

Arbeitsbericht der GI Fachgruppe WI-VM – Arbeitskreis „Vorgehensmodelltypen“: Rahmen zur Auswahl von Vorgehensmodellen

Christop Filß, Warschau
Reinhard Höhn, Wien
Stephan Höppner, Berlin
Martin Schumacher, Linz
Herbert Wetzel, Augsburg,

Stand: März.2005

Inhaltsverzeichnis

1.	Zusammenfassung	184
2.	Entscheidungssituation Vorgehensmodell	184
2.1.	Motivation der Vorgehensmodell-Auswahl:	184
2.2.	Gang der VM-Entscheidung	185
2.3.	Ansätze von VM-Eigenschaften	186
2.3.1.	Kriterienskala oder Gegenpole (Verlage)	186
2.3.2.	Dimensionen des Software-Entwicklungsprozesses (Hesse 1998)	187
2.3.3.	Phasendurchlauf-Struktur (Bremer)	188
2.3.4.	CC-RIM	189
2.3.5.	Chroust(1994)	190
3.	Typisierung der Vorgehensmodelle, Kriterien	191
3.1.	Ausgewählte Dimensionen	191
3.2.	Generizität	193
3.3.	Submodelle	193
3.4.	Phasenabdeckung	194
3.5.	Objekt-Domäne	195
3.6.	Branchenfokus	196
3.7.	Formalisierungsart	196
4.	Bekannte Vorgehensmodelle und ihre Beurteilung	197
4.1.	IDEF	197
4.2.	UML	202
4.3.	ZACHMANN Framework	204
4.4.	V-Modell 97	207
4.5.	V-Modell XT (Version 1.0)	210
4.6.	RUP	215
4.7.	St.Gallen Business Engineering Modell und PROMET	219
4.8.	OTK	223
5.	Ergebnisse	226
6.	Ausblick	227

1. Zusammenfassung

Der Markt bietet eine Vielzahl von Vorgehensmodellen, Frameworks für Vorgehensmodelle, Methodensammlungen, Tool-Kits und er stellt auch immer wieder neue Vorgehensmodelle vor. Bei jedem neuen Projekt muss immer wieder die Frage gestellt werden, ob das im Hause eingesetzte Vorgehensmodell auch für die neuen Fragestellungen geeignet ist, ob neue Fragestellungen Erweiterungen erfordern, ob weitere Vorgehensmodelle eingeführt werden müssen, eventuell sogar komplett alte Ergebnisse zu einem neuen Vorgehensmodell migriert werden müssen oder ob man sich auf ein Framework einigt und dessen fallweise projektindividuelle Ausgestaltung zulässt. Man denke hierbei etwa an Themen wie Web-Applikationen, Knowledge-Management Projekte, Data Warehouse Lösungen für die gerade die Modelle mit dem höchsten Verbreitungsgrad (RUP, ARIS) nicht gerüstet sind.

In Berlin, im GI Workshop der Fachgruppe WI-VM, (VM04) am 22.4.2004 wurde die Idee geboren, das Entscheidungsproblem des Vorgehensmodelleinsatzes in einem Arbeitskreis zu behandeln:

- Vorgehensmodell einsetzen ja/nein
- wenn ja welches VM
- in welchem Umfang sollen VM eingesetzt werden, VM oder VM-Framework, etc.
- wie sind VM abzugrenzen, ist XP, sind agile Methoden in die Kategorie der VM einzuordnen

Die Arbeitsgruppe hat es sich zum Ziel gesetzt, die „Entscheidungsproblematik Vorgehensmodell“ zu beleuchten, mit Hilfe einer Kategorisierung einen Überblick über die bestehenden Vorgehensmodelle zu geben, und Kriterien für die Auswahlentscheidung vorstellen.

2. Entscheidungssituation Vorgehensmodell

2.1. Motivation der Vorgehensmodell-Auswahl:

Ausgangspunkt ist, dass vor dem Einsatz eines Vorgehensmodells die Entscheidung steht welches Vorgehensmodell implementiert werden soll, es gibt Alternativen. Das ist alles andere als trivial und eine Fehlentscheidung ist mit erheblichen Kosten verbunden. Alle Teilnehmer kennen aus eigener Erfahrung, Vorgehensmodell-Mix, Vorgehensmodelle die als „Schrank-ware“ enden und sogar Umschwenken auf eine anderes Vorgehensmodell mitten im Projekt, weil die gesetzten Projekt-Regeln nicht durchgehalten werden können. Warum setzt man sich überhaupt mit dem Thema „Vorgehensmodell“ auseinander, was erwartet man sich von dem Einsatz eines Vorgehensmodells, was sind die Befürchtungen der Beteiligten. Ein Vorgehensmodell soll zunächst helfen eine komplexe Handlungsfolge zur Herstellung eines komplexen Gegenstandes (Gewerk, Software, System, Gebäude, Anlage) zu ordnen, den Überblick über die Abfolge zu bewahren. Einige Fragestellungen hierzu sind :

- Wie ist Akzeptanz im Unternehmen, erlauben die Kultur und Mentalität der Mitarbeiter den Einsatz eines VM oder sogar nur eine bestimmte Kategorie eines VM: welche Akzeptanzbarrieren belasten die Auswahl
- Methodenvertrautheit, bereits gemachte Erfahrungen mit Tools,
- Gibt es Auswahl-Einschränkungen durch im Unternehmen vorherrschende Standardsoftware z.B. SAP-Nähe von ARIS
- Welche Typen und Ansätze zur Typisierung der Vorgehensmodelle
- Welcher Qualitätsbeitrag ist zu erwarten, Beitrag von VM zur Qualität der Projektergebnisse
- Welche VM-Frameworks gibt es, welche Architekturen haben VM
- Sind Vorgehensmodelle Benchmark-fähig, d.h. Aufnehmen von /Erweitern durch Kennzahlen
- Können Vorgehensmodelle bestehende Benchmarksysteme (z.B. CMMI, ITIL, ...) unterstützen
- Welche Implementierungsfähigkeiten bringen VM mit, wie ist die Wartbarkeit von VM

- Gibt es Erfahrungswerte und Statistiken zum Einsatz von VM, wie ist die Marktdurchsetzung von VM, welche Lebensdauer haben VM in Unternehmen
- Kann man ein VM aufgrund neuer Erkenntnisse anpassen, Anpassbarkeit von VM oder Anpassung des Unternehmens an das VM
- Ist V-Modell 200x ein Kandidat, welche Migrationsempfehlungen von V-Modell 97, welche Supportmöglichkeiten gibt es
- Welchen Nutzen bringt ein VM
- Liefern die VM einen Beitrag zur Qualität der Projektergebnisse und zur Wissenssicherung
- Bieten VM Toolunterstützung, wie weit ist die Automatisierung von Vorgehensmodellen fortgeschritten
- Gibt es Standardmodelle die mir bei Partnerschaftsprojekten eine Zusammenarbeit erleichtern
- Welche Kosten sind mit einer VM-Implementierung und Erhaltung verbunden
- Welche Kosten hilft mir ein gut genutztes VM zu ersparen
- Welche Risiken hilft das VM zu vermeiden, bzw. gegenzusteuern
- Gibt es Entscheidungshilfen bei der Auswahl von VM für bestimmte Projekte

2.2. Gang der VM-Entscheidung

Das Entscheidungsfeld ist komplex und vielschichtig, die Entscheidung ist mit Risiken verbunden und bietet einen nicht vorauskalkulierbaren Nutzen. Vor der Auswahl eines Vorgehensmodells steht deshalb die grundsätzliche Entscheidung ob ein neues VM eingesetzt werden soll oder mit dem bestehenden Modellen weitergearbeitet wird. Die Möglichkeiten (Entscheidungsraum 1) sind:

- das letzte eingesetzte Vorgehensmodell auch für die neuen Fragestellungen verwenden
- das letzte eingesetzte Vorgehensmodell auf neue Fragestellungen erweitern
- weitere Vorgehensmodelle einführen
- zu einem neuen Vorgehensmodell migrieren
- ein Framework einführen das fallweise projektindividuell ausgestaltet werden kann
- ist eventuell sogar ein individuelles VM neu herzustellen und einzuführen

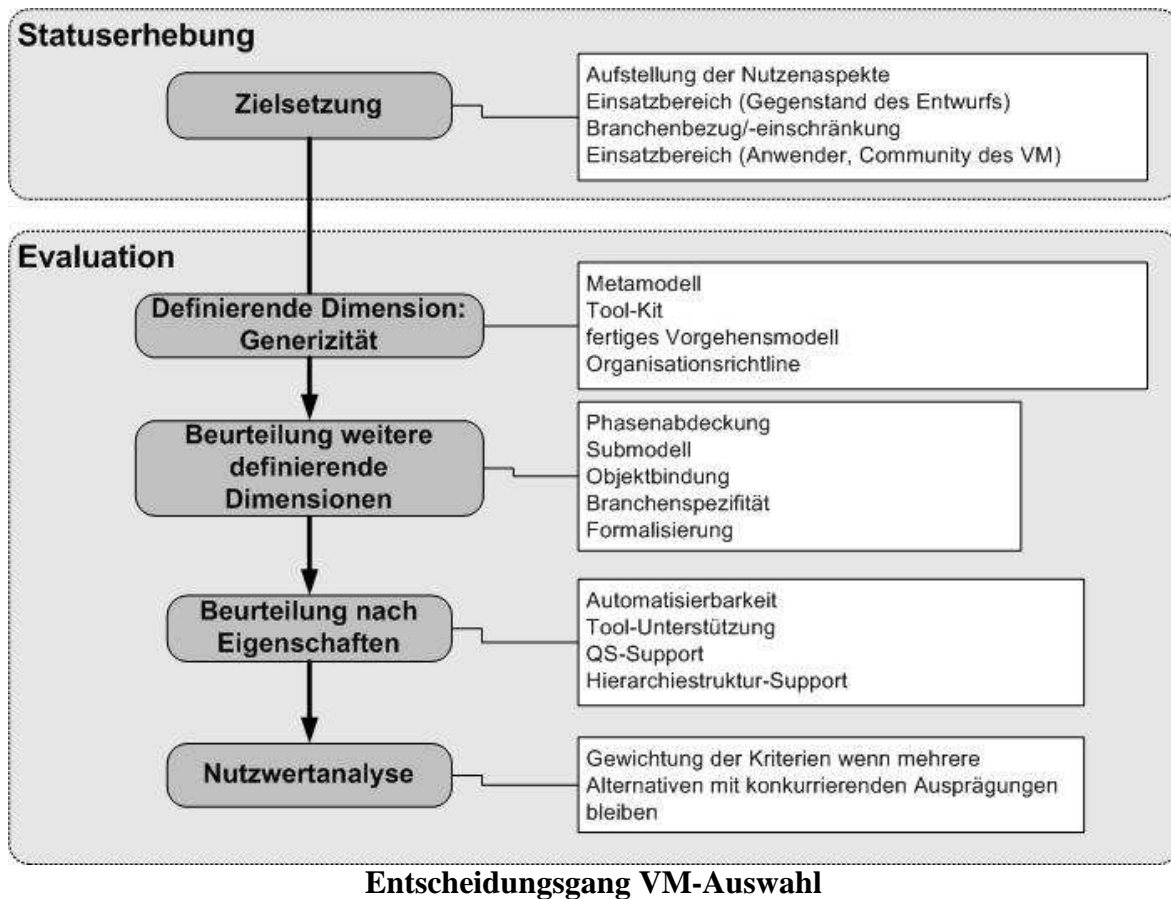
Ist die Entscheidung zu Gunsten eines neuen bestehenden Vorgehensmodells getroffen worden, beginnt die Aufstellung der Ziele, die mit dem neuen VM verfolgt werden und den daraus resultierenden Requirements.

Die nächste grundsätzliche also schwerwiegende Entscheidung, von der alle anderen anzuwendenden Kriterien abhängen ist die Entscheidung für eine Ausbaukategorie (Generalisierungsstufe oder Abstraktionsstufe). Soll ein fertig ausgebautes VM eingesetzt werden, dann ist keine Konfigurationsarbeit mehr erforderlich und auch nicht das Wissen aufzubauen wie konfiguriert werden kann. Oder soll ein Framework, das für eine große Breite von Projekttypen ausgebaut werden kann, bevorzugt werden, zu dem Preis neben dem methodischen Wissen auch noch das Meta-methodische aufbauen zu müssen.

Mit der Festlegung der Ausbaukategorie reduziert sich das Angebot der VM-Typen. Auf die reduzierte Menge werden die Kriteriengruppen der „charakteristischen Dimensionen“ angesetzt. Der Begriff definierend ist hier bevorzugt worden, da es sich nicht nur um Eigenschaften handelt sondern um Kategorien von Vorgehensmodellen. Das betrifft z.B. die Phasen: Entwurfsmodell, Implementierungsmodell, Strategiemodell etc. Das betrifft auch die Submodelle QS-Modell, PM-Modell, SE-Modell etc.

Neben diesen charakteristischen Dimensionen sind im Sinne der performanten VM-Anwendung noch Eigenschaften der Tool-Unterstützung zu evaluieren: Automatisierung, Benutzerunterstützung, etc.

Wenn dann noch Alternativen bleiben und man eine objektive Gegenüberstellung der Kriterien bezüglich der Alternativen erreichen will, müssen die einzelnen Kriterien noch gewichtet werden. Dabei hilft z. B. die Nutzwertanalyse.



2.3. Ansätze von VM-Eigenschaften

Für die Ermittlung der charakteristischen Eigenschaften können ein paar Ergebnisse aus der Literatur und von früheren Workshops der Fachgruppe nützliche Dienste erweisen. Im Folgenden werden deshalb eine Auswahl bekannter Ansätze analysiert und daraus die Zusammenstellung der beherrschbaren aber notwendigen Auswahl von Dimensionskandidaten gewonnen.

2.3.1. Kriterienskala oder Gegenpole (Verlage)

Die folgende Tabelle zeigt einmal in der Übersicht aus den Fragestellungen oben gewonnene Gegenpole mit ausgewählten Vor- und Nachteilen (verändert nach Verlage, 1998). Auch wenn einige Eigenschaften negativ klingen, sind Vor- und Nachteile bieten die sorgfältig erwogen werden müssen.

Erkenntnis: zwischen den Gegenpolen gibt es mehrere Zwischenstufen, die als graduelle Ausprägung des Kriteriums verwendet werden können.

1	Pol I	Gegenpol
	Vertrautheit	Novität
	Hohe Akzeptanz bei Konservativ-Denkern	Hohe Akzeptanz bei Neugierigen, Euphorie bei Spezialisten, Ablehnung bei Lernmüdem Personal
	Standardisierungsgrad	Individualität
	Anpassungen bestenfalls durch Parameter	Anpassungen möglich
	Offenheit	Abgeschlossenheit
	Anbindung weiterer Strukturen ist auf-	Keine Anpassungsaufwände, keine

wändig	Migrationsprobleme
Anpassbarkeit	Starrheit
Flexibilität auf neue Erkenntnisse reagieren zu können, Anpassungsmöglichkeiten erlernen nur Experten, Vorbereitung zum Einsatz erforderlich	Keine Anpassungsaufwände, keine Migrationsprobleme, schnell einsatzfähig

Kriteriengegenpole

Obwohl in Verlage der Focus der Ausführungen auf Prozessmodellierungssprachen liegt, sind einige Eigenschaften auch für die Vorgehensmodellthematik anwendbar, da ein Vorgehensmodell der IT-Entwicklung ja eine Folge von Arbeitsschritten, einen Prozess darstellt. Angelehnt an Verlage sind folgende VM-Eigenschaften interessant:

- **Modellnähe:** schnelle Erkennbarkeit der den Prozessmodellelementen zugehörigen realen Prozesselementen. Realitätsnähe der Symbolik
- **Meßunterstützung:** Zuordnung von Messgrößen und Kriterien zu den Prozessbestandteilen, Einrichtung von Messplänen
- **Konkretisierbarkeit:** generische Prozessmodellbestandteile auf konkrete Situationen anpassen, Tailoring im Sinne von Detaillierung
- **Formalitätsgrad:** eindeutige Symbolisierung, konsistenzfördernde Darstellung der Prozesse
- **Verständlichkeit:** Kommunikation der Inhalte zwischen den sich verständigenden Rolleninhabern eines Projekts
- **Ausführbarkeit:** Interpretationsfähigkeit, Maschinenlesbarkeit, Version für die Verarbeitung durch Rechner herstellbar
- **Nachvollziehbarkeit:** Möglichkeiten die Entstehung des Prozessmodells nach Zuständigkeiten festzustellen

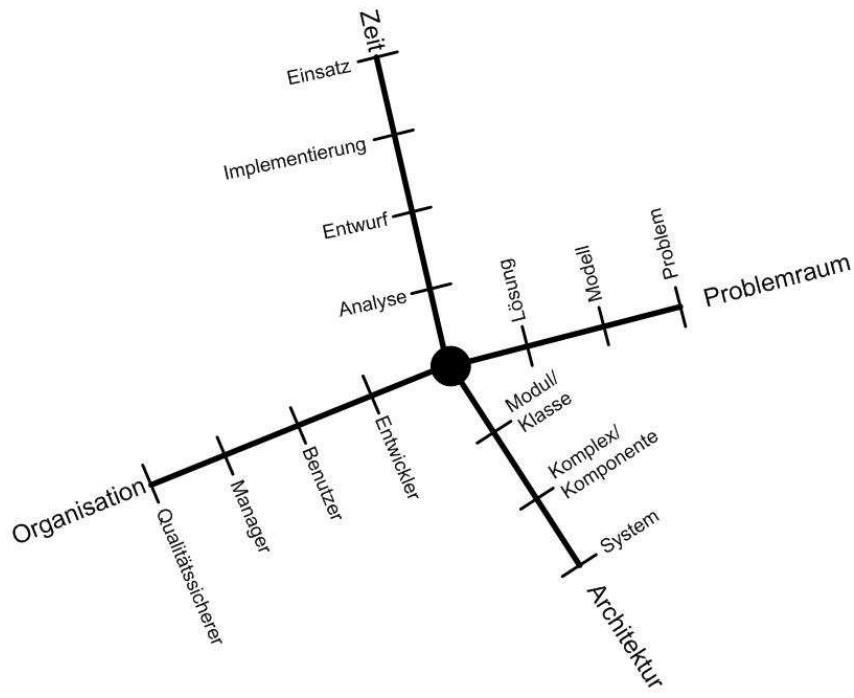
2.3.2. Dimensionen des Software-Entwicklungsprozesses (Hesse 1998)

Hesse stellt ein vierdimensionales Modell zur evolutionären Softwareentwicklung vor, wenn auch auf OO-Vorgehensmodelle beschränkt, (Jacobson OOSE, Rumbaugh OOM, Shlaer und Mellor OOSA, Booch OOA+D) und schon etwas in die Jahre gekommen (1998), trotzdem für unsere Zwecke nützlich.

Die Dimensionen des Software-Entwicklungsprozesses nach Hesse 1998 sind

- Architektur: Zerlegungsstufe der Softwarebestandteile
- Problemraum: Problemlösungsgrad, Nähe zur Lösung
- Zeit: Entwicklungsphase
- Organisation: Rollen

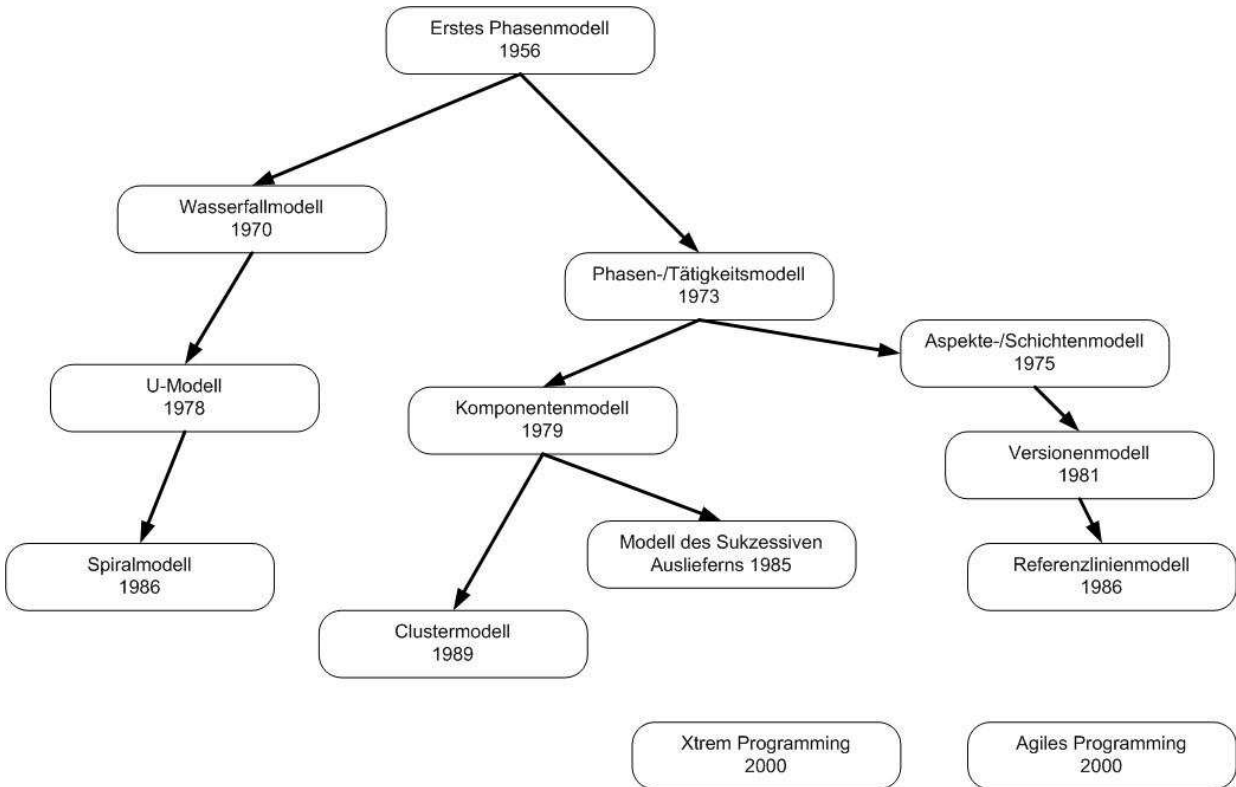
Zu den Dimensionen sind Ausprägungen aufgezählt. Folgende Abbildung zeigt dies.



Dimensionen des Software-Entwicklungsprozesses (Hesse)

2.3.3. Phasendurchlauf-Struktur (Bremer)

Alle Vorgehensmodelle unterteilen ein komplexes Aufgabenpaket in kleinere Pakete. Einige Pakete können erst bearbeitet werden, wenn andere Pakete fertiggestellt sind. Dadurch kommt zwangsläufig eine hierarchische Gliederung der Aufgaben und Ablaufstruktur zustande. Auf der obersten Ebene der Gliederung ist die Ablaufstruktur eine **Phasenaufteilung**. In der Geschichte des System-Engineering haben sich mehrere Phasenaufteilungen und mehrere Empfehlungen zum Durchlaufen der Phasen entwickelt. Die „Genealogie“ von Bremer gibt einen Überblick bezogen auf die Entwicklungsschemata in der Softwareentwicklung.



Phasendurchlaufvarianten nach Bremer

Der Phasendurchlauf kann, selbst wenn von den Autoren eines VM eine Empfehlung abgegeben wird, abweichend gestaltet werden. Die Phasenablauf-Varianten sind demnach kein Klassifizierungskriterium für Vorgehensmodelle, sondern ein Organisationsrichtlinie.

Bremer charakterisiert die Phasendurchlaufvarianten mittels der Eigenschaft „**Vorgehensweise**“:

- Sequentiell
- Inkrementell
- Iterativ
- Partizipativ
- Evolutionär

Die Zuordnung zeigt folgende Tabelle.

	Sequentiell	Inkrementell	Iterativ	Partizipativ	Evolutionär
Phasen/Wasserfallmodell	X				
U-Modell	X				
Phasen- /Tätigkeitenmodell	X				
Komponentenmodell	X	X			
Spiralmodell	(X)	X			(X)
Modell des Suzessiven Ausliefern		X	X		(X)
Clustermodell		X	X		(X)
Aspekte- /Schichtenmodell		X		X	
Versionen- /Referenzmodell		X	X	X	X

Phasendurchlaufvarianten und Vorgehensweisen

2.3.4. CC-RIM

Zur Beschreibung und zum Vergleich ist ein Beschreibungsansatz, ein Vergleichsrahmen nötig. Die bekannten Vorgehensmodelle unterscheiden sich auch in ihren Beschreibungsansätzen. Die im Folgenden aufgeführten Beschreibungsansätze dienen uns zur Orientierung und zur Auswahl der für uns interessanten Beschreibungselemente. Die Frage die sich stellt ist also: Bietet das ausgewählte Vorgehensmodell Beschreibungselemente die wir in unser Modell übernehmen möchten. Ein Ansatz hierzu bietet CC-RIM.

Elemente der VM- Beschreibung nach CC-RIM (Competence Center Rechnergestützte Informationsmodellierung) sind:

- **Entwurfsaktivitäten:** Aktivitäten zur Erzeugung von Entwurfsergebnissen (z.B. Datenmodell erstellen, Qualitäts-Check durchführen, Projekt planen), (kann Sub-Aktivitäten enthalten)
- **Rollen:** Menschen die Entwurfsaktivitäten ausführen, Entwurfsergebnisse produzieren, (z.B. Projektleiter, Analkytiker, etc.)
- **Entwurfsergebnisse:** Produkte der Entwurfsaktivitäten (z.B. Datenmodell, Klassendiagramm, Projektplan,)
- **Metadatenmodell:** Gesamtes Vorgehensmodell als Datenbankschema, zur Ablage der Ergebnisse und zur Verfolgung der Arbeitsstatus des Prozesses des Durchlaufens des Vorgehensmodells, in einer Datenbank
- **Techniken:** Anleitungen zur Herstellung eines Entwurfsergebnisses, z.B. Durchführung der Normalisierungsregeln beim Datenmodellieren,

2.3.5. Chroust(1994)

Im „Modelle der Software-Entwicklung“ von Chroust sind mehrere Dimensionen einschließlich ihrer möglichen Ausprägungen nachzuschlagen. Einige Dimensionen der VM- Beschreibung sind die Niveaus von Vorgehensmodellen und ihre Realisationsstufe.

Niveaus (Granularität) von Vorgehensmodellen

- Universell: prinzipielle Schritte, Aufzählung von Richtlinien, globale Ablaufstruktur
- Weltlich: Abfolge von Schritten, Voraussetzungen für Aktivitäten
- Atomar: präzise Definition aller Daten und Prozessschritte

Realisationsstufe von Vorgehensmodellen

- Prinzip
- Methode
- Methodik
- Verfahren
- Methodenbündel
- Methodologie
- Werkzeug
- Technologisches Verfahren
- Technik

Chroust stellt auch eine Zuordnung von Methoden z Phasen auf:

	E/R	JSD	SA	SADT	ISAC	ADT	ANI- MOSS	SD	CD	HIPO	LITOS	JSP	LCP	PC	StP
Anforderungsanalyse	X	X	X	X	X	X									
Fachkonzeption	X	X	X	X	X	X	X	X							
Grobentwurf	X	X	X	X	X	X	X	X	X	X	X				
Feinentwurf						X	X	X	X	X	X	X	X	X	X
Implementierung						X	X				X	X	X	X	X
Testen												X			

Phasen von Vorgehensmodellen und ihre Methoden (Chroust 1994):

Chroust liefert weiterhin eine synoptische Darstellung der Phasenstruktur zu bis 1994 bekannten Phasenmodelle:

	Balzert 82	Boehme 84	DoD 85	Enders 86	Hesse 4	IBM (ADPS) 87	IBM (AD Cycle 89
1 Planung	System Feasibility	System Concept	Projektvor-schlag	Analyse			Requirement
2 Definition	SW Plans and Requirements	System Requirements Analysis	Planung I	Definition	Define Model/Design	Define Mo-del/Design	Analysis and Design
3 =		Software Requirements Analysis	Planung II	=	=	=	=
4 Entwurf	Product Design	Preliminary Design	=	Systement-wurf	=	=	=
5 =	Detailed Design	Detailed Design	=	Komponenten-entwurf	Refine Mo-del/Design	=	=
6 Implementierung	Code	Coding and Unit Testing	Realisierung I	Modul-Implementierung	Product Code	Produce	

7	=	Integration	Component Testing, SW Configuration Item Testing	=	Subsystem-Integration	Integrate and Install System	Build and Test
8	=	Implementation	System Integration and Testing	=	System-Integration	=	=
9	Abnahme und Einführung	Operation and Maintenance	=	Realisierung II	Installation	=	Production and Maintenance
10	Pflege und Wartung	=	=	Einsatz	Betrieb und Wartung	Monitor System	=

Phasenstruktur bekannter Phasenmodelle (Chroust 1994)

Das Gleichheitszeichen zeigt an, dass die Themen des Feldes vom darüberstehenden Thema umfasst sind.

3. Typisierung der Vorgehensmodelle, Kriterien

3.1. Ausgewählte Dimensionen

Der Entwurf eines multidimensionalen Ansatzes kann aus den Kandidaten für Dimensionen aus den dargestellten Modell-Ansätzen gewonnen werden:

- Phasenabdeckung
- Phasendurchlaufbindung, Vorgehensweisen
- Konfigurierbarkeit
- Methodenbindung, Entwurfsergebnisse, Technikempfehlungen
- Branchenbezug
- Formalisierungsgrad
- Generizität (Metamodell, Framework, Referenzmodell, ..)
- Hierarchie-Support
- Automatisierbarkeit, Ausführbarkeit
- Strukturpakete (SE, PM, QS, Benchmarking, KM, CM)
- Entwurfsobjektbindung (Web, OO, KM, Technische Systeme, DWH....)
- Einsatzflexibilität bzw. Bindungsstärke, Abhängigkeitspotential
- QS-Unterstützung, Messunterstützung
- Qualitäts-(Reifegrad) Zertifizierungssupport
- Rollenmodell, Organisationseinheiten
- Architekturstufe, Zusammenbaustufe, Zerlegungsgrad,
- Nachvollziehbarkeit, Historie
- Verständlichkeit, Kommunizierbarkeit, Mehr-Rollen-Verständnis

Einige Dimensionen sind charakteristisch für den Typ des Vorgehensmodells andere sind eher als zusätzliche Eigenschaftsdimensionen zu sehen. Der Vorschlag des AK-VW-Typen für die charakteristischen Elemente des VM- Vergleich sind:

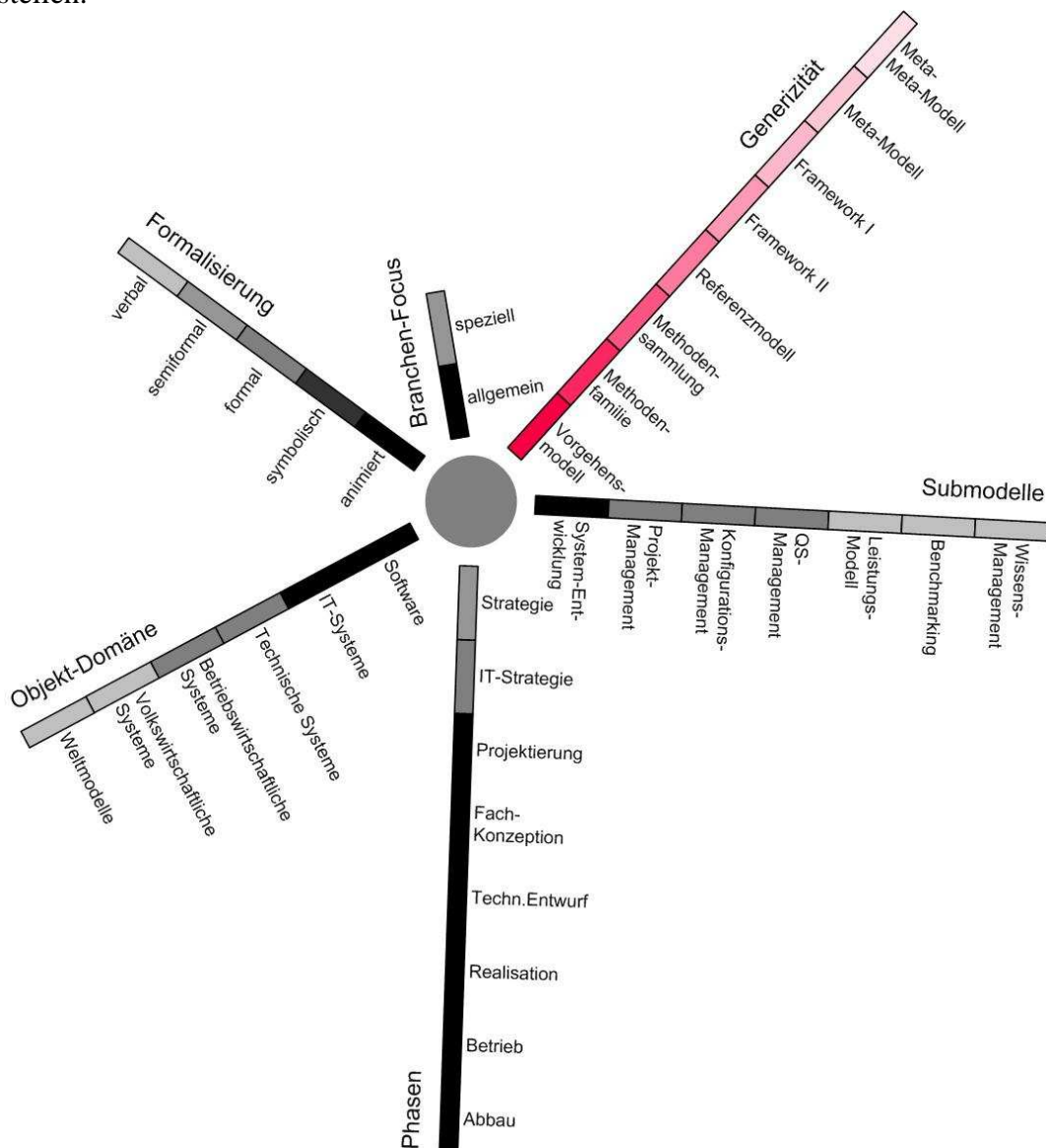
- Generizität (Metamodell, Framework, Referenzmodell, ..)
- Phasenabdeckung
- Submodelle (SE, PM, QS, Benchmarking, KM)
- Entwurfsobjektbindung (Web, OO, KM, Technische Systeme,)
- Branchenspezifität
- Formalisierung

Die Abbildung unten zeigt die charakteristischen Dimensionen der Einordnung der Vorgehensmodelle, diese sind grundlegend für eine Einsatzentscheidung. Die Graustufen geben einen Hinweis auf die Ausprägung der Dimensionen, Gradation genannt. Einige Ausprägungen einer Dimension besitzen eine Gradation. Dabei treten folgende Fälle auf:

- Einige Dimensionen haben Ausprägungen die einfach bis mehrfach auftreten können (z.B. eine oder mehrere Phasen), dargestellt durch Färbung mehrerer Felder auf der Dimensionsachse.
- Einige Dimensionen können eine Ausprägung haben die alle anderen Ausprägungen umfasst (Skalierbarkeit der Dimension) z.B. Technisches System umfasst IT-System umfasst IT-Hardware und IT-Software, dargestellt durch Färbung mehrerer Felder auf der Dimensionsachse.
- andere Dimensionen haben genau eine Ausprägung (Referenzmodell oder Framework), und schließen zusätzliche Ausprägungen aus (Exklusivität)
- wieder Dimensionen andere können eine Ausprägung in Abstufungen bieten, (Software generell, auf spezielle Softwaretypen eingeschränkt), dargestellt durch Graustufen der Felder auf der Dimensionsachse.
- einige Ausprägungen können mehrere Abstufungen umfassen (Skalierbarkeit der Ausprägung), dargestellt durch Graustufen der Felder auf der Dimensionsachse.

Im Falle von Abstufungen muss zu den Graustufen eine Legende erstellt werden..

Die Graustufen in der unten stehenden Sterngrafik sollen die Bedeutung für die Softwareentwicklung darstellen.



Charakteristische Dimensionen der Vorgehensmodelle

3.2. Generizität

Unter VM werden mehrerer verschiedene Ansätze unterschiedlicher Konkretisierung und Verwendungsverpflichtung subsummiert. Die Betrachtung muß also etwas weiter gefaßt werden, als der strenge Begriff „Vorgehensmodell“ umfaßt. Was zur Erstellung eines der Firmensituation angemessenen VM hilft ist nicht nur ein komplettes fertig ausgebautes nur noch auf ein Projekt anzuwendendes VM, sondern auch Ordnungsrahmen, Checklisten zum Vorgehen, Metamodelle, Referenzmodelle können in Frage kommen. Diese Varianten stellen **VM-Ausbau-Kategorien** dar, oder können im umgekehrten Blickwinkel als **Generizitätskategorien** bezeichnet werden.

Die Eigenschaften die zu evaluieren sind, sind der Fertigstellungsgrad, der ausdrückt wieviel noch zu entwerfen ist um zu einem konkreten Vorgehensmodell zu kommen und die Anpassbarkeit, die ausdrückt welche Abweichungen vom vorgegebenen Modell zugelassen sind, wieviel Bindung das Vorgegebene Modell fordert. Ob weitere Bestandteile z.B. von anderen VM, oder selbst entworfene, integrierbar sind.

Kriterium: Generizität, alternativ: Abstraktheit, Fertigstellungsgrad

Kritische Frage: Ist das Modell fertig ausdefiniert, gibt die zu durchlaufenden Schritte vor, nennt zu erzeugenden Ergebnisse, schreibt die zu verwendenden Methoden vor, oder ist das Modell anpassbar?

Folgende VM-Ausbau-Kategorien können unterschieden werden.

VM-Kategorie	Erklärung	Beispiel
1 Meta-Metamodell	Frei definierbarer Objektesatz mit einigen Grunddefinitionen, z.B. Entitätstypen, Attributtypen, Beziehungstypen, Metamodell-Kit	Design O/A, Metadesign
2 Metamodell	Datenbankstruktur aus Ergebnistypen zur Ablage von Entwicklungsergebnissen	Rochade CC-RIM
4 Framework I	Ordnungschema, Checkliste, nicht Toolgestützt	Zachmann
5 Framework II	Toolgestützter Ordnungsrahmen, Directory-Struktur	V-Modell
6 Referenzmodell	Vordefinierte Bestandteile eines Vorgehensmodells, erweiterbar und kürzbar	V-Modell
7 Methodensammlung	Zueinander passende Methodensammlung, muß nicht vollständig sein	UML
3 Methodenfamilie mit Repository	Zueinander passende Methodensammlung deren Ergebnisse über DD, Repository oder Enzyklopädie auf Konsistenz geprüft werden	
8 Vorgehensmodell	Begrenzter Methodenvorrat,	ARIS

VM-Ausbau-Kategorien

3.3. Submodelle

Die die Produktion der Primärergebnisse überwachenden Begleitmodelle. Erzeugung von Sekundärergebnissen. Sekundär deshalb, weil ein Bezug zu den Primärergebnissen besteht. Ohne Primärergebnis hat das Sekundärergebnis keinen Gegenstand.

Vorgehensweisen können sich auch auf einzelne Modelle konzentrieren. Aus der Sicht der Vorgehensmodell-Komponenten (Submodelle) sind dies die folgenden speziellen Modelle:

- Systementwicklungsmodell,
- Projektmanagementmodell, ,
- Qualitätssicherungsmodell,
- Konfigurationsmodell, ,

- **Benchmarkingmodell**
- **Leistungsmodell**
- **Wissensmodell**

Die Modelltypen sind gegeneinander nicht abgrenzbar, überschneiden sich gegenseitig und enthalten sich z.Teil.

Kriterium: Submodell alternativ: Modelkomponenten, Strukturkomponente

Kritische Frage: Welche parallelen Modelle können die Erzeugung der Primärergebnisse unterstützen

	Ausprägung	Erklärung	Beispiel
1	Systementwicklung	mit dem Schwerpunkt auf die Strukturierung und Abfolge der Produktionsschritte, Nennung derer Produktionsergebnisse und der Mittel und Methoden mit denen produziert wird	V-Modell 97
2	Projektmanagement	mit dem Schwerpunkt auf die Aufgabenstrukturierung, der Zuordnung von Terminen und Terminabhängigkeiten, Ressourcen zur Abwicklung der Aufgaben	PRINCE, PMBOK, ASQF
3	Konfigurationsmanagement	zur Objektgliederung, Analyse zu die Grundbestandteile, Synthese der Grundbestandteile zu höheren Baugruppen	
4	Qualitätsicherung	mit dem Ziel Qualitätsmerkmale zu erklären, zu bewerten und die Herstellung der Qualität entsprechend der vorgegeben Qualitätsziel zu erzeugen	TQM, QFD, EFQM
5	Leistungsmodell	mit dem Ziel Leistungspositionen für Angebote zu schaffen, Kosten zu ermitteln, Budgets zu berechnen, Abnahmekriterien zur Anerkennung der Leistung als Lieferung	BVB,
6	Benchmarking	mit dem Ziel allgemeine Zielgrößen zu definieren, deren Erreichung zu überprüfen, Abhängigkeiten der Erreichung von Maßnahmen	CMM, CMMI, KMMM
7	Wissensmanagement	mit dem Schwerpunkt auf die Wissensgewinnung, Wissenserhaltung aus den und der Wissensbeistellung zur Abwicklung der Aufgaben	

Ausprägungen der Dimension Submodelle

3.4. Phasenabdeckung

Gemessen an einem alle mögliche Phasen umfassenden Vorhaben kann ein Modell nur für bestimmte Phasen entwickelt worden sein.

Die BW-Strategie kann auch als Start eines BW-Phasenmodells gesehen werden. Beim Einsatz von zwei Phasenmodellen, eines für IT, eines für BW, ist darauf zu achten. Hier ist BW mit aufgenommen, um zu zeigen, dass die IT von der BW Vorgaben braucht um den Unternehmenszielen angemessene und zutragende Lösungen zu schaffen.

Kriterium: Phasenabdeckung, alternativ: Phasenreichweite

Kritische Frage: Welche Phasen eines Systementwicklungsprozesses werden vom Modell unterstützt

	Definition	Erklärung	Beispiel
1	BW-Strategie	Zieldefinition, Vision, Politik, Strategische Massnahmen	Methoden wie bei BSC Wie Catalyst
2	IT-Strategie	Projektportfolio, IT-Budget, Multiprojektmanagement	
3	Projektierung	Strukturplan, Balkenplan, Netzplan, Einsatzplan	
4	Konzeption	Requirements, Fachliche Anforderungen, Pflichtenheft, Grobkonzept	Fachsprache der Anwender Informationsmodell Datenmodell
5	Techn. Entwurf	Technische Spezifikation, Technisches Konzept, Feinkonzept	
6	Realisation	Programmierung, Beschaffung, Datenerfassung, Funktionstesten, Montage, Installation, Integrationstesten,	fertiges System (SW, HW, Maschine, Anlage)
7	Betrieb	Nutzung, Anwenderbetreuung, Systemmonitoring, erhalten der Verfügbarkeit Betriebsauswertungen, Kostenerfassung, Datensicherung,	
8	Abbau	Demontage, Datenarchivierung, Migration zu neuere Anwendung	

Phasenabdeckung

3.5. Objekt-Domäne

Das Modell kann nur auf einen speziellen Gegenstand angewendet werden, Modell ist auf besondere Entwicklungsgegenstand, (Domäne) ausgerichtet, es gibt eine **Entwurfsobjektbindung**.

Die Strukturierung nach dem Gestaltungsobjekt kann selbst wieder in Arten gegliedert werden, z.B. Software weiter in Web –Applikationen, OO-Anwendungen, Real-Time Software, Knowledge Management etc. und die wieder weiter in **Softwaretypen** DWH, CMS, ERP, DMS, KMS, XPS, DSS, etc. Einige Vorgehensmodelle beziehen nur auf diese dritte Stufe der Zerlegung, den Softwaretyp.

Kriterium: Entwurfsobjektbindung, alternativ: Entwurfskategorienbindung, Entwurfsgegenstandsfokus

Kritische Frage: Ist das Modell speziell auf einen bestimmten Gegenstand konzipiert, wenn ja welcher Entwurfsgegenstand ist das.

	Ausprägung	Erklärung	Beispiel
	Software	Web, OO, Real-Time, Knowledge Management	Content Management System, ERP
1	Hardware	Rechner, Netze	Firmen-LAN, Firmen-WAN, Server-Park,
2	Technische Systeme	Maschinen, elektrotechnische Systeme, verkehrstechnische Systeme, haustechnische Systeme, verfahrenstechnische Systeme, humantechische Systeme	Rechenzentrum, Kraftwerk, Autobahnanlage und Betriebsgesellschaft
3	Betriebswirtschaftliche Systeme	Unternehmen mit Anlagen, Personen, Behörden, Betriebe,	Bahn AG Kraftwerksbetrieb Service Provider
4	Volkswirtschaftliche Systeme	Staaten, Länder, Ländergruppen, mit Betriebswirtschaftlichen Organisationen, Behörden	

Ausprägungen der Dimension Entwurfsobjektbindung

3.6. Branchenfokus

Die Frage nach dem Branchenbezug ist aus der Softwaresicht schlecht nachzuvollziehen, da die Methoden der Softwareentwicklung für Banken dieselben sind wie die für einen Industriebetrieb oder eine Behörde. Wenn man allerdings den Entwurfsbogen etwas weiter spannt und nicht nur Software sondern auch technische Systeme einbezieht, kommen bei technischen Anlagen Maßzeichnungen, Gebäude-Architekturentwürfe, Formenentwürfe, technische Berechnungen, Auslegungsplanungen dazu. Eine speziell auf die Anlagenindustrie zugeschnittenen Lösung könnte z.B. eine methodische Integration des Funktionalen Entwurfs und der Anforderungen mit Auslegung beinhalten.

Es gibt betriebswirtschaftliche Referenzmodelle speziell für einige Branchen. Eine Klassifikation von Vorgehensmodellen nach Branchen ist uns nicht bekannt.

Kriterium: Branchenbindung, alternativ: Branchenfokus

Kritische Frage: Ist das Modell auf eine bestimmte Branche oder Branchengruppe ausgerichtet oder allgemein über alle Branchen einsetzbar.

	Ausprägung	Erklärung	Beispiel
1	Industrie	Maßzeichnungen, Gebäude-Architekturentwürfe, Formenentwürfe, technische Berechnungen, Auslegungsplanungen, Zuverlässigkeits und Ausfall-Analysen	
2	Finanzen	Banken, Versicherungen, enthält besondere Elemente für Geldlogistik, Zahlungsverkehr	
3	Government, öffentl. Dienst, Dienstleistung	Juristische Relevanz, politische Elemente,	
4	Militär	Für Strategien, Einsatzplanungen, Schlachtfeld-Organisation, Szenario-Diagramm, Geografie-Entwürfe, Strategieelemente	

Ausprägungen der Dimension Branchenbindung

Für die Balkeneinfärbung wird folgendermaßen verfahren:

- grau = für alle Branchen anwendbar,
- weiß = nicht anwendbar auf Branche xxx
- schwarz = Branche wird konkret unterstützt

3.7. Formalisierungsart

Modelle können als Anweisungen in Vorschriften und Organisationsrichtlinien festgelegt sein, mittels formaler Methoden und Regelwerke beschrieben werden, als Metamodelle hinterlegt werden..

Kriterium: Formalisierungsart, alternativ: Formalisiertheit

Kritische Frage: Mit welchen Mitteln (Sprachen) ist das Modell formalisiert worden.

	Ausprägung	Erklärung	Beispiel
1	Verbale Beschreibung	Text, Normalsprache, Fachsprache	Richtlinie, Organisationsanweisung
2	Programmiersprache	Syntax, Semantik, Alphabet	C#, Visual Basic, XML, HTML, SQL
3	Höhere Programmiersprache	Makros, Spezialsprache, Syntax, Semantik, Alphabet, Regeln, Programmbausteine	OWL, RDF,
4	Grafik	Grafische 2D-Symbole, Symbol-Semantik, Verwendungsregeln, Symbolverbindungen, Textergänzungen,	E/R, IDEF-ERM, SERM, OntoEdit, UML-Klassendiagramm, Use Case
5	3D? Animation??	Wie 4 nur in 3D-Raum eingebettet, Navigation in 3.Dimension	?

Ausprägungen der Dimension Formalisierungsgrad

Bremer, G., Genealogie von Entwicklungsschemata in Kneuper R., u.a., Vorgehensmodelle für die betriebliche Anwendungsentwicklung, 1998

Verlage, M., Modellierungssprachen für Vorgehensmodelle, in Kneuper R., u.a., Vorgehensmodelle für die betriebliche Anwendungsentwicklung, 1998

4. Bekannte Vorgehensmodelle und ihre Beurteilung

4.1. IDEF

Historie und Motivation

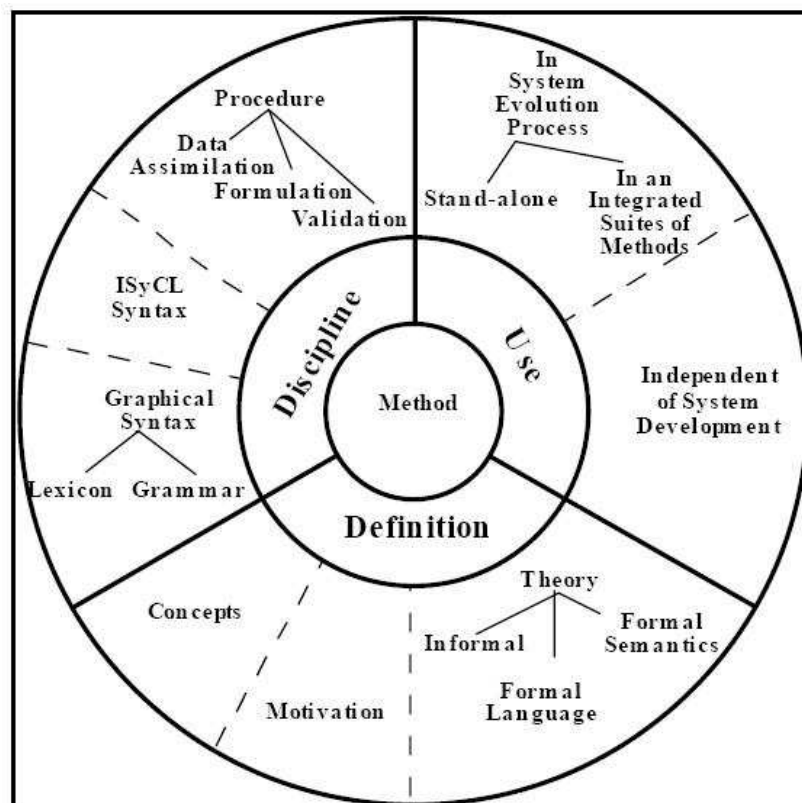
In den 1970er Jahren versuchte die U.S. Air Force im Programm für Integrated Computer Aided Manufacturing (ICAM) die Produktivität durch systematische Anwendung der Computer Technologie zu steigern. Dabei wurde die Notwendigkeit für bessere Analyse und Kommunikationswerkzeuge für die in den Projekten involvierten Personen festgestellt.

Als Ergebnis wurden die IDEF (ICAM Definition) Methoden erstellt, die dazumal aus IDEF0 (Funktionsmodell), IDEF1 (Informationsmodell) und IDEF2 (Dynamisches Modell) bestanden.

1983 wurde die IDEF1 Methode zur IDEF1X (IDEF1 Extended) Methode weiterentwickelt, wodurch eine semantische Datenmodellierungsmethode geschaffen wurde. Anfang der neunziger Jahre waren die IDEF0 und IDEF1X Methoden in den USA bereits weit verbreitet.

1991 begann das National Institute of Standards and Technology (NIST) mit Hilfe des U.S. Department of Defense, Office of Corporate Information Management (DoD/CIM), die IDEFx Methoden zu Standards zu erheben.

Die entwickelten Methoden wurden bei Corporate Information Management (CIM), Concurrent Engineering (CE) und Computer Integrated Manufacturing Projekten erfolgreich eingesetzt, bevor 1992 als erstes die Methode IDEF1 in Form einer Rekonstruktion eines Technical Reports [MAYER1992] als Standard veröffentlicht wurde. Der Ansatz wird durch die folgende Grafik verdeutlicht:



IDEF Methoden im Zusammenhang

In den folgenden Jahren wurden bis 1995 die Standards IDEF0 bis IDEF4 und der Standard IDEF9 veröffentlicht. Ursprünglich waren 14 verschiedene Standards zur Abdeckung aller Modellierungs- und Beschreibungsarten geplant, die aber nicht alle ausgearbeitet wurden.

Während die IDEF Methoden auf damals bereits bewährte Methoden aufsetzten und diese adaptierten, basieren heutige UML Erweiterungen im Bereich Business Modelling teilweise wiederum auf IDEF [NORAN2003].

Definition bzw. Einordnung

Die IDEFx Methoden sind eine frei verfügbare Methodensammlung zur Erstellung von Software und zum Business Modelling. Die Standardbeschreibungen und erläuternde Texte können bei [KBI2005] geladen werden.

Eine Liste von Tools, die IDEF Modellierung unterstützen, wird in [HOMM2001] gegeben.

Es ist nicht angedacht, die in den unterschiedlichen Methoden entstandenen Diagramme und Beschreibungen untereinander zu vernetzen oder in ein gemeinsames Repository zu geben, wodurch sich die IDEF Methoden als Methodensammlung kennzeichnet im Gegensatz zu einer Methodenfamilie.

Aufbau

Die IDEFx Methodensammlung umfasst folgende Methoden mit ihren Beschreibungen:

IDEF0	F unction M odeling
IDEF1	I nformation M odeling
IDEF1X	D ata M odeling
IDEF3	P rocess D escription C apture
IDEF4	O bject O riented D esign
IDEF5	O ntology D escription C apture
IDEF6	D esign R ationale C apture
IDEF8	U ser I nterface M odeling
IDEF9	S cenario-driven I S D esign
IDEF10	I mplementation A rchitecture M odeling
IDEF11	I nformation A rtifact M odeling
IDEF12	O rganization M odeling
IDEF13	T hree S chema M apping D esign
IDEF14	N etwork D esign

IDEF Methodensammlung inklusive der nicht entwickelten Methoden [MAYER1992]

In der Entwicklung abgeschlossen und tatsächlich erhältlich sind die Standards IDEF0, 1, 1x, 3, 4, 5 und 9. Die Standards 2 und 6 werden in [MAYER1992] beschrieben, sind aber nicht erhältlich.

Neben der Beschreibung der Syntax zur Erstellung der Diagramme in den Methoden wird großes Augenmerk auf die Beschreibung der Methodik (Guidelines) gelegt, mit der bei der Erstellung der Diagramme vorgegangen werden soll. Alleine beim IDEF0 Standard nimmt diese Beschreibung etwa die Hälfte des Umfangs der gesamten Standardbeschreibung ein.

Während die Methoden IDEF0 und IDEF1 zur Analyse und Modellierung dienen, werden die Methoden IDEF3 und IDEF5 in beschreibender Weise verwendet und dienen der Dokumentation des entworfenen Systems und der Kommunikation zwischen den beteiligten Personen.

Die in den Diagrammen verwendete Symbolik beschränkt sich auf nicht animierte 2D Symbole, die mit Pfeilen verbunden werden, wobei die Gestaltung der Symbole und Pfeile, sowie die Art der Anordnung Bedeutung in sich trägt. So macht es etwa in einem IDEF0 Diagramm einen großen Unterschied, von welcher Seite aus ein Pfeil auf ein Symbol trifft.

Funktionen, Informationen, Prozesse

Da IDEF Diagramme zum Zwecke der Dokumentation und Kommunikation gedacht sind, sind sie durchwegs so ausgelegt worden, dass sie in hierarchischer Detaillierung erstellt werden können. Hierbei wird gute Unterstützung und Anleitung gegeben, wie die Hierarchisierung vorgenommen werden soll, damit das erstellte Diagramm in sich Sinn ergibt und verständlich bleibt.

Funktionen, Informationen und Prozesse werden mit jeweils eigenen IDEF Methoden beschrieben. Wird das zu erstellende System objektorientiert entworfen, so ist dazu die IDEF4 Methode zu verwenden anstatt der Methoden IDEF0 und IDEF1.

Hervorgehoben wird, dass IDEF0 keinerlei zeitliche Abfolge in den Funktionen beschreibt, dies ist den prozessbeschreibenden IDEF3 Diagramme vorbehalten.

Die Ursprünge von IDEF1 liegen unter anderem in den ER Diagrammen von Peter Chen, IDEF1 erlaubt aber zusätzlich auch die Beschreibung von nicht in der Software abgelegten Informationen, wie zum Beispiel Personen und Akten. Weitere Ursprünge von IDEF1 sind ELKA Diagramme und das relationale Modell von Codd.

Vor dem heute üblichen Tailoring wird ausdrücklich gewarnt, um die Diagramme konsistent zu halten und die kurze Einlernzeit beim Austausch von Diagrammen zum Zweck der Kommunikation zu erhalten.

Qualitätskriterien werden in Zeiten angegeben, die ein unbedarfter Benutzer zur Einarbeitung in die erstellten Diagramme brauchen darf. Andernfalls wäre das Diagramm zu kompliziert ausgefallen.

IDEF1X stellt eine Erweiterung von IDEF1 dar, indem es versucht, eine Brücke herzustellen zwischen logischer Datenmodellierung der sprachlichen Beschreibung von Entitäten der realen Welt. Die Automatisierung der Ergebnisse aus IDEF1X gestaltet sich demnach schwieriger als die Automatisierung der Erstellung eines physikalischen Datenmodells eines IDEF1 Diagramms.

Die nicht standardisierte IDEF2 Methode diente zur grafischen Simulation und erforderte seiner Natur entsprechend einen Spezialisten im Simulationsdesign, da sich die Simulationserstellung kompliziert gestaltete und schwer lesbar war.

IDEF3 besteht aus zwei Untermethoden, den Process Flows und den Object State Transitions. Die Process Flow Beschreibungen dienen dem Erfassen des Wissens über die Abläufe im System, während die Object State Transition Beschreibungen angeben, welche Veränderungen ein Objekt unter verschiedenen Umständen durchlaufen darf.

IDEF4 Diagramme sind den heute üblichen Klassendiagrammen sehr ähnlich, wobei es fünf Unterdigramme umfasst, die folgende Aspekte beschrieben: Vererbung, Typisierung, Methodenaufprotokolle, Methodenähnlichkeiten und Methodenaufzugsberechtigungen.

Obwohl IDEF4 als Entwurfsmethode konzipiert wurde, wird es meist als Dokumentationsmethode eingesetzt.

Charakteristik im Detail

Generizität

Da die Vernetzung von Diagrammen nicht vorgesehen ist, genauso wenig wie ein gemeinsames Repository, grenzt sich IDEF von typischen Vertretern der Methodenfamilien ab.

Als Sammlung von Methoden zur Modellierung von Aspekten eines Systems in den Bereichen Corporate Information Management Systems, Concurrent Engineering und Computer Integrated Manufacturing klassifiziert sich IDEF somit klar als Methodensammlung.

Da kein Vorgehen außerhalb der Modellierung der einzelnen Methodendiagramme angegeben wird, handelt es sich bei IDEF auch nicht um ein Vorgehensmodell.

Phasenabdeckung

Die IDEF0 Methode wird ausdrücklich zur Analyse von Anforderungen empfohlen, sofern sie in einem hohen Abstraktionsgrad durchgeführt wird. Dies gilt auch für die restlichen IDEF Methoden, deren Ergebnisse in Folge in die Grobkonzepte eingehen.

Werden die erstellten Diagramme mit großer Sorgfalt und Genauigkeit erstellt, können sie auch zur Unterstützung der technischen Spezifikation und des Feinkonzepts verwendet werden.

Sub-Komponenten

Durch den Analyse sowie Dokumentations- und Wissensaustauschcharakter der IDEF Methoden kann das Submodell Wissensmanagement mit dem Schwerpunkt auf Wissensgewinnung und Wissensserhaltung zu den abgedeckten Sub-Komponenten gezählt werden.

Objekt-Domäne

Die IDEF Methoden sind ausschließlich für die Entwicklung von Software gestaltet.

Branchenfokus

Unter den IDEF Methoden finden sich keine auf bestimmte Branchen zugeschnittenen Methoden, obwohl der Ursprung der IDEF Methoden in der militärischen Softwareentwicklung liegt.

Formalisierung

In den zur Verfügung stehenden Standardbeschreibungen, die ausschließlich im PDF-Format erhältlich sind, liegen die IDEF Methoden in verbaler Form, unterstützt durch erläuternde 2D Grafiken vor.

Die Notation und Verwendungsart der Diagramme wird sehr ausführlich in verbaler Form beschrieben.

QS-Beitrag

Zur Überprüfung der Qualität der erstellten Diagramme werden sehr einfache Tests beschrieben, die nur selten per Regelwerk oder automationsunterstützt nachvollzogen werden können. Tools zur Erstellung der Diagramme können aber bei der Hierarchisierung Unterstützung geben, indem der Modellierer gezwungen wird, in Diagrammen höherer Ebene eine geringere Komplexität als in untergeordneten Diagrammen zu modellieren.

System-Zerlegung

Die einzelnen IDEF Methoden existieren weitgehend parallel zueinander und sind nicht vernetzt.

Automatisationsgrad

Es sind einige Tools erhältlich [HOMM2001], die jedoch nicht alle IDEF Methoden abdecken. Zumeist werden die Methoden IDEF0, IDEF1 und IDEF3 unterstützt, wobei die Toolunterstützung teilweise soweit geht, dass aus IDEF1 Diagrammen physikalische Datenmodelle erstellt werden können.

Tool-Support

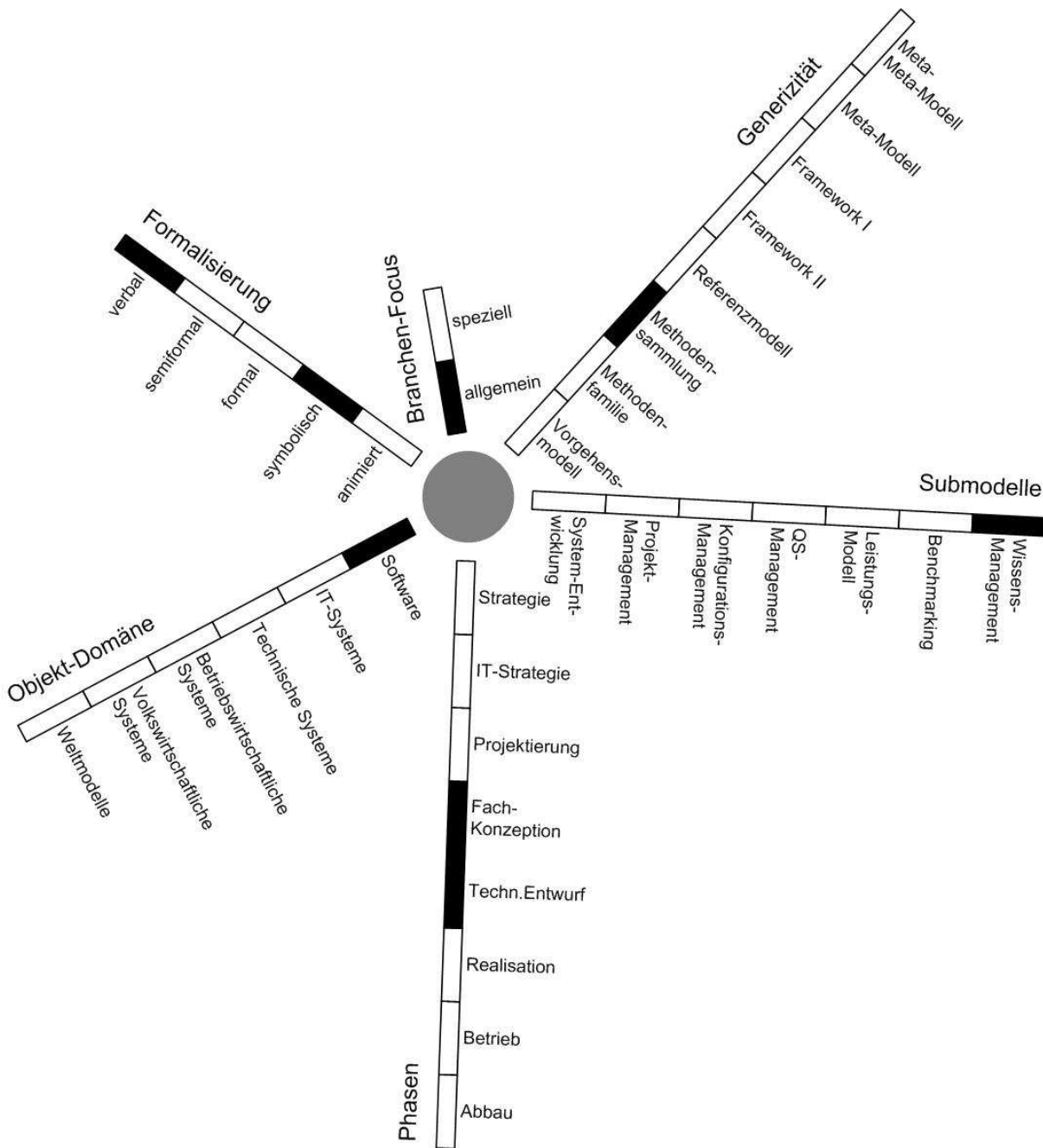
Die IDEF Methodensammlung besteht an sich nur aus einer Methodenauswahl, es gibt aber einzelne Tools, die bis zum generierenden Support ausgestaltet sind.

Bewertung

Die IDEF Methodensammlung war in den neunziger Jahren im U.S. amerikanischen Behörden, Militär und Industriebereich verbreitet, hat aber durch das Aufkommen von UML an Bedeutung verloren, zumal es teilweise Eingang in UML gefunden hat. Von den ursprünglich 14 angekündigten Methodenstandards wurden sieben veröffentlicht, davon der letzte vor etwa zehn Jahren.

Nichts desto trotz gibt es immer noch eine Reihe von Tools, die einzelne IDEF Methoden unterstützen.

Das folgende Sterndiagramm zeigt die Ausprägung der einzelnen definitorischen Kriterien.



Charakteristik IDEF

Literatur

[MAYER1992] Mayer: IDEF1 Information Modeling; A Reconstruction of the Original Air Force Wright Aeronautical Laboratory Technical Report AFWAL-TR-81-4023: Knowledge Based Systems, Inc., 1992.

[NORAN2003] Noran: Business Modelling: UML vs. IDEF: Griffith University, 2003.

[KBI2005] Knowledge Based Systems, Inc.: IDEF Family of Methods: <http://www.idef.com/>, 2005.

[HOMM2001] Hommes: Business Process Modelling Tools (By Technique): <http://is.twi.tudelft.nl/~hommes/toolsub.html>, 2001.

www.idef.com

4.2. UML

Historie und Motivation

Die Unified Modelling Language (UML) gründet sich auf die Zusammenführung der Notationen OMT (Object Modelling Technique) von J. Rumbaugh und OOD (Object Oriented Design) von G. Booch im Jahr 1996. Zwei Jahre später wird die UML von der Object Management Group (OMG) als Standard eingeführt und publiziert. Seit der Version 1.1 sind in die UML weiterhin die Ideen des OOSE (Object Oriented Software Engineering) von I. Jacobson eingeflossen.

Durch stetige Weiterentwicklung und durch die Beteiligung diverser Softwarehersteller ist die UML sehr weit verbreitet und liegt derzeit in der Version 2.0 vor.

Definition bzw. Einordnung

Die UML ist eine grafische Modellierungssprache für diverse Systeme. Ihr Haupteinsatzgebiet liegt in der Spezifikation bzw. dem Design von Software.

Die UML stellt keinerlei Informationen zur Vorgehensweise der Softwareentwicklung bereit. Sie ist lediglich eine Sammlung von Notationen für die Entwicklung.

Aufbau

Im Rahmen der UML ist eine Sammlung von Diagrammarten für die jeweils verschiedenen Einsatzzwecke enthalten. Unterschieden werden:

- General Purpose Diagram (Package-Ebene)
- Use Case Diagram (Anwendungsfalldiagramm)
- Class Diagram (Klassenmodell)
- Scenario Diagram
- State-Transition Diagram (Zustandsübergangsdigramm)
- Component Diagram (physikalische Systemebene)
- Deployment Diagram (Einsatzdiagramm)

Charakteristik im Detail

Generizität

Die UML ist eine Methodensammlung. Es existieren keinerlei Informationen hinsichtlich Ablauf bzw. Aufbau des Softwareentwicklungsprozesses, sondern es werden verschiedenste Diagramme zur Verwendung angeboten.

Phasenabdeckung

Als eine Methodensammlung ist es nicht möglich verschiedene Phasen zu erkennen. Eingesetzt wird die UML im Bereich des Softwaredesigns. Dies erstreckt sich über die Fachkonzeption sowie den technischen Entwurf.

Sub-Komponenten

Sub – Komponenten sind nicht vorhanden.

Objekt-Domäne

Die Diagramme der UML können für verschiedene Einsatzzwecke verwendet werden. Ihr Haupteinsatzgebiet liegt jedoch im Bereich der Softwareentwicklung.

Branchenfokus

Die UML ist nicht auf spezielle Softwaresysteme ausgerichtet. Ein allgemeiner Einsatz ist somit möglich.

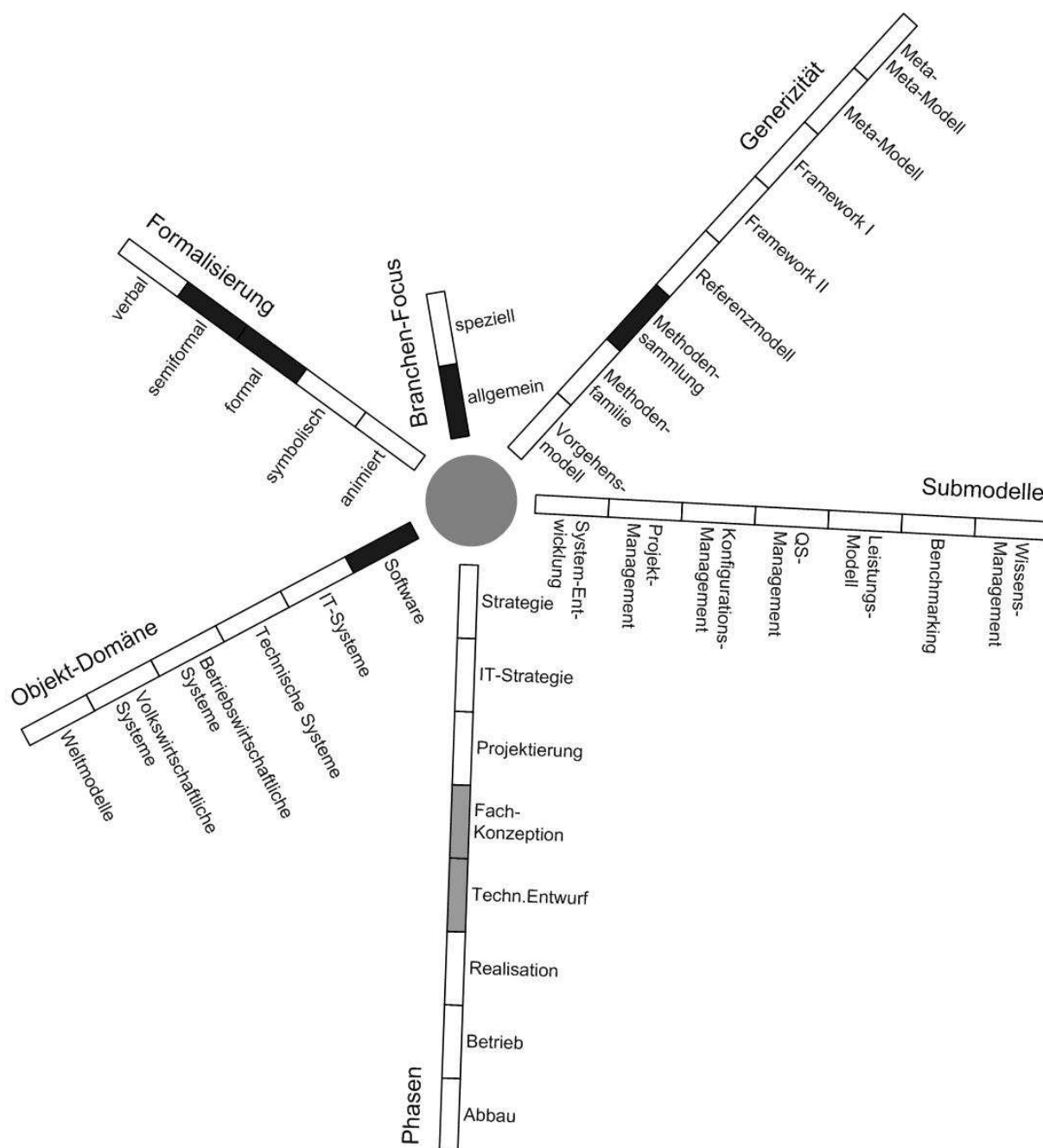
Formalisierung

Je nach Einsatz der Diagrammarten kann die Darstellung als formal oder semiformal angesehen werden.

Bewertung

Als universell einsetzbare Modellierungssprache für Softwaresysteme ist die UML sehr weit verbreitet und bildet den Standard für die Entwicklung von objekt- bzw. komponentenorientierten Systemen.

Das folgende Sterndiagramm zeigt die Ausprägung der einzelnen definitorischen Kriterien.



Charakteristik UML

Literatur

[BaWe99] Bannert, Gabriele; Weitzel, Martin : Objektorientierter Softwareentwurf mit UML. Addison Wesley Longman, 1999.

[FoSc00] Fowler, Martin; Scott, Kendall : UML konzentriert. Addison Wesley Longman, 2000.

4.3. ZACHMANN Framework

Historie und Motivation

1987 publizierte John Zachman einen damals völlig neuartigen Ansatz zur Systementwicklung. Anstatt die bei der Systementwicklung ablaufenden Prozesse als eine Reihe von Schritten aufzulisten, ordnete er sie rund um die Blickwinkel verschiedener Rollen an.

Diese Rollen waren der Planer (Geschäftsführer), die Besitzer der Firma, die System Analytiker, Designer und das System selbst.

Definition bzw. Einordnung

Beim Zachman-Framework handelt es sich um ein Framework I, also ein Ordnungsschema oder eine Checkliste, die nicht durch Tools unterstützt wird.








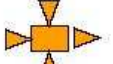
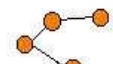
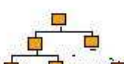
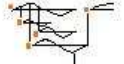

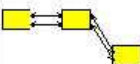
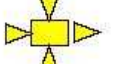
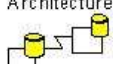
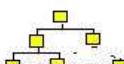




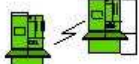
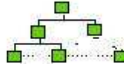








Es werden von Zachman keine Methoden vorgeschlagen, mit der die zu untersuchenden Aspekte der Systementwicklung betrachtet werden sollen, es wird nur vorgegeben, was aus welcher Sicht zu betrachten ist. Die Betrachtung soll dabei möglichst vollständig sein.

Aufbau

Im Laufe der Jahre erweiterte, veränderte und schrumpfte Zachman die ursprünglich eingeführten Rollen, sodass sie heute folgende Perspektiven umfassen: Planer, Besitzer, Designer, Builder, Sub-contractor und das System selbst.

Diese Rollen brachte Zachman in eine Matrix mit den Dingen, die aus jeder der Perspektiven zu betrachten wären. Diese Dinge beschreibt Zachman als das What, How, Where, Who, When und Why.

What, How und When werden oftmals und von vielen Methoden betrachtet. Die Betrachtung der Vernetzung im Where und der Gründe für das zu geschehende im Why waren hingegen neu. Eine Verdeutlichung Aufbaus wird in [Zach2004] gegeben.

	DATA <i>What</i>	FUNCTION <i>How</i>	NETWORK <i>Where</i>	PEOPLE <i>Who</i>	TIME <i>When</i>	MOTIVATION <i>Why</i>	
Objective/Scope <i>Contextual</i> <i>Role: Planner</i>	List of Things Important in the Business 	List of Core Business Processes 	List of Business Locations 	List of Important Organizations 	List of Events 	List of Business Goals/Strategies 	Objective/Scope <i>Contextual</i> <i>Role: Planner</i>
Enterprise Model <i>Conceptual</i> <i>Role: Owner</i>	Conceptual Data/Object Model 	Business Process Model 	Business Logistics System 	Work Flow Model 	Master Schedule 	Business Plan 	Enterprise Model <i>Conceptual</i> <i>Role: Owner</i>
System Model <i>Logical</i> <i>Role: Designer</i>	Logical Data Model 	System Architecture Model 	Distributed Systems Architecture 	Human Interface Architecture 	Processing Structure 	Business Role Model 	System Model <i>Logical</i> <i>Role: Designer</i>
Technology Model <i>Physical</i> <i>Role: Builder</i>	Physical Data/Class Model 	Technology Design Model 	Technology Architecture 	Presentation Architecture 	Control Structure 	Rule Design 	Technology Model <i>Physical</i> <i>Role: Builder</i>
Detailed Representations <i>Out of Context</i> <i>Role: Programmer</i>	Data Definitions 	Program 	Network Architecture 	Security Architecture 	Timing Definition 	Rule Specification 	Detailed Representations <i>Out of Context</i> <i>Role: Programmer</i>
Functioning Enterprise <i>Role: User</i>	Usable Data	Working Function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy	Functioning Enterprise <i>Role: User</i>

Zachmann Framework

Zusätzlich zu der beschreibenden Grafik gibt es das Original Paper von Zachman bei IBM zum Download in Form eines eingescannten Artikels [ZACH1987]. Weitergehende Informationen, sowie Unterstützung bei der Umsetzung kann direkt beim Zachman Institute www.zifa.com erworben werden.

Funktionen, Informationen, Prozesse

Der Einsatz des Zachman Frameworks bei der Systemplanung und Erstellung beläuft sich darauf, dass sichergestellt werden muss, dass alle zu betrachtenden Aspekte aus allen Perspektiven betrachtet werden. Welche Methoden dazu im einzelnen gewählt werden, steht frei.

Charakteristik im Detail

Generizität

Das Zachman Framework ist ein in der Methodenwahl sehr generisches Framework, das aber streng vorgibt, was mit den eingesetzten Methoden zu erreichen ist.

Phasenabdeckung

Durch Betrachtung des Systems aus den Blickwinkeln sowohl des Planers, des Konzeptionisten, als auch des technischen Entwurfs und der Realisierung werden viele Phasen der Systemerstellung abgedeckt. Phasen wie Projektmanagement oder Abbau werden allerdings auch im Zachman Framework nicht abgedeckt.

Sub-Komponenten

Schwerpunktmäßig dient das Zachman Framework der Systementwicklung, indem es jeden Aspekt des Systems aus verschiedensten Blickwinkeln betrachtet.

Objekt-Domäne

Das Zachman Framework betrachtet einen Betrieb sowohl aus Softwaresicht, als auch aus Sicht eines betriebswirtschaftlichen Systems mit Geschäftsprozessen und Personen, die diese abwickeln.

Branchenfokus

Einen bestimmten Branchenfokus gibt es durch die abstrakte Beschreibung eines Betriebes nicht.

Formalisierung

Der Formalisierungsgrad des Zachman Frameworks beschränkt sich auf eine Grafik und zusätzlichen erläuternden Text, ist also sehr niedrig.

QS-Beitrag

Zur Überprüfung der Qualität der erstellten Beschreibungen werden keine automatischen Hilfsmittel angeboten, diese liegt völlig in der Hand des Modellierers.

System-Zerlegung

Jede einzelne Beschreibung eines Aspektes aus einer Perspektive ist für sich in der Ebene der Aspekte ein abgeschlossener Teil, der jedoch über die Ebene der Perspektiven miteinander in Beziehung steht.

Ausgehend vom Planer bauen die weiteren Perspektiven auf den Ergebnissen der vorigen Perspektive auf.

Automatisationsgrad

Da das Zachman Framework ein Ordnungsschema ist, kann nur wenig automatisierte Unterstützung geboten werden, die über die Anbietung von für den jeweilig zu beschreibenden Aspekt geeigneten Methoden hinausgeht.

Werden allerdings ineinander überführbare Methoden zur Beschreibung verwendet, so kann etwa, was den Datenaspekt betrifft, ein hoher Automatisationsgrad durch ein gemeinsames Methoden-Repository erreicht werden.

Tool-Support

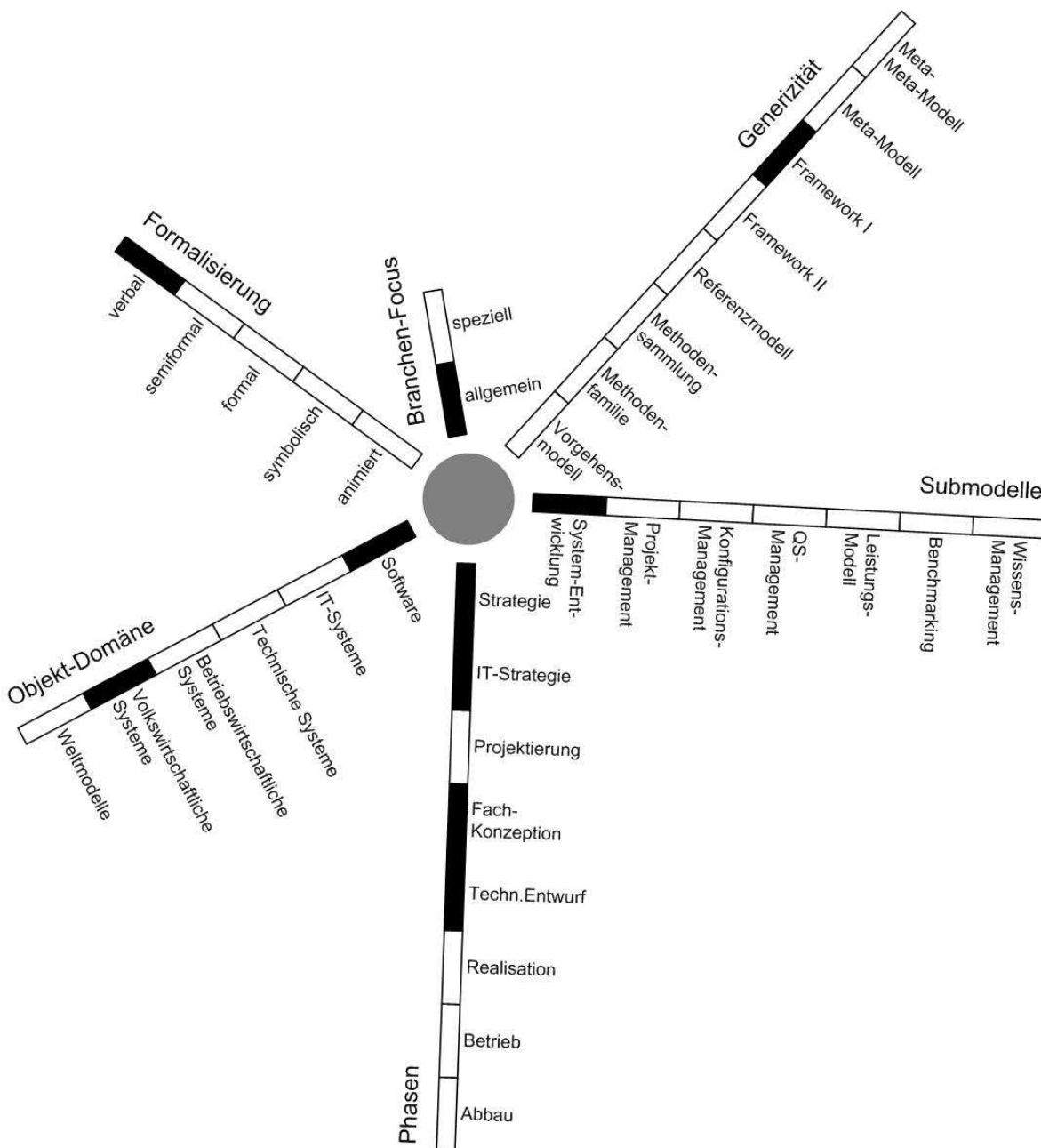
Tools benützen das Zachman Framework, um dem Modellierer eine Übersicht zu geben, welche Aspekte des Systems noch zu modellieren sind und können dann je nach angebotener Methode bis zur Codegenerierung Unterstützung anbieten. Diese Art der Unterstützung liegt dann aber eigentlich schon außerhalb des Bereichs des Zachman Frameworks, da dieses nur Vorschläge für die zu verwendenden Methoden anbietet.

Bewertung

Das Zachman Framework dient als Leitfaden, welche Aspekte aus welchen Perspektiven beleuchtet werden sollen, um eine gesamte Enterprise Architektur zu modellieren. Diese Art der Modellierung kann als Entscheidungsgrundlage genauso dienen, wie als Dokumentations- und Planungswerkzeug. Sie macht nachvollziehbar, welche Entscheidungen warum zu welcher technischen Realisierung geführt haben.

Durch die freie Methodenwahl bleibt das Zachman Framework dabei offen gegenüber neuen Modellierungsmethoden und erhält so seine Aktualität.

Das folgende Sterndiagramm zeigt die Ausprägung der einzelnen definitorischen Kriterien.



Charakteristik ZFW

Literatur

[ZACH2004] Zachman: Enterprise Architecture, A Framework: Zachman Institute, 2004.

[ZACH1987] Zachman: A Framework for Information Systems Architecture:
<http://www.research.ibm.com/journal/sj/263/ibmsj2603E.pdf>, 1987.

4.4. V-Modell 97

Historie und Motivation

Viele IT-Projekte der Bundesbehörden, insbesondere auch in militärischen Bereichen waren durch die Entwickler nicht ausreichend dokumentiert und damit nicht wartbar. Es entstanden große Abhängigkeiten von den ursprünglichen Entwicklern und Einzelpersonen. Manches Mal wurde aus diesen Gründen eine vollständige Neuentwicklung notwendig, obwohl bereits die bestehenden Softwaresysteme umfangreiche Gelder verschlangen. Das V-Modell entstand Ende der Achtziger bis Anfang der Neunziger Jahre für Militärprojekte der Bundesregierung, mit dem Ziel, eine gut dokumentierte und gut wartbare Software durch die Lieferanten entwickeln zu lassen und den hierfür anfallenden Mehraufwand durch Einsparungen in der Betriebs- und Wartungsphase zu überkompensieren. Darüber hinaus wurde mit der Entwicklung des V-Modells das Ziel verfolgt, militärische IT-Systeme mit größerer Erfolgswahrscheinlichkeit und Qualität zu entwickeln. Der Einsatz des V-Modells in Projekten hat insgesamt folgende Ziele:

- Minimierung der Projektrisiken
- Verbesserung und Gewährleistung der Qualität
- Eindämmung der Gesamtkosten über den gesamten Systemlebenszyklus
- Verbesserung der Kommunikation zwischen allen Beteiligten
- Absicherung der Wartbarkeit der entwickelter IT-Systeme für den gesamten Lebenszyklus.

Die Bundesregierung beauftragte die IABG mit der Entwicklung des V-Modells. Das V-Modell wurde in der ersten Version V-Modell 92 für militärische Projekte der Bundesregierung freigegeben und musste durch die Bundesbehörden und deren Lieferanten verpflichtend eingesetzt werden. In der weiteren Version V-Modell 97 – ebenfalls durch die IABG im Auftrag der Bundesregierung entwickelt – dehnte sich der Anwendungsbereich von reinen Softwaresystemen auf komplexe, aus Hard- und Software bestehende Systeme aus. 1997 wurde mit der Veröffentlichung des Entwicklungsstandards für IT-Systeme des Bundes das V-Modell 97 als Vorgabe für den Einsatz im gesamten zivilen und militärischen Bundesbereich gültig.

Durch die Ausdehnung auf die zivilen Bereiche der Bundesbehörden setzten nun auch Lieferanten außerhalb der Rüstungsindustrie das V-Modell ein. Über diesen Weg fand das V-Modell auch den Weg in die Industrie und wird direkt oder in angepasster Form durch privatwirtschaftliche Unternehmen eingesetzt – zunehmend auch freiwillig, da der Wertbeitrag zur Stabilisierung der Entwicklungsprozesse und zur Qualitätssicherung der Projektergebnisse erkannt wurde.

Auch international fand das V-Modell Verbreitung – direkt oder als weiterentwickelte bzw. angepasste Länderversionen. Beispielsweise wurde das V-Modell in Österreich fast unverändert übernommen. In der Schweiz wurde das Vorgehensmodell Hermes – basierend noch auf dem V-Modell 92 – entwickelt und mittlerweile als wesentlich weiterentwickelte Version „neoHERMES“ verfügbar.

Mit der zweiten Version des V-Modells (V-Modell 97) dehnte sich der Anwendungsbereich auch auf alle zivilen Bundesbehörden in Deutschland aus. In Österreich wurde das V-Modell 97 als IT-Bundesvorgehensmodell (IT-BVM) übernommen und an eigene Erfordernisse angepasst.

Mittlerweile setzen national wie international eine Vielzahl von privatwirtschaftlichen Unternehmen und Behörden das V-Modell erfolgreich ein oder haben eigene, angepasste Vorgehensmodelle davon abgeleitet.

Ausgangspunkt in früheren Modellen: Das V-Modell des Bundes baut auf Phasen-Konzepten des ursprünglichen V-Modells (nach Boehm) auf. Dieses V-Modell nach Boehm sieht die immer detail-

liertere Systemspezifikation und Zerlegung in Systemkomponenten auf dem linken, absteigenden Ast des „V“ vor. Auf der untersten Ebene, der Basis des „V“ findet die Realisierung der Systemkomponenten statt. Auf dem rechten, ansteigenden Ast des „V“ findet die Integration der Systemkomponenten zum System sowie der Test Systemtest, Abnahmetest statt.

Definition bzw. Einordnung

V-Modell, V-Modell 97, Entwicklungsstandard des Bundes zur Entwicklung von IT-Systemen (EStBIT)

Das V-Modell ist ein frei verfügbares, kostenlos nutzbares und vollständiges Vorgehensmodell zur Erstellung von IT-Systemen, Software und Hardware. Es deckt alle Phasen der Projektdurchführung zur Entwicklung von IT-Systemen ab. Es ist sowohl geeignet für strukturierte Entwicklungsmethoden für prozedurale Programmiersprachen und auch für objektorientierte Entwicklung, objektorientierte Programmiersprachen. Für agile Entwicklungsprozesse ist das V-Modell nicht oder nur bedingt, d.h. in Teilbereichen z.B. der SW-Realisierung/Programmierung einsetzbar.

Das V-Modell wurde in der ersten Version V-Modell 92 für militärische Projekte der Bundesregierung freigegeben und musste durch die Bundesbehörden und deren Lieferanten verpflichtend eingesetzt werden. In der weiteren Version V-Modell 97 – ebenfalls durch die IABG im Auftrag der Bundesregierung entwickelt – dehnte sich der Anwendungsbereich von reinen Softwaresystemen auf komplexe, aus Hard- und Software bestehende Systeme aus. 1997 wurde mit der Veröffentlichung des Entwicklungsstandards für IT-Systeme des Bundes das V-Modell 97 als Vorgabe für den Einsatz im gesamten zivilen und militärischen Bundesbereich gültig.

Aufbau

Bestandteile, Darstellungsmittel, Symbolik, Symbolvarianten, Struktur, Referenzen

Das V-Modell 97 besteht aus vier Submodellen:

- Projektmanagement
- Konfigurationsmanagement
- Qualitätsmanagement
- Systementwicklung.

In diesen monolithischen Submodellen sind die im V-Modell vorgeplanten Aktivitäten und Projektergebnisse (Produkte) sowie die Rollen enthalten. Die Darstellungen der Abläufe sind für einen schnellen Überblick grafisch dargestellt. Ansonsten sind die Inhalte des V-Modells überwiegend verbal beschrieben.

Im Einzelnen wurden dabei vom Bundesministerium der Verteidigung, Bundesamt für Informationsmanagement und Informationstechnik der Bundeswehr IT-AmtBw A5, und vom Bundesministerium des Innern, Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung (BMI-KBSt), die folgenden Dokumente als Allgemeiner Umdruck (AU) Nr. 250-252 beziehungsweise als Schriftenreihe der KBSt Band 27/1 und 27/2, zur Verfügung gestellt:

- Vorgehensmodell (AU 250)
- Teil 1: Regelungsteil (KBSt Band 27/1)
- Teil 2: Behördenspezifische Ergänzungen (KBSt Band 27/2)
- Teil 3: Handbuchsammlung (KBSt Band 27/2)
- Methodenzuordnung (AU 251)
- Funktionale Werkzeuganforderungen (AU 252).

Für die Bundesbehörden verbindlich ist nur der Regelungsteil AU250 Teil 1 sowie der für die Behörden zutreffende Teil der behördenspezifischen Ergänzungen AU250 Teil 2. Die Handbuchsammlung, die funktionellen Werkzeuganforderungen sowie die Methodenzuordnungen haben lediglich Empfehlungscharakter.

Das V-Modell 97 kann über die Webseiten der KBSt (Koordinierungs- und Beratungsstelle des Bundes), der IABG oder des ANSSTAND e.V. (Verein der Anwender des V-Modells) bezogen werden. Das V-Modell 97 ist auch in englischer Sprache erhältlich. Des Weiteren ist das V-Modell

97 auch als Online-Version GDPA über die Webseite der Uni Bremen in deutscher und englischer Sprache verfügbar und hat über die vollständige Verlinkung eine sehr gute inhaltliche Aufbereitung.

Funktionen, Informationen, Prozesse

Input, Output, Hierarchische Zerlegung, Vernetzung, Ablauf Des Einsatz, Vorbereitung, Abschluss

Die Entwicklungsstrategie eines nach V-Modell durchzuführenden IT-Projektes kann klassisch einem Wasserfallmodell folgen, kann aber auch inkrementellen bzw. iterativen Entwicklungsstrategien folgen. Dabei ist die inkrementelle auch die empfohlene Strategie.

Es bestehen Vorgaben hinsichtlich der Reihenfolge der durchzuführenden Aktivitäten, die in sogenannten Produktflüssen dargestellt sind. Für jede Aktivität sind Eingangsprodukte, deren Status sowie die Ausgangsprodukte und deren Status in einer Produktflusstabelle definiert.

Vor einer Anwendung des V-Modells ist eine Anpassung des V-Modells an das Projekt erforderlich. Diese Anpassung wird als Tailoring bezeichnet. Das Tailoring erfolgt auf der Ebene der Produkte und Aktivitäten im Normalfall subtraktiv, d.h. die nicht benötigten Produkte und Aktivitäten werden unter Angabe von Streichgründen weggelassen. Alternativ dazu können Standard-Tailoring-Matrizen angewendet werden. Anschließend werden die Instanzen der Produkte und Aktivitäten erzeugt, und festgelegt, welche Produkte einer QS-Prüfung unterzogen werden.

Charakteristik im Detail

Generizität

Das V-Modell ist ein vollständiges Vorgehensmodell und kann grundsätzlich auch ohne Anpassungen für IT-Projekte genutzt werden. Darüber hinaus ist es ein Framework zur Ableitung organisationsspezifischer Vorgehensmodelle und ist als Referenzmodell einsetzbar. Das V-Modell enthält eine umfangreiche Methodensammlung.

Phasenabdeckung

Das V-Modell unterstützt nur die Phasen der (auftragnehmerseitigen) Durchführung von IT-Projekten. Nicht einsetzbar ist es für den Entwurf einer betriebswirtschaftlichen, unternehmerischen Strategie oder für die Ableitung / Definition einer IT-Strategie.

Sub-Komponenten

Das V-Modell deckt Wissensmanagement; Benchmarking und Leistungsmodell nicht ab, alle anderen Sub-Komponenten werden vollständig abgedeckt.

Objekt-Domäne

Das V-Modell unterstützt die Erstellung von IT-Systemen inkl. der erforderlichen Hard- und Software.

Branchenfokus

Das V-Modell ist branchenunabhängig einsetzbar.

Formalisierung

Das V-Modell verfügt über eine gewisse grafische Symbolik, ansonsten überwiegt die verbale Beschreibung.

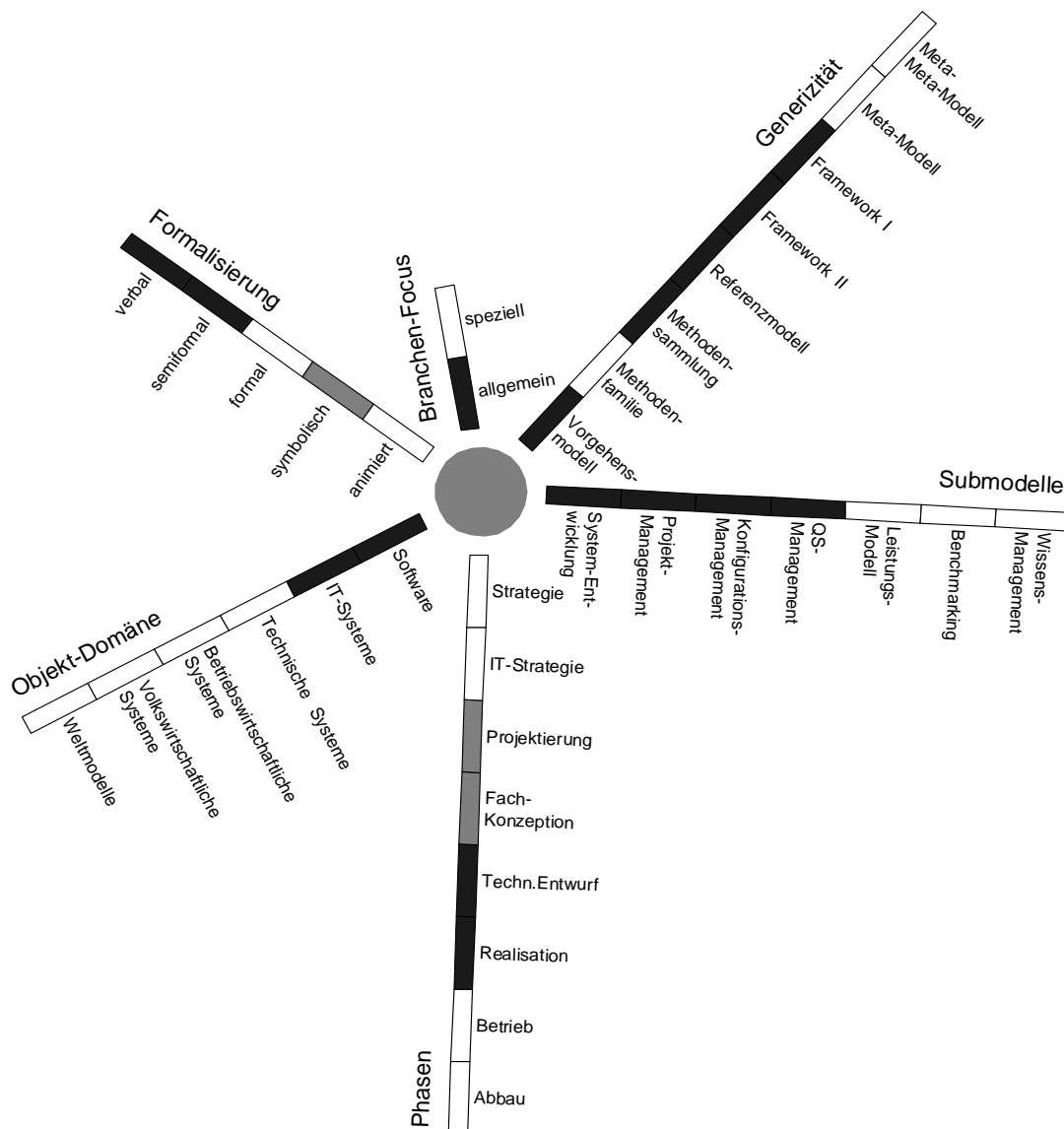
Bewertung

Erfahrungen, Kosten, Nutzen, Maßstäbe, Skalen, Akzeptanzstudien, Toolunterstützung, Implementierungsstatistiken

Das V-Modell 97 ist in vielen Unternehmen und Behörden im Einsatz. Insbesondere größere Unternehmen und Behörden haben sich aus dem V-Modell 92 organisationsspezifisch angepasste Vorgehensmodelle abgeleitet. Trotz seines teilweise negativen Rufes (Schwerfälligkeit, Dokumentationslastig) setzen es teilweise auch KMU ein. Bei kleinen Projekten ist das V-Modell aufgrund der mangelhaften Skalierbarkeit schlecht einsetzbar - es wäre oversized.

Da das V-Modell 97 kostenfrei verfügbar ist, kann jedermann das V-Modell einsetzen oder für die Entwicklung eigener Vorgehensmodelle einsetzen.

Das folgende Sterndiagramm zeigt die Ausprägung der einzelnen definitorischen Kriterien.



Charakteristik V97

Literatur

W. Dröschel / M. Wiemers: Das V-Modell 97

G. Versteegen: V-Modell 97 in der Praxis

4.5. V-Modell XT (Version 1.0)

Historie und Motivation

Zielsetzung, Ausgangspunkt in früheren Modellen, Autorenkreis, Hersteller, Projekt, Forschungsvorhaben, Standardisierungsbestrebungen

Das V-Modell 97 war teilweise technisch veraltet und musste erneuert werden. Die Bundesregierung beauftragte die TU München sowie Industriepartner (EADS, IABG, Siemens, 4soft) mit der Weiterentwicklung des V-Modells auf der Basis des Entwicklungsstandards des Bundes für IT-Projekte (EStBIT). Dieses Entwicklungsteam „WEIT“ integrierte erfahrene Reviewer als Qualitäts-

sicherungsinstanz und führte die Weiterentwicklung des V-Modells kosten-, termin- und qualitätsgerecht durch. Im Februar 2005 wurde das V-Modell XT in der Version 1.0 veröffentlicht und wurde für neue IT-Projekte des Bundes als verbindlich erklärt.

Im Rahmen der Weiterentwicklung zur Version XT wurden folgende Ziele verfolgt:

- Verbesserung der Unterstützung von Anpassbarkeit, Anwendbarkeit, Skalierbarkeit und Änder- und Erweiterbarkeit des V-Modells
- Berücksichtigung des neuesten Stands der Technologie und Anpassung an aktuelle Vorschriften und Normen
- Erweiterung des Anwendungsbereiches auf die Betrachtung des kompletten Lebenszyklus im Rahmen von Entwicklungsprojekten
- Einführung eines organisationsspezifischen Verbesserungsprozesses für Vorgehensmodelle.

Obwohl diese Anforderungen bei der Komplexität des V-Modells eine große Herausforderung für das Entwicklerteam darstellte, gelang es ihnen, das V-Modell inhaltlich zu verbessern und zu erweitern, besser handhabbar und leichter anwendbar zu gestalten.

Das überarbeitete V-Modell XT unterstützt zukünftig neben den sehr häufig auftretenden Migrationen auch die Entwicklung von Hardware sowie von komplexen Systemen (bestehend aus Hard- und Software). Darüber hinaus bietet das V-Modell XT eine ganzheitliche Sicht auf alle Phasen der Systementwicklung. Es bildet somit nicht mehr nur die reine Durchführungsphase von IT-Projekten ab, sondern:

- das V-Modell XT unterstützt nun auch die Planungsphasen sowie die Ausschreibung von IT-Systemen, so dass die Auftraggeberseite hervorragend vom V-Modell XT unterstützt wird
- das V-Modell XT unterstützt die während der Entwicklung auch die Planung der Inbetriebnahmephase, der Betriebsphase, Wartung und Pflege, etc. bis hin zur Außerbetriebnahme, so dass der komplette Systemlebenszyklus abgebildet ist. Dabei ist jedoch nur die Planung und Vorbereitung des Betriebs, der Wartung und Pflege usw. abgedeckt, nicht jedoch die konkrete Durchführung dieser Phasen
- im V-Modell ist auch ein Prozessmodell zur Anpassung und Einführung des V-Modells selbst hinterlegt.

Die neue Version XT des V-Modells erneuert die gesamte Struktur und verbessert damit die Anwendbarkeit massiv. Gleichzeitig wurde Bewährtes vom V-Modell 97 übernommen.

Definition bzw. Einordnung

Bezeichnung, Abgrenzung, Varianten, Kurzzeichen, Klassifikation,

V-Modell, V-Modell XT (eXtreme Tailoring), Entwicklungsstandard des Bundes zur Entwicklung von IT-Systemen (EStBIT)

Aufbau

Bestandteile, Darstellungsmittel, Symbolik, Symbolvarianten, Struktur, Referenzen

Die monolithischen Submodelle des V-Modells 97 werden im V-Modell XT durch modulare, inhaltlich aufeinander aufbauende Vorgehensbausteine ersetzt. Die Vorgehensbausteine kapseln inhaltlich zusammengehörende Produkte, Aktivitäten und Rollen. Die Vorgehensbausteine stellen die Tailoringebene des V-Modells XT dar. Der Vorgehensbaustein ist die zentrale Einheit des Tailoring im V-Modell XT. Ein V-Modell-Anwender kann das V-Modell projektspezifisch anpassen, ohne dass er die in den einzelnen Bausteinen enthaltenen Aktivitäten, Produkte und Rollen bereits im Detail verstanden haben muss. Ein Vorgehensbaustein enthält diejenigen Produkte, Aktivitäten und Rollen, die unter Tailoring-Gesichtspunkten inhaltlich zusammen gehören. Die Auswahl der erforderlichen Vorgehensbausteine sowie die Auswahl der situativ passenden Projektdurchführungsstrategie erfolgt über die Auswahl von vordefinierten Projektmerkmalen. Das Tailoring ist durch den hinterlegten Tailoringmechanismus automatisierbar.

Die Projektdurchführungsstrategien beschreiben die möglichen Reihenfolgen von zu durchlaufenden Entscheidungspunkten (quality gates, milestones). Den Entscheidungspunkten ist die Fertigstellung von Produkten zugeordnet. Hierüber können die Vorgaben bezüglich einer Reihenfolge der Produktbearbeitung bzw. -fertigstellung abgeleitet werden. Die Produktflüsse des V-Modells 97 sind entfallen.

Die Struktur des V-Modells ist durch das Meta-Modell beschrieben. Zentrale Elemente sind die zu erstellenden Produkte als die gewünschten Projektergebnisse. Ausgehend vom Produkt bestehen Beziehungen zu Aktivitäten, die ein Produkt bearbeiten sowie zu Rollen die für ein Produkt verantwortlich sind. Darüber hinaus gibt es Abhängigkeit zwischen Produkten, inhaltliche und erzeugende Produktabhängigkeiten. Über die erzeugenden Produktabhängigkeiten wird ein automatisiertes Erzeugen und Instantiieren von Produkten ermöglicht (ein Systemelement erzeugt automatisch die erforderlichen QS-Produkte zur Qualitätssicherung) und es wird ein Tailoring auch während der Projektlaufzeit ermöglicht, das sogenannte dynamische Tailoring.

Das V-Modell ist in einer strukturierten Dokumentation beschrieben, wobei die Teile 1 und 2 einführenden Charakter haben, die restlichen Teile dienen eher als Nachschlagewerk:

- Teil 1 – Grundlagen des V-Modells
- Teil 2 – eine Tour durch das V-Modell
- Teil 3 – V-Modell-Referenz Tailoring
- Teil 4 – V-Modell-Referenz Rollen
- Teil 5 – V-Modell-Referenz Produkte
- Teil 6 – V-Modell-Referenz Aktivitäten
- Teil 7 – V-Modell-Referenz Konventionsabbildungen
- Teil 8 – Vorlagen
- Teil 9 – Anhang.

Die Dokumentation ist verfügbar in einer PDF-, HTML sowie in einer Word-Version. Darüber hinaus liegt das V-Modell vollständig in XML-Form vor. Die Struktur der XML-Datei ist als XML-Schemata hinterlegt. Die anderen Versionen des V-Modells (PDF, Word, HTML) wurden aus der XML-Datei erzeugt.

Darüber hinaus ist bereitgestellt:

- eine HTML-Version des V-Modells
- eine interaktive Lerntour
- zielgruppenspezifische Schulungsunterlagen
- ein kostenloses OpenSource-Tool „Editor“, mit dem das V-Modell organisationsspezifisch angepasst und gepflegt werden kann
- ein kostenloses OpenSource-Tool „Projektassistent“ mit dem das Tailoring automatisiert unterstützt wird und die anschließende grundsätzliche Projektplanung durchgeführt werden kann. Die Ergebnisse können als projektspezifisches V-Modell nach PDF, Word, HTML sowie als Projektplan nach MS-Project oder Ganttproject (OpenSource) exportiert werden.

Funktionen, Informationen, Prozesse

Die Sichten von Auftraggeberseite und Auftragnehmerseite werden im V-Modell XT getrennt.

Die Projektdurchführungsstrategie eines nach V-Modell durchzuführenden IT-Projektes richten sich nach Projektgegenstand, Projektrolle und weiteren Projektmerkmalen. Es existieren für Systementwicklungsprojekte mehrere Projektdurchführungsstrategien:

- für die auftraggeberseitige Durchführung von IT-System existiert nur eine Projektdurchführungsstrategie: Vergabe und Durchführung von Systementwicklungsprojekten (AG).
- Auftragnehmerseitig kommen folgende Projektdurchführungsstrategien in Frage:
 - Inkrementelle Systementwicklung
 - Komponentenbasierte Systementwicklung

- Agile Systementwicklung
- Wartung und Pflege von Systemen
- Weiterentwicklung und Migration von Altsystemen

Darüber hinaus existiert eine Projektdurchführungsstrategie für Prozessverbesserung von Systementwicklungsprozessen, d.h. Inhalt derartiger Projekte ist die Anpassung und Einführung des V-Modells selbst bzw. die laufende Verbesserung eines eingeführten Vorgehensmodells: Einführung und Pflege eines organisationsspezifischen Vorgehensmodells.

Die Projektdurchführungsstrategien und die darin verknüpften Entscheidungspunkte geben die Reihenfolge der Produktfertigstellung und somit die grundlegende Struktur des Projektverlaufs vor. Die detaillierte Projektplanung und -steuerung wird auf der Basis der Bearbeitung und Fertigstellung von Produkten durchgeführt. Für jedes Produkt ist eine Rolle verantwortlich. Die Produktqualität ist durch definierte Anforderungen an das Produkt und explizite Abhängigkeiten zu anderen Produkten überprüfbar.

Charakteristik im Detail

Generizität

Das V-Modell ist ein vollständiges Vorgehensmodell und kann grundsätzlich auch ohne Anpassungen für IT-Projekte genutzt werden. Darüber hinaus ist es ein Framework zur Ableitung organisationsspezifischer Vorgehensmodelle und ist als Referenzmodell einsetzbar. Als Methodensammlung kann es nur bedingt bezeichnet werden, da es wohl auf Methoden verweist, diese aber nicht vollständig beschreibt und auch noch nicht konsistent ist. Die Methodenbeschreibungen haben sich gegenüber der Version 97 des V-Modells qualitativ deutlich verschlechtert, was sicher auch am Pflegeaufwand für ständiges Updating liegt, um auf dem jeweils aktuellen technologischen Stand zu bleiben.

Phasenabdeckung

Das V-Modell unterstützt alle Phasen bei der Vorbereitung und Durchführung von IT-Projekten. Nicht einsetzbar ist es für den Entwurf einer betriebswirtschaftlichen, unternehmerischen Strategie oder für die Ableitung / Definition einer IT-Strategie.

Phasen des Systemlebenszyklus wie Betrieb und Abbau werden zwar hinsichtlich der Vorausplanung dieser Phasen während der Entwicklung unterstützt. Während der Phasen Betrieb und Abbau findet jedoch keine Unterstützung statt. Schnittstellen zum hierfür verbreitetsten Prozessmodellen (ITIL) sind ansatzweise vorhanden.

Sub-Komponenten

Das V-Modell deckt Wissensmanagement und Benchmarking nicht ab, alle anderen Sub-Komponenten werden vollständig abgedeckt. Zu üblichen Prozessmodellen, die für Benchmarking von Vorgehens-(prozess-)modellen verbreitet eingesetzt werden, existiert eine Konventionsabbildung (CMMI). Es fehlt noch eine Konventionsabbildung zu ISO 15504 (Spice).

Objekt-Domäne

Das V-Modell unterstützt die Erstellung von IT-Systemen inkl. der erforderlichen Hard- und Software. Zur Erstellung von komplexen technischen Systemen ist es nach Ansicht der Autoren bedingt einsetzbar, es fehlen System-zerlegungsmöglichkeiten in mehrere komplexe technische (Teil-)Systeme.

Branchenfokus

Es existiert kein Branchenfokus, das V-Modell ist in allen Branchen einsetzbar.

Formalisierung

Das V-Modell verfügt über eine gewisse grafische Symbolik, ansonsten überwiegt die verbale Beschreibung. Da das V-Modell vollständig in einer XML-Datei vorliegt, ist das V-Modell semiformal beschrieben.

Bewertung

Erfahrungen, Kosten, Nutzen, Maßstäbe, Skalen, Akzeptanzstudien, Toolunterstützung, Implementierungsstatistiken

Das V-Modell XT baute auf den umfangreichen Einsatzerfahrungen des V-Modells 97 auf. Aufgrund der Weiterentwicklung ist es leichter anwendbar geworden. Insbesondere die integrierte Toolunterstützung führt in Verbindung mit den automatisierten Tailoringmöglichkeiten zu einem schnellen Einsatz des V-Modells. Anwendungshemmnisse und Anwendungslücken wurden beseitigt. Bei Auftragnehmern wird die fachliche Erweiterung und die Integration agiler Vorgehensweisen zu verbreiterten Anwendungsmöglichkeiten führen. Besonders zu betonen ist die integrierte Werkzeugunterstützung. Das V-Modell stellt ein Tailoringwerkzeug „Projektassistent“ bereit, mit dem Tailoring und Projektgrobplanung durchgeführt werden können. Kommerzielle Werkzeuge werden durch die formale Bereitstellung des V-Modells als XML-Datei sowie durch die automatisierbaren Tailoringregeln unterstützt. Es existieren bereits kommerzielle Tools, die das V-Modell unterstützen.

Für Auftraggeber bietet das V-Modell mit dem Release XT erstmals eine vollständige und umfangreiche Unterstützung an. Kein anderes verfügbares V-Modell verfügt über so umfangreichen Support für Auftraggeber. Die Auftraggeber profitieren mit dem V-Modell erstmals über ein Prozessmodell für die auftraggeberseitige Projektdurchführung und über eine Sammlung aller hierfür erforderlichen Aktivitäten, Produkte, Rollen und Methoden. Das V-Modell unterstützt darüber hinaus öffentliche Auftraggeber durch umfangreiche Verweise auf existierende Vergabevorschriften und Vertragsvorlagen.

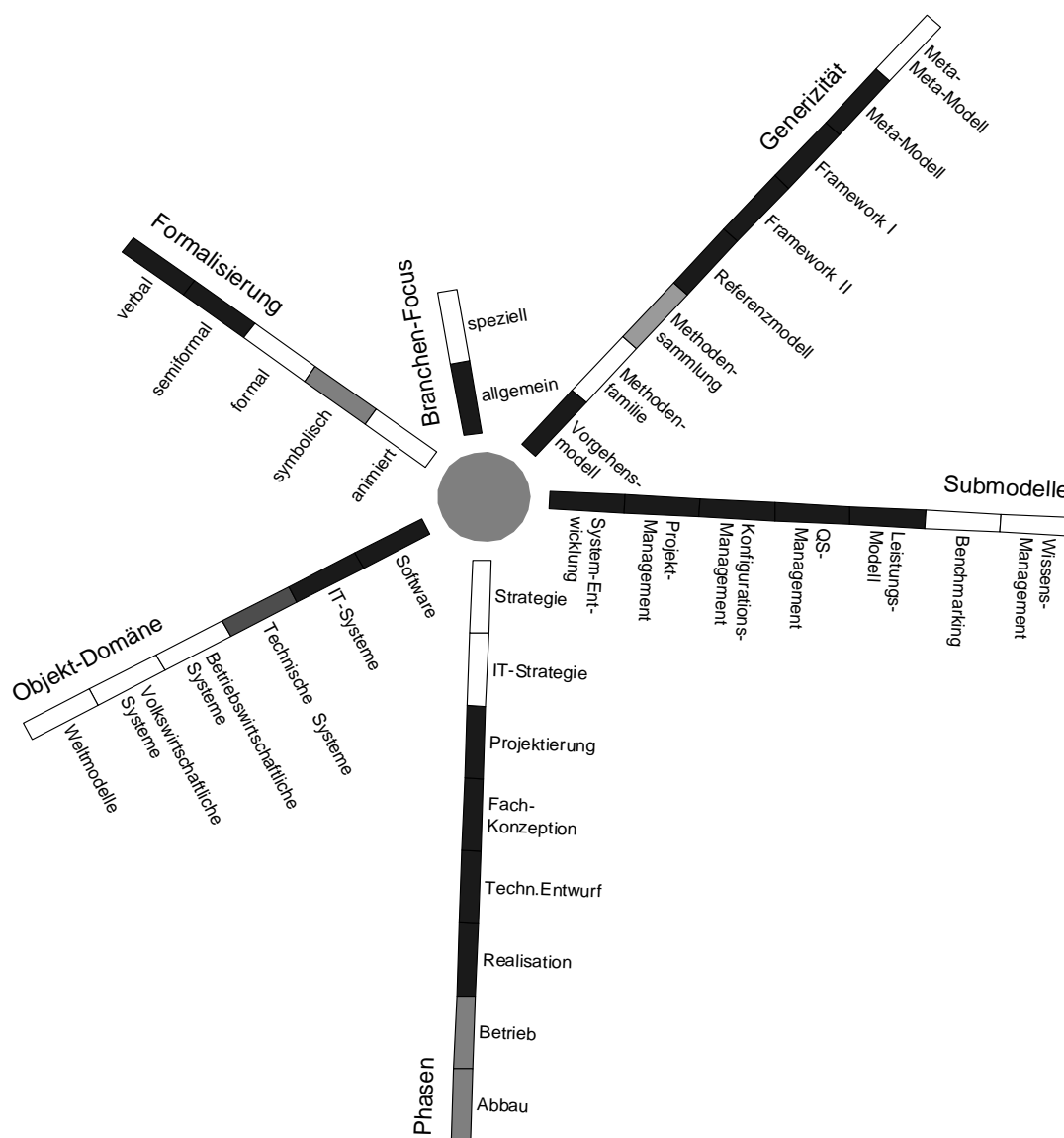
Das V-Modell enthält neben Vorgehens-(prozess-)modellen für die Durchführung von Systementwicklungsprojekten von Auftraggebern und Auftragnehmern auch ein Vorgehens-(prozess-)modell für die Anpassung und Einführung des V-Modells selbst.

Für diese Anpassung des V-Modells stellt es eine Toolunterstützung „Editor“ bereit, mit dem das V-Modell angepasst und ergänzt werden kann und das erzeugte angepasste V-Modell in unterschiedliche Ausgabeformate exportiert werden kann.

Die kostenfrei mitgelieferten OpenSource-Tools „Projektassistent“ und „Editor“ werden den Einsatz und die Verbreitung des V-Modells stark beschleunigen, da die Einstiegshürden extrem gesenkt wurden. Da das V-Modell 97 kostenfrei verfügbar ist, kann jedermann das V-Modell einsetzen oder für die Entwicklung eigener Vorgehensmodelle einsetzen. Das V-Modell ist zur Zeit in Deutsch verfügbar, eine englische Version befindet sich in Vorbereitung.

Weltweit existiert zur Zeit kein vergleichbares Vorgehensmodell, die Bundesrepublik Deutschland und die Projektpartner des Projektteams „WEIT“ haben etwas einmaliges geschaffen. Das V-Modell XT wird sich deutschlandweit, evtl. europa- oder sogar weltweit durchsetzen und einen Namen machen. Bestrebungen zu einer europaweiten Standardisierung dürften diesen Trend erheblich beschleunigen.

Das folgende Sterndiagramm zeigt die Ausprägung der einzelnen definitorischen Kriterien.



Charakteristik VMX

Literatur

Prof. Dr. Andreas Rausch, Stephan Höppner: „V-Modell XT – eine Einführung“. In: Software-Qualitätsmanagement in Theorie & Praxis, Band 3 – Methoden, Vorgehen und Werkzeuge für die Software-Qualitätssicherung, hrg. von Roland Petrasch, Stephan Höppner, Logos Verlag, 2005

Ulrich von Hagen: (siehe www.v-modell-xt.de)

Reinhard Höhn, Stephan Höppner: „Das V-Modell, Fortsetzung einer Erfolgsstory“, erschienen in OCG-Journal Dez 2004

4.6. RUP

Historie und Motivation

Grundlegend fließen im Rational Unified Process (kurz: RUP) zwei verschiedene Entwicklungen zusammen. Einerseits die von der Firma Rational praktizierte Erarbeitung eines iterativen und inkrementellen Softwareprozesses mit architekturzentrierter Vorgehensweise und andererseits der von Ivar Jacobson entwickelte Objectory Process. Der von der gleichnamigen Firma vertriebene Prozess Objectory basiert auf Erfahrungen bei der Entwicklung von Kommunikationssystemen bei Ericsson. Zentrales Konzept bildete die Kommunikation des in Blöcke aufgeteilten Systems. Zur Darstellung

dienen Use-Case- Diagramme (deutsch: Anwendungsfalldiagramme) und weitere das objektorientierte Design unterstützende Darstellungen, welche zur heutigen UML große Ähnlichkeit aufweisen.

Im Jahre 1995 wurde Objectory von Rational übernommen und es entstand durch die Zusammenarbeit von Grady Booch, James Rumbaugh und Ivar Jacobson der Rational Objectory Process (1996). Unter Einbeziehung der Vereinheitlichungsbestrebungen der UML und unter dem Einfluss weiterer Firmen veröffentlichen die oben genannten Autoren den Unified Process (1998). Ein eher generischer Prozess von dem der Rational Unified Process eine spezielle und detaillierte Instanz ist. Der RUP wird ebenfalls im Jahr 1998 erstmalig unter diesem Namen veröffentlicht.

Seit 2003 gehört Rational zum IBM Konzern, welcher den Rational Unified Process weiter unter diesem Namen entwickelt und vertreibt.

Definition bzw. Einordnung

Der Rational Unified Process (kurz RUP) ist ein kommerziell vertriebenes Vorgehensmodell zur Erstellung von Software. Der gesamte RUP kann als Produkt inklusive Wissensdatenbank von IBM / Rational bezogen werden. Ebenso existiert eine Online Version zur Unterstützung der Teammitglieder. Der RUP selbst umfasst keine Software. Durch Produkte der Rational Suite kann eine hohe Werkzeugunterstützung erreicht werden.

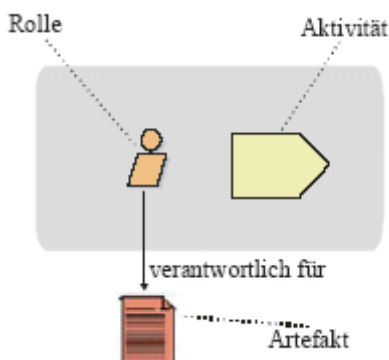
Aufbau

Sowohl die Online- als auch die Papierversion basieren auf neun grundlegenden Büchern mit eigenen Zielen:

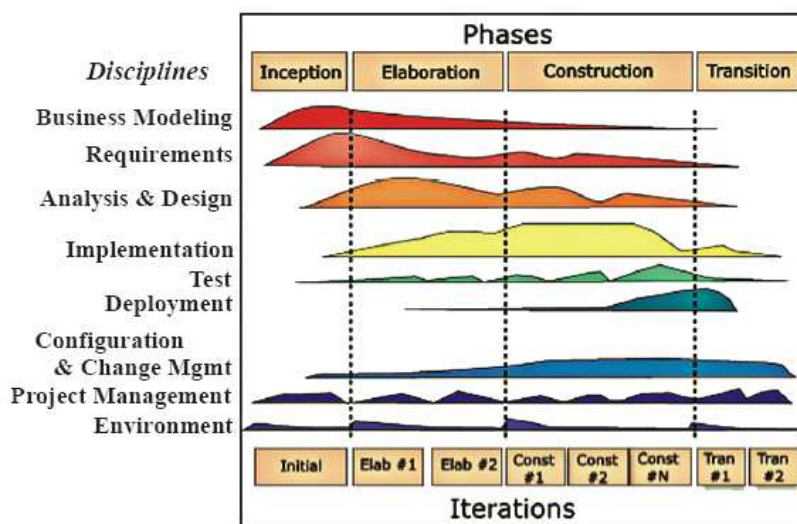
- RUP Einführung ([Kruc1999])
- RUP Prozesshandbuch
- RUP Prozesskonfiguration
- RUP Prozess
- RUP Werkzeugmentoren
- RUP Richtlinien
- RUP Artefakte
- RUP ADA Programmierrichtlinien
- RUP C++ Programmierrichtlinien.

Durchgehend durch die entsprechenden Dokumentationen stehen vier Modellierungselemente zur Beschreibung des Prozesses zur Verfügung. Eine Rolle legt dabei die Verantwortlichkeit einer Person oder einer Personengruppe für eine bestimmte Anzahl von zusammengehörenden Tätigkeiten, den so genannten Aktivitäten fest. Unter Artefakten versteht man alle Produkte, die im Prozessverlauf durch eine Rolle erzeugt, geändert oder kontrolliert werden.

Als viertes Symbol wird der Workflow verwendet, der als Abfolge von Aktivitäten, mit einem konkret nutzbaren Ergebnis, zu verstehen ist.



Die zugrunde liegende Idee der iterativen und inkrementellen Softwareentwicklung in Kombination mit den weiteren Bestandteilen der Bücher lässt sich folgendermaßen darstellen:



Phasenmodell nach RUP

Die horizontale Achse repräsentiert dabei die dynamischen Aspekte des Lebenszyklus. Im zeitlichen Verlauf werden dabei die einzelnen Phasen Konzeptualisierung (Inception), Entwurf (Elaboration), Konstruktion (Construction) und Übergang (Transition) durchlaufen. Jede einzelne Phase besteht aus mindestens einer Iteration, die wiederum wie ein sequenzielles Projekt durchgeführt wird.

Die vertikale Achse zeigt die statischen Aspekte des Prozesses mit den sechs Hauptdisziplinen Geschäftsprozessmodellierung (Business Modeling), Anforderungsanalyse (Requirements), Analyse und Design (Analysis & Design), Implementierung (Implementation), Test, Verteilung (Deployment) und den drei Unterstützungsdisziplinen Projektmanagement, Konfigurations- und Änderungsmanagement (Configuration & Change Management) und der Umgebungsdisziplin (Environment). Bezeichnet werden diese Disziplinen im RUP als Workflows.

Funktionen, Informationen, Prozesse

Die Vielzahl von Aktivitäten eines Prozesses sind im RUP durch verschiedene Typen von Workflows unterteilt. Dies sind Core- und Iterations- Workflows sowie Workflow- Details. Core-Workflows sind die Elementaren Bestandteile des Prozesses. Hier unterscheidet man die oben angeführten neun Verschiedenen.

Iterations- Workflows bieten einen anderen Blickwinkel auf die Entwicklung, in dem sie sich mehr auf die Aspekte einzelner Iterationen beziehen. Hierfür sind im RUP typische Beispiele vorhanden. Jedoch ist ihr Einsatz eher aus pädagogischen Gründen von Interesse.

Input und Output einer Aktivität sowie weitere Informationen werden in den Workflow- Details betrachtet.

Um den RUP einzusetzen existieren diverse Richtlinien, Vorlagen und Tool- Mentoren. Die Tool- Mentoren sind vor allem für den Einsatz des RUP in der ebenfalls von Rational entwickelten Rational Suite vorgesehen.

Charakteristik im Detail

Generizität

Der RUP als ein Modell zum Vorgehen bei der Entwicklung von Software kann in entsprechenden Umgebungen ohne weiteren Anpassungsaufwand angewendet werden. Dies trifft vor allem zu wenn die von Rational entwickelten Softwareprodukte zum Einsatz kommen. Jedoch auch ohne diese Tools kann ein Einsatz, entsprechend der Handbücher erfolgen. Die Bedingungen für die Einordnung als Vorgehensmodell sind somit erfüllt.

Eine Verwendung als Referenzmodell oder Framework II ist durch die starke Verknüpfung mit den Rational Software Produkten ebenfalls gegeben.

Unter bestimmten Umständen kann RUP auch als Methodensammlung oder Methodenfamilie mit Repository angesehen werden. Eine vollständige Einordnung in einer dieser Kategorien ist jedoch nicht sinnvoll, da der Umfang des gesamten RUP wesentlich größer ist und eine solche Nutzung nicht dem Anspruch genügen würde.

Phasenabdeckung

Der RUP unterstützt den Softwareentwicklungsprozess, im Rahmen der wesentlichen Prozess-Workflows, von der Modellierung bis zum Einsatz, der so genannten Verteilung. Unter Verteilung wird im RUP die Installation des Endprodukts beim Kunden und ggf. die Durchführung entsprechender Schulungen verstanden. Es sind keinerlei Elemente wie Systemmonitoring oder Betriebsauswertungen vorhanden. Aus diesem Grund kann die Phasenabdeckung für den Betrieb nicht mit 100 Prozent angenommen werden.

Einer der wesentlichen Unterstützungs- Workflows ist das Projektmanagement, welches entsprechende Aufgaben und Aktivitäten für die Planung und Projektierung vorsieht.

Sub-Komponenten

Sub-Komponenten finden ihre Entsprechungen im RUP durch die entsprechenden Workflows. Die Core- Workflows des Engineering stellen dabei das Subsystem Systementwicklung dar. Dabei ist der Test Workflow die entsprechende Komponente für die Qualitätssicherung. Es sind diverse Qualitätsaspekte sowie verschiedene Testarten zu unterschiedlichsten Zeiten und auf den verschiedensten Ebenen vorgesehen.

Weitere Unterstützungs- Workflows decken die Aufgaben des Konfigurations- & Änderungsmanagements und des Projektmanagements ab.

Objekt-Domäne

Der RUP ist ausschließlich für die Entwicklung von Software gestaltet.

Branchenfokus

Im RUP finden sich keine auf bestimmte Branchen zugeschnittenen Methoden. Durch die Anpassung des Prozesses auf das jeweilige Projekt wird ein Modell erreicht, welches in allen Branchen der Softwareentwicklung eingesetzt werden kann.

Formalisierung

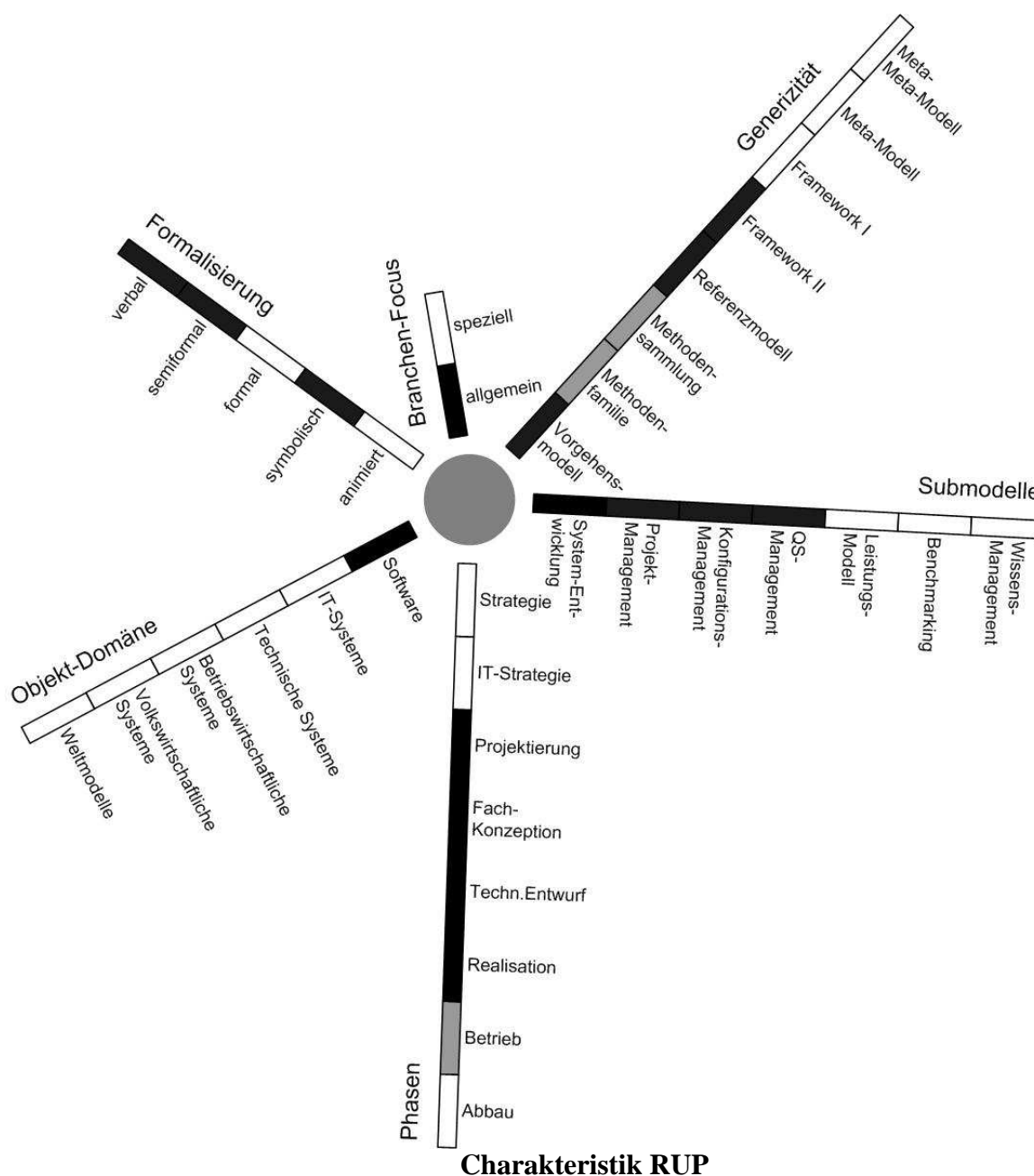
Sowohl in den Handbüchern als auch in der Online Version ist der RUP hauptsächlich in verbaler Form beschrieben. Zur Verdeutlichung gewisser Zusammenhänge und Sachverhalte sind Grafiken (2D) mit eingeflossen.

Der Prozess selber schreibt die Implementierungssprache für das Projekt nicht vor. Jedoch sind die Bücher für C++ und ADA im Prozess enthalten und zeigen somit auch eine Verwendung von Programmiersprachen auf. Diese Darstellungen sind als semiformal anzusehen.

Bewertung

Der Rational Unified Process ist ein in sich geschlossenes mit großer Tool – Unterstützung ausgestattetes Vorgehensmodell. Die kontinuierliche Entwicklung trug dazu bei, dass die Erfahrungen von vielen Projekten im Rahmen der Softwareentwicklung mit eingeflossen sind.

Je nach Einsatzzweck kann das Vorgehen den entsprechenden Gegebenheiten angepasst werden. Hier ist vor allem zu entscheiden in welchem Maß der einzuführende Prozess automatisiert bzw. Tool gestützt werden soll, da dies großen Einfluss auf die Anschaffungskosten hat. Eine Einführung des Prozesses mit allen unterstützenden Tools macht für kleine oder einmalige Projekte sicherlich wenig Sinn. Sind die Projekte größer und wird der RUP für mehrere Projekte verwendet gibt er die Möglichkeit wirtschaftlich und effektiv Software zu entwickeln.



Literatur

- [JaBR1998] Jacobson, Ivar; Booch, Grady; Rumbaugh, James : The Unified Software Development Process. Reading, MA: Addison Wesley Longman, 1998.
- [Kruc1999] Kruchten, Philippe : Der Rational Unified Process - Eine Einführung. Reading, MA: Addison Wesley Longman, 1999.
- [Scot2002] Scott, Kendall : The Unified Process Explained. Pearson Education, 2002.

4.7. St.Gallen Business Engineering Modell und PROMET

Historie und Motivation

Der Ansatz des Business Engineering entstand Anfang der 1990er Jahre am Institut für Wirtschaftsinformatik der Universität St. Gallen (Schweiz) und wird dort seitdem weiterverfolgt. Hintergrund war und ist die Herausforderung der Transformation vom Industrie- zum Informationszeitalter, der sich Wirtschaft und Gesellschaft permanent ausgesetzt sehen. Idee des Business Engineering ist es, diese Transformation in einzelne, besser beherrschbare Schritte aufzuteilen und damit eine systema-

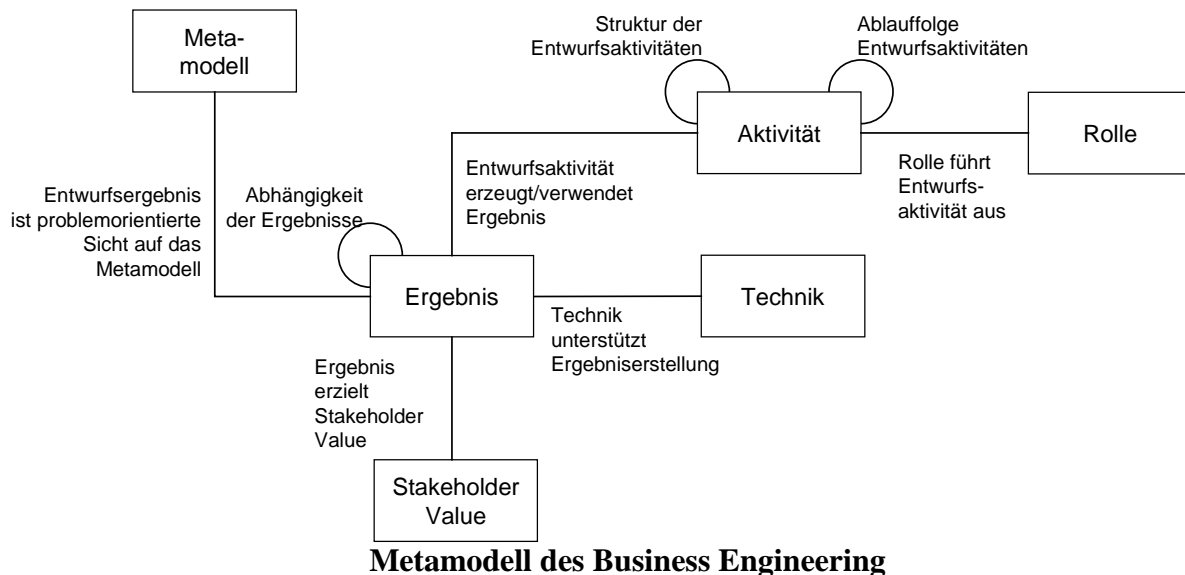
tische Entwicklung von neuen Geschäftslösungen voranzutreiben. Diese Lösungen werden in Form von Geschäftsprozessen definiert und umgesetzt. Die einzelnen Transformationsschritte werden zu Vorgehensmodellen für Projekte zusammengefasst. Es handelt sich also hierbei um einen geschäftsgetriebenen Ansatz [vgl. Blessing/Österle 1999].

Im Laufe der Zeit entstanden diverse Teilkomponenten, welche zum PROMET Methoden-Set zusammengefasst wurden. Dieses Methoden-Set wird heute von einem Spin Off Unternehmen des Institutes betreut, in enger Zusammenarbeit mit dem Institut weiterentwickelt und als lizenzpflichtiges Produkt vertrieben.

Definition bzw. Einordnung

Das Business Engineering zerlegt die Unternehmens-Transformation in Projekte, ein Projekt in den fachlichen Entwurf und in die Führung des Veränderungsprozesses. Den fachlichen Entwurf wiederum in die Ebenen Strategie, Prozess und System. So entstehen überschaubare Projektaktivitäten mit klar definierten Ergebnissen in Form von (Ergebnis-)Dokumenten. Ein Vorgehensmodell schließlich verbindet die Aktivitäten wieder zu einer Reihenfolge im Projektplan.

Um die nötige Stringenz in den Projekten zu erzielen, befolgt das Business Engineering die Grundsätze des Method Engineering [vgl. Gutzwiller 1994, S. 11 ff.].

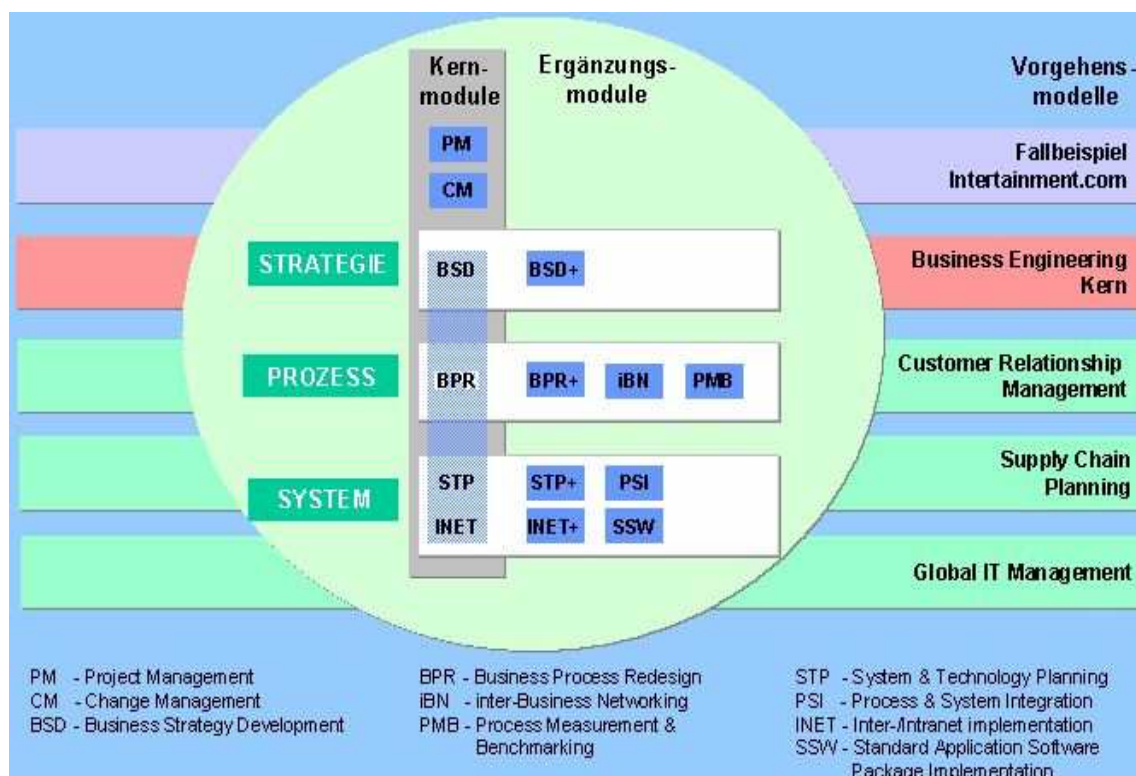


Business Engineering ist **Ergebnis**-orientiert. Jede Projektaktivität führt zu definierten Ergebnissen (meist Dokumenten). Die **Techniken** leiten bei der Erarbeitung der Ergebnisse an. Ein Projekt besteht aus einer Folge von **Aktivitäten** (Vorgehensmodell), die im Projektplan bzw. in Aktivitätsplänen festgelegt werden. Die Techniken helfen bei der Ausführung der Aktivitäten. Das Kriterium für jede Geschäftslösung ist der **Stakeholder Value**. Als Stakeholder werden alle vom Projekt betroffenen Interessengruppen bezeichnet. Die Mitwirkenden in einem Projekt nehmen bestimmte **Rollen** wahr, die mit entsprechenden Aufgaben, Kompetenzen und Verantwortung verbunden sein müssen. Das **Metamodell** ist das Datenmodell des Business Engineering. Es beschreibt die einzelnen Gestaltungsobjekte (z. B. Prozess, Aufgabe, Kunde, Applikation) und die Beziehungen zwischen diesen [vgl. zur Metamodellierung Ferstl/Sinz 1998, S. 117 ff.].

Aufbau

PROMET besteht aus einem Set modular aufgebauter und auf die spezifischen Problem- bzw. Lösungssituationen zugeschnittener Methoden, welche über definierte Schnittstellen miteinander kombiniert werden können. Die **Kernmodule** umfassen die zentralen Techniken für die durchgängige Projektabwicklung von der strategischen Ebene über die Prozessgestaltung bis zur Implementierung einer neuen System-Lösung. **Ergänzungsmodule** liefern für weitere Fragestellungen, z.B. die Steuerung des kontinuierlichen Verbesserungsprozesses nach Projektabschluss (PMB), die Integration neuer Lösungen in die bestehende Infrastruktur (PSI) oder die Standardsoftware-

Einführung (SSW), spezifische Lösungen. **Vorgehensmodelle** kombinieren Elemente der einzelnen Module über die Modulgrenzen hinweg zu einem Leitfaden für definierte Projektsituationen wie z.B. für ein CRM-Einführungsprojekt [vgl. IMG 1997a / IMG 1997b].



PROMET Komponenten

Charakteristik im Detail

Generizität

Business Engineering bzw. PROMET hat ein Metamodell mit vordefinierten Ergebnistypen bzw. Templates, welche an die jeweilige Projektsituation angepasst werden können. Es umfasst Checklisten, welche in strukturierter Form im Intranet bzw. Extranet bereitgestellt sind. Auf die Ergebnisse und Techniken kann entweder direkt oder über modulübergreifende Referenz-Vorgehensmodelle zugegriffen werden, z.B. für ein CRM-Projekt auf Ergebnistypen der Strategie, Prozess- und Systemebene.

Phasen

Schwerpunkt ist – wie der Name schon sagt – die Gestaltung von Geschäftslösungen. Abgedeckt sind die Strategie-Entwicklung (Unternehmensstrategie und IT-Strategie) sowie die Projektierung und Fachkonzeption. Technischer Entwurf und Realisation sind nur teilweise abgedeckt. Hier liegt der Schwerpunkt eindeutig bei der Umsetzung von Standardsoftware-Lösungen. Betrieb und Abbau von Lösungen sind nicht abgedeckt.

Submodelle

Das Leistungsmodell ist der zentrale Bestandteil des Ansatzes. Aus den angestrebten Leistungen der Geschäftsprozesse gegenüber internen oder externen Kunden werden die Anforderungen an die Geschäftsprozesse und Systemlösungen abgeleitet. Benchmarking und Projekt-Management sind ebenfalls abgedeckt. QS-Management nur teilweise.

Objekt-Domäne

Klarer Schwerpunkt ist die Beschreibung der betriebswirtschaftlichen Anforderungen – abgeleitet aus der Unternehmensstrategie - und deren Umsetzung in Form von Geschäftsprozessen. Weiterhin die Spezifikation und Implementierung einer Softwarelösung soweit es sich dabei um eine Stan-

dardsoftware-Lösung handelt. Systementwicklungs-Projekte und technische Systemlösungen sind nicht abgedeckt.

Branchenfokus

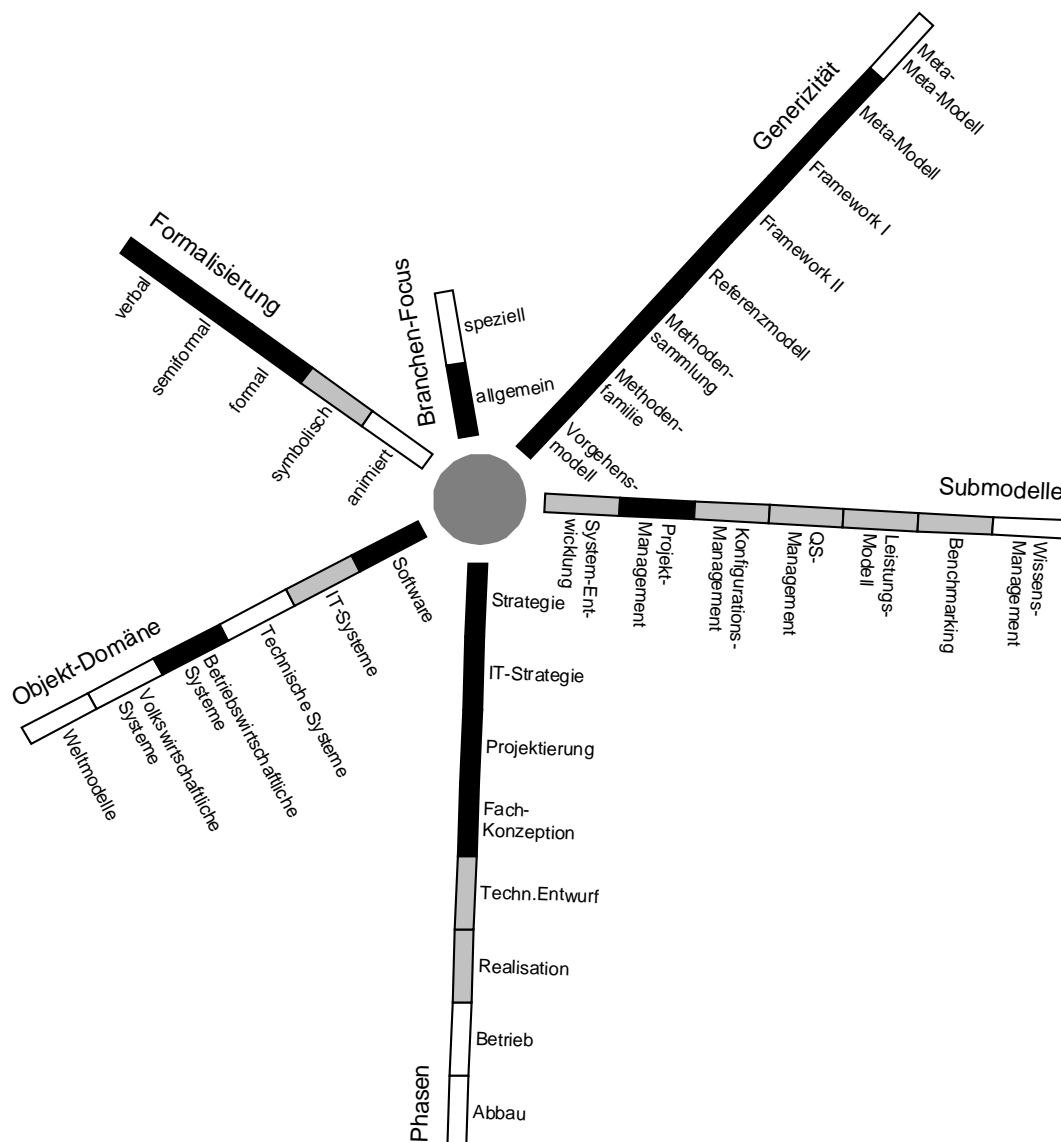
Das Modell hat keine spezifische Industrie-Ausrichtung, d.h. ist branchenneutral

Formalisierung

Die Dokumentation existiert in elektronischer Form als Intranet- / Extranet-Lösung. Auf die Ergebnis-Dokumente kann per Mausklick über ein Vorgehensmodell zugegriffen werden. Alternativ kann ein Ergebnis auch über die Modulsicht direkt ausgewählt werden. Außerdem gibt es schriftliche Dokumentationen in Form von Methodenhandbüchern je Modul.

Bewertung

Der Ansatz hat sich laut Herstellerangabe seit über zehn Jahren bei einer Vielzahl von Projekten bewährt. Der Schwerpunkt liegt eindeutig in der Ableitung der geschäftlichen Anforderungen aus der Unternehmensstrategie und bei der Erstellung der fachlichen Feinspezifikation. Die Umsetzung in eine Systemlösung wird nur dann unterstützt, wenn es sich dabei um eine Standardsoftware-Lösung handelt. Für Systementwicklungsprojekte ist eine Kombination mit anderen Modellen wie V-Modell XT oder RUP denkbar. Der laufende Betrieb von Lösungen wird nicht unterstützt. Das folgende Sterndiagramm zeigt die Ausprägung der einzelnen definitorischen Kriterien.



Charakteristik PROMET

Literatur

[Blessing/Österle 1999] Blessing, D., Österle, H., Business Engineering Model, Arbeitsbericht des Instituts für Wirtschaftsinformatik der Universität St. Gallen, St. Gallen, 1999

[Ferstl/Sinz 1998] Ferstl., O. K., Sinz, E., E., Grundlagen der Wirtschaftsinformatik, Band 1, 3. Aufl., Oldenbourg, München/Wien, 1998

[Gutzwiller 1994] Gutzwiller, T., Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen, Physica, Heidelberg, 1994

[IMG 1997a] Information Management Gesellschaft, PROMET®-BPR – Methodenhandbuch für den Entwurf von Geschäftsprozessen, Version 2.0, St. Gallen et al., 1997

[IMG 1997b] Information Management Gesellschaft, PROMET®-SSW – Methodenhandbuch für die Einführung von Standardanwendungssoftware, Version 3.0e, St. Gallen u. a., 1997

<http://www.promet-web.com>

http://www.promet-web.com/prod/start/default.asp?art=serv_a5d&lan=d&kap=serv

4.8. OTK

Historie und Motivation

Ausgangspunkt in früheren Modellen: OTK ist auf der Basis von CommonKADS, einem Vorgehensmodell für Knowledge Management Applikationen entwickelt worden. Zur Zeit von CommonKADS war die Ontologie-basierte Software-Entwicklung ohne Bedeutung. Mit dem Bestreben heutige Web-Technologien in Richtung „Semantisches Web“ semantisch anzureichern, sind einige Bedeutungsspezifizierende Fortentwicklungen entstanden. Eine zukunftssträchtige Entwicklung diesbezüglich sind Ontologien. Ontologien sind ein Hilfsmittel, eine Methode für den Entwurf von Wissensstrukturen wie auch implementierbare, automatisierbare Konstrukte und haben als solche einen Life Cycle.

Zielsetzung von OTK ist es den gesamten LifeCycle von Ontologiebasierten Applikationen zu unterstützen. OTK ist im Rahmen von IST-Programm der EU entstanden. Bzgl. der Standardisierungsbestrebungen ist folgendes anzumerken: Wie alle Web-Technologien letztendlich von W3C auf Standardisierungsempfehlungen überprüft werden, befindet sich auch die Ontologie-Sprache OWL auf dem Weg in den Status „Empfehlung“. OTK wird nicht zu einem Standard erhoben werden.

Definition bzw. Einordnung

Bezeichnung: On-to-knowledge Methodology, kurz OTK

Abgrenzung: OTK kann als Fortsetzung von einigen CommonKADS-Komponenten zur Ontologie-Entwicklung angesehen werden.

Varianten: keine bekannt

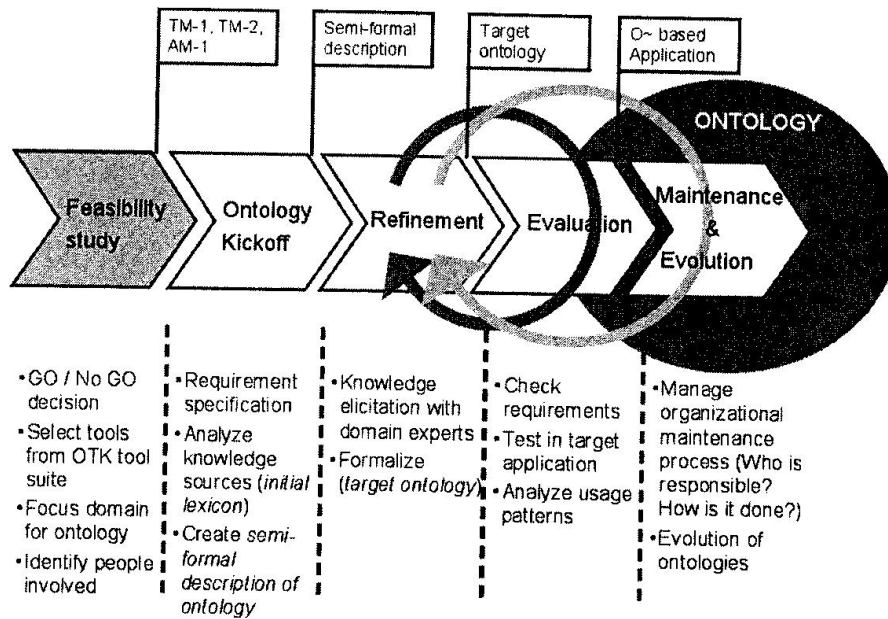
Klassifikation: Vorgehensmodell zur Entwicklung von Ontologie-basierten Software-Lösungen

Aufbau, Funktionen, Informationen, Prozesse

OTK ist ein Prozessmodell das in einer bestimmten Schrittefolge verschiedenen Produktionsphasen durchläuft. Für einige Phasenprodukte stehen Darstellungsmittel und Symbolik zur Verfügung.

OTK ist unabhängig und bezieht nicht von anderen VM einen Input, liefert keinen Output, an andere VM sondern führt unmittelbar zu einer Softwarelösung.

Die folgende Abbildung stellt den Entwicklungsprozess über alle Phasen vor.



Entwicklungsprozess nach OTK

Charakteristik im Detail

Generizität

OTK ist ein Referenzmodell das sich um eine Entwurfsmethode im engeren Sinne, den Ontologie-Entwurf, rankt. Zu den einzelnen Phasen sind Empfehlungen für Methoden wie z.B.

- Feasibility Study (Phase und Ergebnis):
 - Use Cases nach UML
 - Organisation nach CommonKADS, bestehende aus Problembeschreibung, Organisations-Focus, Prozesszerlegungsstruktur, Wissensnutzung, Machbarkeitsbeurteilung,
 - Task Modell, bestehend aus Task Analysis nach CommonKADS und Knowledge Item Analysis nach CommonKADS
 - Agent Model nach CommonKADS
- Kick Off Phase:
 - Ontology-Zielsetzung
 - Design Guidelines
 - Wissensquellen-Katalog
 - Nutzungsszenarios
 - Erhebung der Kompetenzfragen
 - Definition Ontology-Interface, Aufstellung des semi-formal beschriebenen Ontologiemodells (initial draft), Anschluß an Wissensmanagement-Applikationen
- Refinement Phase
 - Wissenserhebung, Abstimmung mit den Domain-Experten
 - Modifizierung der semi-formal beschriebenen Ontologien
 - Formalisierung der semi-formalen Beschreibung mittels Ontologie-Entwurf (z.B. erweiterte Klassendiagramme aus UML, OIL, DAML OIL, OWL)
- Evaluationsphase
 - Verbesserung der Ontologiebestände, Schließung von Lücken, Korrektur von Fehlern, Aufdeckung und Reparatur semantischer Widersprüche
- Maintenance und Evolutionsphase
 - Kontinuierliche Verbesserung der formalen Ontologien, oder auch aufnehmen neuer Domänen und Ergänzung der semi-formalen, Beschreibungen

Die Methoden stehen in einem Lieferverhältnis: Folgemethoden sind auf die Ergebnisse der Vorgängermethoden angewiesen. In diesem Sinne liegt ein Komplettes Vorgehensmodell mit einer Methodenfamilie vor.

Phasenabdeckung

OTK startet bei der Projektstrukturierung und endet mit der Wartung der Informationsbestände. Der Strategiebezug ist implizit in der Feasibility-Study als Anforderung eingeflossen. Eine Unternehmensstrategie wird nicht explizit formuliert. Eine Phase zum Abbau ist nicht definiert.

Sub-Komponenten

Schwerpunkt der Phasen ist die Systementwicklung eingeschränkt auf die Ontologie-Applikationsentwicklung.

Das Vorgehensmodell selbst ist als Anleitung zur Generierung, Verwaltung, Auswertung, Weiterentwicklung von Wissen ein Baustein des Wissensmanagementprozesses und enthält damit implizit ein Wissensmanagementzyklus, eingeschränkt auf Ontologien als Wissensträger.

Außer der Nutzungsanalyse von Ontologien sind keine QS-Modelle und keine Prozessbenchmarkingmodelle definiert.

Projektmanagementmodell und Leistungsmodell (Angebote, Ausschreibungen) sind nicht Gegenstand.

Objekt-Domäne

Die Methoden sind auf die Entwicklung von Ontologien und der Applikationen zur Verwaltung von Ontologien über deren gesamten Lebenszyklus ausgerichtet. Damit ist die Objektdomäne Software, aber nicht die gesamte Palette der möglichen Softwaretypen, sondern Wissensmanagement-Applikationen und dort speziell ontologiebasierte WM-Softwarelösungen.

Branchenfokus

OTK ist nicht auf spezielle Branchen ausgerichtet, und auch nicht einschränkend auf Branchen anzuwenden. Ontologien helfen Informationen bzw. Wissen zu verwalten um eine effizientere und semantisch aussagekräftigere Suche zu gestatten. Ontologien werden daher eher in Web-Services, und begleitende Dienstleistung (z.B. Handel, Bibliotheken, Suchdienste, Unternehmenslexikon, ...) Anwendung finden. Spezielle Ontologien werden auch als Referenzmodelle gehandelt, z.B. Stücklistenzerlegung von Kraftfahrzeugen, Medikamententypisierung, Krankheiten charakteristik.

Formalisierung

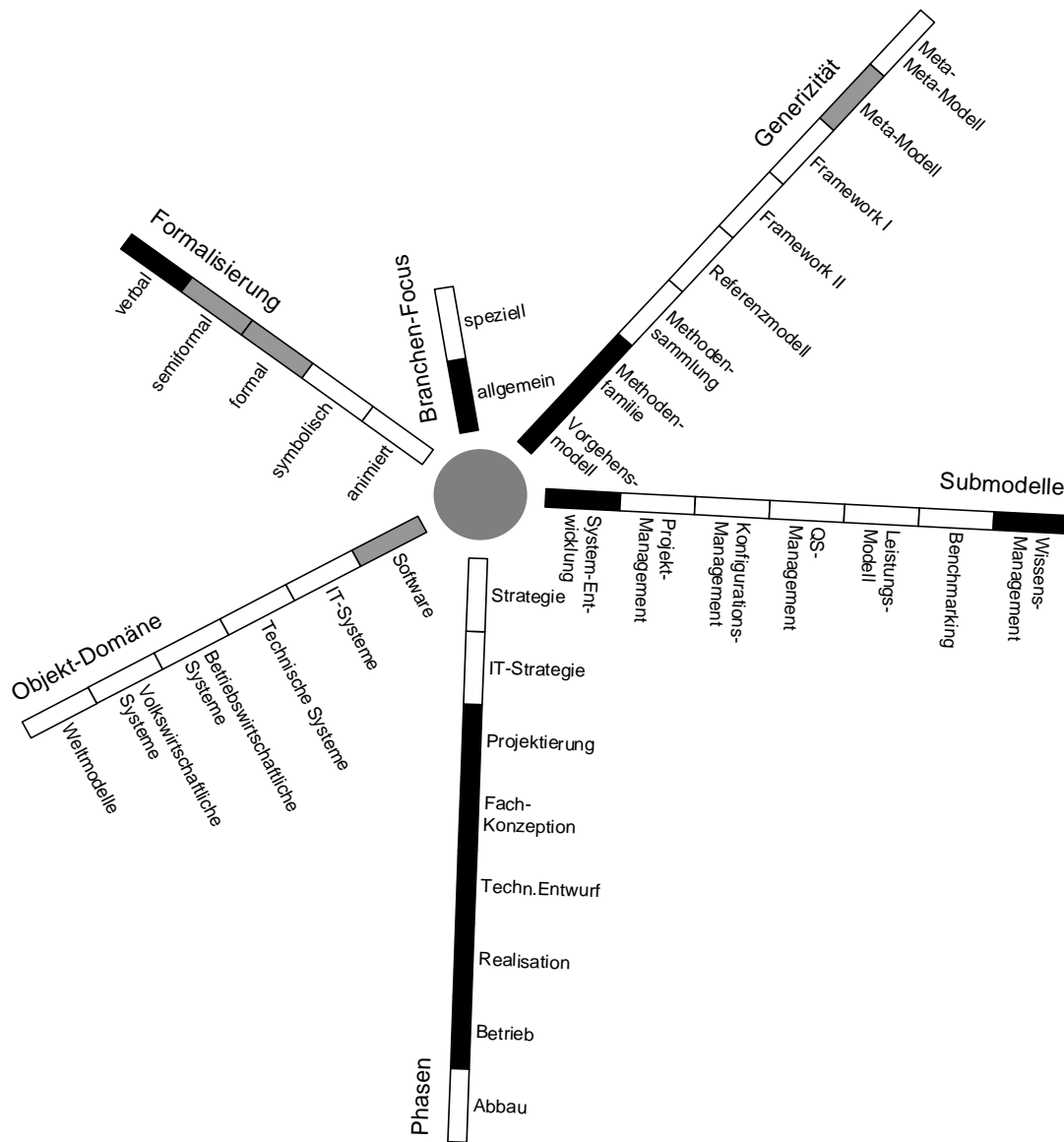
Alle Ergebnisse der OTK-Phasen sind verbal beschreibbar.

Die Wissensbestände zu den Domänen, etc. werden in der Reihenfolge ihrer Konkretisierung zunächst semiformal und dann in Form von Ontologien formal beschrieben, mittels spezieller Programmiersprachen (DAML+OIL, OWL, u.a.). Für Ontologien existiert eine auf Programmiersprachenebene aufgesetzte Symbolik. Für Anwendungsfälle wird die UML Symbolik von Use-Cases genutzt. Der Formalisierungsgrad ist damit teilweise symbolisch.

Bewertung

Erfahrungen, Kosten, Nutzen, Maßstäbe, Skalen, Akzeptanzstudien, Toolunterstützung, Implementierungsstatistiken

Das folgende Sterndiagramm zeigt die Ausprägung der einzelnen definitorischen Kriterien.



Charakteristik OTK

Literatur

Davies J., Fensel D., Harmelen, van F., Towards the Semantic Web, Wiley, 2003

5. Ergebnisse

Der Begriff des Vorgehensmodells ist nicht eindeutig. Die Spanne der Interpretation reicht von einem Metamodell das einem Anwender hilft Daten strukturiert und mit referentieller Integrität in einer Datenbank abzulegen, über Methodensammlungen mit grafischer Unterstützung, über auf einem Data Dictionary (Meta Modell) operierenden frei auswählbaren Methodenset, über in Phasen und Lifecycle-orientierten Gruppierungen von Erzeugnisgruppen (Artefakte aus Methodenanwendungen) bis zur streng am Entwicklungsprozess aufgereihten Methodenkette. Für die Entscheidungsfindung des Anwenders die für das Projekt passende Unterstützung zu bekommen sind alle Ausprägungen interessant und unter der Frage „passendes Vorgehensmodell für eine Entwicklungsaufgabe auswählen und konfigurieren“. Wo für den einen Anwender die Freiheit der Methodenwahl und die Konfigurierbarkeit der Phasen erst die Projektgestaltung ermöglicht ist der andere Anwender mit den Freiheitsgraden überfordert und das Projekt wird mangels breiter Akzeptanz scheitern.

Das Entscheidungsbild kann daher nicht auf eine strenge Definition des Begriffs „Vorgehensmodell“ konzentriert werden, z.B. mit der Erfüllung aller Konzepte wie

- Vollständige Projektkomponenten (QS, KM, PM,..)
- wohldefinierte Phasen (Requirements, Entwurf, Implementierung, ...)
- konsistente Methoden (ERM, DFD, PFD, Zustandsmodellierung, Dialogmodellierung, Erstellung Klassendiagramm, ...)
- Spezifizierte Ergebnistypen, Produkten, Artefakten (Datenmodell, Prozessmodell, Evaluationsstudie,...)

Die Entscheidungssituation erfordert eine breite Auslegung des Begriffs „Vorgensmodell“ umfassend alle Ausprägungen wie:

- Metamodell
- Framework
- Methodenkette
- Methoden-Pool

Die Ausstattung mit Konzepten ist ein Gütekriterium eines im umfassenden Sinne verstandenen Vorgehensmodells. In diesem Sinn ist auch die zu Beginn des Arbeitskreises gestellte Frage ob denn XP, alle agilen Methoden überhaupt zum Kreis der Vorgehensmodelle zu rechnen sind, mit einem eindeutigen „Ja“ zu beantworten. Allerdings stehen agile Methoden und XP am untersten Ende der Skala der Güte der VM-Ausstattung. Was einen geringen Lernaufwand begründet und was eventuell deren Beliebtheit erklärt.

6. Ausblick

Es soll nicht verschwiegen werden, dass um einen effizienten Satz von Beschreibungsdimensionen zu bekommen auch noch einige Strukturprobleme zu klären sind:

- Problem 1: Dimensionenminimalität
Vermeidung des gegenseitigen Enthaltenseins von Kandidaten, Minimierung der Anzahl der Dimensionen, Aufstellen eines Systems unabhängiger Dimensionen
- Problem 2: Echte Dimensionen
Verhindern das eine Ausprägung oder Eigenschaft einer Dimension für eine Dimension gewählt wird, ist eventuell nicht auf Anhieb sichtbar,
- Problem 3: singuläre Dimensionen
Vermeidung das der Dimensionskandidat zu grob gefasst ist und mehrere Dimensionen umfasst
- Problem 4: kompatible Begrifflichkeit
 - Gleiche Begriffe weichen häufig in der Bedeutung leicht voneinander ab,
 - für die gleiche Bedeutung sind oft verschiedene Begriffe in Verwendung.
- Problem 5: Abgegrenztheit
Modelltypen sind in der Regel gegeneinander nicht abgrenzt, überschneiden sich gegenseitig und enthalten sich zum Teil.
- Problem 6: Definition
Minimalauffassung des Begriff Vorgehensmodell umfasst die Prozesssicht, d.h. das Durchlaufen von Phasen, nicht, dann sind auch „Kataloge wie CMM Vorgehensmodelle.
- Problem 7:Phasen-/Objekt-Mehrdeutigkeit
Die Gestaltung von Betriebswirtschaft, Umwelt, Technischen Systemen kann sowohl als Gestaltungsobjekt als auch als Vorphasen, vor dem eigentlichen IT-Projekt aufgefasst werden. Also z.B. die Gestaltung der Strategie als Vorphase zur Gestaltung der IT-Strategie.

Die Klärung dieser Fragen, wie auch die Auswertung der Betrachtung auf weitere interessante Vorgehensmodelle im weitesten Sinne (ARIS, NIAM, DWTK, CMMI, QFD, 6-Sigma, PMBOK, WebML, SPEM, PRINCE...) soll bis zum kommenden Workshop 2006 erarbeitet werden. Der Arbeitskreis ist offen für jeden, der sich für das Thema interessiert.