD 26262 BL11									
▪ Concept phase (Part 3) > 3-4: Item definition > 3-4.4: Provisions (normative) > 3-4.4.1 ▪ Product development, Software level (Part 6) > 6-5: Specification of software safety requirements > 6-5.4: Provisions (normative) > 6-5.4.1: General									

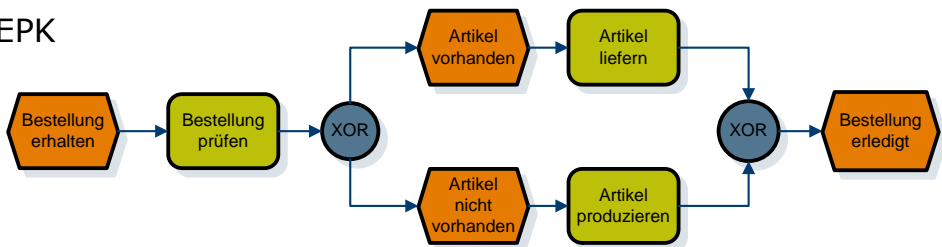
Software Requirements. As soon as
Change & Problem Resolution [Change]

Typischerweise workflow-orientiert

- Knoten
 - Aktivitäten
 - Verzweigung/Verbindung
 - Parallelisierung/Synchronisation
- Kanten
 - Kontrollfluss
 - Informationsfluss

Beispiele

- EPK/eEPK
- BPMN
- BPEL
- XPDL



Mangel an Prozessmodellierungssprachen für Entwicklungsprozesse

Definition von Modellierungssprachen mittels Metamodellen

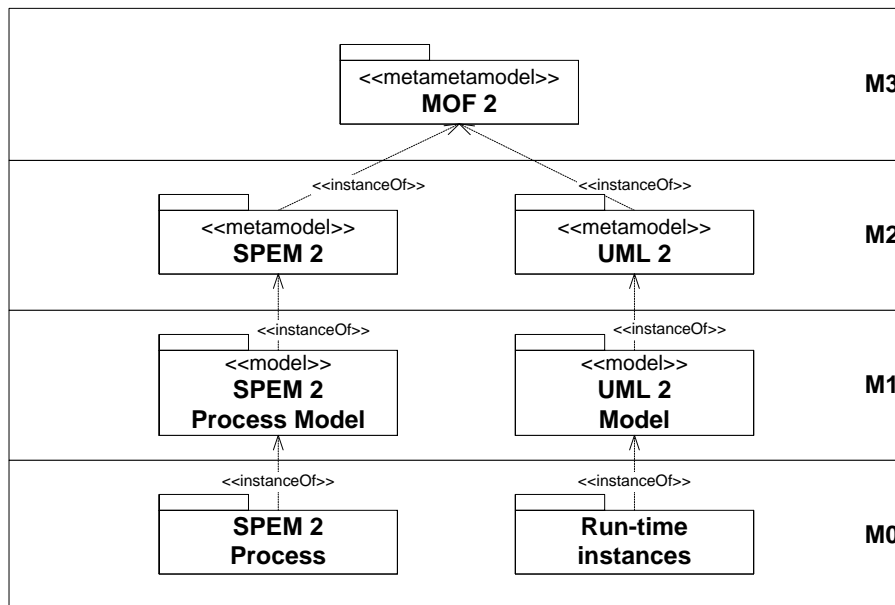
⇒ Definition von Metamodellen zur Modellierung von Software-Entwicklungsprozessen

Ergebnis

- Software Process Engineering Metamodel (SPEM)
- Software Engineering Metamodel for Development Methodologies (SEMDM) → ISO/IEC 24744



SPEM basiert auf der Vier-Ebenen-Architektur der OMG



Version 1.0 in 2002

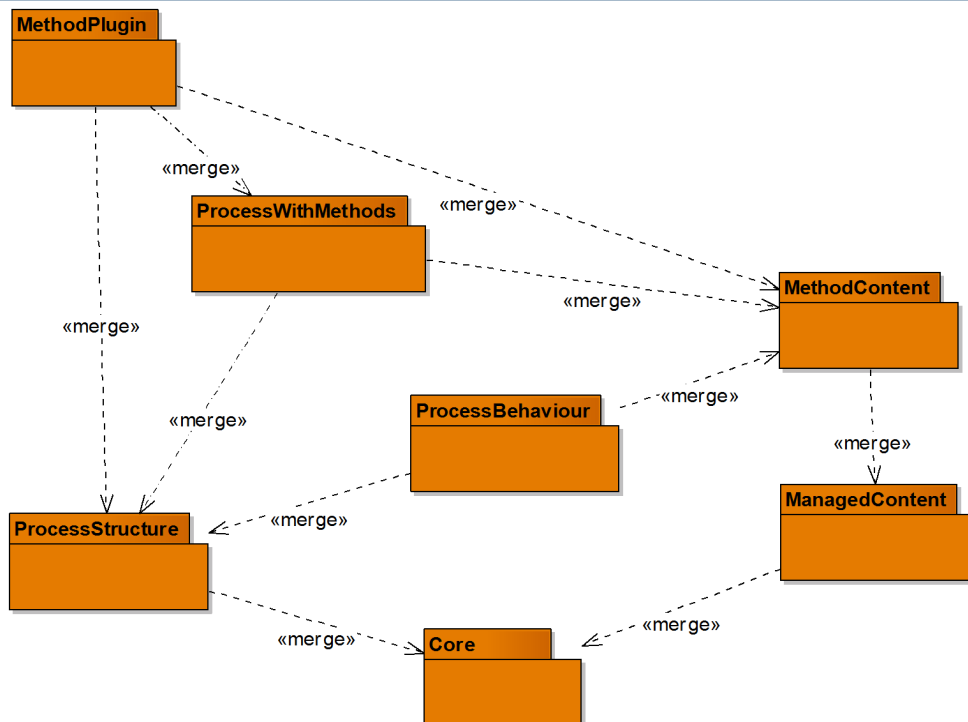
Version 1.1 in 2005

Hauptkritik an 1.x-Versionen

- mangelnde Unterstützung bei der Prozessausführung
- Mehrdeutigkeit bestimmter Konzepte

⇒ Version 2.0 in 2008

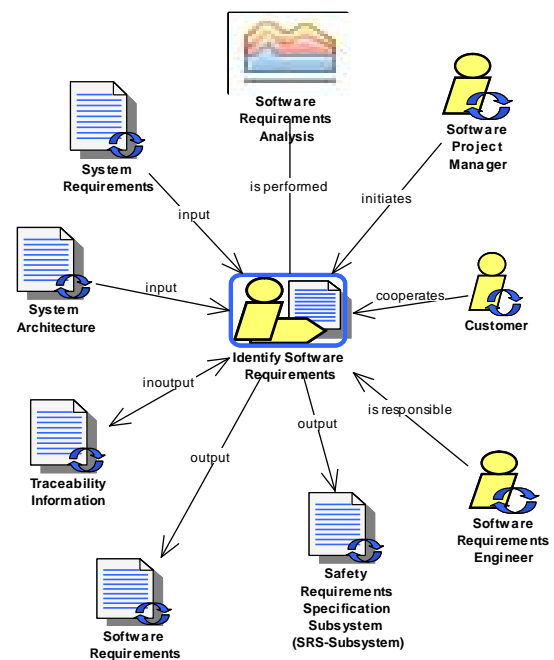
SPEM – Aktuelle Struktur



SPEM – Vorteile und Beispiel einer Aktivität

Vorteile von SPEM

- einfach zu verstehen für Personen, die mit UML2 vertraut sind (OMG Spezifikation)
- UML-Profil für SPEM zur Umsetzung in UML-Tool
- modularer Aufbau



Analytische Probleme

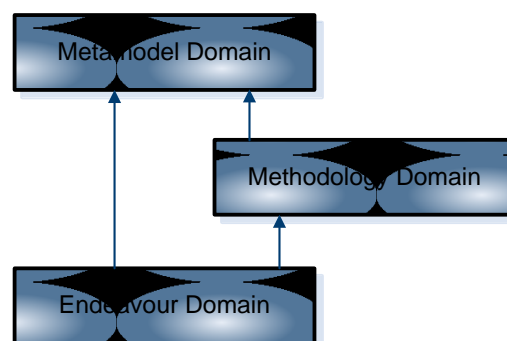
- fehlende Begriffsdefinitionen (z.B. Prozess)
- keine Regeln zur Erweiterung des Metamodells
- Fehler und Widersprüche
- für minimalistisches Metamodell zu umfangreich
 - über 70 Klassen
 - 236 Seiten

Modellierungsprobleme

- keine Regeln zur Modellierung mit SPEM
- Unterschiede zwischen Metamodell und UML-Profil
- Phase ist kein Element des Metamodells
- P&C Entwicklungsprozess von Continental ließ sich mit SPEM nur über Anpassungen und Erweiterungen von SPEM abbilden

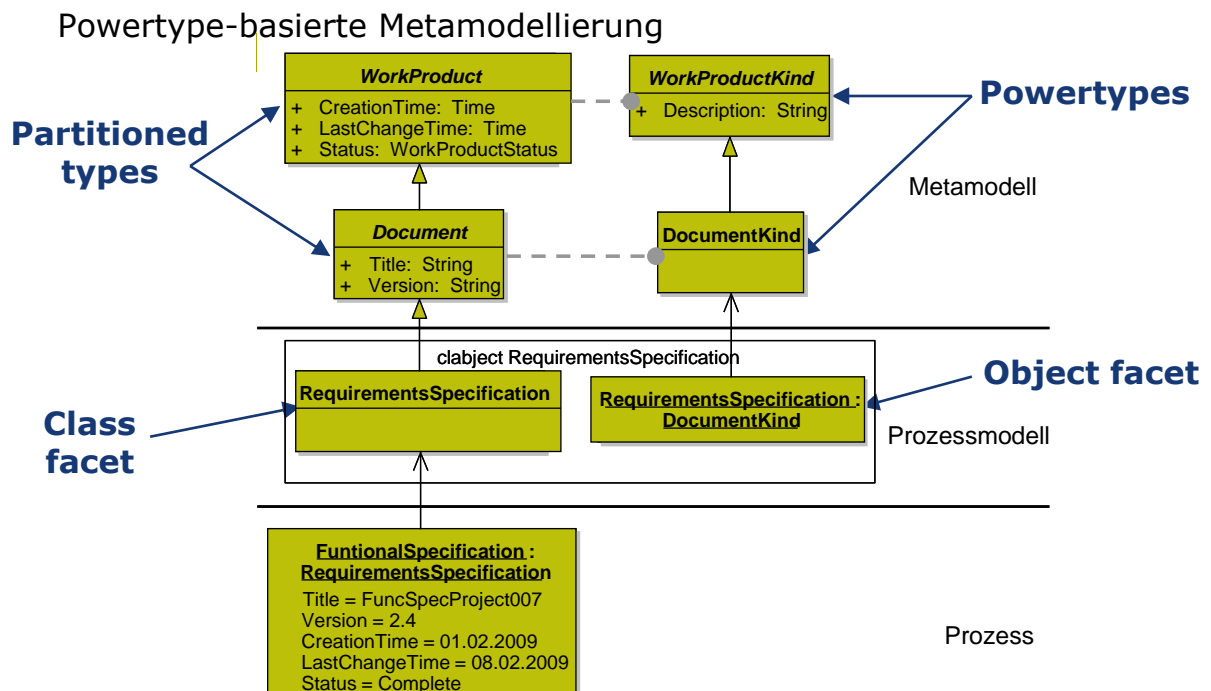
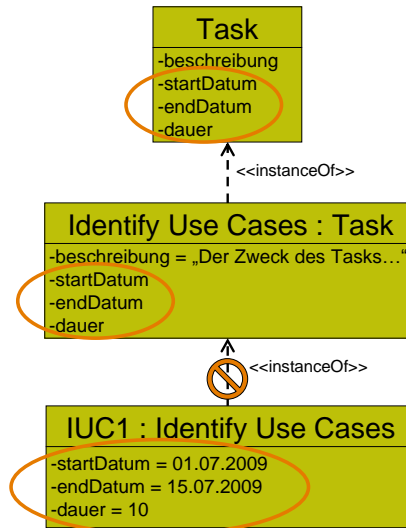
Unterschied zur Vier-Ebenen-Architektur der OMG

- Metamodell beinhaltet Klassen zur Definition der Prozesselemente zur Laufzeit (Endeavour)
- nicht nur Instanz-von-Beziehungen zwischen zwei Ebenen, sondern auch Vererbungsbeziehungen



Problem bei üblicher Instanz-von-Beziehung zwischen Ebenen

- bestimmte Attribute von Klassen im Metamodell benötigen erst zwei Ebenen tiefer Werte

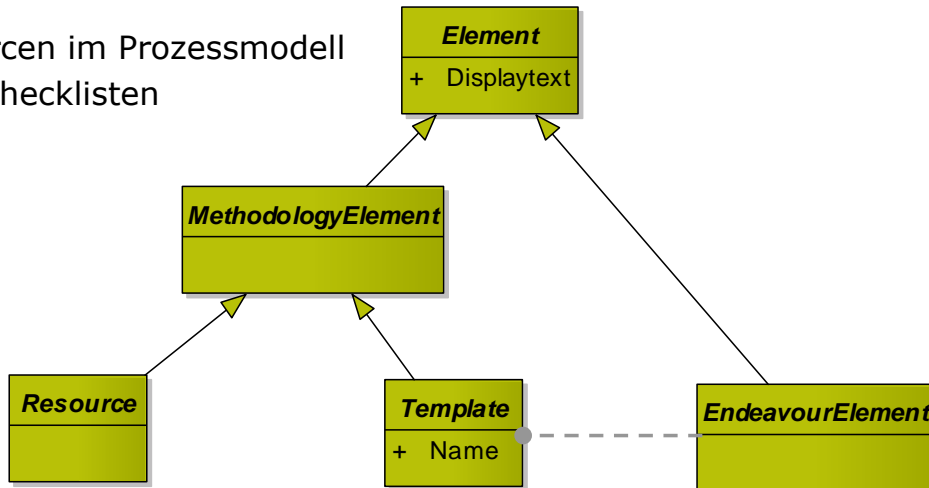


Veröffentlichung als ISO/IEC 24744 in 2007

Modellierung von Prozessmodell- und Projektebene

Ressourcen im Prozessmodell

- Checklisten



Strukturierung nach:

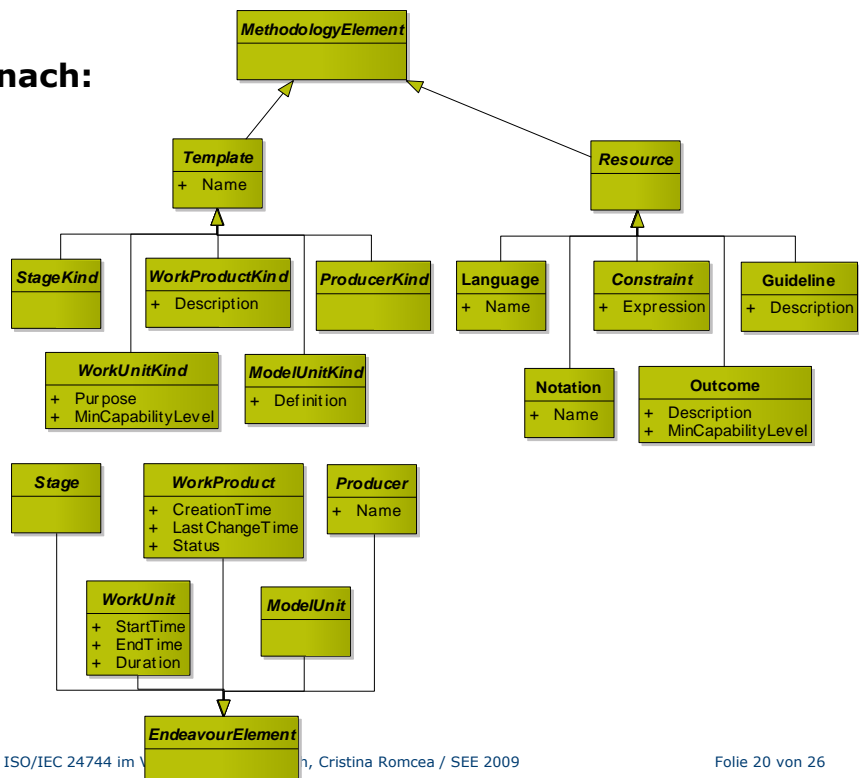
Prozess

- Arbeit
- Produkt
- Produzent
- zeitliche Aspekte

Modellierung

- Modell-einheiten

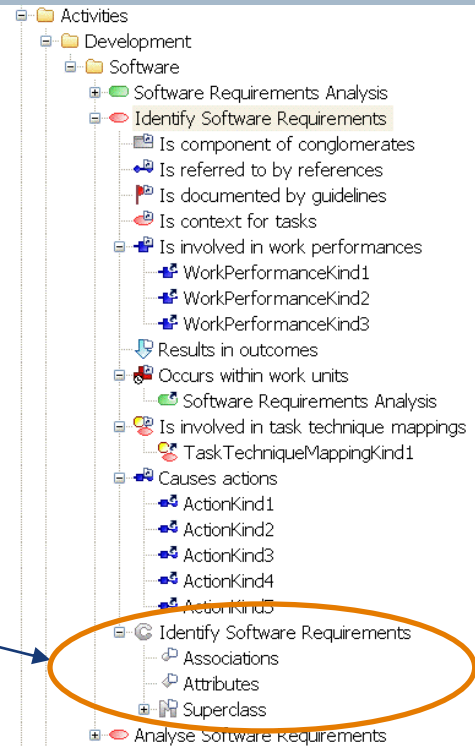
Ressourcen



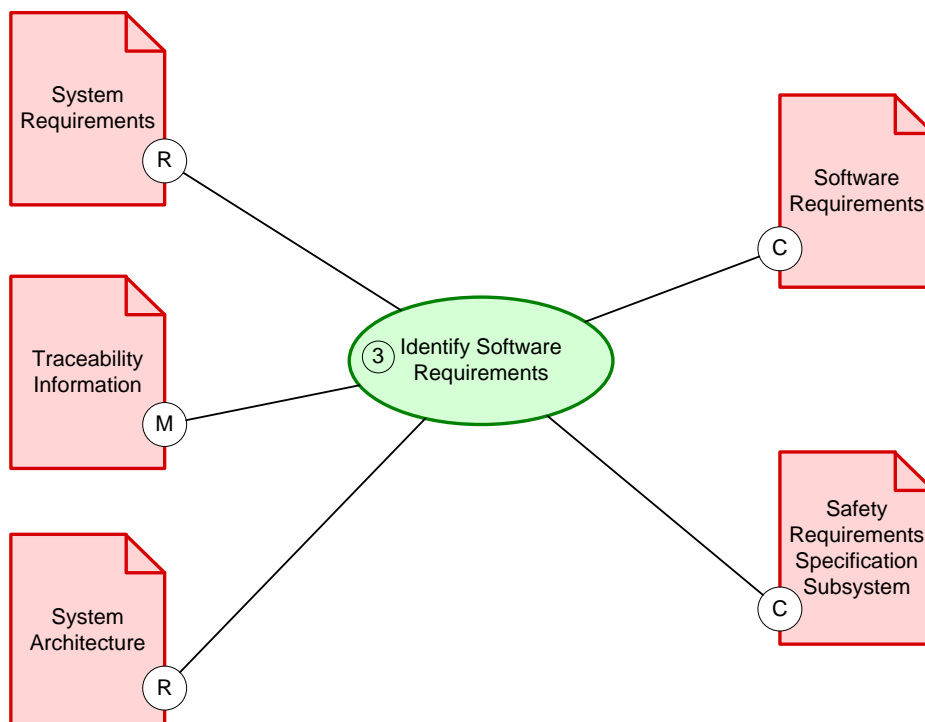
ISO/IEC 24744 – Beispiel einer Aktivität

Prozessmodellierungstool

- „Stages Composer“ von Method Park SW AG



ISO/IEC 24744 – ActionDiagram mit Visio



ISO/IEC 24744 – Vor- und Nachteile



Vorteile

- relativ klein
 - 27 Powertype Patterns + 10 weitere Klassen
 - gut strukturiert (Prozess, Produkt, Produzent)
- Regeln zur Benutzung
- Regeln zur Erweiterung

Nachteile

- P&C Entwicklungsprozess von Continental ließ sich mit ISO/IEC 24744 nicht vollständig abbilden (nur über Erweiterungen des Metamodells möglich)
- Erweiterungen nur eingeschränkt möglich
 - existierende Elemente dürfen nicht verändert werden
 - Bsp.: um neue Assoziation zwischen zwei Elementen einzuführen, müssen erst eigene Elemente definiert werden

Vergleich von SPEM und SEMDM



Aspekt	SPEM	ISO/IEC 24744
Komplexität	<ul style="list-style-type: none"> ▪ hoch ▪ schwer zu überblicken ▪ lange Einarbeitungszeit 	<ul style="list-style-type: none"> ▪ mittel ▪ gut strukturiert ▪ kurze Einarbeitungszeit
Benutzung	<ul style="list-style-type: none"> ▪ keine Regeln zur Benutzung ▪ viele Möglichkeiten (Metamodell, UML-Profil) 	<ul style="list-style-type: none"> ▪ Regeln zur Benutzung ▪ die Existenz von je zwei Klassen für ein Element ist verwirrend
Erweiterbarkeit	<ul style="list-style-type: none"> ▪ festes Metamodell ▪ keine Regeln ▪ drei Compliance Points 	<ul style="list-style-type: none"> ▪ festes Metamodell ▪ Regeln zur Erweiterung ▪ nur eingeschränkt
Process Enactment (Ausführung von Prozessen)	<ul style="list-style-type: none"> ▪ nur Export zu Projektmanagement-Tools und Workflow-Engines angedeutet 	<ul style="list-style-type: none"> ▪ wird durch Modellierung direkt unterstützt

Gegenwärtig vorhanden

- Metamodelle wie SPEM und ISO/IEC 24744 bieten eine gute Grundlage für die Definition von Entwicklungsprozess-Modellen
- fest vorgegebene Metamodelle sind aber zu starr, um wirklich jeden Prozess modellieren zu können
- die Spezifikationen von SPEM und ISO/IEC 24744 sind primär für Werkzeughersteller geeignet, nicht für Prozessmodellierer

Benötigt

- flexible Metamodelle, die auf einer gemeinsamen Basis beruhen
- Metamodelle, die Process Enactment unterstützen
- Unterstützung für Prozessmodellierer



Vielen Dank
für Ihre Aufmerksamkeit.



10.2. Werkzeuggestützte Konsistenzsicherung von Software-Entwicklungsprozessen

Dr. Frank Marschall

Software Engineering Quality, Quality Business Services, Systems Integration

T-Systems Enterprise Services, GmbH

Dachauer Str. 651

80995 München

Frank.Marschall@t-systems.com

Abstract

Der hier vorgestellte Beitrag beschreibt den Einsatz von Werkzeugen zur Entwicklung und Prüfung von Software-Entwicklungsprozessen und die damit verbundenen Erfahrungen in einem großen Software-Haus. Exemplarisch wird ein Werkzeug zur Simulation von Entwicklungsprozessen vorgestellt.

Ausgangssituation

Die Standardisierung von Entwicklungsprozessen im Software-Engineering stellt eine große Herausforderung für Unternehmen dar. Einerseits gilt es, zahlreiche bestehende Ansätze aus dem Unternehmen und Vorgaben aus Modellen und Standards wie CMMI zu einem durchgängigen, konsistentem Vorgehen zu vereinen. Andererseits müssen für unterschiedliche Projekttypen, wie z.B. Individualentwicklung, SAP-Projekte, etc, jeweils angepasste Standardprozesse zur Verfügung gestellt werden. Hierbei gilt es, möglichst große Teile des Vorgehens in den verschiedenen Prozessbeschreibungen wieder zu verwenden.

Innerhalb der T-Systems führt die Service Unit Systems Integration (SI) Großprojekte in der Software-Entwicklung durch. Der Software-Entwicklungsprozess der Systems Integration wird durch das Software-Engineering-Book (SE Book) beschrieben und laufend weiterentwickelt. So wurden aus dem bestehenden Entwicklungsprozess für Individualprojekte Standardprozesse für SAP-Projekte und Business-Intelligence-Projekte entwickelt. Weitere Varianten werden folgen. Verbesserungsinitiativen, wie ein internes CMMI-Projekt, Initiativen der unternehmensinternen Software-Engineering-Community und Feedback aus dem kontinuierlichen Verbesserungsprozess der SI, führen ca. alle sechs Monate zu einem neuen Release der Prozess-Familie.

Die wachsende Zahl von Standardprozessen für unterschiedliche Projekttypen und zusätzlichen Ausprägungen, die durch das Tailoring dieser Prozesse entstehen können, stellt hohe Anforderungen an die Qualitätssicherung der Prozesse. Für jede Version jeder Variante des SE Books muss sichergestellt werden, dass sie keine Inkonsistenzen, wie z.B. nicht verwendete Ergebnisse, überflüssige Vorlagen etc. enthält. Weiterhin muss gewährleistet werden, dass jede Variante alle Aktivitäten und Ergebnisse, die für die Projektdurchführung nötig sind, aufweist, aber keine unnötigen Ergebnisse oder undurchführbaren Aktivitäten enthält.

Der Vortrag stellt Ansätze und Techniken zur Entwicklung und Qualitätssicherung der Entwicklungsprozesse der T-Systems vor. Als wesentliche Voraussetzungen für eine effiziente Qualitätssicherung erweisen sich hierbei ein modellbasierter Entwicklungsprozess und der Einsatz darauf aufbauender Werkzeuge.

Ansatz

Inkonsistente Prozessbeschreibungen führen zu (berechtigten) Vorbehalten seitens der Anwender und verhindern in vielen Fällen eine sinnvolle Werkzeugunterstützung für die Prozessdurchführung. Um Qualität und Konsistenz der zahlreichen Prozessvarianten sicherzustellen, wird eine Reihe von eigens entwickelten Werkzeugen genutzt. Diese umfassen Werkzeuge zur Erzeugung von Standardprozessen aus Prozessbausteinen, zur Visualisierung der Prozesse, zur Simulation und Verifikation von Prozessabläufen und zur Überprüfung des Modells auf ungewünschte Eigenschaften. Typische Fehler, von mehrfachen Glossareinträgen über fehlende Vorlagen bis zu nicht mehr aktuellen URLs in Literaturverweisen, lassen sich anhand von automatisierten Überprüfungen, analog zu Regressionstests in der Software-Entwicklung, effizient automatisiert erkennen.

Der Vortrag gibt einen Überblick über die eingesetzten Werkzeuge und zeigt Nutzen und Grenzen einer werkzeugunterstützten Qualitätssicherung auf. Exemplarisch wird ein Werkzeug zur Simulation von Prozessabläufen vorgestellt, welches Traces möglicher Prozessabläufe erzeugt. Das Werkzeug muss hierzu einige Eigenschaften des SE Book Metamodells berücksichtigen:

- Jede Aktivität hat Mengen von Ergebnistypen als Ein- und Ausgabe. Für jede Ein- oder Ausgabe ist ein Zustand (in Bearbeitung, fertig gestellt, intern freigegeben, extern freigegeben) festgelegt. Der Eingabezustand legt fest, in welchem Zustand ein Ergebnis mindestens vorliegen muss, der Ausgabestatus bezeichnet den höchstwertigen Zustand, den das Ergebnis nach Abschluss der Aktivität erreichen kann.
- Einzelne Ein- und Ausgaben können optional sein (z.B. der Ergebnistyp Fehler-Ticket)
- Ergebnistypen sind hierarchisch organisiert. Ein Ergebnistyp kann sich also aus einer Menge untergeordneter Ergebnistypen zusammensetzen.

Das Werkzeug simuliert interaktiv oder durch Angabe einer Strategie mögliche Abläufe und liefert einen Trace aller Prozessschritte. Für jeden Schritt wird hinterlegt, warum einzelne Aktivitäten in der jeweiligen Situation nicht durchführbar sind.

Können bestimmte Aktivitäten eines Prozesses nicht durchgeführt werden, lassen sich anhand der Traces die Ursachen hierfür feststellen. So können beispielsweise wechselseitige Abhängigkeiten zwischen zwei oder mehr Aktivitäten erkannt werden. Simulierte Abläufe, die eigentlich nicht möglich sein sollten, weisen darauf hin, dass Autoren ihre Vorstellungen nicht adäquat modelliert haben.

Durch Umkehrung der Simulation können, ausgehend von einem Ergebnistyp in einem Zustand (z.B. Anforderungsspezifikation, extern freigegeben), minimale Mengen initial erforderlicher Ergebnistypen berechnet werden. Diese werden zum Projektstart benötigt, um das Ergebnis mit dem gegebenen Vorrat an Aktivitäten zu erzeugen (z.B. Projekt-Assignment, Vertrag und Angebotsunterlagen in einem bestimmten Zustand).

Zusammenfassung

Ein modellbasiertes, konsistentes Prozessmodell stellt die Grundlage für zahlreiche Werkzeuge zur Projektunterstützung, wie z.B. Werkzeuge zur Generierung von Projektplänen, dar. Die vorgestellte Werkzeugunterstützung kann eine inhaltliche, fachliche Prüfung eines Prozessmodells nicht ersetzen. Unsere Erfahrungen haben jedoch gezeigt, dass vie-

le der Techniken, wie Simulation, Visualisierung von Abhängigkeiten oder Regressions-tests, die in der Software-Entwicklung eingesetzt werden, auch für die Konsistenzprüfung modellbasierter Entwicklungsprozesse eine große Effizienzsteigerung mit sich bringen. Gerade durch Simulationsläufe lassen sich Modellierungsfehler und falsche Annahmen bereits während der Prozessmodellierung sehr effizient und interaktiv identifizieren und beheben.



SEE 2009

Werkzeuggestützte Konsistenzsicherung von Software-Entwicklungsprozessen

Dr. Frank Marschall



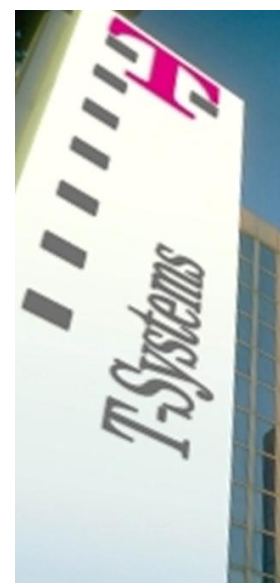
SEE 09

Dr. Frank Marschall, 27.05.2009

Page 1

Werkzeuggestützte Konsistenzsicherung Einleitung

- T-Systems Enterprise Services / Systems Integration
 - IT-Dienstleister für Großprojekte
 - 12.000 Mitarbeiter in Deutschland
 - Kooperation mit Cognizant (> 50.000 Mitarbeiter)
- Eigenes Vorgehensmodell zur SW-Entwicklung mit Varianten
- Erfahrungen bei Abbildungen auf Vorgehensmodelle von Kunden und Partnern
 - Oft kein Modell
 - Oft Modell ohne Semantik
 - Oft zahlreiche Inkonsistenzen
- Qualitätssicherung des eigenen Entwicklungsprozesses
 - Handwerkliche Fehler
 - Zahlreiche, wiederkehrende Modellierungsfehler
 - Zeitraubende Prüfungen



SEE 09

Dr. Frank Marschall, 27.05.2009

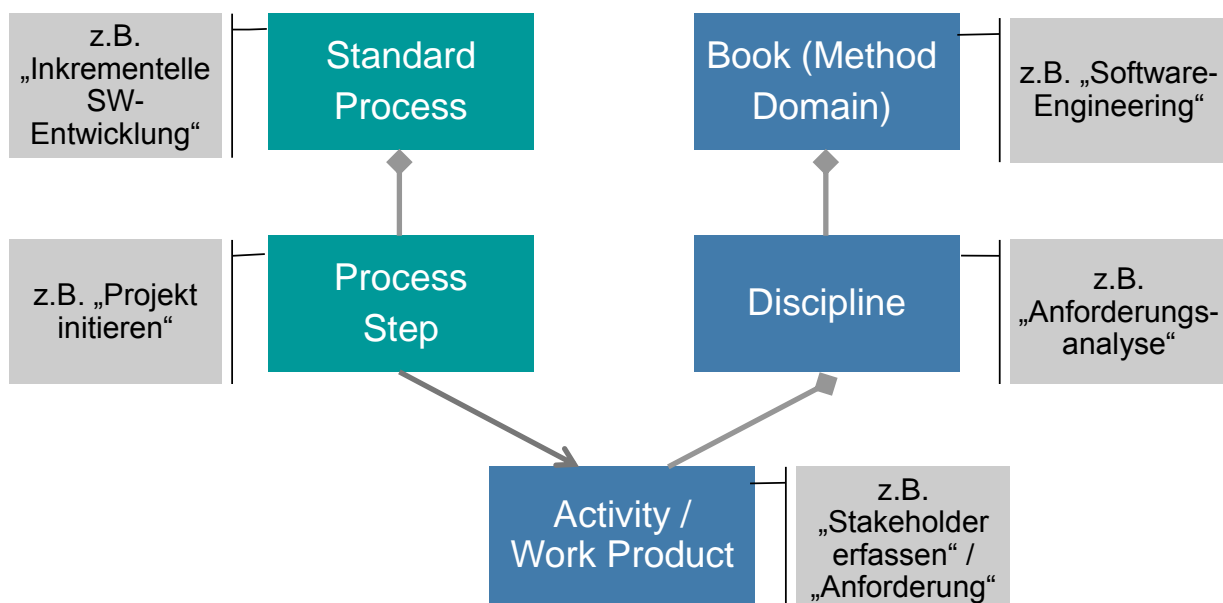
Page 2

Werkzeuggestützte Konsistenzsicherung Qualität von Vorgehensmodellen

- Inkonsistenzen und Lücken in Vorgehensmodellen
 - verwirren, halten auf oder leiten fehl
 - pflanzen sich in nachgelagerter Projektunterstützung fort
 - führen zu sinkender Akzeptanz der Anwender
- Lange Diskussionen um die „richtige Interpretation“
- Einheitliche Vorgehensweisen werden nicht umgesetzt
- Vorteile eines einheitlichen Vorgehensmodells, wie z.B.
 - unternehmensweite Prozessoptimierung,
 - einheitliche Dokumentation und Kommunikation,
 - Vergleichbarkeit von Projekten, etc
 gehen dem Unternehmen verloren
- Kaum weiterführende Unterstützung möglich, z.B.
 - Projektplangenerierung
 - Werkzeugunterstützung zur Aufwandsabschätzung,
 - Werkzeuggestützte Nachverfolgung des Projektfortschritts, etc.

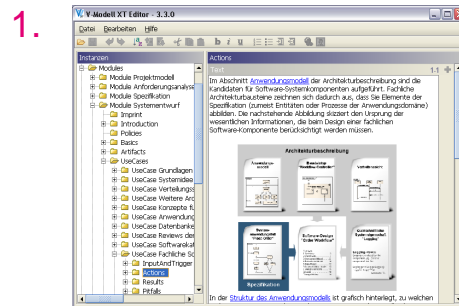


Werkzeuggestützte Konsistenzsicherung Prozessarchitektur

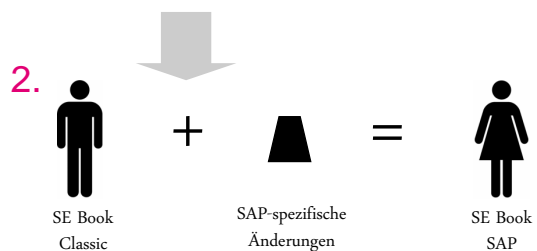


Werkzeuggestützte Konsistenzsicherung Von der Methodensammlung zum Prozess im Projekt

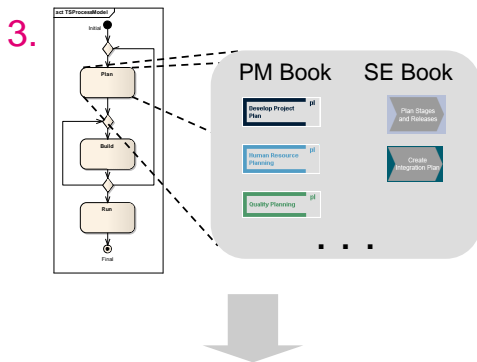
1. Entwicklung der SE-Methodenbausteine mit dem V-Modell XT Editor
 - Organisiert nach Disziplinen
 - Eigenes Metamodell
 - API zum Zugriff auf Modellelemente



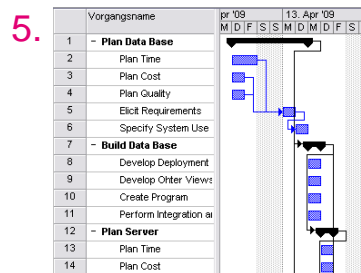
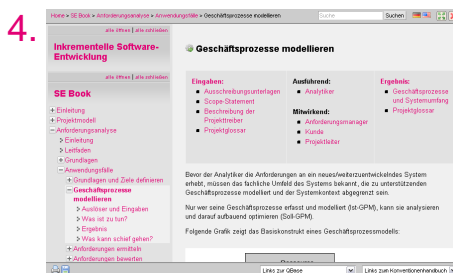
2. Technologie- und domänenspezifische Anpassung der Books durch automatisiertes
 - Hinzufügen neuer Elemente
 - Ersetzen bestehender Element
 - Ändern bestehender Elemente
 - Löschen bestehender Elemente



Werkzeuggestützte Konsistenzsicherung Von der Methodensammlung zum Prozess im Projekt



3. Konfiguration der Book-Inhalte zu Standardprozessen
4. Projektspezifisches Tailoring eines Standardprozesses und Generierung der Prozessbeschreibung
5. Instanzierung des getailorerten Prozesses im Projekt



Werkzeuggestützte Konsistenzsicherung Ausgangssituation

- Zahlreiche Standardprozesse für unterschiedlichen Anwendungsdomänen und Technologien
- Kontinuierliche Weiterentwicklung bestehender Entwicklungsprozesse
- Methodenautoren meist erfahrene Experten aus operativen Einheiten, jedoch oft ohne Erfahrungen in Modellierung von SW-Entwicklungsprozessen
- Problem: Effiziente Qualitätssicherung zahlreicher Varianten
 - Formale Fehler vermeiden
 - Nur sinnvolle Abläufe ermöglichen
 - Vermeidung von Inkonsistenzen
 - Keine überflüssigen Aktivitäten oder Ergebnisse
 - Fail Fast: Fehlentwicklungen und Missverständnisse frühzeitig erkennen (nicht erst im Freigabe-Review)

→ Werkzeugunterstützung zur Unterstützung der Qualitätssicherung



Werkzeuggestützte Konsistenzsicherung Ergebnistypen und Aktivitäten

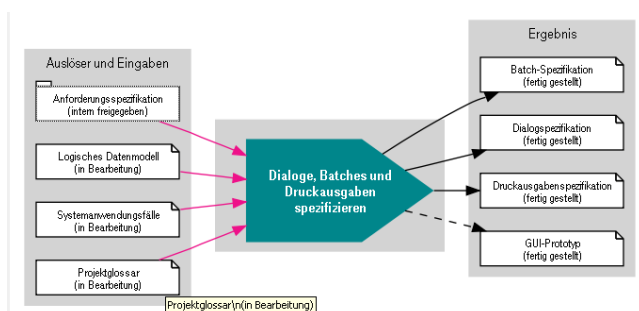
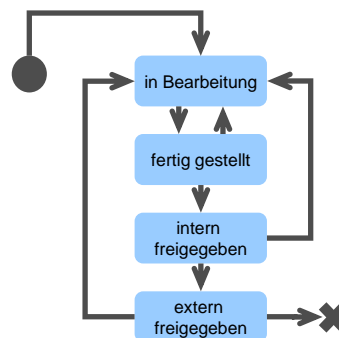
Ergebnistypen

- Hierarchisch organisiert
- Zustandsmodell

Aktivitäten

- Ergebnisse mit minimal gefordertem Zustand als Eingabe
- Ergebnisse mit bestmöglichem Zustand als Ausgabe
- Ergebnisse die als Ein- und Ausgabe auftauchen werden bearbeitet

→ Durch Ein-/Ausgaben der Aktivitäten ergeben sich Einschränkungen für mögliche Abläufe



Werkzeuggestützte Konsistenzsicherung „Statische Analyse“

- Evolutionär entstandenes Werkzeug zur Konsistenzsicherung
- Analog zu Regressions-Tests wird das Prozessmodell auf unerwünschte Eigenschaften untersucht

Beispiele für Fehler:

- Ein-/Ausgaben ohne Zustände
- Mehrfache, identische Ein-/Ausgaben
- Mehrfache Glossareinträge
- Fehlende / überflüssige Hilfsmittel
- Nicht erreichbare externe URLs
- ...

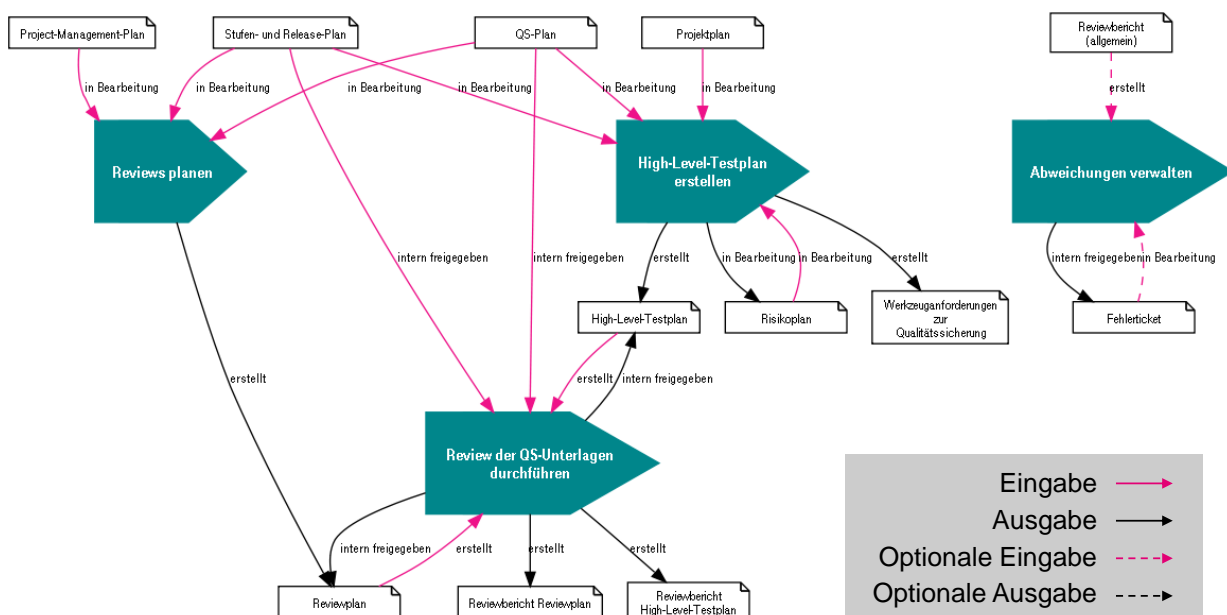
Beispiele für „Bad Smells“:

- Aktivitäten ohne Ein-/Ausgaben
- Nicht verwendete Ergebnistypen
- Leere Namens- oder Textfelder
- Nicht referenzierte Anhänge
- ...

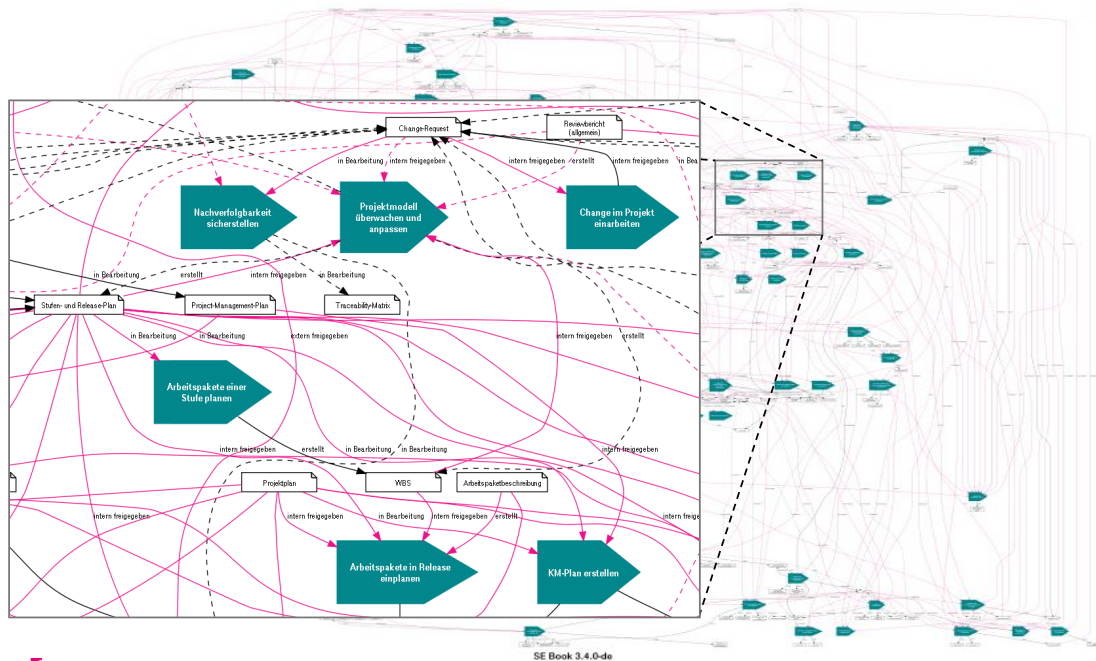
- Spart Zeit bei der Qualitätssicherung
- Erlaubt Fokussierung auf inhaltliche Fragen
- **Aber:** Keine Unterstützung bei Analyse der Abhängigkeiten und möglichen Abläufe



Werkzeuggestützte Konsistenzsicherung Abhängigkeiten innerhalb einer Disziplin



Werkzeuggestützte Konsistenzsicherung Abhängigkeiten innerhalb des SE Books Individual



Werkzeuggestützte Konsistenzsicherung „Dynamische Analyse“

- Ablauf der Projektdurchführung wird simuliert
- Einschränkungen zur Analyse
 - Ein Durchlauf ohne Iterationen
 - Keine Parallelität
 - Jede Aktivität wird höchstens einmal ausgeführt
 - Jedes Ergebnis existiert höchstens ein mal
 - Alle Aktivitäten werden atomar ausgeführt
- Einschränkungen erlauben dennoch Klärung der Fragestellung
- Vor der Simulation:
 - Initialzustand herstellen
 - Dekomposition aller Ergebnistypen



Werkzeuggestützte Konsistenzsicherung Simulation: Suchstrategien

▪ Fragestellungen

- Können mit den Aktivitäten alle erforderlichen Ergebnisse erzeugt werden?
- Gibt es überflüssige Aktivitäten oder Ergebnistypen?
- Funktioniert der Workflow wie gedacht?
- ...

Manuell gesteuerte Abläufe:

- Gewünschte Reihenfolge der Aktivitäten wird manuell (interaktiv) vorgegeben

→ Ist ein Ablauf wie gedacht möglich?
Ist ein Standardprozess so ausführbar?

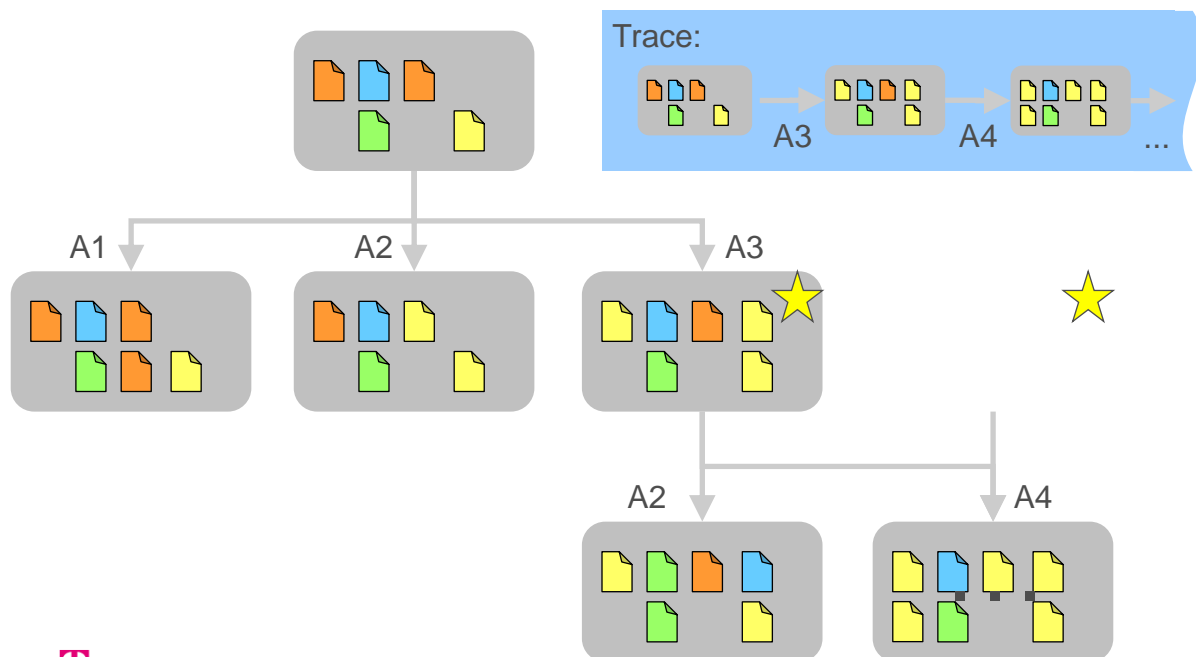
Generierung von Abläufen:

- Schlechtesten Zustand verfolgen
- Z.B. der, der die meisten höchstwertigen Ergebnisse beinhaltet

→ Gibt es unerwünschte Abläufe?
(z.B. solche, die Ergebnisse ohne die fachlich nötigen Eingaben produzieren)

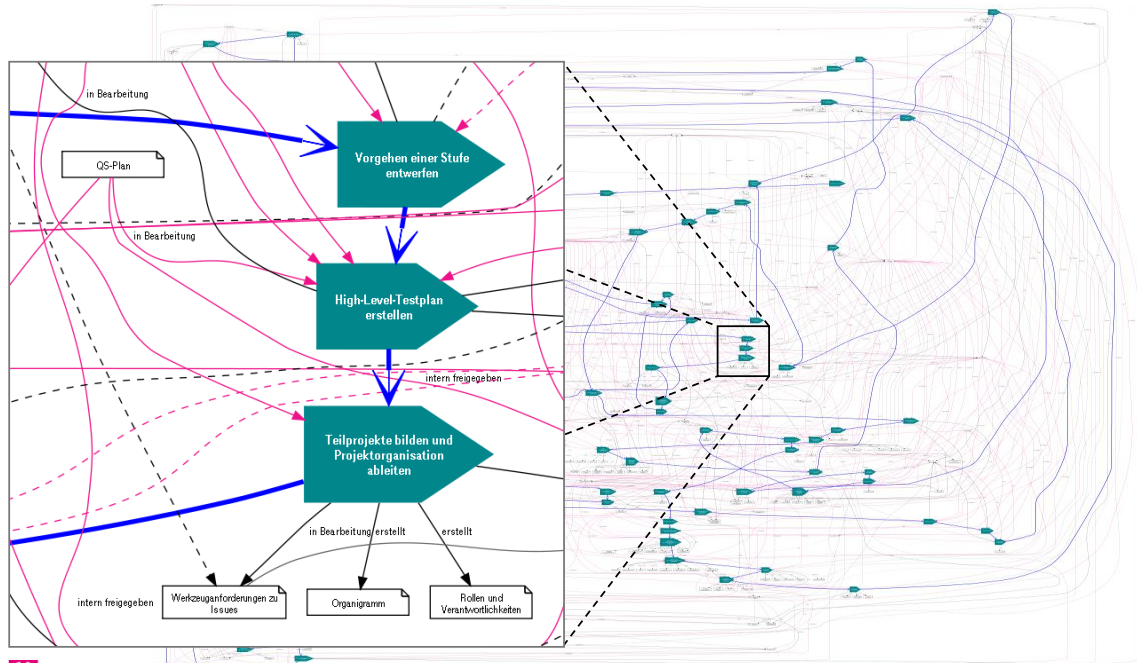


Werkzeuggestützte Konsistenzsicherung Generierung von (unerwünschten) Abläufen



Werkzeuggestützte Konsistenzsicherung

Simulation: Gefundener, möglicher Ablauf



Werkzeuggestützte Konsistenzsicherung

Ausgabe: Traces aus Zuständen

Bisher ausgeführte Aktivitäten:

Change-Management etablieren
 Grundlagen und Ziele definieren
 Geschäftsprozesse modellieren

...

Programm dokumentieren
 Programm testen

**Soll dieser Ablauf
 so möglich sein?**

Bisher erzeugte Ergebnisse:

Programmierauftrag (Finished)
 Project-Management-Plan (Approved)
 WBS (Approved)
 Projektplan (Approved)

...

Unit-Testfall (Approved)
 Quellcode (Approved)
 Testprotokoll Unit-Test (Finished)

**Ist alles vorhanden?
 Passt die Reihenfolge?**

Aktuell nicht ausführbare Aktivitäten:

Cannot apply

Reviews der Integration durchführen because
 of missing input: „Beschreibung der
 Projekttrandbedingungen
 (intern freigegeben), ...“

Warum geht das nicht???

Cannot apply

Review der Anforderungsspez durchführen
 because of missing input:
 Reviewplan (intern freigegeben)

Cannot apply

Review der QS-Unterlagen
 because of missing input:
 High-Level-Testplan (fertig gestellt), ...

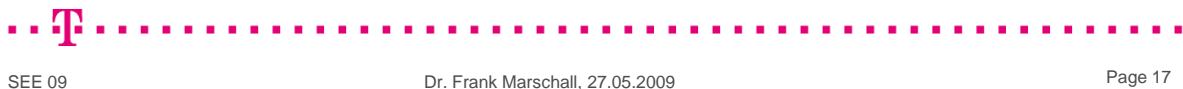
Cannot apply

High-Level-Testplan erstellen
 because of missing input:
 Stufen- und Release-Plan (in Bearbeitung)

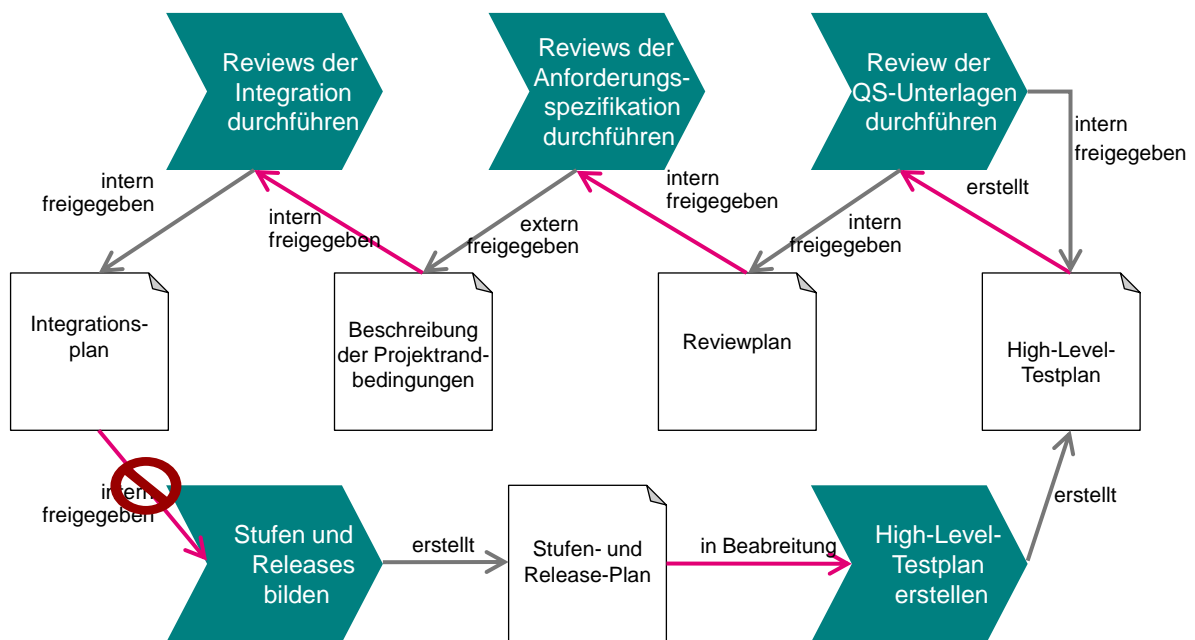


Werkzeuggestützte Konsistenzsicherung Typische, gefundene Fehler

- Ergebnisse liegen nach Durchlauf nicht (im gewünschten Zustand) vor
 - Ergebnis fehlt, weil ein Teilergebn nirgends erzeugt wird
 - Bestehender Ergebnistyp wurde nicht genutzt, sondern neuer Ergebnistyp erzeugt
 - Fehlende Eingaben aus Vorgängerprozessen (z.B. Sales-Prozess)
 - ...
- ➔ Formal nachweisbar:
Ohne Aktivitäten, die Ergebnisse in einen minderwertigeren Zustand überführen, liefert jeder mögliche Ablauf die selbe Ergebnismenge.
- Fehlerhafte Abläufe
 - Abläufe, die „eigentlich“ nicht möglich sein sollten
 - Nicht ausführbare Aktivitäten durch zyklische Abhängigkeiten
 - ...

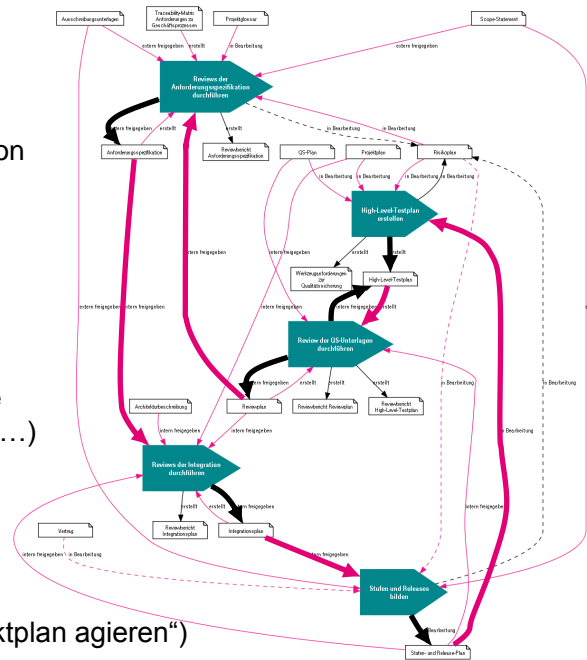


Werkzeuggestützte Konsistenzsicherung Zyklische Abhängigkeiten erkennen



Werkzeuggestützte Konsistenzsicherung Validierung von Traces

- Analyse der erzeugten Abläufe
 - erfordert Erfahrung
 - ist überwiegend Validierung, nicht Verifikation
- Vorgehensmodelle sind (meist) nicht formal vollständig und konsistent
 - Implizite Annahmen („Projektplan ist (fast) immer erforderlich.“)
 - Generische Ergebnisse (Reviewbericht)
 - Fehlende Ein-/Ausgaben für Ausnahmefälle (Fehlerticket, Change-Request, Risikoliste, ...)
 - Generische Aktivitäten („Abweichungen verwalten“)
- ➔ Formalen Schwächen berücksichtigen
 - ➔ Constraints in der Simulation (z.B.: „Nur der Projektleiter darf ohne Projektplan agieren“)
 - ➔ Durch erweiterte Modellierung (z.B. VMXT-Produktabhängigkeiten)



Werkzeuggestützte Konsistenzsicherung Mögliche Verbesserungen und Erweiterungen

- Auflösen der Atomizität
- Rückwärtssimulation: Welche Ergebnisse müssen initial vorliegen?
 - ➔ Können Techniken aus dem Model-Checking helfen?
- Abbildung realer Projekte (Stufen, Iterationen, ...):
 - ➔ Ergebnisse müssen Kontext kennen
 - ➔ Einsatz bei der Generierung von Projektplänen

Werkzeuggestützte Konsistenzsicherung

Zusammenfassung



No magic ...



... but useful

- Vorgestellte Techniken erlauben keinen automatisierten Korrektheitsnachweis, aber
 - sind sehr effiziente Hilfsmittel bei der Fehlersuche
 - ersparen viele manuelle Überprüfungen
 - liefern sofort Feedback
 - sind mit wenig Aufwand zu realisieren
- Sinnvolle Ergänzung zu bestehenden QS-Maßnahmen
- Ermöglichen Schaffung konsistenter Vorgehensmodelle
 - Verbesserte Nutzerakzeptanz
 - Voraussetzung für weitergehende Werkzeugunterstützung
- Voraussetzungen
 - Metamodell mit klarer Semantik
 - Technischer Zugriff (API)
 - Explizites Wissen auch um Schwächen der Modellierung



10.3. Erstellung und Bereitstellung von Prozessbeschreibungen, die Mitarbeiter nützlich finden

Uwe Schmitz
wibas IT Maturity Services GmbH
Otto-Hesse-Str. 19B
64293 Darmstadt
uwe.schmitz@wibas.de

Abstract

Es werden leider immer wieder (zu) umfangreiche Prozessbeschreibungen erstellt, die an den Bedürfnissen der Projekte und Mitarbeiter vorbeigehen. Das Ergebnis sind „Papier-“ oder „Intranetmonster“, die weniger Mehrwert für die Geschäftsziele der Unternehmen darstellen, als von allen Beteiligten gewünscht. Dieser Vortrag beleuchtet die notwendigen Aspekte, um typische Fehler zu vermeiden. Er diskutiert das zielorientierte Vorgehen, um Mehrwert bringende Prozessbeschreibungen zu erstellen und bereitzustellen. Hierbei werden Methoden und Tools gezeigt, die es Ihnen erlauben praktische Prozessbeschreibungen zu erstellen, über die die Beteiligten hinterher sagen: „Ja, sie haben mir geholfen“.

Motivation

Zu viele Prozessbeschreibungen werden von Mitarbeitern nicht genutzt. Die Ursachen hierfür sind vielschichtig: Die Prozessbeschreibungen sind zu umfangreich. Sie stellen „Theorie“ dar und gehen somit an den Bedürfnissen der Anwender vorbei. Es werden Prozessbeschreibungen von Prozessgruppenmitarbeitern erstellt, die aber nicht ausreichend mit den Praktikern (sprich Kunden der Prozessbeschreibungen) abgestimmt sind. Es werden initial zu viele Dinge beschrieben, die in der Form noch nicht „ausprobiert“ (pilotiert) wurden. Die Beschreibungen sind entweder sehr generisch, somit zwar „nicht falsch“, aber auch nicht hilfreich - oder sehr detailliert und spezifisch, so dass sie nur auf wenige Projekttypen zutreffen.

Die Konsequenz solcher Fehler ist: Die hohen Investitionen in das Erstellen von Prozessbeschreibungen zahlen sich nicht aus.

Dieser Beitrag beschreibt wie man solche Fehlermuster bei der Erstellung von Prozessbeschreibungen vermeidet und zeigt stattdessen Prinzipien auf, um zu Mehrwert bringenden Prozessbeschreibungen zu kommen. Es werden anhand des wibas Expertise Portals praktische Beispiele gegeben.

Ansatz

Erstellen Sie keine umfangreichen, gedruckten Werke. Die Suchmöglichkeiten sind begrenzt. Man hat sie nicht zur Hand, wenn benötigt. Sie sind schneller veraltet, als sie verteilt werden können. Es ist keine Integration in elektronische Workflows möglich. Stattdessen sollten die Beschreibungen in Web Portalen verfügbar sein. Top Level Darstellung sollten als Ausdrucke (Poster, Flyer) verfügbar sein. Selektiv sollten Subprozesse und Methoden ausdrückbar sein.

10.3. Erstellung und Bereitstellung von Prozessbeschreibungen

Die Prozessbeschreibungen sollten direkt in eine Projektplanung umsetzbar sein. Allerdings sind die Prozessbeschreibungen als „Starter-kits“ für die Projekte zu sehen. Es sind keine 1:1 Vorlagen, die ohne Mitdenken in Projektpläne zu transformieren sind.

Es gibt nicht „die eine“ Prozessbeschreibung, die für alle Projekttypen gleichermaßen zutrifft. Stattdessen muss es mit Hilfe eines einfachen Satzes von Kriterien möglich sein, aus den Prozessbeschreibungen das adäquate Tailoring für ein spezifisches Projekt zu erzeugen. Hierbei sind z.B. Projektgröße, Komplexität, Budget, Projektlebenszyklusmodell zu berücksichtigen.

Verwenden Sie zunächst Übersichten. Modellieren Sie detaillierte Flows nur, wenn Sie diese ausprobiert haben und die Beteiligten diesen Detaillierungsgrad für notwendig befunden haben.

Die Informationen müssen innerhalb der Prozessbeschreibungen einfach zu finden sein. Es dürfen nur wenige Schachtelungsebenen vorhanden sein und es müssen mächtige Suchfunktionen bereitgestellt werden. Definieren Sie Ihre Prozesse inkrementell und stellen Sie die Prozessbeschreibungen entsprechend bereit. Achten Sie darauf, die Organisation nicht mit einer zu großen Menge an Informationen/Vorgaben zu überfordern. Fokussieren Sie sich zunächst und erweitern Sie Ihre Prozessbeschreibungen auf der Basis gemachter Erfahrung. Involvieren Sie Mitarbeiter aus den Projekten so früh wie möglich in die Prozessdefinition und -pflege, also auch in die Erstellung der Prozessbeschreibungen. Wenn keine Mitarbeiter aus den Projekten zur Verfügung stehen, stoppen Sie Ihre Definitionsarbeit, eskalieren Sie den Zustand und finden Sie mit dem Higher-Level Management eine Lösung.

Messen Sie kontinuierlich, ob Ihre Prozessbeschreibungen tatsächlich verwendet werden. Unter anderem sollte Ihr Tooling hierzu die Aufrufe pro Zeiteinheit messen, auswerten und darstellen können. Managen Sie die Erwartung der Kunden (sprich: Projekte, Mitarbeiter). Versprechen Sie realistische Eigenschaften der Prozessbeschreibungen. Machen Sie deutlich, was die Mitarbeiter bei dem Aufsetzen der Projekte (sprich: Instantiieren der Prozesse) zu tun haben.

Lassen Sie die Projekte nicht mit den Prozessbeschreibungen allein. Unterstützen Sie die Projekte bei der Anwendung der Prozesse (kontinuierliches Coaching!!!) Verfassen Sie die Prozessbeschreibungen in der Sprache der Mitarbeiter - übernehmen Sie nicht 1:1 die Sprache der unterliegenden Modelle (CMMI, SPICE, PMI ...) Wählen Sie ein Tool, welches den Mitarbeitern erlaubt Feedback einfach einfließen zu lassen und den Prozessverantwortlichen dennoch die „redaktionelle Hoheit“ vorbehält.

Bewertung

Wenn sie richtig gemacht werden, sind Prozessbeschreibungen wertvolle Assets einer Organisation. Vermeiden Sie theoretische Werke. Gestalten Sie stattdessen Ihre Prozessbeschreibungen auf schlanke Art und Weise mit Hilfe von Prozessgruppenmitarbeitern und Praktikern in einem einfachen Tooling. Stellen Sie eine technisch gut handhabbare Möglichkeit bereit, Prozessbeschreibungen zu erstellen, zu publizieren und zu pflegen. Legen Sie Wert auf eine Rückmeldeschleife aus der Praxis.

Bei der Berücksichtigung dieser Prinzipien kommen Sie zu Prozessbeschreibungen, die einen echten Mehrwert für Ihre Organisation darstellen und die von den Mitarbeitern als wertvoll beurteilt werden.

Uwe Schmitz

Erstellung und Bereitstellung von Prozessbeschreibungen, die Mitarbeiter nützlich finden

Software & Systems Engineering Essentials (SEE), Berlin, 27.05.2009



Veränderung führen. Wissen teilen.



Ein mögliches Problem: Prozessbeschreibungen stellen weniger Mehrwert für die Organisation dar, als von allen Beteiligten gewünscht.

- „Eigentlich“: Prozessbeschreibungen sind wertvolle Assets der Organisation. Sie definieren
 - » eine gemeinsame Sprache
 - » ein gemeinsames Verständnis über die Arbeitsweisen
 - » ...
- Die gelebte Realität sieht anders aus
 - » „Das passt bei uns nicht“
 - » „So kann ich das nicht machen“
 - » „Ich muss Termine halten“
 - » ...

Prozessbeschreibungen und gelebte Realität
haben häufig wenig miteinander gemein



- 1 Große Prozessbücher
- 2 Prozesse sind Theorie
- 3 „Die eine“ Prozessbeschreibung



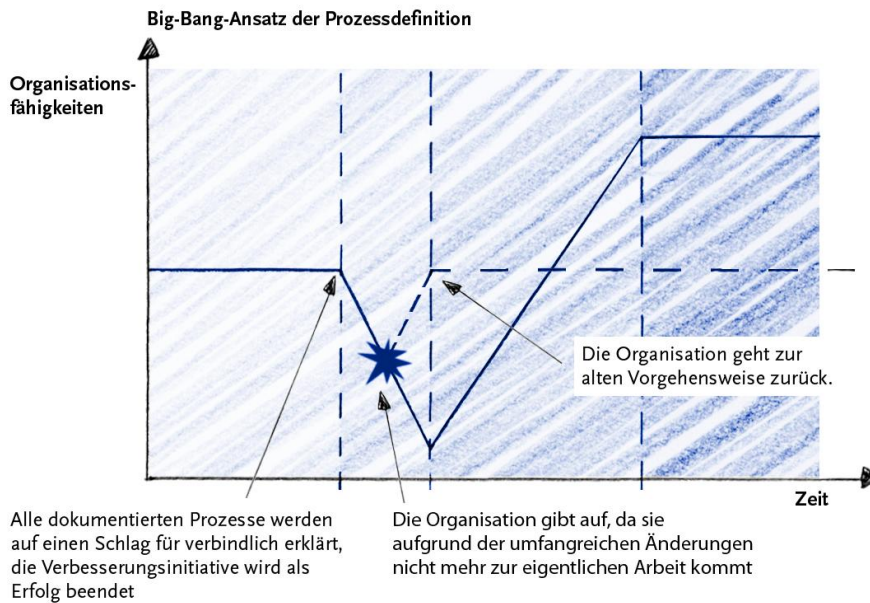
Fehlermuster: Große Prozessbücher stellen eine große Hürde für die Anwender dar.

- Prozessbeschreibungen definieren zu viele Details
- Benötigte Information ist nicht einfach zu finden
- Die Anwender sind überfordert





Effekt: Die Anwender gehen wieder zu ihrer alten Arbeitsweise zurück.



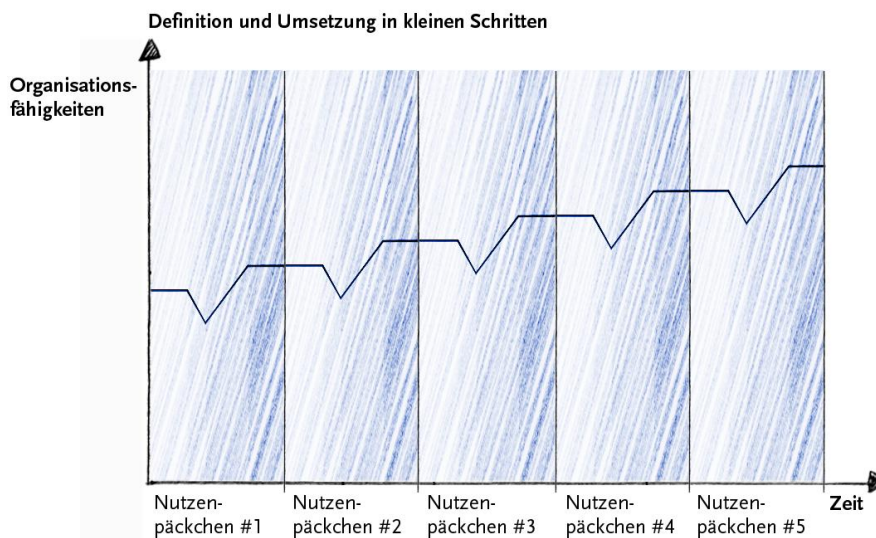
SEE 2009

- 5 -

© 2009 wibas GmbH



1. Fokussieren Sie sich zunächst und erweitern Sie Ihre Prozessbeschreibungen auf der Basis gemachter Erfahrung.



SEE 2009

- 6 -

© 2009 wibas GmbH



2. Strukturieren Sie die Prozessbeschreibungen in kleinen Einheiten.

- Verwenden Sie zunächst Übersichten
 - » Diese Übersichten sollten als Ausdrucke (Poster, Flyer) verfügbar sein
- Es dürfen nur wenige Schachtelungsebenen vorhanden sein
- Es müssen mächtige Suchfunktionen bereitgestellt werden
- Modellieren Sie detaillierte Flows nur, wenn die Anwender
 - » diese ausprobiert haben
 - » diesen Detaillierungsgrad für notwendig befunden haben



1	Große Prozessbücher
2	Prozesse sind Theorie
3	„Die eine“ Prozessbeschreibung



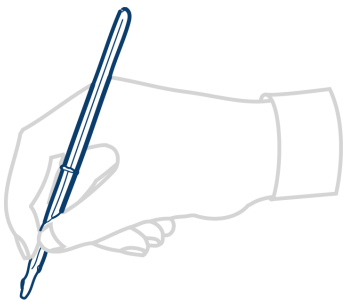
Fehlermuster: Prozessbeschreibungen werden ohne die Beteiligung der Anwender erstellt.

- Anwender empfinden die Prozessbeschreibungen als
 - » nicht nützlich oder sogar nicht anwendbar
 - » zu schwergewichtig
 - » zuviel Overhead
 - » ...

➤ Vorwurf des Elfenbeinturms



Effekt: Die Anwender wehren sich gegen die Ausführung der Prozessbeschreibungen.



Die Prozesse werden entwickelt



Die Prozesse werden ausgerollt

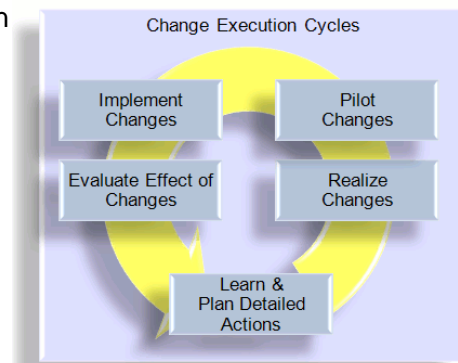


„Das passt bei uns nicht“
„So kann ich das nicht machen“



Involvieren Sie die Anwender im gesamten Lebenszyklus der Prozessentwicklung und –bereitstellung.

- Erarbeiten Sie die Prozessbeschreibungen zusammen mit den Anwendern
 - » Die Sprache der Anwender wird in den Prozessbeschreibungen genutzt
- Pilotieren Sie die neuen Prozesse
 - » Die Prozessbeschreibungen sind in der Planung der Projekte nutzbar
- Unterstützen Sie die Anwender bei der Nutzung
 - » Schwachstellen werden aufgedeckt und können behoben werden
- Messen Sie die Nutzung
 - » z.B. Aufrufe pro Zeiteinheit (Welche Prozesselemente werden genutzt?)
- Holen Sie Feedback ein
 - » Lassen Sie die Anwender die Prozessbeschreibungen selbst verbessern



1	Große Prozessbücher
2	Prozesse sind Theorie
3	„Die eine“ Prozessbeschreibung



Fehlermuster: Es gibt nicht „die eine“ Prozessbeschreibung, die für alle Projekttypen gleichermaßen zutrifft.

- Ein generischer Prozess für Alle (one size fits all)
 - » Wieder die Gefahr des Vorwurfs der Theorie (siehe Fehlermuster 2)
- Viele Projekttypen mit vielen Details
 - » Wieder die Gefahr der großen Prozessbücher (siehe Fehlermuster 3)



Definieren sie so viel wie nötig und so wenig wie möglich.

- Definieren Sie einen einfachen Satz von Kriterien, die ein adäquates Tailoring für ein spezifisches Projekt ermöglichen.
 - » Beispiele für solche Kriterien: Projektgröße, Komplexität, Budget, Projektlebenszyklusmodell
- Machen Sie deutlich, was die Mitarbeiter bei dem Aufsetzen der Projekte (sprich: Instantiieren der Prozesse) zu tun haben
- Die Prozessbeschreibungen inkl. der Tailoring-Kriterien sollten direkt in eine Projektplanung umsetzbar sein



Diskussion

Wie erreiche ich das bisher gesagte?

Wie involviere ich die betroffenen Personen?

Wie vermeide ich zu viele Details?

Wie?



Ein paar „wies“ (für die „Prozessler“)

- Gehen **Sie** zu den Projekten
- Verstehen Sie sich eher als Coach, der die Projekte *enabled*, anstatt jemand, der alles besser weiß und den Projekten Vorschriften macht
- „Schwitzen“ **Sie** in den Projekten/mit den Projekten
- Die „richtigen“ Prozessler müssen zu den Projekten gehen
- Erkennen Sie Überforderungssituationen der Projekte und reagieren Sie entsprechend
- Sind Sie kurzfristig zu Kompromissen bereit, die unterhalb Ihrer Qualitätsanforderungen liegen und haben Sie eine Strategie zukünftig den Rest zu erreichen
- Verankern Sie ernsthafte Lessons Learned / Retrospektiven in Ihren Entwicklungsprozessen
-



Zusammenfassung



Die Fehlermuster können mit wenigen aber nachhaltigen Korrekturen einen größeren Mehrwert für die Organisation schaffen.

Fehlermuster

- Große Prozessbücher
- Prozesse sind Theorie
- „Die eine“ Prozessbeschreibung

Korrektur

- Fokus
- Kleine Einheiten
- Involvierung der Anwender
- So viel wie nötig, so wenig wie möglich



Die Referenten



Uwe Schmitz
Dipl.-Ing.
Senior Executive Consultant
CMMI Assessor, Certified Scrum Master, Scrum Practitioner, Agile Coach
Mobil: +49 172 660 88 89
E-Mail: uwe.schmitz@wibas.de



VERÄNDERUNG FÜHREN. WISSEN TEILEN.



11. Session 11: Test und Wartung von IT-Systemen

Inhalt

11.1. Funktionales Agiles Testen durch einen Keyword-Driven Ansatz . . .	436
11.2. SW quality improvements by using quality KPIs	446
11.3. Effort Reduction and Error Prevention during Software Development and Maintenance	467

11.1. Funktionales Agiles Testen durch einen Keyword-Driven Ansatz

Hans-Joachim Brede
Bredex GmbH
Mauernstr. 33
38100 Braunschweig
hans-joachim.brede@bredex.de

Abstract

Dieser Vortrag stellt den Ansatz des Keyword-Driven Testens vor. Es wird erklärt, warum diese Methode sich besonders gut für agile Funktionstests durch die GUI eignet.

Motivation

Das Ziel der agilen Software-Entwicklung ist die regelmäßige Lieferung getesteter, funktionierender Software. Dabei setzt der Entwicklungsprozess auf Kommunikation und Flexibilität. Um eine fehlerfreie Software nach einem kurzen Zyklus herauszugeben, muss man ebenso im Testprozess agil vorgehen.

Agiles Testen hat schon im Bereich JUnit-Tests Erfolg genossen. Agile Funktionstests erweisen sich allerdings als schwieriger einzusetzen. Unter "Funktionstests" verstehen wir das Testen einer Anwendung (Workflows, Business-Logik) durch die GUI. Eine große Schwierigkeit bei agilen Funktionstests liegt darin, dass Anforderungen, Workflows und die GUI selbst häufig verändert werden. Daraus resultiert ein Bedarf an Testwartung. Gerade beim Testen durch die GUI können solche Wartungsarbeiten mühsam und zeintensiv sein. Ein zweites Problem stammt aus den kurzen Zyklen im agilen Projekt. Da es keine traditionelle Test- oder Abnahmephase gibt, muss das Testen parallel zu der Entwicklung stattfinden. In vielen Projekten wird gewartet, bis die GUI fertig ist, um mit der Testautomatisierung anzufangen. Für einen agilen Prozess ist dies allerdings zu spät.

Aus diesen Problemen kann man schließen, dass ein agiler Testprozess besondere Herausforderungen hat. Man muss eine zeitnahe Testautomatisierung leisten können - sprich, eine zunächst von der GUI unabhängige Testerstellung. Außerdem muss man dafür sorgen, dass Tests ein hohes Grad an Wartbarkeit haben, um Tests schnell an die neusten Veränderungen anzupassen.

Ansatz

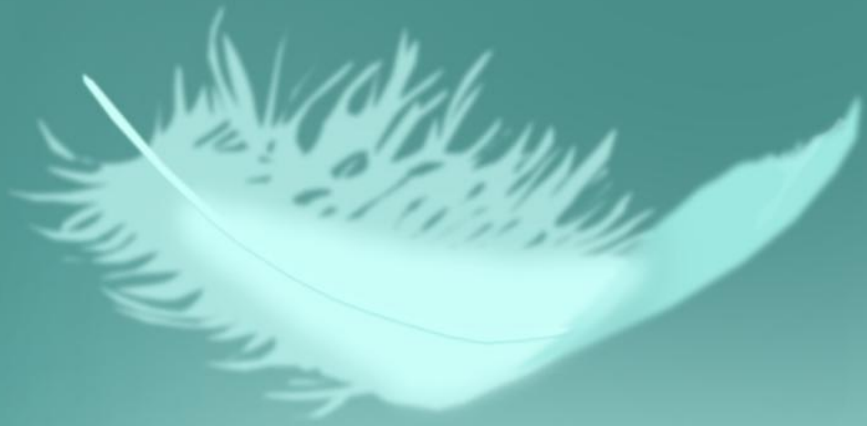
Ein sehr häufig verwendeter Ansatz, Funktionstests zu erstellen, ist das Aufnehmen von Aktionen in einer laufenden Anwendung (Capture-Replay). Die aufgenommenen Skripten können anschließend bearbeitet (modularisiert, parametrisiert) werden. Allerdings ist dieser Ansatz für agile Prozesse eher schlecht geeignet. Zum Einen, weil die Testautomatisierung erst anfangen kann, wann die GUI fertig ist. Weiterhin kann man nur so viel aufnehmen, wie schon funktioniert. Tests aufzuzeichnen birgt außerdem die Gefahr, dass man gegen die Implementierung und nicht gegen die Anforderungen testet. Auch

nach dem Aufnehmen ist die Testautomatisierung längst nicht fertig. Um aufgenommene Tests überhaupt später wiederausführen oder warten zu können, müssen Skripte zunächst programmatisch angepasst werden - Implementationsdetails und Daten müssen ausgelagert werden, und Module erstellt. Im Gegensatz dazu bietet der Keyword-Driven Ansatz für das agile Funktionstesten eine Möglichkeit, die oben genannten Probleme zu lösen. In diesem Ansatz wird den gleichen Prinzipien der Software-Entwicklung gefolgt: modularer Aufbau von Anfang an, Wiederverwendung (Referenzierung) von Modulen und dadurch eine einfache Pflege. „Keywords“ sind Module der Testfunktionalität, die mit einem Namen (Schlüsselwort) versehen werden, um deren Funktionalität zu erklären. Simple Keywords sind z.B. „OK Button Klicken“ oder „Benutzername Eingeben“. Daraus können wiederum komplexe Keywords entstehen, beispielsweise „Login Durchführen“, „Maske Ausfüllen“. Keyword Tests bestehen aus Sequenzen solcher Bausteine. So sind Keywords dem agilen Funktionstest gerecht. Schon aus den Anforderungen und parallel zu der Software-Entwicklung können Keywords entworfen und zusammengestellt werden. Das Warten auf eine GUI entfällt. Die Modularität und Wiederverwendung ermöglichen ein exponentielles Testwachstum, und reduzieren den Wartungsaufwand: Veränderungen können zentral durchgeführt werden um einen ganzen Test zu aktualisieren und Module sind leicht auszutauschen.

Es gibt schon Tools und Frameworks auf dem Markt, die Keyword-Driven Testen unterstützen. Manche erfordern das programmatische erstellen von Keywords, z.B. FIT (für White-Box Tests, die nicht über die GUI laufen) oder Quality-Center. Andere stellen vorgefertigte Keywords zur Verfügung und erfordern dadurch für einen Black-Box GUI-Test keinen Programmieraufwand (z.B. GUIDancer).

Bewertung

Keywords eignen sich allgemein gut für die Automatisierung von Funktionstests, weil diese aus vielen wiederkehrenden Aktionen bestehen, die sich sehr gut in Keywords abbilden lassen. Auch die Modularität ist ein großer Vorteil in Angesicht der vielen Veränderungen die den Funktionstest betreffen können. Der besondere Charm für agile Funktionstests liegt darin, dass Tests frühzeitig erstellt werden können. Somit bleibt die Testautomatisierung im Schritt mit der Entwicklung und sichert dadurch, dass jede neue Version gut getestet ist.



Funktionales Agiles Testen

Ein Keyword-Driven Ansatz

Hans-Joachim Brede, BREDEX GmbH

© 2008 Brede GmbH

Version 2.3



Agenda

- ▶ **Agile Entwicklung – Agiles Testen**
- ▶ **Funktionale Agile Tests – Herausforderungen**
- ▶ **Der Keyword-Driven Ansatz**
- ▶ **Vorteile für Projekte**



Agilität im Prozess

▶ Agile Prozesse

Regelmäßige Lieferungen funktionierender Software
Flexible Anforderungen

▶ Softwarequalität muss gewährleistet werden

Testprozess muss auch agil sein
Test-First, TDD z.B. JUnit Tests
... es fehlen allerdings die Akzeptanztests (z.B. GUI Tests)

Keywords 19.05.2009 3



Agiles Funktionales Testen

▶ Testen einer Anwendung durch die GUI

Business-Logik
Workflows
Häufig manuell getestet

▶ Müssen im agilen Prozess automatisiert werden

Automatisierter build&test Prozess
Zeitnahe Testergebnisse & Feedback

Keywords 19.05.2009 4



Herausforderungen

► Früher Anfang mit Testaktivitäten

Gleichzeitig mit Entwicklung, aus den Anforderungen
Test & Entwicklung parallel stattfinden
Idealerweise – vor der Entwicklung! Test-First

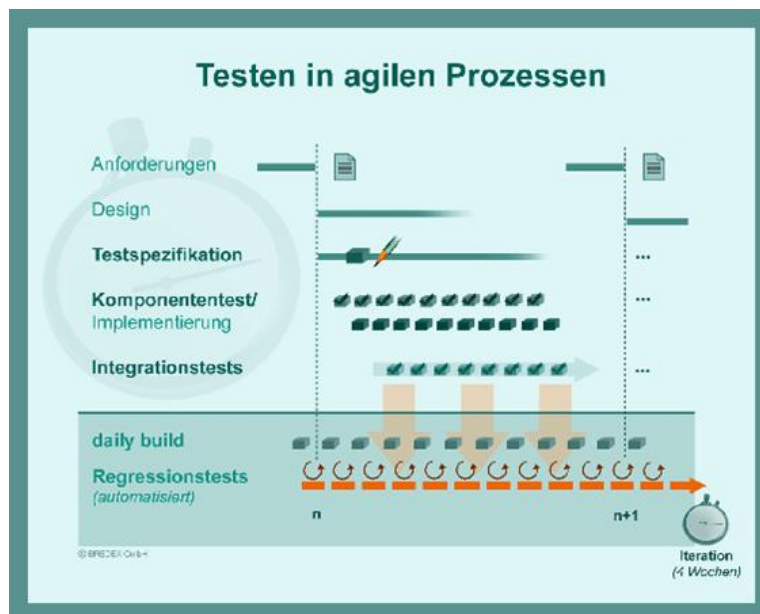
► Testwartung

Workflows, GUI, Anforderungen verändern sich häufig
Tests müssen gegenüber täglichen Veränderungen aktuell
bleiben

Keywords 19.05.2009 5



Testen im agile Prozessen



Keywords 19.05.2009 6



Neues Denken für den Test erforderlich

▶ **Aufnehmen und Wiederabspielen nicht geeignet**

Warten, bis die Anwendung verfügbar ist

Kann nur schon funktionierende Teile testen

An Implementationsdetails gebunden

Programmaufwand, um Wartbarkeit zu gewährleisten

Testet IST-Zustand, nicht SOLL-Zustand

Keywords

19.05.2009

7



Der Keyword-Driven Ansatz

▶ **Basiert auf gleiche Prinzipien der SW-Entwicklung**

Do it once and only once!

Modularität von Anfang an

Wiederverwendbarkeit ergibt Wartbarkeit

▶ **GUI Tests bestehen aus wiederkehrende Aktionen**

Zu Bibliotheken zusammengefasst

Tester kann Bibliotheken benutzen

Kleine Keywords kombinieren – größere Keywords

Keywords

19.05.2009

8



Keywords

▶ Mit sprechendem Namen versehen

Enter Username in Login Dialog (param: USERNAME)

Close Search Dialog with OK Button

Fill in New Project Wizard (param: PROJECT_NAME)

▶ Hierarchischer Aufbau

Open Any Dialog From Menu (params: MENUPATH, TITLE) →

Open Search Dialog From Menu

Open New Project Dialog From Menu

▶ Wenn man schon zum 2. Mal eine Aktion braucht

→ Keyword erstellen

Keywords

19.05.2009

9



Keywords – was steht dahinter?

▶ In manchen Frameworks (e.g. FIT, Quality Center)

Programmcode, von Automation Experts geschrieben

▶ In anderen Tools (e.g. GUIDancer)

Fertige, beliebig kombinierbare Keywords mit Tool ausgeliefert

Keinen Programmieraufwand für das Testen

Keywords

19.05.2009

10



Einen Keyword-Test erstellen

- ▶ **Anhand von fertigen Keywords**

Parallel zu der Entwicklung

- ▶ **Login Dialog wird entwickelt**

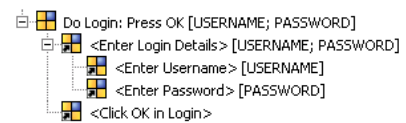
Keywords für Login Dialog →

Enter Username

Enter Password

Click OK in Login

Click Cancel in Login



Keywords 19.05.2009 11



Vorteile für Tests

- ▶ **Tests sind lesbar**

- ▶ **Zeitnahes Testen**

- ▶ **Wiederkehrende Aktionen als Keywords abbildbar**

Breite Wiederverwendung

Zentrale Veränderungen um Test aktuell zu halten

- ▶ **Modularität**

Module lassen sich gut austauschen

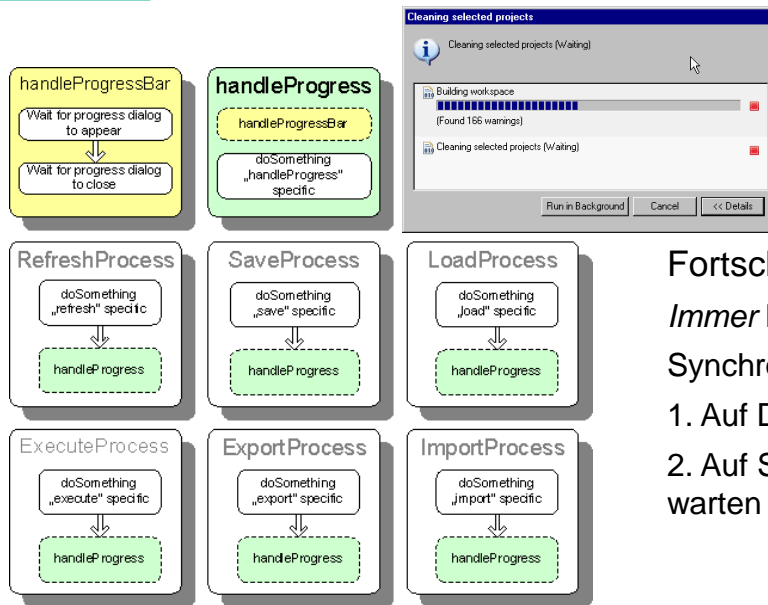
Neue Module (aus bestehenden) erstellen und hinzufügen

- ▶ **(Ohne Programmaufwand)**

Keywords 19.05.2009 12



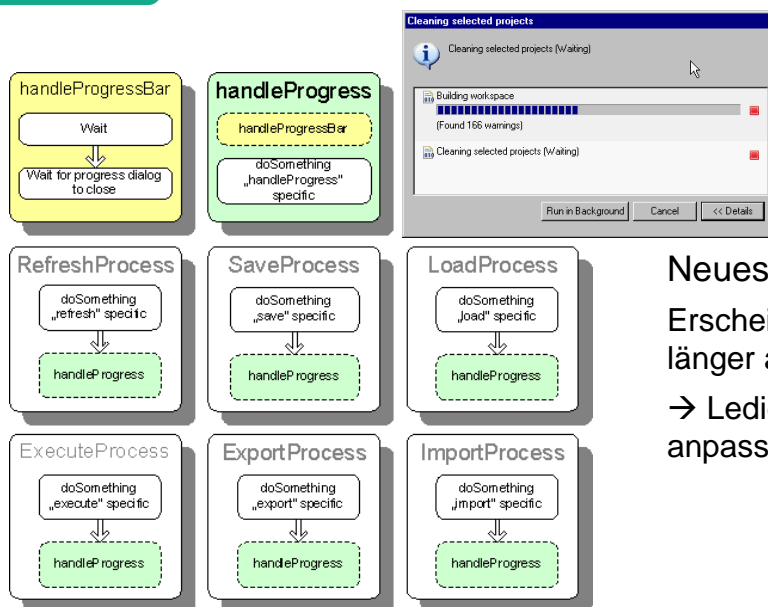
Beispiel



Fortschrittdialog:
 Immer bei längeren Aktionen
 Synchronisation im Test
 1. Auf Dialog warten
 2. Auf Schließen von Dialog warten



Beispiel



Neues Verhalten
 Erscheint nur bei Aktionen, die länger als 1 Sek. dauern
 → Lediglich handleProgressBar anpassen



Fazit

- ▶ **Agiles Funktionales Testen → Herausforderung**
- ▶ **Capture-Replay eignet sich nicht**
- ▶ **Keywords sind dem agilen Prozess gerecht**
- ▶ **Erlauben zeitnahe Automatisierung ohne große Wartungsaufwände**
- ▶ **→ jede Iteration gut getestet**

11.2. SW quality improvements by using quality KPIs

Dr. Hartwig Schwier
Corporate SW Quality, Test and Validation
Océ Printing Systems
Siemensallee 2, Postfach 1260
85581 Poing
Hartwig.Schwier@oce.com

Abstract

At the beginning of 2007 software quality key performance indicators (KPI) on product level were introduced to report the quality of the software products and to initiate software product quality improvement activities. Significant improvements over the period of about 12-24 months could be achieved. It was proven that quality indicators can be used for reporting as well as for initiating and controlling improvement activities. Following a PDCA cycle we are going to introduce modified metrics and KPIs to safeguard the achieved results and start another improvement cycle.

Introduction

Océ supplies digital printing systems, software and services for the production, reproduction, distribution and management of documents in small and wide format for professional users in offices, educational institutions, industry, advertising and the graphics arts market. To make the print workflow faster and more efficient R&D Application Software within Océ develops and maintains modular document workflow software. These products are commercial-off-the-shelf (COTS) software products and licensed to customers. The base products can be configured by options and additional components can be added to support integration with system and work processes already in place across the full range of printing environments. In 4 R&D sites worldwide approx. 220 engineers develop and test about 10 product lines with independent life cycles.

Motivation

At the beginning of 2007 software quality key performance indicators (KPI) on product level were introduced

- to report the quality of the software products
- to initiate software product quality improvement activities

In addition, all quality product indicators contribute to a global Q-indicator, which represents the overall product quality of R&D Application SW. The defined quality indicators reflect the actual "quality in use" of the software products from a customer point of view. Besides the number of defects and change requests, trends, priorities or backlogs of the open field reports are also considered.

Definition of quality indicators

Problem reports

As the quality indicators should reflect the actual "quality in use" of the software products from a customer perspective, the evaluation of the metrics is based on a corporate-wide database, which contains all customer and service records worldwide for released products. The central maintained database is accessible for service, QA and development engineers. Therefore, the usage of these data allows a high degree of transparency, objectivity and reliability. Each database record includes priorities, time stamps, descriptions, responsibilities and the actual status for the whole problem fixing process (from a customer to development action and back again to the customer)

Metrics

Despite the fact that the usage of metrics and KPIs is highly recommended, real practical examples for software KPIs and their application can seldom be found in literature. Therefore, we analyzed state-of-the-art metrics definitions (see references listed in the appendix), selected the most promising for our purposes and tailored them for the Océ R&D Application Software processes to cover all relevant quality aspects, e.g. number of problems, trends, severity of problems and time-to-solve (backlog). Product-specific parameters allow making the definitions applicable to the variety of all 10 products groups. The relative weight for each metric was derived systematically from experience and observations, e.g. starting from initial statements like „too many open problems“ or „response time to solve is too long“. The metrics are evaluated for each product separately. All metrics are normalized to a range of 0 to 100%. In Figure 11.1 the defined metrics, the description and the relative weight are summarized.

Metric	Weight	Description
TCP	9	Trend of Customer Problems measures the deviation from reference level. Defined as unilateral capability index and displayed as pseudo control chart.
ASCP	3	Average Severity of Customer Problems
Backlog	5	Backlog of open problems (response-time-to-solve)
PCI	9	Problems per Customer Installation
CRI	1	Change Requests per Customer Installations

Abbildung 11.1.: Defined metrics, their description and their relative weight

TCP

TCP (trend of customer problems) is defined as an unilateral capability indicator and measures the deviation of total number of problems from a product specific reference value.

$$TCP = \frac{\text{Number of Problems} - \mu}{3 \sigma}$$

μ : Level or average value
 σ : Deviation standard deviation
 μ and σ are defined for each product

11.2. SW quality improvements by using quality KPIs

ASCP

ASCP stands for „Average severity of customer problems“ and describes the problem severity per product.

$$\text{ASCP} = \sum_i \frac{g_i * NP_i}{g_{\max} * NP}$$

NP	Total number of problems
NP _i	Number of problems per severity
g _i	Weight per severity
g _{max}	Maximum weight
i	Metric identifier

Backlog

The backlog metrics is used to measure the measures the fraction of open problems, which exceed the specified response time per severity class.

$$\text{Backlog} = \sum_i \frac{g_i * NTP_i}{g_{\text{sum}} * NP_i}$$

NTP _i severity	Number of open problems per exceeded the response time criteria
NP _i	Number of problems per severity
g _i	Weight per severity
g _{sum}	Sum of all weights
i	Metric identifier

PCI and CRI

The metrics PCI and CRI measure the problems and change requests per customer installations respectively.

$$\text{PCI} = \frac{\text{Number of problems}}{\text{Number of installations}}$$

$$\text{CRI} = \frac{\text{Number of change requests}}{\text{Number of installations}}$$

Product q-indicator

For each product a specific q-indicator is calculated as weighted sum:

$$q = \frac{\sum_i w_{mi} * m_i}{\sum_i w_{mi}}$$

m_i **Metric**
 w_{mi} **Weight for metric**
 i **Metric identifier**

Q-KPI

The Q-KPI describes the overall quality level of all products.

$$Q = \frac{\sum_i w_i * q_i}{\sum_i w_i}$$

q_i **q-Indicator per product group**
 w_i **Weight for product group**
 i **Product group identifier**

Interpretation

The metrics and the quality indicator are interpreted in the same way and summarized in Figure 11.2.

Value ranges	Interpretation and color assignment
$0 \leq m, q, Q \leq 30$	Quality level is good
$30 < m, q, Q \leq 60$	Quality level is acceptable, but to be improved
$60 < m, q, Q \leq 100$	Quality level poor (to be corrected)

Abbildung 11.2.: Interpretation

Implementation

Evaluation

For the evaluation of the quality indicators a set of Excel templates and spreadsheets were developed. In the first steps the raw data are manually extracted, filtered and sorted. In these steps the raw data are checked regarding consistency and if needed consolidated.

11.2. SW quality improvements by using quality KPIs

The successive evaluation steps are automated, which allows a reliable evaluation of the indicators within half a day.

Communication and acceptance

The specification of the metrics and Q-indicators were published in December 2006, presented and discussed in several workshops during the first quarter of 2007 with all involved R&D sites and service organizations.

Acceptance

As engineers of the service organization confirmed that the introduced quality-indicators present the actual product quality quite well, confidence was built up. After the workshops the quality indicator were accepted as common quality measure pretty fast, as

- the complete evaluation process is based on a common database, simple and highly transparent
- the reporting is done regularly on a monthly basis and on corporate level and the R&D management supported of the metrics programs strongly.

Reporting

Q-KPI

The Q-KPI is reported monthly as red (poor), yellow (acceptable) and green (good), but also in more detail in a percent range from 0 (very good) to 100 (very poor). In Figure 11.3 the global Q-KPI (27.48, green) of December 2007 is shown as a pie-chart. The color of each pie displays the status of a product group and the size of each pie represents the impact on the global Q-indicator respectively.

Product q-indicator

Figure 11.4 shows the typical cover page of such a report, which summarizes all important findings per product in a radar diagram. The smaller the size of the area defined by the blue product line, the higher the quality of the product. The status of a product group is displayed also in the pie-chart; the size of the pie represents the impact on the overall Q-Indicator.

Observations and improvements

After the introduction at the beginning of 2007, a detailed analysis showed that the number of problems was too high, that there was a strong deviation from the reference values, that the average problem priority was too high and the problem fixing time too long. The following root causes were identified:

- Inconsistencies in the problem database and in the problem handling process, e.g. no systematic record handling or limited attention on record maintenance.
- In the planning of product releases, the attention was more directed on new functionality than on product quality improvements.

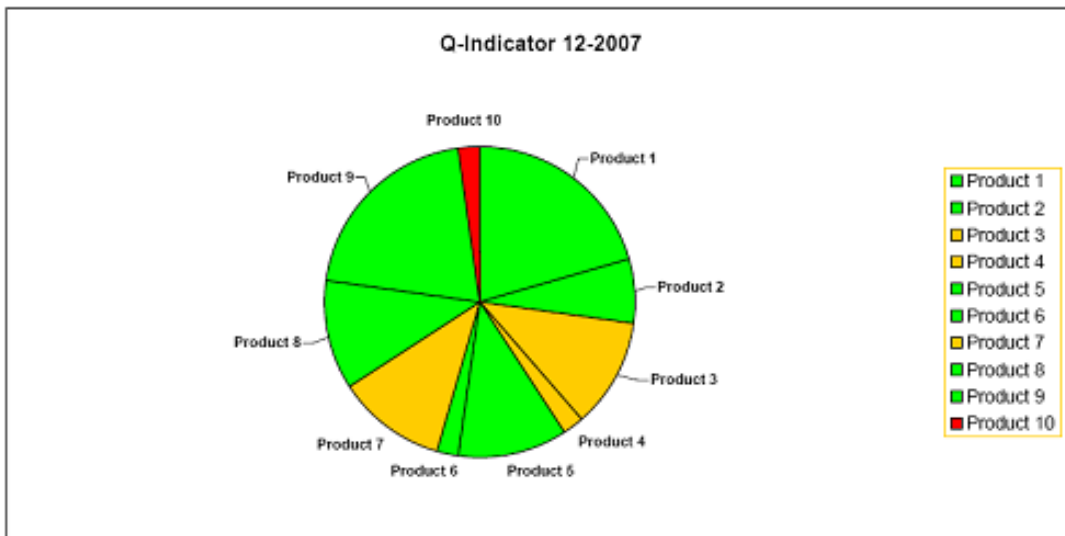


Abbildung 11.3.: Graphical presentation of the Global Q-Indicator (27.48, green) of December 2007 as a pie-char

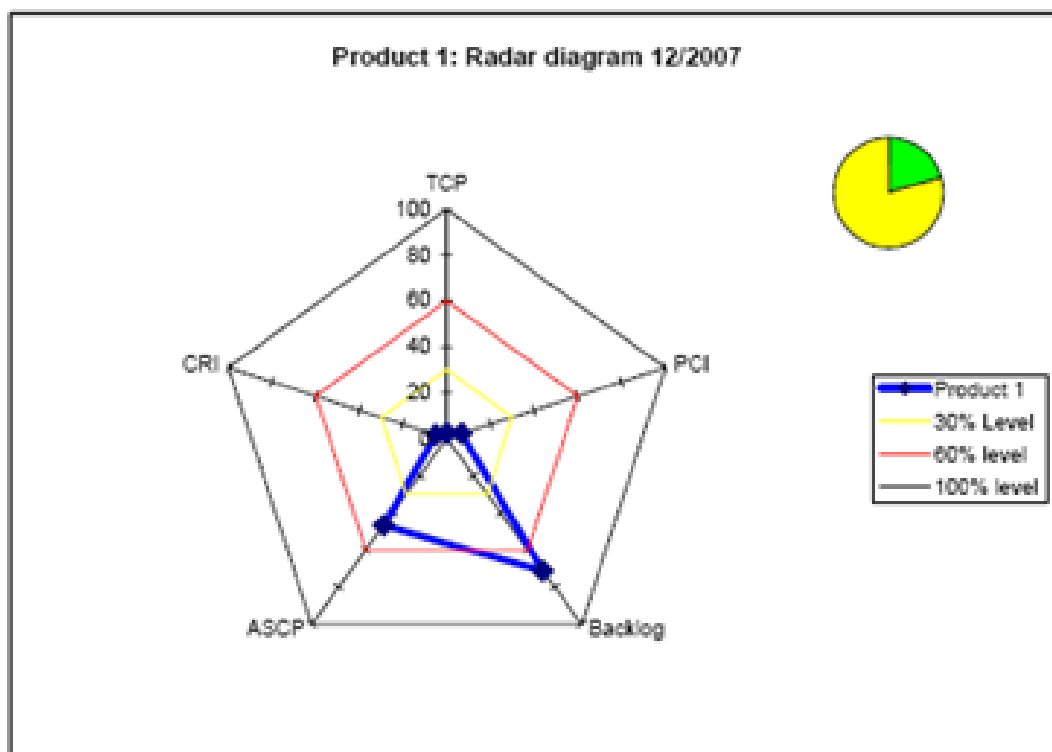


Abbildung 11.4.: Product quality indicator displayed as radar diagram

11.2. SW quality improvements by using quality KPIs

For each product improvement targets were defined, actions like

- Purge the database with the support of service and development.
- Initiate and define product quality improvements explicitly either as part of the regular functional enhancements or as separate releases.

were agree and the outcome controlled. These small PDCA cycles were repeated several times unless the original set targets were reached.

Results

Q-KPI

By December 2007, the overall Q-indicator had improved by about 40% from 45 (yellow) to 27 (close to green). Figure 11.3 shows the result of December 2007 as a pie-chart and Figure 11.5 displays the improvements achieved in 12 months.

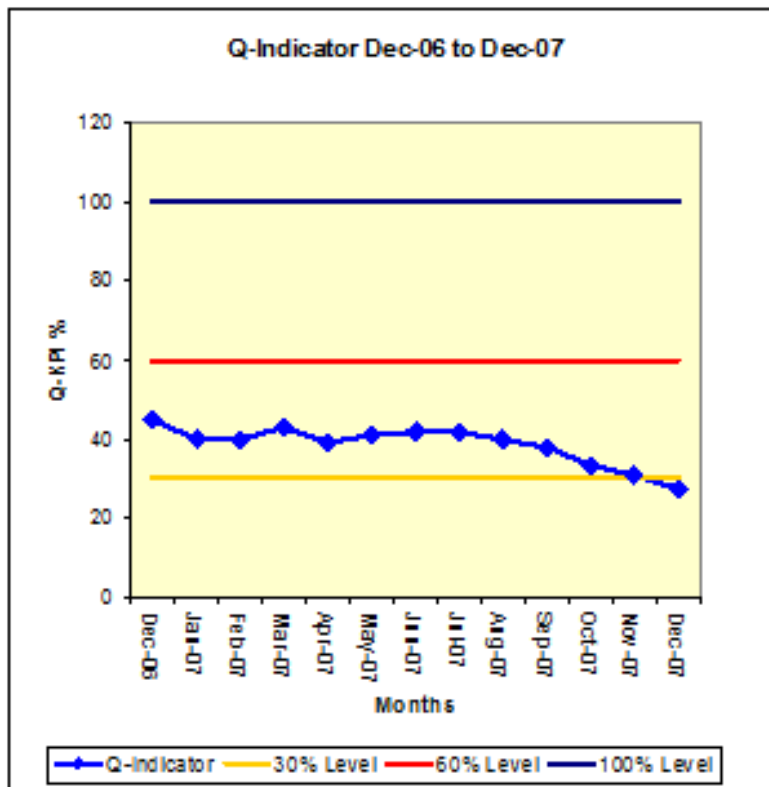


Abbildung 11.5.: Achieved improvements of the overall Q-Indicator in the year 2007

Product quality indicator

Figures 11.6 shows the significant improvements over the period of about 12 months for one product. The green polygon describing the quality in 2007 is almost completely included in the 30% level (yellow line). In addition, the area defined by the green polygon is only a fraction of the area defined by the red polygon (December 2007) indicating the degree of quality improvement.

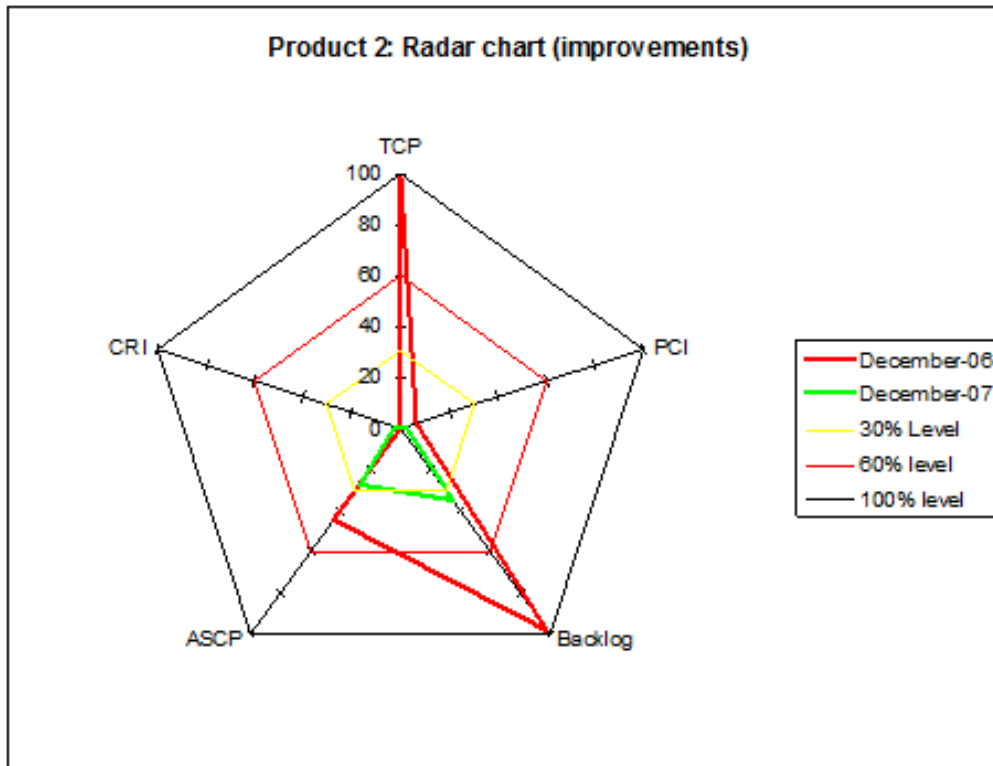


Abbildung 11.6.: Radar chart indicating the improvements

The reduction of number of problem records of about 66% could be achieved as a combined action of four (!) bug-fix releases and additional maintenance activities of purging the problem database (see Figure 11.7)

Besides this significant reduction in open problems the average priority of the customer problems was reduced to value close to priority 3. From a customer point of view, this means to our experience a high product quality in functionality, performance and usability.

The columns in Figure 11.8 represent the number of open problems per priority. Besides a significant reduction in open problems (as discussed before) the distribution of the data of December 2007 in green can obviously be fitted as a Gaussian distribution with a mean value close to priority 3. From a customer point of view, this means to our experience a high product quality in functionality, performance and usability.

Conclusion

During 2007 software quality indicators were introduced in a systematic way. The defined KPIs describe several aspects of product quality, e.g. the number of problems, the severity of problems and the problem handling. It was proven that quality indicators can be used for reporting as well as for initiating and controlling improvement activities.

To be successful a systematic approach, regular reporting and the support from all departments involved is needed. Following a PDCA cycle we are going to introduce modified metrics and KPIs to safeguard the achieved results and start another improvement cycle.

11.2. SW quality improvements by using quality KPIs

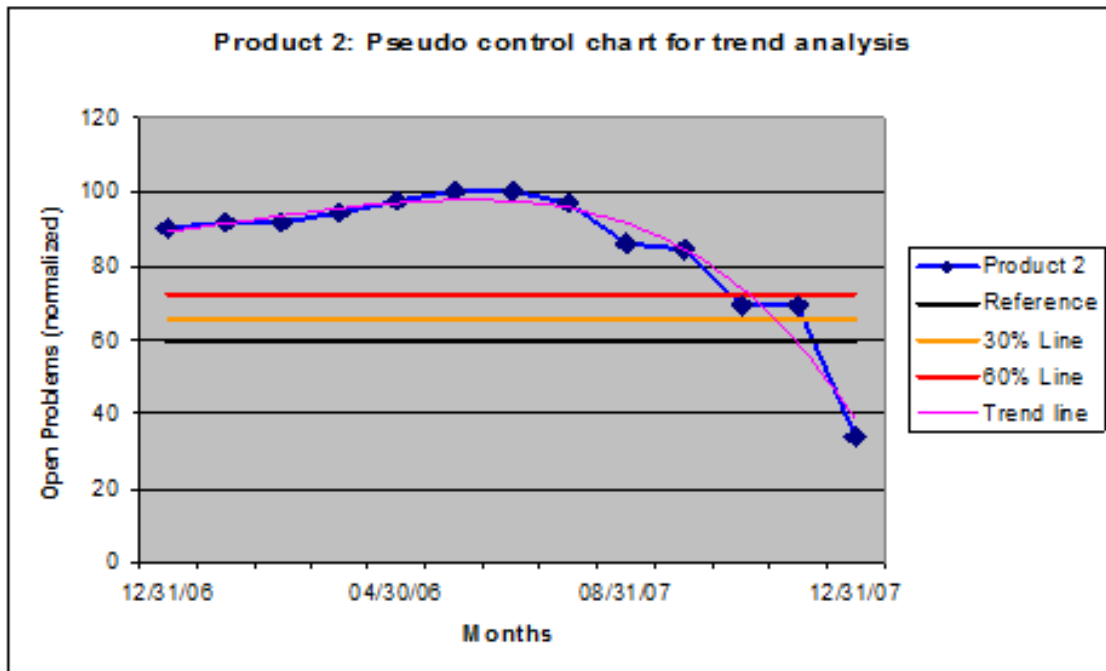


Abbildung 11.7.: Decrease in the number of problems for one major product

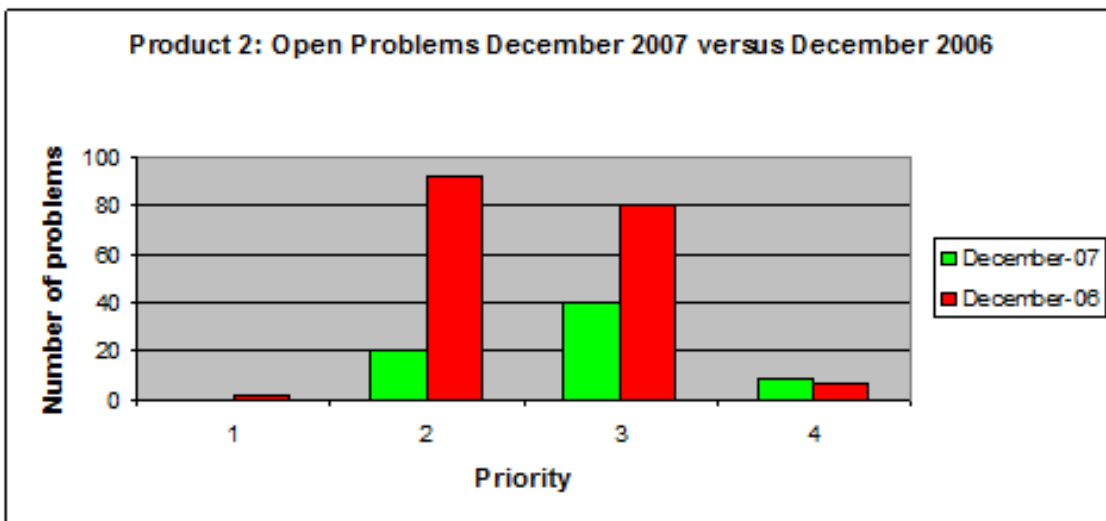


Abbildung 11.8.: Open Problems and average problem severity

Literature

The introduced metrics are based on state-of-the art definitions and adapted to the available data and defined processes at R&D Application Software.

The following sources were used:

1. Kan, H. Metrics and models in software quality engineering 2nd edition, Addison Wesley, 2003
2. Galin, Daniel Software Quality Assurance Addison Wesley, 2004
3. Craig, Rick D; Jaskiel, Stefan P. Systematic Software Testing Artech House, 2002

First results were published

1. H. Schwier, Testing experience 3-2008



SW quality improvements by using quality KPIs

Dr. Hartwig Schwier
Corporate SW Quality, Test and Validation
Océ Printing Systems
Siemensallee 2, Postfach 1260
85581 Poing
Hartwig.Schwier@oce.com

1

Introduction



Océ supplies digital printing systems, software and services for the production, reproduction, distribution and management of documents in small and wide format for professional users in offices, educational institutions, industry, advertising and the graphics arts market.

To make the print workflow faster and more efficient R&D Application Software within Océ develops and maintains modular document workflow software. These products are commercial-off-the-shelf (COTS) software products and licensed to customers. The base products can be configured by options and additional components can be added to support integration with system and work processes already in place across the full range of printing environments.

In 4 R&D sites worldwide approx. 220 engineers develop and test about 10 product lines with independent life cycles.

2

Objectives



Q-KPI (quality key performance indicators) on product level:

- to report the quality of the software products (quality in use)
- to initiate software product quality improvement activities

The Q-KPIs are based on external defects and change requests and should cover

- number
- trends
- priorities
- backlogs

All quality product indicators contribute to a global Q-KPI, which represents the overall product quality of R&D Application SW.

3

Data source



Data source

- Corporate-wide database for all customer and service records worldwide.
- Each database record includes priorities, time stamps, descriptions, responsibilities and the actual status

4

Overview: Metrics



Metric	Weight	Description
TCP	9	<u>T</u> rend of <u>C</u> ustomer <u>P</u> roblems measures the deviation from reference level. Defined as unilateral capability index and displayed as pseudo control chart.
ASCP	3	<u>A</u> verage <u>S</u> everity of <u>C</u> ustomer <u>P</u> roblems
Backlog	5	<u>B</u> acklog of open problems (response-time-to-solve)
PCI	9	<u>P</u> roblems per <u>C</u> ustomer <u>I</u> nstallation
CRI	1	<u>C</u> hange <u>R</u> equests per Customer <u>I</u> nstallations

The metrics are evaluated for each product separately. All metrics are normalized to a range of 0 to 100%.

5

Metric: TCP



TCP (trend of customer problems) is defined as an unilateral capability indicator and measures the deviation of total number of problems from a product specific reference value.

$$TCP = \frac{\text{Number of Problems} - \mu}{3 \sigma}$$

μ : Level or average value

σ : Deviation standard deviation

μ and σ are defined for each product

6

Metric: ASCP



ASCP stands for “Average Severity of Customer Problems” and describes the problem severity per product.

$$\text{ASCP} = \sum_i \frac{g_i * \text{NP}_i}{g_{\text{max}} * \text{NP}}$$

NP	Total number of open problems
NP _i	Number of problems per severity
g _i	Weight per severity
g _{max}	Maximum weight
i	Metric identifier (1,...4)

7

Metric: Backlog



The backlog metrics is used to measure the measures the fraction of open problems, which exceed the specified response time per severity class.

$$\text{Backlog} = \sum_i \frac{g_i * \text{NTP}_i}{g_{\text{sum}} * \text{NP}_i}$$

NTP _i	Number of open problems per severity exceeded the response time criteria
NP _i	Number of problems per severity
g _i	Weight per severity
g _{sum}	Sum of all weights
i	Metric identifier (1,...4)

8

Metrics: PCI and CRI



The metrics PCI and CRI measure the problems and change requests per customer installations respectively.

$$\text{PCI} = \frac{\text{Number of Problems}}{\text{Number of Installations}}$$

$$\text{CRI} = \frac{\text{Number of Changes Requests}}{\text{Number of Installations}}$$

9

Product q-indicator



Product q-indicator

For each product a specific q-indicator is calculated as weighted sum:

$$q = \frac{\sum_i w_{mi} * m_i}{\sum_i w_{mi}}$$

m_i Metric
 w_{mi} Weight for metric
 i Metric identifier

10

Q-KPI



Q-KPI

The Q-KPI describes the overall quality level of all products:

$$Q = \frac{\sum_i w_i * q_i}{\sum_i w_i}$$

q_i q-Indicator per product group
 w_i Weight for product group
 i Product group identifier

11

Interpretation



Interpretation

The metrics and the quality indicator are interpreted in the same way.

Value ranges	Interpretation and color assignment
$0 \leq m, q, Q \leq 30$	Quality level is good
$30 < m, q, Q \leq 60$	Quality level is acceptable, but to be improved
$60 < m, q, Q \leq 100$	Quality level poor (to be corrected)

12

Communication and acceptance



Communication

The specification of the metrics and Q-indicators were published in December 2006, presented and discussed in several workshops during the first quarter of 2007 with all involved R&D sites and service organizations.

Acceptance

As engineers of the service organization confirmed that the introduced quality-indicators present the actual product quality quite well, confidence was built up. After the workshops the quality indicator were accepted as common quality measure pretty fast, as

- the complete evaluation process is based on a common database, simple and highly transparent
- the reporting is done regularly on a monthly basis and on corporate level and the R&D management supported of the metrics programs strongly.

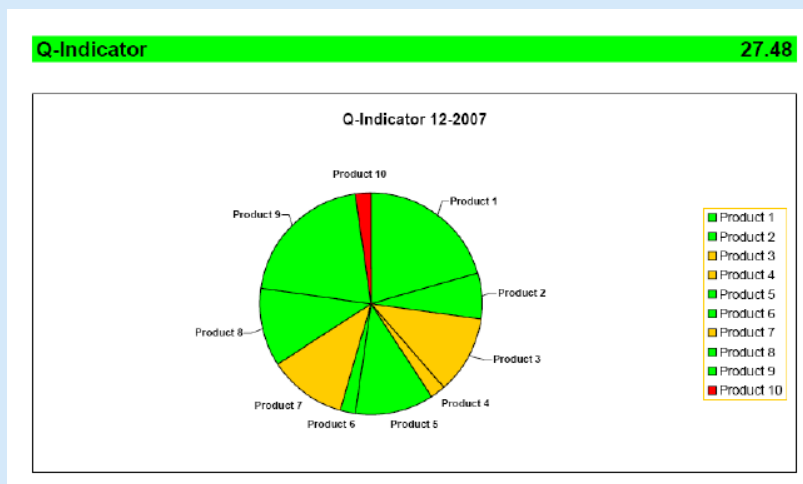
13

Reporting: Q-KPI



Q-KPI

The Q-KPI is reported monthly as red (poor), yellow (acceptable) and green (good), but also in more detail in a percent range from 0 (very good) to 100 (very poor).



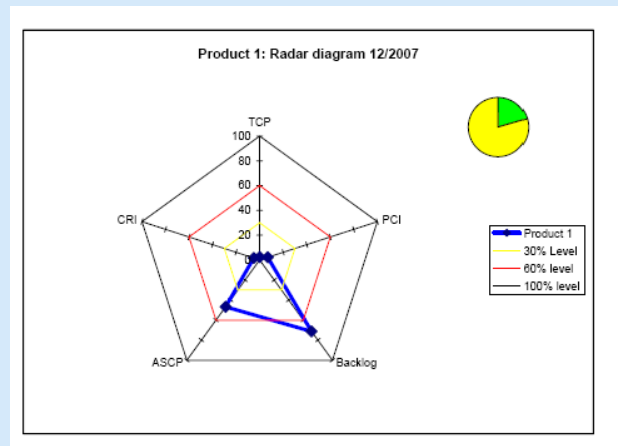
14

Reporting: Product quality indicator



Product quality indicator

The product quality indicator displayed as radar diagram



15

Observations and improvements



After the introduction at the beginning of 2007, a detailed analysis showed that the number of problems was too high, that

- there was a strong deviation from the reference values
- the average problem priority was too high
- the problem fixing time too long.

The following root causes were identified:

- Inconsistencies in the problem database and in the problem handling process,
- In the planning of product releases, the attention was more directed on new functionality than on product quality improvements.

For each product improvement targets were defined:

- Purge the database with the support of service and development.
- Initiate and define product quality improvements explicitly

The actions were agree and the outcome controlled.

These small PDCA cycles were repeated several times unless the original set targets were reached.

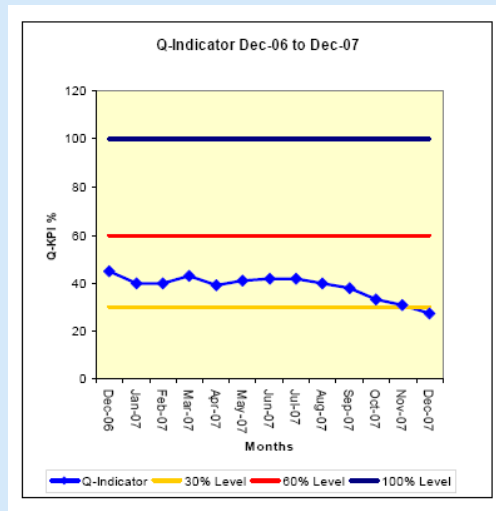
16

Results: Q-KPI



Q-KPI

By December 2007, the overall Q-indicator had improved by about 40% from 45 (yellow) to 27 (close to green)



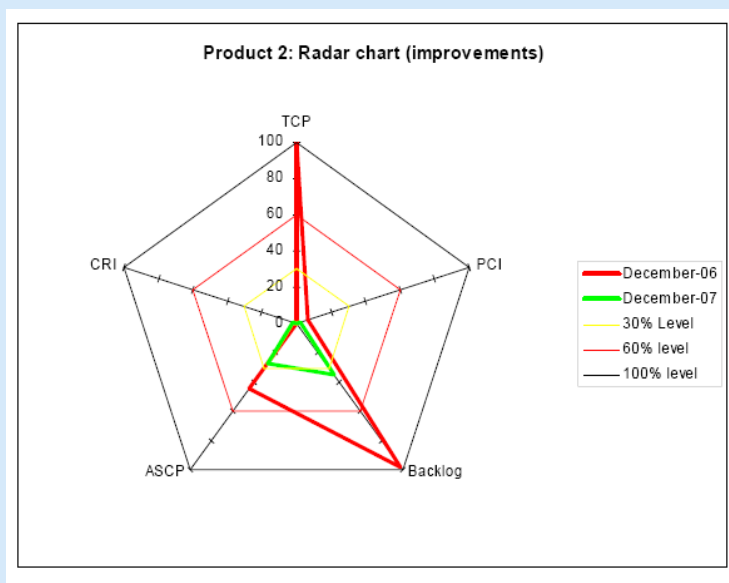
17

Results: Product quality indicator



Product quality indicator

The green polygon describing the quality in 2007 is almost completely included in the 30% level (yellow line).



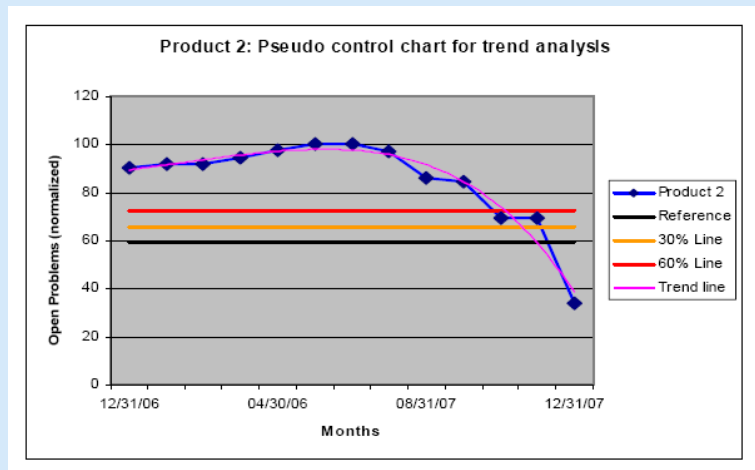
18

Results: Product quality indicator



Product quality indicator – Number of defects

The reduction of number of problem records of about 66% could be achieved as a combined action of four (!) bug-fix releases and additional maintenance activities of purging the problem database.



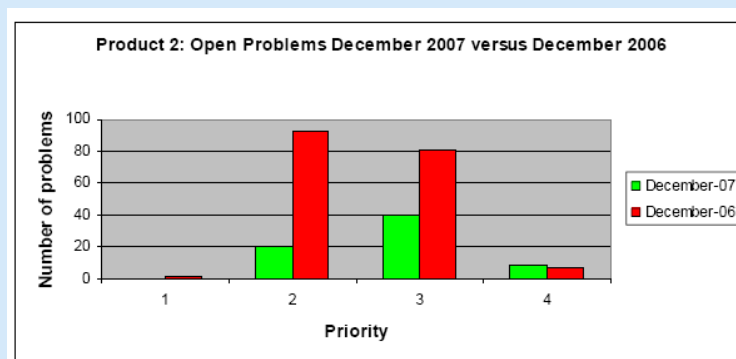
19

Results: Product quality indicator



Product quality indicator – ASCP

Besides this significant reduction in open problems the average priority of the customer problems was reduced to value close to priority 3. From a customer point of view, this means to our experience a high product quality in functionality, performance and usability.



20

Conclusion



During 2007 software quality indicators were introduced in a systematic way. The defined KPIs describe several aspects of product quality, e.g. the number of problems, the severity of problems and the problem handling. It was proven that quality indicators can be used for reporting as well as for initiating and controlling improvement activities.

To be successful a systematic approach, regular reporting and the support from all departments involved is needed.

Following a PDCA cycle we are going to introduce modified met.

21

Literature



The introduced metrics are based on state-of-the art definitions and adapted to the available data and defined processes at R&D Application Software.

The following sources were used:

Kan, H.
Metrics and models in software quality engineering
2nd edition, Addison Wesley, 2003

Galin, Daniel
Software Quality Assurance
Addison Wesley, 2004

Craig, Rick D; Jaskiel, Stefan P.
Systematic Software Testing
Artech House, 2002

First results were published:

H. Schwier, Testing experience 3-2008

22

11.3. Effort Reduction and Error Prevention during Software Development and Maintenance

Gürcan Gülesir, Lodewijk Bergmans
BloomWise
Zilverling Building, EWI/SE
University of Twente
7500 AE, Enschede, The Netherlands
gulesir@bloomwise.com
{gulesirg | bergmans}@ewi.utwente.nl

Abstract

The development and maintenance of today's software systems is an increasingly effort-consuming and error-prone task. A major cause of the effort and errors is the gap between the design and implementation. In practice, software design is often informally documented, or not documented at all. Therefore, (a) the design cannot be properly communicated between software engineers, (b) it cannot be automatically analyzed for finding and removing faults, (c) the conformance of an implementation to the design cannot be automatically verified, and (d) the design documents cannot be kept up-to-date with the evolving source code. As a result, the development of proper design documentation is often considered a large burden with very little pay-off, especially if -it is expected that- the design documentation is not kept up-to-date with the actual software.

In this talk, we address these problems for organizations that develop control-intensive software systems. We present the following solutions, each addressing the respective problem stated above: (a) a graphical language called Vibes, which enables engineers to express requirements on the behavior implemented in source code, (b) a software tool called CheckSpec, which automatically verifies the consistency between multiple specifications written in Vibes, (c) a software tool called CheckSource, which automatically verifies whether a given implementation fulfills the requirements written in Vibes, and (d) a software tool called CreateVisuaL, which semi-automatically generates Vibes specifications from existing (legacy) source code. As a result, writing the Vibes specifications once during design, has multiple pay-offs during product development and during maintenance (since now it can be automatically ensured that design documentation and code remain consistent over the life-time of the product).

Controlled experiments at ASML (www.asml.com) indicate that CheckSource is beneficial during some of the typical control-flow maintenance tasks: 75% effort reduction, and prevention of one error per 140 lines of source code.

The key innovation of this solution is the graphical language Vibes with its patent-pending feature: context-sensitive wildcards [1]. This feature makes Vibes specifications more evolvable (i.e. less susceptible to changes), and more concise.

In this talk, we will explain how the solution ensures that (a) the source code fulfills the requirements (i.e. prevents bugs), (b) the design documents are kept up-to-date with the evolving source code, and (c) common maintenance tasks require substantially less effort.

Literaturverzeichnis

1. Gülesir, G.; *Evolvable Behavior Specifications Using Context-Sensitive Wildcards*. Ph.D. Thesis. University of Twente. 2008

Effort Reduction and Defect Prevention during Software Development and Maintenance

 *Gürcan Güleşir*
Lodewijk Bergmans
BloomWise
gulesir@bloomwise.com

Software Engineering, EEMCS, University of Twente
{gulesirg, bergmans}@ewi.utwente.nl

Objectives

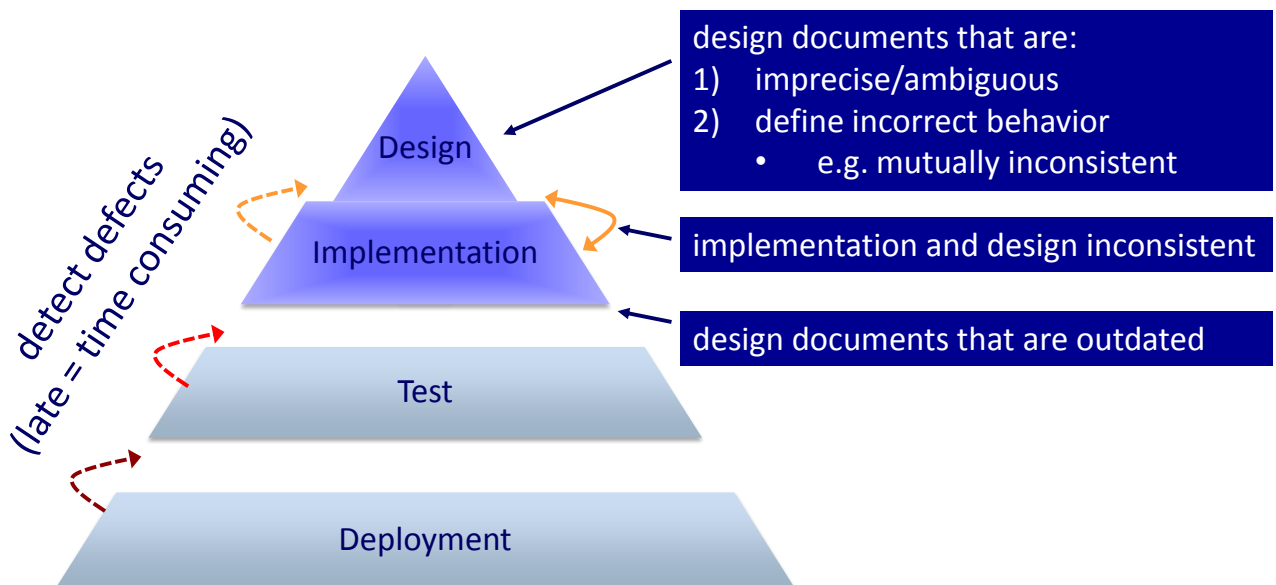


- goals:
 - design documentation that is precise & concise & provides insights
 - detect defects during design, implementation & maintenance
 - keep (architecture) design and implementation consistent
 - keep design documentation up-to-date
- means:
 - a simple visual notation for behavior design
 - incremental, agile way of working
 - complements existing methods, tools & processes
 - tool support, e.g. for automated checks
 - Example: experiments have shown substantial error & effort reduction

Problems We Address



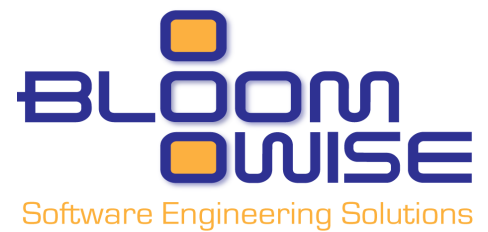
Common Problems in software development



Our Scope



- We focus on design and documentation of behavior
 - Especially control flow



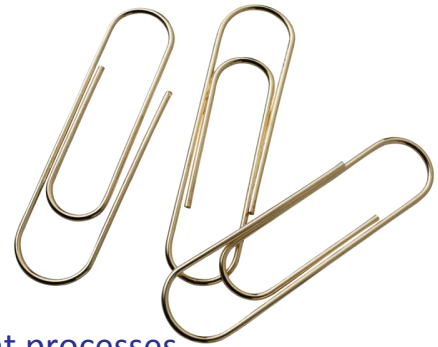
Our Solution



Solution



- Simple, graphical, notation: 'Vibes'
 - 'Visual behavior specification'
 - subset of UML (UML profile)
- Way of working (WoW)
 - Light-weight (agile)
 - Incremental
 - Easy to integrate into existing development processes
- Software tools to use Vibes and support WoW
 - Reduce effort
 - Prevent defects

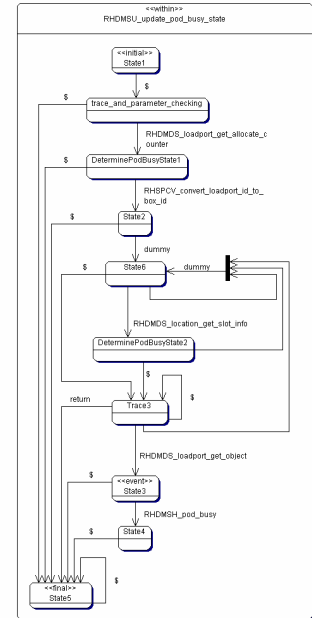
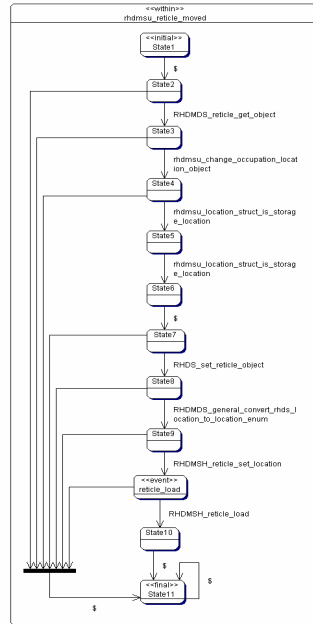
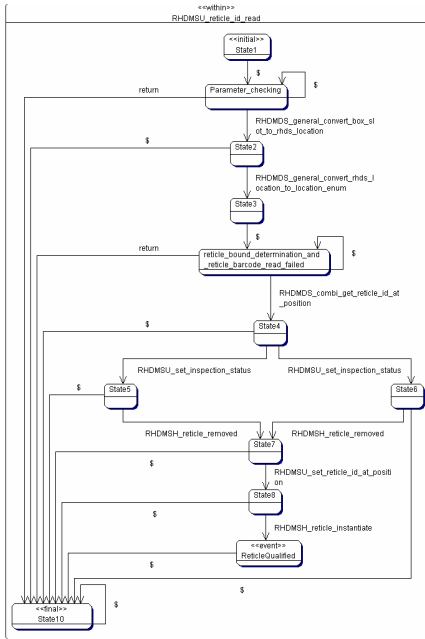


Vibes Notation

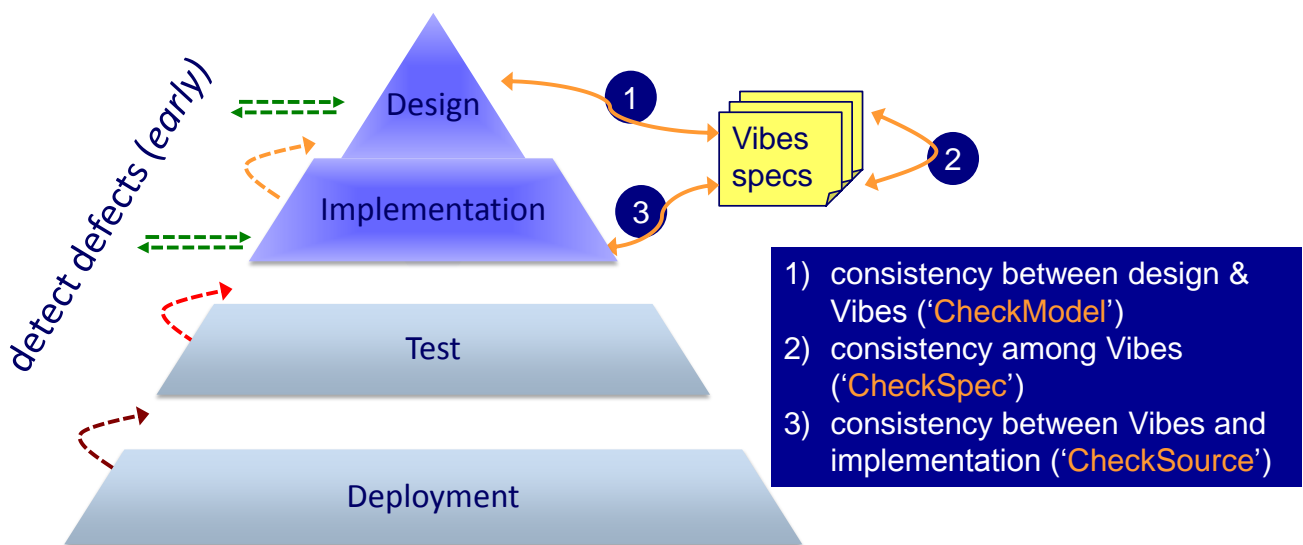


- Approach: minimal specification of behavioral design constraints
 - Avoid over-specification
 - Generic, independent of a specific implementation (details)
 - Support system evolution & product lines
- Allows for expressing design constraints on control flow behavior:
 - logical constraints
 - temporal constraints
 - repetition
- Unique *context-sensitive wildcard* ("\$") facilitates abstraction and evolvability by hiding details instead of omitting
 - patent pending

Realistic Examples of Vibes Specifications



Development process with Vibes approach



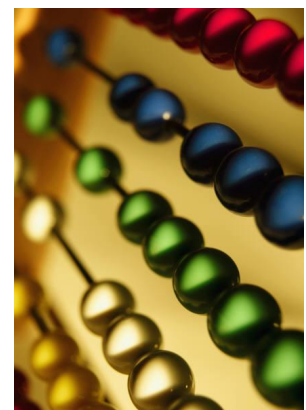
Evaluation of the Solution



Evaluation (experiences at ASML/Ideals)



- How steep is the learning curve?
 - Training for reading Vibes specifications and using CheckSource: 15 minutes
 - Training for creating Vibes specifications: 1 hour
- How much effort does it save?
 - In experiments with 24 ASML developers, CheckSource saved 75% of the effort
 - For control flow-related debugging tasks
- How many defects does it prevent?
 - In experiments with 24 ASML developers, CheckSource prevented 7 defects per 1000 LoC
 - For control flow-related debugging tasks



The Business Case for Vibes



- Warning: “Lies, damn lies, statistics!”
 - Feel free to fill in your own estimations or measurements
- On average **37%** of developer time spent on debugging [IDC]
- Assume **50%-75%** of debugging tasks is control-flow related
- Experience at ASML:
 - We *measured* **50%-75%** effort reduction in debugging effort
 - This implies **9-21%** reduction of software development effort
 - Assume cost of creating Vibes documentation is **2-5%** of total sw. dev. effort
- This implies a net overall effort reduction of **4-19%**
 - or: achieve less defects while even spending less time!

Related technologies



- Model-Driven Engineering
 - we believe for most systems, still a lot of behavior needs to be specified
- Model-checking approaches
 - can give more precise results, but this **requires** precise, **complete** input specifications
 - when adding details in implementation, preciseness is lost
 - **scalability** is often an issue (esp. software model checking)
- Static code analysis tools
 - check your software with little effort for low-level problems
 - but do not know *anything* about the system being analyzed
- Testing techniques
 - much less tests needed, but still needed to test for certain data values
 - more costly, cannot show the absence of defects.

All are **complementary** to the Vibes approach

Conclusion



Conclusion



- The Vibes approach reduces the gap between design and implementation
 - Design-time automated consistency checks (*CheckSpec & CheckModel*)
 - Automated conformance checking between design and source code (*CheckSource*)
 - The \$ supports evolvable design documents and abstraction
- Supports light-weight & incremental way of working
 - Can enforce up-to-date design documentation
- *Measured* benefits of Vibes and CheckSource
 - For control-flow related debugging tasks
 - **75% effort reduction and 7 defects per 1000 LoC avoided**

Opinions, Suggestions, Questions?



David Parnas: *[our interpretation]*

“My engineering teachers laid down some basic rules:

- Design before implementing. *[use your brain]*
- Document your design. *[use Vibes]*
- Review and analyze the documented design. *[use CheckDesign]*
- Review implementation for consistency with the design. *[use CheckSource]*

These rules apply to software at least as much as they do to circuits or machines.”

Backup slides



Vibes Use Cases (1/2)

How does it fit in the development process?



1. During detailed design
 - Document the control flow constraints of functions
2. To verify the implementation
 - Ensure that it corresponds to the design documents
3. When adding a new feature
 - Easier to see how this interacts with the design
4. To detect bugs during source code evolution
 - Ensure that updated code does not break the design

Vibes Use cases (2/2)

How does it fit in the development process?



5. To distinguish bugs from features
 - Explicitly describe control flow that is not allowed.
6. To improve the efficiency and effectiveness of source code inspections
 - Through better design documentation
7. To manage software complexity through compositional reasoning
 - Complex designs can be modularly specified, composed and verified

An Alternative Business Case for Vibes



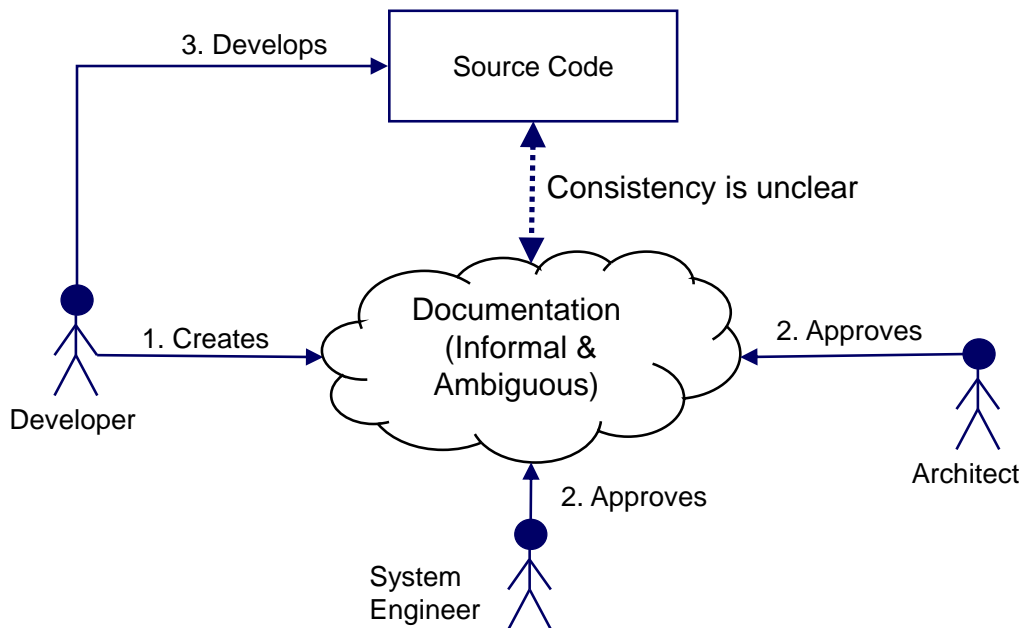
- Assumptions:
 - Software maintenance takes 80-90% of total Sw.Dev. Costs
 - 50% of maintenance is spent on code understanding.
 - 50-75% effort reduction when using Vibes in debugging
 - Estimated part of debugging that involves code understanding is 80-90%
 - Estimated cost of creating & updating diagrams is 2-5% of sw.dev. costs
- Conclusion: Vibes may reduce the net sw. dev. effort by 3-21%

Other (future) applications



- Reverse engineer Vibes from source code
- Use Vibes diagrams to 'weave in' code in the right places
 - E.g. state update information
- Generate skeleton code from the design
- Various alternatives of 'deep analysis'
- Extend Vibes notation to explicitly include conditions on transitions
- Apply to object-oriented languages
- Automatically minimize the size and complexity of Vibes specifications
- ...

Current Software Development Process

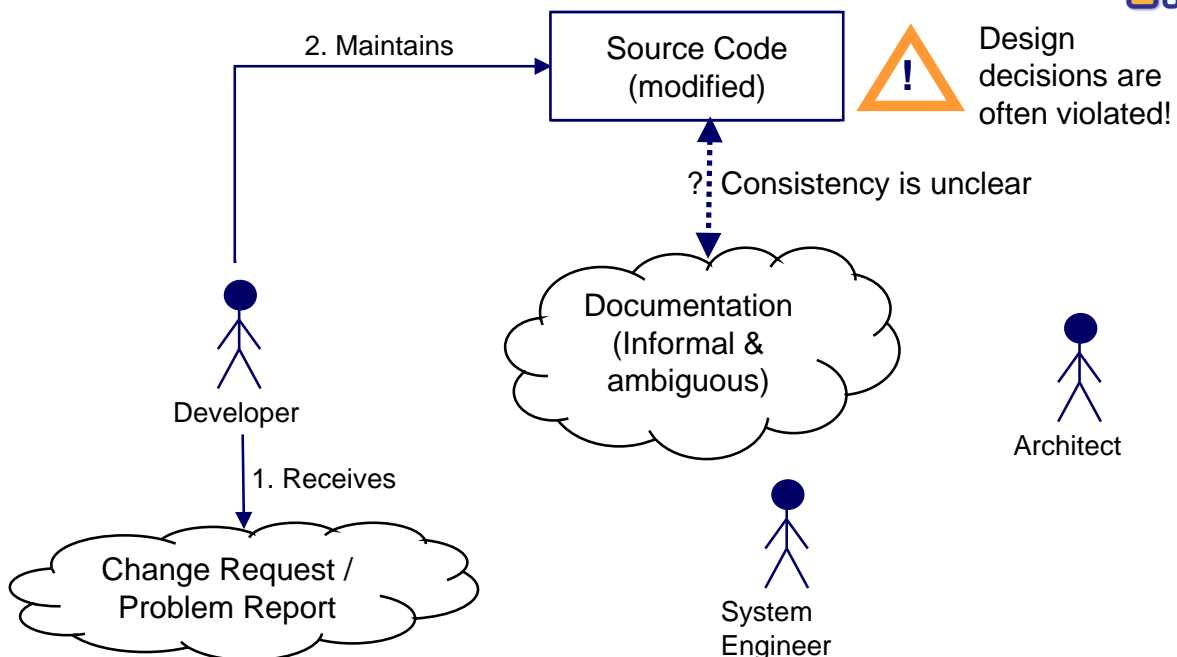


Problems



1. The developer, architect, and system engineer may (mis)interpret the design in different ways
 2. The software design can be internally inconsistent
 - And this cannot be systematically verified
 3. The implementation can be inconsistent with the design
 - And this cannot be systematically verified
- Consequences
 - Excessive development effort
 - Large number of defects

Current Software Maintenance Process

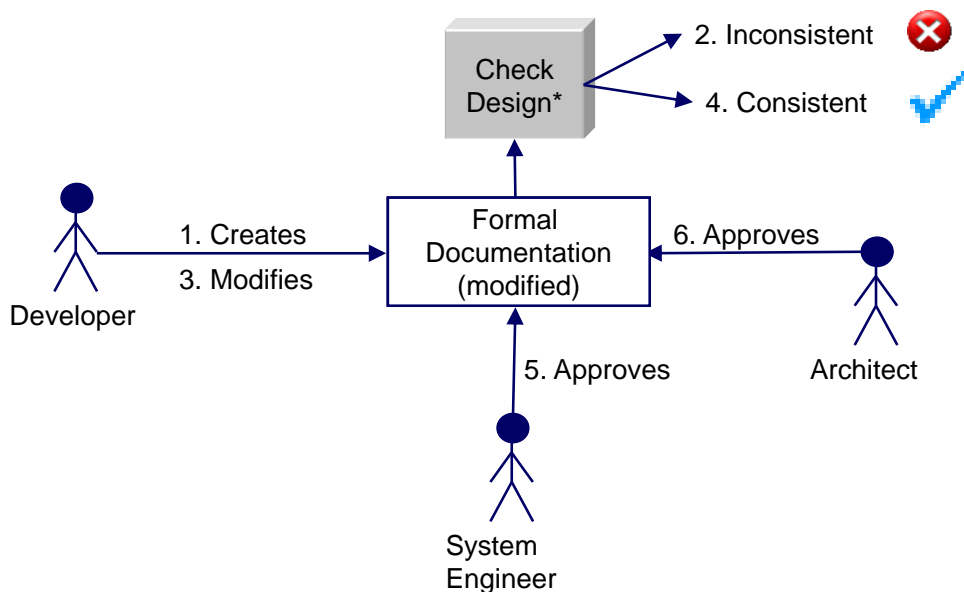


Problems (continued)



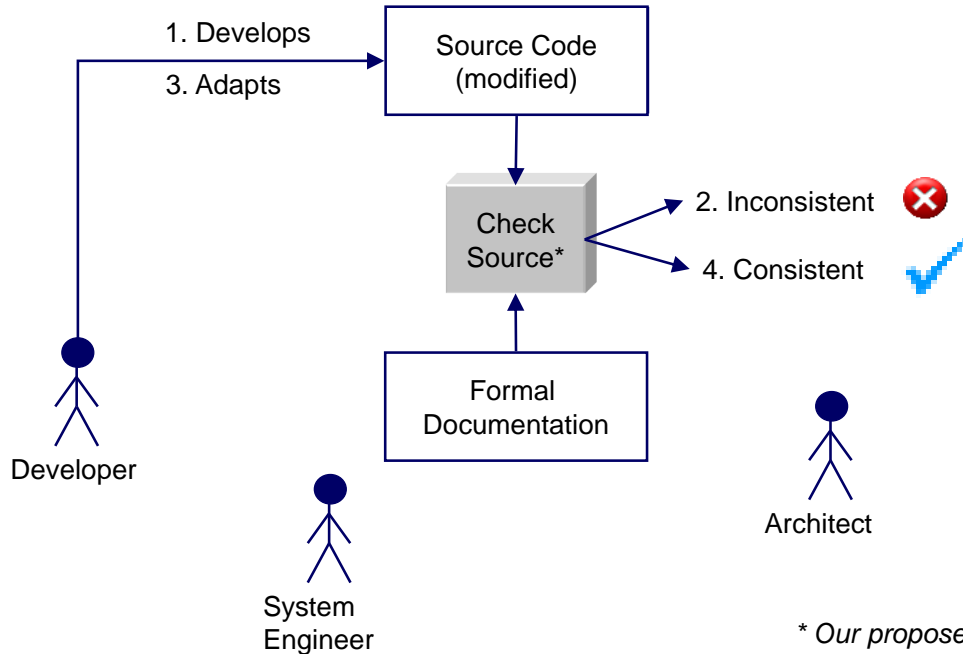
4. Documentation is outdated
 - Documentation becomes useless
 - The ROI of the documentation effort is very low
 5. System engineers and architects are no longer -systematically- involved
 6. Modified source code often violates design decisions
-
- Consequences
 - Excessive maintenance effort
 - Large number of defects

Proposed Software Design Process

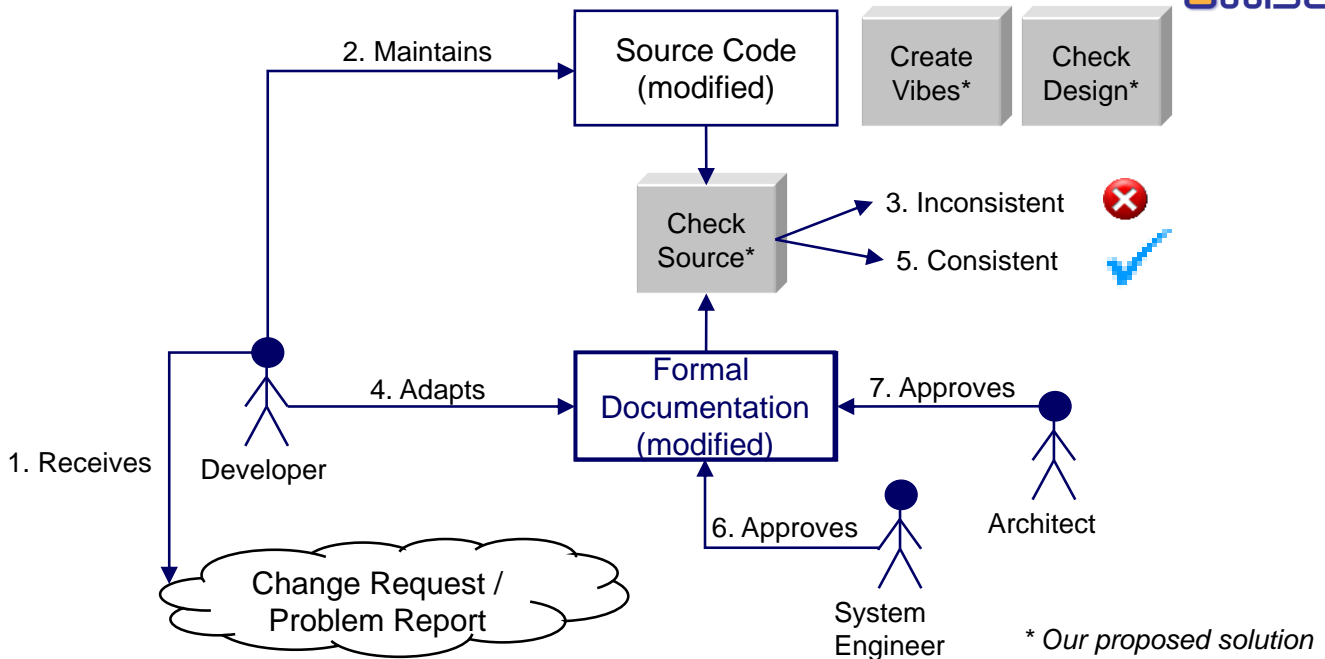


* Our proposed solution

Proposed Software Implementation Process



Proposed Software Maintenance Process



Related/alternative technologies



- **Model-Driven Engineering**
 - generation is mostly useful if specific domain knowledge is involved (e.g. platform, state machine, ..)
 - still most functionality needs to be *programmed*
 - → our approach is complementary:
 - for less defects in behavior models during design
 - whenever behavior is designed and implemented
- **Model-checking**
 - gives more precise results, but requires precise (complete) input
 - when adding details in implementation, preciseness is lost
 - scalability is often an issue (esp. software model checking)
- **Testing techniques**
 - can test effect of certain data values
 - effort consuming (but worth it)
 - but can never be exhaustive (only show defects)

12. Session 12: Werkzeuge zur Kollaboration in Projekten

Inhalt

12.1. Generierung von Prozess Templates für IBM Jazz auf Basis des V-Modell XT	486
12.2. Unibase - a unified CASE Tool	499
12.3. Anforderungen nachweisbar erfüllen	508

12.1. Generierung von Prozess Templates für IBM Jazz auf Basis des V-Modell XT

Angelo Gülle

angelo.guelle@bearingpoint.com

Abstract

Ein weiterer Baustein um formale Vorgehensmodelle werkzeuggestützt in die täglichen Arbeit zu integrieren. Kernthema ist die Erstellung eines *Open-Source Prototypen*, der Tailoringergebnisse des V-Modell XT Projektassistenten in Prozess-Templates für IBM-Jazz transformiert. Es werden die Werkzeuge Rational Team Concert mit IBM-Jazz, sowie der Rational Method Composer vorgestellt.

Der Vortrag fasst die Ergebnisse einer Diplomarbeit vom 03.11. - 03.02.2008 der FHTW-Berlin und Bearingpoint GmbH zusammen.

Motivation

Das V-Modell XT ist ein generisches Vorgehensmodell. Neben einer schriftlichen Dokumentation beinhaltet es Produktvorlagen, sowie die Referenzimplementierung zweier Werkzeuge zur Anpassung(Tailoring) des V-Modells.

IBM Jazz ist eine Entwicklungsplattform zur kollaborativen Arbeit aller Projektbeteiligten über den gesamten Lebenszyklus eines Softwareprodukts. Die Phasen des Softwarelebenszyklusses werden durch einzelne Applikationen abgedeckt, die sich gemeinsam über die Jazz Plattform integrieren.

Vorgehensmodelle laufen Gefahr nicht oder nur unzureichend angewendet zu werden, wenn die Werkzeuge zur Bearbeitung von Aufgaben innerhalb eines Projekts diese Vorgehensmodelle nicht unterstützen bzw. erzwingen. Dieses Projekt soll exemplarisch die Lücke zwischen dem V-Modell XT als Vorgehensmodell und IBM Jazz als Werkzeug schließen.

Es gilt die Erstellung eines Prototyps zur Transformierung eines getailorten V-Modell XT Projekts in Process Templates, die durch die Entwicklungsplattform IBM Jazz verarbeitet werden können. Dabei steht nicht allein die technische Machbarkeit im Vordergrund, sondern auch die theoretischen Überlegungen zur inhaltlichen Transformierung zwischen den beiden META-Modellen.

Ansatz

Zum Zeitpunkt der Arbeit existierte nur eine produktreife IBM Jazz-Applikation zur Entwicklung und Wartung von Software, das Rational Team Conert(RTC) 1.0. Inzwischen sind weitere Tools für Anforderungsanalyse, Test und Qualitätssicherung verfügbar. IBM bietet mit dem Rational Method Composer(RMC) ein Werkzeug zum Tailoring des Rational Unified Process(RUP) an, dass über Methoden-Plugins weitere Vorgehensmodelle importieren kann. Ebenso existiert eine experimentelle Brücke zwischen dem RMC und dem RTC, so dass sich Prozesse des RUP direkt als Process-Templates für IBM Jazz exportieren lassen. Diese Möglichkeiten werden genutzt, indem aus dem Tailoringergebnis des V-Modell XT Projektassistenten 1.3 ein Methoden-Plugin für den RMC generiert wird.

Bewertung

Durch die Übernahme des V-Modells XT in den RMC, ist ein wesentlich umfangreicheres Werkzeug zum Tailoring entstanden. Es lassen sich auf diesem Wege sehr feingranulare Tailorings erstellen und das V-Modell mit anderen Vorgehensmodellen kombinieren. Die RMC-RTC-Brücke ist in ihrem Umfang noch sehr beschränkt, integriert jedoch eine sehr brauchbare, interaktive Dokumentation in IBM-Jazz. Über diese lassen sich auch die zugehörigen Beispielprojekte und Produktvorlagen des getailorten Projekts abrufen. Es ist zu hoffen, dass es zukünftig auch für weitere IBM-Jazz Applikationen Brücken aus dem RMC geben wird, so dass der gesamte Lebenszyklus, und damit das gesamte V-Modell XT, in IBM-Jazz abgebildet werden kann.



Guten Tag!



Generierung von Prozess Templates für
IBM-Jazz auf Basis des V-Modell XT

Angelo Gülle

Diplomarbeit - Angewandte Informatik

Betreuer:
Prof. Dr. Hermann Heßling
Dipl. -Inf. Thorsten Hoffmann



Fachhochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

IBM - Jazz

© 02/2009 Angelo Gülle

3

- Kollaborationsplattform für Softwareentwicklung
- „Andocken“ von Tools zur Unterstützung des Softwarelebenszyklus:
 - RRC: Rational Requirements Composer BETA (Analyse)
 - RTC: Rational Team Concert 1.0 (Entwicklung, Wartung)
 - RQM: Rational Quality Manager BETA (Test)
 - Third Party Tools...

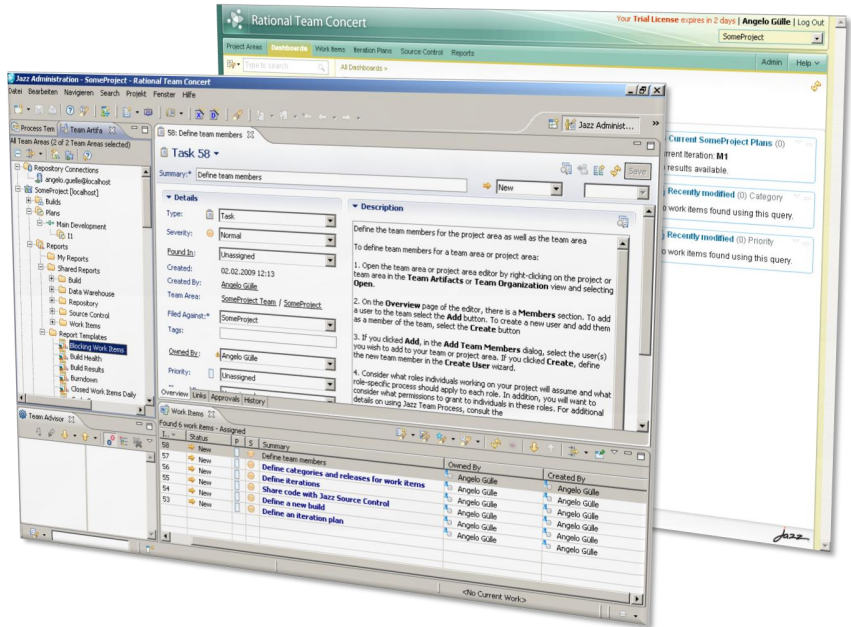


© 02/2009 Angelo Gülle

4

- Entwicklung und Wartung von Software
- Aufgabenmanagement
- Projektmanagement
- Reporting

- Rollenbasiertes Regelwerk als XML-Beschreibung: „Process Template“ !



© 02/2009 Angelo Gülle

5

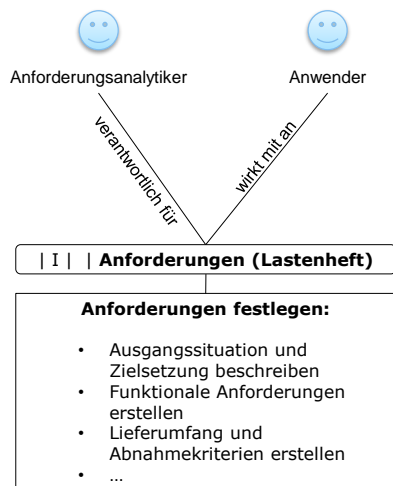
V-Modell XT

© 02/2009 Angelo Gülle

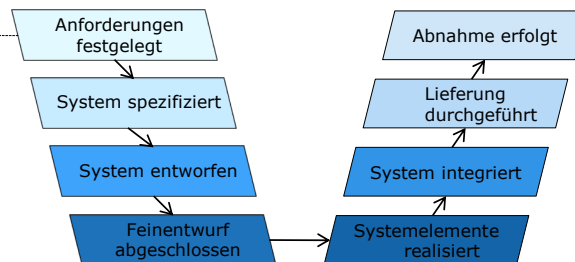
6

V-Modell XT 1.3

■ skalierbares Vorgehensmodell



- WER? -> Rolle
- WAS? -> Produkt
- WIE? -> Aktivität
- WANN? -> Entscheidungspunkt

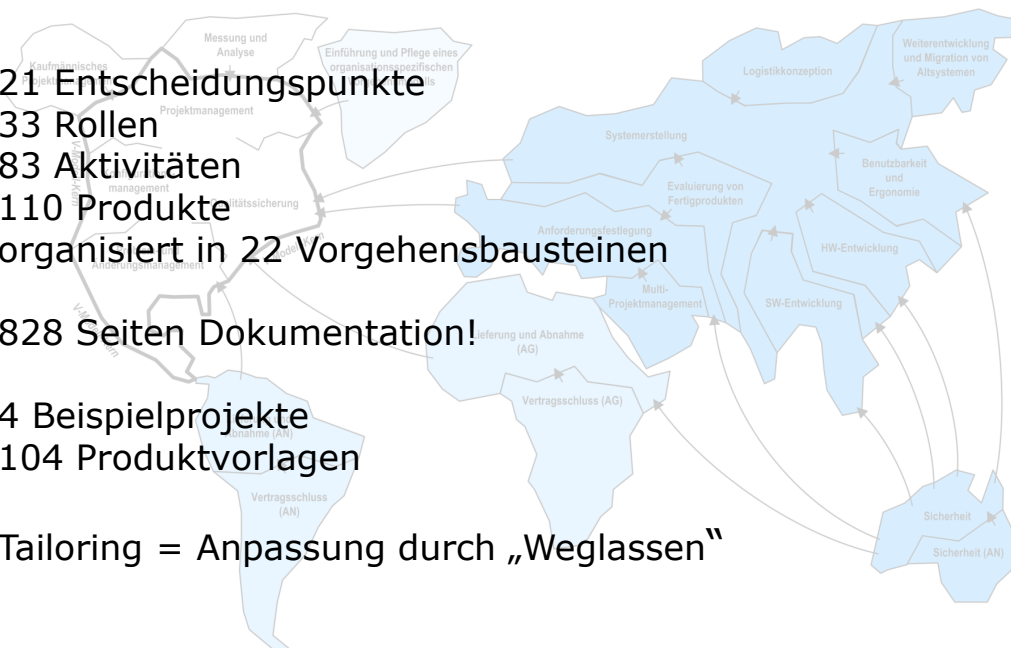


© 02/2009 Angelo Gülle

7

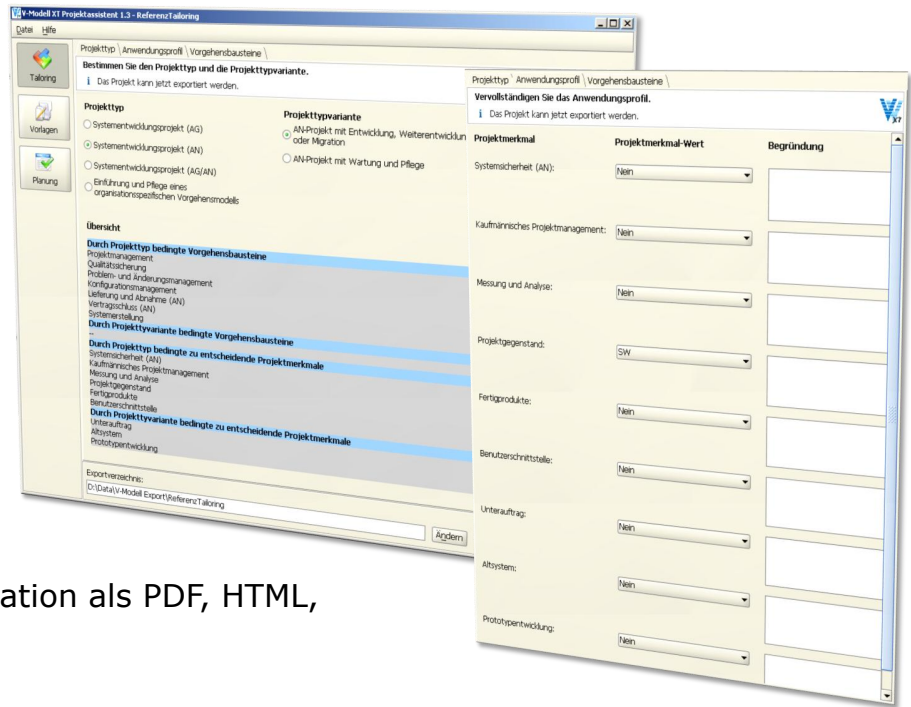
V-Modell XT 1.3 eXtreme Tailoring

- 21 Entscheidungspunkte
- 33 Rollen
- 83 Aktivitäten
- 110 Produkte
- organisiert in 22 Vorgehensbausteinen
- 828 Seiten Dokumentation!
- 4 Beispielprojekte
- 104 Produktvorlagen
- Tailoring = Anpassung durch „Weglassen“



8

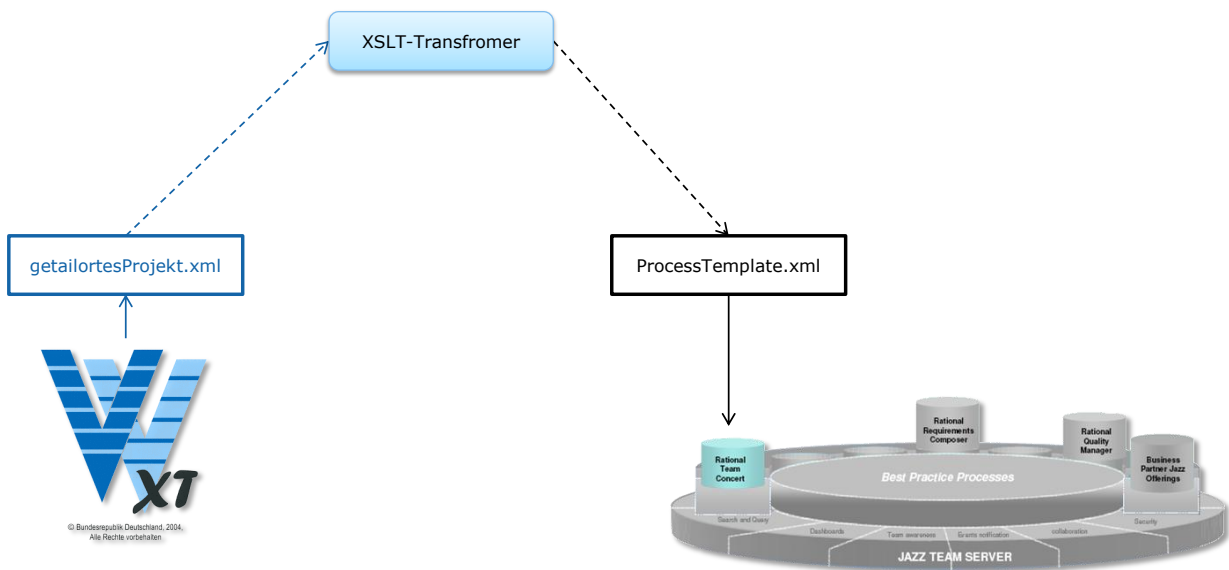
- Werkzeuggestütztes Tailoring über Projektmerkmale
- Open-Source



- Export der Dokumentation als PDF, HTML, ODT oder XML

© 02/2009 Angelo Gülle

9



© 02/2009 Angelo Gülle

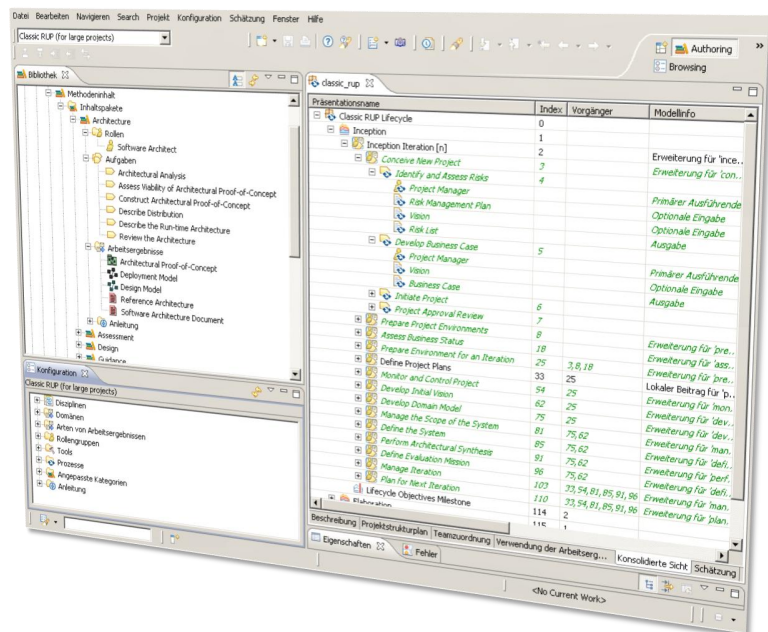
10

Rational Method Composer

© 02/2009 Angelo Gülle

11

- Erweiterung des quellenoffenen Eclipse Process Frameworks (EPF)
- Zusammenstellung von Prozessen auf Basis des Rational Unified Process (RUP)
- Tailoring mittels sog. „Konfigurationen“
- Export als PDF oder HTML
- RMC-RTC-Bücke generiert „Process Template“ für IBM-Jazz!
- Weitere Vorgehensmodelle als Methoden-Plugins nachladbar (z.B. SCRUM)



© 02/2009 Angelo Gülle

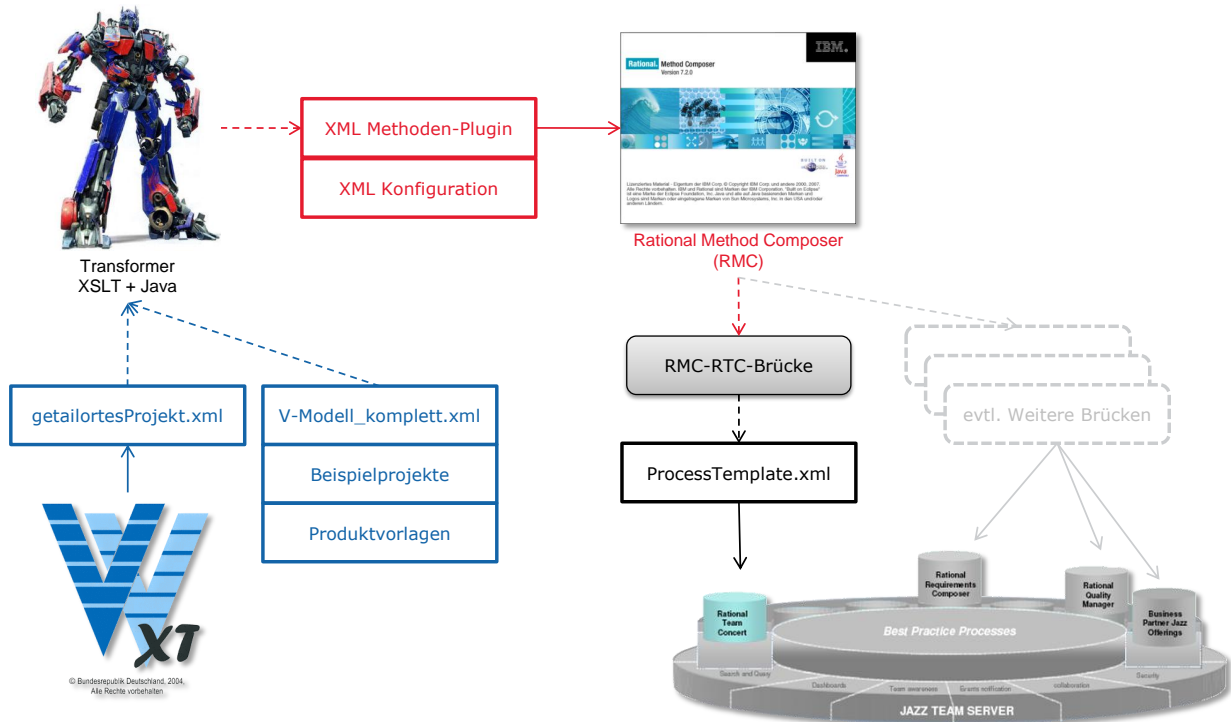
12

Umweg über den Rational Method Composer (RMC)

- Umsetzung des **un-getailorten** V-Modells XT als Methoden-Plugin
- Umsetzung des **getailorten** V-Modells XT als Konfiguration
- Integration der V-Modell XT Produktvorlagen (.rtf-Files)
- Integration der V-Modell XT Beispielprojekte

© 02/2009 Angelo Gülle

13



© 02/2009 Angelo Gülle

14

~~Lösung 1: VMXT -> RTC(Jazz)~~

Lösung 2: VMXT -> RMC -> RTC(Jazz)

Vorteile:

- Zukünftige Brücken in andere Jazz Werkzeuge -> Abbildung des gesamten Softwarelebenszyklus
- Abbildung des kompletten V-Modells XT im RMC nahezu verlustfrei möglich
- Feingranulares, nachträgliches Tailoring im RMC
- Sicherstellung der Kompatibilität zu zukünftigen Jazz Versionen
- Kombination des V-Modells XT mit anderen Vorgehensmodellen

Nachteile:

- Keine direkte Kontrolle über das Ergebnis in Jazz
- Bisher geringe Funktionalität der RMC-RTC Brücke

Demo

12.1. Generierung von Prozess Templates für IBM Jazz auf Basis des V-Modell XT

Datei Hilfe

Willkommen beim

V-Modell XT Projektassistent 1.3

Wählen Sie eine der folgenden Optionen:

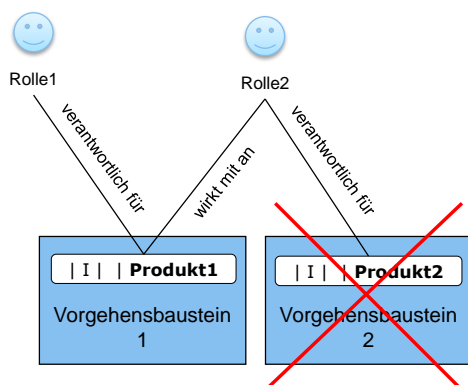
- Neues Projekt anlegen
- Bestehendes Projekt öffnen

Jazz / RTC V-Modell XT RMC Demo **Einschränkungen** Fazit

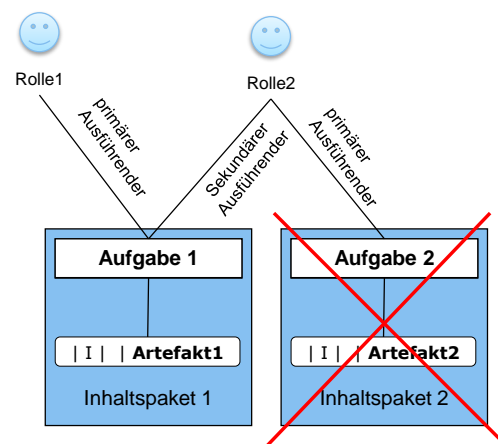
Einschränkungen: Tailoring



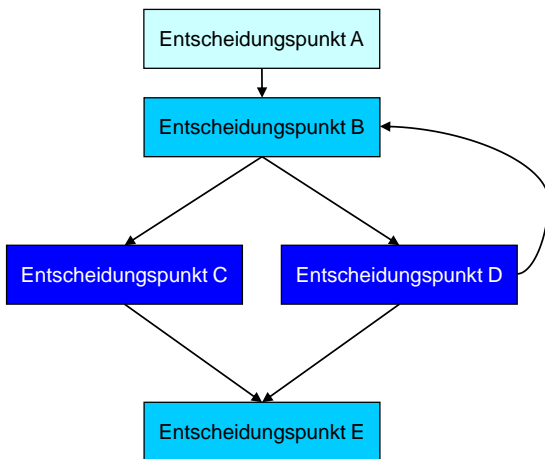
V-Modell XT



Eclipse Process Framework (RMC)



V-Modell XT



Eclipse Project Framework (RMC)

Index	Iteration	Vorgänger
1	Iteration A	
2	Iteration D	4, 4
3	Iteration C	4
4	Iteration B	2, 2
5	Iteration E	2, 3

Fazit



Fazit

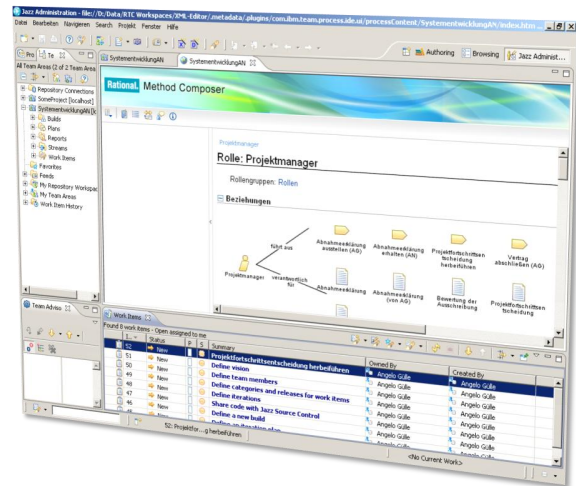
Was wurde geschafft:

- Abbildung des V-Prozessmodells
- Abbildung der V-Modell XT Rollen
- Abbildung der V-Modell XT Aktivitäten
- Integration einer interaktiven Prozessdokumentation
- Direkter Zugriff auf Produktvorlagen und Beispielprojekte

- **Schaffung eines sehr mächtigen Werkzeugs zur Prozessmodellierung mit dem V-Modell XT**

Was ist wünschenswert:

- Eine präzisere Prozesssteuerung durch das Jazz-Regelwerk
- Die Umsetzung für den kompletten Softwarelebenszyklus in weiteren Jazz-Tools (RRC, RQM, ...)



Management
& Technology
Consultants

Kontakt:
thorsten.hoffmann@bearingpoint.com

Vielen Dank!

12.2. Unibase - a unified CASE Tool

Jonas Helming, Maximilian Koegel

Technische Universitaet Muenchen

Department for Computer Science, Chair for Applied Software Engineering

Bolzmannstraße 3, D-85748 Garching bei München, Germany

{helming | koegel}@in.tum.de

Abstract

Many tools in software engineering projects support the visualization and collaborative modification of a custom set of artifacts. These include tools for requirements engineering, UML tools for the design, project management tools, developer tools and many more. In typical software engineering projects the artifacts from the respective tools are stored separately in various repositories including databases, file systems or software configuration management systems. To achieve traceability between these artifacts, different tools and especially their underlying models have to be integrated. This is a non-trivial task and is especially difficult due to limited extensibility of commercial tools in particular. We propose a unified CASE tool, which offers a unified repository for software engineering projects and specific views on this model for project participants like architects, project manager or developers. Unibase is an open-source project based on the Eclipse platform.

Motivation

The integration of a heterogeneous tool-landscape is a difficult endeavor for software engineering projects. The need for integration is triggered by the requirement for traceability between various artifacts. On one hand, parts of the specification shall be traceable inside of the system model to the origin requirements. On the other hand parts of the project model like persons, iterations or bug reports should be traceable to the affected part of the system model. Both kinds of traceability are essential preconditions for an effective management of change.

The difficulty of tool integration in software engineering projects is dimorphic: (1) First from a design point of view the integration of different tools is constrained by missing or not well-documented APIs and poor use of open exchange standards. (2) Second from an analytical point of view the underlying models of different tools have to be integrated. Many tools lack in extensibility and adaptability to integrate their model with other repositories.

Proposed Solution

This presentation introduces a unified CASE tool: unibase". The tool essentially consists of three parts: (1) A uniform and adaptable model for software engineering projects, (2) a unified repository for collaborative modification of the model, (3) unified views for textual, tabular and graphical browsing and modification of the model, tailored to specific activities.

Unified Model

We propose a unified model for software engineering projects. This model explicitly connects artifacts from the system model, i.e. the model of the system under construction, with the project model, i.e. the model of the project itself. The system model contains model elements like Requirements or Use Cases but also includes the design using artifacts such as UML ([8]) classes or flow charts. The project model contains management artifacts such as tasks or iterations as well as the organizational structure. Both models define which artifact types can be linked and the unified model defines which artifact types from the different models can be linked. The former are also referred to as intra-model links while the latter are inter-model links. The model implementation is based on the Eclipse Modeling Framework and therefore easily extensible. A modification of the unified model is even possible during project run-time with existing model instances.

Unified Repository

The unified repository is a server node which is able to store and version any instance of the unified model. It supports collaboration by providing the changes between two versions of a model instance and thereby providing an update and commit operation as known from other software configuration management systems such as Subversion. The server implementation is independent of the specific model in use.

Unified Views

Unicase provides basic views to browse and modify instances of the unified model, including graphical draw panes. These views are independent from a specific model. They work generically with any instance of a unified model. As specific activities in a software engineering project require customized views, unicase provides various specific views such as a task view for the developer or status views for the project manager. Custom views can be added easily by a plug-in mechanism.

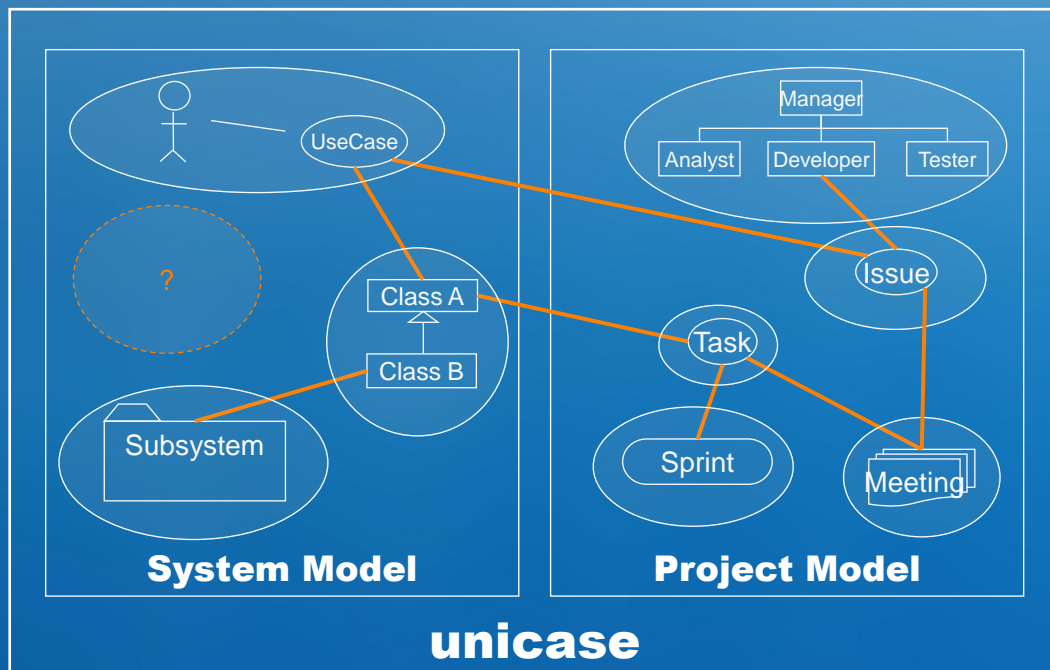
Conclusion

We are currently evaluating unicase in a student project with 25 participants with an industrial client - the Airport Munich. So far we identified two essential shortcomings of the approach, we are currently working on: (1) Effective use of the system requires a better integration with source code in terms of possibilities for code generation and traceability to parts of the source code. (2) Large projects generate huge models and require innovative approaches to consolidate them over time in terms of managing obsolete and duplicate artifacts.

Unicase

□ □ □ □
A unified CASE Tool

Jonas Helming
Technische Universität München
{helming, koegel}@in.tum.de

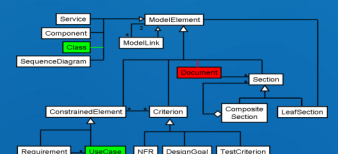
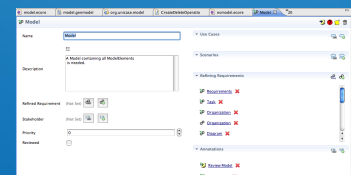
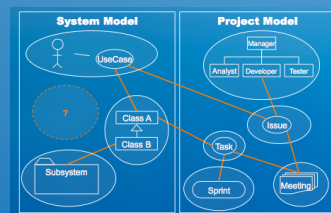


Stakeholder

- I want one integrated tool for my project
- I want to change and adapt my artefact types
- I want to extend and implement CASE-Tool support
- I need versioning for my models

Core requirements

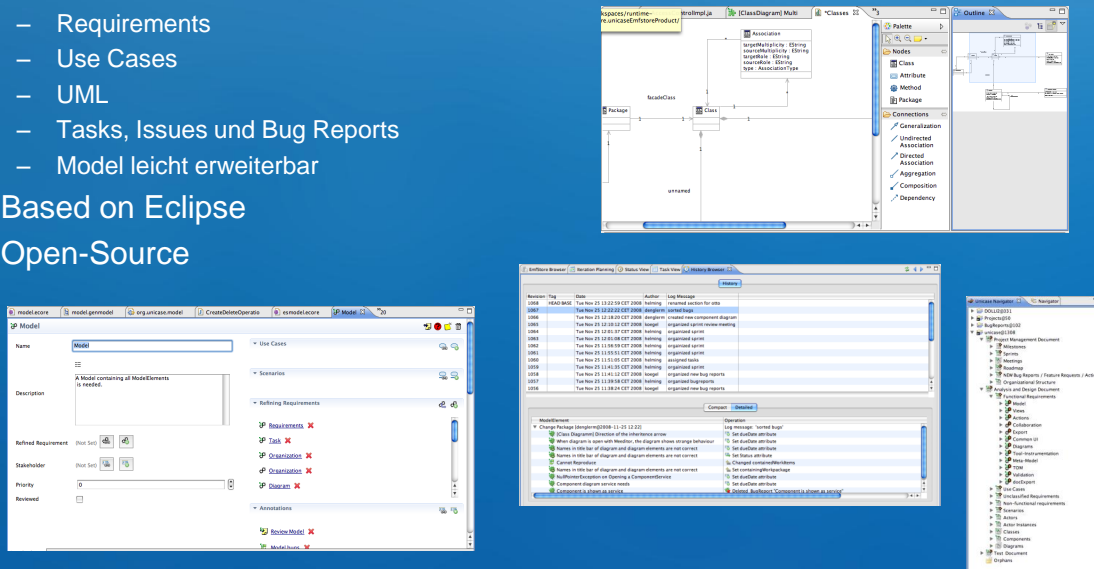
- Extensible model:
 - Generic editors and server
 - Support for migration of existing instances
- Extensible editors:
 - Model Element Editor: Widgets
 - Diagram Editor: GMF
- Collaboration support:
 - Conflict detection and merging
 - Versioning
 - Operation-based not state-based
- Integrated Project Management





Unicase - a unified CASE Tool

- Modellelemente:
 - Requirements
 - Use Cases
 - UML
 - Tasks, Issues und Bug Reports
 - Model leicht erweiterbar
- Based on Eclipse
- Open-Source



Jonas Helming - Unicase



Existing solutions

- Jazz
 - Focus on work items
 - Not open-source
- Open Software Engineering Environment (OSEE)
 - No offline mode
 - High complexity
 - No EMF

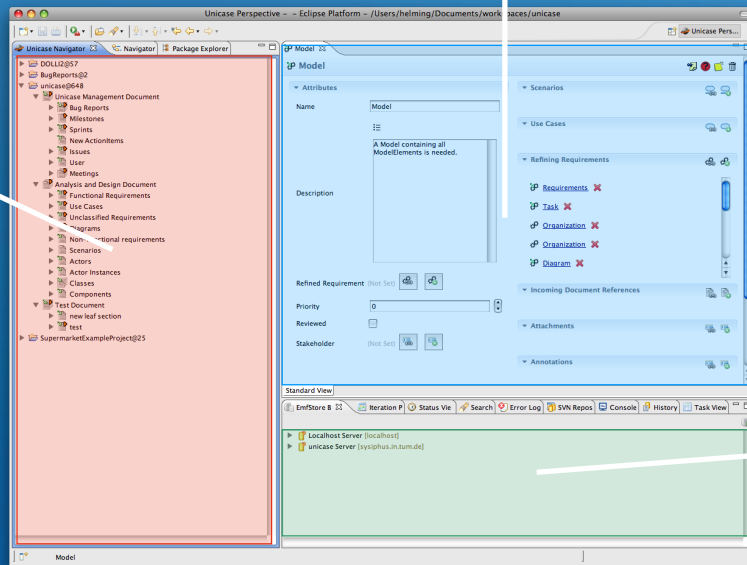


Jonas Helming - Unicase

Views

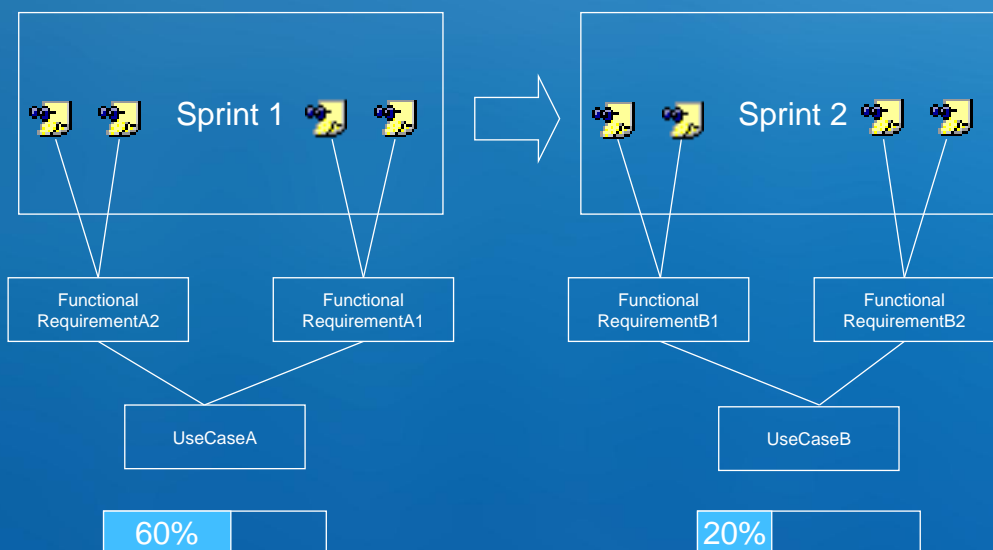
Model Element Editor

Navigator



EMFStore Browser

Example 1: Planning based on Requirements

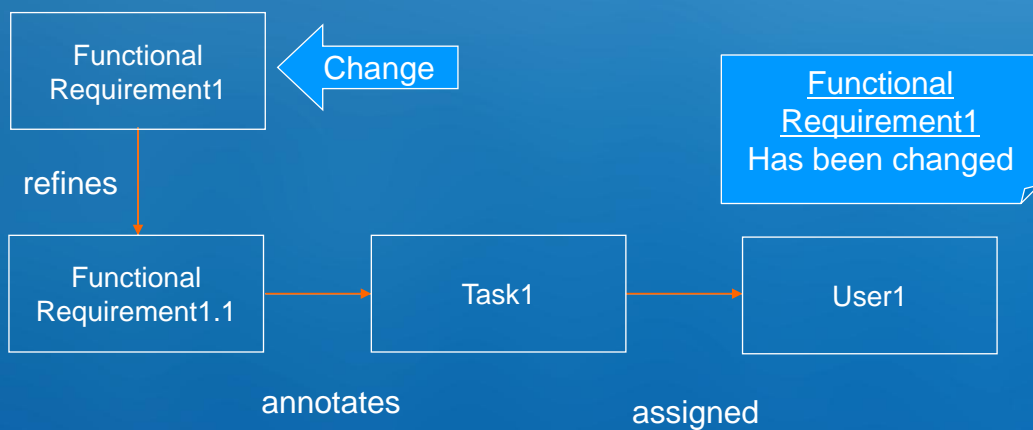


WorkItem	Annotated	Assigned to	Priority	Estimate
▼ Annotatable Model elements	N/A	N/A	0	5 / 10
▼ Add annotations	N/A	N/A	0	0 / 5
Implement Button	✚ Add annotation	N/A	0	5
▼ View Annotations	N/A	N/A	0	5 / 5
Implement Annotations view	✚ View Annotatic	helming	0	5
▼ Requirements Modelling	N/A	N/A	0	13 / 22
Functional requirement needs a stakeholder attribute	✚ Requirements	helming	0	7
model classes of requirements	✚ Requirements	hoechts	0	6
No Type attribute for NFRs	✚ Requirements	helming	0	9

Jonas Helming - Unicase





9

Example 2: Change Notification



Jonas Helming - Unicase

10

You updated your project to version 1815	06.04.2009 11:09
 Your WorkPackage Backlog has changed.	03.04.2009 13:13
 FunctionalRequirement Modelling has been modified. Trace to the changed element: Milestone content not visible in navigator => Milestone Modelling => Task Modelling => Project Modelling => Modelling	03.04.2009 13:10
 shtereva sent you a notification about this change: Created MEDiagram Usersession proxy. in Usersession proxy	31.03.2009 16:06
 You have been assigned an ActionItem Make UseCase Step widget look nice and usable	11.03.2009 09:36

Extensibility

- Open-Source (EPL)
- Model
 - Generic Views
 - Change during project runtime
 - Meta-Model evolution
 - Generic repository
- Views
 - EMF
 - GMF
- Integration with Eclipse projects



Unicase Project Setup

- Collaboration partners:
 - Academic:
 - TUM
 - LMU
 - University of Heidelberg
 - Industrial:
 - beoole
 - msg
 - Airport Munich
 - Siemens Corporate Research
- Development team:
 - 4 (+4) doctoral candidates
 - 8 students



Jonas Helming - Unicase

13

It's open source, you can join...
www.unicase.org

helming@in.tum.de

12.3. Anforderungen nachweisbar erfüllen

Alexander Auerbach

Abstract

In vielen Softwareentwicklungsprojekten ist die konsistente Nachverfolgbarkeit der Anforderungen von ihrer Definition über die Implementierung bis zum Test eine entscheidende Voraussetzung für den Erfolg. Insbesondere wer nachweisen muss, dass der Entwicklungsprozess konform nach CMMI, SPICE oder V-Modell XT erfolgt, benötigt eine effektive und effiziente Traceability-Lösung. Der Ansatz der komponentenbasierten Modellierung ermöglicht eine solche Lösung mit Werkzeugunterstützung und ohne organisatorischen Overhead. Im Ergebnis ist es möglich, jederzeit für eine Anforderung festzustellen, wo sie im Code erfüllt wird und mit welchem Test das bewiesen wird. Für den hier vorgestellten Ansatz steht mit netCCM Studio Professional umfassende Werkzeugunterstützung bereit.

Die Herausforderung: Konsistenz im Entwicklungsprozess

Eine entscheidende Voraussetzung für den Projekterfolg in professionellen Softwareentwicklungsprozessen ist die durchgängige Nachverfolgung der an das Endprodukt gestellten Anforderungen. Als besondere Schwierigkeit stellt sich dabei heraus, dass sich die Anforderungen während des Entwicklungsprozesses praktisch immer ändern. Bei der mit professionellen Software-Entwicklungsprojekten verbundenen Komplexität ist damit die Beantwortung der Frage, welche Anforderungen von einer bestimmten Version der erstellten Software wirklich erfüllt werden, nicht leicht zu beantworten.

Betrachtet man bei der Suche nach einer Lösung dieses Problems die zentralen Aktivitäten des Softwareentwicklungsprozesses gemäß ISO 12207, so fällt auf, dass für eine erfolgreiche Softwareentwicklung die Ergebnisse der einzelnen Aktivitäten konsistent zueinander sein müssen. So muss zum Beispiel ein Softwaredesign zu den Softwareanforderungen passen, die erfüllt werden sollen. Abbildung 12.1 zeigt die wichtigsten Konsistenzbeziehungen.

Bei der Betrachtung dieser Aktivitäten fällt auf, dass praktisch jede Aktivität von unterschiedlichen Rollen im Entwicklungsprozess wahrgenommen wird. Damit verbunden sind jeweils spezialisierte Werkzeuge für die jeweiligen Aktivitäten, was zwangsläufig zu Medienbrüchen führt. Das erfordert typischerweise erheblichen manuellen Aufwand für die Überbrückung dieser Medienbrüche.

Eine Traceability-Lösung erfordert unter diesen Bedingungen einen hohen organisatorischen Aufwand und Nachweislücken bei Anforderungserfüllung lassen sich kaum vermeiden.

Komponentenbasierte Modellierung

Das Vermeiden von Medienbrüchen bzw. das Schaffen automatischer Medienübergänge ist die Motivation für komponentenbasierte Modellierung. Dabei wird die Struktur der zu erstellenden Software, also Systemarchitektur und Softwaredesign, in den Mittelpunkt gestellt.

12.3. Anforderungen nachweisbar erfüllen

Es ist naheliegend, dass im Softwaredesign Komponenten auch zu größeren Einheiten zusammengestellt werden können, die dann selbst wieder zu noch größeren Einheiten zusammengestellt werden. Eine solche Einheit wird Aggregation genannt. Sie erscheint im Modell wie eine Komponente, verfügt jedoch weder über Code noch über Metadaten. Das Verhalten einer Aggregation wird vielmehr durch das Zusammenwirken ihrer Komponenten definiert. Die Anschlusspunkte einer Aggregation ergeben sich ebenfalls aus ihrer Definition als Zusammenstellung von Komponenten. Aggregation können neben Komponenten auch andere, kleinere Aggregationen enthalten.

Abbildung 12.2 zeigt ein einfaches Modell in dem komponentenbasierten Modellierungswerkzeug netCCM Studio. Auf der linken Seite ist in der Baumstruktur die Systemarchitektur bestehend aus einem Executable (Standalone Application PIM) und einer Bibliothek (BusinessLogic) zu erkennen. Das Diagramm auf der rechten Seite zeigt die oberste Ebene des Softwaredesigns bestehend aus einer Komponente und vier Aggregationen.

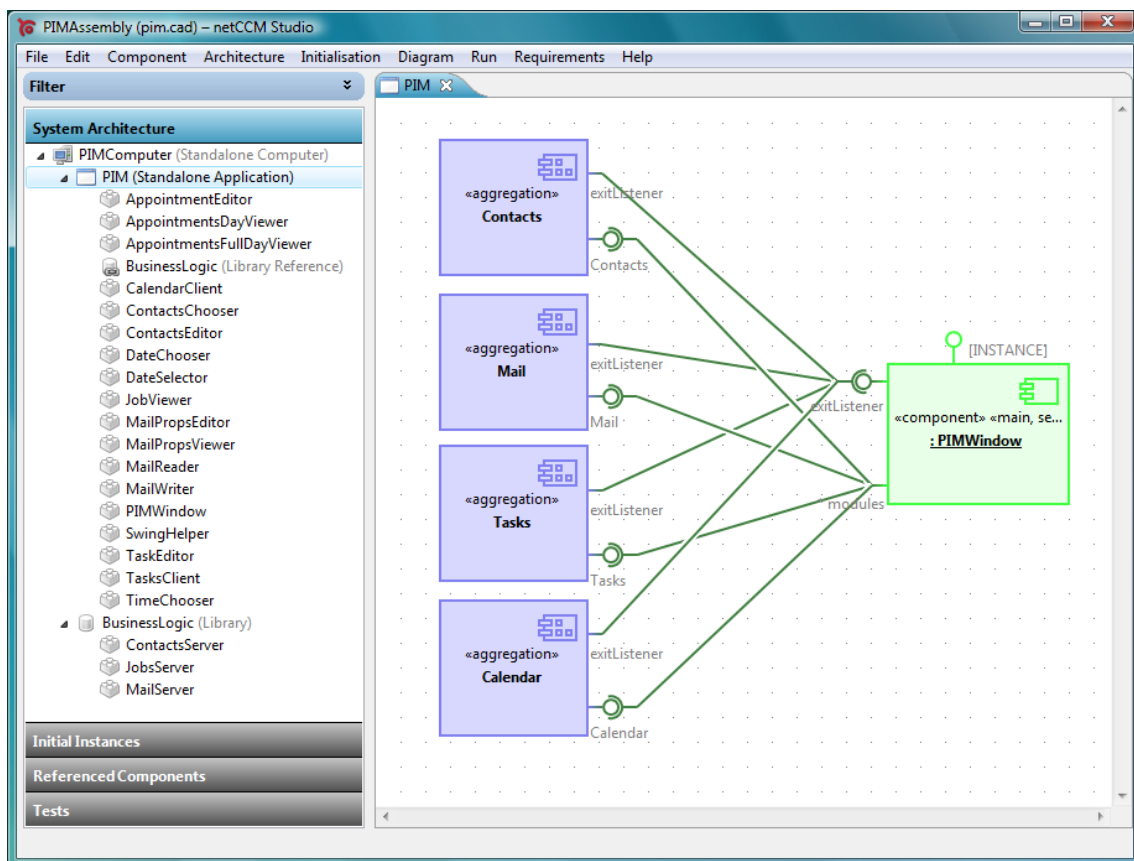


Abbildung 12.2.: Modellierungswerkzeug netCCM Studio

Konsistenz von Modell und Code

Die Konsistenz zwischen Systemarchitektur und Softwaredesign kann dabei einfach sichergestellt werden, indem nur definierte Systemelemente durch ein Softwaredesign verfeinert werden können. Außerdem können im Softwaredesign nur Komponenten verwendet werden, die dem entsprechenden Systemelement direkt oder indirekt über Abhängigkeitsbeziehungen zugeordnet wurden. Betrachtet man ein solches Modell von

Systemarchitektur und Softwaredesign nicht bottom-up aus Sicht der Komponenten sondern top-down aus Sicht des Softwaresystems, so stellt es eine formal definierte, vollständige und konsistente funktionale Dekomposition des Softwaresystems dar.

In dem Modell stehen damit alle Informationen über das Softwaresystem zur Verfügung: Das Verhalten wird von den Komponentenimplementierungen definiert und die Struktur beschreibt das Modell. Damit kann aus einem solchen Modell automatisch das Softwaresystem erstellt werden. Dafür wird für die im Modell definierte Struktur Infrastrukturcode generiert. Außerdem werden für die zu den Metadaten der Komponenten gehörenden abstrakt definierten Schnittstellen äquivalente Schnittstellen in der jeweiligen Programmiersprache generiert. Anschließend werden alle Komponentenimplementierungen und der generierte Infrastrukturcode kompiliert und entsprechend der Systemarchitektur zu Systemelementen gelinkt. Ist dieser automatische Vorgang erfolgreich, so ist sichergestellt,

- dass das Modell vollständig ist,
- dass das Modell konsistent ist und
- dass die Komponentenimplementierungen konsistent zu ihren Metadaten entwickelt wurden.

Damit wird die Verantwortung für das Konsistenthalten von Architekturbeschreibung und Code zu den Entwicklern verlagert, die diese Konsistenz im Gegensatz zu anderen Projektrollen auch problemlos sicherstellen können. Denn die Aufgabe von Entwicklern besteht darin, kompilierbaren Code zu liefern. Verstößen sie gegen die Architekturdefinition, so kann der Code nicht kompiliert werden. Gleichzeitig können sie die Architekturbeschreibung einfach im Modell anpassen, wenn es zur Erfüllung einer Anforderung erforderlich ist. Das Modell wird damit zum verlässlichen Kommunikationsmittel zwischen Entwicklern und Architekten. Abbildung 12.3 zeigt die Integration des komponentenbasierten Modellierungswerkzeugs netCCM Studio in die Werkzeugkette.

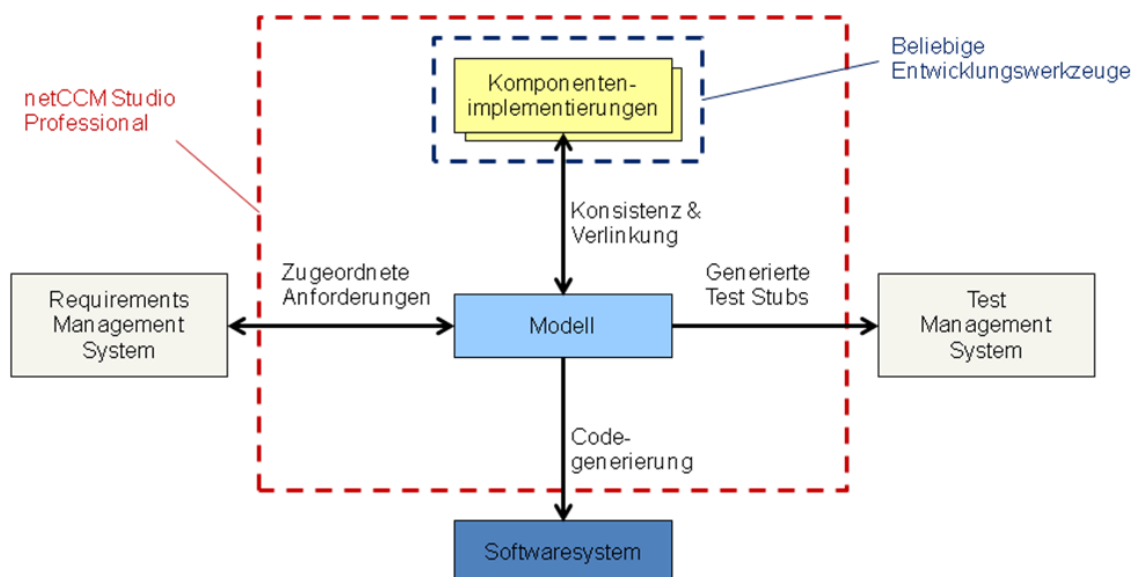


Abbildung 12.3.: Integration des komponentenbasierten Modellierungswerkzeugs netCCM Studio in die Werkzeugkette

Nachverfolgen von Anforderungen

Jedem Modellelement, das der Architekt für die Erfüllung einer oder mehrerer Anforderungen anlegt, können diese sofort zugeordnet werden. Dafür können die Anforderungen selbst in den üblichen Werkzeugen verwaltet werden. Sie müssen lediglich über eine ID identifizierbar sein.

Damit besteht die Möglichkeit, Anforderungen entsprechend des Softwaredesigns zu verfeinern. So kann eine allgemeine Anforderung einer Aggregation zugeordnet werden, die aus mehreren Komponenten besteht. Diese allgemeine Anforderung kann nun zu spezifischeren Anforderungen verfeinert werden, die beschreiben, wie die einzelnen Komponenten zur Erfüllung der allgemeinen Anforderung beitragen sollen.

Im Ergebnis muss der Entwickler, der eine einzelne Komponente implementiert, nicht alle Anforderungen kennen. Die für seine Arbeit relevanten Anforderungen wurden durch den Architekten mit der Zuordnung zu der entsprechenden Komponente bereits herausgefiltert. Gleichzeitig erhält der Architekt ein Werkzeug, mit dem er leicht überblicken kann, ob er in seinem Softwaredesign alle gestellten Anforderungen berücksichtigt hat. Die Anforderungen selbst werden dabei in den üblichen Werkzeugen verwaltet. Sie müssen lediglich über eine ID identifizierbar sein.

Nachweis der Erfüllung von Anforderungen durch Testen

Analog zur Generierung des Infrastrukturcodes für die Erstellung des Softwaresystems kann auch eine Testinfrastruktur aus einem Modell automatisch generiert werden. Dafür werden Modellelemente, denen Anforderungen zugewiesen wurden, für das Testen markiert. Für jedes markierte Modellelement kann jederzeit ein so genannter Teststub generiert werden.

Dabei handelt es sich um ein Programm, das den Code enthält, den das Modellelement repräsentiert. Außerdem stellt es eine Nutzerschnittstelle bereit, mit der der Teststub bedient werden kann. Dabei greift der Generator für den Teststub auf die Modellinformationen zurück, in denen alle angebotenen Schnittstellen beschrieben sind. In der generierten Nutzerschnittstelle können dann alle angebotenen Methoden mit Parameterwerten aufgerufen werden, woraufhin die jeweiligen Rückgabewerte angezeigt werden.

Mit einem Teststub kann damit ein beliebiger Teil des Softwaresystems isoliert getestet werden, indem seine Umgebung simuliert wird. Der Test kann dabei sowohl manuell als auch automatisch mit einem Testautomatisierungssystem erfolgen.

Abbildung 12.4 zeigt die Auswirkungen auf den Software-Entwicklungsprozess, die sich aus dem Einsatz eines komponentenbasierten Modellierungswerkzeugs ergeben.

Zusammenfassung

Mit dem Ansatz komponentenbasierter Modellierung können Anforderungen in beliebigen Softwareentwicklungsprozessen von ihrer Definition über die Implementierung bis zum Test konsistent nachverfolgt werden. Dabei kommen in den einzelnen Aktivitäten des Software-Entwicklungsprozesses die üblichen spezialisierten Werkzeuge zum Einsatz. Daneben vermeidet ein zusätzlich eingesetztes komponentenbasiertes Modellierungswerkzeug wie netCCM Studio Professional Medienbrüche bzw. automatisiert Medienübergänge und sorgt so für durchgängige Konsistenz. Auf diese Weise kann für jede

Anforderungen nachweisbar erfüllen

Alexander Auerbach

*netCCM GmbH
Rudower Chaussee 25
12489 Berlin*

www.netccm.com
a.auerbach@netccm.com



Inhalt

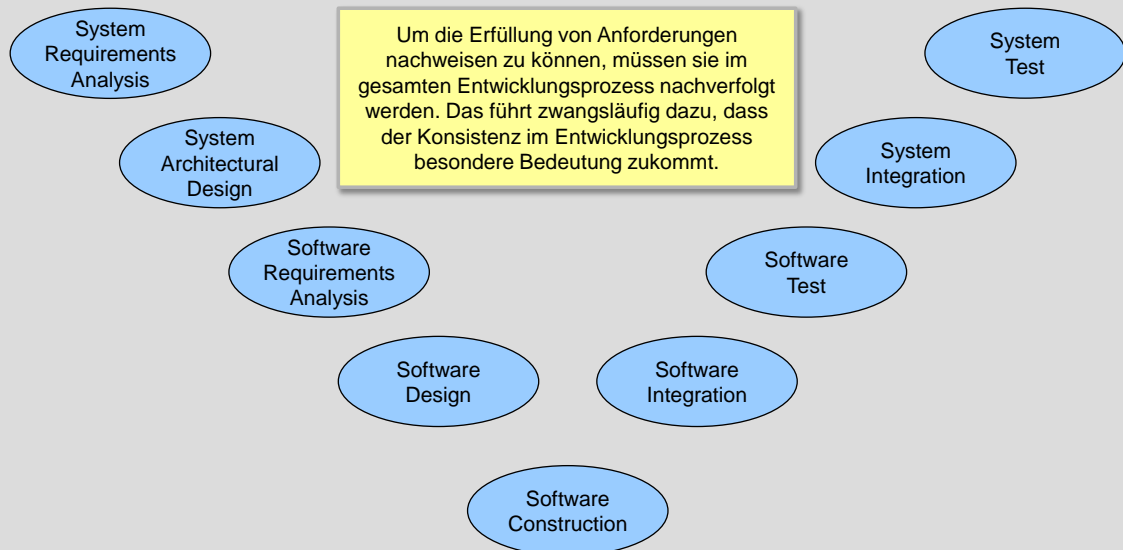
- Anforderungen nachweisbar erfüllen:
Konsistenz im Entwicklungsprozess
- Komponentenbasierte Modellierung
 - Nachverfolgen von Anforderungen
 - Nachweis der Erfüllung von Anforderungen durch Tests
- Werkzeugunterstützung
- Zusammenfassung

Folie 2

www.netccm.com
a.auerbach@netccm.com



Konsistenz im Software-Entwicklungsprozess

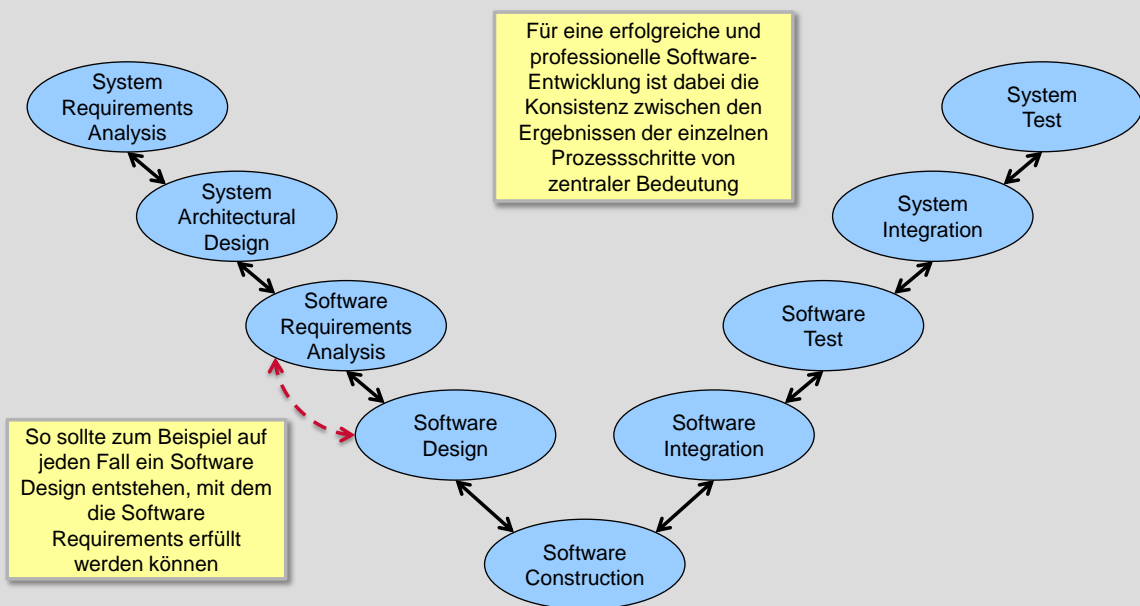


Folie 3

www.netccm.com
a.auerbach@netccm.com



Konsistenz im Software-Entwicklungsprozess

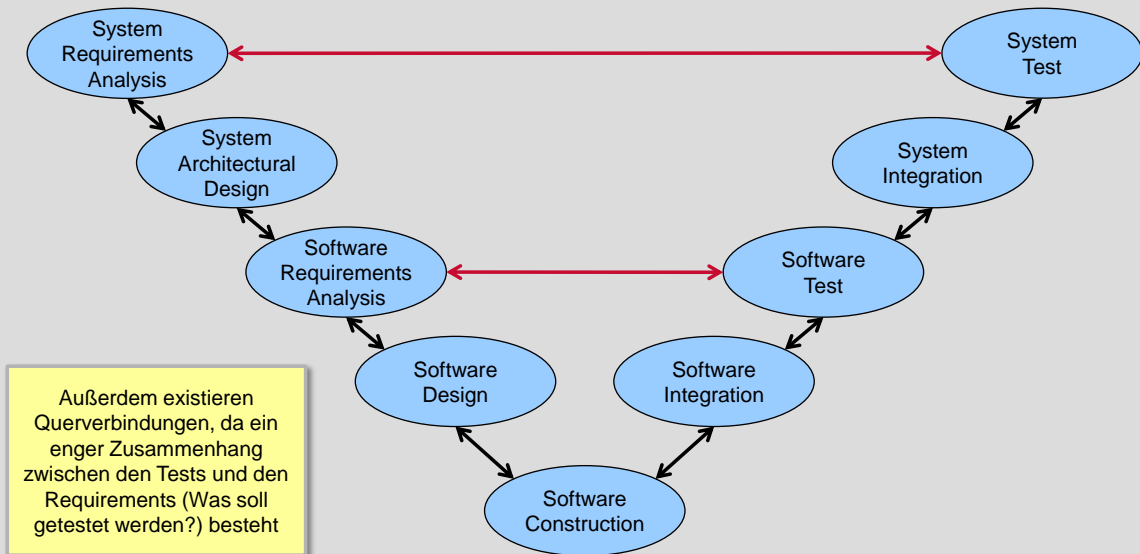


Folie 4

www.netccm.com
a.auerbach@netccm.com



Konsistenz im Software-Entwicklungsprozess

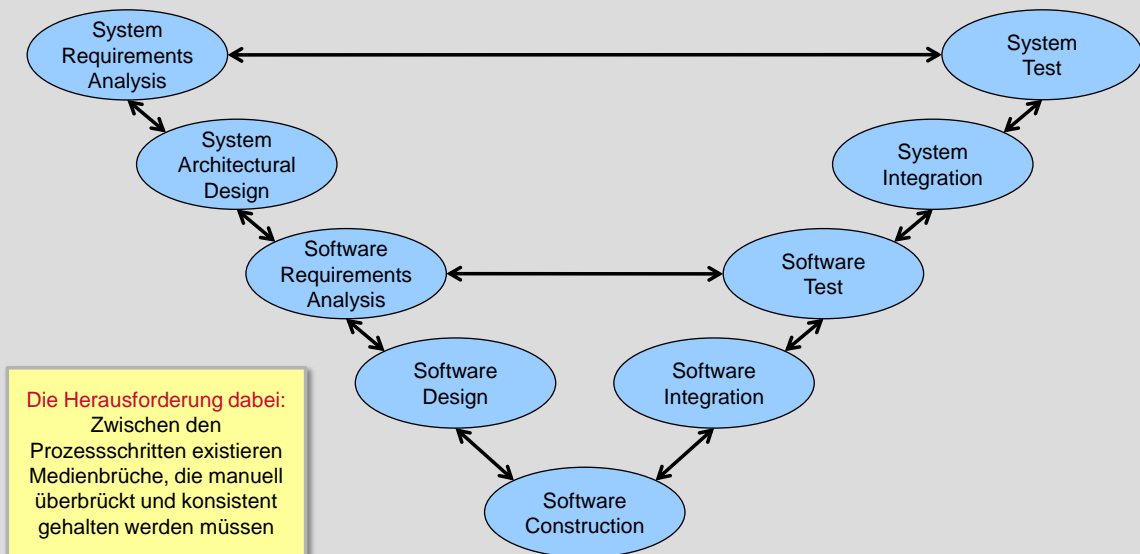


Folie 5

www.netccm.com
a.auerbach@netccm.com



Konsistenz im Software-Entwicklungsprozess

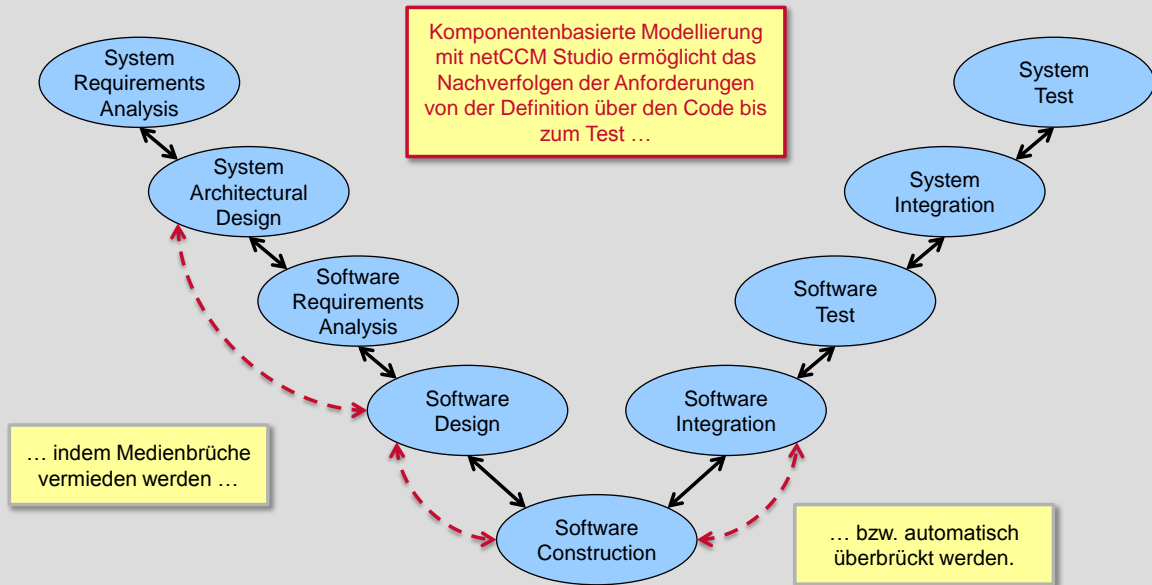


Folie 6

www.netccm.com
a.auerbach@netccm.com



Komponentenbasierte Modellierung

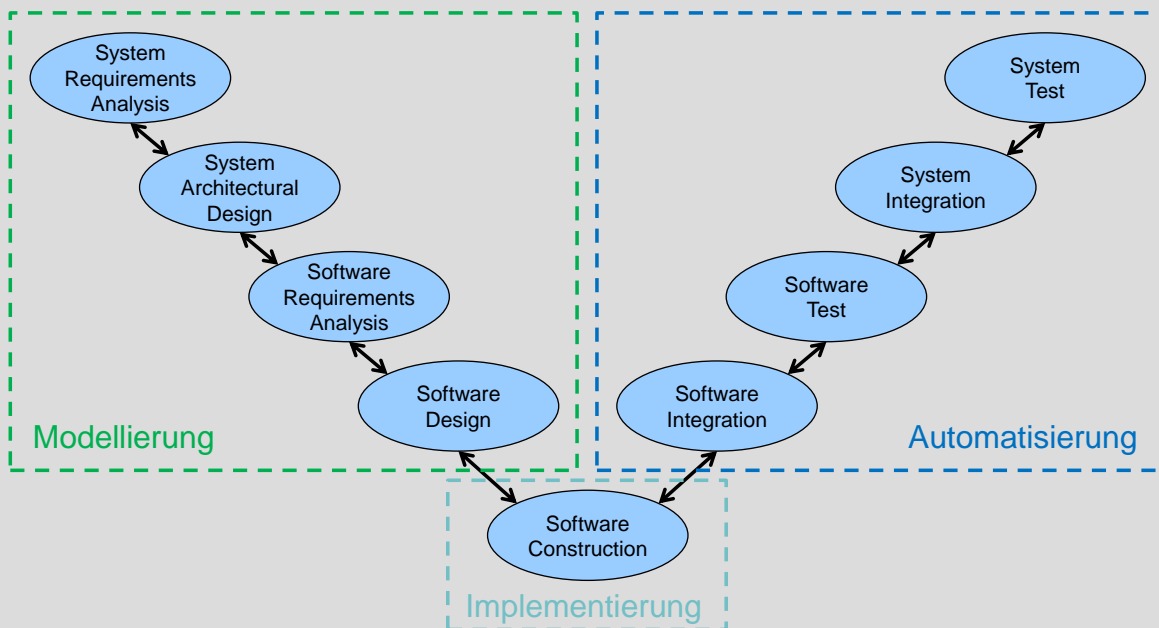


Folie 7

www.netccm.com
a.auerbach@netccm.com



Komponentenbasierte Modellierung

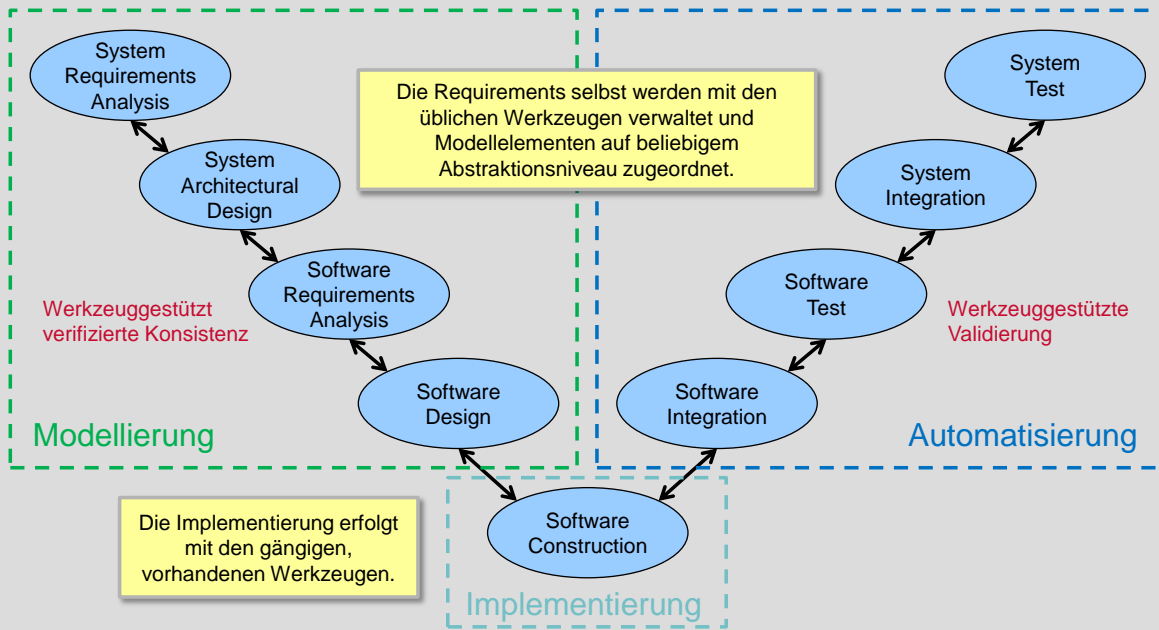


Folie 8

www.netccm.com
a.auerbach@netccm.com



Komponentenbasierte Modellierung

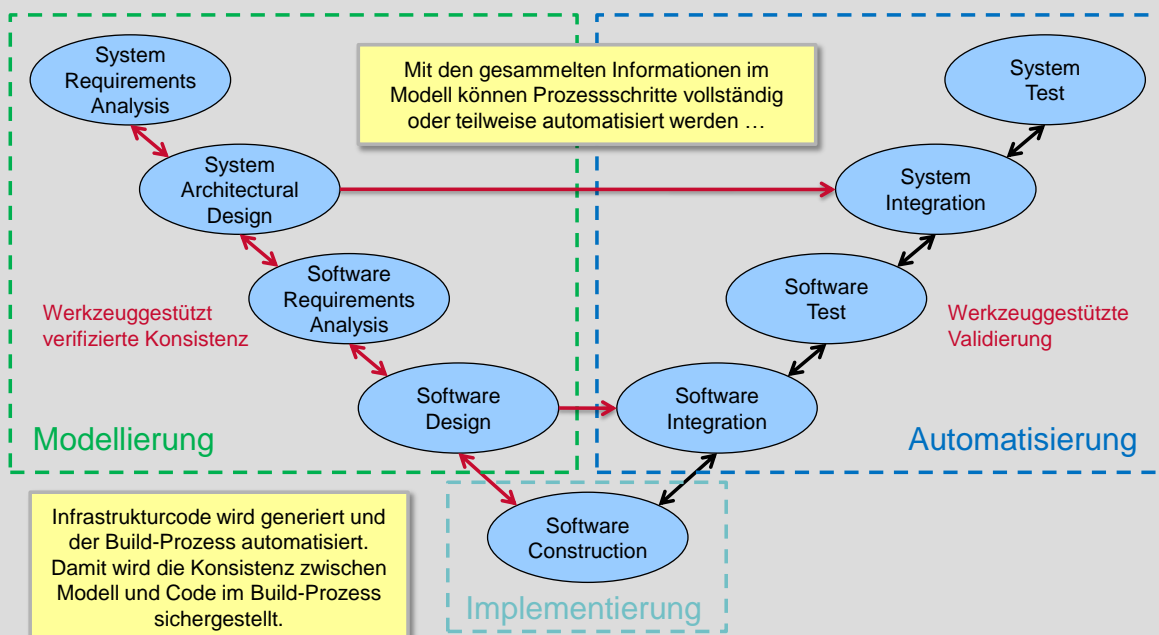


Folie 9

www.netccm.com
a.auerbach@netccm.com



Nachverfolgen von Anforderungen mit netCCM Studio Professional

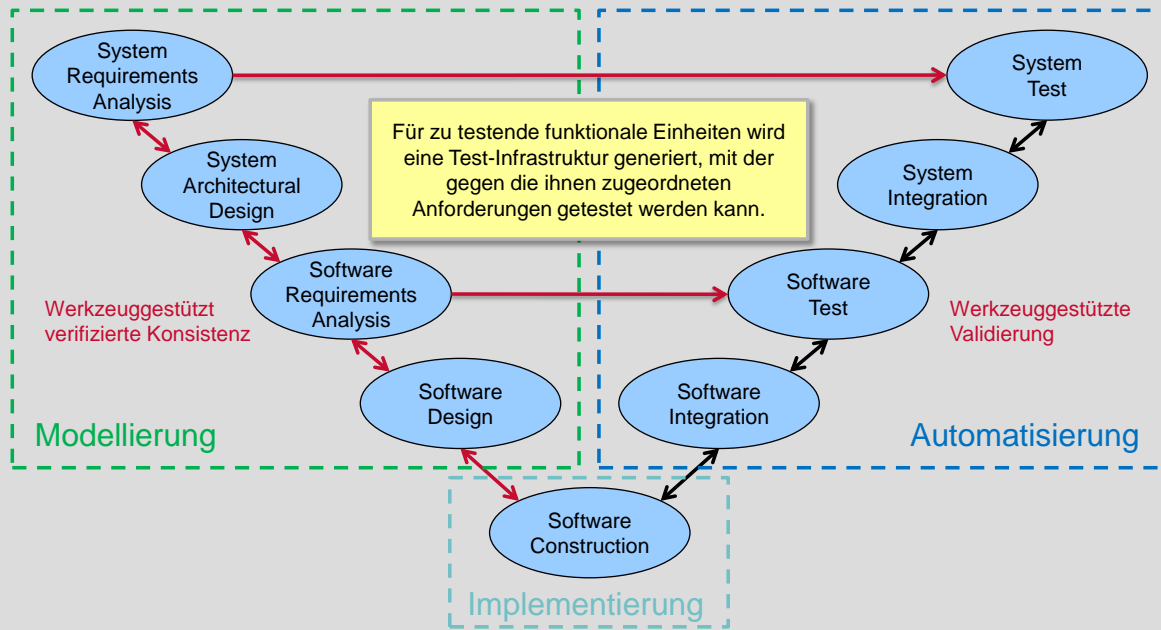


Folie 10

www.netccm.com
a.auerbach@netccm.com



Nachverfolgen von Anforderungen mit netCCM Studio Professional

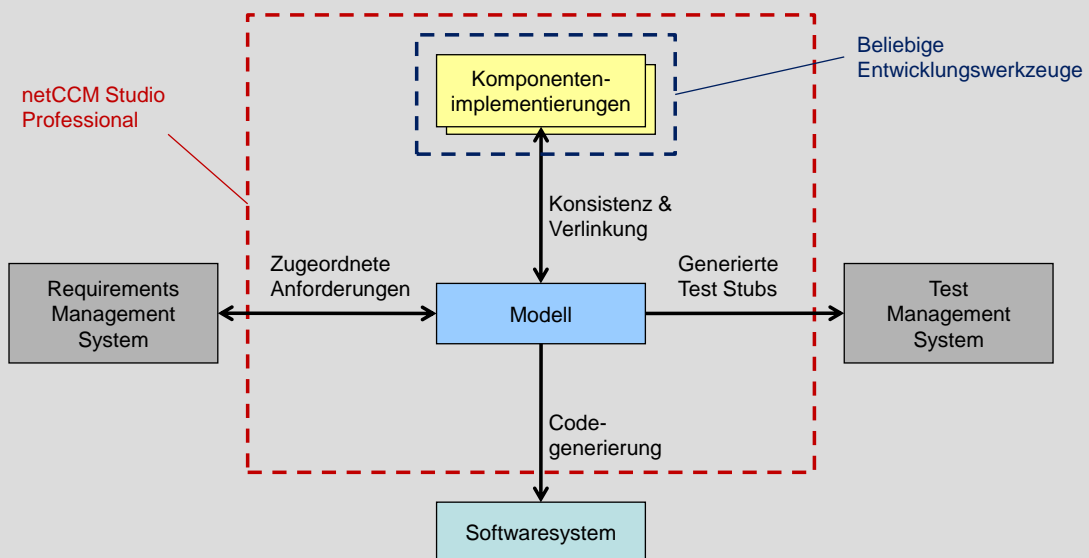


Folie 11

www.netccm.com
a.auerbach@netccm.com



Werkzeugunterstützung mit netCCM Studio

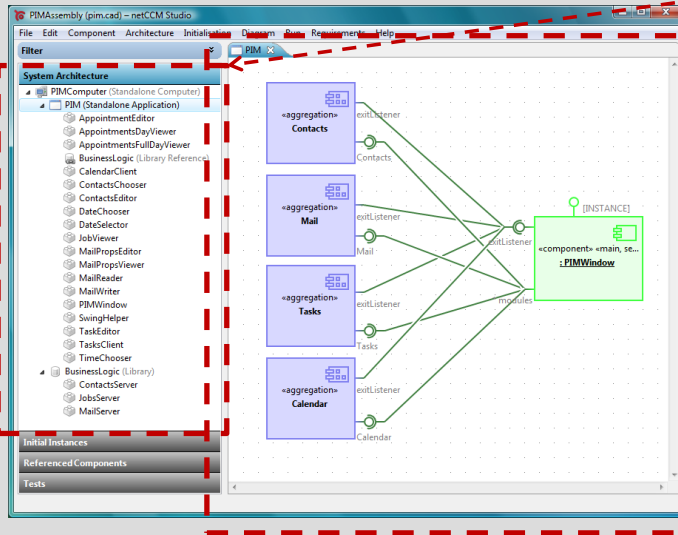


Folie 12

www.netccm.com
a.auerbach@netccm.com



Werkzeugunterstützung mit netCCM Studio



Die Systemarchitektur kann meist in 5 Minuten definiert werden. Sie spezifiziert, welche Executables, Webservices, Bibliotheken u.s.w. zu dem Softwaresystem gehören. Diesen Systemelementen werden Komponenten, also Codeteile zugeordnet.

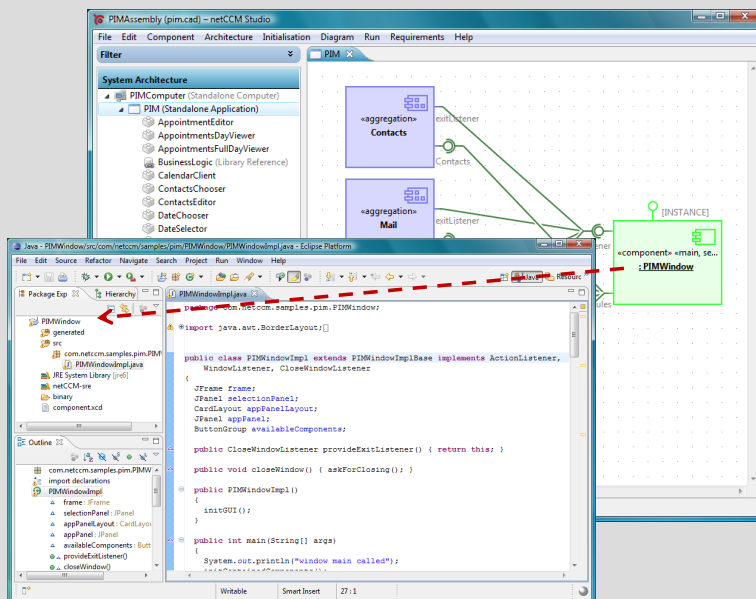
Im Softwaredesign werden die Komponenten miteinander verknüpft.

Folie 13

www.netccm.com
a.auerbach@netccm.com

netCCM
SOFTWARE ENGINEERING SOLUTIONS

Werkzeugunterstützung mit netCCM Studio



Zu jeder Komponente gehört Code, der mit den verfügbaren, vertrauten Werkzeugen wie Eclipse, Visual Studio oder NetBeans entwickelt wird. Die Konsistenz zwischen Modell und Code wird kontinuierlich sichergestellt.

Folie 14

www.netccm.com
a.auerbach@netccm.com

netCCM
SOFTWARE ENGINEERING SOLUTIONS

Zusammenfassung

- Sowohl Entwicklern als auch Testern stehen die für sie jeweils relevanten Requirements jederzeit zur Verfügung
- Der Nachweis der Erfüllung von Anforderungen erfolgt praktisch automatisch
- Zertifizierung nach SPICE, CMMI oder V-Modell XT wird drastisch vereinfacht
- Aufwände für manuelles Pflegen und Konsistenthalten von Arbeitsergebnissen werden vermieden

Folie 15

www.netccm.com
a.auerbach@netccm.com



Zusammenfassung

netCCM Studio Professional

- garantiert die Konsistenz der Arbeitsergebnisse des Entwicklungsprozesses über die gesamte Lebensdauer der Software.
- ist die einzige Lösung, die durchgängig werkzeuggestützt das Verfolgen von Anforderungen von der Definition über den Code bis zum Test ermöglicht.

Vielen Dank für Ihre Aufmerksamkeit.
Weiter Informationen finden Sie unter www.netccm.com

Folie 16

www.netccm.com
a.auerbach@netccm.com

