

Software-Service-Co-Design

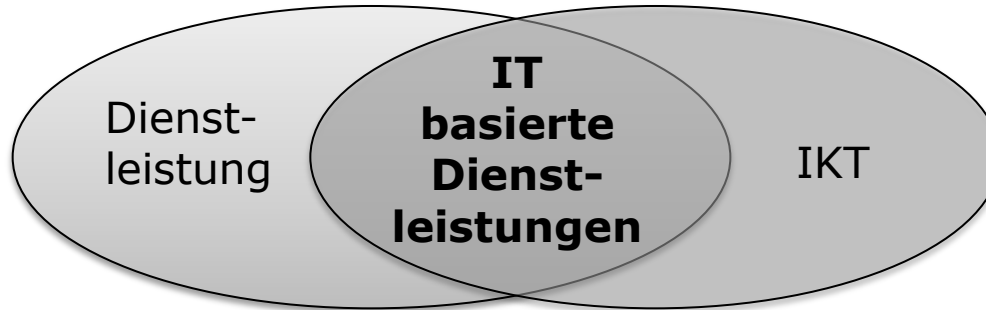
- Eine Methodik für die Entwicklung komponentenorientierter IT-basierter Dienstleistungen -

Kyrill Meyer

meyer@informatik.uni-leipzig.de

26.05.2009

Wandel zu IT-basierten Dienstleistungen

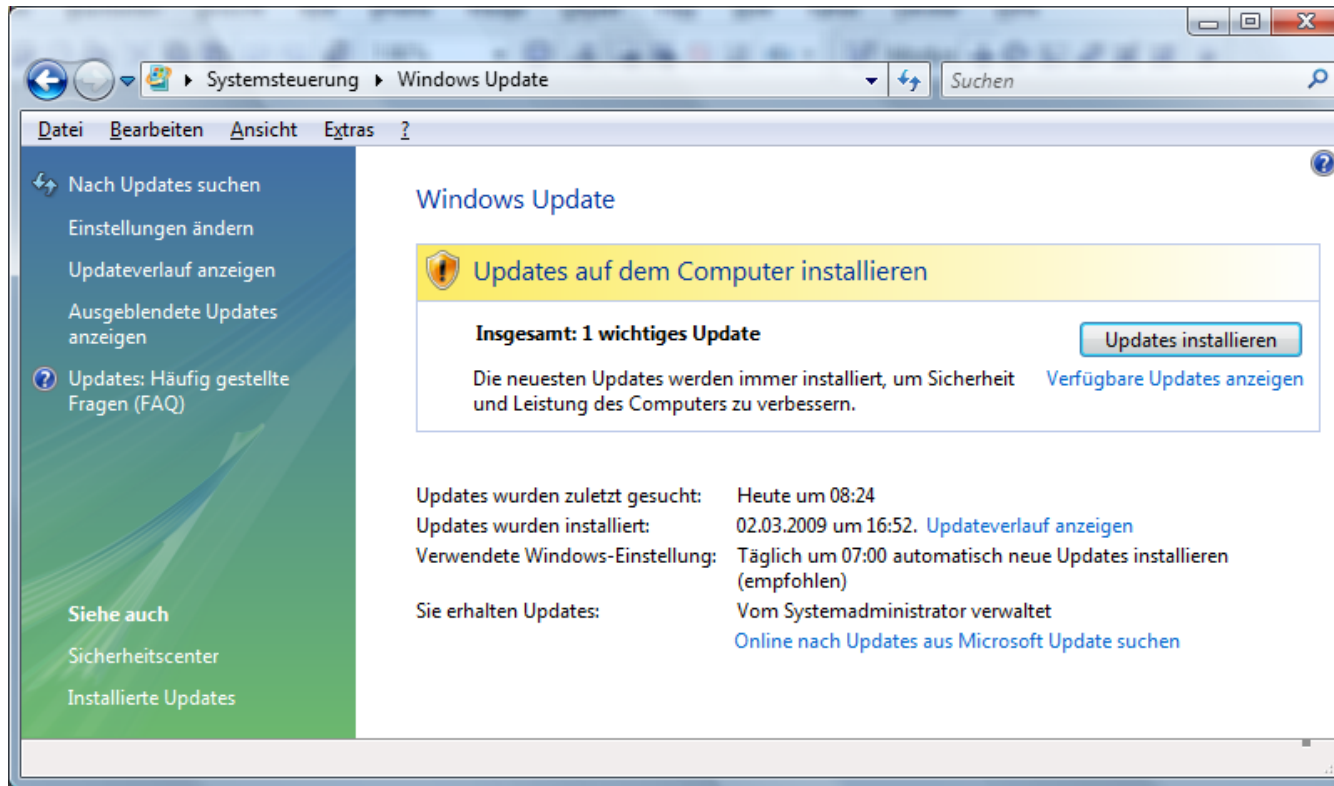


- Beispiele:
 - B2C: E-Business (z.B. Amazon, I-Tunes), Online-Banking, Ticket Terminals, DHL Packstation
 - Governmental: E-Government, E-Learning
 - B2B: CRM-Dienstleistungen, Remote Services, IT-Services



IT-basierte Dienstleistungen: Beispiele

- Software Updates



IT-basierte Dienstleistungen: Beispiele

- E-Commerce

Amazon.com: Advances in Services Innovations: Dieter Spath, Klaus-Peter Fähnrich: Books - Mozilla Firefox

File Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

http://www.amazon.com/Advances-Services-Innovations-Dieter-Spath/dp/35402

amazon.com Hello. Sign in to get personalized recommendations. New customer? Start here. FREE 2-Day Shipping, No Minimum Purchase

Your Amazon.com Today's Deals Gifts & Wish Lists Gift Cards Your Account | Help

Shop All Departments Search Books GO Cart Your Lists

Books Advanced Search Browse Subjects Hot New Releases Bestsellers The New York Times® Best Sellers Libros En Español Bargain Books Textbooks

Advances in Services Innovations and over 240,000 other books are available for Amazon Kindle – Amazon's new wireless reading device. Learn more

Click to LOOK INSIDE!

Advances in Services Innovations (Hardcover)

by Dieter Spath (Editor), Klaus-Peter Fähnrich (Editor) "As the significance of the service sector has grown, competition in many service markets has increased markedly over the last few years..." (more)

Key Phrases: favorable customer experiences, low contact intensity, early entrant advantages, New York, Journal of Marketing, Deutsches Institut (more...)

★ ★ ★ ★ ☆ (1 customer review)

List Price: \$115.00

Price: \$91.73 & this item ships for FREE with Super Saver Shipping. Details

You Save: \$23.27 (20%)

Special Offers Available

In Stock.

Ships from and sold by Amazon.com. Gift-wrap available.

Want it delivered Friday, March 13? Order it in the next 13 hours and 14 minutes, and choose One-Day Shipping at checkout. Details

15 new from \$90.81 8 used from \$88.78

Quantity: 1

Add to Shopping Cart

or

Sign in to turn on 1-Click ordering.

or

Add to Cart with FREE Two-Day Shipping

Amazon Prime Free Trial required. Sign up when you check out. Learn More

More Buying Choices

23 used & new from \$88.78

Have one to sell? Sell yours here

Add to Wish List

Share your own customer images

Search inside this book

Start reading Advances in Services Innovations on your Kindle

IT-basierte Dienstleistungen: Beispiele

- I-Tunes Plattform

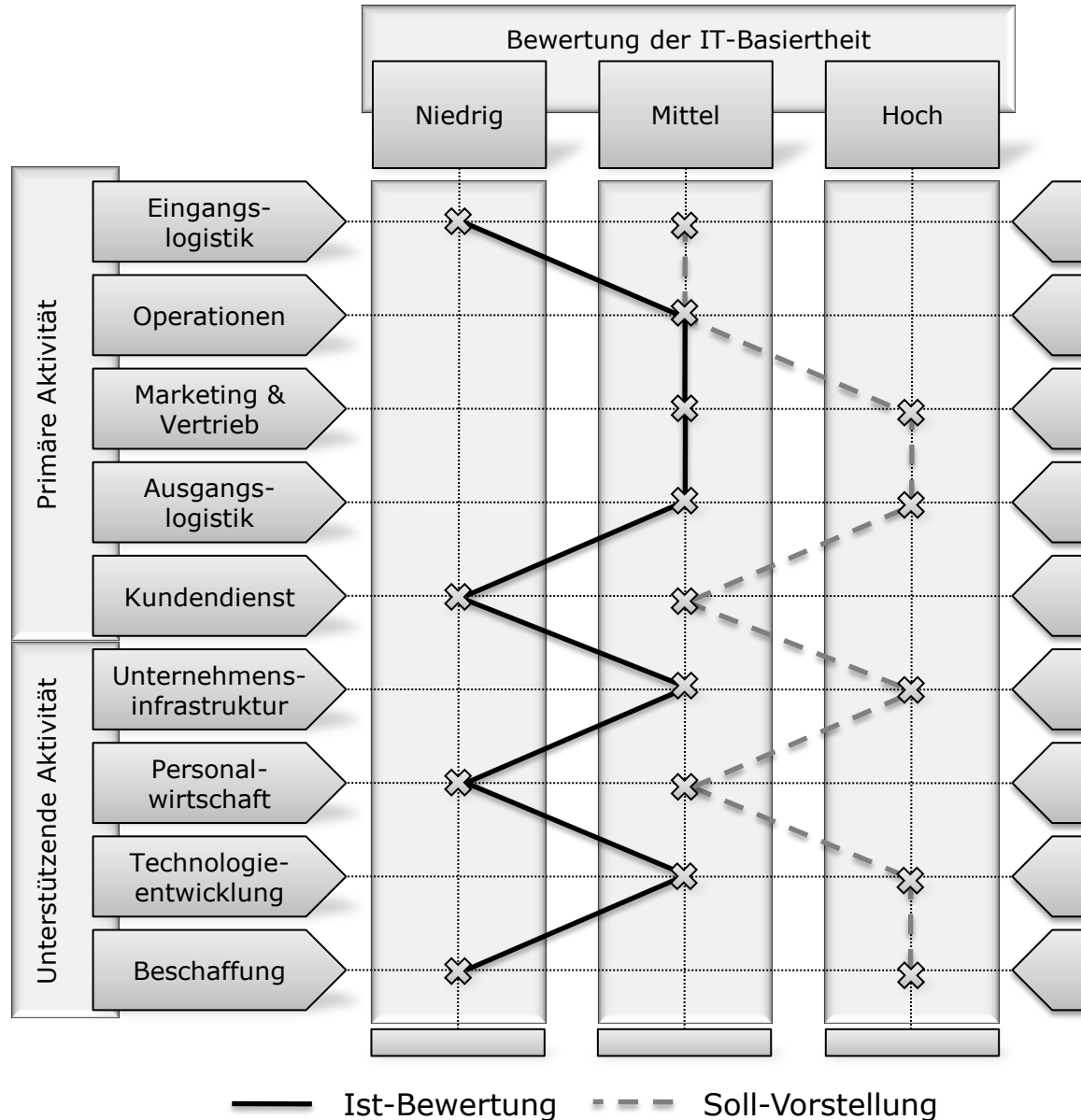


IT-basierte Dienstleistungen: Beispiele

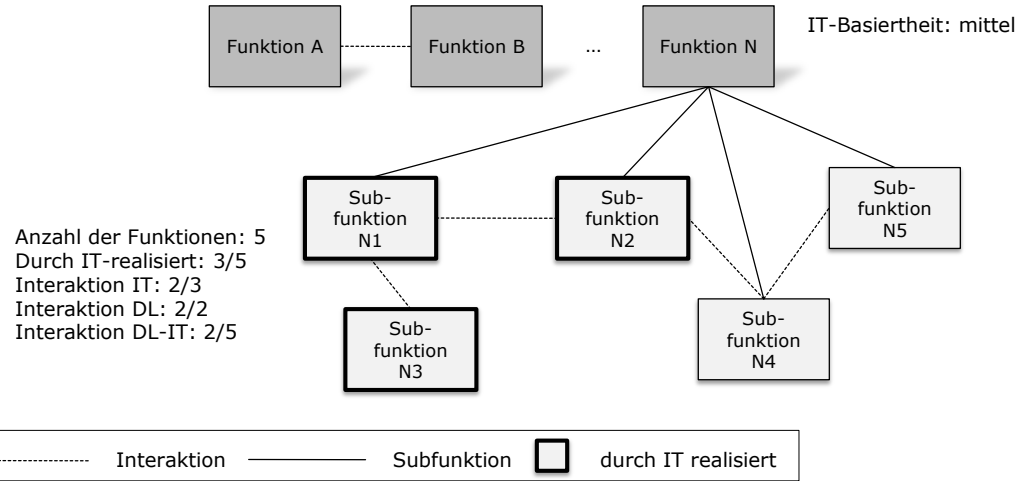
- Klassische Dienstleistungen werden durch IT unterstützt



IT-Basiertheit anhand Porters Value Chain



Grad der IT-Basiertheit

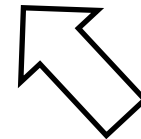
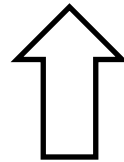
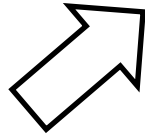
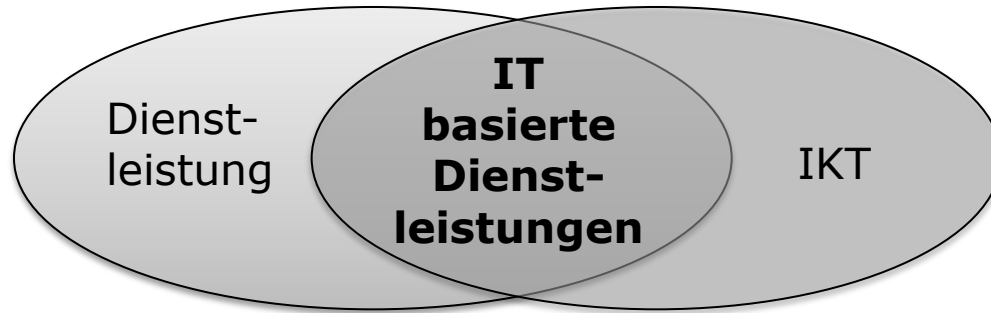


Gewichtung	Funktion	Bewertungsskala			
		B ₁	B ₂	...	B _m
λ ₁	F ₁				
λ ₂	F ₂				
...	...				
λ _n	F _n				

$$G_{IT} = \sum_{x=1}^m \sum_{y=1}^n \lambda_y BF_{xy} \left(\frac{100}{m-1} \right) (m-x).$$

Problem und Lösungskonzeption

Problemdimensionen



Komplexitäts-
beherrschung

Technologie-
einsatz

Domänenwissen

Entwicklungs-
vorgehen

- Notwendig: Entwicklungsgegenstand abgrenzen
- Granularität
- richtiger Grad von Abstraktion

- Automatisierung wo sinnvoll
- Keine Technologie-dominanz

- Einbeziehung des Wissens über die Services

- Kombination von top-down und bottom-up Ansätzen

Lösungsidee



Notwendig ist eine **Methodik für die Entwicklung komponentenorientierter IT-basierter Dienstleistungen**, die ein Software-Services-Co-Design ermöglicht.

Dafür zu erarbeitende Elemente:

- ein konsistentes Systemverständnis des Entwicklungsgegenstandes
- eine geeignete Technik der Modellierung und Notation
- eine Strategie, die durch den Entwicklungsprozess führt

Frage 1: Welche Wechselwirkungen bestehen zwischen Software und Dienstleistungen in Bezug auf IT-basierte Dienstleistungen?

Frage 2: Welche Techniken der Modellierung und Notationen können für die Beschreibung der Entwicklung IT-basierter Dienstleistungen zum Einsatz kommen?

Frage 3: Welche Methodik(en) der Entwicklung (Vorgehensweisen) IT-basierter Dienstleistungen lassen sich unter Berücksichtigung der Komplexität identifizieren und beschreiben?

Zentrale Ergebnisse der Forschung

Zentrales Ergebnis: Co-Design-Methodik

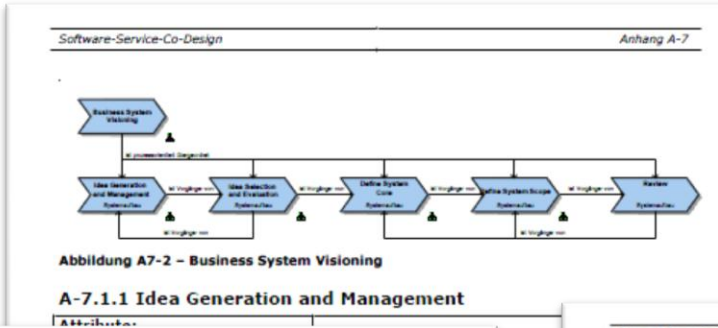


Abbildung A7-2 - Business System Visioning

A-7.1.1 Idea Generation and Management

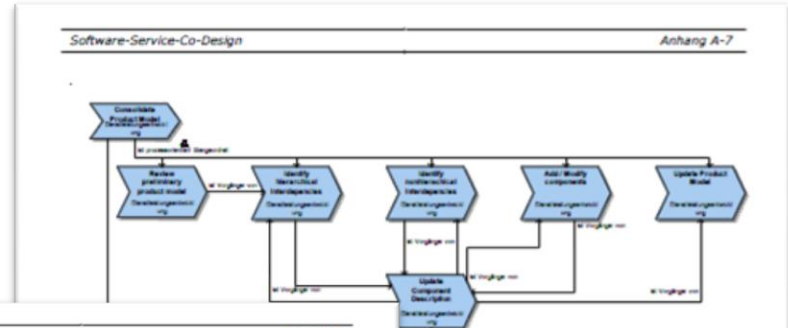


Abbildung A7-3 - Generate and document ideas

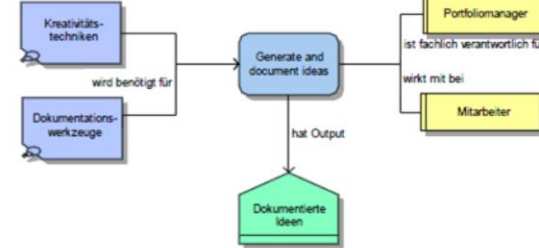


Abbildung A7-3 - Generate and document ideas

A-7.1.2 Idea Selection and Evaluation

Attribute:	
Name	Idea Selection and Evaluation
Ergebnis	Chancen und Risiken neuer Ideen; Machbarkeitsstudie
Voraussetzung	Durchgeführte Sammlung von Ideen (Idea Generation and Management)
Beschreibung/Definition	Aus der Menge der einmal gesammelten Vorschläge für neue Lösungen oder Erweiterungen muss eine möglichst qualifizierte Entscheidung abgeleitet werden, ob die Idee umgesetzt werden soll. Dazu sind die Ideen zu selektieren und zu bewerten, was häufig mit Hilfe sogenannter Machbarkeitsstudien erfolgt. Dabei spielt die strategische Ausrichtung des Unternehmens meist eine entscheidende Rolle.
Ziel	Ziel ist die Auswahl und Bewertung von Ideen, wobei frühzeitig Umsetzungshindernisse identifiziert werden sollen, um kostspielige Fehlentwicklungen zu vermeiden.
Entwicklungslayer	Systemaufbau
Beziehungen:	
Ist Vorgänger von	Define System Core
Folgt auf	Idea Generation and Management
Ist prozessorientiert untergeordnet	Business System Visioning
Ist Vorgänger von	Idea Generation and Management
Hat Input	Portfoliodokumentation
steht unter fachlicher Verantwortung von	Portfoliomanager
Wird bearbeitet unter Mitwirkung von	Evaluator

Software-Service-Co-Design Anhang A-7

A-7 Modellierte Methodik

Nachfolgend wird die mithilfe der erarbeiteten Modellierungskonvention in ARIS umgesetzte Methodik für das Software-Service-Co-Design im Detail dargestellt. Dabei wird sowohl die grafische als auch die textuelle Repräsentation dokumentiert.

Gruppe:	Methodik
Attribute:	
Name:	Software-Service-Co-Design

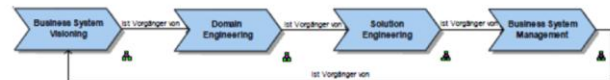
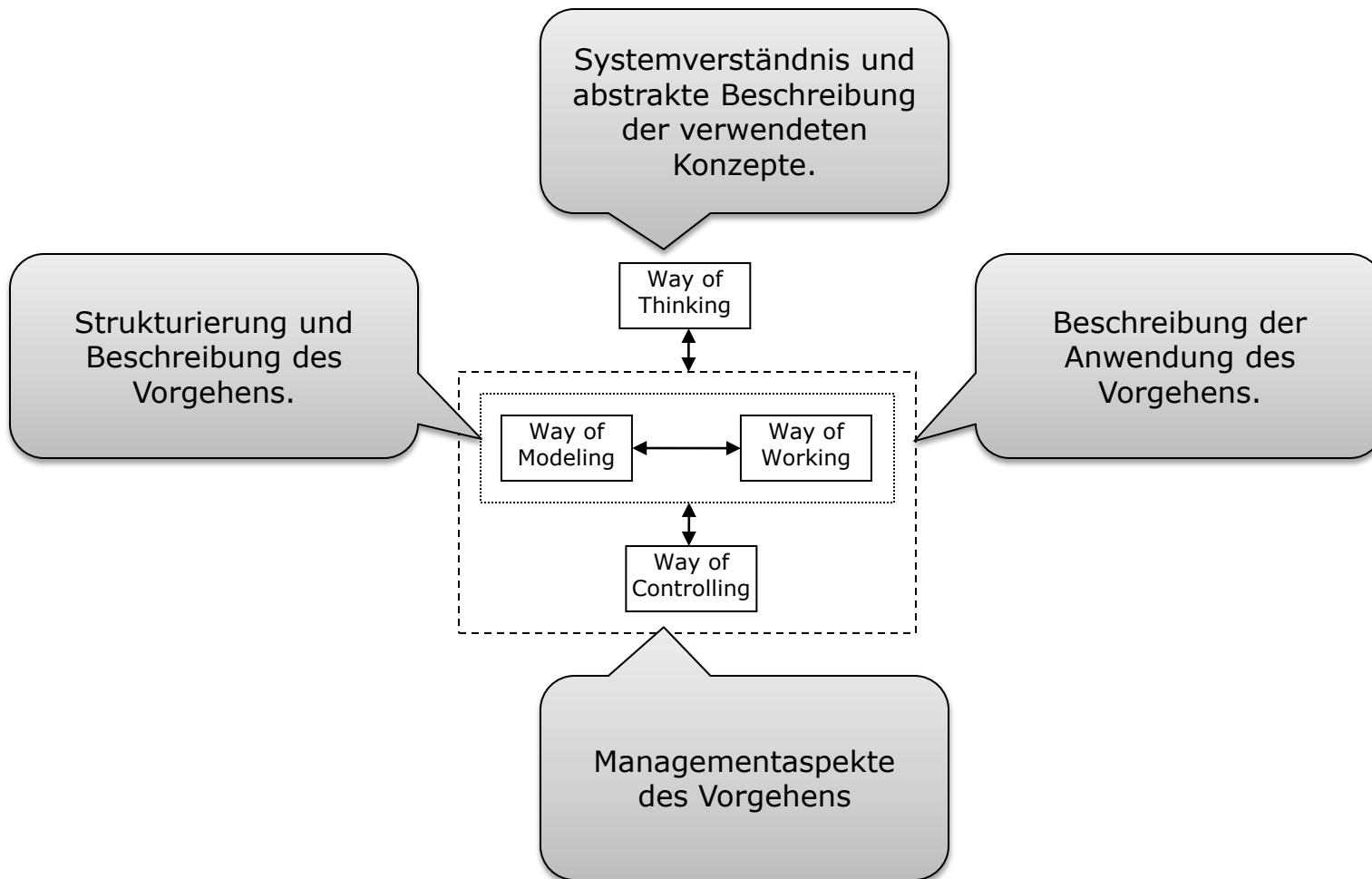


Abbildung A7-1 - Software-Service-Co-Design

A-7.1 Business System Visioning

Attribute:	
Beschreibung/Definition	Diese Phase startet den Prozess und definiert wichtige ökonomische Aspekte. Die Vision eines Portfolios von Leistungselementen ist zu formulieren oder fortzuschreiben und die Zielsetzung der Leistungssysteme im Sinne grundlegender Geschäftsfelder zu erarbeiten. Hierbei kann auf Reifegrade zurückgegriffen werden.
Name	Business System Visioning
Beziehungen:	
Ist Vorgänger von	Domain Engineering
Folgt auf	Business System Management
Hinterlegte Modelle:	
A-7.1 Business System Visioning	Wertschöpfungskettendiagramm
Beziehungen:	
Ist prozessorientiert übergeordnet	Idea Generation and Management
Ist prozessorientiert übergeordnet	Define System Scope
Ist prozessorientiert übergeordnet	Review
Ist prozessorientiert übergeordnet	Define System Core
Ist prozessorientiert übergeordnet	Idea Selection and Evaluation

Framework für Entwicklungsmethodiken



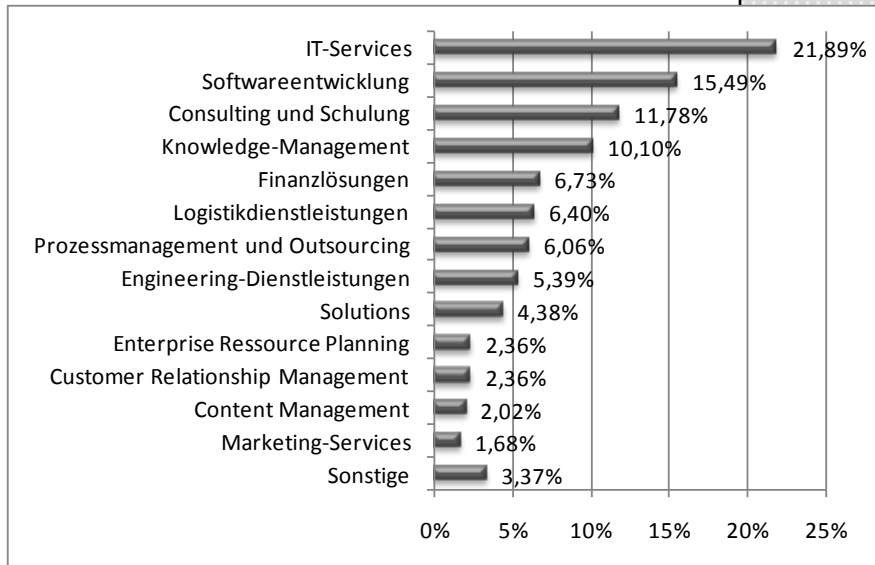
Sol, H.G. (1992). Information systems development: a problem-solving approach. In W. W. Cotterman & J. A. Senn (Ed.), *Challenges and strategies for research in systems development* (pp. 151-161). New York, NY, USA: John Wiley Sons, Inc.

Empirische Untersuchungen

- Anforderungsanalyse und Begriffsklärung

Frage nach der Definition	IT-basierte Dienstleistungen („Was verstehen Sie unter IT-basierten Dienstleistungen?“)		
Durch Auswertung identifizierte Kategorie	IT-begleitende Dienstleistungen	Dienstleistung und IT als hybride Leistung	durch IT unterstützte Dienstleistungen
Anzahl der Nennungen	6	7	8
Beispiele der Antworten der Experten	„Dienstleistungen in Ergänzung zur Software“; „softwarebegleitende Dienstleistungen“; „begleitende Dienstleistungen zu IT“	„Lösungen im gesamten IT-Spektrum“; „Bündel aus Informationstechnik und Dienstleistungen“; „Bereitstellung von Solutions“	„Dienstleistungen, die eine IT-Realisierung zur Folge haben“; „Dienstleistungen, deren Prozesse durch IT unterstützt werden“; „Leistungserbringung erfolgt unter Zuhilfenahme von IT-Systemen“

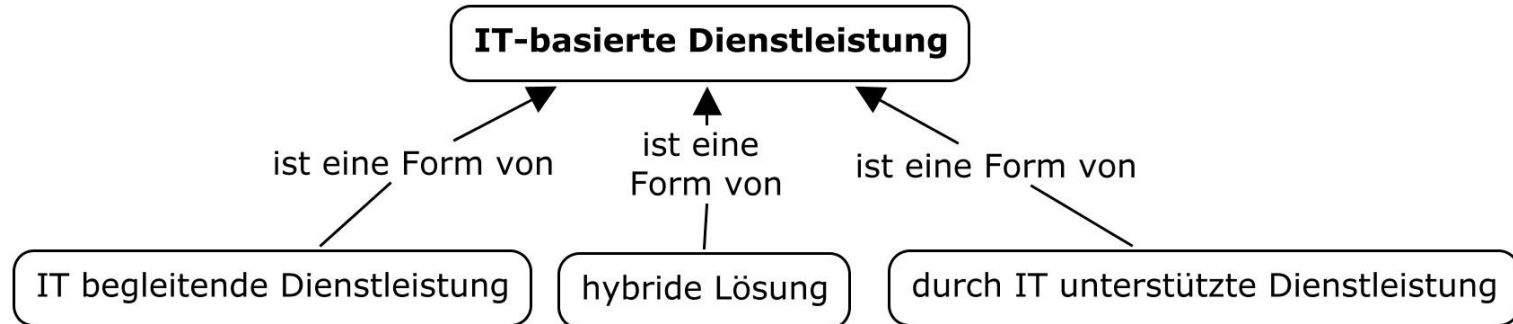
Breitenerhebung



Experteninterviews

Ausprägungen IT-basierter Dienstleistungen

- Erarbeitung einer Definition IT-basierter Dienstleistungen

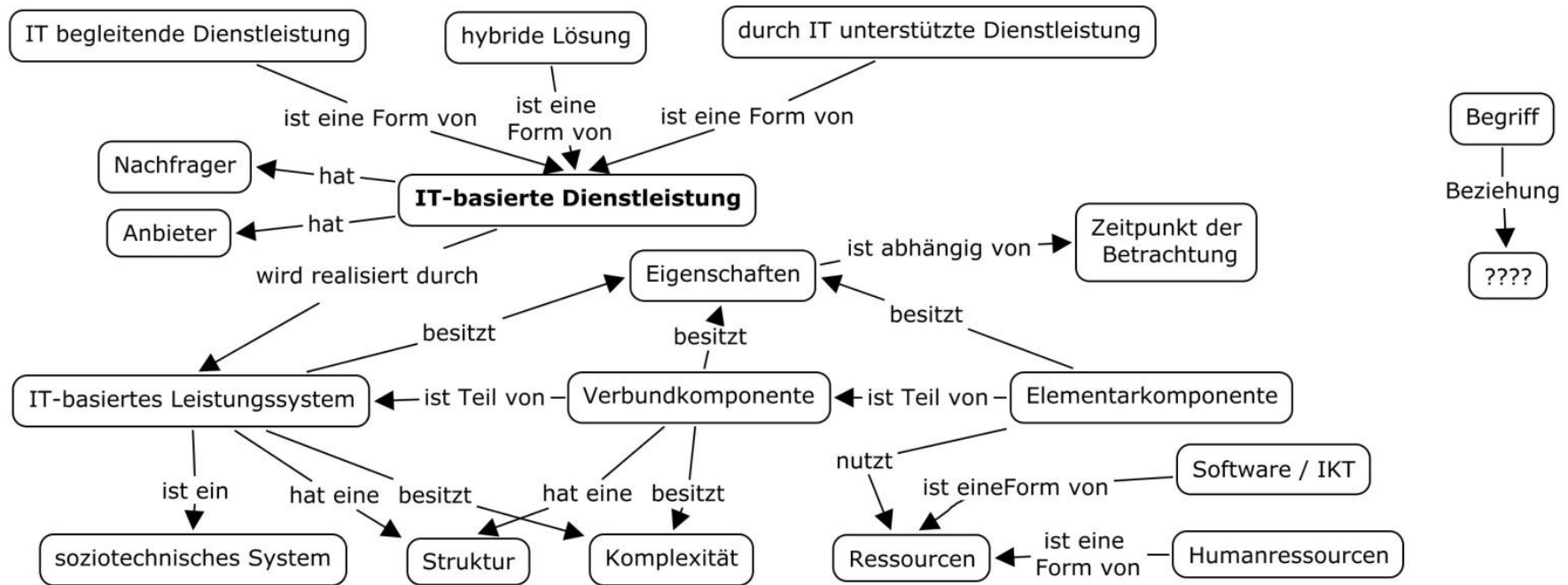


Bsp.: Software mit Hotline Bsp.: Fernwartung Bsp.: Schulung mit Online-Tutorial

IT-basierte Dienstleistungen sind Lösungen, deren Nutzen für den Kunden zu einem maßgeblichen Teil durch den Einsatz von Informations- und Kommunikations-Technologien (IKT) unter Verwendung von vernetzter Software entsteht. Sie treten auf in Form von Dienstleistungen, deren effiziente Gesamterbringung nur durch den Einsatz von IKT gewährleistet werden kann, als begleitende Dienstleistungen zu Produkten der IKT sowie als hybride Lösungen bestehend aus Dienstleistungen und IKT-Produkten.

Beziehungen im soziotechnischen System

- Beschreibung grundlegender Beziehungen und Konzepte für IT-basierte Dienstleistungen als Concept Map



Matrix für das Software-Service-Co-Design

- Komplexität als Unterscheidungskriterium

Komplexität der Software (Art und Anzahl der Teilelemente, Interaktionskomplexität)	hoch	Komplexes Softwareprodukt <i>Software Engineering</i>	IT-begleitende Dienstleistung <i>Software-Service-Co-Design</i>	Komplexe hybride Leistung <i>Software-Service-Co-Design</i>
	mittel	Softwareprodukt	Integrierte hybride Dienstleistung	Durch IT unterstützte Dienstleistung
	niedrig	<i>Kein Engineering</i> Trivialleistung	<i>Service Engineering</i> Dienstleistungsprodukt	
		niedrig	mittel	hoch

Komplexität der Dienstleistung

(Art und Anzahl der Teilelemente,
Interaktionskomplexität)

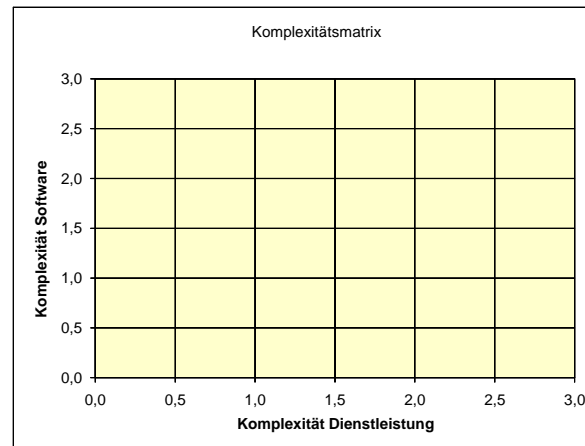
Erfassung der Komplexität

Komplexitätsmatrix

Dieses Tool dient der Einordnung von Dienstleistungs-/Software-Bündeln in einer zweidimensionalen Matrix.
Als Maß wird die Komplexität angesetzt.
Die Skala reicht von 0 "sehr niedrig" bis 3 "sehr hoch" (ganzzahlig)
Bitte alle grünen Felder ausfüllen

Bezeichnung der IT-basierten Dienstleistung Nr. 1

Nr.	Frage	Bewertung (0=stimme nicht zu, 3=stimme voll zu)	Gewicht	Wert
Dienstleistungskomplexität				
DL_1	Das Unternehmen versteht sich als Dienstleister.		0,5	0
DL_2	Der größte Anteil der Aufgaben des Unternehmens liegt in kundennahen Tätigkeiten.		0,5	0
DL_3	Der Innovationsgrad der neuen Dienstleistungen ist sehr hoch.		0,5	0
DL_4	Die für die Dienstleistungsentwicklung benötigte Personalkapazität ist sehr hoch.		0,5	0
DL_5	Es werden Dienstleistungen entwickelt, die aus sehr vielen Teilleistungen bestehen.		0,5	0
DL_6	Die einzelnen Dienstleistungen müssen sehr detailliert ausgearbeitet und beschrieben werden.		0,5	0
			Summe DL	0,00
Softwarekomplexität				
SW_1	Das Unternehmen versteht sich als Software-Anbieter.		0,5	0
SW_2	Der größte Anteil der Aufgaben liegt in der Software-Entwicklung.		0,5	0
SW_3	Der Innovationsgrad der neuen Software ist sehr hoch.		0,5	0
SW_4	Die für die Softwareentwicklung benötigte Personalkapazität ist sehr hoch.		0,5	0
SW_5	Es wird Software entwickelt, die aus sehr vielen Modulen besteht.		0,5	0
SW_6	Die Software-Module zeichnen sich durch viele Abhängigkeiten untereinander aus.		0,5	0
			Summe SW	0,00
Interaktionskomplexität				
INT_1	Der Abstimmungsbedarf zwischen Dienstleistungs- und Softwareentwicklung ist sehr hoch.		0,25	0
INT_2	Viele Aufgaben werden von Dienstleistungs- und Softwareentwicklern gemeinsam durchgeführt.		0,25	0
INT_3	Es werden regelmäßige Reviews mit Software- und Dienstleistungsverantwortlichen durchgeführt.		0,25	0
INT_4	Software- und Dienstleistungsentwicklung sind ein gemeinsamer Prozess.		0,25	0
			Summe INT	0,00



Erarbeitung von Anforderungen

- Anforderungen an die Methodik zur Entwicklung IT-basierter Dienstleistungen

Systemverständnis und abstrakte Beschreibung der verwendeten Konzepte.

Beschreibung der Anwendung des Vorgehens.

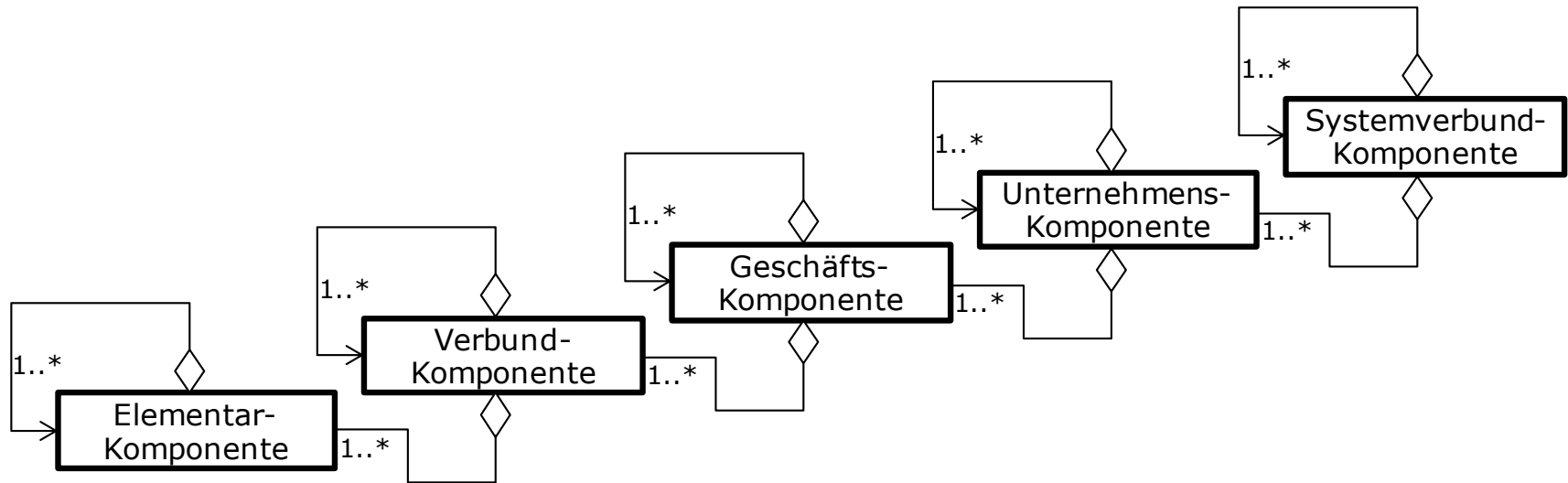
Managementaspekte des Vorgehens

Strukturierung und Beschreibung des Vorgehens.

Quelle der Ableitung	Typologisierung	Experteninterviews	Literaturanalyse	existierende Empirie	eigene Empirie
Anforderung					
Way of Thinking					
Komposition von Teilleistungen	x			x	x
kontinuierliche Erfassung von Anforderungen			x		
Komponentenkonzept	x		x	x	x
Way of Working					
Lebenzyklus				x	x
Aufbau und Erweiterung von Systemen	x	x			
komponentenorientierter Ansatz	x		x	x	x
Flexibilität und Anpassbarkeit		x		x	x
implementationsunabhängig				x	
Way of Controlling					
Unterscheidung des Typs	x	x			
Management und Fortschrittskontrolle		x	x	x	x
Rücksprünge und Iterationen					x
Rolle von Ressourcen	x				x
Way of Modelling					
klare Struktur der Methodik			x	x	x
grafische Repräsentation					x
maschinenlesbare Repräsentation					x

Identifikation von Granularitätsstufen

- Teil des Komponentenkonzeptes



- niedrigste Ebene der Abstraktion
- nicht sinnvoll weiter zu zergliedern

- feingranulare Leistungen
- nicht als solche vermarktbar
- IT Komponenten oder humane Leistung

- kohärente Sammlung von Funktionalität
- vermarktbare Leistung
- Verarbeitungsdienste

- komponentenorientiertes Leistungssystem
- Lösung im Rahmen eines Geschäftsproblems

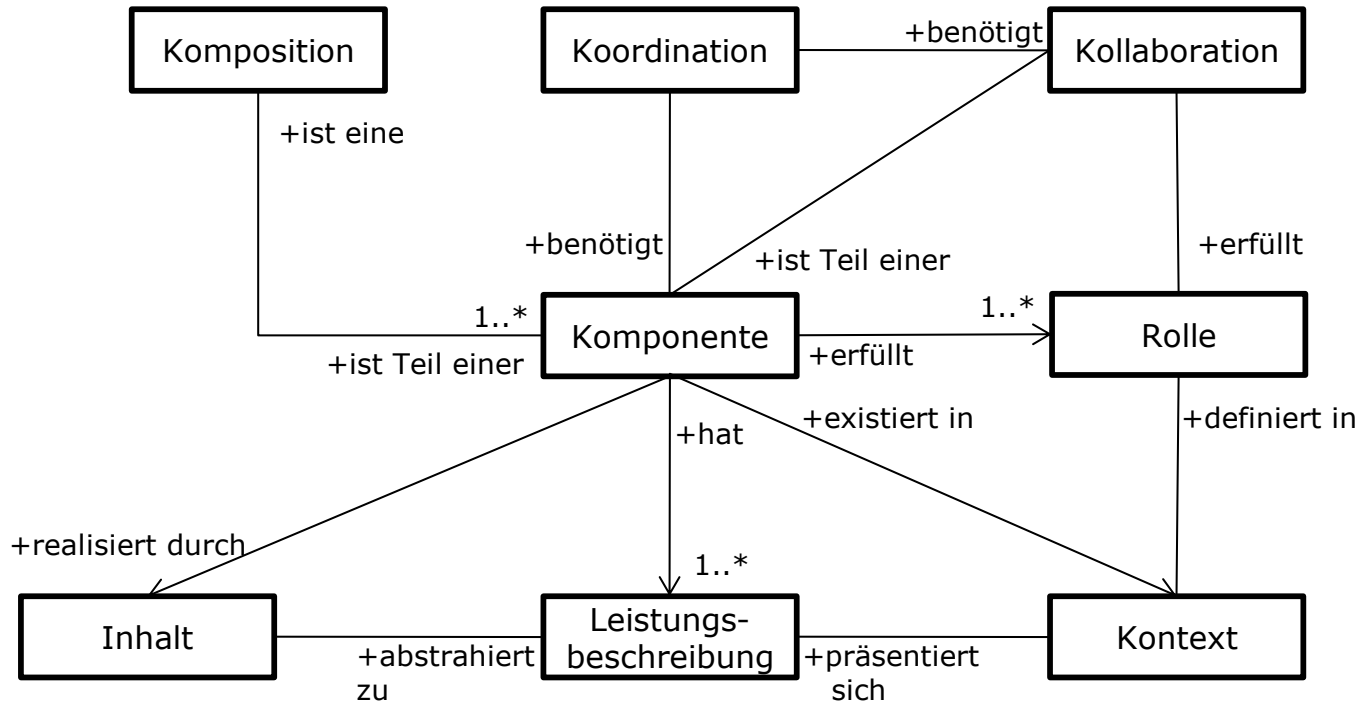
- Zusammenfassung von Unternehmenskomponenten
- Ziel der Kooperation zwischen Leistungssystemen

Erarbeitetes Komponentenkonzept

-Kompositions-
funktion für
Verbundkomponenten

-Zeitliche Ablauf
der Interaktion

-Interaktion zwischen
Komponenten



- interne Realisierung

- Spezifikation des
Verhaltens

- Problembereich außerhalb
der Komponente

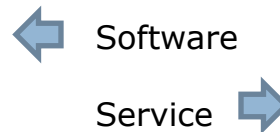
Unterschiedliche Entwicklungsansätze

Systemverständnis (way of thinking):
Die Anforderungen an Software lassen sich eindeutig beschreiben und festlegen. Die Entwicklung eines Softwaresystems ist geprägt durch die initiale Erfassung dieser Anforderungen und deren direkte Umsetzung.

Entwicklungsschema (way of working)	Generisches Modell (way of controlling)	Strukturierung (way of modeling)
Aktivitätsorientiert (Softwareentwicklung als eine Kette von Aktivitäten, die sequenziell abgearbeitet wird)	Phasenmodell (Benington 1956, Sommerville 2007), Wasserfallmodell (Royce 1970)	Strukturiert die Phasen Analyse, Entwurf, Implementierung, Integration, Installation und Einsatz
	V-Modell 97 bzw. V-Modell XT (IABG und Bundesrepublik Deutschland 2007)	Strukturiert Definition, Entwurf, Modulimplementierung mit zugeordneten Modultests, Integrationstests und Systemtests. Das V-Modell XT stellt eine Flexibilisierung dar.
Iterativ (Softwareentwicklung als eine Sequenz von Entwicklungszyklen)	Iteratives Verbesserungsmodell (Basili und Tuner 1975)	Die Phasen Analyse, Entwurf, Implementierung und Einsatz werden als Inkremente verstanden und wiederholt durchlaufen.
	Spiralmodell (Boehm 1988)	Arbeitet mit Quadranten, die wiederholt durchlaufen werden: Zielfestlegung, Alternativen bestimmen und Risiken festlegen, Entwicklung und Überprüfung und Planung der nächsten Phase.

Die Anforderungen an Software sind meist unklar und nur schwer initial zu erfassen. Die Entwicklung eines Softwaresystems geht mit einer kontinuierlichen Schärfung der Anforderungen einher.

Evolutionär (Softwareentwicklung als rückbezoglicher Prozess, d. h. durch den Einsatz von Software entstehen neue Anforderungen, welche zu neuen Versionen der Anwendung führen)	Evolutionäres Modell (Balzert 2001)	Über Kernanforderungen wird eine Nullversion erstellt, die durch ergänzende Anforderungen weiterentwickelt wird.
	Prototypenmodell (Prototyping) (Balzert 2001)	Art des Prototyps wird geklärt und dieser dient als Ausgangspunkt für die inkrementelle Weiterentwicklung.
Agil (Softwareentwicklung durch	Extreme Programming	Unterscheidet zwischen



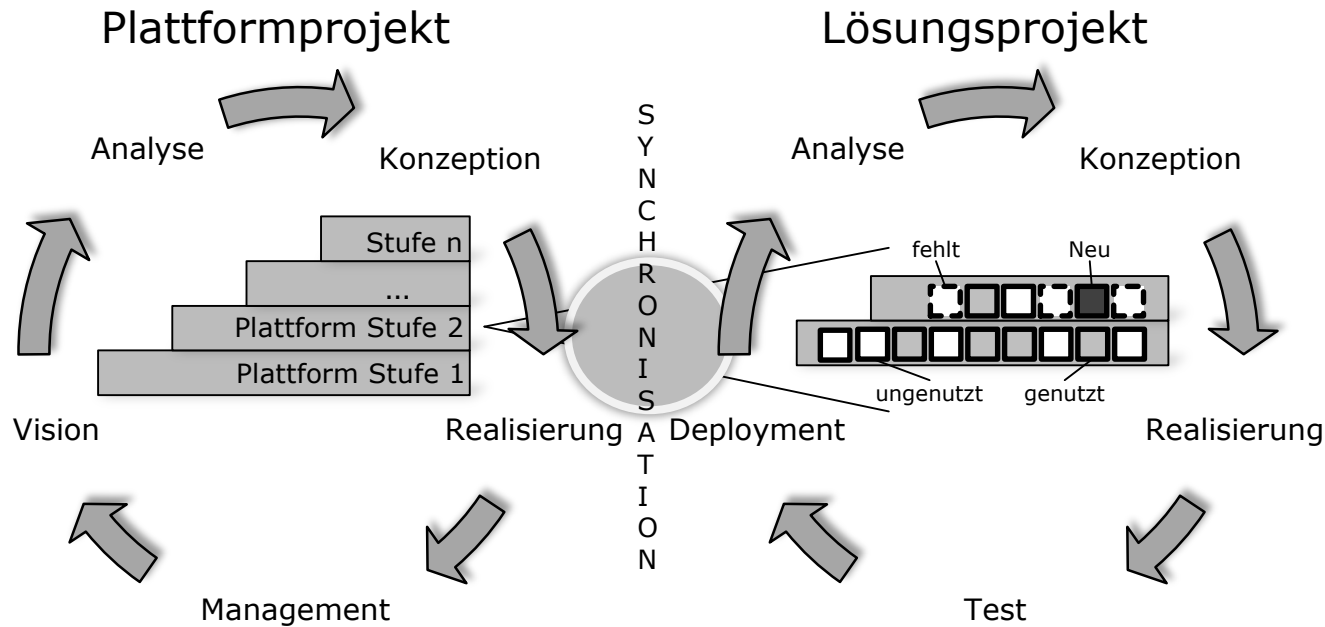
Systemverständnis (way of thinking):
Die Anforderungen an Dienstleistungen lassen sich eindeutig beschreiben und festlegen. Die Entwicklung ist geprägt durch die initiale Erfassung dieser Anforderungen und deren direkte Umsetzung.

Entwicklungsschema (way of working)	Generisches Modell (way of controlling)	Strukturierung (way of modeling)
Aktivitätsorientiert (Dienstleistungsentwicklung als eine Kette von Aktivitäten, die sequenziell abgearbeitet wird)	Modell von Edvardsson und Olsson (Edvardsson und Olsson 1996)	Segmentiert in Service Concept/Service Process/Service System.
	Modell nach Scheuing und Johnson (Scheuing und Johnson 1989)	Zerfällt in vier Stufen (Direction, Design, Testing, Introduction) mit 15 Phasen.
	Modell nach Jaschinski (Jaschinski 1998)	Zerfällt in 3 Hauptphasen (Definieren, Konzipieren, Umsetzen).
	Modell nach DIN Fachbericht 75 (DIN e.V. 1998)	Strukturiert die Phasen Ideenfindung, Anforderungen, Design, Einführung, Erbringung und Auflösung.
Iterativ (Dienstleistungsentwicklung als eine Sequenz von Entwicklungszyklen)	Modell nach Ramaswamy (Ramaswamy 1996)	Segmentiert Service Design und Service Management mit jeweils 4 Phasen. Die letzte Phase kann als Input für eine neue Sequenz dienen.
	Modell nach Shostack (Kingman-Brundage und Shostack 1991)	Unterscheidet zwischen den Phasen Design, Implementation, Documentation, Introduction. Anforderungen werden iterativ initial erfasst.
	Modell nach FHG IAO (Meiren und Barth 2002)	Strukturiert die Phasen Definition, Anforderungsanalyse, Konzeption/Modellierung, Implementierung und Markteinführung mit fakultativ anwendbaren Unterphasen. Konzeption und Realisierung werden iterativ durchgeführt.

Die Anforderungen an Dienstleistungen sind beständiger Veränderung unterworfen und lassen sich zu keinem Zeitpunkt vollständig erfassen. Die Entwicklung ist geprägt durch eine kontinuierliche Durchführung von Anpassungen.

Evolutionär (Dienstleistungsentwicklung als rückbezoglicher Prozess, d. h. bei der Erbringung von Dienstleistungen entstehen neue Anforderungen,	Pro-Services Referenzmodell (Kunau et al. 2005)	Strukturiert multiperspektivisch die Entwicklung mithilfe der Generierung von Sichten. Unterschieden wird zwischen Service Engineering, Service
---	---	---

Systemverständnis für das Co-Design



- langfristige Entwicklung der Plattform
- definiert Entwicklungsstufen

- kurzfristige Kundenprojekt
- Nutzung und Erweiterung der Plattform

Literaturanalyse zu Entwicklungsmethodiken

- Analyse existierender Entwicklungsansätze

Modul / Modell \ Ursprung	Idea Generation	Idea Selection	Definition	Re-quirements	Conception	Realization	Test	De-ployment	Usage	Phase-Out
<i>Aktivitätsorientierte Modelle</i>										
Wasserfallmodell (SW)			Definition	Analyse	Entwurf	Imple-mentierung	Tests	Installation	Einsatz	
Phasenmodell (SW)			Operational Plan	Program Specifications	Coding Specifications	Coding	Testing	Shakedown	System Evaluation	
Edvardsson/ Olsson (DL)			Service Concept Development		Service System Development					
					Service Process Development					
Scheuing/ Johnson (DL)	Formulation of new Service Objectives & Strategies									
	Idea Generation	Idea Screening	Concept Developing							
			Concept Testing	Business Analysis						
				Projekt Authorisation	Service Design&Testing	Process& System Design&Testing				
						Marketing Program Design&Testing				
						Personnel Training	Service Testing & Pilot Run			
							Test Marketing	Full-Scale Launch		
								Post-Launch Rewiew		

Untersuchung zur Modellierung

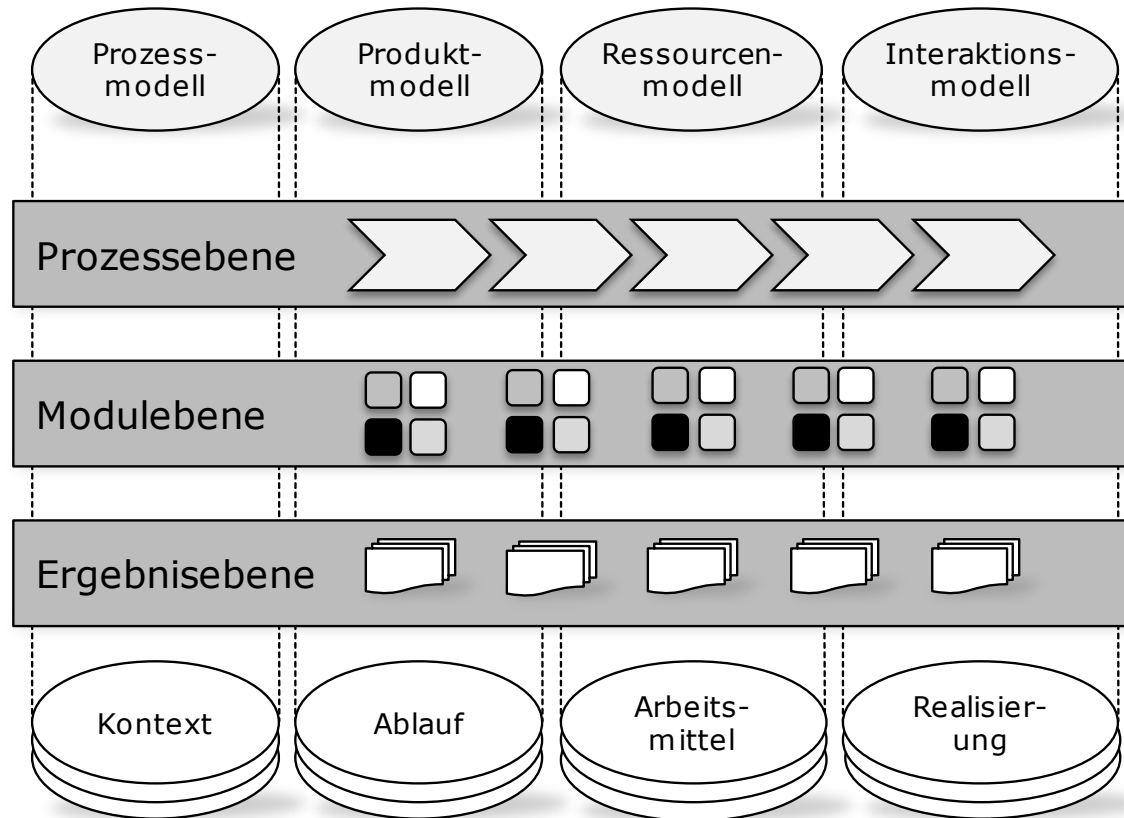
- Wahl der ARIS Scouts als Modellierungsansatz

		ARIS EPK	Service Blueprints	ARIS Scout	BPMN	UML
Formal	Korrektheit	◐	◐	◐	◐	◐
	Vollständigkeit	◐	◐	◐	◐	◐
	Einheitlichkeit	●	◐	◐	○	○
	Wiederverwendung/ Hierarchiebildung	●	○	●	◐	●
	Methodische Analyse	◐	◐	◐	◐	◐
Anwen- der	Einfachheit	●	●	●	●	◐
	Anwendbarkeit	◐	◐	●	◐	◐
	Verständlichkeit	◐	◐	●	◐	◐
	Anschaulichkeit	◐	◐	●	◐	◐
Anwendung	Erweiterbarkeit	○	○	◐	◐	◐
	Ausdrucksmächtigkeit	●	◐	●	●	●
	Operationalisierbarkeit	◐	○	◐	◐	●
	Iteration/ Rekursion	●	○	●	◐	●

Legende: Kriterium ○ - nicht erfüllt, ◐ - teilweise erfüllt, ● - erfüllt

Ebenen des Entwicklungsvorgehens

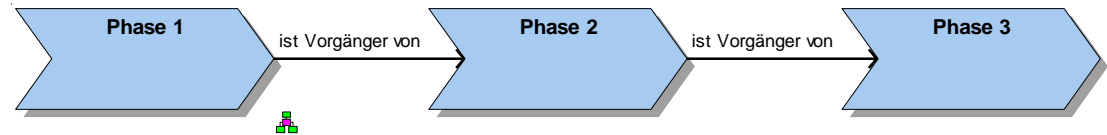
- Umsetzung des Systemverständnisses und des Komponentenkonzeptes in ein Entwicklungsvorgehen für ein Software-Service-Co-Design



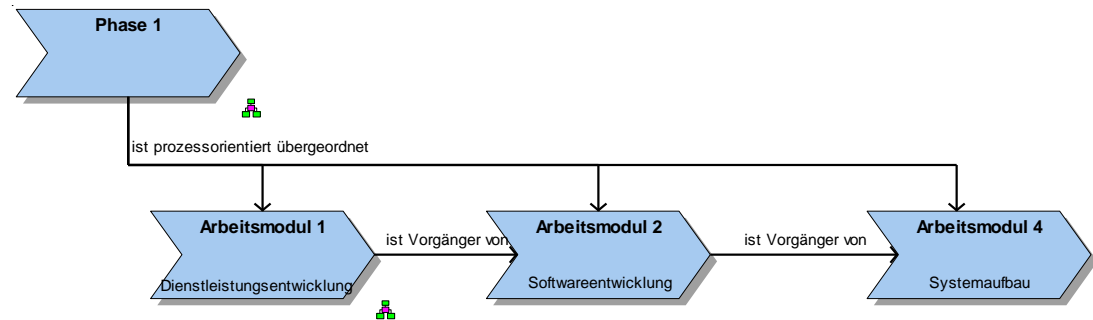
Modellierungskonventionen

- Modellierung mit Hilfe von ARIS

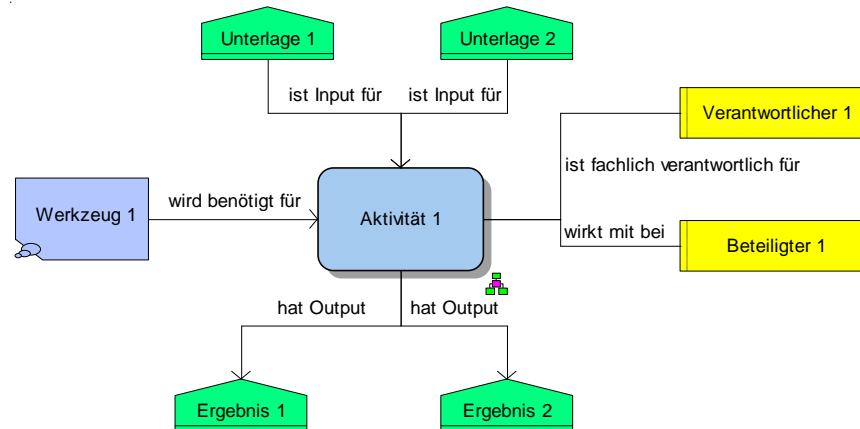
- Phasen



- Arbeitsmodule



- Aktivitäten

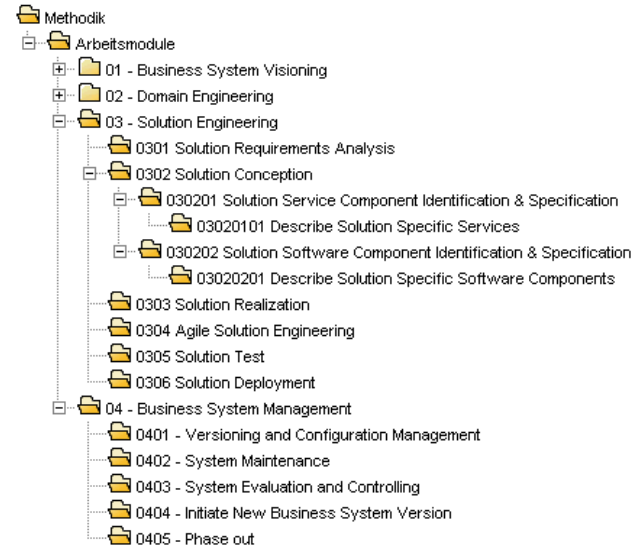
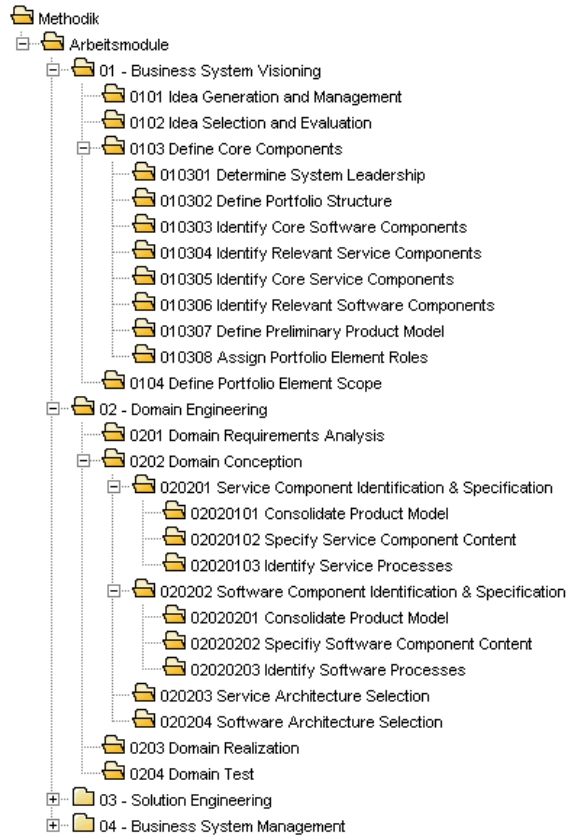


Attribute von modellierten Objekten

- Modellierung mit Hilfe von ARIS
 - Nutzung von Attributen zur Auszeichnung der einzelnen Objekte der Methodik

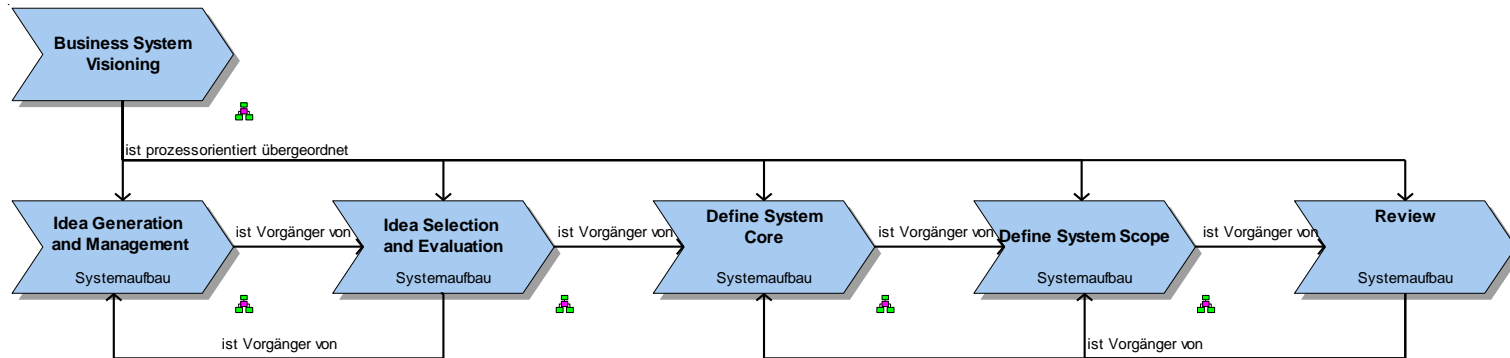
Attribut	Beispiel
Name des Objektes	<i>Konzeption Dienstleistungsprozess</i>
Zuordnung zum Entwicklungslayer	<i>Dienstleistungsentwicklung</i>
Beschreibung (detaillierte textuelle Beschreibung)	<i>[textuelle Beschreibung des Vorgehens]</i>
Voraussetzungen	<i>Ideenbewertung durchgeführt</i>
Ergebnisse	<i>Dokumentiertes Prozessmodell</i>
Ziel	<i>Erarbeitung Konzept der DL-Prozesse</i>
Werkzeuge	<i><serviceblueprinting></i>

Übersicht über die Methodik

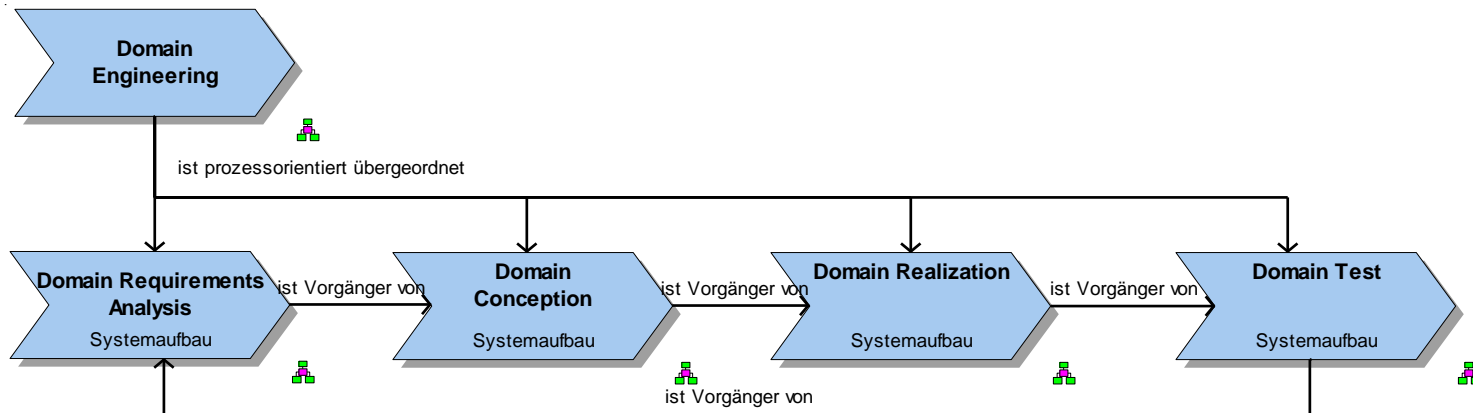


Kurzeinblick in die Methodik (1)

- Business System Visioning:
 - Ideenfindung, Bewertung und Strategieentwicklung

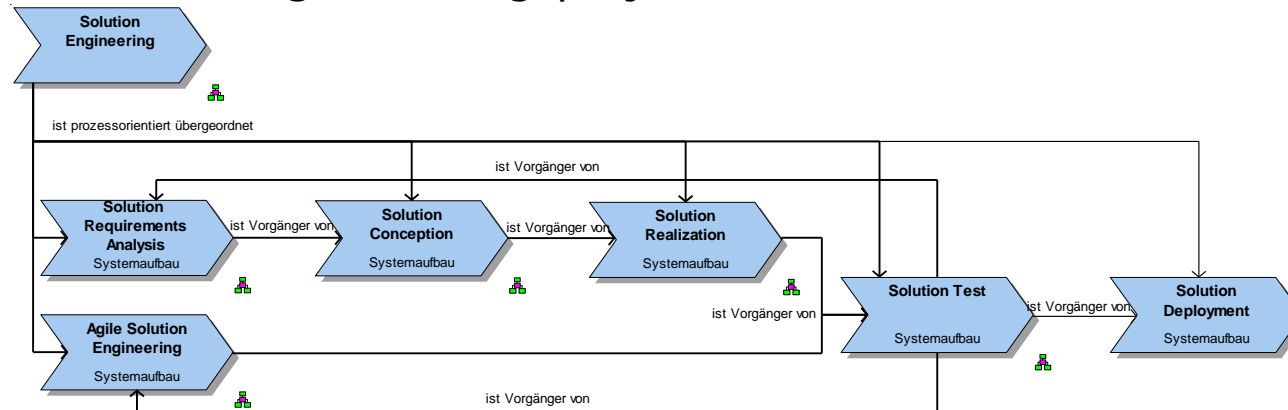


- Domain Engineering:
 - Aufbau der Leistungsplattform

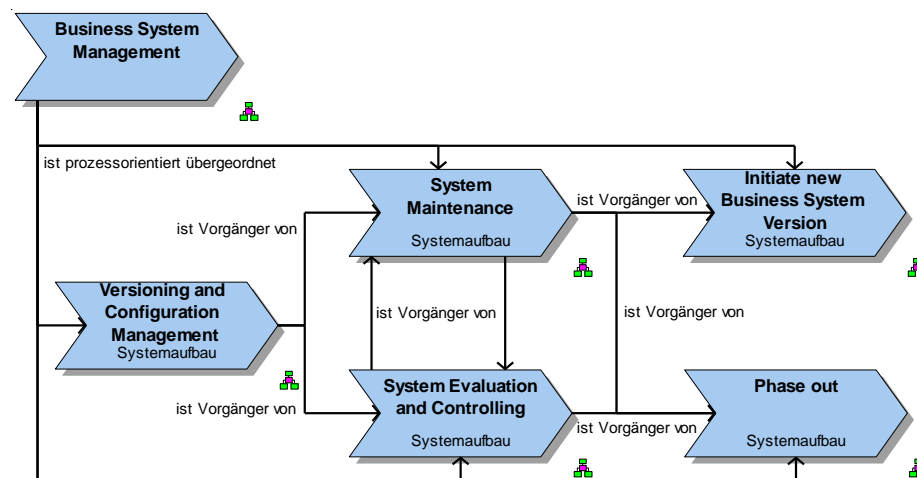


Kurzeinblick in die Methodik (2)

- Solution Engineering:
 - Umsetzung in Lösungsprojekten

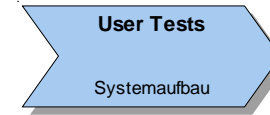
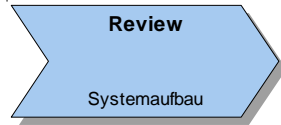


- Business System Management:
 - Betrieb, Variantenmanagement und Auslaufen

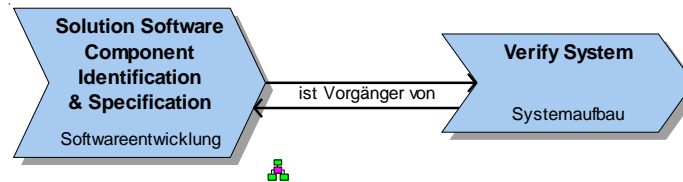


Wichtige Hilfsmittel

- Kontrollelemente (Tests Reviews)

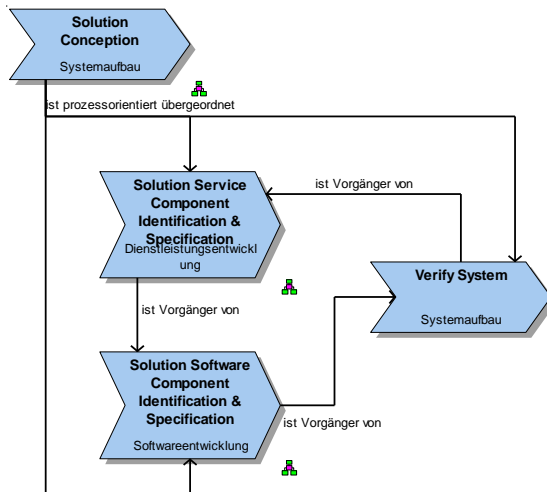


- Iterationsmöglichkeiten

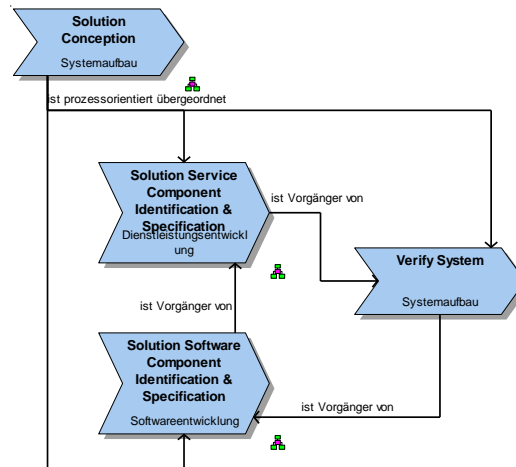


- Unterscheidung des Typs IT-basierter Dienstleistungen

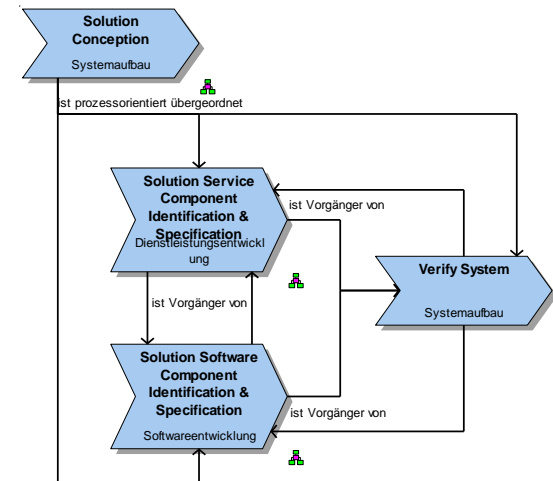
IT-begleitende Dienstleistung



durch IT unterstützte Dienstleistung

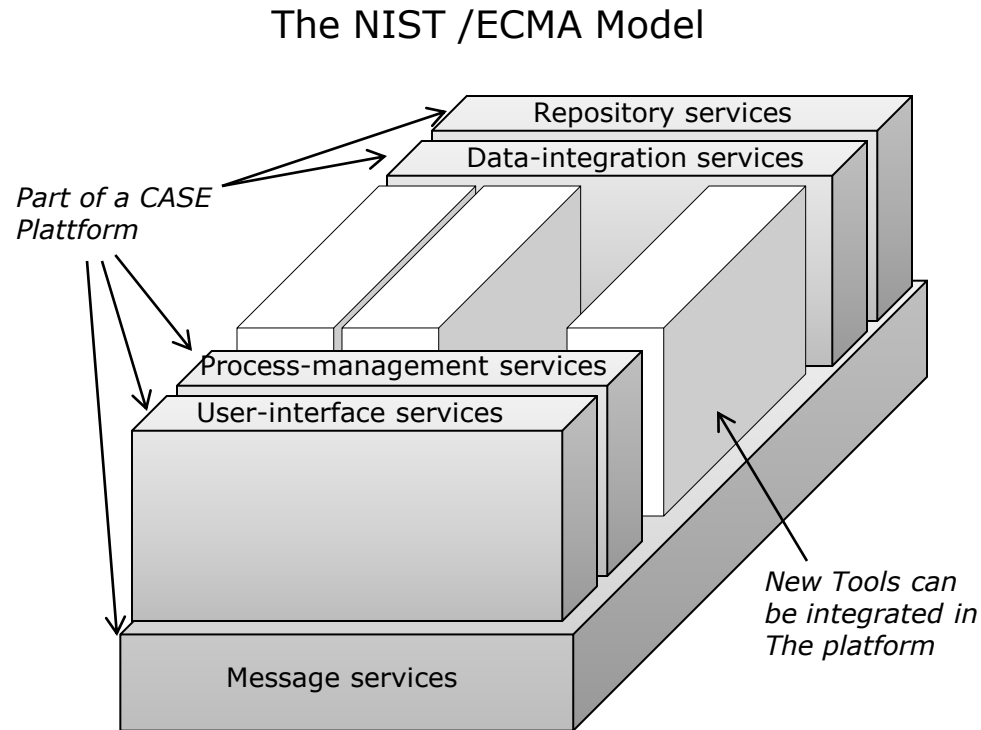


hybride Leistung



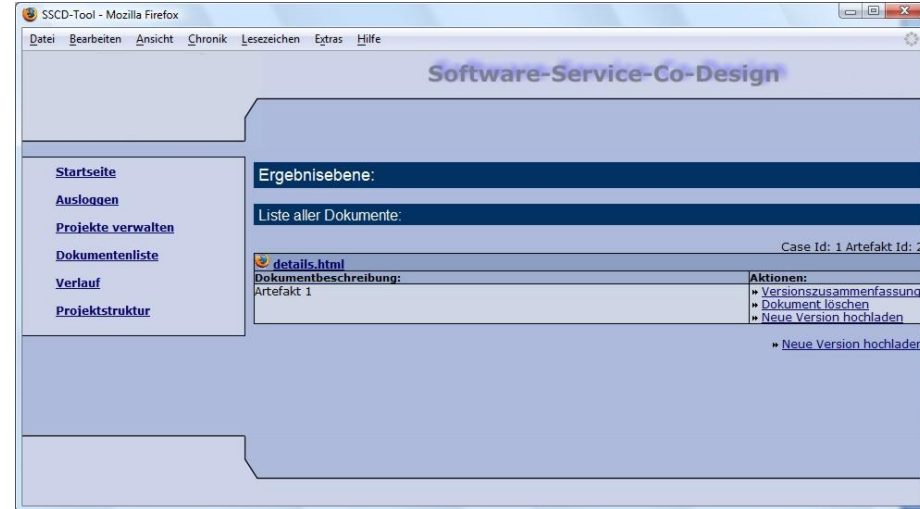
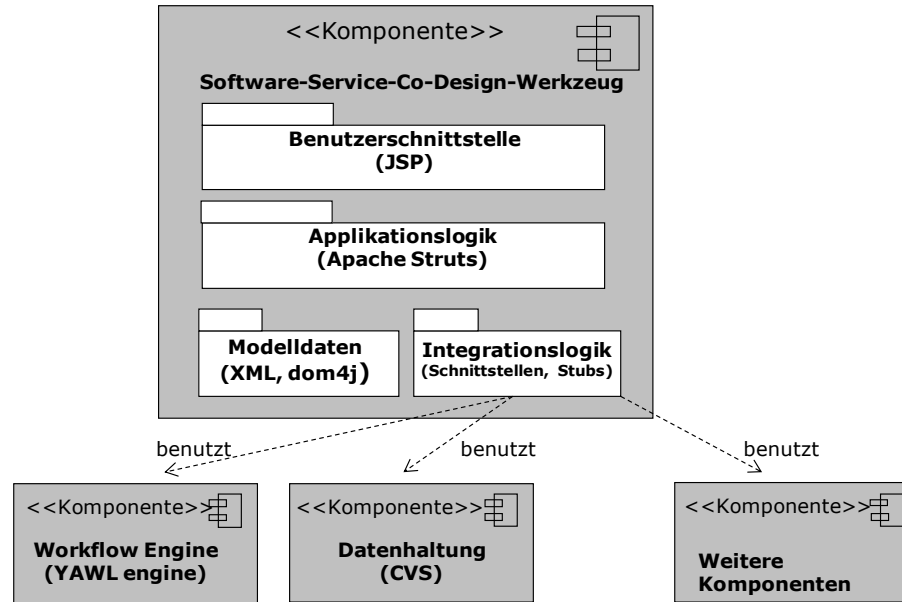
Werkzeugunterstützung für die Methodik

- Tool Support im Sinne eines „Computer Aided Service Engineerings“



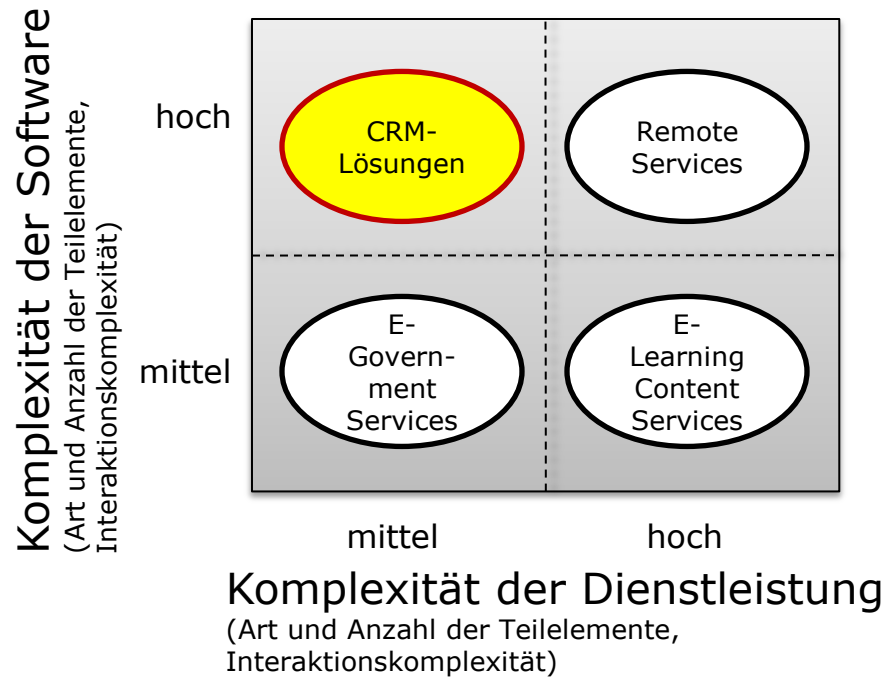
NIST, National Institute of Standards and Technology; ECMA, European Computer Manufacturers Association (1993): *Reference Model for Frameworks of Software Engineering Environments*. Gaitersburg: NIST / ECMA, 1993.

Co-Design-Tool: Prototyp als Demonstrator



Anwendung und Evaluation

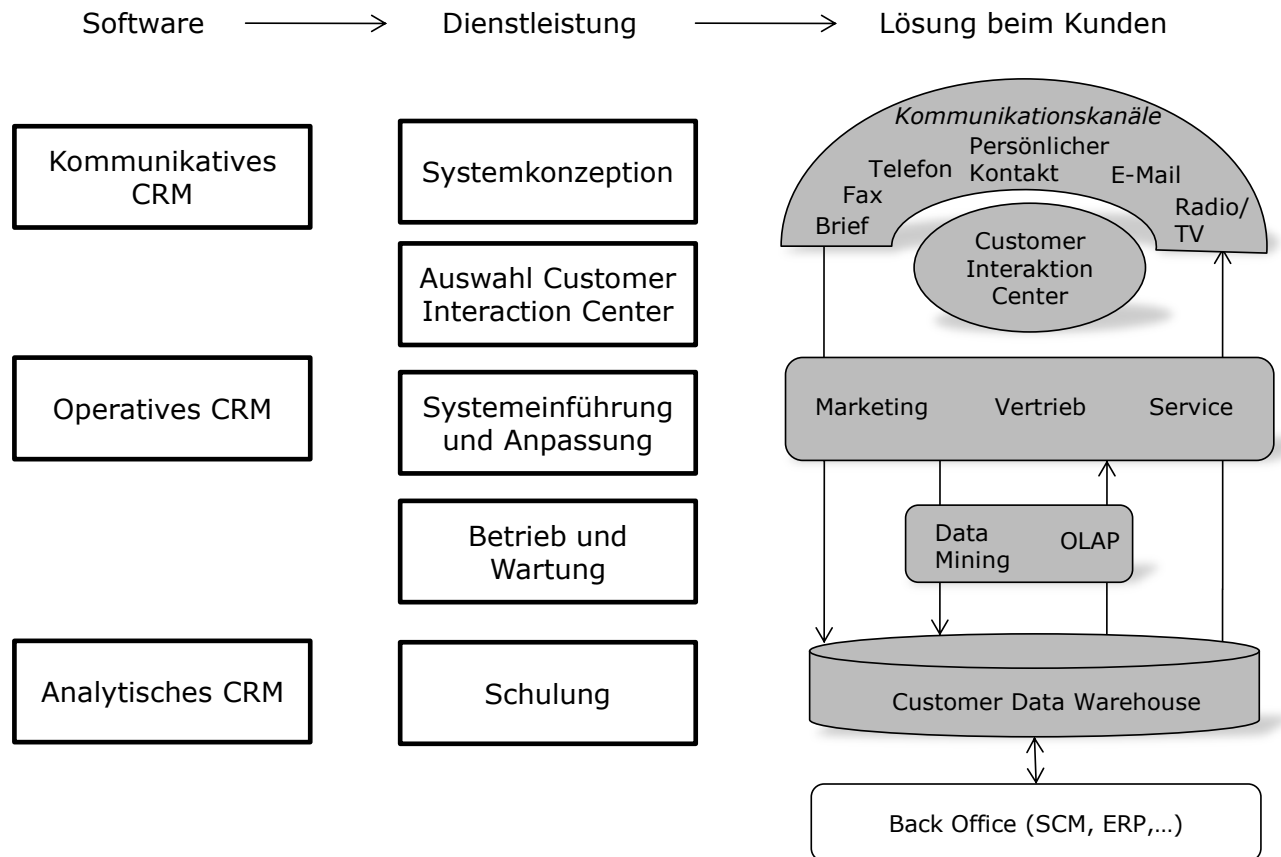
- Evaluation des Ansatzes über die Durchführung von Fallstudien



Fallbeispiel CRM Lösung: Ideenfindung



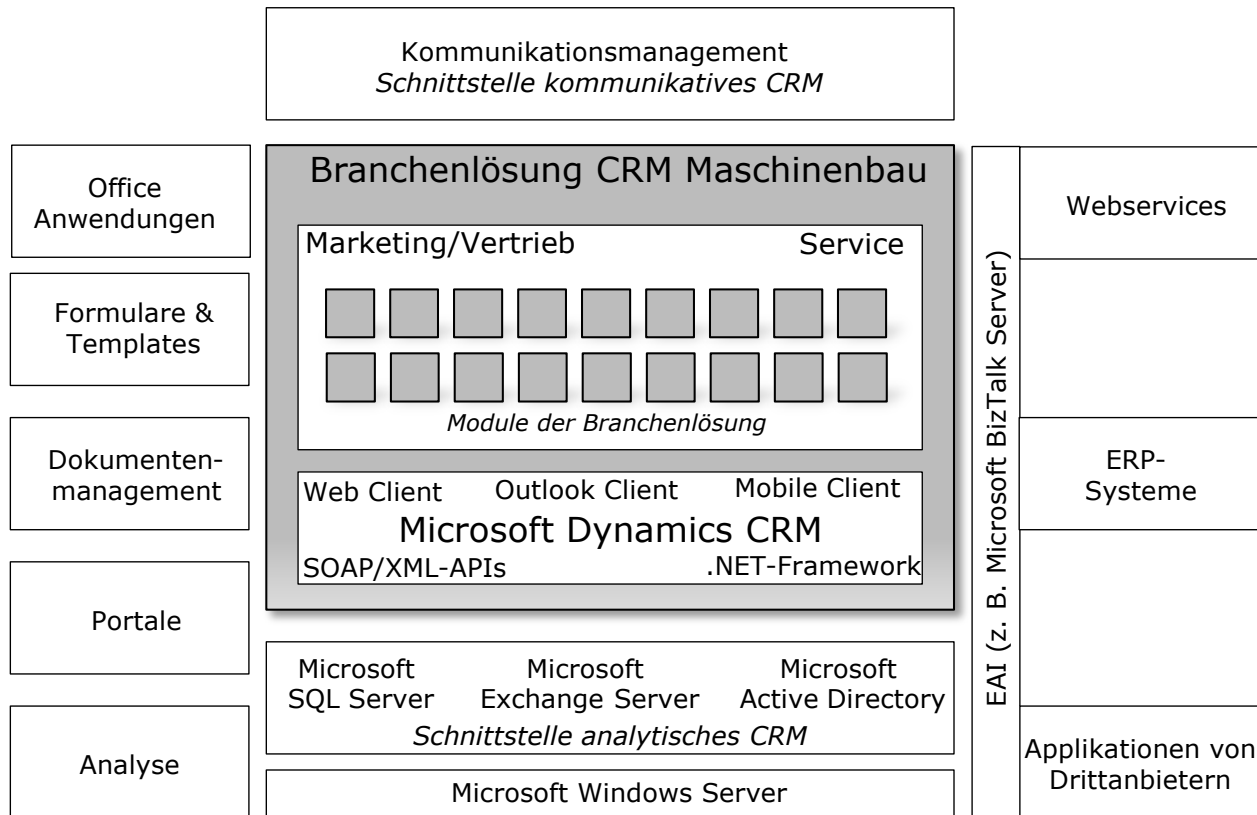
- Analyse im Business System Visioning



Fallbeispiel CRM Lösung: Systemarchitektur



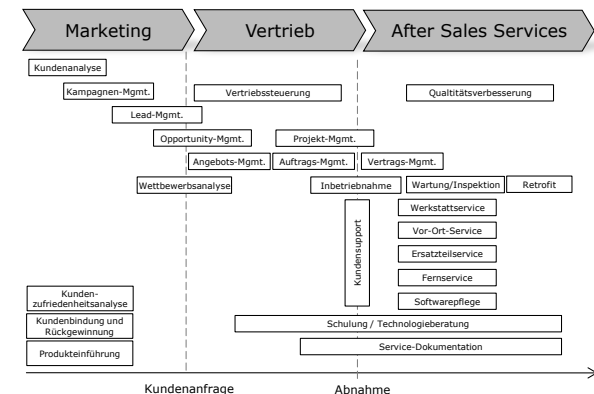
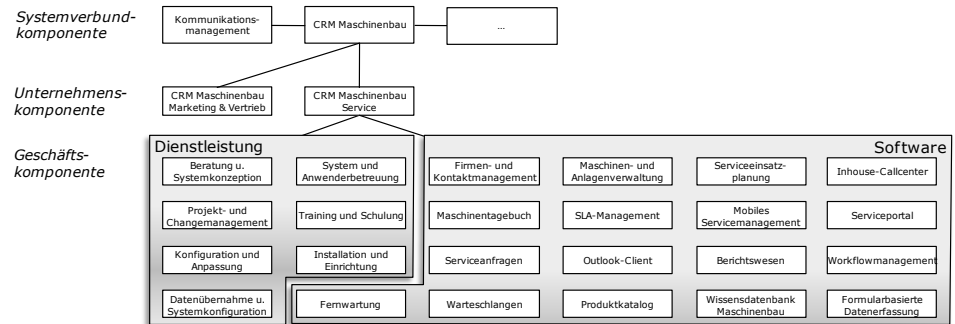
- Domain Engineering: Konzeption einer Plattform für den Anlagen- und Maschinenbau



Fallbeispiel CRM Lösung: Domain Engineering

- Aufnahme von Anforderungen
- Ableitung von Komponenten verschiedener Granularität
- Prozessanalyse

	Wiederverwertbare Anforderungen	Variable Anforderungen
Business Requirements	<ul style="list-style-type: none"> • Geringe Gesamtkosten • Schnelle Einführung • Automatisierung von Routine-tätigkeiten • Komponentenorientierung und leichte Konfiguration • Kenntnisse der Anwendungsdomäne • ... 	<ul style="list-style-type: none"> • Umfang der Gesamtleistung • Individuelle Preisgestaltung • ...
Service Requirements	<ul style="list-style-type: none"> • Geringer Schulungsaufwand • Best Practice Prozesse für Routinetätigkeiten • Einfacher Zugang zum Support • ... 	<ul style="list-style-type: none"> • Sich ändernde Marktbedingungen • Erbrachte Services • ...
Software Requirements	<ul style="list-style-type: none"> • Einfache Bedienbarkeit • Zentrale Installation • Hohe Produktivität • Einfacher Informationsaustausch (WebServices) • Workflowunterstützung • ... 	<ul style="list-style-type: none"> • Skalierbarkeit • Integration von Kundendaten • Konfiguration • Einbindung in die konkrete Systemlandschaft • Legacy-Systeme • ...

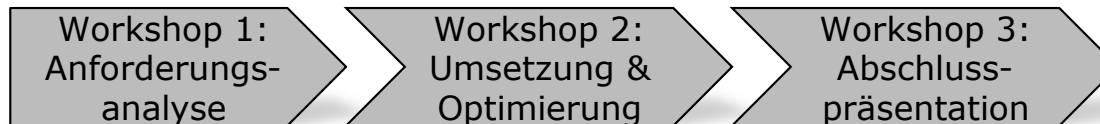


Fallbeispiel CRM Lösung: Solution Engineering



- Solution Engineering: Komponente Beratung und Systemkonzeption

- Evaluation der Szenarien
- Anpassung der Szenarien
- Aufnahme der IT-Infrastruktur und Systeme
- Vorstellung des Testsystems



- Vorstellung von Best Practice-Prozessen
- Einführung der Branchenlösung
- Definition und Besprechung der Kernprozesse
- Beschreibung der ausgewählten Szenarien

- Vorstellung der Szenarien
- Vorstellung der Anforderungsbeschreibung
- Vorstellung des Projektplanes
- Besprechung weiterer Schritte

Zusammenfassung und Reflexion

Zentrale Ergebnisse



Zentrales Ergebnis ist die **Methodik für die Entwicklung komponentenorientierter IT-basierter Dienstleistungen**, die ein Software-Services-Co-Design ermöglicht.

Die erarbeiteten Elemente dafür sind:

- ein konsistentes Systemverständnis des Entwicklungsgegenstandes basierend auf einer durchgängigen Komponentenorientierung
- eine geeignete Technik der Modellierung und Notation
- eine Strategie und die Benennung von Phasen, Arbeitsmodulen und -aktivitäten, die durch den Entwicklungsprozess führen



UNIVERSITÄT LEIPZIG



- Verbundprojekt, gefördert vom BMBF, PT Softwaretechnik im DLR
- FKZ: 01ISC36B
- <http://servcase.informatik.uni-leipzig.de/>

Danke für Ihre Aufmerksamkeit!



Kyrill Meyer

Universität Leipzig / Betriebliche Informationssysteme

Johanniskasse 26; 04103 Leipzig

meyer@informatik.uni-leipzig.de

www.bis.uni-leipzig.de