

Institut für Informatik der
Technischen Universität München

Vom Vorgehensmodell zum Projektplan

Michael Andreas Josef Gnatz

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Peter Hubwieser

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Dr. h.c. Manfred Broy
2. Jun.-Prof. Dr. Andreas Rausch,
Universität Kaiserslautern

Die Dissertation wurde am 29.06.2005 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 28.10.2005 angenommen.

Vom Vorgehensmodell zum Projektplan

Michael A. J. Gnatz

Kurzfassung

Studien zufolge müssen viele Softwareentwicklungs-Projekte heute in Bezug auf Zeit, Kosten und Qualität als nur eingeschränkt erfolgreich eingestuft werden. Der Einsatz geeigneter Vorgehensmodelle kann wesentlich zum Erfolg eines Projektes beitragen. Vorgehensmodelle können als Checklisten für die Durchführung von Projekten dienen, Dokumentvorlagen vorgeben oder auch als Eingabe für Workflow-Maschinen mit Aufgaben-Listen für die einzelnen Projektbeteiligten dienen. Die vorliegende Arbeit hat die Ableitung von Projektplänen als eine andere Art der Anwendung von Vorgehensmodellen im Fokus.

Kernthema der Arbeit ist die Konzeption eines Vorgehensmodells als Abstraktion bewährter Projektpläne, mit dem Ziel, aus einem entsprechenden Vorgehensmodell auf einfache Weise einen Projektplan ableiten zu können. Im Zentrum steht die präzise Definition von Mechanismen und damit die Spezifikation einer Werkzeugunterstützung für die projektspezifische Anpassung eines Vorgehensmodells sowie die Ableitung eines Projektplans aus einem Vorgehensmodell. Vorgehensmodelle werden also eigens für den Anwendungsfall Projektplanung ausgelegt.

Wir definieren die Semantik eines Vorgehensmodells im Kontext der Projektplanung präzise mit Hilfe eines Meta-Modells für Vorgehensmodelle. Dieses legt die Konzepte für die Beschreibung eines Vorgehensmodells, die strukturellen Zusammenhänge dieser Beschreibungskonzepte sowie Konsistenzbedingungen bezüglich der Anwendung der Beschreibungskonzepte fest. Damit dient das Meta-Modell als Modellierungssprache für Vorgehensmodelle. Wir bilden das Meta-Modell für Vorgehensmodelle zudem auf ein Modell für Projektpläne ab. Damit dient das Meta-Modell zugleich als Spezifikation einer Werkzeugunterstützung für die Ableitung von Projektplänen aus einem entsprechend dem Meta-Modell beschriebenen Vorgehensmodell.

Das Meta-Modell soll die Beschreibung aller Aspekte eines Vorgehensmodells erlauben, die sich aus der Abstraktion „guter“ Projektpläne ergeben. Die Ausdrucksmächtigkeit priorisieren wir dabei höher als die Einfachheit der Beschreibungsmittel. Wir nehmen eine Analyse der Schwachstellen des V-Modells XT als Ausgangspunkt für die Konzeption eines Meta-Modells für Vorgehensmodelle, die sich auf die Projektplanung als Anwendungsfall ausrichten.

Den methodischen Einsatz der Beschreibungsmittel, die Ausdrucksmächtigkeit der Beschreibungsmittel und die Stimmigkeit der erarbeiteten Konzepte zeigen wir anhand zahlreicher Beispiele auf Ebene des Vorgehensmodells sowie auf Projektebene. Zudem geben wir eine Methodik für die Durchführung von Projekten zur Erstellung von Vorgehensmodellen an. Neben der Erstellung unterstützen die konzipierten Beschreibungsmittel zudem die Änder- und Erweiterbarkeit von Vorgehensmodellen. Vorgehensmodelle sind oft „langlebige Systeme“. Zu diesem Zweck beinhaltet das Meta-Modell ein Modulkonzept.

Ein Spezialisierungskonzept ermöglicht die Beschreibung von Vorgehensmodellen auf unterschiedlichen Spezialisierungsebenen. Damit unterstützt das Meta-Modell die Beschreibung abstrakter Modelle sowie organisations- und domänenspezifischer Ausprägungen dieser abstrakten Modelle.

Danksagung

Die Arbeit motiviert sich aus vielfältigen Erfahrungen, bereichernden Diskussionen und Ergebnissen des Projektes WEIT zur Erstellung des V-Modells XT. Dieses Projekt ermöglichte das praxisnahe Studium vielfältiger Aspekte der Vorgehensmodellierung und Projektplanung in doppelter Hinsicht – zum einen anhand der ingenieurmäßigen Vorgehensweise im Projekt selbst, zum anderen anhand des V-Modells als Projektgegenstand.

Zuerst möchte ich mich bei Prof. Dr. Dr. h. c. Manfred Broy für das angenehme und befruchtende Arbeitsumfeld, die Freiheit zum wissenschaftlichen Arbeiten und die Betreuung des Erstellungsprozesses der vorliegenden Arbeit bedanken. Mein Dank gilt nicht weniger J. Prof. Dr. Andreas Rausch für zahlreiche hilfreiche Kommentare und Anregungen, die zum Gelingen der Arbeit beigetragen haben, und schließlich für die Übernahme des Zweitgutachtens.

Für die Durchsicht und zahlreiche hilfreiche Kommentare zu früheren Versionen des Manuskriptes gebührt mein Dank Jan Friedrich, Rupert Gnatz, Ulrike Hammerschall, Inga Küffer, Marco Kuhrmann und Tilman Seifert. Für die redaktionelle Durchsicht von Teilen des Manuskripts bedanke ich mich bei Helene Gnatz.

Nicht zuletzt bedanke ich mich bei meinen Eltern, die mich motiviert haben, meinen Weg einzuschlagen und bei Antonina für die Geduld, die sie während der Erstellung dieser Arbeit mit mir hatte.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Thematische Einordnung und Zielsetzung	4
1.2	Beitrag der Arbeit.....	8
1.3	Aufbau der Arbeit.....	10
1.4	Vergleich verwandter Arbeiten	13
2	State-Of-The-Art: Vorgehensmodelle und Projektplanung	17
2.1	Vorgehensmodelle: Überblick und Einordnung.....	17
2.1.1	Merkmale zur Klassifikation von Vorgehensmodellen.....	18
2.1.2	Projektdurchführungsstrategie.....	19
2.1.3	Anpassbarkeit.....	24
2.1.4	Inhaltliches Spektrum	30
2.1.5	Abstraktionsgrad.....	33
2.1.6	Anwendung und Werkzeugunterstützung	36
2.2	Definition Vorgehensmodell	39
2.3	Projektmanagement und Projektplanung	42
2.3.1	Projektmanagement und Projektplanung im V-Modell XT	42
2.3.2	Projektplanung: Vorgehen und Methodik	46
2.3.3	Praktische Probleme der Projektplanung	54
2.4	Definition Projektplan	56
2.5	Zusammenfassung.....	58
3	Schwachstellenanalyse des V-Modells XT	59
3.1	Vorgehensbausteine	59
3.2	Abläufe	68
3.3	Produkte	74
3.4	Abstraktion im V-Modell	80
4	Modellierungstechnik für Vorgehensmodelle	87
4.1	Grundlegendes zur Definition von Modellierungssprachen	87
4.2	Definition der verwendeten Modellierungstechnik.....	90
4.2.1	Klassen und Objekte	90
4.2.2	Generalisierung und Spezialisierung.....	97
4.3	Modellierungsansatz für Vorgehensmodelle und verwandte Ansätze	104
5	Modellierung von Vorgehensmodellen	107
5.1	Die Modelle im Überblick.....	107
5.2	Strukturmodell	110
5.2.1	Planungsmodell – Planungselemente	110
5.2.2	Produktmodell – Produkte	112
5.2.3	Planungsmodell – Planung der Produkterstellung	116
5.2.4	Planungsmodell – Strukturierung des Projektablaufs	122
5.3	Dynamisches Modell	127

5.3.1	Planungsmodell – Ablaufbeziehungen und Iterationen.....	127
5.3.2	Produktmodell – Abhängigkeitsbeziehungen	132
5.4	Spezialisierungskonzept.....	142
5.4.1	Spezialisierung auf Ebene des Vorgehensmodells.....	142
5.4.2	Studie: Spezifische Ausprägungen des V-Modells XT.....	147
5.4.3	Beziehungen zwischen Spezialisierungs-Ebenen: Generische Vorgehensweisen	148
5.5	Zusammenfassung.....	154
6	Erstellung und Anwendung von Vorgehensmodellen.....	157
6.1	Modell zur inhaltlichen Strukturierung	157
6.2	Tailoring-Modell.....	169
6.3	Studie: Tailoring anhand von Beispielen des V-Modells XT	174
6.4	Anwendung von Vorgehensmodellen: Tailoring und Ableitung eines Projektplans	177
6.5	Methodisches Vorgehen der Erstellung von Vorgehensmodellen	184
6.6	Modellierung von Texten.....	191
6.7	Werkzeugkonzept.....	194
7	Zusammenfassung und Ergebnisse	197
	Abbildungsverzeichnis	i
	Quellenverzeichnis	v
	Anhang.....	xi
A.	Vorgehens-Meta-Modell im Überblick.....	xi

1 Einleitung

„The trouble comes from the fact that there is so much tinkering with software. It is not made in a clean fabrication process, which it should be. ...

What we need is software engineering.“ [Bauer93]

Die Software-Industrie steht heute vor nie da gewesenen Herausforderungen. Der Anteil von Software im Alltag und damit ihre Bedeutung steigen seit vielen Jahren kontinuierlich an. Zudem lässt sich ein stetiger Zuwachs an Umfang und Komplexität moderner Softwaresysteme beobachten. Kunden fordern heute Software, die eine Vielzahl von Funktionen aufweist, einfach zu benutzen ist und hohen Qualitätsansprüchen gerecht wird. Diese Herausforderungen müssen in einem dynamischen Projektumfeld bewältigt werden, in dem sich Technologien häufig ändern, Mitarbeiter oft die Firma wechseln, Teams räumlich verteilt arbeiten und globaler Wettbewerb die Kosten drückt und immer kürzere Entwicklungszeiten verlangt.

Software ist immateriell und wird praktisch nicht durch physikalische Gesetze begrenzt. Im Gegensatz zu materiellen, technischen Produkten scheint hier das „Denkmögliche“ machbar zu sein. Mit dem Zuwachs an Komplexität einher geht allerdings das Problem der Beherrschbarkeit der Software-Entwicklung beziehungsweise Software-Pflege und -Wartung.

Der Begriff „Software-Krise“ wurde Mitte der 60er Jahre geprägt (vgl. [NR69]). Software wurde oft nicht termingerecht und zu höheren Kosten als ursprünglich geschätzt fertig gestellt sowie entsprach vielfach nicht den Anforderungen ihrer Nutzer. Die Software-Krise war der Auslöser, Software-Entwicklung analog zu den „klassischen“ Ingenieurdisziplinen zu systematisieren und zu organisieren. Der Begriff „Software Engineering“ spiegelt die Auffassung wider, dass die Erstellung und Wartung von Software kein künstlerischer oder „handwerklicher“ Prozess sein sollte, sondern ein ingenieurmäßig ablaufender Prozess, an dem oft viele Spezialisten beteiligt sind. Unter einem ingenieurmäßig ablaufenden Prozess verstehen wir im Gegensatz zum reinen „Handwerk“ einen Prozess, bei dem die Beteiligten über wissenschaftlich fundierte Informationen über ihr Tun verfügen.

Ein **ingenieurmäßiges Vorgehen** ist aufgrund der inhärenten Komplexität von Software insbesondere auch für die Softwareentwicklung anzustreben. Vergleicht man ein Softwaresystem mit einem anderen von Menschen geschaffenen materiellen System und beide Systeme weisen die gleiche Anzahl von „Elementen“ auf, so ist das Softwaresystem meist als komplexer einzustufen. Sind zwei Teile eines Softwaresystems identisch oder ähnlich, so werden sie nicht wie im Falle von materiellen Dingen dupliziert, sondern in einem Unterprogramm gekapselt. Die Skalierung von Software bedeutet also nie eine reine Wiederholung von Elementen, sondern immer einen Zuwachs an neuen Elementen beziehungsweise neuer Funktionalität und damit an Komplexität.

Im Gegensatz zu Ingenieur-Disziplinen wie dem Bauwesen verfügt die vergleichsweise junge Softwarebranche auch heute noch nicht über verbindliche Strukturen wie etwa eine Bauaufsichtsbehörde. So gibt es weder verbindliche Normen, die die Anwendung von Verfahren und Methoden vorschreiben, noch eine einer Bauaufsichtsbehörde vergleichbare Institution. Der Weg der Softwareentwicklung von einem kunstvollen Handwerk, das noch in den Kinderschuhen steckt, hin zu einer ausgereiften Ingenieur-Disziplin ist noch lange nicht abgeschlossen. Dies ist nicht weiter verwunderlich, da dieser Prozess auch im Falle des Bauwesens durchaus einen längeren Zeitraum beansprucht hat.

Die ingenieurmäßige Entwicklung von Software umfasst die Anwendung von Architekturmodellen, Beschreibungstechniken, Werkzeugen, Methoden und Vorgehensmodellen. Fortschritte auf diesem Gebiet werden heute durch neue technische Möglichkeiten und immer komplexere Anwendungen relativiert. So ist es nicht verwunderlich, dass einige Softwarepro-

jekte auch heute nicht von Erfolg gekrönt ist. Laut Untersuchungen der Standish Group werden in den USA etwa 28% aller Projekte vor ihrer Fertigstellung abgebrochen, 46% verzeichnen signifikante Termin- oder Kostenüberschreitungen oder aber verfügen bei Fertigstellung nicht über die versprochene Funktionalität. Lediglich 26% aller Projekte werden uneingeschränkt erfolgreich abgeschlossen, das heißt im Termin- und Kostenrahmen mit der versprochenen Funktionalität und in angemessener Qualität (vgl. [SGI01]).

Die in den meisten Softwareentwicklungsprojekten zu bewältigenden Aufgaben sind vielfältig und komplex miteinander verflochten und werden daher von unerfahrenen Projektleitern oft unterschätzt. Abhilfe versprechen Vorgehensmodelle, da sie den Erfahrungsschatz früherer Projekte sammeln und dadurch helfen, die gegenläufigen Kräfte Kosten, Zeit und Qualität geeignet auszubalancieren.

Vorgehensmodelle werden dem Bedarf nach verbesserter Kommunikation und Koordination gerecht, der beispielsweise aus der externen Vergabe der Entwicklung von Software oder der Entwicklung in räumlich und zeitlich verteilten Teams resultiert. Sind Vorgehensmodelle alleine auch nicht der „heilige Gral“ erfolgreicher IT-Projekte, so kann ihr Einsatz doch wesentlich zum Erfolg beitragen. Vorgehensmodelle sind eine wesentliche Grundlage für einen arbeitsteiligen, einheitlichen und wiederholbaren Prozess.

In letzter Zeit sind viele neue Vorgehensmodelle entstanden, angefangen von den so genannten agilen Methoden wie Extreme Programming, über Open-Source-Softwareentwicklung bis hin zu Vorgehensmodellen zur Entwicklung von Produktlinien. Prominentes Beispiel für ein junges Vorgehensmodell ist das V-Modell XT, an dessen Erstellung im Rahmen des Projektes WEIT der Autor beteiligt war. Aus den in diesem Umfeld gewonnenen Erfahrungen motiviert sich die vorliegende Arbeit. Neben zahlreichen Abstimmungsprozessen auf inhaltlicher Ebene waren insbesondere die Beschreibungsmittel des V-Modells XT, das heißt die „Modellierung“ des V-Modells XT zentrale Diskussionsthemen. Die Erarbeitung eines klaren gemeinsamen Verständnisses der verwendeten Beschreibungsmittel in einem großen Team hat sich für die Erstellung des V-Modells XT als essentiell erwiesen.

Unter einem **Vorgehensmodell** verstehen wir einen standardisierten organisatorischen Rahmen für den idealen Ablauf eines Entwicklungsprojektes. Vorgehensmodelle werden in Form zu erstellender Produkte, durchzuführender Aktivitäten und zu besetzender Rollen beschrieben. Ein Vorgehensmodell ist damit ein Modell beziehungsweise eine Abstraktion einer möglichen Vorgehensweise im konkreten Projekt. Vorgehensmodelle abstrahieren von projektspezifischen Vorgehensweisen mit dem Ziel einer breiteren Einsetzbarkeit der Modelle für unterschiedliche Arten von Projekten. Nichtsdestotrotz sollen Vorgehensmodelle einfach an den Kontext eines konkreten Projektes anpassbar und damit einfach anwendbar sein. Insbesondere das „Tailoring“ von Vorgehensmodellen, das heißt die Anpassung eines Vorgehensmodells durch Zuschneiden des Modells auf die im Kontext eines konkreten Projektes relevanten Teile, findet heute in der Praxis großes Interesse (vgl. [DGR03]).

Der Nutzen der Anwendung von Vorgehensmodellen kann allgemein in Effizienzsteigerungen gesehen werden. Ein Vorgehensmodell kann als Lektüre verstanden werden, die zum Nachdenken über das eigene Vorgehen anregt. Die bloße Beschäftigung der Projektmitarbeiter mit der Materie kann bereits in einer verbesserten Effizienz resultieren. Viele der heute verbreiteten Vorgehensmodelle scheinen allein den Anwendungsfall einer Lektüre zu erfüllen. Von einem qualitativ hochwertigen Vorgehensmodell sollte allerdings mehr zu erwarten sein. Wird ein ingenieurmäßiges Vorgehen angestrebt, muss ein Vorgehensmodell eher Regelungscharakter als anregenden Charakter haben. Zu diesem Zweck muss die Anwendung eines Vorgehensmodells im Projekt klar vorgegeben sein. Der Nutzen heutiger Vorgehensmodelle ist unter dem Aspekt einer klaren Richtlinie für die Projektdurchführung betrachtet oft fraglich. Vorgehensmodelle weisen heute oft einen hohen Interpretationsspielraum auf. Ein hohes Maß an erforderlicher Interpretation bei der Anwendung reduziert aber das Maß an Hilfestellung, das ein Vorgehensmodell einem Projekt bietet. Eine Voraussetzung für die Erhöhung der Erfolgswahrscheinlichkeit eines Projektes durch ein Vorgehens-

modell ist die präzise vorgegebene Anwendung und einfache Anwendbarkeit des Vorgehensmodells im Projekt.

Vorgehensmodelle wie das V-Modell zielen auf ein ingenieurmäßiges Vorgehen für die Entwicklung von Software ab. Das kürzlich veröffentlichte V-Modell XT resultiert aus der Weiterentwicklung des V-Modells 97. Allerdings ist das V-Modell XT sicherlich nicht als das „Maß aller Dinge“ für Entwicklungsprojekte zu sehen. Weiterentwicklungen auf inhaltlicher Ebene wie auf Ebene der Grundkonzepte sind wünschenswert. Wir beleuchten in der vorliegenden Arbeit Schwachstellen des V-Modells XT hinsichtlich der Ableitbarkeit einer Projektplanung. Die Erstellung eines detaillierten Projektplans auf Basis des V-Modells ist weder präzise definiert noch auf einfache Weise möglich. Dies begründet sich zum Teil aus der Anforderung nach Anwendbarkeit des V-Modells XT in allen Arten von Entwicklungsprojekten im IT-Umfeld. Die Inhalte des V-Modells XT weisen somit einen sehr hohen Abstraktionsgrad auf, der die Ableitung eines Projektplans nicht präzise vorgibt.

In jedem Fall sind konkretere Vorgehensmodelle als das V-Modell XT wünschenswert. Dabei kann es sich um organisationsspezifische Modelle, die sich auf das Geschäftsfeld einer Firma ausrichten, oder um domänenspezifische Modelle handeln, beispielsweise für die objektorientierte Entwicklung betrieblicher Informationssysteme. Diese Einschränkung des Spektrums der Anwendbarkeit ermöglicht Vorgehensmodelle, die sich spezifischer auf bestimmte Klassen von Projekten ausrichten. Im Rahmen der vorliegenden Arbeit fokussieren wir Vorgehensmodelle, die konkrete Vorgaben in Bezug auf die Projektplanung machen.

In Kapitel 1.1 stellen wir die thematische Einordnung und Zielsetzung der Arbeit dar. In Kapitel 1.2 beschreiben wir den Beitrag der Arbeit. In Kapitel 1.3 beschreiben wir den Aufbau der Arbeit. Kapitel 1.4 widmen wir schließlich dem Vergleich der vorliegenden Arbeit mit verwandten Arbeiten.

Inhalt des Kapitels		Seite
1.1	Thematische Einordnung und Zielsetzung	4
1.2	Beitrag der Arbeit	8
1.3	Aufbau der Arbeit	10
1.4	Vergleich verwandter Arbeiten	13

1.1 Thematische Einordnung und Zielsetzung

Kernthema der Arbeit ist die Konzeption eines Vorgehensmodells als Abstraktion bewährter Projektpläne, mit dem Ziel, aus einem entsprechenden Vorgehensmodell auf einfache Weise einen Projektplan ableiten zu können. Im Zentrum steht die präzise Definition von Mechanismen und damit die Spezifikation einer Werkzeugunterstützung für die projektspezifische Anpassung eines Vorgehensmodells sowie die Ableitung eines Projektplans aus einem Vorgehensmodell. Vorgehensmodelle werden also eigens für den Anwendungsfall Projektplanung ausgelegt.

Wir definieren die Semantik eines Vorgehensmodells im Kontext der Projektplanung präzise mit Hilfe eines Meta-Modells für Vorgehensmodelle. Dieses legt die Konzepte für die Beschreibung eines Vorgehensmodells, die strukturellen Zusammenhänge dieser Beschreibungskonzepte sowie Konsistenzbedingungen bezüglich der Anwendung der Beschreibungskonzepte fest. Damit dient das Meta-Modell als Modellierungssprache für Vorgehensmodelle. Wir bilden das Meta-Modell für Vorgehensmodelle zudem auf ein Modell für Projektpläne ab. Damit dient das Meta-Modell zugleich als Spezifikation einer Werkzeugunterstützung für die Ableitung von Projektplänen aus einem entsprechend dem Meta-Modell beschriebenen Vorgehensmodell.

Das Meta-Modell soll die Beschreibung aller Aspekte eines Vorgehensmodells erlauben, die sich aus der Abstraktion „guter“ Projektpläne ergeben. Die Ausdrucksmächtigkeit priorisieren wir dabei höher als die Einfachheit der Beschreibungsmittel. Wir nehmen eine Analyse der Schwachstellen des V-Modells XT als Ausgangspunkt für die Konzeption eines Meta-Modells für Vorgehensmodelle, die sich auf die Projektplanung als Anwendungsfall ausrichten.

Den methodischen Einsatz der Beschreibungsmittel, die Ausdrucksmächtigkeit der Beschreibungsmittel und die Stimmigkeit der erarbeiteten Konzepte zeigen wir anhand zahlreicher Beispiele auf Ebene des Vorgehensmodells sowie auf Projektebene. Zudem geben wir eine Methodik für die Durchführung von Projekten zur Erstellung von Vorgehensmodellen an. Neben der Erstellung unterstützen die konzipierten Beschreibungsmittel zudem die Änder- und Erweiterbarkeit von Vorgehensmodellen. Vorgehensmodelle sind oft „langlebige Systeme“. Zu diesem Zweck beinhaltet das Meta-Modell ein Modulkonzept.

Ein Spezialisierungskonzept ermöglicht die Beschreibung von Vorgehensmodellen auf unterschiedlichen Spezialisierungsebenen. Damit unterstützt das Meta-Modell die Beschreibung abstrakter Modelle sowie organisations- und domänenspezifischer Ausprägungen dieser abstrakten Modelle.

Abbildung 1–1 illustriert das Thema der Arbeit. Voraussetzung für die Ableitung eines Projektplans aus einem Vorgehensmodell ist die Eignung des Vorgehensmodells für diesen Zweck. Die Eignung bestimmt sich dabei unter anderem aus der geeigneten Strukturierung des Vorgehensmodells in Form eines Meta-Modells.

Unter einem Vorgehens-Meta-Modell verstehen wir ein Modell eines Vorgehensmodells. Im Vorgehens-Meta-Modell werden Beschreibungselemente wie *Produkt*, *Aktivität*, *Rolle* und *Ablauf* sowie strukturelle Beziehungen zwischen diesen Beschreibungselementen definiert. Konsistenzbedingungen schränken die Anwendung der Beschreibungselemente bei der Beschreibung eines Vorgehensmodells ein. Das Vorgehens-Meta-Modell dient dem Prozessingenieur bei der Erstellung und Pflege von Vorgehensmodellen als Sprache.

In Bezug auf die Elemente des Vorgehens-Meta-Modells definieren wir Regeln, die die projektspezifische Anpassung eines Vorgehensmodells und die Ableitung von Projektplänen vorgeben. Durch die projektspezifische Festlegung bestimmter Planungsparameter durch den Projektleiter kann somit aus einem Vorgehensmodell, das nach den Vorgaben des Vorgehens-Meta-Modells beschrieben ist, ein Projektplan abgeleitet werden. Im Kontext einer Werkzeugunterstützung kann das Vorgehens-Meta-Modell als logisches Datenmodell

für die Speicherung von Vorgehensmodellen gesehen werden, sowie als Spezifikation einer Werkzeugunterstützung für die Ableitung von Projektplänen.

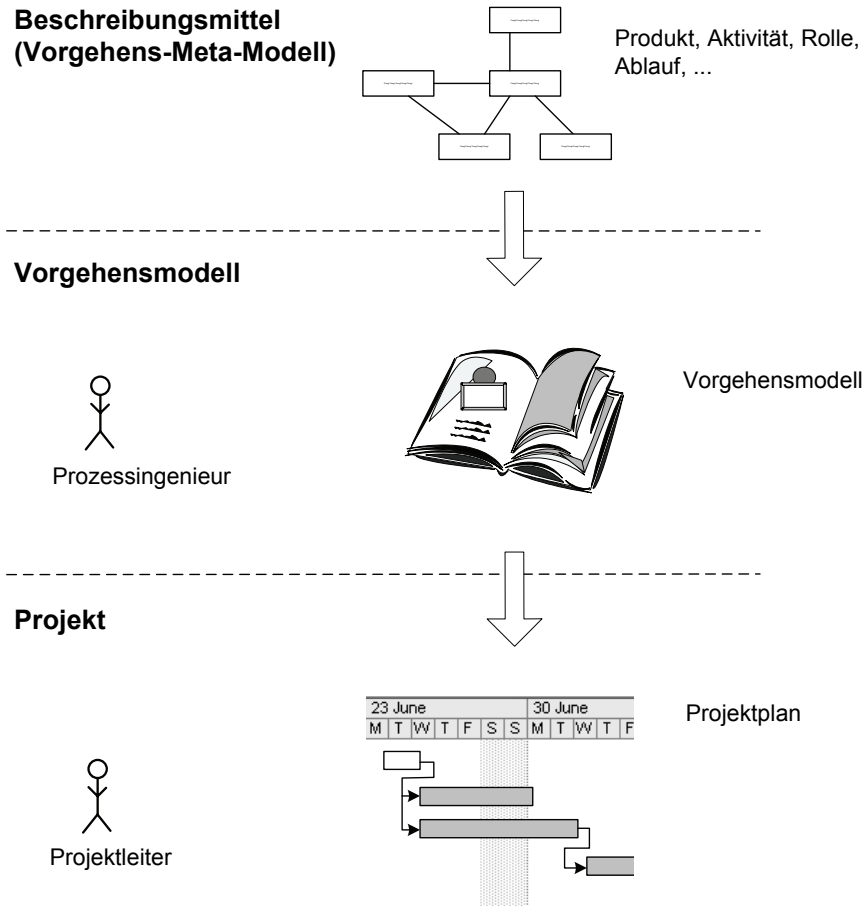


Abbildung 1–1: Thema der Arbeit

Abbildung 1–2 zeigt die Anwendungsfälle, die wir in der vorliegenden Arbeit adressieren. Organisationen, Prozessingenieure, Projektleiter und Entwickler sind die adressierten Zielgruppen.

Organisationen wie Unternehmen oder auch die öffentliche Hand sind an Vorgehensmodellen als verbindlichen Vorgaben für die Abwicklung von Entwicklungsprojekten interessiert. Dazu eignen sich Vorgehensmodelle, die Projektpläne auf abstrakter Ebene vorgeben. Eine Verifizierung der korrekten Anwendung des Vorgehensmodells im Projekt kann in einer Konformitätsprüfung des Projektplans bestehen.

Prozessingenieure sind mit der Erstellung beziehungsweise Pflege von Vorgehensmodellen befasst. Vorgehensmodelle werden selbst in Projekten erstellt. Für die arbeitsteilige Erstellung eines Vorgehensmodells sind die Verteilbarkeit von Aufgaben und damit die Planbarkeit des Erstellungsprozesses unabdingbar. Projekte zur Erstellung von Vorgehensmodellen benötigen ein Konzept zur Aufteilung eines Vorgehensmodells in modulare Bearbeitungseinheiten.

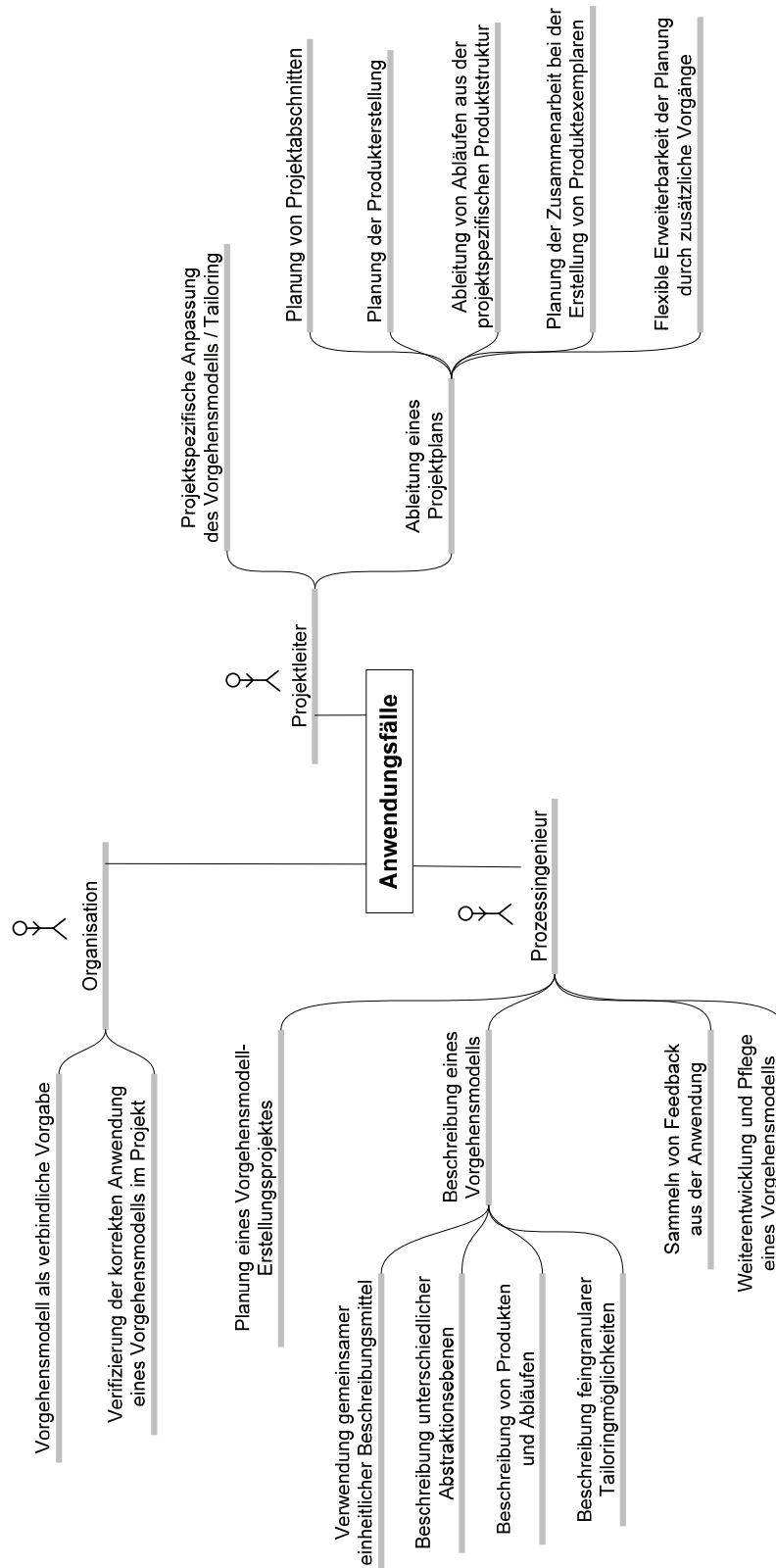


Abbildung 1–2: Adressierte Anwendungsfälle

Um eine einheitliche Beschreibung eines arbeitsteilig erstellten Vorgehensmodells zu erreichen, benötigen Prozessingenieure eine gemeinsame Sprache. Diese Sprache als Vehikel zur Beschreibung von Vorgehensmodellen wird als Vorgehens-Meta-Modell bezeichnet. Auch wenn ein Vorgehensmodell im Team erstellt wird sollte es aussehen wie „aus einem Guss“. Aus Sicht der einfachen Anwendbarkeit und letztlich der Qualität eines Vorgehensmodells kommt der Einheitlichkeit der Beschreibung eine große Bedeutung zu. Die Beschreibungsmittel sollen zudem „einfach“ sein, also leicht erlernbar und dem Prozessingenieur ein Hilfsmittel. Die Ziele Einfachheit und Einheitlichkeit sind allerdings teilweise konträr zum Ziel der Ausdrucksmächtigkeit. In der vorliegenden Arbeit priorisieren wir anders als beispielsweise im V-Modell XT die Ausdrucksmächtigkeit der Beschreibungsmittel höher als ihre Einfachheit und die Unterstützung der Einheitlichkeit des beschriebenen Vorgehensmodells. Die Beschreibungsmittel sollen jede als sinnvoll erachtete Abstraktion eines idealen Projektverlaufs beschreibbar machen.

Die Beschreibungsmittel sollen die Beschreibung von Vorgehensmodellen unterschiedlicher Abstraktionsebenen erlauben. Sowohl allgemeine abstrakte Vorgehensmodelle als auch abgeleitete organisations- oder domänenspezifische Ausprägungen dieser Vorgehensmodelle sollen beschrieben werden können. Ein Spezialisierungskonzept trägt dabei auch zur Vermeidung von Redundanzen bei.

Die Beschreibungsmittel sollen zudem die Beschreibung von Produkten und Abläufen als Vorgabe für die Projektplanung erlauben. Unter Produkten verstehen wir dabei die abstrakte Festlegung der Projektergebnisse, unter Abläufen Reihenfolgebeziehungen für Vorgänge, beispielsweise in Form von Vorgangfolgen der Netzplantechnik. Die Möglichkeiten des Tailorings sollen im Vorgehensmodell auf feingranularer Ebene beschreibbar sein. Unter dem Begriff Tailoring verstehen wir die projektspezifische Anpassung von Vorgehensmodellen durch die Auswahl der für ein Projekt relevanten Teile eines Vorgehensmodells.

Neben der Erstellung sind die Weiterentwicklung und Pflege von Vorgehensmodellen von Bedeutung. Wie im Falle des V-Modells handelt es sich bei vielen Vorgehensmodellen um „langlebige Systeme“. Für die Weiterentwicklung ist in Analogie zur Wartung von Software-Systemen die Lokalität von Änderungen im Vorgehensmodell auf Basis eines Modulkonzeptes wünschenswert. Als Anregung für Weiterentwicklungen dient dem Prozessingenieur Feedback aus der Anwendung im Projekt. Wünschenswert ist eine integrierte Werkzeugunterstützung für Vorgehensmodellierung und Projektplanung, die das Sammeln von Feedback aus der Anwendung eines Vorgehensmodells im Projekt unterstützt.

Projektleiter sind mit der Anwendung des Vorgehensmodells in einem Projekt betraut. Neben der projektspezifischen Anpassung des Vorgehensmodells durch Tailoring ist es Aufgabe des Projektleiters, aus einem Vorgehensmodell einen Projektplan abzuleiten. Im Projektplan muss der Projektablauf auf grobgranularer Ebene durch Projektabschnitte entsprechend den Vorgaben des Vorgehensmodells strukturiert werden. Auf Basis der im Vorgehensmodell beschriebenen und im Projekt zu erstellenden Produkte ist die Produkterstellung zu planen.

Bestimmte Abläufe im Sinne von Reihenfolge-Beziehungen der Vorgänge zur Produkterstellung können aus der projektspezifischen Produktstruktur abgeleitet werden. Bestimmte im Vorgehensmodell beschriebene Inhalte der Produktexemplare implizieren Ablauf-Beziehungen im Projektplan. Legt beispielsweise ein Software-Architekturdokument Inklusions-Beziehungen zwischen Komponenten fest, kann daraus eine Reihenfolge für Integration und Test der Komponenten abgeleitet werden.

An der Erstellung eines Produktes können im Projekt mehrere Bearbeiter beteiligt sein. Dies zu planen ist Aufgabe des Projektleiters. Ein Vorgehensmodell als Basis für die Projektplanung sollte Muster zur Planung der Zusammenarbeit mehrerer Bearbeiter bei der Erstellung von Produktexemplaren beinhalten. Beispielsweise erfordern Konsistenzabhängigkeiten zwischen Produkten im Allgemeinen eine Planung der Zusammenarbeit der jeweiligen Bearbeiter.

Vorgehensmodelle als Abstraktionen der Projektebene können in der Praxis hinsichtlich aller zu planenden Vorgänge nicht vollständig sein. Ein Projektplan muss also zudem durch Vorgänge flexibel erweiterbar sein, die sich nicht aus dem Vorgehensmodell ableiten. Entsprechende Erweiterungen auf Projektebene können den Prozessingenieuren als Feedback für die Weiterentwicklung des Vorgehensmodells dienen.

„Out of Scope“

Zudem geben wir eine Reihe von Anwendungsfällen an, die wir in der Arbeit nicht betrachten. Wir konzipieren Vorgehensmodelle nicht als Hilfestellung für den einzelnen Entwickler. Vorgehensmodelle dienen hier vorrangig dem Projektleiter bei der Erstellung einer Projektplanung. Der Beschreibung von Methodik für die Entwicklung von IT-Systemen, also der Beschreibung der Aufgaben der Rolle des Entwicklers gilt dabei nicht das Hauptaugenmerk. Allerdings sind auch aus Sicht der Projektplanung Entwurf und Realisierung einer Software im Vorgehensmodell zu beschreiben, da beispielsweise ein Architekturdokument dem Projektleiter Einheiten für die Planung der Realisierung nahe legt.

Vorgehensmodelle können Werkzeugen für die Automatisierung von Workflows als Eingabe dienen. Im Gegensatz zu verwandten Arbeiten betrachten wir nicht die „Ausführung“ eines Vorgehensmodells in Form von „Task-Management“. Task-Management beziehungsweise so genannte „prozessorientierte Softwareentwicklungsumgebungen“ (vgl. dazu Kapitel 1.4) dienen der Unterstützung des einzelnen Entwicklers bei der Auswahl und Reihung zu erledigender Aufgaben.

Der ingenieurmäßige Ablauf eines Entwicklungsprojektes erfordert allerdings das Hinterfragen und die Anpassung der verwendeten Entwicklungsmethodik. Ein Vorgehensmodell sollte also auf der Ebene der eigentlichen Entwicklungsarbeit keine zu strikte Vorgabe sein. Entwicklung ist im Gegensatz zu industrieller Produktion stets mit Innovation verbunden und erfordert daher in gewissem Umfang Kreativität und in jedem Fall immer Abweichungen vom zuvor Entwickelten. Daher erachten wir eine Workflow-Unterstützung des Entwicklungsprozesses als nicht zielführend, betrachten sie also nicht weiter im Rahmen der Arbeit. Als sinnvolle Granularität für die Vorgaben eines Vorgehensmodells sehen wir den Projektplan an. Ein Projektplan ist dabei im Projektablauf nicht unumstößlich, sondern kann bei Bedarf angepasst werden.

Wir widmen uns zudem nicht der Beschreibung eines Rollenmodells. Ein Rollenmodell ist Bestandteil der meisten Vorgehensmodelle. Im Rahmen der Projektplanung dient ein Rollenmodell der Formulierung von Anforderungen hinsichtlich der Zuordnung geeigneter menschlicher Ressourcen zu Vorgängen. Die Konzepte des Rollenmodells stellen allerdings für die vorliegende Arbeit keine Herausforderung bereit und werden daher nicht weiter betrachtet.

Wir konzipieren Beschreibungsmittel für Vorgehensmodelle nicht im Hinblick auf die Erstellung einer Papierfassung eines Vorgehensmodells. Wir geben auch keine Abbildung in eine solche Form an. Grafische oder textuelle Notationen für strukturelle Aspekte eines Vorgehensmodells werden nicht thematisiert.

1.2 Beitrag der Arbeit

Viele der heute verbreiteten Vorgehensmodelle sind als Lektüre konzipiert. In der vorliegenden Arbeit modellieren wir Vorgehensmodelle präzise. Wir definieren die Semantik des Vorgehens-Meta-Modells und damit die Semantik des Vorgehensmodells selbst mit dem Ziel der Unterstützbarkeit durch Werkzeuge. Als Semantik bezeichnen wir die Abbildung einer Sprache in eine semantische Domäne, wie in Abbildung 1–3 dargestellt.

Als semantische Domäne des Vorgehens-Meta-Modells dient ein Modell für Projektpläne. Dieses erscheint als semantische Domäne geeignet, da es sich dabei um ein vergleichsweise einfach strukturiertes und im Allgemeinen gut verstandenes Modell handelt. Der Projektplan enthält im Gegensatz zum Vorgehensmodell beispielsweise keinen Tailoring-Mechanismus, keine Produktabhängigkeiten, keine Iterationen und keine generischen Vor-

gehensweisen. Unter generischen Vorgehensweisen verstehen wir dabei Aktivitäten im Vorgehensmodell, die sich nicht auf ein Projektergebnis, sondern auf alle Projektergebnisse beziehen, wie beispielsweise die Qualitätssicherung.

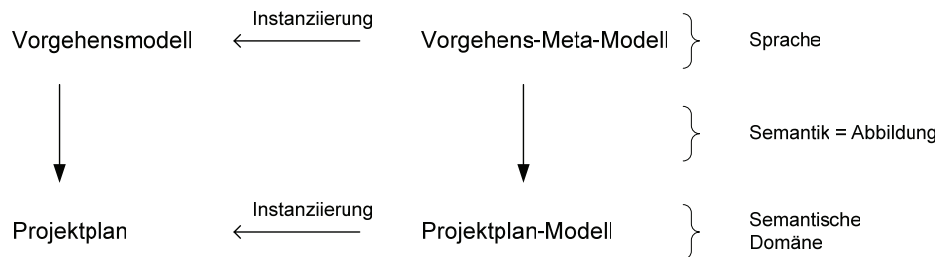


Abbildung 1–3: Präzise Definition der Semantik von Vorgehensmodellen

Bei der Konzeption des Vorgehens-Meta-Modells verfolgen wir einen pragmatischen Ansatz. Wir definieren das Meta-Modell ausgehend von Beispielen. Wir betrachten Meta-Modellierung und die Untersuchung von Modellierungstechniken nicht als Selbstzweck, sondern ziehen eine praxisgetriebene Modellbildung dem Aspekt der Eleganz und Schönheit eines Meta-Modells vor. Der Beitrag der Arbeit ist daher vor allem auch pragmatischer Natur.

Die Ergebnisse der vorliegenden Arbeit sind für unterschiedliche Zielgruppen von Interesse.

- Aus Sicht des **Projektleiters** eines Entwicklungsprojektes vereinfacht ein Vorgehensmodell entsprechend dem Konzept der Arbeit die Erstellung von Projektplänen. Anspruchsvolle Aufgabe der Projektplanung ist die Zerlegung der Gesamtaufgabe in handhabbare Arbeitspakete mit klar definierten Abhängigkeiten. Selbst bei kleinen Entwicklungsprojekten sind die Abhängigkeiten mannigfaltig und von nicht zu unterschätzender Komplexität. Die Strukturierung eines Projektes in Form eines Projektplans ist eine intellektuelle beziehungsweise kreative Leistung des Projektleiters, die vor allem auf Erfahrungswissen beruht. Dies birgt im Falle eines unerfahrenen Projektleiters das Risiko des Scheiterns eines Projektes in sich. Entsprechend dem Konzept der vorliegenden Arbeit dient ein Vorgehensmodell als konserviertes Erfahrungswissen der Projektplanung. Die Anwendung des Vorgehensmodells in Form von Tailoring und Ableitung eines Projektplans kann dabei durch ein Werkzeug unterstützt werden.
- Eine präzise Definition der Semantik des Vorgehens-Meta-Modells dient aus Sicht der **Prozessingenieure** dem gemeinsamen Verständnis der im Vorgehensmodell zu beschreibenden Elemente. Das konzipierte Vorgehens-Meta-Modell stellt eine Sprache zur Beschreibung von Vorgehensmodellen dar. Das erarbeitete durchgängige Beispiel illustriert und dokumentiert die Anwendung der Konzepte. Die Konzepte sind dabei einerseits einfach anwendbar, zum anderen aber auch entsprechend dem verwendeten Modellierungsansatz in ihrer Ausdrucksmächtigkeit nahezu beliebig erweiterbar.

Ein Spezialisierungskonzept unterstützt die Modellierung abstrakter Vorgehensmodelle sowie organisations- und domänenspezifischer Ausprägungen eines abstrakten Modells.

Die Methodik für die Erstellung von Vorgehensmodellen beschreibt die Durchführung eines entsprechenden Projektes auf Granularität der Projektplanung. Ein Modulkonzept ermöglicht einem Projektleiter die Planung der arbeitsteiligen Erstellung eines Vorgehensmodells. Im Rahmen der Weiterentwicklung eines Vorgehensmodells unterstützt das Modulkonzept die Lokalität beziehungsweise die Verfolgung der Auswirkungen von Änderungen.

Wesentliche Aufgabe des Prozessingenieurs ist es, die Lebendigkeit eines Vorgehensmodells durch praktisch anwendbare und auch tatsächlich angewendete Inhalte des

Vorgehensmodells sicherzustellen. Die Integration von Vorgehensmodell-Editor und Projektplanungskomponente in einer Werkzeugumgebung ermöglicht den Prozessingenieuren die Aufsammlung von Feedback unmittelbar aus der Anwendung.

- Weitere Zielgruppe der Arbeit sind **Organisationen**, die Entwicklungsprojekte durchführen. Über den Nutzen von Vorgehensmodellen als Hilfestellung hinaus ist Organisationen wie Unternehmen oder der öffentlichen Hand an verbindlichen Vorgaben gelegen. Durch die präzise Modellierung eines Vorgehensmodells in Form eines abstrakten Projektplans kann diese Verbindlichkeit erreicht werden. Die konzipierte Werkzeugunterstützung für die Projektplanung ermöglicht zugleich die Validierung der „ordnungsgemäßen“ Anwendung eines Vorgehensmodells. Abweichungen des abgeleiteten Projektplans von den Vorgaben eines Vorgehensmodells können berechnet werden.

Nicht zuletzt untersuchen wir in der Arbeit das V-Modell XT. Das für das V-Modell XT konzipierte Meta-Modell sowie die Grundmechanismen für Tailoring und Projektplanung werden kritisch hinterfragt. Wir identifizieren Verbesserungspotential und zeigen Lösungsansätze zu den identifizierten Problemen auf. Dabei beziehen wir uns vor allem auf das Meta-Modell, das im Vergleich zu den Inhalten des V-Modells XT langlebiger sein wird. Wir stellen allerdings nicht explizit den Bezug zu den Konzepten des V-Modells XT her, sondern konzipieren ein Vorgehens-Meta-Modell im Rahmen der Arbeit von Grund auf neu. Dieses Vorgehens-Meta-Modell ist allein als Verbesserungsansatz hinsichtlich der werkzeuggestützten Ableitbarkeit eines Projektplans aus einem Vorgehensmodell zu sehen. Das V-Modell XT verfolgt zudem das Ziel eines übersichtlichen Nachschlagewerks für alle Projektbeteiligten, dem wir uns in der Arbeit nicht widmen.

Die Arbeit leistet zudem einen Beitrag als Anwendungsbeispiel der UML und des Modellierungsansatzes der Meta-Modellierung. Soweit die aktuellen Spezifikationen der UML für Klassendiagramme beziehungsweise der OMG zur Meta-Modellierung hinsichtlich der in der Arbeit benötigten Konzepte unpräzise sind, definieren wir die Semantik der verwendeten Konzepte im Rahmen der Arbeit präzise.

1.3 Aufbau der Arbeit

Wir geben einen Überblick über die Gliederung der folgenden Kapitel.

Kapitel 2 – State-Of-The-Art: Vorgehensmodelle und Projektplanung

In diesem Kapitel stellen wir den State-of-the-Art auf dem Gebiet der Vorgehensmodelle und Projektplanung dar. Dabei definieren wir die für die Arbeit wesentlichen Begriffe und Grundlagen. Um den Begriff des Vorgehensmodells plastisch zu machen, geben wir zunächst einen Überblick über ausgewählte Vorgehensmodelle von den historischen Wurzeln bis hin zu jüngsten Entwicklungen. Die betrachteten Vorgehensmodelle werden dabei in ein allgemeines Schema zur Klassifikation eingeordnet. Wir geben zudem eine Begriffsdefinition für Vorgehensmodelle in Form einer Ontologie der Elemente an, die zur Beschreibung von Vorgehensmodellen üblicherweise herangezogen werden.

Einen wesentlichen Aspekt des Themas der Arbeit stellt der Themenkomplex Projektmanagement und Projektplanung dar. Wir stellen das in der Literatur verbreitete methodische Vorgehen bei der Erstellung von Projektplänen dar, unter anderem die Erstellung von Produkt- und Projektstrukturplänen, sowie die Netzplantechnik. Dies führt uns entsprechend den Betrachtungen über Vorgehensmodelle zu einer Begriffsdefinition, die wir in Form einer Ontologie des Aufbaus von Projektplänen angeben. Die Darstellung der praktischen Probleme der Projektplanung hebt die erforderliche intellektuelle beziehungsweise kreative Leistung des Projektleiters bei der Erstellung eines Projektplans hervor. Diese beruht vor allem auf Erfahrungswissen. Dies birgt im Falle eines unerfahrenen Projektleiters das Risiko eines

Scheiterns des Projektes in sich. In der vorliegenden Arbeit erstellen wir daher ein Konzept für die Konservierung des mit der Planung verbundenen Erfahrungswissens in Form eines Vorgehensmodells.

Kapitel 3 – Schwachstellenanalyse des V-Modells XT

In diesem Kapitel analysieren wir eine Reihe von Schwachstellen des V-Modells XT, die sich aus dem Meta-Modell des V-Modells XT ergeben. Die Beschreibungsmittel des V-Modells XT führen teilweise zu einer ungünstigen inhaltlichen Umsetzung des V-Modells. Die Folge sind Verständnisprobleme auf Seiten der Anwender, Schwierigkeiten im Rahmen des Weiterentwicklungsprozesses sowie eine eingeschränkte Unterstützbarkeit des V-Modells durch Werkzeuge. Insbesondere hinsichtlich der Unterstützbarkeit des V-Modells durch ein Werkzeug für die Projektplanung decken wir Schwachstellen auf.

Wir beschreiben die Konzepte des Meta-Modells zu Vorgehensbausteinen, Abläufen und Produkten. Außerdem betrachten wir Abstraktionsprinzipien im V-Modell XT. Eine entsprechend präzise Darstellung des Meta-Modells des V-Modells XT findet sich derzeit weder im V-Modell XT selbst, noch in sonstigen Publikationen.

Die identifizierten Schwachstellen und Probleme des V-Modells XT erläutern wir jeweils anschaulich anhand von Beispielen. Das Kapitel dient uns als Ausgangspunkt für die Konzeption eines verbesserten Meta-Modells für Vorgehensmodelle, das der Zielsetzung der Arbeit gerecht wird.

Kapitel 4 – Modellierungstechnik für Vorgehensmodelle

In diesem Kapitel legen wir den Grundstein für die Modellierung von Vorgehensmodellen. Wir definieren die in den folgenden Kapiteln verwendete Modellierungstechnik für Vorgehensmodelle. Als Modellierungstechnik verwenden wir Klassendiagramme und Constraints in Form prädikatenlogischer Ausdrücke. Da Klassendiagramme in der UML teilweise nicht präzise definiert sind, beziehungsweise bestimmte Aspekte der UML in der Arbeit spezifisch eingesetzt werden, definieren wir in diesem Kapitel bestimmte Aspekte dieser Modellierungstechnik semantisch präzise. Dabei legen wir Instanziierung sowie die Generalisierung und Spezialisierung von Klassen und Assoziationen fest. Als Modellierungsansatz für die Modellierung von Vorgehensmodellen dient in den folgenden Kapiteln die Einteilung in Meta-Ebenen, die sich durch das Konzept der Spezialisierung abgrenzen.

Kapitel 5 – Modellierung von Vorgehensmodellen

Das Kapitel widmet sich dem Kernthema der Arbeit. Wir entwerfen sukzessive ein Vorgehens-Meta-Modell, das die Anwendung im Rahmen der Projektplanung und die Unterstützbarkeit durch Werkzeuge für die Projektplanung ermöglicht. Wir nehmen in diesem Kapitel nicht das Meta-Modell des V-Modells XT als Ausgangspunkt, sondern konzipieren ein Meta-Modell von Grund auf neu.

Wir gliedern die Darstellung dieses Vorgehens-Meta-Modells in ein Strukturmodell und ein dynamisches Modell. Wir definieren Konzepte zur Beschreibung von Vorgehensmodellen wie beispielsweise Produkt, Produktabhängigkeit, Vorgang oder Projektabschnitt sowie deren Beziehungen untereinander. Wir illustrieren die Anwendung der Konzepte des Vorgehens-Meta-Modells anhand eines durchgängigen Beispiels auf Ebene des Vorgehensmodells sowie auf Projektebene.

Zudem betrachten wir den Modellierungsansatz der Unterteilung in (Meta-)Ebenen im Zusammenhang mit dem in der Arbeit verwendeten Spezialisierungskonzept. Die Ebenen Vorgehens-Meta-Modell, Vorgehensmodell und Projekt grenzen sich durch das in der Modellierungstechnik definierte Konzept der Spezialisierung voneinander ab. Spezialisierung ist

allerdings auch auf der Ebene des Vorgehensmodells von Bedeutung. In einem Entwicklungsprojekt stellt Modellbildung eine gängige Technik dar. Modellbildung, beziehungsweise Generalisierung als Gegenstück zur Spezialisierung, muss also auch im Vorgehensmodell beschreibbar sein, insbesondere da Modellbildung einen Einflussfaktor auf die Projektplanung darstellt. Dies zeigen wir anhand eines Beispiels.

Spezialisierung ist zudem aus dem Blickwinkel der Vermeidung von Redundanz im Vorgehensmodell erforderlich. Nicht zuletzt unterstützt ein Spezialisierungskonzept die Modellierung organisations- und domänenspezifischer Ausprägungen eines abstrakten Vorgehensmodells. Dies zeigen wir anhand eines Beispiels aus dem V-Modell XT. Als alleinige Beziehung zwischen den Ebenen Vorgehens-Meta-Modell und Vorgehensmodell ist Spezialisierung nicht ausreichend. Dies verdeutlichen wir am Beispiel einer generischen Vorgehensweise zur Qualitätssicherung. Generische Vorgehensweisen wie die Qualitätssicherung beziehen sich beispielsweise auf alle Exemplare eines Produktes.

Kapitel 6 – Erstellung und Anwendung von Vorgehensmodellen

In diesem Kapitel betrachten wir die Anwendung der im vorangehenden Kapitel erarbeiteten Konzepte zur Modellierung von Vorgehensmodellen. Unter Erstellung verstehen wir die Anwendung der konzipierten Beschreibungsmittel im Erstellungsprozess eines Vorgehensmodells. Anwendung bedeutet die Anwendung des Vorgehensmodells in einem Projekt, also im Kontext der Arbeit die Ableitung eines Projektplans.

Wir entwerfen zunächst ein Konzept zur Strukturierung der Inhalte eines Vorgehensmodells. Diese Strukturierung der Inhalte ist Voraussetzung für die Definition von Tailoring-Einheiten, also für die projektspezifische Anpassung eines Vorgehensmodells, sowie für die Definition von Bearbeitungs-Einheiten im Erstellungsprozess eines Vorgehensmodells. Das Konzept zur inhaltlichen Strukturierung und zu Tailoring-Einheiten illustrieren wir anhand zahlreicher Beispiele, unter anderem anhand von Beispielen aus dem V-Modell XT. Den Mechanismus zur projektspezifischen Anpassung eines Vorgehensmodells und Ableitung eines Projektplans definieren wir präzise mit Hilfe von Constraints.

Aufbauend auf dem Modell zur inhaltlichen Strukturierung definieren wir ein Konzept für modulare Bearbeitungs-Einheiten für die Erstellung eines Vorgehensmodells. Dieses bildet die Grundlage einer Methodik für die Erstellung eines Vorgehensmodells auf Basis der Konzepte des erarbeiteten Vorgehens-Meta-Modells.

Im Kontext der Anwendung von Vorgehensmodellen sind beschreibende Texte zu den modellierten Elementen eines Vorgehensmodells von Bedeutung. Vor dem Hintergrund des Tailorings konzipieren wir diese als Beschreibungs-Bausteine, die die „Modellierung von Texten“ erlauben.

Abschließend stellen wir ein Werkzeugkonzept für die Erstellung und Anwendung von Vorgehensmodellen vor, das die in der Arbeit definierten Konzepte zur Anwendung bringt.

Kapitel 7 – Zusammenfassung und Ergebnisse

Dieses Kapitel fasst die wesentlichen Ergebnisse der Arbeit zusammen und bewertet deren Anwendbarkeit.

Anhang A – Vorgehens-Meta-Modell

Im Anhang A zeigen wir das in den vorangehenden Kapiteln sukzessive erarbeitete und jeweils in Ausschnitten dargestellte Vorgehens-Meta-Modell im Überblick.

1.4 Vergleich verwandter Arbeiten

Wir betrachten in diesem Kapitel eine Reihe verwandter Arbeiten und positionieren die vorliegende Arbeit in deren Kontext.

Es gibt einige kommerzielle Werkzeuge für die Projektplanung, wie beispielsweise Microsoft Project (vgl. [MSProj03]) oder Primavera Project Planner (vgl. [Primav04]). Neben dem Anwendungsfall Projektpläne neu zu erstellen, bietet beispielsweise Microsoft Project eine Reihe von Musterprojektplänen. Dabei handelt es sich um Sammlungen typischer Vorgänge für unterschiedliche Projekttypen. Ein Musterprojektplan kann durch Kopieren und geeignetes Benennen von Vorgängen projektspezifisch angepasst werden. Regeln für die Anpassung werden durch Musterprojektpläne allerdings nicht unterstützt. Ein Konzept zur Instanziierung ist nicht Bestandteil eines Musterprojektplans.

**Projektplanungs-
Werkzeuge**

Meta-Modelle geben eine Sprache für die Beschreibung von Vorgehensmodellen beziehungsweise Prozessmodellen vor. Die Begriffe Vorgehensmodell und Prozessmodell verstehen wir im Kontext der Arbeit als synonym. Ein Meta-Modell für die Beschreibung von Prozessmustern findet sich beispielsweise in [GMP+03] oder auch in [DGH02]. Die Konzepte einer Werkzeugunterstützung für Prozessmuster sind allerdings nur rudimentär ausgearbeitet (vgl. [GMP+02]). Weitere Beispiele von Meta-Modellen beziehungsweise Prozessmodellierungssprachen aus dem akademischen Umfeld sind PROMENADE, das als Erweiterung der UML konzipiert ist (vgl. [RF00]), oder SPEARMINT (vgl. [BHK+98]). Zielrichtung dieser Ansätze ist einzig die Beschreibung von Vorgehensmodellen.

Meta-Modelle

Im kommerziellen Umfeld findet sich als Meta-Modell für die Vorgehensmodelle das so genannte „Software Process Engineering Metamodel“ (vgl. [SPEM01]). Dieses wird durch die „Rational Process Workbench“ (RPW) unterstützt (vgl. [RPW04]). Anwendungsfall ist dabei der Rational Unified Process (vgl. [KK03]), bei dessen Modellierung die RPW zum Einsatz kommt. Werkzeugunterstützung über das Editieren und Tailoring eines Vorgehensmodells hinaus ist im Kontext von SPEM und RPW nicht vorgesehen.

Weitere Beispiele sind das „Process Interchange Format“ (vgl. [LGJ+98]), die „Process Specification Language“ für Fertigungsprozesse (vgl. [SKR98]), das „Workflow Reference Model“ der Workflow Management Coalition (WfMC) als Standard für die Beschreibung von Workflows (vgl. [WfMC95] und [WfMC99]) oder ARIS für Modellierung und Analyse von Geschäftsprozessen (vgl. [Scheer98]), um nur einige Beispiele zu nennen. Diese Modelle zielen allesamt nicht auf eine Werkzeugunterstützung für die Projektplanung ab. Ein Überblick zu Meta-Modellen kann beispielsweise [BB00] und [BB01] entnommen werden.

Der Ansatz in [Chroust94] beziehungsweise [Chroust00] spezifiziert ein Kalkül auf Basis eines Meta-Modells, mit dem aus partiellen Vorgehensmodellen ein Vorgehensmodell synthetisiert werden kann. Operationen wie Differenz, Substitution oder Vereinigung von Vorgehensmodellen dienen diesem Zweck. Eine Untersuchung der Praxistauglichkeit in Form eines Fallbeispiels ist der Arbeit nicht zu entnehmen. Aufgrund der mannigfaltigen Abhängigkeiten innerhalb eines Vorgehensmodells erachten wir einen Markt für Vorgehensmodell-Fragmente, die zu einem stimmigen Vorgehensmodell synthetisiert werden können (vgl. [Chroust00]), für nicht realisierbar.

Im akademischen Umfeld widmeten sich Anfang der 90er viele Arbeiten den so genannten „Process-centered Software Engineering Environments“ (PSEE) (vgl. [CG98]). Dabei handelt es sich um Softwareentwicklungsumgebungen, die den Benutzer durch eine möglichst umfangreiche Automatisierung von Aufgaben unterstützen. Ein PSEE stützt sich dabei auf ein explizit hinterlegtes Prozessmodell ab, das entsprechend einer bestimmten Prozessmodellierungssprache beschrieben wird. Vergleichbar einer Workflow-Maschine zielt ein PSEE auf das Verwalten anstehender Aufgaben eines Bearbeiters, die Unterstützung der Zusammenarbeit der Bearbeiter sowie die Ausführung von im Softwareentwicklungsprozess benötigten Werkzeugen ab.

**Process-centered
Software
Engineering
Environments
(PSEE)**

Die eingesetzten Prozessmodellierungssprachen fallen dabei in unterschiedliche Kategorien. Unterschieden werden können regelbasierte Sprachen wie Merlin (vgl. [JPS+94]), Sprachen

basierend auf Petri-Netzen wie FUNSOFT-Netze (vgl. [EG90]) oder beispielsweise APPL/A basierend auf der Programmiersprache Ada (vgl. [SHO95]). Als eine Mischform kann SPADE gelten (vgl. [BFG91]). Prozesse werden in SPADE als Petri-Netze dargestellt, Prozessergebnisse werden objektorientiert modelliert und die mit den Transitionen des Petri-Netzes verbundenen Aktionen werden in einer operationalen Sprache beschrieben.

In EPOS werden Prozessinstanzen mit Hilfe von Techniken der künstlichen Intelligenz abgeleitet (vgl. [CHL+94] und [LC93]). EPOS verwendet die Prozessmodellierungssprache SPELL, die für die zu erledigenden „Tasks“ Ein- und Ausgangsprodukte sowie Vor- und Nachbedingungen definiert. Die Planungskomponente in EPOS ermittelt zu einem Prozessmodell und einem auf Projektebene instanziierten Produktmodell ein passendes „Task network“, das die jeweiligen Produktexemplare erstellt. Mit Techniken wie „backward chaining“ werden Tasks so verkettet, dass jeweils benötigte Eingangsprodukte als Ausgangsprodukte in vorangehenden Tasks erstellt werden und Vorbedingungen durch Nachbedingungen vorangehender Tasks erfüllt werden. Als Beispiel wird ein Prozess für das Übersetzen von Quellcode und das anschließende Binden dargestellt. Mit Hilfe definierter Abhängigkeiten der Codefragmente ist dieser Prozess automatisierbar.

Der in MILOS verfolgte Ansatz reichert die Workflow-Funktionalität der PSEEs um Funktionalität für die Projektplanung an (vgl. [GMH00] und [MDBG+00]). Die Planung eines Prozesses kann kooperativ durch mehrere Personen durchgeführt werden. Ein Notifikationsmechanismus kommuniziert Pläne und Planänderungen an die Bearbeiter sowie Planabweichungen an die Ersteller der Planung. Die Prozessbeschreibungssprache basiert auf Aktivitäten mit Vor- und Nachbedingungen in Bezug auf die Zustände von Produkten sowie alternativen Methoden für die Durchführung von Aktivitäten. Prozess-Modellierung, Planung und Workflow-Unterstützung können „verschränkt“ erfolgen. Während der Ausführung eines Workflows ist also sowohl eine Anpassung des Prozessmodells als auch eine Anpassung des Plans möglich.

Die Prozessmodellierungssprache DYNAMITE definiert so genannte dynamische Aufgabennetze mit den Konzepten Produktfluss und Kontrollfluss (vgl. [HJK+96]). Die Ausführung eines dynamischen Aufgabennetzes wird entsprechend Petri-Netzen durch die Aktivierung „bereiter“ Aktivitäten formalisiert. Jeder Durchlauf durch das Aufgabennetz erzeugt neue Versionen der jeweiligen Aufgaben und Produkte. Regeln für die Ausführung von Aufgabennetzen werden anders als bei Petri-Netzen mit Hilfe von schema-erhaltenden Graphgrammatiken festgelegt (vgl. [SWZ95]).

Das AHEAD-System stützt sich auf die Sprache DYNAMITE ab und stellt dem Benutzer eine aus den Regeln einer Graphgrammatik generierte Benutzeroberfläche für die Planung, Überwachung und das Konfigurationsmanagement von Entwicklungsprozessen zur Verfügung (vgl. [Westfe01]). Durch ein entsprechendes „Matching“ der linken Seite einer Regel im Aufgabennetz kann das Aufgabennetz beispielsweise so transformiert werden, dass ein Produkt als Eingabe einer Aktivität konsumiert wird. Mit Hilfe von Regeln, die strukturelle Änderungen des Aufgabennetzes abhängig vom Ausführungszustand beschreiben, ist das verschränkte Editieren und Ausführen von Aufgabennetzen möglich. AHEAD unterstützt zudem die Evolution von Prozessmodellen. Tasks können von den Vorgaben eines Prozessmodells abweichen und zu einem späteren Zeitpunkt in das Prozessmodell integriert werden (vgl. [HSW03]).

Beispiel in-Step

Als Beispiel eines moderneren PSEE aus dem kommerziellen Umfeld betrachten wir das Werkzeug in-Step der Firma Microtool (vgl. [in-Step]). In-Step integriert Workflow-Unterstützung, Projektplanungs- und Konfigurationsmanagement-Funktionalität auf Basis eines Vorgehensmodells. Vorgehensmodelle werden entsprechend einem festgelegten Meta-Modell definiert und für ein Projekt instanziiert.

Das Meta-Modell von in-Step orientiert sich dabei am Meta-Modell des V-Modells 97 (vgl. [VM97a]). Produkte und Aktivitäten werden entsprechend diesem Meta-Modell eigenständig definiert. Der Produktfluss einer Aktivität legt Eingangs- und Ausgangsprodukte sowie die jeweiligen Eingangs- und Ausgangs-Zustände dieser Produkte fest. Zusätzlich kann ein

Kontrollfluss der Aktivitäten in Form von Vorgangsfolgen der Netzplantechnik definiert werden. Für Produkte und Aktivitäten des Vorgehensmodells als auch für weitere Elemente wie beispielsweise Anforderungen oder Risiken können Zustandsautomaten definiert werden. Diese spiegeln den jeweiligen Bearbeitungszustand in Bezug auf das Konfigurationsmanagement wider und legen zudem Aktionen der Workflow-Unterstützung wie beispielsweise die Benachrichtigung eines Bearbeiters oder die Ausführung beliebiger Programme fest. Für die Instanziierung der Produkte im Projekt lassen sich im Vorgehensmodell Kardinalitäten festlegen. Ein Konzept für Produktabhängigkeiten (vgl. dazu Kapitel 5.3.2) ist allerdings nicht vorgesehen.

Neben der Führung der Projektmitarbeiter entsprechend dem aus dem Vorgehensmodell abgeleiteten Workflow und der Repository-Funktionalität für Produktexemplare realisiert in-Step Funktionalitäten eines Projektmanagement-Werkzeugs. Für die instanziierten Aktivitäten kann eine Terminplanung erstellt werden. Die Erfassung der Arbeitszeiten der Projektmitarbeiter bezüglich der geplanten Aktivitäten sowie Änderungen des Bearbeitungszustands durchgeführter Aktivitäten durch die Projektmitarbeiter dienen der Projektkontrolle.

Der in der vorliegenden Arbeit verfolgte Ansatz ist allein auf das Thema Projektplanung zugeschnitten. Das konzipierte Vorgehens-Meta-Modell ist daher im Vergleich zu in-Step wie zu den übrigen PSEEs einfacher. Beispielsweise sieht die Produkt-Aktivitäts-Schnittstelle der vorliegenden Arbeit für jedes zu erstellende Produkt genau einen Vorgang zur Produkterstellung vor. Auf der Ebene der Produktbearbeitungen lassen sich einer Produkterstellung mehrere Vorgänge zuordnen, soweit diese für die Planung relevant sind. Die Definition eines Zustandsautomaten der Aktivitäten oder Produkte ist für die Ableitung eines Projektplans beispielsweise nicht zweckdienlich. Als nachteilig einzustufen ist die Komplexität der Prozessmodellierungssprache eines PSEE, die zu einem hohen Aufwand bei der Erstellung oder Weiterentwicklung eines Vorgehensmodells führt. Die Einspeisung von Erfahrungen eines Projektes in ein organisationspezifisches Vorgehensmodell und damit die „Lebendigkeit“ eines Vorgehensmodells gestalten sich beispielsweise mit in-Step sehr aufwendig.

Andererseits übertreffen wir mit dem Ansatz der vorliegenden Arbeit die Möglichkeiten von in-Step wie die der meisten PSEEs an Ausdrucksmächtigkeit. Ein Vorgehensmodell kann unabhängig vom definierten Vorgehens-Meta-Modell entsprechend dem Ansatz der vorliegenden Arbeit durch Spezialisierung der Konzepte der Modellierungstechnik basierend auf Klassen, Assoziationen und Constraints durch den Prozessingenieur beliebig erweitert werden (vgl. dazu Kapitel 5.4.1). Zudem kann ein Vorgehensmodell mehrere Instanzierungsbeziehungsweise Spezialisierungsebenen in sich vereinigen, um beispielsweise organisationspezifische Ausprägungen modellieren oder Redundanz im Vorgehensmodell vermeiden zu können (vgl. dazu Kapitel 5.4.2). Generische Vorgehensweisen wie die Qualitätssicherung können wir präzise beschreiben. Zwar erlaubt in-Step die Modellierung von Beziehungen zwischen Produktexemplaren, allerdings nicht von Beziehungen der Produkte auf Ebene des Vorgehensmodells. Auch Beziehungen zwischen Elementen des Vorgehensmodells und Elementen des Meta-Modells können mit in-Step nicht modelliert werden. Im Kontext generischer Vorgehensweisen existiert kein Konzept um beispielsweise die Beziehung eines Prüfprotokolls und des jeweils geprüften Produktexemplars auf Ebene des Vorgehensmodells als Vorgabe für die Projektebene zu modellieren (vgl. dazu Kapitel 5.4.3).

Positionierung der vorliegenden Arbeit

2 State-Of-The-Art: Vorgehensmodelle und Projektplanung

In diesem Kapitel stellen wir den State-of-the-Art auf dem Gebiet der Vorgehensmodelle und Projektplanung dar. Dabei definieren wir für die Arbeit wesentliche Begriffe und Grundlagen. Um den Begriff des Vorgehensmodells plastisch zu machen, geben wir in Kapitel 2.1 zunächst einen Überblick über ausgewählte Vorgehensmodelle an, angefangen von den historischen Wurzeln bis hin zu modernen Vorgehensmodellen. Die betrachteten Vorgehensmodelle werden dabei zur Klassifikation in ein allgemeines Schema eingeordnet. Wir geben zudem in Kapitel 2.2 eine Begriffsdefinition für Vorgehensmodelle in Form einer Ontologie der Elemente an, die zur Beschreibung von Vorgehensmodellen üblicherweise herangezogen werden.

Den Themenkomplex Projektmanagement und Projektplanung erläutern wir in Kapitel 2.3. Wir stellen das in der Literatur derzeit verbreitete methodische Vorgehen für die Erstellung von Projektplänen dar. Dies führt uns entsprechend den Betrachtungen über Vorgehensmodelle zu einer Begriffsdefinition, die wir in Kapitel 2.4 in Form einer Ontologie des Aufbaus von Projektplänen angeben.

Den Zusammenhang zwischen Vorgehensmodellen und Projektplänen konzipieren wir in den Kapiteln 5 und 6. In diesem Kapitel betrachten wir Vorgehensmodelle und Projektplanung isoliert, ohne dabei auf die Zusammenhänge einzugehen.

Inhalt des Kapitels	Seite
2.1 Vorgehensmodelle: Überblick und Einordnung	17
2.2 Definition Vorgehensmodell	39
2.3 Projektmanagement und Projektplanung	42
2.4 Definition Projektplan	56
2.5 Zusammenfassung	58

2.1 Vorgehensmodelle: Überblick und Einordnung

Die Definition des Begriffs Vorgehensmodell als Abstraktion einer idealen Vorgehensweise bedingt eine große Anzahl möglicher Ausprägungen konkreter Vorgehensmodelle. Tatsächlich existiert heute eine Vielzahl unterschiedlicher Vorgehensmodelle. Neben Normen, Standards und Konventionen, die meist breite Gültigkeit für sich beanspruchen, verfügt jedes größere Unternehmen über ein oder mehrere eigene hausinterne Vorgehensmodelle.

Mit dem Einsatz eines Vorgehensmodells werden im Allgemeinen die folgenden Ziele verfolgt:

- Ein Vorgehensmodell soll durch die Definition einheitlicher Begriffe und Ergebnisse zur Verbesserung der Kommunikation der Projektbeteiligten beitragen.
- Ein standardisiertes Vorgehen soll die Vollständigkeit und Qualität der zu liefernden Ergebnisse gewährleisten.
- Ein Vorgehensmodell zielt in mannigfaltiger Weise auf eine Eindämmung der Entwicklungskosten beziehungsweise der Kosten über den gesamten Lebenszyklus eines Systems ab.

In diesem Kapitel versuchen wir die Facetten des schillernden Begriffs Vorgehensmodell genauer darzustellen. Wir geben, ohne Anspruch auf Vollständigkeit, einen Überblick über gängige Vorgehensmodelle, angefangen von den historischen Wurzeln des „Wasserfallmodells“ bis hin zu neueren Entwicklungen wie dem V-Modell XT, und versuchen eine Einordnung unterschiedlicher Arten von Vorgehensmodellen in ein allgemeines Schema zur Klassifikation.

Abschnitte in Kapitel 2.1		Seite
2.1.1	Merkmale zur Klassifikation von Vorgehensmodellen	18
2.1.2	Projektdurchführungsstrategie	19
2.1.3	Anpassbarkeit	24
2.1.4	Inhaltliches Spektrum	30
2.1.5	Abstraktionsgrad	33
2.1.6	Anwendung und Werkzeugunterstützung	36

2.1.1 Merkmale zur Klassifikation von Vorgehensmodellen

Eine exakte Klassifikation von Vorgehensmodellen erfordert einen umfangreichen Katalog von Merkmalen. Abbildung 2–1 zeigt fünf mögliche Kategorien, die einen Überblick über unterschiedliche Arten von Vorgehensmodellen vermitteln sollen und damit dem Zweck dieses Kapitels genügen.

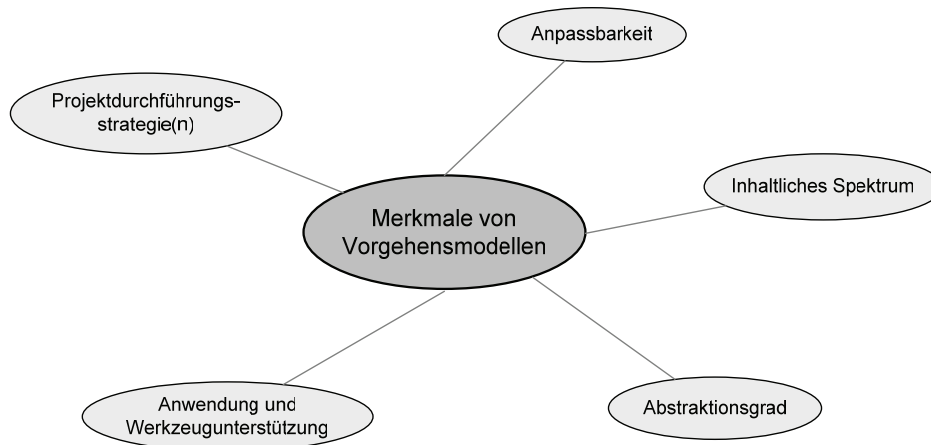


Abbildung 2–1: Merkmale zur Klassifikation von Vorgehensmodellen

Das Wasserfallmodell bildet den historischen Ausgangspunkt der Entwicklung von Vorgehensmodellen für die Softwareentwicklung. Im Wasserfallmodell steht das Merkmal einer Projektdurchführungsstrategie im Vordergrund. Während das Wasserfallmodell keine breite Abdeckung des möglichen inhaltlichen Spektrums aufweist, geht der Trend moderner Vorgehensmodelle in diese Richtung. Der mögliche Abstraktionsgrad von Vorgehensmodellen erstreckt sich über die grobgranularen Vorgaben des Wasserfallmodells bis hin zum Projektplan. Im Zusammenhang mit dem Spiralmodell betrachten wir das Merkmal der Anpassbarkeit von Vorgehensmodellen. Die Anwendung von Vorgehensmodellen schließlich reicht von

einer Anwendung im Rahmen von Ausbildung und Schulung bis hin zur Werkzeugunterstützung, die die Automatisierung von Abläufen in einem Projekt umfasst.

Die dargestellten Merkmale werden in den folgenden Kapiteln anhand von Beispielen erläutert.

2.1.2 Projektdurchführungsstrategie

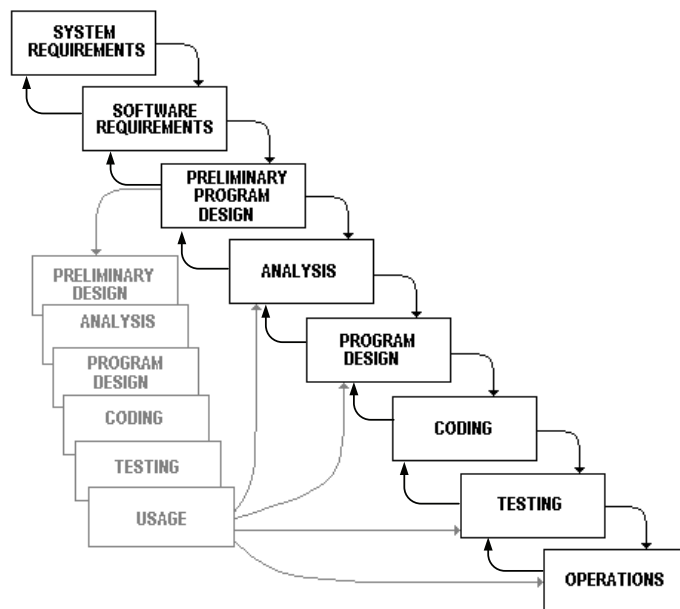
In diesem Kapitel geben wir einen Überblick über die Entwicklung von den Wurzeln, die sich im Wasserfallmodell manifestieren, über die evolutionäre und inkrementelle Entwicklung bis hin zum Modell der agilen Entwicklung. Diese Modelle bezeichnen wir als Projektdurchführungsstrategien.

Unter einer **Projektdurchführungsstrategie** verstehen wir analog zur Definition in [VM-XT05] ein Vorgehensmodell beziehungsweise den Teil eines Vorgehensmodells, das den idealen Ablauf eines Projektes auf der Granularität von Phasen oder Entscheidungspunkten beschreibt.

Das Wasserfallmodell

Die Historie von Vorgehensmodellen reicht bis zu dem wohl ältesten von Herbert Benington beschriebenen 9-Phasen-Modell aus dem Jahr 1956 zurück (vgl. [Bening56]). Dieses Vorgehensmodell wurde in den 50er Jahren in einem großen Entwicklungsprojekt für ein Luftverteidigungssystem angewendet und bildete die Basis für das später publizierte und aufgrund seiner grafischen Darstellung als Wasserfallmodell bezeichnete Vorgehensmodell.

(a) Darstellung aus [Royce70]



(b) Vereinfachte Darstellung (ohne verschränktes Prototyping)

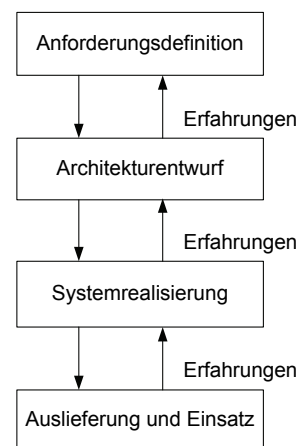


Abbildung 2–2: Das Wasserfallmodell nach [Royce70]

Charakteristisch für das 9-Phasen-Modell ist die Vorgabe, dass jede Phase von der Analyse der Anforderungen bis zur Auslieferung in der vorgeschriebenen Reihenfolge in voller Breite durchzuführen ist. Das 9-Phasen-Modell ist ein reines Top-Down-Vorgehen, das heißt vor Eintritt in eine neue Phase muss die vorangehende Phase abgeschlossen werden. Am Ende einer Phase steht jeweils ein fertig gestelltes Dokument. Das 9-Phasen-Modell wird daher wie das Wasserfallmodell als „dokument-getrieben“ bezeichnet.

Während das Vorgehensmodell von Benington einen rein sequentiellen Ablauf von Phasen vorgab, präsentierte Winston Royce 1970 ein erweitertes Phasenmodell, das einerseits Rücksprünge in vorangehende Phasen erlaubte und zum anderen das Konzept des Prototyping integrierte, wie in Abbildung 2–2 (a) dargestellt. Das Wasserfallmodell ist also entgegen vielfacher Behauptung kein rein sequentielles Modell.

Royce erkannte bereits das Kosten-Problem, das durch Rücksprünge in nicht unmittelbar vorangehende Phasen bei großen Projekten verursacht würde. Rücksprünge beispielsweise von der Test- in die Designphase aufgrund spät erkannter Mängel im Design würden eine Wiederbelebung der Codierphase erforderlich machen. Dieses Risiko versuchte Royce durch Angabe einer Reihe von Maßnahmen zu minimieren.

Eine Maßnahme bestand nach Royce im Prinzip des „Do it twice“, das in Abbildung 2–2 (a) durch graue Kästen dargestellt ist. Diese zeigen einen Durchlauf durch die Phasen, der verschränkt zu dem eigentlichen Phasen-Ablauf ist und Feedback-Schleifen enthält. Dabei handelt es sich um eine frühe Variante des Prototyping. Die Systemversion, die an den Kunden ausgeliefert wird, ist dabei in Bezug auf kritische Teile die zweite Version, die bereits gewonnene Erkenntnisse bei der Umsetzung der ersten Version beinhaltet. Weitere Maßnahmen waren die Erstellung detaillierter und umfangreicher Designspezifikationen, die frühzeitige Planung und Steuerung der Testphase, sowie die kontinuierliche Einbeziehung des Kunden auch bezüglich Details der Entwicklung.

Ein häufiger Einwand gegen das Wasserfallmodell ist, dass es Abweichungen vom streng sequentiellen Ablauf und damit Prototyping verbietet. Dies liegt allerdings nicht am Modell, sondern an der falschen Interpretation, dass nicht eine Zeile Code geschrieben werden darf, bevor der Entwurf abgeschlossen ist. Die Publikation [Royce70] beschreibt das Gegenteil. In [Bening56] wird Prototyping nicht explizit beschrieben. Allerdings wurde in dem Projekt, das das 9-Phasen-Modell zur Anwendung brachte, Prototyping frühzeitig und intensiv eingesetzt (vgl. [Boehm83]).

In diesen frühen Vorgehensmodellen sind also bereits viele noch heute gültige Vorgehensweisen beschrieben, die später als Merkmale moderner Vorgehensmodelle erneut erfunden wurden.

Die Anwendung des Wasserfallmodells im Projekt bringt Vor- und Nachteile mit sich. Das Wasserfallmodell gibt einen klaren, aus Sicht des Projektmanagements beziehungsweise des Auftraggebers einfach zu verstehenden und gut kontrollierbaren Ablauf vor. Am Wasserfallmodell kritisch gesehen werden kann, dass ein einmal festgelegter Entwicklungsablauf starr eingehalten wird, ohne dass beispielsweise Projekt- und Entwicklungsrisiken berücksichtigt werden.

Dokumentation ist zweifellos wichtig:

„The first rule of managing software development is ruthless enforcement of documentation requirements. ... In order to produce 5 million dollars of software I would estimate a 1500 page specification is about right in order to achieve comparable control.“
[Royce70]

Es besteht aber die Gefahr, dass Dokumentation für wichtiger erachtet wird als Kundenzufriedenheit.

Abbildung 2–2 (b) zeigt eine schematische Darstellung des Wasserfallmodells, die den Vergleich mit den im Folgenden dargestellten Modellen erleichtern soll.

Evolutionäres und Inkrementelles Modell

Das Ziel des Wasserfallmodells ist es, in den frühen Phasen die Anforderungen des Auftraggebers vollständig zu ermitteln, wobei unklare Anforderungen unter Umständen noch durch Prototyping geklärt werden können. Anschließend wird das System als Ganzes entwickelt und dem Auftraggeber als Ganzes zur Verfügung gestellt.

Allerdings ist die vollständige Formulierung der Anforderungen oftmals schwierig, da sich manche Wünsche an das System erst beim Einsatz des Systems ergeben. Dies gilt für die Entwicklung von Individual-Software und insbesondere auch bei der Entwicklung eines Systems für einen anonymen Markt.

Aus diesen Überlegungen entstand die Idee der evolutionären Entwicklung, die sich unter anderem im Spiralmodell von Barry Boehm manifestiert (vgl. [Boehm86] und Kapitel 2.1.3). Dabei wird das System stufenweise in mehreren Iterationen entwickelt, gesteuert durch die Erfahrungen, die der Auftraggeber beziehungsweise die Nutzer mit dem System machen. Ausgehend von einem Kernsystem, das einen Teil der Anforderungen abdeckt, werden auf Grund der im Einsatz gewonnenen Erfahrungen sukzessive neue Versionen des Systems realisiert und ausgeliefert, wie in Abbildung 2–3 (a) dargestellt. Evolutionäre Entwicklung wird zum Teil auch als evolutionäres Prototyping beziehungsweise als iteratives Modell bezeichnet.

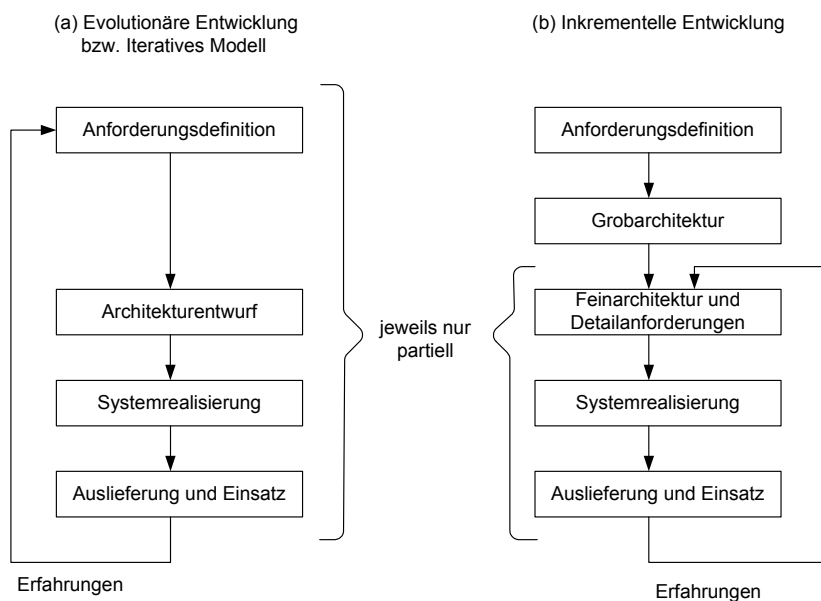


Abbildung 2–3: Evolutionäre Entwicklung versus inkrementelle Entwicklung

Als Vorteil ist zu vermerken, dass die Entwicklung eines Systems im Vergleich zum Wasserfallmodell in kleinere Einheiten überschaubarer Größe aufgeteilt wird. Hierdurch ist es möglich, den Projektablauf an die im Projektverlauf gewonnenen Erfahrungen anzupassen beziehungsweise immer wieder neu zu definieren. Allerdings bringt die evolutionäre Entwicklung auch Nachteile mit sich. Aus Sicht des Managements geht bei der evolutionären Entwicklung die langfristige Planbarkeit eines Projektes verloren, da es bei nicht festgelegter Anzahl von Iterationen auch nicht möglich ist das Projektende abzusehen. Aus technischer Sicht besteht die Gefahr, dass aufgrund neu hinzugekommener Anforderungen die Architektur des Systems überarbeitet werden muss.

Dies führte zur Idee der inkrementellen Entwicklung. Der Literatur ist keine präzise Unterscheidung zwischen inkrementeller und evolutionärer Entwicklung zu entnehmen. Inkremen-

telle Entwicklung wurde bereits 1975 von Basili und Turner vorgeschlagen (vgl. [BT75]) und hat sich im V-Modell 97 als präferierte Entwicklungsstrategie niedergeschlagen (vgl. [Dröschel00]). Die inkrementelle Entwicklung stellt eine Kombination aus dem Wasserfallmodell und der evolutionären Entwicklung dar. Dabei wird versucht, den Vorteil der langfristigen Planbarkeit des Vorgehens mit dem Vorteil der Berücksichtigung des Feedbacks aus dem Einsatz des Systems zu vereinen.

Bei der inkrementellen Entwicklung ist das Ziel, die Anforderungen zu Beginn möglichst vollständig zu erfassen und eine Grobarchitektur, die alle Anforderungen umfasst, zu modellieren. Es wird also der Gesamthorizont des Systems abgesteckt und auf dieser Basis eine Planung der einzelnen Inkremente erstellt. Diese werden wie bei der evolutionären Entwicklung sukzessive realisiert und ausgeliefert, wie in Abbildung 2–3 (b) dargestellt. Beim Einsatz gewonnene Erfahrungen können in die Umsetzung des Folge-Inkrementes einfließen. Hinsichtlich einer über den Projektverlauf stabilen Architektur bietet das inkrementelle Vorgehen günstigere Voraussetzungen als die evolutionäre Entwicklung, da die Gesamtarchitektur des Systems zumindest in Form einer Grobarchitektur festgelegt wird.

Die Idee der inkrementellen beziehungsweise evolutionären Entwicklung findet sich in ähnlicher Form auch in modernen Vorgehensmodellen wie Extreme Programming (vgl. [Beck99] und [XP03]) oder den so genannten Agilen Methoden (vgl. [Taa04]) wieder. Während in den frühen Modellen Iterationszyklen von mehreren Monaten angedacht waren, werden in diesen Ansätzen zum Teil extrem kurze Zyklen angestrebt.

Agile Entwicklung

Die agile Entwicklung weist aufgrund der Vorgabe vieler Methoden für die Zusammenarbeit im Entwicklungsteam den Charakter eines Vorgehensmodells auf (vgl. [Taa04]). In diesem Kapitel betrachten wir allerdings einzig den Aspekt der agilen Entwicklung als Projektdurchführungsstrategie.

Es handelt sich dabei um eine Strategie, die minimale Anforderungen an die Projektdurchführung hinsichtlich eines im Voraus geplanten Vorgehens stellt. Dies beruht auf der Erkenntnis, dass vorausschauende Planung über weitere Zeiträume nicht immer möglich ist.

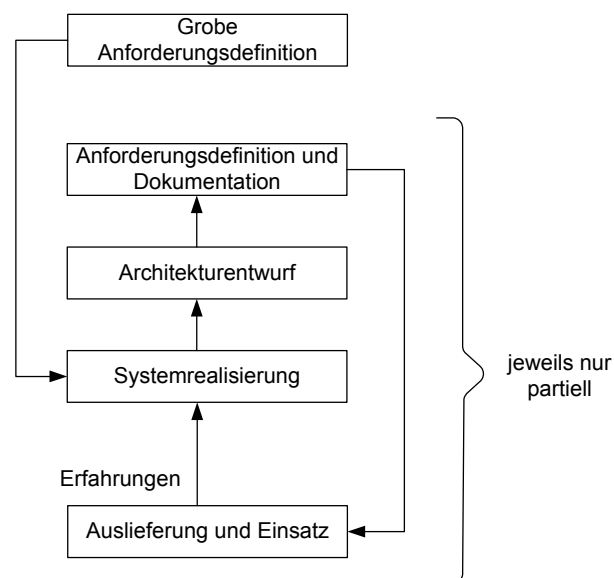


Abbildung 2–4: Agile Entwicklung

Wie in Abbildung 2–4 dargestellt, erfolgt die Anforderungsdefinition zu Beginn des Projektes auf grober Granularität. Bei neuartigen Systemen mit hohen Realisierungsrisiken ist eine vollständige Anforderungsdefinition zu Projektbeginn schwierig beziehungsweise nicht machbar. Im Rahmen der agilen Systementwicklung ist durch den unmittelbaren Einstieg in die Realisierung des Systems sichergestellt, dass keine Spezifikation für Teile erstellt wird, die sich später als nicht realisierbar erweisen. Die agile Entwicklung durchläuft also die Phasen des klassischen top-down-orientierten Wasserfallmodells in umgekehrter Reihenfolge.

Voraussetzung für eine erfolgreiche Anwendung dieser Strategie ist die intensive Einbeziehung des Kunden in die Anforderungsdefinition und das Testen im Einsatz. Durch sehr kurze Iterationszyklen steht ein Teilsystem frühzeitig zur Verfügung. Erfahrungen aus dem Einsatz können frühzeitig in die Entwicklung einfließen.

Nebenläufiges Modell

Einen anderen Ansatz stellt das nebenläufige Modell dar. In der Kette vom Wasserfallmodell zum evolutionären Modell stellt es den logischen nächsten Schritt dar.

Das nebenläufige Modell stammt ursprünglich aus der Fertigungsindustrie und wird auch als „Concurrent Engineering“ oder „Simultaneous Engineering“ bezeichnet (vgl. [Spect91]). Früher gab es oft eine strikte Trennung zwischen der Entwicklung und Fertigung eines Systems, so dass es erst nach Abschluss der Entwicklung eines Prototyps eine Überprüfung hinsichtlich Fertigungstauglichkeit gab. Dies verzögerte oftmals die termingerechte Fertigstellung eines Systems.

Concurrent Engineering verfolgt das Ziel, alle betroffenen Abteilungen von der Entwicklung über die Fertigung bis hin zum Vertrieb möglichst frühzeitig einzubeziehen und außerdem möglichst viele Schritte auf dem Weg von der Produktidee zum Markt zu parallelisieren.

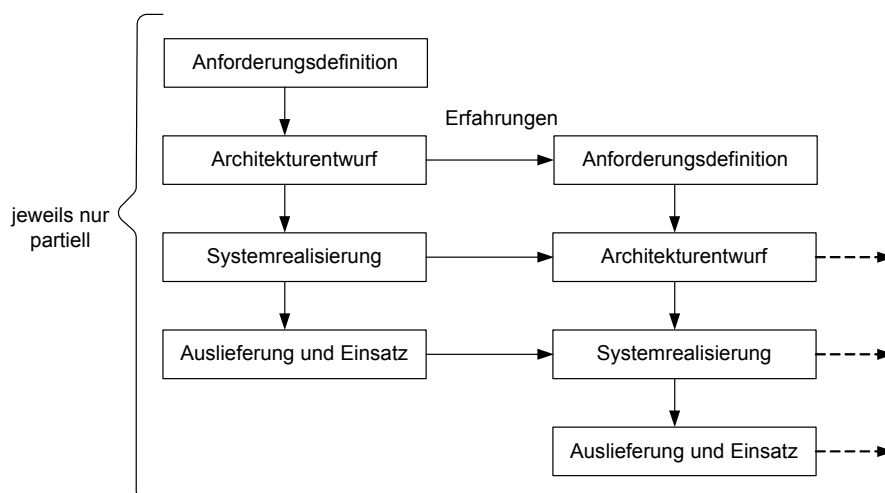


Abbildung 2–5: Nebenläufige Entwicklung nach [Balzert98] (vereinfacht)

Im Kontext reiner Softwareentwicklung, bei der der Bereich Fertigung ausgeblendet ist, entspricht der Ablauf des Concurrent Engineering der in Abbildung 2–5 dargestellten Skizze. Dabei werden die Phasen des Wasserfallmodells jeweils um eine Phase versetzt nebenläufig ausgeführt.

Mit der nebenläufigen Entwicklung ist in jedem Fall ein sehr hoher Aufwand verbunden, um in der Planung Engpässe zu antizipieren. Das hier skizzierte Modell wird in der Literatur im Kontext der Entwicklung von Software bislang nicht beschrieben. Eine Anwendung des

nebenläufigen Modells bei der Entwicklung erscheint auch nicht unbedingt Erfolg versprechend.

2.1.3 Anpassbarkeit

In diesem Kapitel beschreiben wir den Aspekt der Anpassbarkeit von Vorgehensmodellen. Dabei gehen wir auf das Spiralmodell, den Prozessmusteransatz und das V-Modell XT ein.

Unter **Anpassbarkeit** verstehen wir die einem Vorgehensmodell inhärente Fähigkeit, ohne zu großen Aufwand an den Kontext eines Projektes anpassbar zu sein.

Das Wasserfallmodell strukturiert ein Projekt in Phasen. Es propagiert bei wörtlicher Interpretation eine Planung in Form von Phasen, die zu Projektbeginn erstellt wird und sich bis zum Ende eines Projektes erstreckt. Dies birgt den Nachteil, dass lang andauernde Projekte nicht angemessen auf ein sich veränderndes Projektumfeld reagieren können. Die aktive Beobachtung des Projektumfelds tritt in den Hintergrund.

Das Spiralmodell

Um diesen Mangel zu beheben, schlug Barry Boehm 1986 das so genannte Spiralmodell für Softwareentwicklung und -pflege vor (vgl. [Boehm86]). Charakteristisch für dieses Modell ist, dass es nicht wie das Wasserfallmodell „dokument-getrieben“ oder „prototyp-getrieben“ ist, sondern die wiederholte Analyse der Risiken als zentrales Element in den Vordergrund stellt. Im Vergleich zum Wasserfallmodell enthält das Modell also einen so genannten „look ahead“-Schritt (vgl. [Boehm86]).

Das Spiralmodell sieht, wie in Abbildung 2–6 dargestellt, vier Schritte vor, die im Projekt iterativ zu durchlaufen sind. In einem ersten Schritt werden die Ziele bezüglich einem Teil des zu erstellenden Systems identifiziert und alternative Möglichkeiten bezüglich der Realisierung des Systems abgewägt – beispielsweise unterschiedliche Architektur-Entwürfe, Wiederverwendung oder der Einsatz von Fertigprodukten.

Im zweiten Schritt erfolgt eine Abwägung der erarbeiteten Alternativen unter Berücksichtigung der Ziele und Rahmenbedingungen. Werden Risiken identifiziert, so müssen Strategien beziehungsweise Maßnahmen eingeplant werden – beispielsweise Prototyping.

Im dritten Schritt wird eine Projektdurchführungsstrategie als weiteres Vorgehen festgelegt. Dabei kann es sich unter anderem um eine Anwendung des Wasserfallmodells, des inkrementellen Modells, des evolutionären Modells oder auch um eine Kombination dieser Modelle handeln. Wie viel Zeit in Aufgabenbereiche wie Projektmanagement, Konfigurationsmanagement oder Qualitätssicherung investiert wird, ist getrieben vom Risikomanagement. Auch der Architekturentwurf wird durch die Risikoanalyse beeinflusst – nur mit hohem Risiko behaftete Teile eines Systems werden detailliert entworfen und durch Prototyping evaluiert.

Im vierten Schritt erfolgt schließlich eine Bewertung der vorangegangenen Schritte, sowie die Planung der nächsten Schritte einschließlich der benötigten Ressourcen für das Projekt. Ein erneuter Durchlauf der beschriebenen Schritte startet auf einer entsprechend detaillierteren Ebene unter Berücksichtigung der bereits erarbeiteten Ergebnisse.

Das Spiralmodell enthält andere Vorgehensmodelle als Spezialfälle und kann somit als eine Art Meta-Vorgehensmodell verstanden werden. Dabei formuliert Boehm bereits Kriterien für die Auswahl eines oder die Kombination mehrerer dieser Modelle abhängig von bestimmten Projektmerkmalen (vgl. [Boehm86]). Das Wasserfallmodell wird beispielsweise bei Vorliegen stabiler Anforderungen, das evolutionäre Modell bei unstabilen Anforderungen, aber nicht zu engen Vorgaben hinsichtlich Budget und Zeit als geeignet eingestuft.

Die grundlegende Idee der Prozessmuster ist die Beschreibung eines Vorgehensmodells in Form modularer Einheiten entsprechend einem festgelegten Beschreibungsschema. Die Beschreibung eines Prozessmusters besteht aus einem initialen Kontext, der ein allgemeines, wiederkehrendes Problem in Entwicklungsprojekten beschreibt, einem Vorschlag einer bewährten und allgemeinen Lösung, sowie einem Ergebniskontext als Ergebnis der Anwendung des Prozessmusters. Prozessmuster geben Hilfestellung zu einzelnen Arbeitsschritten, die sich in der Praxis bewährt haben. Sie sind pragmatische Anleitungen für die Abwicklung einzelner Aktivitäten in einem Projekt.

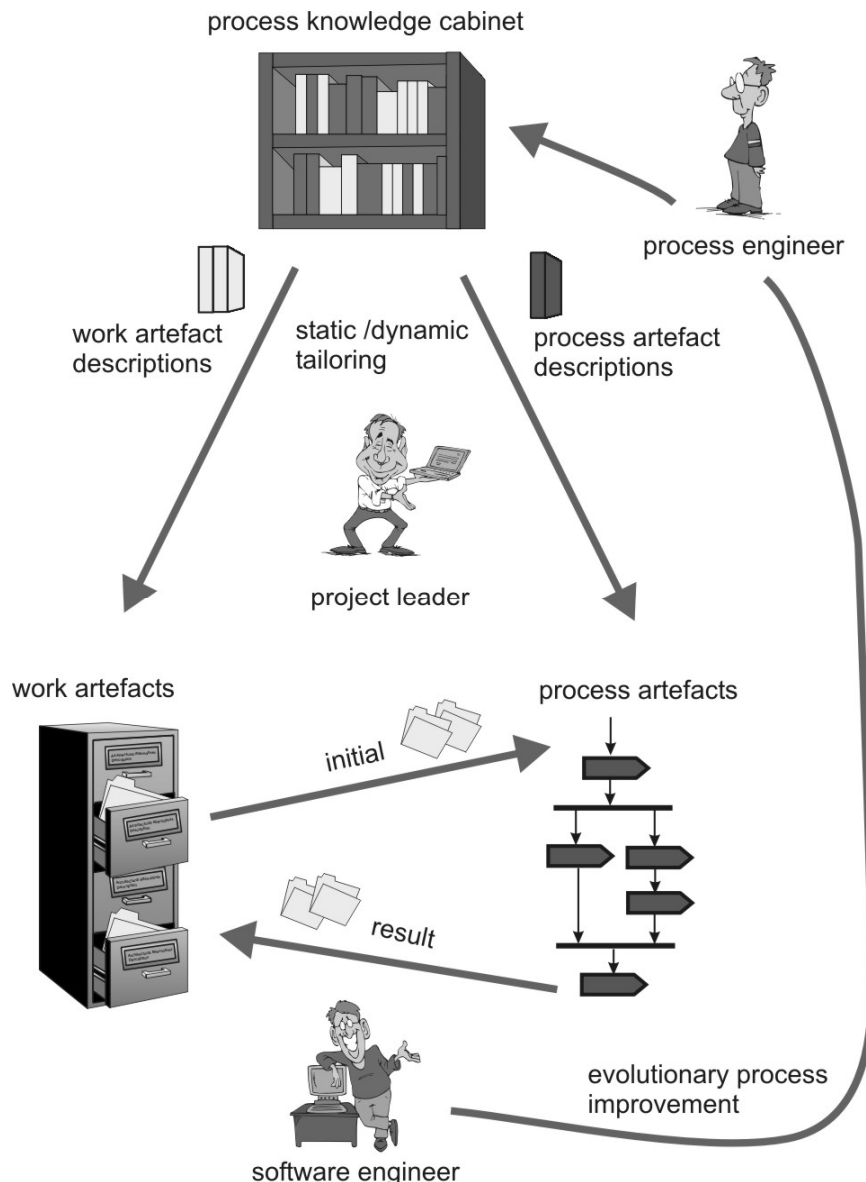


Abbildung 2–7: Lebendiger Entwicklungsprozess basierend auf Prozessmustern aus [GMP+03]

Der Aspekt Anpassbarkeit weist, wie in Abbildung 2–7 dargestellt, zwei Facetten auf. Einerseits ist das Vorgehensmodell, das aus Prozessmustern als modularen Einheiten besteht, Gegenstand eines organisationsweiten Verbesserungsprozesses. Das Vorgehensmodell wird also durch die Person des Prozessingenieurs laufend unter Berücksichtigung des

Feedbacks aus Projekten weiterentwickelt. Andererseits kann der in einem Projekt gelebte Prozess laufend den aktuellen Bedürfnissen angepasst werden.

Dies wird durch eine ergebnis-orientierte Arbeitsweise ermöglicht. Dabei werden zu Beginn des Projektes die zu erstellenden Ergebnisse („work artefacts“) definiert. Eine Vorauswahl, welche Prozessmuster in einem Projekt zur Anwendung kommen sollen, wird bewusst nicht getroffen. Das Zurechtschneiden des Vorgehensmodells („Tailoring“) bezüglich der anzuwendenden Prozessmuster beziehungsweise durchzuführenden Aktivitäten wird nicht zu Beginn des Projektes durchgeführt. Vielmehr soll dem Projektleiter und dem Projektteam die Möglichkeit gegeben werden, abhängig von der konkreten Projektsituation bestimmte Prozessmuster auszuwählen.

Beispielsweise könnte ein Prozessmuster beschreiben, wie man ausgehend von einem Designmodell konstruktiv Testfälle für die Testspezifikation erstellen kann. Ob und wann dieses Prozessmuster ausgeführt wird, ist aber nicht festgelegt, sondern wird vom Projektleiter zusammen mit dem Projektteam bei Bedarf entschieden.

Von einmal gewählten Prozessmustern kann jederzeit abgewichen werden. Basierend auf dem Zustand der definierten Ergebnisse können je nach Bedarf im Projekt andere Prozessmuster als ursprünglich geplant zur Fertigstellung der Ergebnisse verwendet werden.

Das V-Modell XT

Während das Spiralmodell und der Prozessmusteransatz nur ein Rahmenwerk zur Verfügung stellen, enthält das V-Modell XT eine inhaltliche Ausgestaltung eines Anpassungsmechanismus (vgl. [VM-XT05]).

Das V-Modell XT beinhaltet einen zweistufigen Mechanismus zur projektspezifischen Anpassung. Dieser hat einerseits Regelungscharakter, da das V-Modell Vorgabe der öffentlichen Hand für die Durchführung von Entwicklungsprojekten sein kann. Zum anderen hat das Tailoring die einfache Handhabbarkeit durch den Anwender zum Ziel.

Beim so genannten **Tailoring** werden aus der Menge der Vorgehensbausteine die für das Projekt relevanten Teile ausgewählt.

Dies geschieht anhand einer Liste vordefinierter Projektmerkmale, die als Kriterien für die Auswahl von Vorgehensbausteinen dienen. Im Rahmen der Projektplanung erfolgt dann die Auswahl der Produkte, die im Projekt zu erstellen sind.

Im Rahmen des Tailorings werden im V-Modell XT entsprechend den unterstützten Projekttypen Entscheidungspunkte ausgewählt, wie in Abbildung 2–8 dargestellt. Auf Basis dieser gewählten Entscheidungspunkte gibt eine Projektdurchführungsstrategie, die wir hier nicht abbilden, einen groben Rahmen für den Ablauf des Projektes vor. Die Projektdurchführungsstrategie ist dabei ebenfalls im Rahmen des Tailorings auszuwählen.

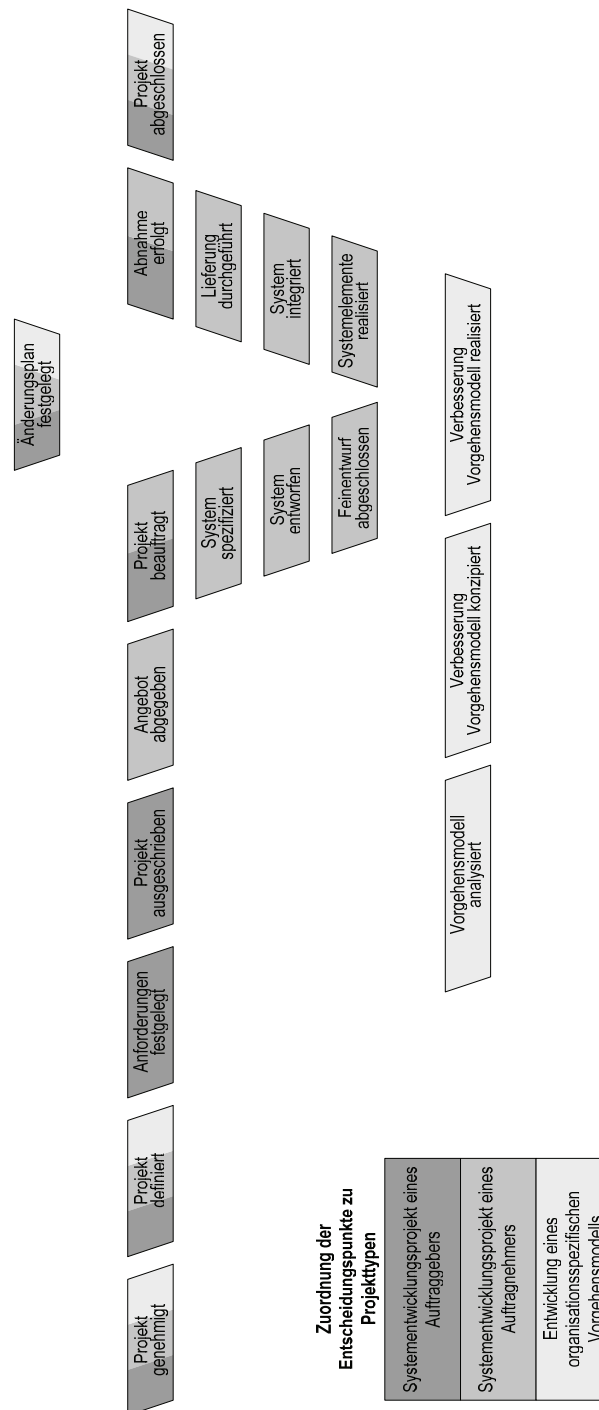


Abbildung 2–8: Tailoring im V-Modell XT – Entscheidungspunkte aus [VM-XT05]

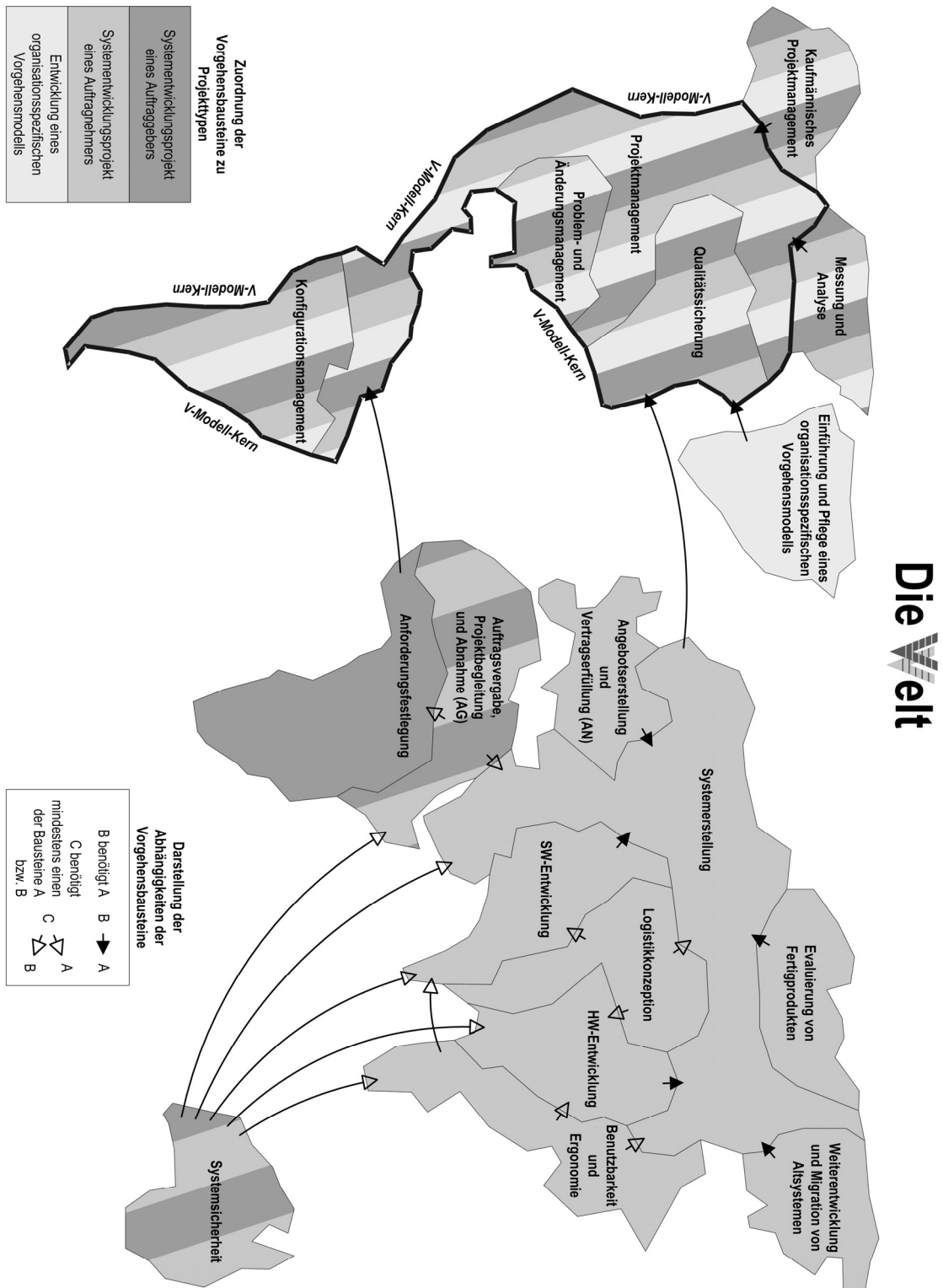


Abbildung 2–9: Tailoring im V-Modell XT – Vorgehensbausteine aus [VM-XT05]

Vorgehensbausteine legen im V-Modell XT unter anderem die Inhalte der gewählten Entscheidungspunkte fest. Je nach Auswahl bestimmter Vorgehensbausteine sind im Rahmen der Entscheidungspunkte unterschiedliche Ergebnisse vorzulegen. In Abbildung 2–9 sind die Vorgehensbausteine des V-Modell XT ersichtlich, die im Projekt angewendet werden können oder nicht. Vorgehensbausteine dienen als modulare Einheiten für die Beschreibung von Aktivitäten, Ergebnissen und Rollen. Die in Abbildung 2–9 dargestellten Abhängigkeiten schränken die gültigen Auswahlmöglichkeiten für Vorgehensbausteine im Rahmen eines Projektes ein.

Die Bandbreite der Vorgehensbausteine reicht von der Beschreibung klassischer Managementaufgaben wie Projekt- und Konfigurationsmanagement, sowie spezifischer Managementaufgaben des Auftraggeber- und des Auftragnehmerprojektes, über die Entwicklung von Software oder Hardware und die Migration von Altsystemen bis hin zur Einführung und Pflege organisationsspezifischer Vorgehensmodelle. Das inhaltliche Spektrum als Merkmal zur Klassifikation von Vorgehensmodellen betrachten wir auch im folgenden Kapitel.

2.1.4 Inhaltliches Spektrum

In diesem Kapitel betrachten wir das inhaltliche Spektrum als Merkmal zur Klassifikation von Vorgehensmodellen. Das Wasserfallmodell weist ein sehr schmales inhaltliches Spektrum auf, nämlich allein die Entwicklung von Software. Das Spiralmodell hat bereits den Anspruch, auch für die Pflege und Wartung von Software anwendbar zu sein.

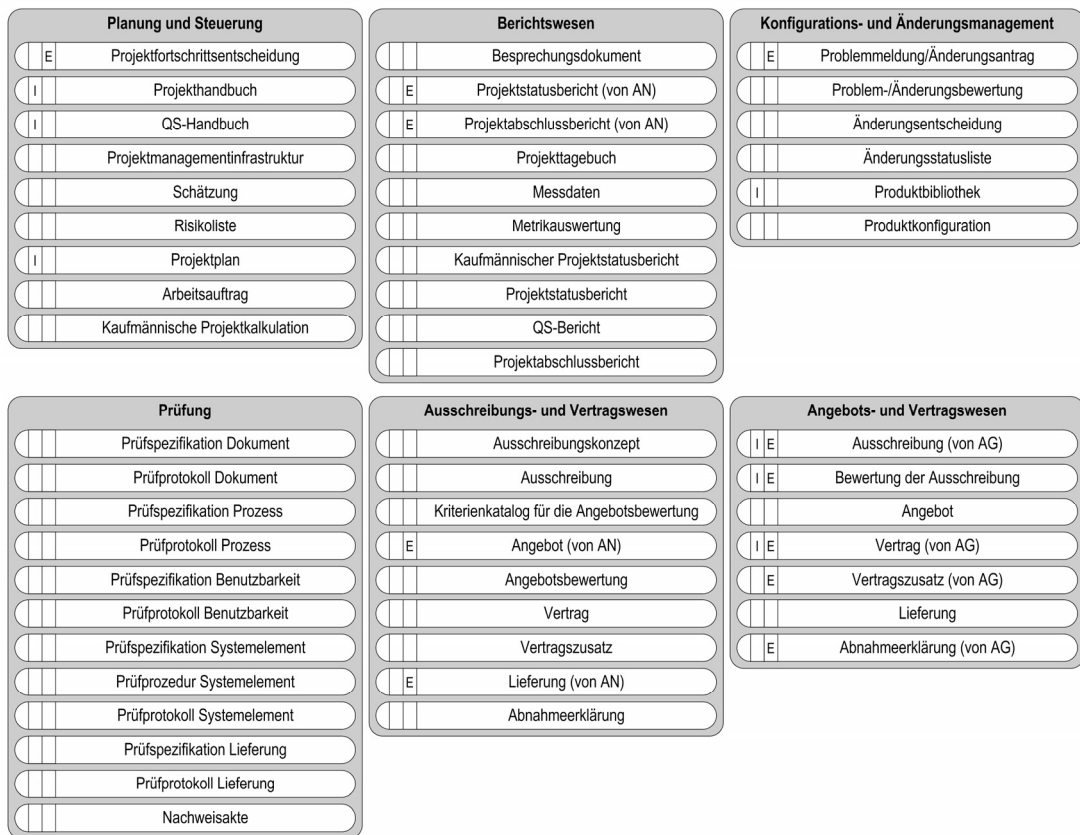
Dem möglichen inhaltlichen Spektrum eines Vorgehensmodells sind theoretisch keine Grenzen gesetzt. Die neuere Entwicklung scheint in Richtung einer immer breiteren Anwendbarkeit von Vorgehensmodellen für die unterschiedlichsten Bereiche zu gehen. In diesem Kapitel geben wir einen Überblick über das inhaltliche Spektrum von Vorgehensmodellen am Beispiel des V-Modells XT (vgl. [VM-XT05]).

Abbildung 2–10 zeigt einen Überblick über die im V-Modell XT definierten Produktgruppen. Produktgruppen sind disjunkte Mengen von Produkten. Produkte sind Beschreibungen der zu erstellenden Ergebnisse in einem Projekt (vgl. dazu Kapitel 3.3). Da Aktivitäten im V-Modell XT analog zu den Produkten strukturiert sind, gibt die Abbildung auch einen Überblick über die in einem V-Modell Projekt durchzuführenden Aktivitäten.

Die in Abbildung 2–10 dargestellten Produktgruppen untergliedern sich in

- das Projektmanagement im weiteren Sinne – hierunter fallen neben Planung, Steuerung und Berichtswesen auch das Konfigurationsmanagement, die Qualitätssicherung sowie das Ausschreibungs- und Vertragswesen –,
- die Entwicklungsaufgaben – angefangen von der Anforderungsanalyse über die Spezifikation und den Entwurf bis hin zur Realisierung und Dokumentation des Systems – und
- die Aufgaben der Organisation – diese sind im V-Modell XT nur für Prozessverbesserungsprojekte beschrieben.

Erweitertes Projektmanagement



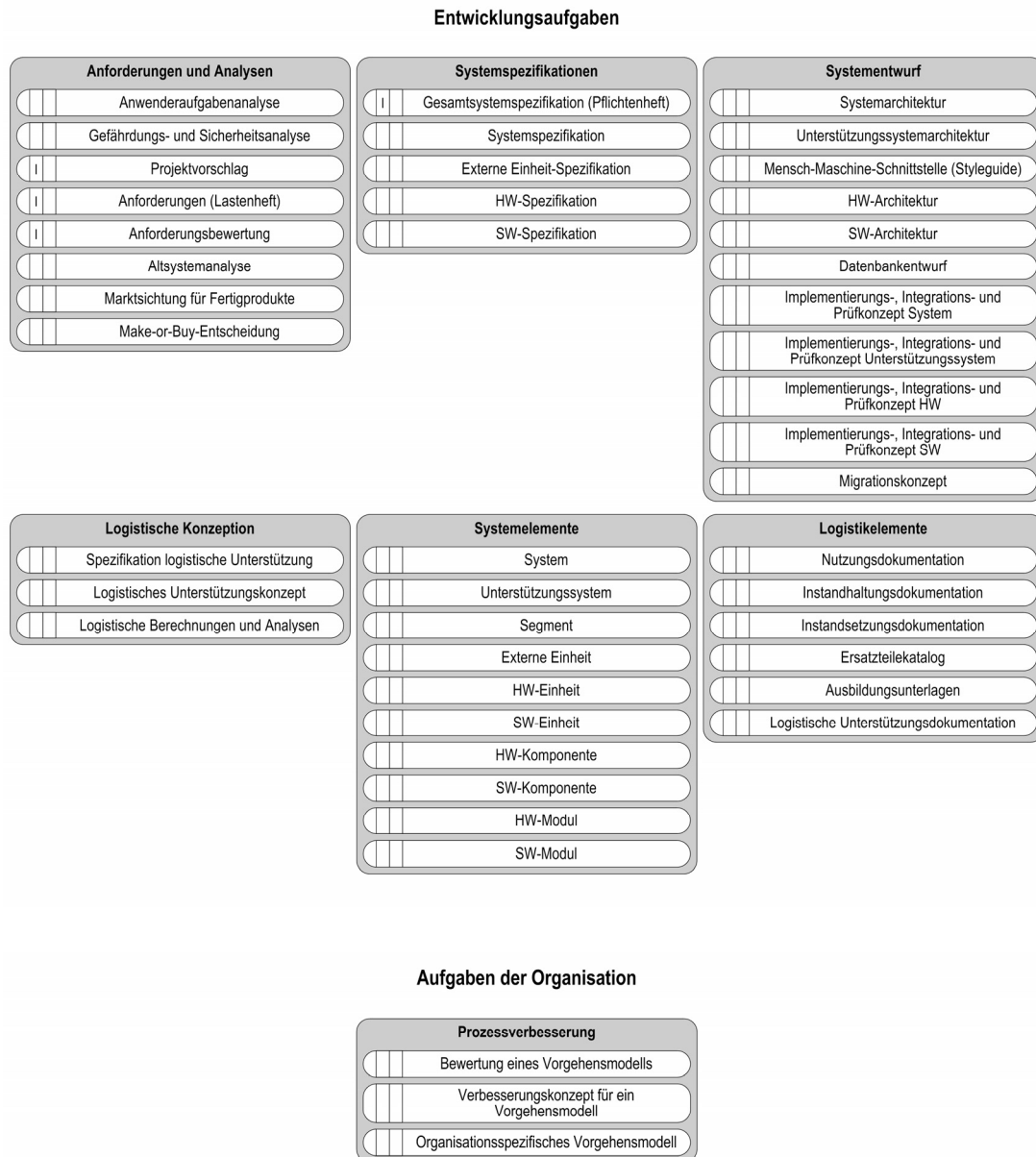


Abbildung 2–10: Inhaltliches Spektrum im V-Modell XT – Produktgruppen aus [VM-XT05]

Einen schematischen Überblick über die Dimensionen des inhaltlichen Spektrums von Vorgehensmodellen gibt Abbildung 2–11. Vorgehensmodelle lassen sich entsprechend der Aspekte Entwicklungsgegenstand, Systemlebenszyklusabschnitt und Aufgabenbereich klassifizieren. Das V-Modell XT unterscheidet als sehr abstraktes Vorgehensmodell lediglich die Aspekte Hardware und Software im Bereich der Entwicklungsaufgaben im Lebenszyklusabschnitt Entwicklung. Im Allgemeinen wäre beispielsweise auch eine Differenzierung in unterschiedliche Vorgehensweisen im Bereich Projektmanagement für Hardware- und Softwareentwicklungsprojekte, oder eine Differenzierung der Entwicklungsaufgaben für Neuentwicklungsprojekte und Pflege- und Wartungsprojekte vorstellbar.

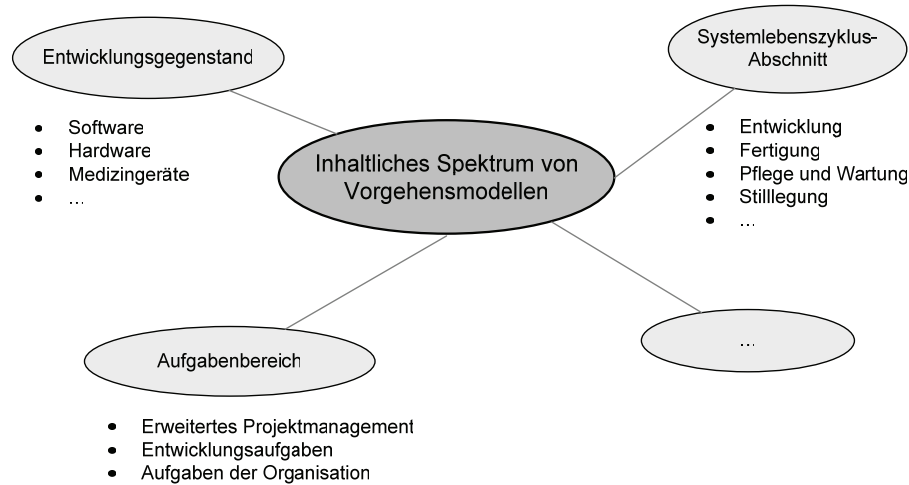


Abbildung 2–11: Inhaltliches Spektrum von Vorgehensmodellen

Viele Vorgehensmodelle richten sich, spezifischer als das V-Modell XT oder vergleichbare Modelle wie HERMES (vgl. [HERMES03]) auf genau einen spezifischen Entwicklungsgegenstand aus, wie beispielsweise die Entwicklung von Medizingeräte-Software (vgl. [AAMI03]).

Andere Vorgehensmodelle beinhalten anders als das V-Modell XT einen großen Teil Entwicklungsmethodik. Der Rational Unified Process (vgl. [KK03]) fokussiert beispielsweise die Objektorientierung als Entwicklungsparadigma. Andere Vorgehensmodelle stellen die Anwendung von Notationen wie die Unified Modeling Language zur Spezifikation des zu entwickelnden Systems in den Vordergrund (vgl. [Breu01]). Im Software-Engineering sind eine Reihe von Entwicklungsmethoden entstanden, wie „Structured Analysis / Structured Design“ (SA/SD) (vgl. [DeMarco78] und [YC79]), Catalysis (vgl. [DW98]), der Unified Software Development Process (vgl. [JBR99]) oder Objectory (vgl. [Jacob92]), um nur einige zu nennen.

2.1.5 Abstraktionsgrad

Als weiteres Merkmal zur Klassifikation von Vorgehensmodellen betrachten wir den Abstraktionsgrad. In [Chroust92] wird der Abstraktionsgrad auch als Niveau eines Vorgehensmodells bezeichnet. Dabei wird unterschieden zwischen so genannten universellen Vorgehensmodellen, die „prinzipielle Schritte“ beziehungsweise eine globale Ablaufstruktur beschreiben, weltlichen Vorgehensmodellen, die eine Abfolge von ausführbaren Schritten detailliert beschreiben, und atomaren Vorgehensmodellen, die eine Automation des Softwareentwicklungs-Prozesses ermöglichen und damit am detailliertesten sind.

Wir stellen die Bandbreite von abstrakten Modellen, die lediglich beschreiben, „Was“ zu tun ist, hin zu konkreten Modellen, die eine Antwort auf das „Wie“ geben, in Abbildung 2–12 dar. Am oberen Ende sind Projektdurchführungsstrategien wie das Wasserfallmodell zu finden, am unteren Ende projektspezifisch angepasste Vorgehensmodelle beziehungsweise Projektpläne. Projektpläne haben ebenfalls Modellcharakter, da sie als Abstraktionen des tatsächlichen Ablaufs im Projekt gelten können.

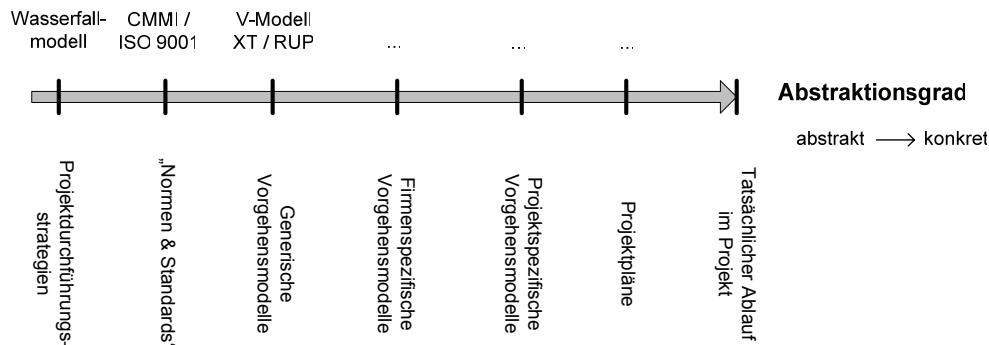


Abbildung 2–12: Abstraktionsgrad von Vorgehensmodellen

Projektdurchführungsstrategien wie das Wasserfallmodell regeln lediglich den Ablauf in Form von Phasen, ohne deren Ausgestaltung detailliert zu beschreiben. **Normen und Standards** wie ISO-9001, ISO-15288 oder CMMI formulieren Anforderungen an die im Projekt anzuwendenden Vorgehensmodelle (vgl. [Thaller00], [ISO15288], [Humph90], [CMMI]). Dabei werden keine Abläufe in Form eines Aktivitätsflusses oder Produktinhalte beschrieben, sondern Anforderungen, denen die Abläufe in einem Projekt beziehungsweise die Ergebnisse genügen müssen.

Generische Vorgehensmodelle wie das V-Modell XT dagegen geben konkrete Abläufe in Form von Projektdurchführungsstrategien, Aufgaben in Form von Aktivitätsbeschreibungen sowie Produktinhalte vor (vgl. [VM-XT05]). Als generisch werden diese Modelle aufgrund ihrer breiten Anwendbarkeit und der für eine Anwendung erforderlichen projektspezifischen Anpassung bezeichnet.

Allerdings beschreibt das V-Modell XT gemäß seinem Anspruch, methoden-neutral und damit breit einsetzbar zu sein, keine eigenen Methoden für die Entwicklung, sondern kann im Rahmen seiner Anwendung mit unterschiedlichen bestehenden Methoden ausgestaltet werden. Die Beschreibung der Produkte und Aktivitäten ist bewusst abstrakt gehalten. Die Bandbreite des möglichen Entwicklungsgegenstandes reicht vom betrieblichen Informationssystem bis hin zum militärischen Kettenfahrzeug.

Der Übergang zwischen dem „Was“ und dem methodischen „Wie“ ist bei generischen Vorgehensmodellen fließend. Der Rational Unified Process (RUP) (vgl. [KK03]) kann ebenfalls als generisches Vorgehensmodell bezeichnet werden. Der RUP ist allerdings als weniger abstrakt einzustufen als das V-Modell. Die Bandbreite des möglichen Einsatzes beschränkt sich im Falle des RUP auf den Bereich Softwareentwicklung, als Methodik wird die objektorientierte Entwicklung mit der UML beschrieben.

Der Abstraktionsgrad kann bereits anhand der verwendeten Beschreibungsmitteln abgelesen werden, wie in Abbildung 2–13 dargestellt. In einer Aktivität wird beschrieben, „Was“ zu tun ist, also letztlich das Ergebnis der Aktivität. Eine Konkretisierung stellt die Einbettung in eine mögliche Ablaufreihenfolge für die Ausführung von Aktivitäten dar. Wird diese feingranular, beispielsweise auf der Ebene von Teilaktivitäten beschrieben, ist der Übergang zur Methodik und dem „Wie“ nicht groß.

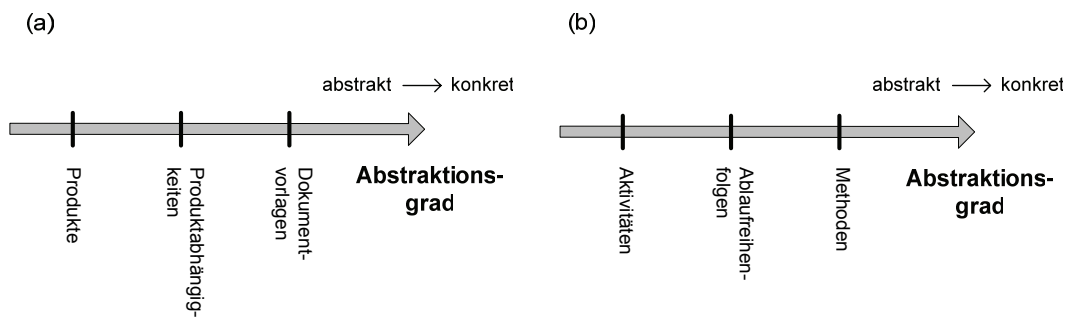


Abbildung 2–13: Abstraktionsgrad in Bezug auf die Beschreibungsmittel von Vorgehensmodellen

Die Beschreibung von Produkten reicht von einer reinen Beschreibung des Sinns und Zwecks über eine mehr oder weniger detaillierte Beschreibung von Produktabhängigkeiten bis hin zur Vorgabe von Dokumentvorlagen. Dokumentvorlagen geben eine Struktur der Inhalte vor und weisen damit bereits methodischen Charakter auf.

Der Abstraktionsgrad eines Vorgehensmodells lässt sich allerdings nicht an den Beschreibungsmitteln allein ablesen. Vorgehensmodelle weisen trotz konsequenter Anwendung festgelegter Beschreibungsmittel oft große Unterschiede hinsichtlich des Abstraktionsgrades ihrer Inhalte auf. So ist eine Aktivität „Steuerungsmaßnahmen ergreifen“ deutlich abstrakter als eine Aktivität „Projektstatusbericht erstellen“, da bei ersterer die Aufgabenbeschreibung nicht vollständig gefasst werden kann, bei letzterer dagegen eine Dokumentvorlage angegeben werden kann (vgl. [VM-XT05]).

Firmenspezifische Vorgehensmodelle – so genannte Hausstandards – sind auf die jeweiligen Geschäftsfelder einer Firma zugeschnitten. Firmenspezifische Modelle können entweder durch Anpassung eines generischen Vorgehensmodells oder aber von Grund auf neu entwickelt werden. Generische Vorgehensmodelle wie das V-Modell 97 werden dabei häufig angepasst, da ihr unmittelbarer Nutzen für ein Projekt aufgrund ihrer hohen Abstraktion als beschränkt einzustufen ist.

Beispielsweise werden laut [Plögert02] auf Basis des V-Modell 97 folgende Arten von typischen firmenspezifischen Anpassungen vorgenommen:

- Auswahl der Teile des V-Modell 97, die für Projekte der Firma relevant sein können in einem so genannten „Vor-Tailoring“
- Konkretisierung von Kriterien für das projektspezifische Tailoring
- Konkretisierung der Elemente zur Strukturierung der zu entwickelnden Systeme, also Konkretisierung der Elemente der so genannten Erzeugnisstruktur des V-Modell 97
- Inhaltliche Anpassung von Aktivitätsbeschreibungen und Dokumentvorlagen
- Detailliertere Hinweise zur Anwendung von Methoden und Werkzeugen im Projekt
- Erweiterung des Rollenmodells um organisationsspezifische Rollen
- Ergänzung firmenspezifischer Aktivitäten, beispielsweise für die Durchführung von Vorstudien in der Phase Projekt-Vorlauf

Die mögliche Bandbreite der Beschreibungsmittel eines firmenspezifischen Vorgehensmodells unterscheidet sich nicht von der eines generischen Vorgehensmodells. Firmenspezifische Modelle unterscheiden sich von generischen Modellen lediglich im Abstraktionsgrad der Inhalte. Der Übergang vom generischen Vorgehensmodell über das firmenspezifische Vorgehensmodell zum projektspezifischen Vorgehensmodell lässt sich im Allgemeinen über

beitern. Rollenbeschreibungen können beispielsweise in Unternehmen als Stellenbeschreibungen und als Anforderungen an die Qualifikation neuer Mitarbeiter herangezogen werden.

Die überwiegende Mehrheit der Vorgehensmodelle, die in die heutige Projekt-Praxis Einzug gefunden haben, dient den Projektbeteiligten als anregende Lektüre und als Hilfestellung. Vorgehensmodelle dienen dem Projektleiter als Ideengeber für die Erstellung einer initialen Projektplanung. Der Gebrauch der Terminologie eines Vorgehensmodells ist zudem oft geeignet, einen professionellen Eindruck zu vermitteln. Vorgehensmodelle dienen also nicht zuletzt der Präsentation eines nach außen.

In der Praxis durchdringt ein Vorgehensmodell bei dieser Art der Anwendung das Projekt gerade so intensiv, wie es den Projektbeteiligten günstig erscheint. Dies bedeutet, dass Vorgehensmodelle meist nur insoweit angewendet werden, als ihre Anwendung keinen Mehraufwand bedeutet und die Zeit nicht drängt. Bietet das jeweilige Vorgehensmodell Dokumentvorlagen, werden diese oft als Grundlage der Projektdokumente verwendet und im Projektverlauf je nach Bedarf angepasst. Die Anwendung eines Vorgehensmodells im Ganzen stellt die Ausnahme dar, wie sich beispielsweise in Workshops mit Anwendern des V-Modells 97 gezeigt hat (vgl. [DGR03]).

Vorgehensmodelle wie das V-Modell, die der Durchführung von Entwicklungsprojekten der öffentlichen Hand dienen, haben über den Anspruch einer anregenden Lektüre hinaus regelnden Charakter. Der Wunsch einer Regelung des Projektablaufs durch die Vorgabe eines Vorgehensmodells ist auch bei nicht-öffentlichen Auftraggebern und dem hausinternen Management meist gleichermaßen vorhanden.

In dieser Darstellung beschränken wir uns auf die Anwendung von „dokument-, beziehungsweise „aktivitäts-getriebenen“ Vorgehensmodellen. Die Erstellung von Dokumenten zur entwicklungsbegleitenden Planung oder zur Spezifikation des Systems ist im Allgemeinen nur eine, wenn auch die für die Durchführung größerer Projekte einzig Erfolg versprechende Vorgehensweise. Während es sich beim V-Modell XT ein dokument-getriebenes Vorgehensmodell handelt, tritt die Erstellung von Dokumenten beispielsweise bei Extreme Programming (vgl. [Beck99]) in den Hintergrund.

Durch die Schaffung einer einheitlichen Begriffswelt, die Festlegung durchzuführender Aktivitäten in einer gegebenen Reihenfolge und die Festlegung von Dokumenten mit einheitlichen Inhalten und einheitlichem Aufbau wird das Ziel verfolgt, unterschiedliche Projekte vergleichbar zu machen. Diese Vergleichbarkeit, zumindest auf der Granularität von Phasen, ist die Basis für Kontrolle und Steuerung der Projekte durch den Auftraggeber beziehungsweise das hausinterne Management. Die Definition von Entscheidungspunkten im V-Modell XT dient vornehmlich diesem Zweck.

Aus der Sicht der Projektdurchführenden haben Vorgehensmodelle wie das V-Modell 97 oder das V-Modell XT also den Charakter von Checklisten zur Überprüfung der Vollständigkeit der Planung im Sinne durchzuführender Aktivitäten sowie der Überprüfung zu erstellender Dokumente hinsichtlich Aufbau und Vollständigkeit der Inhalte. Eine konsequente Anwendung eines Vorgehensmodells mit entsprechendem Regelungscharakter erfordert also die wiederholte Konsultation des Vorgehensmodells während der gesamten Projektlaufzeit.

Werkzeugunterstützung von Vorgehensmodellen

Ein Schritt in Richtung Werkzeugunterstützung zur Anwendung eines Vorgehensmodells im Projekt ist die Ableitung eines Projektplans beziehungsweise weiterer Planungsdokumente. Da dies geeignet ist, dem Projektleiter Arbeit abzunehmen, ist eine entsprechende Werkzeugunterstützung der Akzeptanz eines Vorgehensmodells sehr förderlich. Das händische Kopieren von Aktivitäten des Vorgehensmodells in den Projektplan und der Aufbau eines Abhängigkeitsgeflechts zwischen diesen Aktivitäten entfallen damit.

Der Bereich einer weitergehenden Automatisierung der Anwendung von Vorgehensmodellen ist mannigfaltig. In [Osterw87] wird die Idee der „Ausführung“ von Prozessen vergleichbar

der Ausführung eines Programms beschrieben: „Software processes are software too“. Dies führte zu einer Forschungsrichtung, die auf die Entwicklung so genannter prozessorientierter Softwareentwicklungsumgebungen abzielt (vgl. [CG98]). Durchzuführende Aktivitäten werden dabei in eine Ablaufmaschine beziehungsweise Workflow-Maschine eingebettet. In [Chroust92] wird diese Art der Anwendung von Vorgehensmodellen als Task-Management bezeichnet. Task-Management dient der Unterstützung des einzelnen Entwicklers bei der Auswahl und Reihung zu erledigender Aufgaben.

Unterschiedliche Grade der Automatisierung sind vorstellbar (vgl. [CG98]). Eine Ablaufmaschine für Prozesse kann entweder nur auf Initiative des Anwenders aktiv werden, oder aber eine aktive Führung in Form einer Liste zu erledigender Aufgaben anbieten. Dabei kann wiederum unterschieden werden in das bloße Vorschlagen zu erledigender Aufgaben sowie die verbindliche Vorgabe von Aufgaben, die in einen festgelegten Ablauf eingebettet sind und vom Benutzer abzuarbeiten sind. Darüber hinaus gibt es Ablaufmaschinen, die bestimmte Aufgaben automatisch, das heißt ohne Benutzerinteraktion, ausführen.

Einige kommerzielle Werkzeuge wie beispielsweise Maestro (vgl. [Merbeth93]) fallen in die Kategorie der prozessorientierten Entwicklungsumgebungen. Maestro bietet unter anderem eine Workflow-Unterstützung basierend auf der Verfolgung von Dokument-Zuständen an. Auch in moderneren Werkzeugen wie in-Step wurde eine Ablaufmaschine basierend auf einem Vorgehensmodell realisiert (vgl. [in-Step]). In in-Step wurde der im V-Modell 97 beschriebene Ablauf basierend auf Aktivitäten, die die Zustände von Produkten ändern, umgesetzt (vgl. Kapitel 1.4).

Im Umfeld der universitären Forschung gibt es eine Reihe von Ideen beziehungsweise Prototypen für Werkzeuge zur Anwendung von Vorgehensmodellen. In [MDBG+00] wird eine Ablaufmaschine für die Entwicklung in verteilten Teams konzipiert. Das Szenario besteht auch hier in der Ableitung eines Projektplans aus einem Vorgehensmodell und in der Erstellung von Aufgabenlisten für Team-Mitglieder. Der Schwerpunkt liegt in [MDBG+00] in der Angabe von Regeln für den Umgang mit Änderungen des Projektplans. So wird ein Mechanismus für die Benachrichtigung räumlich und zeitlich verteilt arbeitender Team-Mitglieder entworfen. Ein regel-basierter Mechanismus ermöglicht die Kennzeichnung von nachfolgend geplanten Aufgaben, die von bereits erarbeiteten Ergebnissen abhängen, als obsolet.

In [Westfe01] wird eine prototypische Werkzeugumgebung zur Unterstützung des Projektmanagements beschrieben. Diese wird aus einer formalen Spezifikation basierend auf Graphersetzungsregeln generiert. Die Bestandteile dieser Managementumgebung sind in die Bereiche Projektplanung, Workflow-Unterstützung mit einer Agenda zu bearbeitender Dokumente für die jeweiligen Bearbeiter von Aufgaben und Konfigurationsmanagement einzuordnen.

Letztlich basieren sämtliche Werkzeuge, die einen wie auch immer gearteten Ablauf beinhalten, zumindest implizit auf einem Vorgehensmodell. So lassen sich beispielsweise in eine Entwicklungsumgebung eingebettete Code-Transformationen – so genannte Refactoring-Schritte – als Umsetzung eines Vorgehensmodells verstehen.

2.2 Definition Vorgehensmodell

Nach Überblick und Klassifikation anhand unterschiedlicher Merkmale geben wir in diesem Kapitel eine Definition für den Begriff des Vorgehensmodells an. Da es wie in vielen anderen Bereichen der Informatik auch eine einheitliche Definition für den Begriff Vorgehensmodell nicht gibt, definieren wir Vorgehensmodelle entsprechend unserem Verständnis:

Ein **Vorgehensmodell** legt einen standardisierten organisatorischen Rahmen für den idealen Ablauf eines Entwicklungsprojektes in Form einer Beschreibung der zu erstellenden Produkte, der durchzuführenden Aktivitäten und der zu besetzenden Rollen fest. Ein Vorgehensmodell ist damit ein Modell beziehungsweise eine Abstraktion einer möglichen Vorgehensweise im konkreten Projekt. Ein Vorgehensmodell kann auch als „Handlungsmuster“ bezeichnet werden. Als synonym betrachten wir den ebenfalls gebräuchlichen Begriff Prozessmodell.

Ein Vorgehensmodell ist also vergleichbar mit einem Kochrezept oder Leitfaden, dessen Befolgung zu einem vorhersagbaren Ergebnis führt, während der Einsatz praktischer Erfahrung allein dies nicht immer gewährleistet.

Da obige Begriffsdefinition nicht plastisch und anschaulich ist, geben wir in Abbildung 2–15 eine allgemeine Ontologie für Vorgehensmodelle an. Die Ontologie definiert Begriffe und Zusammenhänge, die zur Beschreibung von Vorgehensmodellen herangezogen werden, also eine Begriffswelt. Die hier angegebene Ontologie dient ausschließlich der Visualisierung wesentlicher begrifflicher Zusammenhänge. Sie wird nicht dem Anspruch eines Vorgehens-Meta-Modells (vgl. dazu Kapitel 5). In Abbildung 2–15 kennzeichnen schwarze Pfeile jeweils die Leserichtung einer Assoziation.

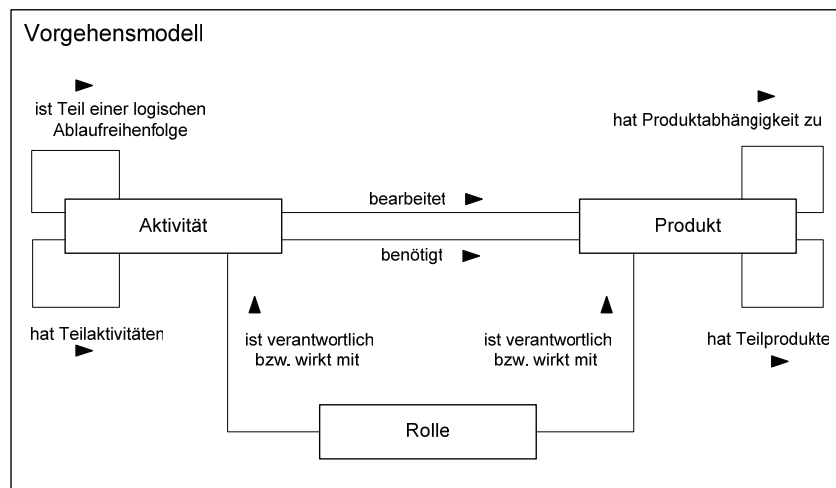


Abbildung 2–15: Allgemeiner Aufbau von Vorgehensmodellen (Ontologie)

Ein Vorgehensmodell dient der Beschreibung der im konkreten Projekt durchzuführenden **Aktivitäten**, zu erstellenden **Produkte** und mit Personen zu besetzenden **Rollen**. Genau genommen handelt es sich dabei jeweils um Beschreibungen dieser Elemente, also beispielsweise um Aktivitätsbeschreibungen. Wir definieren die Begriffe hier allerdings vereinfachend ohne das Suffix „-beschreibung“. Wir geben zunächst Definitionen der Begriffe an und gehen anschließend auf ihre Zusammenhänge ein.

Ein **Produkt** ist eine Beschreibung des Ergebnisses oder auch eines Zwischenergebnisses, das im Laufe eines Projektes entstehen soll. Ein Produkt kann beispielsweise

- eine Beschreibung des Aufbaus und Inhalts eines Softwarearchitektur-Dokuments,
- eine Beschreibung einer Softwareeinheit im Allgemeinen, oder
- eine Beschreibung einer physikalisch greifbaren Entität, beispielsweise einer Leiterplatte oder Autokarosserie, oder
- eine Beschreibung einer für die Projektdurchführung zu installierenden Software, beispielsweise eines Werkzeugs für das Konfigurationsmanagement, sein.

Eine **Rolle** ist eine Beschreibung der Aufgaben, Verantwortlichkeiten und Fähigkeiten einer Person, die zur Durchführung eines Entwicklungs-Projektes benötigt werden. Eine Rolle ist nicht unmittelbar als Beschreibung genau einer Person zu sehen, da im Projekt eine Person mehrere Rollen in sich vereinigen kann oder auch eine Rolle im Projekt von mehreren Personen wahrgenommen werden kann. Beispiele für Rollen sind

- die Rolle Projektleiter oder
- die Rolle Softwarearchitekt.

Eine **Aktivität** ist eine Beschreibung einer im Projekt zu erledigenden Aufgabe oder durchzuführenden Tätigkeit. Aktivitäten können dabei sehr grobgranular eine Beschreibung des „Was ist zu tun“ enthalten, oder aber eine methodische Anleitung „Wie etwas zu tun ist“ in Form einzelner Arbeitsschritte beinhalten. Beispiele für Aktivitäten sind

- die Erstellung eines Softwarearchitektur-Dokumentes oder
- die Durchführung von Maßnahmen zur Projektsteuerung.

In diesem Kapitel stellen wir einzig die wesentlichen Begriffe dar. Die Begriffe Werkzeug und Methode sind in Abbildung 2–15 subsumiert. Die Beschreibung von Werkzeugen kann in die Aktivitäts- oder Produktbeschreibung integriert sein. Aktivitäten beziehungsweise Teilaktivitäten können methodischen Charakter haben. Eine Unterscheidung ist also nicht erforderlich.

Wie in Abbildung 2–15 dargestellt, können Produkte und Aktivitäten beliebig in feingranuläre Beschreibungselemente wie Teilprodukte oder Teilaktivitäten hierarchisch untergliedert werden. Teilprodukte können beispielsweise Kapitel eines Dokumentes, Teilaktivitäten einzelne Arbeitsschritte sein.

Produkte können unterschiedliche Arten von **Produktabhängigkeiten** wie beispielsweise gegenseitige Konsistenzabhängigkeiten oder Erzeugungsabhängigkeiten aufweisen. Produktabhängigkeiten können sehr abstrakt oder aber detailliert und präzise formuliert sein. Abstraktes Beispiel einer Konsistenzabhängigkeit ist die Abhängigkeit „Das Produkt Softwarearchitektur-Dokument muss alle Anwenderanforderungen des gleichnamigen Produktes berücksichtigen“. Beispiel einer gerichteten Erzeugungsbeziehung sind die Vorgaben, die das Softwarearchitekturdokument hinsichtlich der im Projekt zu realisierenden Softwareeinheiten macht. Architekturdokument und Softwareeinheit stehen damit in einer Erzeugungsabhängigkeit.

Aktivitäten sind Teil einer logischen Ablaufreihenfolge. In der dargestellten Ontologie formulieren wir die Zusammenhänge, die sich in den meisten verbreiteten Vorgehensmodellen finden, wenn auch beispielsweise nicht im V-Modell XT (vgl. [VM-XT05]). Eine **logische Ablaufreihenfolge** legt einen idealen schrittweisen oder auch nebenläufigen Ablauf über einer Menge von Aktivitäten fest. Ablaufreihenfolgen in Vorgehensmodellen sind häufig als Anhaltspunkt für die Projektdurchführung zu verstehen und nicht als verbindliche Vorgabe. Beispielsweise kann eine logische Ablaufreihenfolge die Durchführung der Aktivität „Anwen-

deranforderungen festlegen“ vor der Durchführung der Aktivität „Softwarearchitektur erstellen“ festlegen.

Die mit der Durchführung von Aktivitäten verbundene Aufgabe kann in der Erstellung, der Bearbeitung oder auch der Fertigstellung von Produkten bestehen – diese Beziehungen haben wir in der Assoziation **bearbeitet** in Abbildung 2–15 subsumiert –, muss es aber nicht. So ist beispielsweise eine Aktivität „Softwarearchitektur erstellen“ unmittelbar dem Produkt „Softwarearchitektur“ zugeordnet. Andere Aktivitäten wie „Steuerungsmaßnahmen ergreifen“ können nicht unmittelbar einem Produkt zugeordnet werden. Der Aktivität „Steuerungsmaßnahmen ergreifen“ könnte zwar ein Produkt „Action-Item-Liste“ zugeordnet werden, dies würde allerdings der Aufgabe des Projektleiters nicht gerecht werden.

Bestimmte Produkte können Voraussetzung für die Durchführung von Aktivitäten sein, da zuvor erarbeitete Ergebnisse bei ihrer Durchführung berücksichtigt werden sollen. Diesen Zusammenhang drückt die Assoziation **benötigt** in Abbildung 2–15 aus. Beispielsweise wird das Produkt „Anwenderanforderungen“ zumindest in grobgranularer Form als Eingangsprodukt für die Durchführung der Aktivität „Softwarearchitektur erstellen“ benötigt. Eingangsprodukt-Beziehungen zwischen Aktivitäten und Produkten sind analog zu Ablaufreihenfolgen häufig nicht als verbindliche Vorgaben zu verstehen, sondern lediglich als Anhaltspunkt. Zum Teil sind Beschreibungsmittel für Vorgehensmodelle als redundant einzustufen. Beispielsweise können Eingangsprodukt-Beziehungen auch mit Hilfe von Produktabhängigkeiten modelliert werden.

Rollen schließlich sind gemäß ihrer Aufgaben und Fähigkeitsprofile **verantwortlich** für die Erstellung von Produkten beziehungsweise die Durchführung von Aktivitäten, **wirken** aktiv an der Produkterstellung **mit** oder sind nur beratend tätig.

2.3 Projektmanagement und Projektplanung

Das folgende Kapitel beleuchtet den Themenkomplex Projektmanagement und Projektplanung. Projektmanagement und Projektplanung stellen einen Teil des thematischen Umfelds dar, in dem wir uns mit der vorliegenden Arbeit bewegen.

Wir geben einen Überblick über das Management von Projekten anhand des Vorgehensbausteins Projektmanagement aus dem V-Modell XT (vgl. [VM-XT05]). Wir gehen dabei insbesondere auf die Projektplanung ein. Wir stellen derzeit gängigen methodischen Ansätze der Projektplanung, unter anderem die Erstellung von Produkt- und Projektstrukturplänen, sowie die Netzplantechnik dar. Abschließend zeigen wir praktische Probleme auf.

Abschnitte in Kapitel 2.3		Seite
2.3.1	Projektmanagement und Projektplanung im V-Modell XT	42
2.3.2	Projektplanung: Vorgehen und Methodik	46
2.3.3	Praktische Probleme der Projektplanung	54

2.3.1 Projektmanagement und Projektplanung im V-Modell XT

In diesem Kapitel stellen wir das Thema Projektmanagement anhand des V-Modells XT dar. Unser Ziel ist dabei analog der Zielsetzung des V-Modells XT der Aufbau einer klaren Begriffswelt.

Für die Begriffsbestimmung hilfreich ist zunächst der Begriff des Projektes.

Ein **Projekt** ist ein Vorhaben, das im Wesentlichen durch eine Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist. Ein Projekt ist definiert durch die Zielvorgabe, die zeitlichen, finanziellen und personellen Rahmenbedingungen, sowie allgemein durch die Abgrenzung gegenüber anderen Vorhaben.

[DIN69901]

Die Definition des Begriffes Projektmanagement entleihen wir dem V-Modell XT:

Das **Projektmanagement** umfasst alle Aufgaben, um die Aktivitäten des Projektteams zu planen, zu kontrollieren und zu steuern, damit das Projektziel sicher erreicht werden kann beziehungsweise Probleme frühzeitig erkannt und beseitigt werden können. [...] Das Management eines Projektes stellt einen wesentlichen Einflussfaktor auf den Projekterfolg dar. [...]

[VM-XT05], Vorgehensbaustein „Projektmanagement“

Projektmanagement im weiteren Sinne umfasst, abgegrenzt von den Entwicklungsaufgaben und den Aufgaben der Organisation, alle Managementaufgaben im Rahmen eines Projektes. Einen Überblick vermitteln die Produktgruppen aus dem Bereich Projektmanagement des V-Modells XT in Abbildung 2–16.

In der Aktivitätsgruppe Planung und Steuerung werden alle planenden und steuernden Aktivitäten zusammengefasst, die über den gesamten Projektlebenszyklus wiederholt ausgeführt werden müssen.

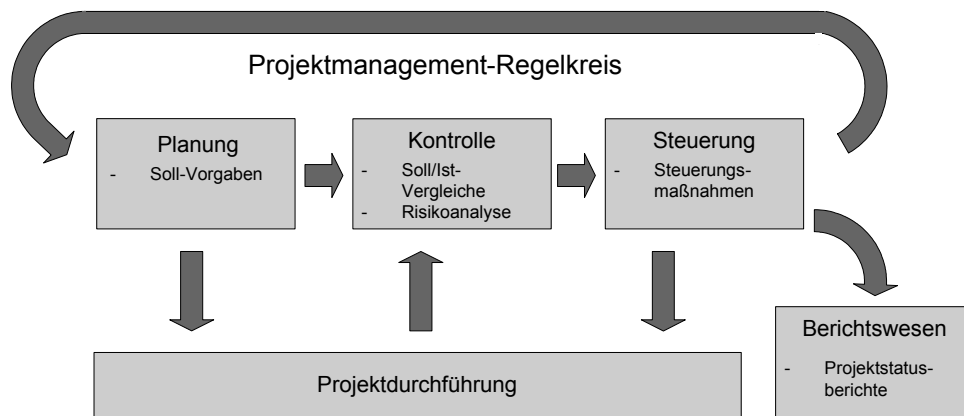


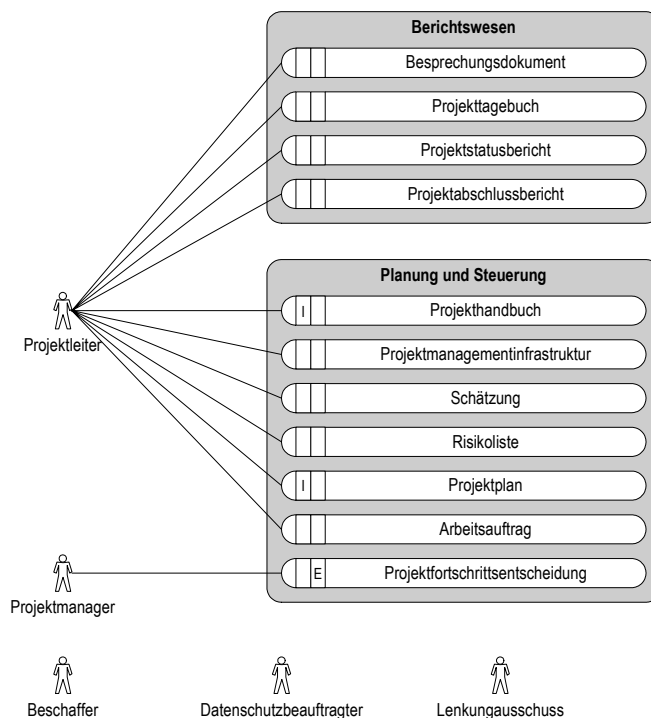
Abbildung 2–17: Projektmanagement-Regelkreisprinzip

Abbildung 2–17 verdeutlicht den iterativen Charakter des Projektmanagements in Form eines Regelkreisprinzips. Das Projekt wird entsprechend der Soll-Vorgaben der Planung durchgeführt. Im Rahmen der Kontrolle wird der Projektfortschritt regelmäßig dem geplanten Soll gegenüber gestellt. Dabei werden Risiken analysiert. Der Soll/Ist-Vergleich und die Risikoanalyse fließen dann in die Projektstatusberichte ein. Durch steuernde Maßnahmen wird Einfluss auf die Projektdurchführung genommen, und die geplanten Soll-Vorgaben werden bei Bedarf modifiziert. Ziel ist die Optimierung des magischen Dreiecks aus Zeit, Kosten und Qualität unter Berücksichtigung des konkreten Zieles und der Einhaltung der Randbedingungen des Projekts.

[VM-XT05], Aktivitätsgruppe „Planung und Steuerung“

Der Vorgehensbaustein Projektmanagement des V-Modells XT beschreibt die folgenden im Projekt zu erstellenden Ergebnisse und verantwortlichen Rollen.

[...] Das Projektmanagement beschreibt die Projektinitialisierung, die Projektplanung, die Projektdurchführung und den Projektabschluss. Zentrale Produkte sind das Projekthandbuch, das die organisatorischen Rahmenbedingungen festlegt, der Projektplan, die Risikoliste, sowie die Produkte des Berichtswesens, das der Dokumentation sowie der internen und externen Verbreitung aller Projektereignisse und -ergebnisse dient. [...]



[...]

Abbildung 2–18: Vorgehensbaustein Projektmanagement

[VM-XT05], Vorgehensbaustein „Projektmanagement“

Die Definition des Begriffes Planung entnehmen wir ebenfalls dem V-Modell XT.

Planung ist die Vorbereitung zukünftigen Handelns. Planung bedeutet, im Voraus zu entscheiden, wie ein Ziel erreicht werden soll, das heißt, was zu tun ist, wie es zu tun ist, wann es zu tun ist und wer es zu tun hat. Ziel dieser Aktivität ist die Durchführung der Planung der Produkte, der notwendigen Aktivitäten, der Ressourcen und der Termine für das Projekt. [...]

[VM-XT05], Aktivität „Projekt planen“

Eine textuelle Definition des Begriffes Projektplan findet sich ebenfalls im V-Modell XT – eine präzise Definition geben wir in Kapitel 2.4 an.

Für die gesicherte und geordnete Durchführung eines Projektes ist ein solider **Projektplan** zwingend erforderlich. Der Projektplan beschreibt die gewählte Vorgehensweise des Projektes und legt detailliert fest, was, wann und von wem zu tun ist. Der Projektplan ist damit die Basis für die Kontrolle und Steuerung des Projektes. Für den Projektplan ist der Projektleiter verantwortlich. Die Erstellung und Bearbeitung des Projektplanes erfolgt aber in Abstimmung mit allen Projektbeteiligten. [...]

[VM-XT05], Produkt „Projektplan“

2.3.2 Projektplanung: Vorgehen und Methodik

In diesem Kapitel stellen wir Methodik und Techniken der Projektplanung dar. Dabei gehen wir auf die gängige in der Literatur beschriebene Methodik der Erstellung von Projektplänen ein. Die Darstellung orientiert sich an [BMI04], [Balzert98] und [PM04]. Das V-Modell XT zitieren wir nur in Teilen, da die Beschreibung der Projektplanung aufgrund des Meta-Modells hinter dem V-Modell XT sehr speziell für dessen Kontext ist. Zum anderen werden methodische Herangehensweisen wie die Netzplantechnik im Rahmen des V-Modells nicht behandelt.

Das Kapitel zeichnet den Weg der Erstellung eines Projektplans über die Erstellung eines Produktstrukturplans, eines Projektstrukturplans, sowie die Netzplantechnik, die Ressourcenplanung und die Terminplanung nach. Zur Veranschaulichung stellen wir die Methodik anhand eines durchgängigen Beispiels dar. In der Literatur lässt sich ein entsprechendes Beispiel nicht finden.

Produktstrukturplan und Projektstrukturplan

Wie in der Literatur beschrieben, führt die Erstellung eines Projektplans zunächst über die Erstellung eines Produktstrukturplans sowie im Anschluss daran die Erstellung eines Projektstrukturplans.

[...] Bei Entwicklungsprojekten wirkt sich die Struktur des zu entwickelnden Systems bestimmend auf den Projektplan aus. Bestimmte Aktivitäten, wie beispielsweise die Realisierung einer Software-Einheit, sind für jede vorgesehene SW-Einheit eigens einzuplanen. Die hierarchische Anordnung aller Komponenten eines Systems in einem Produktstrukturplan kann ein methodisches Hilfsmittel zur Erstellung des Projektplanes sein. Im Produktstrukturplan ist die Struktur des gesamten zu entwickelnden Systems zu erfassen. [...]

(In der Produktstruktur) sind nicht nur die Komponenten des zu entwickelnden Systems zu berücksichtigen, sondern auch Produkte aus dem Bereich Management. [...]

[VM-XT05], Teilaktivität „Produkt- und Aktivitätsstruktur vollständig planen“

Produkt- und Projektstrukturplan stellen eine logische Strukturierung der anstehenden Aufgaben dar, beinhalten also noch keine Termin- oder Ressourcenplanung. Wir definieren den Projektplan wie folgt.

Unter einem **Produktstrukturplan** verstehen wir die Menge aller projektspezifisch zu erstellenden Projektergebnisse, sowohl technische Komponenten als auch Dokumente.

Ein Produktstrukturplan wird dabei meist hierarchisch dargestellt, beispielsweise entsprechend einer strukturellen „ist enthalten“-Beziehung.

Präzise Definitionen der Begriffe Produkt- und Projektstrukturplan sind in der Literatur nicht zu finden. Ein Beispiel eines Produktstrukturplans ist in Abbildung 2–19 dargestellt. Neben den Komponenten Anmeldung und Kundenmanagement werden in diesem Beispiel auch Entwicklungsdokumente sowie eine zu erstellende Testumgebung aufgeführt.

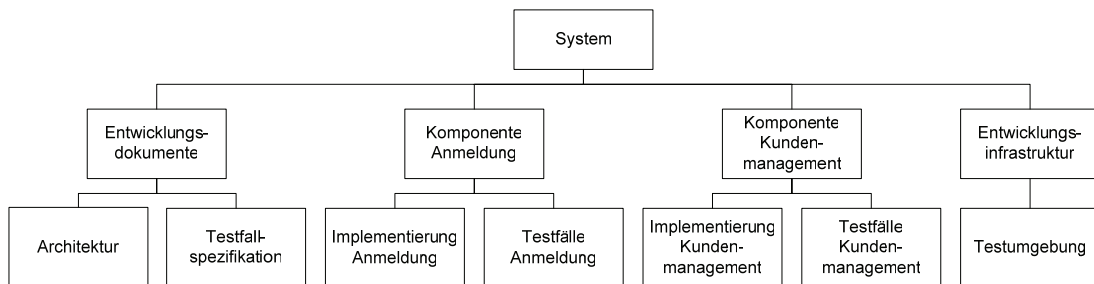


Abbildung 2–19: Produktstrukturplan: Beispiel

Auf Basis des Produktstrukturplans kann ein Projektstrukturplan erstellt werden. Diesen definieren wir wie folgt.

Im **Projektstrukturplan** werden Arbeitspakete beziehungsweise Teilaufgaben jeweils mit Bezug zu den Elementen des Produktstrukturplans erfasst. Unter einem **Arbeitspaket** verstehen wir dabei eine Gruppierung von projektspezifisch durchzuführenden **Teilaufgaben**. Teilaufgaben werden im Kontext der Netzplantechnik auch als Vorgänge bezeichnet.

Diese Definition deckt sich mit den Definitionen in [DIN69901]. Einige Anhaltspunkte für die Strukturierung von Projektstrukturplänen entnehmen wir dem V-Modell XT.

Folgende Punkte sind bei der Formulierung von Arbeitspaketen zu beachten:

- Für jedes Arbeitspaket darf es nur einen Verantwortlichen geben.
- Für jedes Arbeitspaket muss eine eindeutige Aufgabenbeschreibung formuliert werden, deren Erfüllung auch überprüft werden kann.
- Arbeitspakete sollen in sich abgeschlossene Bearbeitungs-Einheiten sein, die über klare Schnittstellen zu anderen Arbeitspaketen verfügen.
- Die geplante Bearbeitungszeit sollte relativ zur Projektlaufzeit nicht zu lange sein, da Kontrolle sonst nur schwer möglich ist.
- Das Zusammenstellen von Arbeitspaketen ist risikoabhängig. Mit hohem Risiko behaftete Aufgaben sind in kleinere Arbeitspakete zu zerlegen als Routineaufgaben.

[VM-XT05], Teilaktivität „Arbeitspakete planen“

Abbildung 2–20 zeigt ein Beispiel eines Projektstrukturplans. Arbeitspakete wurden in diesem Beispiel einerseits auf Basis der Teamstruktur gebildet. Die Arbeitspakete können im Rahmen der Ressourcenplanung unterschiedlichen Teams zugeordnet werden. Zum anderen wurden Arbeitspakete in diesem Beispiel entsprechend dem zeitlichen Ablauf des Pro-

jekt es gebildet, also für die einzelnen Fertigstellungsstufen der Elemente der Produktstruktur.

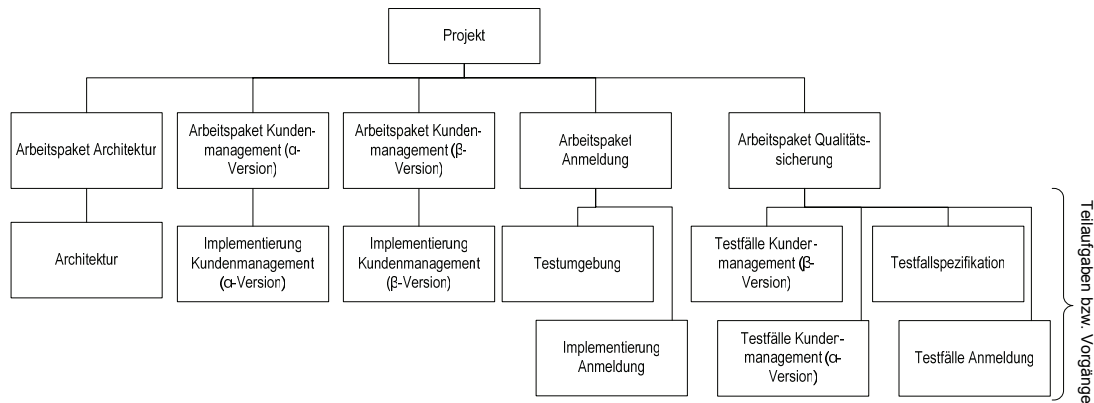


Abbildung 2–20: Projektstrukturplan: Beispiel

Die Blattebene des Projektstrukturplans in Abbildung 2–20 kann unmittelbar der Blattebene des Produktstrukturplans in Abbildung 2–19 zugeordnet werden. Präzise Aussagen zu dem Zusammenhang zwischen Produkt- und Projektstrukturplan sind der Literatur nicht zu entnehmen.

Im V-Modell XT werden für die Erstellung des Produkt- und Projektstrukturplans eigene Produktexemplare nicht explizit gefordert. Produkt- und Projektstrukturplan werden als methodische Schritte auf dem Weg zur Erstellung des eigentlichen Projektplans gesehen.

Eine Aufteilung in einen Produktstrukturplan und einen Projektstrukturplan ist im V-Modell nicht vorgesehen. Die Integrierte Planung muss im Projektverlauf stets als Ganzes überarbeitet werden, um einen konsistenten Planungsstand zu erreichen. Beispielsweise führt eine Änderung der Produktstruktur in der Regel stets zu einer Änderung der [...] (eingepannten Aktivitäten) und damit der Projektstruktur.

[VM-XT05], Produkt „Integrierte Planung“

Netzplantechnik

Die Netzplantechnik stellt ein methodisches Hilfsmittel dar, das der Überführung eines Projektstrukturplans in einen Plan mit Start- und Endterminen für die einzelnen durchzuführenden Aktivitäten dient.

Gemäß [DIN69900-1] ist der **Netzplan** eine graphische oder tabellarische Darstellung von Vorgängen und deren Abhängigkeiten. Im Rahmen der Netzplantechnik werden die einzuplanenden beziehungsweise im Projekt durchzuführenden Aktivitäten als **Vorgänge**, die Abhängigkeiten als **Vorgangfolgen** bezeichnet. Im Kontext der Projektplanung verwenden wir im Folgenden den Begriff Vorgang.

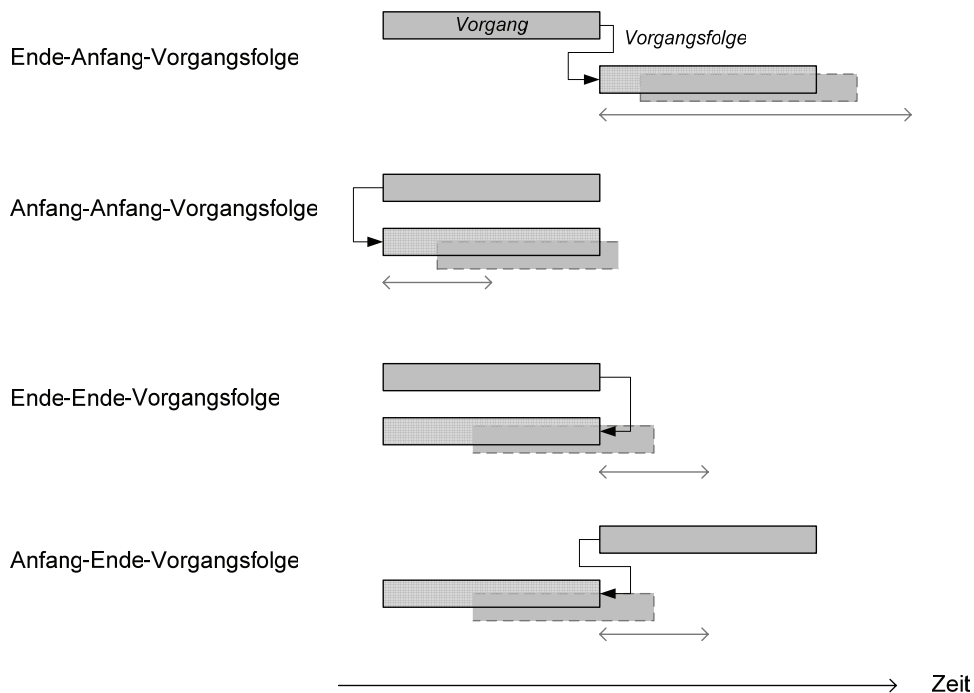


Abbildung 2–21: Vorgangsfolgen

Die am weitesten verbreitete Notation im Rahmen der Netzplantechnik ist der Vorgangsknotennetzplan. In einem Vorgangsknotennetzplan werden die Vorgänge als Knoten dargestellt und die Abhängigkeiten als Kanten. Als Abhängigkeitstypen treten die folgenden binären Beziehungen auf, wie auch in Abbildung 2–21 dargestellt:

- **Normalfolge** beziehungsweise Ende-Anfang-Vorgangsfolge: Ein Vorgang kann begonnen werden, sobald sein Vorgänger endet.
Beispiel: Für die Bearbeitung ist die Fertigstellung eines anderen Vorgangs Voraussetzung. Dabei werden erarbeitete Ergebnisse wie beispielsweise ein Architekturdokument als Input an den Folgevorgang Implementierung übergeben.
- **Anfangsfolge** beziehungsweise Anfang-Anfang-Vorgangsfolge: Ein Vorgang kann begonnen werden, sobald sein Vorgänger begonnen wurde.
Beispiel: Für die Bearbeitung einer Aufgabe ist die Zusammenarbeit zweier Bearbeiter oder Teams erforderlich. Die zeitliche Verfügbarkeit eines Bearbeiters bestimmt den Anfang der Arbeiten des anderen Bearbeiters.
- **Endfolge** beziehungsweise Ende-Ende-Vorgangsfolge: Ein Vorgang kann beendet werden, sobald sein Vorgänger beendet wurde.
Beispiel: Für die Fertigstellung eines Vorgangs müssen die Ergebnisse eines anderen Vorgangs fertig gestellt vorliegen. Mit dem Testen kann beispielsweise bereits während der Implementierung begonnen werden, abgeschlossen werden kann das Testen aber erst nach Fertigstellung der Implementierung.
- **Sprungfolge** beziehungsweise Anfang-Ende-Vorgangsfolge: Ein Vorgang kann beendet werden, sobald sein Vorgänger begonnen wurde.
Beispiel: Ein Vorgang wird an einen anderen Bearbeiter übergeben. Der ursprüngliche Bearbeiter muss so lange zur Verfügung stehen, bis die Übergabe an den neuen Bearbeiter unter Anleitung des ursprünglichen Bearbeiters erfolgt ist.

Vorgangsfolgen sind gerichtete binäre Abhängigkeiten, die bei einer terminlichen Verschiebung eines Vorgangs oder einer Änderung der Bearbeitungsdauer eines Vorgangs zu Änderungen der Termine anderer Vorgänge führen. Verschiebt sich beispielsweise das Ende eines Vorgangs, so verschiebt sich auch der Anfang seines Nachfolgers in einer Ende-Anfang-Vorgangsfolge (vgl. [MSProj03]). Während die Normalfolge einen sequentiellen Ablauf von Vorgängen vorgibt, erlauben Anfangs-, End- und Sprungfolge eine Parallelisierung der eingeplanten Vorgänge. Für Vorgangsfolgen können zudem positive oder negative **Zeitabstände** definiert werden. Zusammengehörige Vorgänge sind dann überlappt oder zeitverzögert auszuführen. Beispielsweise kann eine Normalfolge mit einem positiven Abstand von zwei Tagen annotiert werden, so dass der Folgevorgang erst zwei Tage nach Abschluss des Vorgängers begonnen werden kann.

Vorgänge werden im Rahmen der Netzplantechnik mit einer Reihe von Attributen versehen, wie beispielsweise dem geschätzten Aufwand, der geschätzten Bearbeitungsdauer, dem frühesten beziehungsweise spätesten Start- und Endtermin, sowie dem geplanten Start- und Endtermin. Die Netzplantechnik dient der Optimierung der Start- und Endtermine der einzelnen Vorgänge und damit des gesamten Projektes.

Zunächst muss der Aufwand der einzelnen Vorgänge durch eine Schätzung ermittelt werden. Die Dauer der Vorgänge kann mit Hilfe der Ressourcenplanung jeweils aus dem geschätzten Aufwand ermittelt werden, wie im folgenden Abschnitt noch zu erläutern ist.

Zwei Arten der Terminrechnung sind auf Basis der geschätzten Vorgangsdauern unter Berücksichtigung der Vorgangsfolgen durchzuführen. In der so genannten **Vorwärtsrechnung** werden die frühesten Start- und Endtermine der Vorgänge durch Verfolgung aller Pfade im Vorgangsknoten-Graphen, ausgehend vom geplanten Projektanfang durch Aufsummierung der Aktivitätsdauern, ermittelt. In der **Rückwärtsrechnung** werden die spätesten Termine bestimmt. Dabei wird vom geplanten Projektende ausgegangen. Dieses kann entweder vorgegeben oder aber mit Hilfe der Vorwärtsrechnung als frühestes Projektende bestimmt worden sein. Durch Verfolgung aller Pfade vom Projektende zeitlich gesehen rückwärts werden späteste Start- und Endtermine der einzelnen Vorgänge berechnet.

Die Differenz zwischen spätestem und frühestem Anfangstermin eines Vorgangs wird als Pufferzeit bezeichnet. Die geplante Dauer eines Vorgangs kann in der Projektdurchführung um die **Pufferzeit** überschritten werden, ohne den Endtermin des Projektes zu verzögern. Je größer die Pufferzeit ist, umso geringer ist die Wahrscheinlichkeit, dass eine Verzögerung des jeweiligen Vorgangs ursächlich für die Verzögerung des Projektendes sein wird. Besitzt ein Vorgang keine Pufferzeit, so spricht man von einem kritischen Vorgang. Kritische Vorgänge bilden einen so genannten **kritischen Pfad**.

Ein kritischer Pfad zeigt an, welche Vorgänge besonders sorgfältig überwacht werden müssen, damit ein Projekt termingerecht abgeschlossen werden kann. Wird ein kritischer Vorgang verzögert, führt dies, falls die Verzögerung nicht wieder aufgeholt werden kann, zu einer Gesamtprojektverzögerung.

Beispiel eines Netzplans

Abbildung 2–22 zeigt einen Vorgangsknotennetzplan. In diesem Beispiel werden einige der Elemente der untersten Ebene des Projektstrukturplans aus Abbildung 2–20 in ihrem Abhängigkeitsgeflecht dargestellt. Der uns bekannten Literatur sind keine anwendungsnahen Beispiele für den Einsatz der Netzplantechnik zu entnehmen.

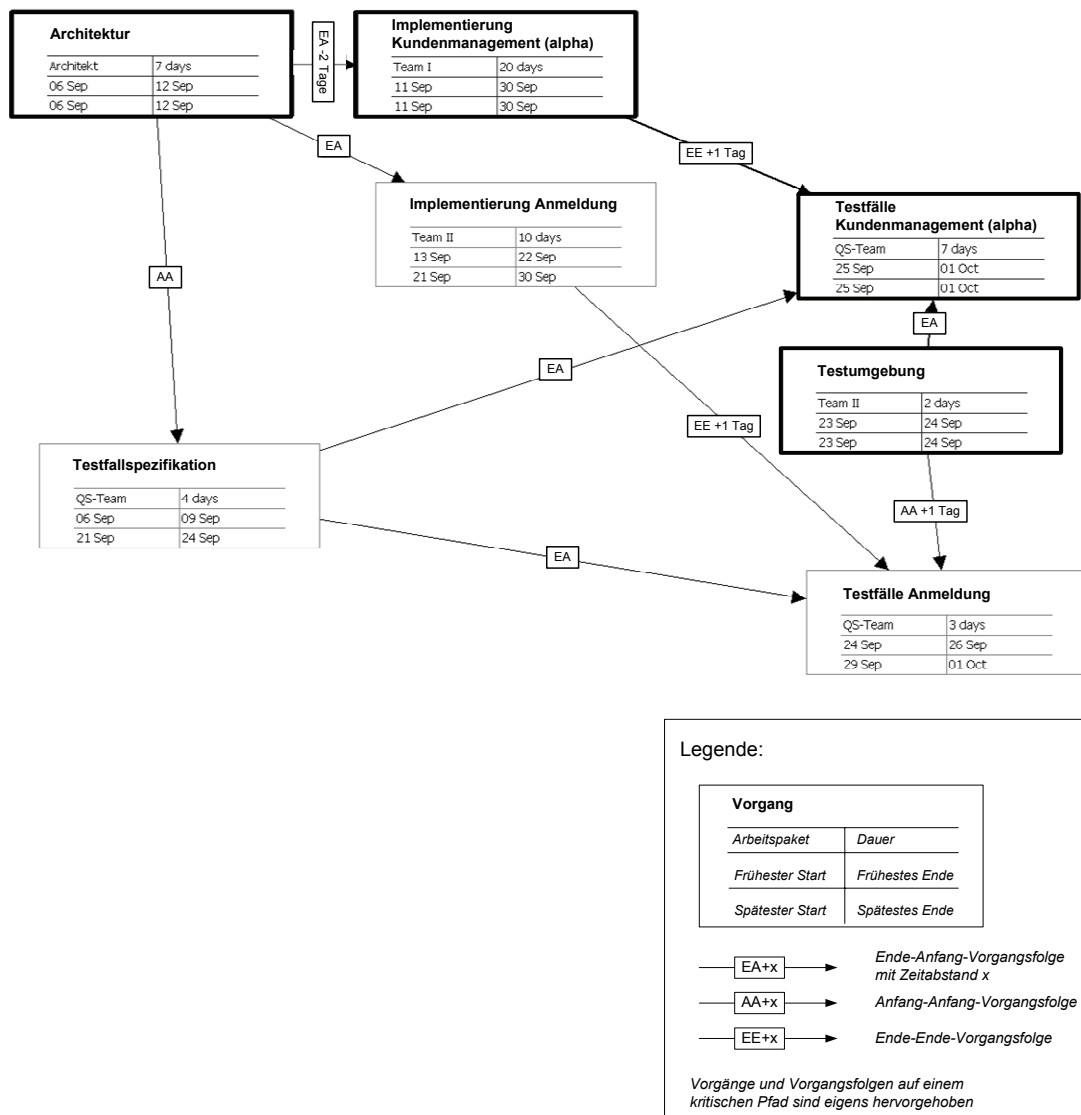


Abbildung 2–22: Netzplan mit Vorgangsfolgen und Terminrechnung: Beispiel

Im Beispiel muss die Erstellung der Architektur des Systems abgeschlossen sein, bevor mit der Implementierung der einzelnen Komponenten begonnen werden kann. Die Implementierung des Kundenmanagements muss allerdings bereits mit einer Zeitversetzung von 2 Tagen vor dem Abschluss der Erarbeitung der Architektur starten, damit in der Architektur erstes Feedback aus der Implementierung berücksichtigt werden kann.

Der Test der Komponenten kann bereits parallel zur Implementierung beginnen, darf aber nicht enden, bevor nicht die Implementierung abgeschlossen ist. Durch die Richtung der Abhängigkeit und den Abhängigkeitstyp Ende-Ende ist festgelegt, dass ein Verzug der Implementierungsvorgänge eine Verschiebung der Testvorgänge zur Folge haben wird.

Der Test der Anmeldekomponente kann einen Tag, nachdem der Aufbau der Testumgebung begonnen hat, beginnen. Die zufrieden stellende Funktionsweise der Testumgebung kann dadurch bereits während der Erstellung überprüft werden. Der Test des Kundenmanagements wird dagegen erst nach Abschluss des Aufbaus der Testumgebung gestartet. Während Ende-Anfang-Vorgangsfolgen meist eine Fertigstellung und Weitergabe eines

Projektergebnisses darstellen, werden parallelisierende Folgen wie die Anfangs-Anfangs-Folge oft für Vorgänge mit intensivem Abstimmungsbedarf der Beteiligten verwendet.

Entsprechend den Ergebnissen der Terminrechnung zur Bestimmung der frühesten und spätesten Anfangs- und Endtermine bilden die Vorgänge „Architektur“, „Implementierung Kundenmanagement“, „Testumgebung“ und „Testfälle Kundenmanagement“ einen kritischen Pfad.

Ressourcenplanung

Die rein schematische Berechnung der Termine anhand der geschätzten Aufwände geht von der Annahme der freien Verfügbarkeit von Ressourcen aus, also der freien Verfügbarkeit sowohl von Personal als auch von Material. Da dies in der Praxis nicht gegeben ist, sind Termin- und Ressourcenplanung stets verschränkt durchzuführen. Im Rahmen der Personaleinsatzplanung ist die zeitliche und räumliche Verfügbarkeit qualifizierten Personals zu berücksichtigen. Die Auslastung der Ressource Personal muss bei der Planung dahingehend optimiert werden, dass weder Überlast noch mangelnde Auslastung der Mitarbeiter die Folge ist.

Prinzipiell bietet sich als Variante zur Optimierung der Auslastung zum einen die **terminorientierte Ressourcenplanung** an. Dabei werden bei vorgegebenen Terminen die Zuordnung von Mitarbeitern zu Aktivitäten und damit die Aktivitätsdauern variiert. Zum anderen kann die Optimierung der Auslastung mit Hilfe der **aufwandsorientierten Ressourcenplanung** erfolgen. Dabei werden bei vorgegebenen maximalen Aufwänden die Termine variiert.

Der Personaleinsatzplanung kommt in der Praxis eine große Bedeutung zu, da Projektmitarbeiter den größten Einflussfaktor auf Erfolg oder eben Misserfolg eines Projektes darstellen. Personaleinsatzplanung und Materialplanung als Teile der Ressourcenplanung betrachten wir im Rahmen der Arbeit nicht weiter, da sie im Rahmen der Arbeit nicht relevant sind.

Terminplanung

Um den Netzplan in einen Terminplan überzuführen, können für die einzelnen Vorgänge Termine gewählt werden, die im Zeitabschnitt der jeweils frühesten und spätesten Termine liegen. Hierbei können so genannte **Vorgangseinschränkungen** formuliert werden. Vorgänge können dabei beispielsweise mit Einschränkungen wie „so früh wie möglich“ oder „so spät wie möglich“, beziehungsweise „Anfang nicht später als“ oder „muss anfangen am“ zusammen mit einem festen Datum annotiert werden.

Nach Festlegung der geplanten Start- und Endtermine bietet sich die Darstellung als so genanntes **Balkendiagramm** an, auch Gantt-Chart genannt. Ein Balkendiagramm ist dabei gemäß [DIN69901] lediglich eine spezielle Darstellungsform eines Netzplans. Das Balkendiagramm ordnet alle Elemente in ihrer zeitlichen Abfolge über einer Zeitachse an, so dass anders als im Netzplan auch Dauern und Zeitdifferenzen graphisch dargestellt werden.

Abbildung 2–23 zeigt einen Balkenplan, der die Vorgänge des Beispiel-Netzplans des vorangehenden Abschnitts beinhaltet. Während der Netzplan früheste und späteste Termine zeigt, sind im Balkenplan die tatsächlich geplanten Termine dargestellt. Dabei ist die Erstellung der Testfallspezifikation nicht mit dem entsprechend dem Netzplan frühesten möglichen Beginn geplant, sondern beginnt beispielsweise aufgrund der Verfügbarkeit des QS-Teams erst später, ohne dabei im Beispiel eine Verschiebung der folgenden Vorgänge nach sich zu ziehen.

Bei der eigentlichen Terminplanung ist die Abstimmung mit allen externen wie internen Beteiligten essentiell. Neben der Einbeziehung des Kunden ist die Terminkoordination beispielsweise mit den Bereichen Systemerstellung und Qualitätssicherung wesentlich.

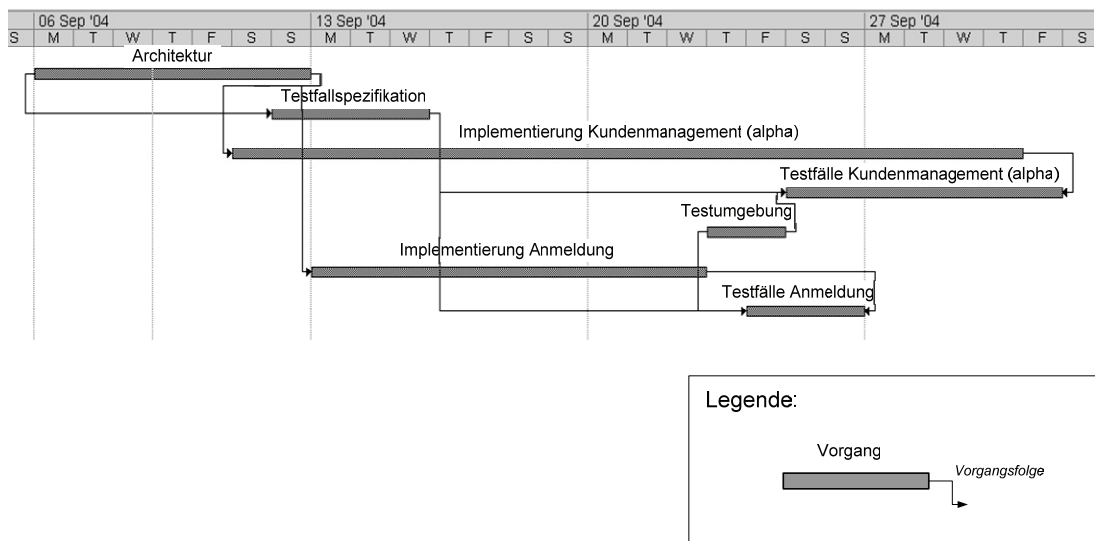


Abbildung 2–23: Balkendiagramm: Beispiel

Meilensteine und Grobplanung

Ein Projektplan muss neben dem Zweck der terminlichen Abstimmung zwischen Vorgängen unterschiedlicher Bearbeiter der Kontrolle des Projektes durch das Management dienen. Meilensteine dienen durch das gezielte und rechtzeitige Erkennen von Projektverzögerungen der Projektkontrolle. Meilensteine synchronisieren Projekt Ereignisse wie Projektanfang, Projektende oder den Abschluss wichtiger Aktivitäten. Projektergebnisse, die beispielsweise für nachfolgende Entwicklungsschritte, Unteraufträge oder Weiterentwicklungen einen definierten Ausgangspunkt bilden, bieten sich ebenfalls für die Festlegung eines Meilensteins an.

Ein **Meilenstein** ist definiert als ein geplanter Prüfzeitpunkt, zu dem eine Menge von Projektergebnissen beziehungsweise Produkten fertig gestellt und überprüfbar sein soll. Neben dem Termin der Abnahme des Meilensteins sind die entscheidungsbefugten Personen festzulegen. Meilensteine müssen stets nach objektiven Kriterien überprüfbar sein und sind daher an die Fertigstellung von überprüfbaren Ergebnissen geknüpft. Zudem ist eine annähernde Gleichverteilung über die Projektlaufzeit anzustreben, sowie zeitliche Abstände aufeinander folgender Meilensteine, die sich in der Größenordnung von nicht mehr als vier Wochen bewegen (vgl. [VM-XT05]).

Bei der Planung eines Projektes wird in der Regel zunächst ein **Grobplan** erstellt, in dem die wesentlichen Rahmendaten wie Projektanfang und Projektende, sowie die wesentlichen Meilensteine eingetragen werden. Das V-Modell XT macht beispielsweise Vorgaben bezüglich der in einem Grobplan mindestens einzuplanenden Meilensteine in Form von so genannten Entscheidungspunkten. Entscheidungspunkte dienen im V-Modell XT als Sollbruchstellen eines Projektes und müssen der Begutachtung durch den Auftraggeber oder durch das hausinterne Management standhalten.

Die Detailplanung fügt sich entsprechend einem Top-Down-Vorgehen in die im Grobplan definierten Abschnitte zwischen den Meilensteinen ein. Für die Strukturierung von Projektplänen stehen in der Netzplantechnik so genannte Sammelvorgänge zur Verfügung. Ein **Sammelvorgang** ist eine hierarchische Gruppierung von Vorgängen beziehungsweise anderen Sammelvorgängen.

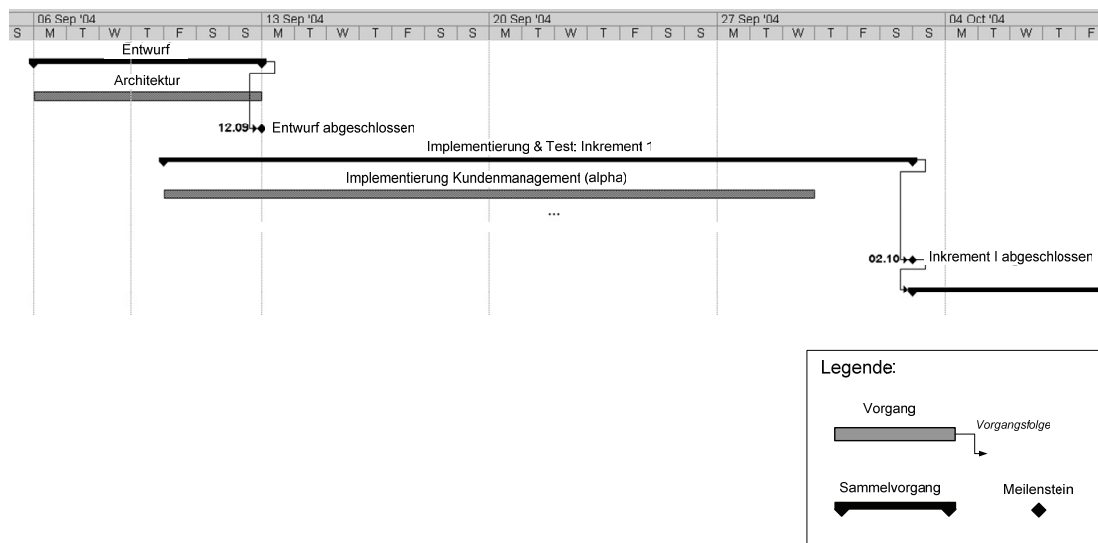


Abbildung 2–24: Balkendiagramm mit Sammelvorgängen und Meilensteinen: Beispiel

Meilensteine werden analog zu Vorgängen in das Abhängigkeitsgeflecht im Netzplan beziehungsweise im Balkendiagramm mit einbezogen. Abbildung 2–24 zeigt aufbauend auf dem Plan aus Abbildung 2–23 ein Balkendiagramm mit Sammelvorgängen und Meilensteinen. Für die Grobplanung von Projekten gelten die gleichen methodischen Herangehensweisen in Form der Erstellung von Produktstrukturplan, Projektstrukturplan und Netzplantechnik, die in den vorangehenden Abschnitten beschrieben sind.

2.3.3 Praktische Probleme der Projektplanung

In diesem Kapitel beleuchten wir einen Teil der Schwierigkeiten, die bei der Anwendung der im vorangehenden Kapitel beschriebenen Methodik für die Projektplanung in der Praxis zu bewältigen sind.

Planung ist die Vorbereitung zukünftigen Handelns. Planung bedeutet, im Voraus zu entscheiden, wie ein Ziel erreicht werden soll.

[...] Nachdem das Voraussehen des künftigen Projektverlaufs mit Unsicherheiten verbunden ist und sich der tatsächliche Ablauf nur bedingt dem geplanten angleichen kann, ist eine wiederholte Überarbeitung der Planung in Projekten eher der Normalfall als die Ausnahme. Die Aktivität Projekt planen ist eine fortlaufende Aktivität, die sich von der Projektinitialisierung bis zum Projektende erstreckt. [...]

[VM-XT05], Aktivität „Projekt planen“

Anspruchsvolle Aufgabe der Projektplanung ist die Zerlegung der Gesamtaufgabe in handhabbare Arbeitspakete mit klar definierten Schnittstellen. Eine Unwägbarkeit zu Beginn des Projektes stellt die Produktstruktur dar, die die Planung wesentlich beeinflusst.

Wurden auf Basis einer Produktstruktur eine Projektstruktur und mit dieser die einzuplanenden Vorgänge identifiziert, ist die Ermittlung von Bearbeitungsdauern nicht unproblematisch. Einerseits sind Schätzungen des Umfangs in der Regel mit einem großen Unsicherheitsfaktor verbunden, da sie auf reinem Erfahrungswissen beruhen. Andererseits ist die Ableitung der Bearbeitungsdauer aus dem geschätzten Umfang nicht trivial. Die Produktivität von Bearbeitern beziehungsweise Teams ist sehr unterschiedlich. Zudem ist die Korrelation

zwischen der Anzahl der einer Aktivität zugeordneten Mitarbeiter und der Dauer der Aktivität in Abhängigkeit vom geschätzten Umfang nicht linear. So ist keineswegs gesichert, dass ein „Mehr“ an Bearbeitern eine anstehende Aufgabe in kürzerer Zeit löst, unter anderem aufgrund eines erhöhten Kommunikationsaufwands.

Die Netzplantechnik bietet den Vorteil der Formulierung von Abhängigkeiten zwischen Vorgängen. Sie zwingt damit zur Systematik bei der Analyse der Schnittstellen beziehungsweise Zusammenhänge der geplanten Vorgänge. Damit wird eine zeitliche Optimierung der Abläufe möglich. Abhängig von der identifizierten Vorgangsfolge kann eine parallele oder zumindest überlappende Bearbeitung erfolgen. Die Festlegung der Abhängigkeiten im Netzplan ist allerdings nicht offensichtlich, sondern erfordert detaillierte Kenntnisse der zu bewältigenden Aufgaben. Selbst bei kleinen Entwicklungsprojekten sind die Abhängigkeiten mannigfaltig und von einer nicht zu unterschätzenden Komplexität.

Die Strukturierung eines Projektes in Form eines Plans mit handhabbaren Aufgabenpaketen ist also eine intellektuelle beziehungsweise kreative Leistung des Projektleiters. Ausschlaggebend ist dabei Erfahrungswissen. Ein Projektplan für ein umfangreiches Projekt kann sehr leicht unübersichtlich werden. Dies zeigt sich bereits bei Projekten mittlerer Größe wie dem Projekt WEIT zur Erstellung des V-Modells XT. In Summe enthält der Projektplan des WEIT-Projektes rund 300 geplante Vorgänge und Meilensteine. Dabei weist der Projektplan allerdings einige Redundanzen auf.

Der Plan folgt einer Reihe von Prinzipien beziehungsweise Schemata. So werden alle Arbeitspakete in einer vorläufigen α - und einer abschließenden β -Version fertig gestellt. Das Projekt wird in sechs Zeitabschnitte, so genannte „Releases“ unterteilt, wobei am Ende jedes Releases eine Freigabe der Ergebnisse zur externen Qualitätssicherung und zum Kunden erfolgt. Die Arbeitspakete werden den Zeitabschnitten des Projektes zugeordnet. Die Zuordnung erfolgt passend zu den in der geplanten Ergebnisstruktur definierten Abhängigkeiten. So werden beispielsweise Vorgehensbausteine vor anderen Vorgehensbausteinen eingeplant, deren Erarbeitung die Basis für die Erarbeitung anderer Vorgehensbausteine darstellt.

Während das Schema hinter der Erstellung des Plans logisch und einheitlich ist, scheint die Netzplantechnik nicht unmittelbar geeignet zu sein, dieses Schema zu vermitteln. Vielmehr würde der um Verständnis bemühte Betrachter ein Modell eines Netzplans sehen wollen, das die wesentlichen angewandten Prinzipien nicht redundant darstellt. Eine Verifizierung der Durchführbarkeit eines Projektes anhand eines Projektplans in Netzplan-Notation ist schwierig. Geeigneter wäre hierfür der Blick auf die grundlegenden Annahmen, sowie die Verifizierung der korrekten Umsetzung dieser Annahmen im Plan. Wünschenswert wäre also eine Abstraktion dieses Projektplans in Form eines Modells. In den Kapiteln 5 und 6 betrachten wir als durchgängiges Beispiel einen dem WEIT-Projekt angelehnten Projektplan.

2.4 Definition Projektplan

In diesem Kapitel geben wir eine Definition für den Begriff Projektplan in Form einer allgemeinen Ontologie für den Aufbau von Projektplänen. Dabei handelt es sich um eine Zusammenfassung des vorangehenden Kapitels. Den Zusammenhang zu den Konzepten eines Vorgehensmodells stellen wir in den Kapiteln 5 und 6 her. In diesem Kapitel betrachten wir Projektpläne ohne dabei den Bezug zu Vorgehensmodellen herzustellen, da dieser Zusammenhang nicht dem State-of-The-Art entspricht.

Die Definition des Projektplans in diesem Kapitel orientiert sich an der Netzplantechnik (vgl. [Balzert98]), sowie an verbreiteten Planungswerkzeugen wie MS Project (vgl. [MSProj03]).

Ein **Projektplan** beschreibt die gewählte Vorgehensweise eines Projektes und legt fest, was, wann und von wem zu tun ist. Der Grad der Detaillierung kann dabei unterschiedlich gewählt werden.

In Abbildung 2–25 stellen wir den Aufbau von Projektplänen dar. Um die Abbildung übersichtlich zu gestalten, wurde auf die detaillierte Modellierung von Terminen verzichtet. Unter dem Begriff „Termin“ in Abbildung 2–25 subsumieren wir immer den Anfangs- und den Endtermin. In Abbildung 2–25 kennzeichnen schwarze Pfeile die Leserichtung einer Assoziation. Nicht ausgefüllte Pfeilenden repräsentieren das Konzept der Spezialisierung. So sind Sammelvorgänge, Meilensteine und Vorgänge als spezielle Planungselemente zu verstehen.

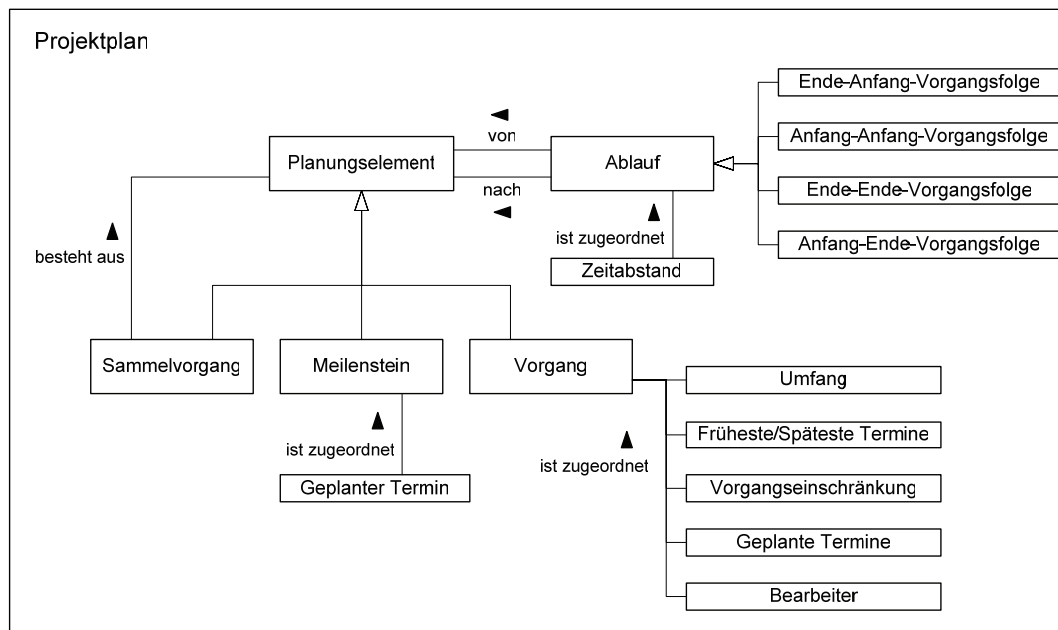


Abbildung 2–25: Allgemeiner Aufbau von Projektplänen (Ontologie)

Zentrales Element jeden Projektplans sind die geplanten Vorgänge.

Unter einem **Vorgang** verstehen wir eine im Projekt zu erledigende zeitlich begrenzte Aufgabe oder eine fortlaufende Tätigkeit.

Vorgänge werden mit unterschiedlichsten Attributen annotiert. Geschätzter Umfang eines Vorgangs, früheste beziehungsweise späteste Anfangs- oder Endtermine, sowie Vorgangseinschränkungen sind Attribute, die in der Netzplantechnik verwendet werden (vgl. Kapitel 2.3.2, Abschnitt Netzplantechnik). Neben frühesten und spätesten Terminen enthält der Projektplan die geplanten Anfangs- und Endtermine der einzelnen Vorgänge. Im Rahmen der Ressourcenplanung werden den Vorgängen zudem Bearbeiter zugeordnet.

Ein **Meilenstein** markiert den Abschluss der Erstellung einer Menge von (Zwischen-) Ergebnissen im Projekt. Im Gegensatz zu einem Vorgang wird ein Meilenstein auf der Zeitachse als Punkt eingeplant.

Sammelvorgänge dienen der Strukturierung von Projektplänen und damit der Übersichtlichkeit.

Ein **Sammelvorgang** ist eine hierarchische Gruppierung von Vorgängen, Meilensteinen oder anderen Sammelvorgängen. Sammelvorgänge werden nicht zeitlich geplant, vielmehr ergibt sich ihre zeitliche Ausdehnung aus den enthaltenen Planungselementen.

Vorgänge, Meilensteine und Sammelvorgänge fassen wir unter dem Oberbegriff **Planungselemente** zusammen. Vorgangsfolgen definieren **Abläufe** dieser Planungselemente.

Vorgangsfolgen sind Einschränkungen der möglichen zeitlichen Reihenfolge von Planungselementen in Bezug auf mögliche Anfangstermine und Endtermine. Unterschieden werden die Vorgangsfolgen Ende-Anfang, Anfang-Anfang, Ende-Ende und Anfang-Ende.

Vorgangsfolgen können also beispielsweise Vorgänge oder Sammelvorgänge in Beziehung setzen. Erläuterungen der Semantik der unterschiedlichen Arten von Vorgangsfolgen finden sich in Kapitel 2.3.2 im Abschnitt Netzplantechnik. Meilensteine und Sammelvorgänge werden wie Vorgänge in das Geflecht der Vorgangsfolgen eingebunden. Das Attribut Zeitabstand dient der Vorgabe von zeitlichen Abständen in Bezug auf die jeweilige Art einer Vorgangsfolge.

2.5 Zusammenfassung

Wir haben einen Überblick über den State-of-the-Art von Vorgehensmodellen und Methodik und Vorgehen der Projektplanung gegeben. Es existiert heute eine Vielzahl von Vorgehensmodellen. Um dieses Feld überblicksartig darstellen zu können haben wir ein Schema zur Klassifikation von Vorgehensmodellen angegeben. Vorgehensmodelle können demnach anhand der Merkmale Projektdurchführungsstrategie, Anpassbarkeit, inhaltliches Spektrum, Abstraktionsgrad sowie der Striktheit ihrer Anwendung unterschieden werden. Eine beispielhafte Einordnung einiger Vorgehensmodelle entsprechend dieser Merkmale haben wir angegeben.

In der vorliegenden Arbeit betrachten wir Vorgehensmodelle, die unterschiedliche Projektdurchführungsstrategien in sich vereinen, die einen hohen Grad an Anpassbarkeit aufweisen und einem Projekt Bausteine aus einem breit gefächerten inhaltlichen Spektrum zur Auswahl anbieten. Während diese Zielsetzungen denen des V-Modells XT entsprechen, streben wir im Vergleich zum V-Modell XT einen geringeren Abstraktionsgrad und eine weitergehende Unterstützbarkeit durch Werkzeuge an. Wir konzipieren im Folgenden entsprechend dem Thema der Arbeit Vorgehensmodelle, die sich eigens für die Ableitung von Projektplänen eignen. In ihrem Abstraktionsgrad unterscheiden sich diese Vorgehensmodelle allein durch bestimmte projektspezifische Parameter von der Projektebene. Die Einstellung dieser Parameter durch den Projektleiter ermöglicht die Ableitung eines Projektplans.

Um die Begriffe Vorgehensmodell und Projektplan plastisch darzustellen haben wir eine Ontologie des Aufbaus von Vorgehensmodellen sowie des Aufbaus von Projektplänen angegeben. Dabei handelt es sich um die Konzepte, die den meisten heutigen Vorgehensmodellen gemein ist und die den State-of-the-Art auf dem Gebiet der Projektplanung widerspiegeln. Für die Projektplanung haben wir Vorgehen und Methodik entsprechend der Literatur dargestellt, ohne dabei auf den Zusammenhang zwischen Vorgehensmodell und Projektplan einzugehen, da dieser in der Literatur nicht beschrieben wird. Der Weg zum Projektplan führt über die Erstellung eines Produkt- und eines Projektstrukturplans, sowie über die Netzplantechnik. Die Identifikation von Vorgangsfolgen in der Netzplantechnik, sowie die Ressourcen- und die Terminplanung müssen dabei Hand in Hand gehen.

Die Darstellung der praktischen Probleme der Projektplanung hebt die erforderliche intellektuelle beziehungsweise kreative Leistung des Projektleiters bei der Erstellung eines Projektplans hervor, die vor allem auf Erfahrungswissen beruht. Dies birgt im Falle eines unerfahrenen Projektleiters das Risiko eines Scheiterns des Projektes in sich. In den Kapiteln 5 und 6 erstellen wir ein Konzept für die Konservierung des mit der Planung verbundenen Erfahrungswissens in Form eines Vorgehensmodells. Dazu definieren wir in den folgenden Kapiteln Konzepte für die Beschreibung eines Vorgehensmodells als Abstraktion „guter“ Projektpläne.

3 Schwachstellenanalyse des V-Modells XT

Im vorliegenden Kapitel analysieren wir eine Reihe von Schwachstellen des V-Modells XT (vgl. [VM-XT05]). Das Kapitel dient uns als Ausgangspunkt für die Konzeption eines verbesserten Meta-Modells für Vorgehensmodelle, das der Zielsetzung der Arbeit gerecht wird.

In der Analyse betrachten wir Schwachstellen, die sich aus dem Meta-Modell des V-Modells XT ergeben. Die Beschreibungsmittel des V-Modells XT führen teilweise zu einer ungünstigen inhaltlichen Umsetzung des V-Modells. Die Folge sind Verständnisprobleme auf Seiten der Anwender, Schwierigkeiten im Rahmen des Weiterentwicklungsprozesses sowie eine eingeschränkte Unterstützbarkeit des V-Modells XT durch Werkzeuge. Insbesondere hinsichtlich der Unterstützbarkeit des V-Modells XT durch ein Werkzeug für die Projektplanung decken wir Schwachstellen auf. In den weiteren Kapiteln der Arbeit erarbeiten wir zunehmend auf diese Schwachstellen ein verbessertes Meta-Modell für Vorgehensmodelle.

Der Aufbau des Kapitels orientiert sich am Meta-Modell des V-Modells XT. Das Kapitel gliedert sich in die Betrachtung der Konzepte des Meta-Modells zu Vorgehensbausteinen, Abläufen und Produkten, sowie einer Betrachtung der Abstraktionsprinzipien im V-Modell XT. Die Konzepte des Meta-Modells werden jeweils dargestellt und erläutert. Eine entsprechende präzise Darstellung findet sich derzeit weder im V-Modell XT selbst, noch in sonstigen Publikationen. Der Autor der Arbeit bezieht sein Wissen aus der Beteiligung am Erstellungsprozess des V-Modells XT. Die identifizierten Schwachstellen und Probleme des V-Modells XT erläutern wir jeweils anschaulich anhand von Beispielen.

Inhalt des Kapitels	Seite
3.1 Vorgehensbausteine	59
3.2 Abläufe	68
3.3 Produkte	74
3.4 Abstraktion im V-Modell	80

3.1 Vorgehensbausteine

Die Vorgehensbausteine des V-Modells XT sind als modulare Bearbeitungs- und Tailoring-Einheiten konzipiert. Das V-Modell XT besteht derzeit aus 18 Vorgehensbausteinen (vgl. Abbildung 2–9).

Nach einer Erläuterung des Konzeptes Vorgehensbaustein im Meta-Modell des V-Modells XT beleuchten wir Problembereiche. Als Problembereiche stellen sich die Verwendung des Vorgehensbausteins als Bearbeitungs-Einheit sowie als Tailoring-Einheit, die erzwungene Abstraktion von Texten durch den Erweiterungsmechanismus der Vorgehensbausteine sowie die Granularität des Tailorings im V-Modell XT dar.

Abschnitte in Kapitel 3.1	Seite
3.1.1 Vorgehensbausteinconcept des V-Modells XT	60
3.1.2 Bearbeitungs-Einheiten	63
3.1.3 Tailoring-Einheiten	65
3.1.4 Abstraktion durch Erweiterung	66
3.1.5 Granularität des Tailorings	67
3.1.6 Zusammenfassung	67

3.1.1 Vorgehensbausteinconcept des V-Modells XT

Zunächst erläutern wir das Konzept des Vorgehensbausteins.

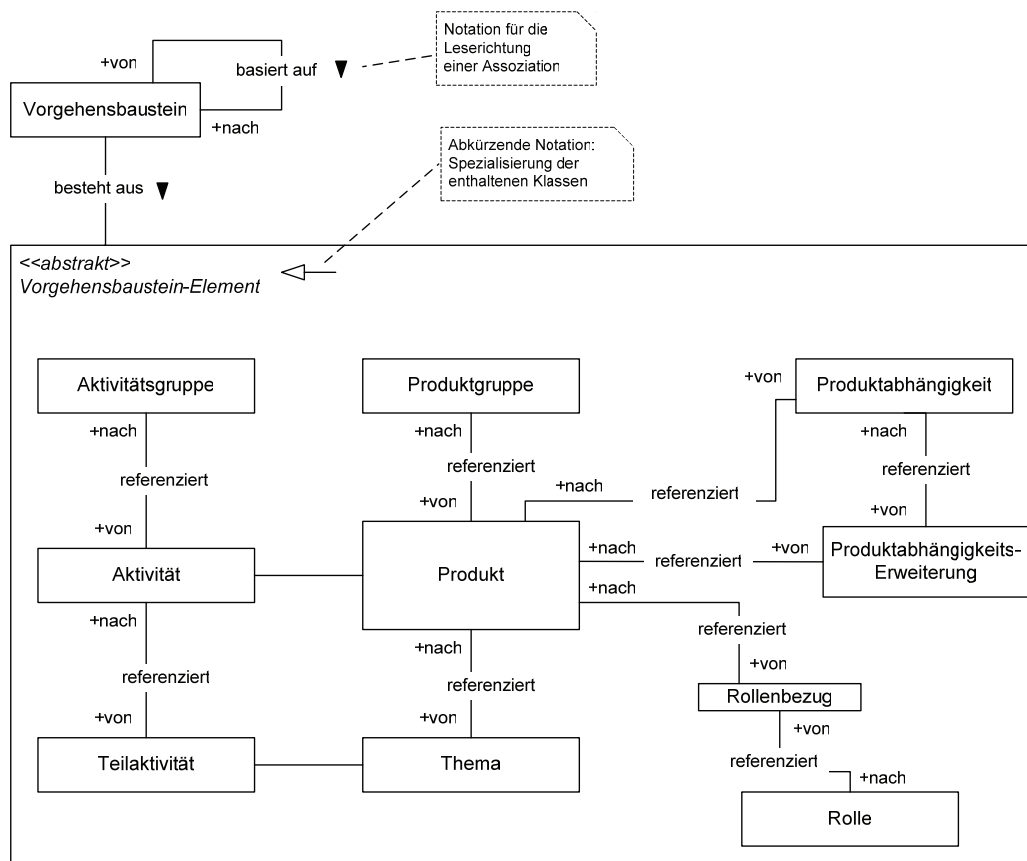


Abbildung 3–1: Vorgehensbausteinconcept des V-Modells XT

In Abbildung 3–1 zeigen wir die zentralen Konzepte des V-Modells XT in Bezug auf Vorgehensbausteine und deren Abhängigkeiten. Die Notation der Abbildung ist die eines UML-Klassendiagramms (vgl. [OMG03a]). Die spezifischen Aspekte der Notation, die nicht dem UML-Standard entstammen, uns aber für eine übersichtliche Darstellung der Abbildung günstig erscheinen, erläutern wir in Abbildung 3–1 mit Hilfe von Notizen. Wir verwenden Spezialisierung hier allein zur übersichtlicheren Darstellung der Bestandteile eines Vorge-

hensbausteins. Für die Formulierung von Constraints verwenden wir eine der Object Constraint Language (OCL) (vgl. [OMG03c]) ähnliche Notation. Die Konzepte dieser Constraint-Sprache sind dabei dieselben, einzig die Syntax ist unterschiedlich zur OCL, dabei aber intuitiv verständlich.

Abbildung 3–1 abstrahiert zum einen von den Multiplizitäten der dargestellten Assoziationen und zum anderen von den unterschiedlichen Ausprägungen und der Richtung der Produktabhängigkeiten. Beides stellen wir erst in den folgenden Kapiteln dar (vgl. Kapitel 3.2, 3.3 und 3.4). Zudem abstrahieren wir in Abbildung 3–1 von den unterschiedlichen Ausprägungen der Assoziationen *referenziert*. Die Referenzierung zwischen Aktivität und Aktivitätsgruppe entspricht beispielsweise einer Aggregations-Beziehung. Für die Betrachtung der Konzepte Vorgehensbaustein und Abhängigkeiten zwischen Vorgehensbausteinen ist dies allerdings nicht relevant.

Vorgehensbausteine fassen die zentralen Beschreibungselemente des V-Modells XT – *Produkte, Aktivitäten* und *Rollen* – zu modularen Einheiten zusammen. Im Rahmen des Tailorings werden Vorgehensbausteine ausgewählt und zu einem projektspezifischen V-Modell zusammengesetzt. Die Komposition von Vorgehensbausteinen im Rahmen des Tailorings folgt klar definierten Regeln. Die Abhängigkeiten der Vorgehensbausteine in Form der Assoziation „*basiert auf*“ sind dabei zu berücksichtigen.

Die in Abbildung 3–1 verwendete Notation der Spezialisierung ist nicht dahingehend zu interpretieren, dass über die Assoziation *referenziert* verbundene Instanzen von *Vorgehensbaustein-Elementen* im selben Vorgehensbaustein angesiedelt sein müssen. Referenzierung kann über die Grenzen der Vorgehensbausteine erfolgen und tut dies im V-Modell XT auch. Vorgehensbausteine stehen über die gerichtete Assoziation „*basiert auf*“ in Beziehung. Ein Vorgehensbaustein basiert auf einem anderen Vorgehensbaustein genau dann, wenn seine Vorgehensbaustein-Elemente Vorgehensbaustein-Elemente eines anderen Vorgehensbausteins referenzieren:

$$\forall e_1, e_2 \in \text{Vorgehensbaustein-Element: } e_1.\text{referenziert.nach}.e_2 \Leftrightarrow \\ e_2.\text{Vorgehensbaustein} = e_1.\text{Vorgehensbaustein} \vee \\ e_2.\text{Vorgehensbaustein} \in e_1.\text{Vorgehensbaustein.basiert auf}^{\text{nach}}$$

In obigem Constraint entspricht die Assoziation *referenziert* den Assoziationen *referenziert* zwischen den Spezialisierungen der abstrakten Vorgehensbaustein-Elemente, also beispielsweise der Assoziation *referenziert* zwischen Aktivitätsgruppen und Aktivitäten. Dieses Constraint ließe sich auch durch einzelne Constraints je Assoziation ausdrücken. Die Richtung der Assoziation *referenziert* muss der Richtung der Assoziation „*basiert auf*“ entsprechen. Die Richtung ist dabei in Form der Assoziations-Enden *von* und *nach* festgelegt. In obigem Constraint legen wir die Richtung der Assoziation „*basiert auf*“ durch den Ausdruck „*basiert auf* ^{nach}“ fest.

Zudem gelten die folgenden Constraints. Die „*basiert auf*“-Assoziation muss transitiv und frei von Zyklen sein. Für Produkte und Themen gilt, dass die jeweils zugeordneten Aktivitäten und Teilaktivitäten in denselben Vorgehensbausteinen liegen müssen. Produktgruppen und Aktivitätsgruppen sind jeweils gleichnamig und ebenfalls denselben Vorgehensbausteinen zugeordnet.

Abhängigkeiten zwischen Vorgehensbausteinen kommen beispielsweise durch das "Einhängen" eines Produktes unter der Produktgruppe eines anderen Vorgehensbausteins oder durch den Verweis einer Produktabhängigkeit auf ein Produkt eines anderen Vorgehensbausteins zustande. Die Notation der Abbildung 3–2 entspricht der eines UML-Instanzdiagramms. Wie in Abbildung 3–2 dargestellt, referenziert beispielsweise das Produkt *SW-Architektur* des Vorgehensbausteins *SW-Entwicklung* die Produktgruppe *Systementwurf*, die im Vorgehensbaustein *Systemerstellung* liegt. Produktgruppen sind im Gegensatz zu Vorge-

hensbausteinen keine Tailoring-Einheiten, sondern disjunkte, inhaltliche Gruppierungen von Produkten.

Für das Tailoring, also die projektspezifische Auswahl von Vorgehensbausteinen, ergibt sich die folgende Regel. Basiert ein Vorgehensbaustein auf einem anderen Vorgehensbaustein, so kann der Vorgehensbaustein nur dann zur Anwendung kommen, wenn seine Basis ebenfalls im Projekt verwendet wird.

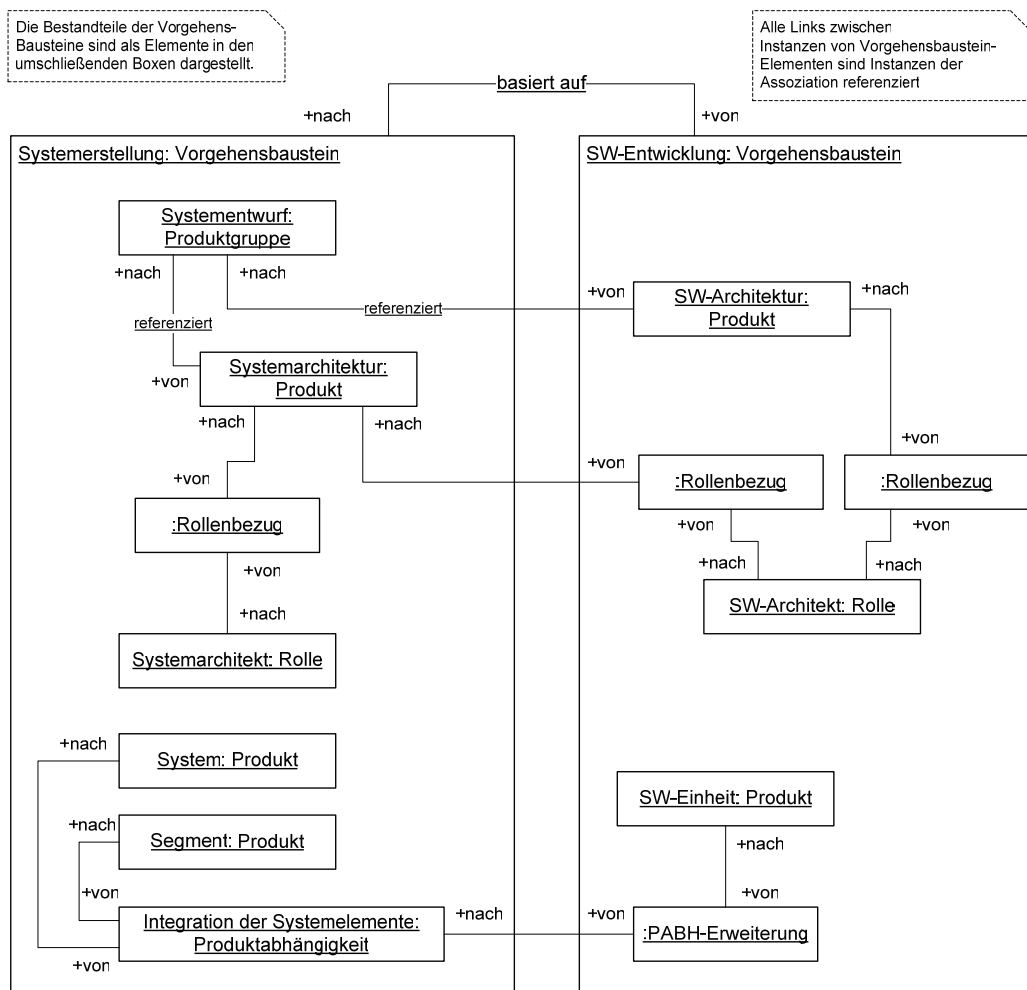


Abbildung 3–2: Instanzdiagramm zum Vorgehensbausteinkonzept (Beispiel): Rollenbeziehungen und Produktabhängigkeitserweiterungen

Das Erweiterungskonzept für Produktabhängigkeiten (*Produktabhängigkeits-Erweiterungen*) dient der Reduzierung der Abhängigkeiten zwischen Vorgehensbausteinen. Da im V-Modell XT Produktabhängigkeiten als n-stellige Assoziationen zwischen Produkten modelliert sind (vgl. dazu Kapitel 3.3), würde das Fehlen dieses Erweiterungsmechanismus zu vielen zusätzlichen Abhängigkeiten zwischen den Vorgehensbausteinen führen. Eine Produktabhängigkeits-Erweiterung kann in einen abhängigen Vorgehensbaustein verschoben werden. Würde beispielsweise die Produktabhängigkeit „*Integration der Systemelemente*“ in Abbildung 3–2 das Produkt *SW-Einheit* direkt referenzieren, würde dies zu einer Abhängigkeit der Vorgehensbausteine führen. Der Vorgehensbaustein Systementwicklung müsste dann auf dem Vorgehensbaustein SW-Entwicklung basieren. Damit wäre die Assoziation

„basiert auf“ nicht mehr zyklensfrei. Beide Vorgehensbausteine könnten dann im Rahmen des Tailorings nur noch gemeinsam gewählt werden. In Konsequenz würde sich anbieten, beide Vorgehensbausteine zu einem zu vereinigen.

Der Bezug einer Rolle zu einem Produkt wird im V-Modell XT dementsprechend modelliert. Das Konzept *Rollenbezug* kann einem Vorgehensbaustein zugeordnet werden. Dies ermöglicht die Modellierung der Mitwirkung des *SW-Architekten* an der *Systemarchitektur* ohne eine zusätzliche Abhängigkeit des Vorgehensbausteins Systemerstellung vom Vorgehensbaustein SW-Entwicklung.

Im Erstellungsprozess des V-Modells XT wurden die Abhängigkeiten der Vorgehensbausteine als Vorgabe für die Erarbeitung der Inhalte der Vorgehensbausteine gesehen. Den Vorgehensbausteinen und ihren Abhängigkeiten kommt also sozusagen die Rolle einer Architektur als Vorgabe für die „Implementierung“ des V-Modells XT zu. Ein Vorgehensbaustein darf gemäß dieser Architektur alle Elemente anderer Vorgehensbausteine referenzieren, auf denen er direkt oder indirekt basiert.

Vorgehensbausteine dienen nicht allein als Tailoring-Einheiten, sondern auch als Bearbeitungs-Einheiten im Erstellungsprozess des V-Modells XT. Neben der Rolle eines eigenständig verantworteten Arbeitspaketes während der Erstellung kommt dem Konzept Vorgehensbaustein auch die Rolle einer modularen Einheit im Weiterentwicklungsprozess des V-Modells XT zu. Die Konzepte Tailoring-Einheit, Bearbeitungs-Einheit und Weiterentwicklung-Einheit finden sich also vereinigt im Konzept des Vorgehensbausteins.

Das Konzept des Vorgehensbausteins führt zu einer Reihe von Anomalien in der inhaltlichen Umsetzung des V-Modells XT. In den folgenden Kapiteln greifen wir ohne Anspruch auf Vollständigkeit einige dieser Probleme auf.

3.1.2 Bearbeitungs-Einheiten

Die Probleme des Vorgehensbaustein-Konzeptes als Bearbeitungs-Einheit veranschaulichen wir anhand des Beispiels der Qualitätssicherung. Abbildung 3–3 zeigt die in diesem und in den folgenden Kapiteln verwendeten Beispiele zu Schwachstellen des Vorgehensbaustein-Konzeptes.

Das V-Modell XT definiert bestimmte Vorgehensbausteine wie Qualitätssicherung oder Projektmanagement als Kernelemente jedes Vorgehens. Diese können im Rahmen des Tailorings nicht entfallen. Die Vorgehensbausteine des Kerns werden von den Vorgehensbausteinen außerhalb des Kerns referenziert. Eine Abhängigkeit der Vorgehensbausteine des Kerns zu Vorgehensbausteinen außerhalb des Kerns darf dagegen nicht existieren, um zyklische Abhängigkeiten der Vorgehensbausteine zu vermeiden. Dies führt dazu, dass Kern-Vorgehensbausteine keine spezifischen Aussagen zu Inhalten aus Vorgehensbausteinen außerhalb des Kerns machen können.

Beispiel
„Qualitätssicherung“

Der Kern-Vorgehensbaustein Qualitätssicherung basiert beispielsweise allein auf dem Vorgehensbaustein Projektmanagement, der ebenfalls zum Kern gehört. Entsprechend seiner Natur ist die Qualitätssicherung allerdings querschnittlich zu anderen Vorgehensbausteinen zu sehen und hätte daher viele Abhängigkeiten zu anderen Vorgehensbausteinen, die allerdings aufgrund der Konsistenzregeln des V-Modells nicht modellierbar sind.

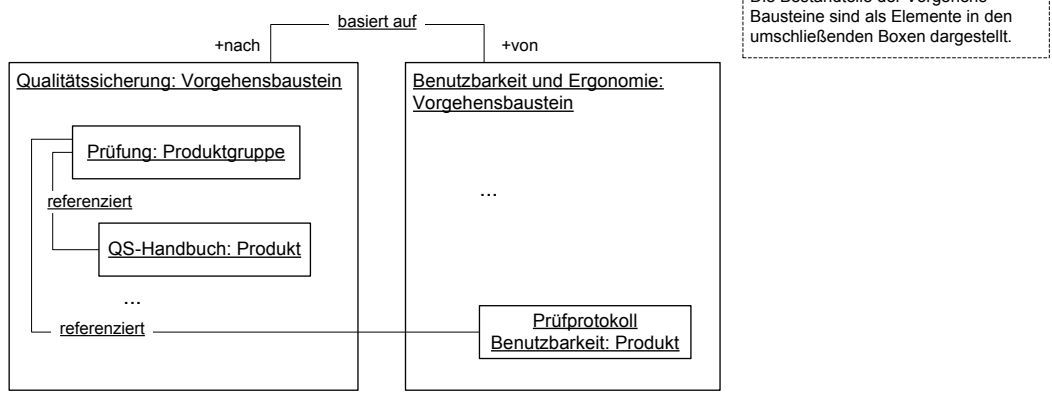
Daher ergänzen Vorgehensbausteine außerhalb des Kerns Produkte in der Produktgruppe *Prüfung*, die im Vorgehensbaustein *Qualitätssicherung* enthalten ist. Beispielsweise ergänzt der Vorgehensbaustein „*Benutzbarkeit und Ergonomie*“ ein „*Prüfprotokoll Benutzbarkeit*“ in der Produktgruppe *Prüfung*, wie in Abbildung 3–3 (a) dargestellt.

Aus Sicht des Vorgehensbausteins als Tailoring-Einheit erscheint die Aufteilung der inhaltlichen Aspekte Qualitätssicherung auf unterschiedliche Vorgehensbausteine durchaus geeignet. Grundlegende Aspekte der Qualitätssicherung, die in jedem V-Modell-Projekt enthalten

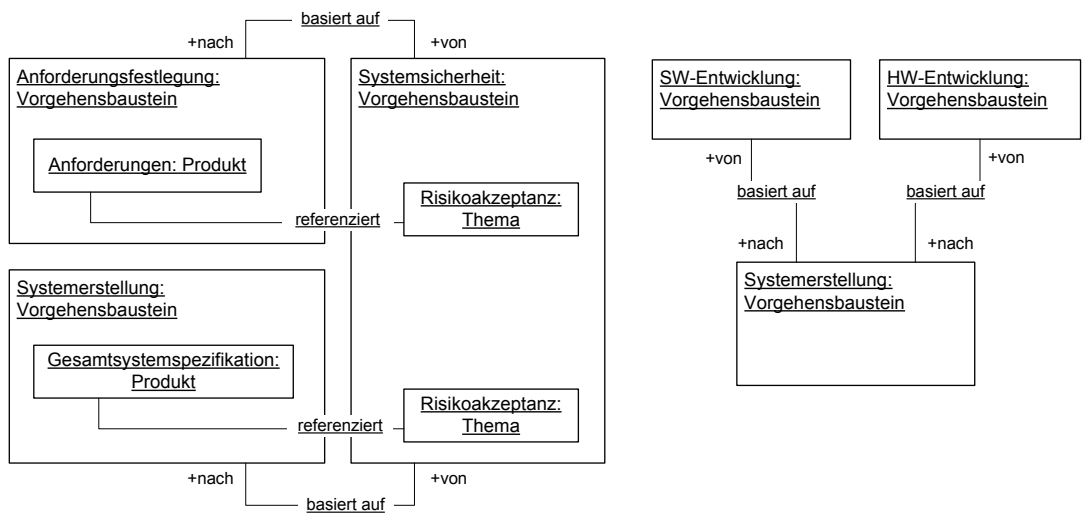
sein müssen, liegen in einem Vorgehensbaustein im Kern, spezifische Aspekte wie Benutzbarkeit liegen in anderen Vorgehensbausteinen.

Als modulare Bearbeitungs-Einheit ist der Vorgehensbaustein Qualitätssicherung damit allerdings ungeeignet, da sich einzelne Aspekte der Qualitätssicherung in anderen Vorgehensbausteinen und damit in anderen Bearbeitungs-Einheiten lokalisieren. Die Beschreibung des Vorgehens zur Qualitätssicherung, sowie künftige Änderungen und Erweiterungen im Rahmen der Pflege und Weiterentwicklung des V-Modells XT, können sich daher nicht im Vorgehensbaustein als modularer Bearbeitungs-Einheit lokalisieren. Eine Anordnung des Vorgehensbausteins Qualitätssicherung außerhalb des Kerns, also mit anderen Vorgehensbausteinen als Basis der Disziplin Qualitätssicherung, würde ein anderes Problem nach sich ziehen, das wir im folgenden Abschnitt darstellen.

(a) Vorgehensbausteine als Bearbeitungs-Einheiten



(b) Vorgehensbausteine als Tailoring-Einheiten



(c) Abstraktion durch Erweiterung

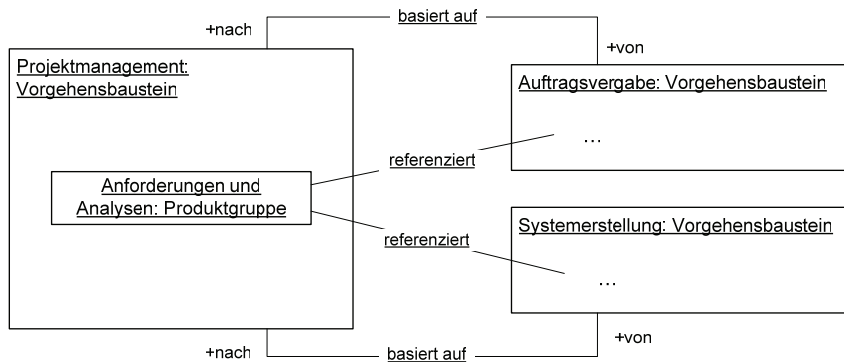


Abbildung 3–3: Beispiele zu Schwachstellen des V-Modells XT: Vorgehensbaustein-konzept (Instanzdiagramm)

Die in Abbildung 3–3 dargestellten Assoziationen „*basiert auf*“ entsprechen teilweise nicht den direkten Abhängigkeiten im V-Modell XT, sondern ergeben sich aus der transitiven Hülle der Vorgehensbaustein-Abhängigkeiten. Die Namen der Elemente in Abbildung 3–3 sind zudem teilweise gekürzt.

3.1.3 Tailoring-Einheiten

Die Probleme des Vorgehensbaustein-konzeptes als Tailoring-Einheit veranschaulichen wir anhand der Beispiele Systemsicherheit und Systemerstellung des V-Modells XT.

Die Disziplin Sicherheit ist wie die Qualitätssicherung von querschnittlicher Natur. Das Thema Sicherheit lokalisiert sich im V-Modell XT im Vorgehensbaustein Systemsicherheit. Dieser Vorgehensbaustein ist im äußeren Kranz der Vorgehensbausteinlandkarte angeordnet, das heißt die Systemsicherheit basiert auf einigen anderen Vorgehensbausteinen, wird aber selbst nicht referenziert. Als Bearbeitungs-Einheit erscheint der Vorgehensbaustein Systemsicherheit geeignet, da er alle inhaltlichen Aspekte der Disziplin Sicherheit kapselt.

**Beispiel
„System-
sicherheit“**

Als Tailoring-Einheit ist der Vorgehensbaustein allerdings problematisch. Die *Systemsicherheit* ergänzt beispielsweise jeweils ein Thema *Risikoakzeptanz* sowohl im Produkt *Anforderungen* als auch im Produkt *Gesamtsystemspezifikation*, wie in Abbildung 3–3 (b) dargestellt. Die beiden Produkte treten aber in einem V-Modell Projekt nie gemeinsam auf, da die Anforderungen in einem Vorgehensbaustein für reine Auftraggeber-Projekte und die Gesamtsystemspezifikation in einem Vorgehensbaustein für Auftragnehmer-Projekte definiert wird.

Als modulare Einheit des Tailorings ist der Vorgehensbaustein Systemsicherheit also ungeeignet, da in jedem Projekt stets einige Elemente dieses Vorgehensbausteins nicht relevant sind. Themen, die sich unter Produkte eines nicht ausgewählten Vorgehensbausteins einhängen, hängen sozusagen „in der Luft“ (vgl. dazu Kapitel 3.3.1). Ein Werkzeug zur Unterstützung des Tailorings muss also je nach Vorgehensbaustein-Auswahl Elemente bestimmter Vorgehensbausteine löschen. Diese Verschiebung des Problems auf ein Tailoring-Werkzeug stellt sich allerdings im Rahmen des Erstellungsprozesses des V-Modells XT als ungünstig dar, da anhand der Beschreibungstechnik das Tailoring für die Prozessingenieure nicht offensichtlich wird.

Entsprechend einem Tailoring-Konzept basierend auf modularen Einheiten sollten also die Inhalte des Vorgehensbausteins Systemsicherheit auf andere Vorgehensbausteine aufgeteilt

sein, sich dabei aber in derselben Bearbeitungs-Einheit lokalisieren. Das Konzept des Vorgehensbausteins als Tailoring- und Bearbeitungs-Einheit zugleich steht dem im Weg.

**Beispiel
„System-
erstellung“**

Als weiteres Beispiel betrachten wir in Abbildung 3–3 (b) die Systemerstellung. Entsprechend der Architektur des V-Modells XT basieren die Vorgehensbausteine *SW-Entwicklung* und *HW-Entwicklung* auf dem Vorgehensbaustein *Systemerstellung*. Die Aufteilung dieser Vorgehensbausteine im Sinne von Bearbeitungs-Einheiten erscheint sinnvoll. Die Vorgehensbaustein-Abhängigkeiten geben für den Erstellungsprozess des V-Modells XT die Erarbeitung einer gemeinsamen Basis für die SW- und HW-spezifischen Entwicklungsthemen vor, nämlich den Vorgehensbaustein *Systemerstellung*. Dies erscheint sinnvoll. Problematisch ist hierbei allerdings die gleichzeitige Verwendung der Vorgehensbaustein-Abhängigkeiten als Konsistenzregeln für die projektspezifische Auswahl von Vorgehensbausteinen.

In einem reinen Softwareentwicklungsprojekt nach V-Model XT ist also stets der Vorgehensbaustein *Systemerstellung* zusammen mit der SW-Erstellung zu wählen. Neben einer SW-Architektur ist also beispielsweise stets eine Systemarchitektur zu erstellen. Sind die Randbedingungen zu Beginn des Projektes hinreichend klar, so dass eine reine Softwareentwicklung abzusehen ist, sollte die Notwendigkeit der Erstellung einer Systemarchitektur im Projekt im Rahmen des Tailorings entfallen können. Die Architektur der Vorgehensbaustein-Landkarte stellt sich in Bezug auf Entwicklungsprojekte ohne Systemanteil als Problem dar.

Unter dem Gesichtspunkt des Tailorings sollte demnach der Vorgehensbaustein *Systemerstellung* nicht die Basis der SW- und HW-Entwicklung sein, sondern eine Erweiterung beider Vorgehensbausteine.

3.1.4 Abstraktion durch Erweiterung

Der Mechanismus der Erweiterung bei der Komposition von Vorgehensbausteinen, also beispielsweise die Erweiterung von Produktgruppen durch Produkte anderer Vorgehensbausteine, führt teilweise zu ungewollter Abstraktion der beschreibenden Texte des V-Modells XT.

Die Beschreibung einer Produktgruppe sollte beispielsweise nur die in der Produktgruppe enthaltenen Produkte im Text referenzieren, die der Vorgehensbaustein gemäß den Abhängigkeiten der Vorgehensbausteine referenzieren darf. Andernfalls könnten bestimmte projektspezifische Auswahlmengen von Vorgehensbausteinen auf Elemente außerhalb dieser Auswahl verweisen. Die Konsistenzbedingungen bezüglich der Abhängigkeiten der Vorgehensbausteine gelten also nicht nur auf Ebene der Assoziationen des Meta-Modells, sondern auch für die beschreibenden Texte zu den einzelnen Elementen des Meta-Modells.

**Beispiel
Produkt-
gruppe „An-
forderungen
und Analy-
sen“**

Die Produktgruppe *„Anforderungen und Analysen“* wird beispielsweise von den Vorgehensbausteinen *Anforderungsfestlegung* und *Systemerstellung* referenziert, wie in Abbildung 3–3 (c) dargestellt. Produktgruppen fassen Produkte nach inhaltlichen Gesichtspunkten zusammen. Da diese Vorgehensbausteine in keiner Abhängigkeitsbeziehung stehen, wurde die Produktgruppe *„Anforderungen und Analysen“* zur Gewährleistung der Konsistenz in einen gemeinsam abhängigen Vorgehensbaustein verschoben, in diesem Fall in den Vorgehensbaustein *Projektmanagement*. Im beschreibenden Text der Produktgruppe *„Anforderungen und Analysen“* kann also eigentlich auf kein enthaltenes Produkt Bezug genommen werden, da das Projektmanagement selbst kein Produkt in die Produktgruppe einhängt.

Bei einer konsistenten Umsetzung des V-Modells XT würde dies zu einer Abstraktion des beschreibenden Textes dieser Produktgruppe führen, die nicht gewollt und nicht sinnvoll ist. Eine stimmige Abstraktion ohne jegliche Kenntnis von Details ist in der Regel nicht möglich. Das V-Modell XT behilft sich an dieser Stelle mit einer Lockerung der Konsistenzregeln in Bezug auf Texte.

Die Konsistenzbedingungen in Bezug auf beschreibenden Text werden im V-Modell XT also nicht konsequent eingehalten. Eine Prüfung dieser Konsistenz könnte sich auf die im Text

enthaltenen Querverweise auf Instanzen von Beschreibungs-Elemente des Meta-Modells abstützen. Durch das Fehlen einer entsprechenden Prüfung wird das V-Modell XT weder dem Ziel des Vorgehensbausteins als Tailoring-Einheit noch als modulare Bearbeitungseinheit vollständig gerecht. Eine textuelle Überarbeitung lokalisiert sich damit nicht notwendigerweise in einem Vorgehensbaustein beziehungsweise der transitiven Hülle der abhängigen Vorgehensbausteine, sondern eventuell in einem anderen nicht als abhängig modellierten Vorgehensbaustein.

3.1.5 Granularität des Tailorings

Die Verwendung von Vorgehensbausteinen als Bearbeitungs-Einheiten beziehungsweise eigenständig verantworteten Arbeitspaketen führte für das V-Modell XT zu einer relativ überschaubaren Menge von 18 Vorgehensbausteinen. Die Vorgehensbaustein-Landkarte ist daher sehr grob partitioniert. Ein feingranulares Tailoring ist damit nicht möglich.

Um in Projekten keinen unnötigen Aufwand zu generieren, enthält das V-Modell XT aus diesem Grund einen weiteren Mechanismus, der sich in Produktabhängigkeiten der Kategorie Erzeugung manifestiert (vgl. dazu Kapitel 3.3.1). Eine erzeugende Produktabhängigkeit definiert, ob eine Instanz beziehungsweise wie viele Instanzen eines Produktes in einem Projekt benötigt werden. Die Mehrzahl der Produkte des V-Modells wird „erzeugt“, ist also im Projekt nicht in jedem Fall zu erstellen.

Die Erzeugung von SW-Einheiten anhand der Vorgaben eines Architekturdokumentes ist ein Beispiel einer erzeugenden Produktabhängigkeit, die durch einen Tailoring-Mechanismus nicht ersetzt werden kann. Andererseits fällt das Führen einer Nachweisakte, die über eine erzeugende Produktabhängigkeit projektspezifisch genau einmal erzeugt wird oder aber nicht erzeugt wird, in die Kategorie Tailoring. Da die Notwendigkeit des Führens von Nachweisen in der Regel bereits zu Projektbeginn abgesehen werden kann, würde sich die Definition eines entsprechenden Projektmerkmals als Tailoring-Kriterium anbieten. Auf der Vorgehensbaustein-Landkarte ließe sich beispielsweise ein Vorgehensbaustein zur erweiterten Qualitätssicherung ergänzen, der auf dem Vorgehensbaustein Qualitätssicherung basiert.

**Beispiel
Produkt
„Nachweis-
akte“**

An anderer Stelle der Vorgehensbaustein-Landkarte fällt die grobe Granularität der Vorgehensbausteine ebenfalls auf. So beinhaltet der Vorgehensbaustein Logistikkonzeption neben dem Produkt Nutzungsdokumentation, die in der Regel für kein Entwicklungsprojekt entfallen darf, weitere Produkte wie den Ersatzteilekatalog, der nur für bestimmte Projekte relevant ist. Der Vorgehensbaustein Logistik ist als Tailoring-Einheit somit vollkommen ungeeignet.

**Beispiel
„Logistik-
konzeption“**

Die geeignete Definition von Tailoring-Einheiten für Produkte, die mit dem Mechanismus der erzeugenden Produktabhängigkeiten nicht mehrfach erzeugt werden, sondern höchstens einfach, würde den Vorteil der einfacheren Anwendbarkeit bieten. Aus Sicht der Anwender des V-Modells wäre ein feingranulares Tailoring des V-Modells in jedem Fall wünschenswert (vgl. [DGR03]).

3.1.6 Zusammenfassung

Wie wir an einer Reihe von Beispielen gezeigt haben, sind das Konzept Vorgehensbaustein des Meta-Modells sowie dessen inhaltliche Umsetzung nicht immer konform mit der Zielsetzung des V-Modells XT. Die aufgezeigten Probleme rühren aus der Tatsache, dass die tatsächlichen Abhängigkeiten der Vorgehensbausteine komplex und vielfältig sind. Die Konsistenzregeln, die aus der Konzeption des Vorgehensbausteins sowohl als Tailoring-Einheit als auch als Bearbeitungs-Einheit folgen, sind verglichen mit der Komplexität der Abhängigkeiten innerhalb eines Vorgehensmodells zu eng gefasst.

Auf der Vorgehensbaustein-Landkarte des V-Modells XT finden sich Beispiele von Querschnitts-Disziplinen, die sich nicht zufrieden stellend in das Konzept des Vorgehensbau-

steins abbilden lassen. Bestimmte Vorgehensbausteine können entweder als Bearbeitungs-Einheit oder als Tailoring-Einheit ihren Zweck nicht erfüllen, da Bearbeitungs-Einheiten und Tailoring-Einheiten nach unterschiedlichen Kriterien orthogonal gebildet werden müssten. Die Verwendung des Vorgehensbausteins als Bearbeitungs-Einheit im Erstellungsprozess des V-Modells XT bringt zudem das Problem eines sehr grobgranularen Tailorings mit sich. Der Erweiterungsmechanismus der Vorgehensbausteine – beispielsweise die Erweiterung einer Produktgruppe durch Produkte eines anderen Vorgehensbausteins – führt schließlich zu einer ungewollten Abstraktion bestimmter Texte des V-Modells XT.

3.2 Abläufe

In diesem Kapitel wenden wir uns dem Ablauf-Konzept im V-Modell XT zu. Zielsetzung des V-Modells XT ist neben der Beschreibung durchzuführender Aktivitäten beziehungsweise zu erstellender Produkte die Beschreibung von Abläufen als Vorgabe für die Projektdurchführung.

In diesem Kapitel stellen wir zunächst das Ablaufkonzept dar. Dieses besteht aus dem Konzept der Projektdurchführungsstrategien sowie aus Aktivitätsdiagrammen auf der Ebene von Teilaktivitäten. Daraufhin beleuchten wir unterschiedliche Problembereiche. Zum einen sind bestimmte Abläufe mit dem Meta-Modell des V-Modells XT nicht beschreibbar. Zum anderen sind Aktivitätsdiagramme der UML für die Ableitung eines Projektplans nur bedingt geeignet, da sie der Netzplantechnik an Ausdrucksmächtigkeit nachstehen.

Abschnitte in Kapitel 3.2	Seite
3.2.1 Ablaufkonzept des V-Modells XT	68
3.2.2 Unvollständige Abläufe	70
3.2.3 Ablauf-Planung auf Basis des V-Modells	72
3.2.4 Zusammenfassung	73

3.2.1 Ablaufkonzept des V-Modells XT

Im Gegensatz zu seiner Vorgänger-Version definiert das V-Modell XT keine Abläufe in Form eines Aktivitätsflusses. Auch ein Produktfluss mit Eingangs- und Ausgangsprodukten wird nicht vorgegeben. Zielsetzung des V-Modells XT ist es, Abläufe flexibel zu gestalten beziehungsweise unterschiedliche Abläufe auf Basis derselben beschriebenen Aktivitäten zu ermöglichen. Das V-Modell XT bietet in Form so genannter Projektdurchführungsstrategien unterschiedliche Rahmen für die Planung auf grobgranularer Ebene und damit die Steuerung von Projekten.

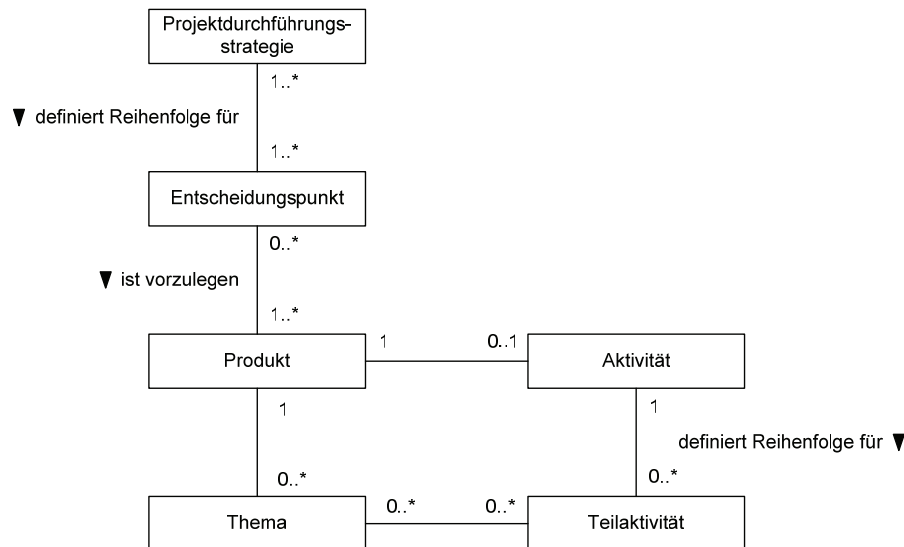


Abbildung 3–4: Ablaufkonzept des V-Modells XT

Wie in Abbildung 3–4 dargestellt, definieren Projektdurchführungsstrategien eine Reihenfolge der im Projekt zu durchlaufenden Entscheidungspunkte. Entscheidungspunkte stellen Meilensteine im Projekt dar, bei deren Erreichen der aktuelle Stand des Projektes überprüft wird. Für jeden Entscheidungspunkt definiert das V-Modell, welche Produktexemplare fertig zu stellen und vorzulegen sind. Die Modellierbarkeit von Reihenfolgen der Entscheidungspunkte einer Projektdurchführungsstrategie ist ebenfalls im Meta-Modell des V-Modells XT hinterlegt. Diesen Aspekt des Meta-Modells stellen wir in Abbildung 3–4 nicht dar, da er für die weiteren Betrachtungen nicht relevant ist.

Projektdurchführungsstrategien

Projektdurchführungsstrategien wie die inkrementelle Systementwicklung oder die agile Systementwicklung definieren eine Reihenfolge der zu durchlaufenden Entscheidungspunkte. Im Falle der inkrementellen Systemerstellung folgt beispielsweise auf den Entscheidungspunkt "Feinentwurf abgeschlossen" der Entscheidungspunkt "Systemelemente realisiert". Im Falle der agilen Systemerstellung ist die Reihenfolge dieser Entscheidungspunkte umgekehrt, dem Entwurf eines Systems geht also die Realisierung voraus. Mit der agilen Systemerstellung wird dabei das Ziel verfolgt, im Projekt keine Spezifikationen zu erstellen, die sich später eventuell als nicht realisierbar herausstellen.

Da für die Entscheidungspunkte abhängig von der im Tailoring gewählten Menge von Vorgehensbausteinen jeweils definiert ist, welche Produktexemplare vorzulegen sind, definiert eine Projektdurchführungsstrategie also auch eine partielle Reihenfolge der Produkterstellung. Entscheidungspunkte teilen den Projektverlauf in Projektabschnitte ein. Die Erstellung der Produkte der Entscheidungspunkte muss spätestens im jeweiligen dem Entscheidungspunkt vorgelagerten Projektabschnitt erfolgen. Da entsprechend dem Meta-Modell des V-Modells XT einem nicht als extern¹ markierten Produkt genau eine fertigstellende Aktivität zugeordnet sein muss, ist damit ebenfalls eine Reihenfolge der Aktivitäten gegeben.

Die Anwendung des Konzeptes der Projektdurchführungsstrategien mündet unmittelbar im Projektplan. Der Projektplan ist im V-Modell XT als Produkt modelliert, im Meta-Modell also nicht weiter präzisiert. Die im Rahmen der Planung relevanten Mechanismen des V-Modells

Planungsmechanismen

¹ Externe Produkte sind Produkte, die nicht im Rahmen eines V-Modell-Projektes erstellt werden, wie beispielsweise der Projektvorschlag, der vor dem eigentlichen Projektbeginn entsteht (vgl. dazu Kapitel 3.3.1).

XT sind im Text beschrieben. Die Planungsmechanismen beruhen zudem auf dem Konzept der erzeugenden Produktabhängigkeiten (vgl. dazu Kapitel 3.3.1).

In einem ersten Schritt müssen die Produkte der Entscheidungspunkte eingeplant werden. Dabei müssen ausgehend von den initialen Produkten entsprechend den Produktabhängigkeiten die erzeugten Produkte der Entscheidungspunkte ermittelt werden. In einem zweiten Schritt müssen die initialen und erzeugten Produkte ermittelt werden, die nicht in Entscheidungspunkten liegen. Projektdurchführungsstrategien und erzeugende Produktabhängigkeiten sind in der Darstellung des V-Modells XT in unterschiedlichen Teilen beschrieben. Eine integrierte Sicht beider Konzepte als Unterstützung für die Ableitung eines Projektplans existiert nicht. Die Durchführung eines entsprechenden Planungsvorgangs gestaltet sich ohne geeignete Werkzeugunterstützung in jedem Fall aufwendig. Der Projektassistent² zum V-Modell XT unterstützt lediglich die Planung eines Ablaufs in Form von Entscheidungspunkten, nicht aber die Erzeugung von Produkten.

Teilaktivitäten

Die einzelnen Schritte zur Erstellung der Produkte werden im V-Modell XT in Form von *Teilaktivitäten* beschrieben. Eine Teilaktivität darf nur *Themen* der *Produkte* bearbeiten, die die ihr zugeordnete *Aktivität* fertig stellt:

$\forall ta \in \text{Teilaktivität:}$
 $\text{thema} \in ta.\text{Thema} \Rightarrow \text{thema}.\text{Produkt} = ta.\text{Aktivität}.\text{Produkt}$

Aktivitäten können eine Ablaufreihenfolge für die ihnen zugeordneten Teilaktivitäten in Form eines Aktivitätsdiagramms der UML definieren (vgl. [OMG03a]). Beispielsweise wird für die Erstellung des Produktes Systemarchitektur eine Reihenfolge der durchzuführenden Teilaktivitäten definiert, die mit der Identifizierung der Architekturtreiber beginnt, mehrere Teilaktivitäten zur Erarbeitung der Architektur nebenläufig anordnet und mit der Teilaktivität zur Bewertung der Architektur entweder endet oder aber selbigen Prozess erneut durchläuft. Aktivitätsdiagramme und Teilaktivitäten beschreiben also methodische Aspekte der Erstellung von Produkten.

3.2.2 Unvollständige Abläufe

Aus den Konzepten des Meta-Modells für Ablaufbeschreibungen des V-Modells XT ergibt sich, dass Abläufe nicht vollständig beschrieben werden können. Dies zeigen wir anhand der Konzepte der Projektdurchführungsstrategien und der Aktivitätsdiagramme.

Projektdurchführungsstrategien

Produkte des selben Entscheidungspunktes

Projektdurchführungsstrategien definieren Reihenfolge-Beziehungen zwischen Entscheidungspunkten und damit zwischen den in diesen Entscheidungspunkten jeweils enthaltenen Produkten. Abläufe zwischen den Produkten desselben Entscheidungspunktes, die also im selben Projektabschnitt erstellt werden, sind nicht beschreibbar. Beispielsweise kann dem Ablaufkonzept entsprechend im V-Modell XT nicht modelliert werden, dass das Produkt Anforderungsbewertung erst aufbauend auf einem zumindest partiell erstellten Produkt Anforderungen erstellt werden soll.

² Siehe <http://www.sourceforge.net/fourever>

Weiterhin sind Produkte, die den Entscheidungspunkten nicht zugeordnet sind, dem Ablaufkonzept des V-Modells auch nicht zugänglich. Zielsetzung des Entscheidungskonzeptes ist aber, selbigen nur Produkte zuzuordnen, die dem Lenkungsausschuss vorzulegen sind. Beispielsweise sind die Produkte SW-Komponente oder Schätzung dem Ablaufkonzept nicht zugänglich, da diese Produkte nicht im Rahmen eines Entscheidungspunktes vorgelegt werden sollen. Die Zuordnung dieser Produkte zu einem Projektabschnitt bleibt also unklar.

Produkte außerhalb von Entscheidungspunkten

Ein Anhaltspunkt für einen Ablauf auf Produktebene kann im Konzept der erzeugenden Produktabhängigkeiten gesehen werden. Eine erzeugende Produktabhängigkeit beschreibt basierend auf einem Ausgangsprodukt eine Bedingung, bei deren Eintreten ein Zielprodukt im Projekt erzeugt werden muss. Dabei handelt es sich um eine gerichtete Abhängigkeit zwischen Ausgangs- und Zielprodukt (vgl. dazu Kapitel 3.3.1).

Erzeugende Produktabhängigkeiten und Abläufe

Beispielsweise ist die Produktabhängigkeit "Produktumfang einer SW-Einheit im System" eine gerichtete Abhängigkeit vom Produkt SW-Architektur zum Produkt SW-Einheit. Inhaltlich besagt diese Produktabhängigkeit, dass in der SW-Architektur die Existenz von SW-Einheiten spezifiziert wird.

Soll im Projekt ein Top-Down-Vorgehen, wie in der Projektdurchführungsstrategie inkrementelle Systementwicklung beschrieben, angewendet werden, impliziert diese Produktabhängigkeit eine Reihenfolge der jeweiligen Aktivitäten. Die Erstellung der SW-Architektur erfolgt damit vor der Erstellung der SW-Einheit. Bestandteil des V-Modells XT sind allerdings auch Bottom-Up-Vorgehensweisen wie die agile Systementwicklung. Eine SW-Architektur kann in diesem Fall auch nach der Realisierung der SW-Einheiten erstellt werden. Vor dem Hintergrund eines Bottom-Up-Vorgehens lässt sich aus der Existenz einer erzeugenden Produktabhängigkeit also keine Reihenfolge der Produkterstellung ableiten. Erzeugende Produktabhängigkeiten haben also auf semantischer Ebene keinen Bezug zum Ablaufkonzept des V-Modells XT.

Aktivitätsdiagramme

Ablaufreihenfolgen auf Ebene der Aktivitäten und Produkte sind also nur unvollständig beschreibbar. Entsprechendes gilt auch für die Ebene der Teilaktivitäten beziehungsweise Themen. Eine Teilaktivität darf nur Themen der Produkte bearbeiten, die die ihr zugeordnete Aktivität fertig stellt (vgl. Kapitel 3.2.1). Es können also keine Reihenfolge-Beziehungen für Teilaktivitäten angegeben werden, die unterschiedlichen Aktivitäten zugeordnet sind.

Teilaktivitäten unterschiedlicher Aktivitäten

Beispielsweise sind im V-Modell XT die Aspekte Dekomposition und Schnittstellenbeschreibung einer Software in getrennten Produkten modelliert. Die Beschreibung der Dekomposition ist im Produkt SW-Architektur, die Schnittstellenbeschreibung im Produkt SW-Spezifikation enthalten. Die Teilaktivitäten, die die Erstellung dieser Produkte beschreiben, sind daher unterschiedlichen Aktivitäten zugeordnet. Ein Ablauf, der die Teilaktivitäten für die Verfeinerung der Schnittstellen, die Abbildung der Schnittstellen auf untergeordnete Systemelemente und die Dekomposition in untergeordnete Systemelemente, in Beziehung setzt, wäre sinnvoll. Eine Herangehensweise zur Identifikation der Dekomposition in Systemelemente wäre in der Verfeinerung der Schnittstellen der Systemelemente zu sehen. Die Abläufe dazu sollten also geeignet ineinander verwoben sein. Im V-Modell XT sind entsprechende verzahnte Abläufe allerdings nicht modellierbar.

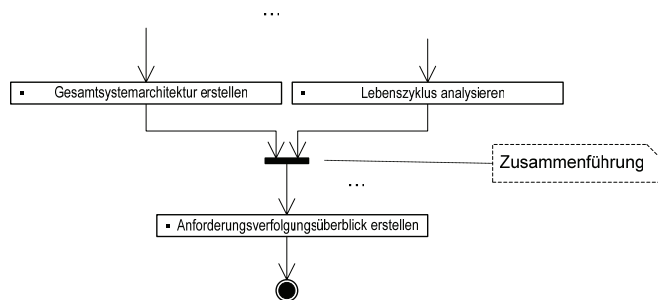
Das Problem der unvollständigen Modellierbarkeit von Abläufen führt bei der Ableitung eines Projektplans zu teilweise unerwünschten Freiheitsgraden. Der Anwendungsfall Planung ist auf Basis des Meta-Modells des V-Modells XT also nur ansatzweise durch ein Werkzeug unterstützbar.

3.2.3 Ablauf-Planung auf Basis des V-Modells

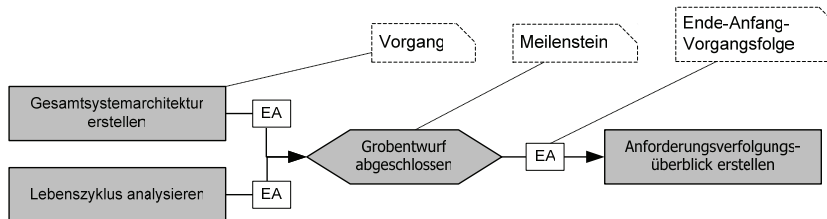
Ausdrucks- mächtigkeit von Aktivi- tätsdiag- rammen

Für die detaillierte Planung der Produkterstellung bietet sich im V-Modell XT die Ebene der Teilaktivitäten an. Teilaktivitäten sind im V-Modell als methodische Arbeitsanleitungen für die Erstellung eines Produktes zu verstehen. Der Ablauf der Teilaktivitäten einer Aktivität kann im V-Modell durch ein Aktivitätsdiagramm der UML beschrieben werden. Die Planung der Produkterstellung erfordert also eine Umsetzung von Aktivitätsdiagrammen in einen Netzplan mit Vorgängen, Meilensteinen und Vorgangsfolgen (vgl. Kapitel 2.4).

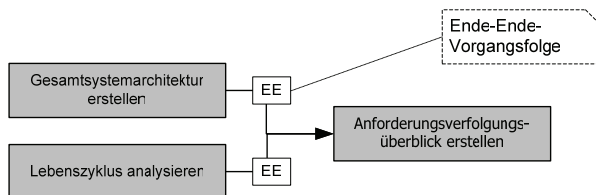
(a) UML-Aktivitätsdiagramm (Beispiel aus V-Modell XT)



(b) Direkte Umsetzung von (a) als Netzplan



(c) Besserer Ablauf mit Nebenläufigkeit als Netzplan



(d) UML-Aktivitätsdiagramm mit Pufferzeiten, äquivalent zu (c)

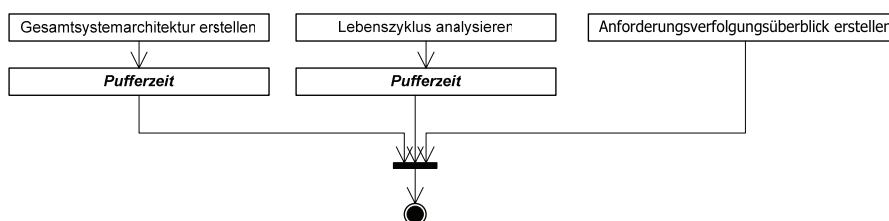


Abbildung 3–5: UML-Aktivitätsdiagramme und Netzplantechnik

Ein Beispiel eines Aktivitätsdiagramms zur Aktivität "Gesamtsystemspezifikation erstellen" des V-Modells XT ist in Abbildung 3–5 (a) dargestellt. Die Umsetzung in einen Netzplan kann unmittelbar durch die Planung der Teilaktivitäten in Form von Vorgängen, die Umsetzung des Kontrollflusses in Ende-Anfang-Vorgangfolgen sowie die Umsetzung von Aufspaltungen und Zusammenführungen in Meilensteine erfolgen. Zusammenführungen im Aktivitätsdiagramm folgen der Semantik, dass die ausgehende Teilaktivität erst angestoßen wird, wenn alle eingehenden Teilaktivitäten abgeschlossen sind (vgl. [OMG03a]).

Abbildung 3–5 (b) zeigt die direkte Umsetzung des Beispiels aus Abbildung 3–5 (a) in einen Netzplan. Die Positionierung der Teilaktivität "Anforderungsverfolgungsüberblick erstellen" in Abbildung 3–5 (a) im Kontrollfluss am Ende spiegelt die Notwendigkeit wider, Anforderungen auf Basis einer abgeschlossenen Version der Gesamtsystemarchitektur zuzuordnen. Damit wird die Konsistenz der Gesamtsystemarchitektur und der Anforderungsverfolgung gewährleistet. Der Beginn des Vorgangs der Anforderungszuordnung sollte allerdings nicht nach Abschluss der Erstellung der Architektur erfolgen, sondern nebenläufig dazu. Der frühe Start der Anforderungsverfolgung wäre sinnvoll, da eine Nicht-Zuordenbarkeit einer Anforderung eine Überarbeitung der Architektur nach sich ziehen muss. Die Umsetzung des Ablaufs in Abbildung 3–5 (b) erscheint also nicht unmittelbar günstig.

Sinnvoll wäre eine nebenläufige Planung beider Vorgänge mit der Randbedingung, dass die Zuordnung der Anforderungen erst abgeschlossen werden kann, nachdem die Erstellung der Architektur abgeschlossen ist. In der Netzplantechnik lässt sich dieser Zusammenhang mit Hilfe von Ende-Ende-Vorgangfolgen ausdrücken. Ein entsprechender Netzplan ist in Abbildung 3–5 (c) dargestellt.

In einem Aktivitätsdiagramm stehen keine Sprachmittel zur Verfügung, die eine Beschreibung sich überlappenden Vorgänge erlauben. UML-Aktivitätsdiagramme sind also für die Beschreibung der Abläufe in einem Projektplan zwar geeignet, ihre Ausdrucksmächtigkeit ist allerdings im Hinblick auf die in der Netzplantechnik beschreibbare Nebenläufigkeit von Vorgängen sehr eingeschränkt.

Theoretisch möglich wäre die Modellierung von Zeitpuffern als Elementen eines Aktivitätsdiagramms, wie in Abbildung 3–5 (d) dargestellt. Dies erscheint uns allerdings als unpraktische und nicht intendierte Verwendung der Notation. Für jede gewünschte Ende-Ende-Vorgangfolge müsste eine eigene Klammerstruktur mit Aufspaltung und Zusammenführung aufgebaut werden.

3.2.4 Zusammenfassung

Das Ablaufkonzept des V-Modells XT stellt sich an einigen Stellen als unvorteilhaft für die einfache Anwendbarkeit des V-Modells im Rahmen der Projektplanung und die Unterstützbarkeit durch ein Planungswerkzeug heraus.

Nicht alle Arten von Abläufen können mit dem Meta-Modell des V-Modells XT beschrieben werden. Produkte, die demselben Entscheidungspunkt zugeordnet sind, Produkte, die keinem Entscheidungspunkt zugeordnet sind, sowie Teilaktivitäten, die unterschiedlichen Aktivitäten zugeordnet sind, sind dem Ablaufkonzept des V-Modells nicht zugänglich. Wie in den vorangehenden Abschnitten am Beispiel gezeigt, wären entsprechende Abläufe wünschenswert. Abläufe auf Ebene der Teilaktivitäten werden im V-Modell XT in Form von Aktivitätsdiagrammen der UML beschrieben. Aktivitätsdiagramme erweisen sich allerdings als weniger ausdrucksmächtig als ihr Abbild im Projektplan, die Netzplantechnik, und damit nur als bedingt geeignet für die Umsetzung in einen Projektplan.

3.3 Produkte

Wir wenden uns dem Konzept des Produktes im V-Modell XT zu. Produkte sind Beschreibungen der Ergebnisse, die in einem Projekt erstellt werden sollen. Nach der Erläuterung des Produktmodellkonzeptes des V-Modells XT gehen wir auf diverse Probleme ein, die sich aus diesem Konzept ergeben.

Produktabhängigkeiten bewegen sich im V-Modell XT auf einer sehr abstrakten Ebene, präzise Zusammenhänge der Produkte werden nicht formuliert. Mangels Ausdrucksmächtigkeit des Meta-Modells setzen Produktabhängigkeiten an einigen Stellen im V-Modell XT Produkte nur indirekt in Beziehung. Die fehlende Möglichkeit einer Formalisierung bestimmter Produktinhalte führt zu weiteren Problemen im Kontext von Werkzeugunterstützung und Verbindlichkeit des V-Modells XT.

Abschnitte in Kapitel 3.3		Seite
3.3.1	Produktmodellkonzept des V-Modells XT	74
3.3.2	Abstrakte Produktabhängigkeiten	76
3.3.3	Indirekte Produktabhängigkeiten	78
3.3.4	Nicht-Formalisierte Produktinhalte	78
3.3.5	Zusammenfassung	79

3.3.1 Produktmodellkonzept des V-Modells XT

Unter dem Begriff Produktmodell fassen wir die Konzepte Produktgruppe, Produkt, Thema und Produktabhängigkeit zusammen. Abbildung 3–6 zeigt diesen Ausschnitt des Meta-Modells des V-Modells XT.

Produktgruppen sind hierarchische disjunkte Gruppierungen von Produkten nach inhaltlichen Gesichtspunkten. *Produkte* sind Beschreibungen der Ergebnisse, die in einem Projekt erstellt werden sollen. *Themen* stellen inhaltliche Aspekte eines Produktes dar. Handelt es sich bei einem Produkt beispielsweise um ein Dokument, ist ein Thema als ein Kapitel dieses Dokumentes zu sehen. *Produktabhängigkeiten* beschreiben inhaltliche Abhängigkeiten zwischen zwei oder mehr Produkten, sind also als Konsistenzbedingungen zu verstehen.

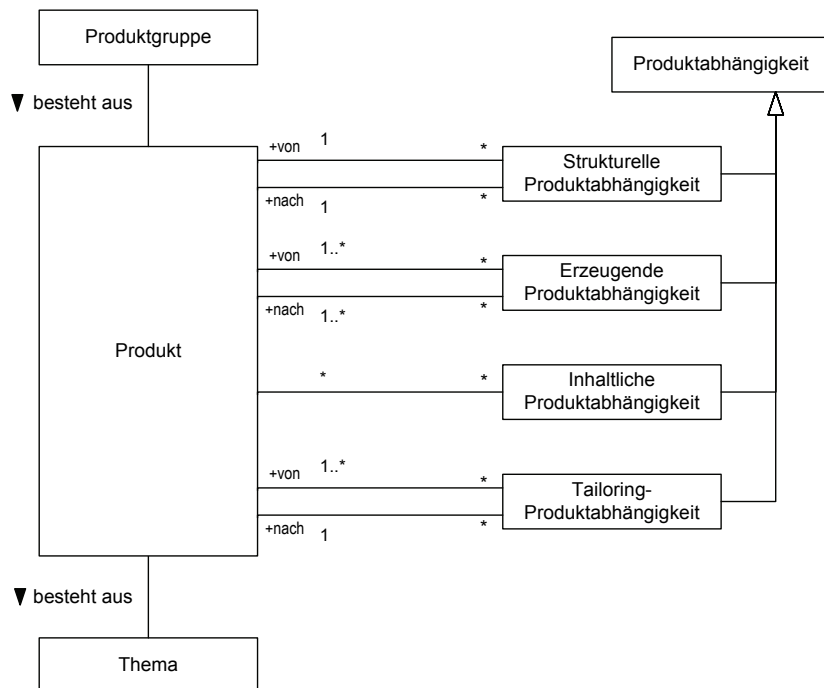


Abbildung 3–6: Produktmodellkonzept des V-Modells XT

Das V-Modell XT unterscheidet, wie in Abbildung 3–6 dargestellt, vier Arten von Produktabhängigkeiten. Das Konzept der Produktabhängigkeits-Erweiterungen stellen wir in Abbildung 3–6 nicht dar (vgl. Kapitel 3.1.1). *Strukturelle Produktabhängigkeiten* dienen im V-Modell XT einzig der Beschreibung einer hierarchischen, logischen Strukturierung des Systems in Form von Systemelementen. Eine allgemeine, präzise Definition zum Begriff der strukturellen Produktabhängigkeit liefert das V-Modell XT nicht. Wir verstehen strukturelle Produktabhängigkeiten als Erweiterung des dreistufig hierarchischen Produktmodellkonzeptes bestehend aus Produktgruppen, Produkte und Themen um ein Beschreibungsmittel für die beliebig tiefe hierarchische Strukturierung von Produkten.

Erzeugende Produktabhängigkeiten sind gerichtete Assoziationen, die für ein oder mehrere Ausgangsprodukte eine Bedingung in Form von Text aufstellen, deren Inkrafttreten die Erzeugung ein oder mehrerer Zielprodukte im Projekt nach sich ziehen muss. Erzeugende Produktabhängigkeiten sind also Regeln für die projektspezifische Instanziierung von Produkten. Unter Instanziierung verstehen wir die Ableitung von Produktexemplaren für ein Projekt aus den Produkten, die das V-Modell XT beschreibt.

Beispielsweise legt das Produkt Systemarchitektur fest, welche SW-Einheiten zu spezifizieren sind. Die SW-Einheit ist ebenfalls als Produkt modelliert. Das V-Modell XT enthält also eine erzeugende Produktabhängigkeit von der Systemarchitektur zur SW-Einheit. Ist die Systemarchitektur im Projekt erarbeitet, müssen nach der Regel dieser Produktabhängigkeit die spezifizierten SW-Einheiten in Form von Produktexemplaren angelegt werden.

Inhaltliche Produktabhängigkeiten beschreiben Zusammenhänge der Inhalte mehrerer Produkte. Anders als erzeugende Produktabhängigkeiten sind inhaltliche Produktabhängigkeiten ungerichtet. Beispielsweise beschreibt die inhaltliche Produktabhängigkeit „Einfluss der Altssystemanalyse auf die Systemerstellung“ den inhaltlichen Zusammenhang der Produkte Altssystemanalyse, Gesamtsystemspezifikation und Systemarchitektur. Erzeugende Produktabhängigkeiten werden im V-Modell zugleich als inhaltliche Produktabhängigkeiten verstan-

den, da bei einer erzeugenden Produktabhängigkeit im Allgemeinen auch die Inhalte der jeweiligen Produkte in Beziehung gesetzt werden.

Tailoring-Produktabhängigkeiten schließlich stellen Bedingungen dar, die zu einer Überarbeitung der projektspezifischen Auswahl von Vorgehensbausteinen führen. Tailoring-Produktabhängigkeiten sind stets, ausgehend von einem oder mehreren Produkten, auf das Produkt Projekthandbuch gerichtet, da dieses das Thema „Projektspezifisches V-Modell“ und damit die Auswahl der Vorgehensbausteine für ein Projekt beinhaltet.

Beispielsweise dokumentiert die Tailoring-Produktabhängigkeit „Vorgaben der Systemarchitektur“ den Einfluss des Produktes Systemarchitektur auf das projektspezifische V-Modell. Wurde eine SW-Einheit in der Systemarchitektur identifiziert, muss der Vorgehensbaustein SW-Entwicklung Bestandteil des projektspezifischen V-Modells sein.

V-Modell 97 im Vergleich

Der Produktfluss des V-Modells 97 formuliert die Zusammenhänge der Produkte in Form von Eingangs- und Ausgangsprodukten für jede Aktivität (vgl. [Dröschel00]). Die Festlegung von Eingangs- und Ausgangsprodukten impliziert im V-Modell 97 einen Aktivitätsfluss und damit genau einen möglichen Ablauf. Das Konzept eines Produktmodells mit explizit und präzise formulierten Produktabhängigkeiten ist als komplexer als das Ablaufkonzept des V-Modells 97 einzustufen. Der Anspruch bestünde in der annähernd vollständigen Formulierung inhaltlicher Zusammenhänge der Produkte. Diese Zusammenhänge sind allerdings in der Praxis sehr mannigfaltig.

Um Produktabhängigkeiten für den Leser des V-Modells XT übersichtlich zu gestalten und den Arbeitsaufwand für Erstellung und Weiterentwicklung zu begrenzen, abstrahiert das V-Modell XT von der präzisen Formulierung von Produktabhängigkeiten. Diese Abstraktionsprinzipien stellen wir zusammen mit den daraus resultierenden Problemen anhand von Beispielen dar. Abbildung 3–7 zeigt ein UML-Instanzdiagramm zu den Schwachstellen des V-Modells XT im Hinblick auf das Produktmodellkonzept. In den folgenden Kapiteln gehen wir auf die dargestellten Probleme ein.

3.3.2 Abstrakte Produktabhängigkeiten

Das Konzept der Produktabhängigkeiten des V-Modells XT erlaubt keine Definition präziser Zusammenhänge der Produkte, sondern ist vielmehr als Abstraktion der präzisen Zusammenhänge zu sehen.

Keine Abhängigkeiten auf Themen- Ebene

Produktabhängigkeiten beziehen sich allein auf die Ebene der Produkte. Abhängigkeiten zwischen den Themen der Produkte sind damit nur im beschreibenden Text einer Produktabhängigkeit formulierbar. Eine Produktabhängigkeit des V-Modells XT kann also als eine Aggregation feingranularer Abhängigkeiten auf Ebene der Themen gesehen werden.

Wie in Abbildung 3–7 (a) dargestellt, setzt beispielsweise eine inhaltliche Produktabhängigkeit die Produkte *QS-Handbuch* und *„Prüfspezifikation Systemelement“* in Beziehung. Entsprechend den Vereinbarungen des QS-Handbuchs soll dabei für Fertigprodukte eine Prüfung erfolgen. Dem beschreibenden Text der Produktabhängigkeit zufolge bezieht sich diese allerdings auf genau ein Thema des QS-Handbuchs, nämlich das Thema *„Vorgaben für die Prüfspezifikation von Fertigprodukten“*.

n-stellige Abhängig- keiten

Zudem handelt es sich bei erzeugenden, inhaltlichen und Tailoring-Produktabhängigkeiten um n-stellige Abhängigkeiten. Diese sind im V-Modell XT jeweils als Aggregationen mehrerer binärer Produktabhängigkeiten zu verstehen.

Wie in Abbildung 3–7 (a) dargestellt, setzt beispielsweise die inhaltliche Produktabhängigkeit

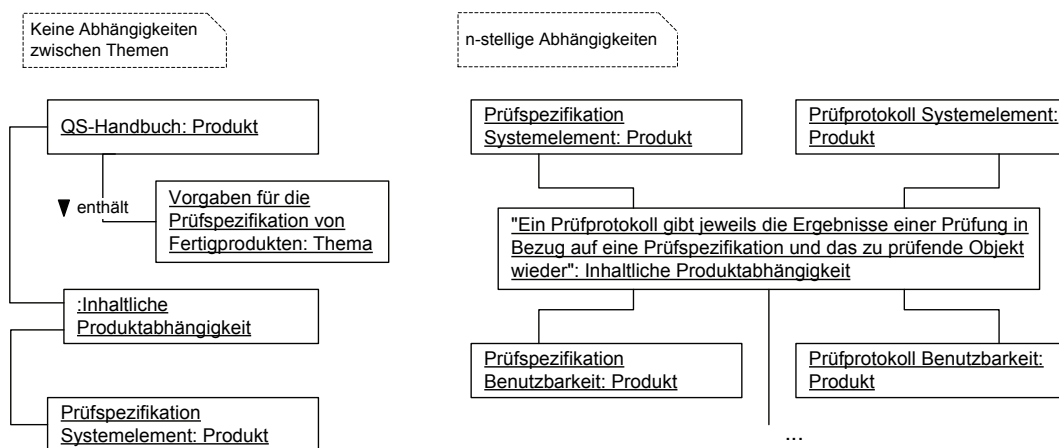
„Ein Prüfprotokoll gibt jeweils die Ergebnisse einer Prüfung in Bezug auf eine Prüfspezifikation und das zu prüfende Objekt wider“

alle Arten von Prüfprotokollen im V-Modell in Beziehung, wie *„Prüfprotokoll Systemelement“* und *„Prüfprotokoll Benutzbarkeit“*, mit den entsprechenden Prüfspezifikationen, wie *„Prüfspezifikation Systemelement“* und *„Prüfspezifikation Benutzbarkeit“*. Diese Art der Modellie-

ung stellt eine Abstraktion vom genauen Zusammenhang eines Produktes „Prüfprotokoll Systemelement“ mit einer „Prüfspezifikation Systemelement“ dar.

Für den Leser des V-Modells XT ist die Aggregation von Produktabhängigkeiten als Abstraktion der genauen Zusammenhänge nicht problematisch, sondern vielmehr dem schnellen Verständnis zuträglich. Aus Sicht einer Werkzeugunterstützung sind abstrakte n-stellige Produktabhängigkeiten allerdings problematisch. Die Zusammengehörigkeit der beiden Produkte „Prüfprotokoll Systemelement“ und „Prüfspezifikation Systemelement“ ist allein über die Korrespondenz der Namen ersichtlich und damit einer Werkzeugunterstützung nicht zugänglich.

(a) Abstrakte Produktabhängigkeiten im V-Modell



(b) Indirekte Produktabhängigkeiten im V-Modell

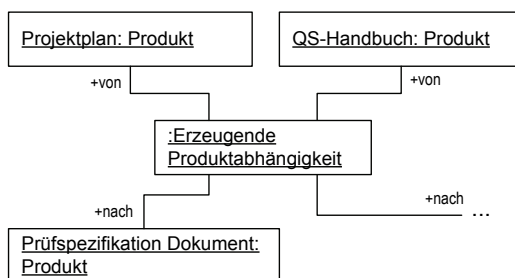


Abbildung 3–7: Beispiele zu Schwachstellen des V-Modells XT: Produktmodellkonzept (Instanzdiagramm)

Das Produktmodell des V-Modells XT dient als Vorgabe für die projektspezifische Produktstruktur. Wird im V-Modell XT ein Produkt – genauer ein Produkttyp – beschrieben, müssen im Projekt ein oder mehrere entsprechende Produktexemplare erstellt werden. Aus Sicht einer Werkzeugunterstützung für die Ableitung eines Projektplans ist von Bedeutung, ob beziehungsweise wie viele Produktexemplare jeweils zu erstellen sind oder welche Parameter zu einer Instanzierung eines Produktes führen.

Entsprechend einer Modellierungssprache wie der UML würde sich hierfür die Modellierung von Multiplizitäten und Constraints anbieten. Unter einer Multiplizität verstehen wir Ein-

Einschränkungen der projektspezifischen Produktstruktur

schränkungen der Vielfachheit der Instanzen einer Klasse. Insbesondere für erzeugende Produktabhängigkeiten im V-Modell XT wäre die Festlegung von Multiplizitäten sinnvoll. Beispielsweise muss es zu jedem „Prüfprotokoll Systemelement“ genau eine „Prüfspezifikation Systemelement“ geben. Dieser Zusammenhang, der immer genau ein Produkt mit genau einem anderen Produkt in Beziehung setzt, könnte in einer Werkzeugunterstützung für die Instanziierung des Produktmodells ausgenutzt werden. Das Produktmodellkonzept des V-Modells XT sieht keine präzise Beschreibung von Einschränkungen der projektspezifischen Produktstruktur vor, um das Modell für den Prozessingenieur einfacher handhabbar zu machen.

3.3.3 Indirekte Produktabhängigkeiten

Generische Vorgehensweisen führen im V-Modell XT zu Produktabhängigkeiten, die die jeweiligen Produkte nur indirekt in Beziehung setzen.

Generische Vorgehensweisen

Das V-Modell XT verfügt nicht über Beschreibungsmittel für generische Vorgehensweisen, das heißt beispielsweise für Aussagen bezüglich aller Produkte. Generische Vorgehensweisen sind im V-Modell XT beispielsweise im Kontext von Qualitätssicherung oder Konfigurationsmanagement zu beschreiben. Das V-Modell XT behilft sich hierbei mit der Formulierung indirekter Produktabhängigkeiten.

Beispielsweise wird eine „*Prüfspezifikation Dokument*“ im V-Modell ausgehend von den Produkten *Projektplan* und *QS-Handbuch* erzeugt, wie in Abbildung 3–7 (b) dargestellt. Die durch diese erzeugende Produktabhängigkeit formulierte Regel besagt, dass für alle im Projektplan eingeplanten Produktexemplare, die einem Dokument entsprechen und im QS-Handbuch als relevant für die Qualitätssicherung vereinbart sind, im Projekt ein Produktexemplar „*Prüfspezifikation Dokument*“ erzeugt werden muss. Diese Regel ist in Form von Text formuliert.

Die Zusammengehörigkeit des zu prüfenden Dokumentes und der jeweiligen „*Prüfspezifikation Dokument*“ ist im V-Modell durch diese Produktabhängigkeit auf formaler Ebene nicht klar und einer Werkzeugunterstützung damit nicht zugänglich. In einem Projektplan wäre allerdings eine entsprechende Kennzeichnung der Zusammengehörigkeit beider Produkte wünschenswert. Beispielsweise sollte in einem abgeleiteten Projektplan über Vorgangsfolgen sichergestellt werden, dass die *Prüfspezifikation* eines Dokumentes vor der eigentlichen Erstellung des Dokumentes zur Verfügung steht.

3.3.4 Nicht-Formalisierte Produktinhalte

Das V-Modell XT beschreibt die Inhalte der Produkte in Form von Text. Für Produkte wird ihr Sinn und Zweck angegeben, sowie die Themen, aus denen ein Produkt besteht. Eine weitergehende Formalisierung der Produktinhalte ist nicht vorgesehen. Aus Sicht einer Werkzeugunterstützung des Planungsvorgangs wäre eine explizite Formulierung der Parameter wünschenswert, die für die Ableitung eines Projektplans aus dem V-Modell XT eingestellt werden müssten.

Beispielsweise wird die Erzeugung eines Exemplars des Produktes SW-Einheit durch die Vorgaben gesteuert, die im Produkt SW-Architektur bezüglich der Aufteilung des Systems in SW-Einheiten gemacht werden. Eine Formalisierung des Inhaltes des Produktes SW-Architektur würde diesen Erzeugungs-Zusammenhang einem Planungswerkzeug zugänglich machen.

Eine für den Zweck der Ableitung eines Projektplans hinreichende Formalisierung der SW-Architektur könnte dabei aus der Festlegung der Namen der SW-Einheiten sowie deren Abhängigkeiten bestehen. Die Spezifikation der SW-Einheiten in der SW-Architektur impliziert dann die projektspezifische Existenz eines Produktexemplars SW-Einheit. Spezifizierte

Abhängigkeiten der SW-Einheiten implizieren bestimmte Vorgangsfolgen im Projektplan. Beispielsweise impliziert die Verwendung einer Software-Einheit B durch eine andere SW-Einheit A zeitlich gesehen die Fertigstellung der SW-Einheit B vor dem abschließenden Test der SW-Einheit A. Erst nach Fertigstellung der verwendeten SW-Einheiten B kann die SW-Einheit A einem abschließenden Test unterzogen werden (vgl. [GDM+04]).

Im Beispiel der SW-Architektur wäre aus Sicht einer Werkzeugunterstützung des Planungsvorgangs diese Formalisierung des Produktinhalts zufrieden stellend. Eine weitergehende Formalisierung könnte beispielsweise der Generierung strukturierter Produktvorlagen dienen.

3.3.5 Zusammenfassung

Das V-Modell XT verfolgt die Idee eines Produktmodells mit explizit formulierten Produktabhängigkeiten. Im Vergleich zu einem Produktfluss wie im V-Modell 97 stellt dies aufgrund der höheren Flexibilität einen Fortschritt dar. Das V-Modell XT formuliert Produktabhängigkeiten allerdings nicht präzise, sondern entsprechend einer Reihe von Abstraktionsprinzipien.

Produktabhängigkeiten im V-Modell XT können als Aggregationen der genauen Zusammenhänge der Produkte verstanden werden. Zum einen werden Produktabhängigkeiten vereinfachend auf Ebene der Produkte formuliert, die beschreibenden Texte beziehen sich aber häufig auf die Ebene der Themen. Zum anderen sind Produktabhängigkeiten als n-stellige Assoziationen definiert. Diese können als Aggregationen der präzisen binären Zusammenhänge verstanden werden. Einschränkungen der projektspezifischen Produktstruktur in Form von Multiplizitäten können mit den Beschreibungsmitteln des V-Modells nicht modelliert werden. Zudem werden bestimmte Produktabhängigkeiten indirekt formuliert, setzen also die betroffenen Produkte nicht unmittelbar in Beziehung. Eine Festlegung der Inhalte eines Produktes, die über die Festlegung der Themen eines Produktes hinausgeht und relevant für die Planung wäre, ist mit den Beschreibungsmitteln des V-Modells XT nicht modellierbar.

Für den Leser des V-Modells XT erscheinen diese Abstraktionen angebracht, da sie der Übersichtlichkeit des Modells dienen. Aus Sicht einer Werkzeugunterstützung für den Planungsvorgang ist die geringe Präzision des Produktmodells nicht hilfreich und setzt den Möglichkeiten einer Werkzeugunterstützung enge Grenzen.

Aus Anwendersicht ist das Fehlen einer weitergehenden Formalisierung des Produktmodells problematisch, da das V-Modell damit an Verbindlichkeit verliert. Gemäß seiner Zielsetzung ist das V-Modell XT ein einheitlicher Standard für die Projektdurchführung, der als verbindliche Vorgabe gelten kann. Die projektspezifisch ausgewählten Vorgehensbausteine werden daher mit dem Auftraggeber abgestimmt. Das Konzept der erzeugenden Produktabhängigkeiten läuft dem entgegen, da nicht alle in der Menge der gewählten Vorgehensbausteine enthaltenen Produkte in Form von Produktexemplaren erzeugt werden müssen. Zudem sind die Regeln für die Erzeugung nicht präzise formuliert und oft auch im Text nicht klar hinterlegt.

Eine Überprüfung der Menge der Produktexemplare eines Projektes durch den Auftraggeber gestaltet sich schwierig, da zu diesem Zweck die Texte der erzeugenden Produktabhängigkeiten interpretiert werden müssen. Eine Automatisierung ist derzeit nicht vorstellbar. Zudem sind die textuellen Beschreibungen der meisten Produktabhängigkeiten sehr abstrakt, so dass sich eine „manuelle“ Überprüfung ebenfalls schwierig gestaltet. Eine Formalisierung der Parameter, die den Aufbau der projektspezifischen Produktstruktur bestimmen, wäre wünschenswert. Damit wären eine einfache Anwendbarkeit des V-Modells sowie eine einfache Überprüfbarkeit der Anwendung gegeben.

Verbindlichkeit des V-Modells XT

3.4 Abstraktion im V-Modell

Das Kapitel beschäftigt sich mit dem Thema Abstraktion im V-Modell XT. Dabei gehen wir auf den generellen Abstraktionsgrad des V-Modells ein. Anhand der Beschreibungsmittel wie beispielsweise dem Konzept der Produkt-Aktivitäts-Schnittstelle kann der Abstraktionsgrad des V-Modells XT abgelesen werden. Das V-Modell XT vereinigt unterschiedliche Abstraktionsebenen in sich. Aufgrund fehlender Beschreibungsmittel zur Modellierung von Abstraktionsebenen sind diese nicht explizit ersichtlich, sondern vielmehr implizit vermischt. Dieses Problem verdeutlichen wir am Beispiel der Systemerstellung, deren Umsetzung im V-Modell XT aufgrund mangelnder Abstraktion einige Redundanzen aufweist.

Abschnitte in Kapitel 3.4	Seite
3.4.1 Abstraktionsgrad des V-Modells XT	80
3.4.2 Produkt-Aktivitäts-Schnittstelle	80
3.4.3 Abstraktionsebenen im V-Modell XT	82
3.4.4 Systemerstellung im V-Modell XT	84
3.4.5 Zusammenfassung	85

3.4.1 Abstraktionsgrad des V-Modells XT

Das V-Modell XT ist im Vergleich zum tatsächlichen Ablauf in einem konkreten Projekt als sehr abstrakt einzustufen (vgl. Kapitel 2.1.5). Dies resultiert aus dem Ziel des V-Modells XT, als standardisierte Vorgehensweise für alle Arten von Systementwicklungsprojekten im IT-Umfeld zu gelten.

Vorteile dieses Abstraktionsgrades sind in der breiten Anwendbarkeit zu sehen, sowie im relativ großen Gestaltungsspielraum eines V-Modell konformen Projektes und nicht zuletzt im vergleichsweise einfachen Erstellungsprozess des V-Modells XT selbst. Von Nachteil ist die relativ unkonkrete Hilfestellung, die das V-Modell einem Projekt bietet. Die Inhalte der Produktexemplare werden je nach Projekt sehr unterschiedlich ausfallen, da das V-Modell lediglich eine sehr allgemeine Beschreibung in Form des „kleinsten gemeinsamen Nenners“ bietet. Methodische Aspekte der Entwicklung, wie beispielsweise der objektorientierte Entwurf von Software, werden im V-Modell XT nicht thematisiert. Vielmehr ist das V-Modell als ein Rahmenwerk für die Projektdurchführung zu sehen.

Der Abstraktionsgrad des V-Modells lässt sich zudem bereits am Konzept des Produktes ablesen. Produkte sind definiert als die in einem Projekt zu erstellenden Ergebnisse. Eine Klassifizierung der Produkte in unterschiedliche Arten ist nicht vorgesehen. Die Bandbreite der möglichen Produktexemplare reicht also beispielsweise von Dokumenten, HW-Teilen und Programmcode bis hin zu installierten Werkzeugen.

3.4.2 Produkt-Aktivitäts-Schnittstelle

Die Produkt-Aktivitäts-Schnittstelle spiegelt ebenfalls den Abstraktionsgrad des V-Modells XT wider. Abbildung 3–8 zeigt die Konzepte *Aktivität*, *Produkt* und *Rolle*, die den meisten Vorgehensmodellen gemein sind (vgl. Kapitel 2.2), im V-Modell XT. Im V-Modell XT steht dabei das Konzept des Produktes als Beschreibung eines Projektergebnisses im Mittelpunkt.

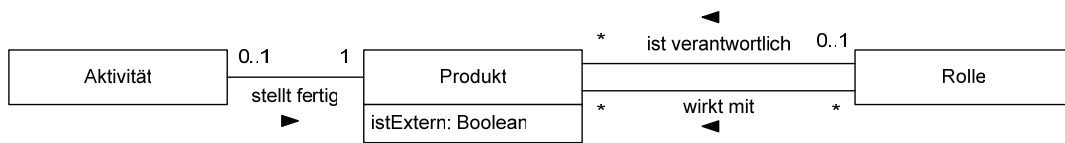


Abbildung 3–8: Die Konzepte Aktivität, Produkt und Rolle im V-Modell XT

Einem Produkt, das nicht als *extern* markiert ist, wird im V-Modell XT genau eine Aktivität, die die Erstellung dieses Produktes beschreibt, und genau eine Rolle, die für die Erstellung des Produktes verantwortlich ist, zugeordnet:

$$\forall p \in \text{Produkt}: p.\text{istExtern} \neq \text{True} \Rightarrow |p.\text{Aktivität}| = 1 \wedge |p.\text{Rolle}| = 1$$

Die Mitwirkung einer Rolle an der Erstellung von Produkten unterliegt keinen einschränkenden Multiplizitäten.

Ein Produkt, das jedoch als *extern* markiert ist, wird nicht im Rahmen eines V-Modell-Projektes erstellt. Externe Produkte wie der Projektvorschlag entstehen beispielsweise vor dem eigentlichen Projektbeginn. Externen Produkten muss keine fertigstellende Aktivität und auch keine verantwortliche Rolle zugeordnet werden.

Produkte, Aktivitäten und Rollen bewegen sich also im V-Modell auf derselben Abstraktionsebene.

Die Aufteilung in die Beschreibung von Produkten einerseits und Aktivitäten andererseits erscheint aufgrund der Zuordnung genau einer Aktivität zu genau einem Produkt nicht sinnvoll. Unter Modellierungsgesichtspunkten betrachtet würde sich aus einer identifizierten Eins-zu-Eins Multiplizität zweier Elemente die Zusammenfassung der betroffenen Elemente zu einem Element empfehlen. Im V-Modell XT werden Produkte und Aktivitäten zudem in unterschiedlichen Anwenderreferenzen dargestellt. Dies ist aufgrund der unmittelbaren Zusammengehörigkeit von Produkt und Aktivität für den Anwender als hinderlich einzustufen.

Anwendungsunfreundlichkeit

Im Rahmen des Erstellungsprozesses des V-Modells XT führt die getrennte Modellierung von Produkten und Aktivitäten zu vielen Redundanzen in den beschreibenden Texten. Die Abgrenzung zwischen einer „Beschreibung des Erstellten“ und einer „Beschreibung des Erstellens“ hat sich im Erstellungsprozess des V-Modells XT als nicht hilfreich herausgestellt. Dies lässt sich an vielen Beispielen im V-Modell XT ablesen. Die Zusammenfassung von Produkt und Aktivität zu einem Konzept würde sich also auch unter diesem Gesichtspunkt anbieten.

Redundanz

Die einschränkenden Multiplizitäten der Produkt-Aktivitäts-Schnittstelle führen zu einer Reihe von Nachteilen bezüglich der Ausdrucksmächtigkeit der Beschreibungsmittel. Aktivitäten ohne ein zugeordnetes fertig zu stellendes Produkt können nicht beschrieben werden. Beispielsweise kann im Rahmen des Projektmanagements eine Aktivität, die allgemein die Tätigkeit der Projektsteuerung beschreibt, aufgrund eines fehlenden unmittelbaren Produktes als Ergebnis dieser Aktivität im V-Modell XT nicht modelliert werden. Die Beschreibung der Projektsteuerung ist zwar zentraler Bestandteil der gängigen Projektmanagement-Literatur, allerdings aus diesem Grund nicht Bestandteil des V-Modells XT.

Mangelnde Ausdrucksmächtigkeit

Die einschränkende Zuordnung genau einer Aktivität zu genau einem Produkt erlaubt auch keine Modellierung mehrerer Aktivitäten zur Erstellung unterschiedlicher Fertigstellungsgrade eines Produktes. So wird beispielsweise zwar eine Anforderungsbewertung als eigene Aktivität modelliert, die Berücksichtigung der Ergebnisse dieser Anforderungsbewertung im Rahmen der Aktivität zur Erstellung der Anforderungen kann allerdings nicht als Aktivität modelliert werden. Der Zusammenhang zwischen Anforderungen und Anforderungsbewer-

tung ist im V-Modell XT stattdessen als inhaltliche Produktabhängigkeit modelliert. Produktabhängigkeiten sind also zum Teil als im Projektplan einzuplanende Aktivitäten zu verstehen.

Im Bereich Systemerstellung erfordern beispielsweise die unterschiedlichen Stufen, die das Produkt System durchläuft, jeweils eigene Aktivitätsbeschreibungen. Das zu erstellende System durchläuft unter anderem die Zustände „integriert“, „installiert“ und beim „Auftraggeber in Betrieb genommen“. Für diese Zustände wären jeweils eigene Aktivitätsbeschreibungen wünschenswert. Das V-Modell XT beschreibt aufgrund restriktiver Beschreibungsmittel allein die Integration des Systems in Form von Aktivitäten.

Das Produkt Besprechungsdokument des V-Modells XT bewegt sich auf einer anderen Abstraktionsebene als die meisten übrigen Produkte, da Besprechungen in vielen unterschiedlichen Kontexten stattfinden können. Eine Möglichkeit, die sinnvollen „Ausprägungen“ eines Besprechungsdokumentes zu modellieren, bestünde in der Angabe dieser Kontexte über die Produkt-Aktivitäts-Schnittstelle. Beispielsweise könnte eine Aktivität zur Erstellung eines Architekturdokumentes zugleich ein Besprechungsdokument als Ergebnis haben, das den Abstimmungsprozess der Architektur im Team dokumentiert. Dies ist im V-Modell ebenfalls nicht modellierbar.

Die einschränkenden Multiplizitäten der Produkt-Aktivitäts-Schnittstelle stellen eine Vereinfachung für den Erstellungsprozess dar. Die Ausdrucksmächtigkeit der Beschreibungsmittel des V-Modells XT wird hierdurch allerdings stark reduziert und den Anforderungen der Praxis an ein Vorgehensmodell zum Teil nicht gerecht.

3.4.3 Abstraktionsebenen im V-Modell XT

Wie sich aus den Beschreibungsmitteln ablesen lässt, bewegt sich das V-Modell XT auf nur einer Abstraktionsebene. Es fehlt ein Konzept für Spezialisierungs-Beziehungen beispielsweise zwischen Produkten. Bei genauer Betrachtung der Inhalte des V-Modells XT im Folgenden zeigt sich allerdings, dass beispielsweise die Produkte nicht auf einer Abstraktionsebene angesiedelt sind. Das V-Modell XT vermischt also Abstraktionsebenen implizit. Dies führt zu Verständnisproblemen auf Seiten der Leser. Im Kontext einer Werkzeugunterstützung können eigentlich unterschiedliche Abstraktionsebenen nicht unterschieden werden.

Wie bereits in Kapitel 3.3.3 dargestellt, enthält das V-Modell XT im Kontext generischer Vorgehensweisen wie der Qualitätssicherung eine Reihe indirekter Produktabhängigkeiten, die von einem Produkt auf alle Produkte zeigen. Das V-Modell XT enthält weitere Beziehungen dieser Art, die nicht explizit modelliert sind. Dies zeigen wir anhand der Beispiele in Abbildung 3–9. Entsprechende Beziehungen können als Indikator für die Vermischung unterschiedlicher Abstraktionsebenen gelten.

Das V-Modell XT beschreibt ein Produkt „*Prüfspezifikation Dokument*“, das Anforderungen an die Prüfung von Dokumenten formuliert. Produktexemplare der „*Prüfspezifikation Dokument*“ müssen für jedes zu prüfende Dokument und auch für jede durchzuführende Prüfung dieses Dokumentes erzeugt werden. Die Umsetzung des Mechanismus zur Qualitätssicherung im V-Modell bezeichnen wir als generisch. Das Produkt „*Prüfspezifikation Dokument*“ bewegt sich auf einer höheren Abstraktionsebene als ein Produkt wie beispielsweise das *Projekthandbuch*, da es sich auf alle Produkte bezieht, die eine bestimmte Bedingung erfüllen.

3.4.4 Systemerstellung im V-Modell XT

Das Fehlen eines Spezialisierungskonzeptes wird im V-Modell XT an der Beschreibung der Disziplin Systemerstellung, insbesondere in Bezug auf die Strukturierung des Systems in Form von Systemelementen, besonders deutlich.

Ein System untergliedert sich im V-Modell XT, wie in Abbildung 3–10 dargestellt, in *Segmente*, *SW- und HW-Einheiten*, „*Externe Einheiten*“, sowie *SW- und HW-Komponenten* und *SW- und HW-Module*. Diese Systemelemente sind im V-Modell XT jeweils als Produkte modelliert.

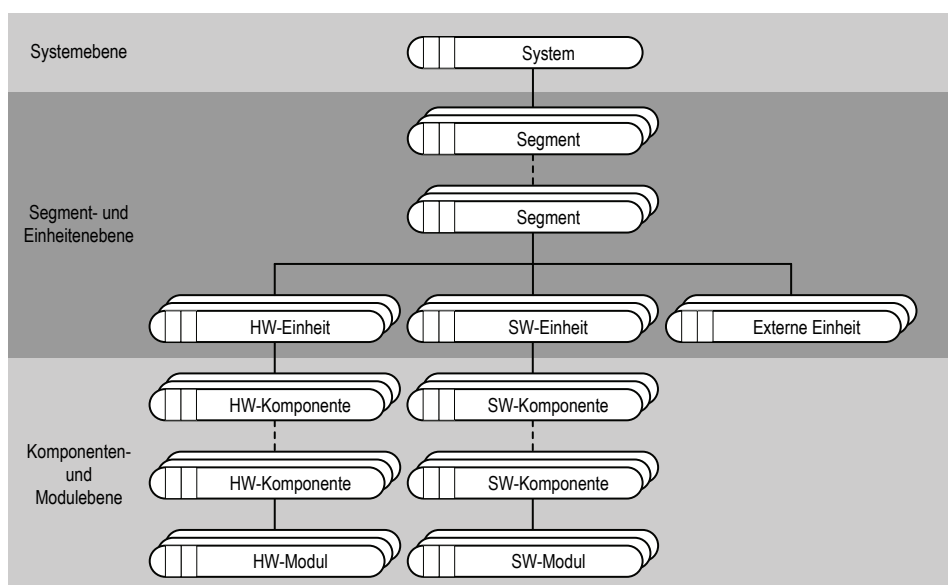


Abbildung 3–10: Systemelemente im V-Modell XT

Redundante Texte

Die Beschreibung der Systemelemente ist im V-Modell XT schematisch. Die Texte zu den Systemelementen sind weitgehend unspezifisch bezüglich des jeweils beschriebenen Systemelements und damit weitgehend redundant. Beispielsweise unterscheiden sich die Texte der Aktivitäten "Zur SW-Komponente integrieren" und "Zur SW-Einheit integrieren" allein durch den Verweis auf das jeweils unterschiedliche Systemelement. Ähnlich verhält es sich mit den Beschreibungen zu Hardware und Software. Die Beschreibungen der Aktivitäten "Zur SW-Einheit integrieren" und "Zur HW-Einheit integrieren" sind beinahe identisch.

Redundante Referenzierung

Weitere Gemeinsamkeiten der Systemelemente sind in der schematischen Umsetzung der erzeugenden Produktabhängigkeiten zu finden. Für SW-Einheiten, SW-Komponenten und SW-Module umfasst die Teilmenge der jeweils von diesen erzeugten Produkte die SW-Spezifikation, die „Prüfspezifikation Systemelement“, die „Prüfprozedur Systemelement“, das „Prüfprotokoll Systemelement“, das „Prüfprotokoll Benutzbarkeit“, die „Prüfspezifikation Benutzbarkeit“ und die „Gefährdungs- und Systemsicherheitsanalyse“. Für SW-Einheiten, SW-Komponenten und SW-Module ist jeweils eine eigene erzeugende Produktabhängigkeit beschrieben. Die Menge der erzeugten Produkte ist dabei jeweils identisch.

Einem Anwender des V-Modells XT wäre eine Beschreibung der Gemeinsamkeiten der Systemelemente anhand eines Produktes Systemelement, sowie eine Beschreibung der marginalen Unterschiede der speziellen Systemelemente wie Segment, SW-Einheit oder SW-Komponente einfacher zu vermitteln. Redundante Beschreibungen führen dagegen

stets zu einem unnötig hohen Lese- und Einarbeitungsaufwand. Ebenso würde sich eine Beschreibung des allgemeinen Erzeugungsprinzips auf abstrakter Ebene anbieten.

Aus Sicht der Pflege und Weiterentwicklung des V-Modells XT führen redundante Texte und redundante Referenzierung ebenfalls zu einem erhöhten Aufwand, da Redundanzen konsistent gehalten werden müssen. Andererseits könnte die Disziplin Systemerstellung im V-Modell XT insgesamt als eine Verfehlung der Abstraktionsebene bezeichnet werden, da die Inhalte unterschiedlicher Beschreibungen zur Systemerstellung starke Ähnlichkeit aufweisen. Die Beschreibung konkreter Systemelemente wie Segment, SW-Einheit oder SW-Komponente wäre aus Sicht der derzeitigen Inhalte der Beschreibung nicht erforderlich.

3.4.5 Zusammenfassung

Die Abstraktionsprinzipien eines Modells geben Aufschluss über die Qualität des Modells. Ein zu hoher Abstraktionsgrad führt zu unbefriedigenden Aussagen bezüglich der modellierten Domäne. Als Indikator für einen zu niedrigen Abstraktionsgrad dagegen können Redundanzen im Modell gelten.

Das V-Modell XT ist den zur Verfügung stehenden Beschreibungsmitteln des Meta-Modells zu Folge auf genau einer Abstraktionsebene angesiedelt. Die Konzepte Produkt und Aktivität sind so miteinander verbunden, dass einem nicht-externen Produkt genau eine Aktivität zugeordnet wird. Produkte und Aktivitäten bewegen sich also auf derselben Abstraktionsebene. Wie wir an einer Reihe von Beispielen gezeigt haben, führt dies zu einer mangelnden Ausdrucksmächtigkeit bezüglich bestimmter Aspekte. Außerdem führt die inhaltliche Umsetzung zu vielen redundanten Texten, da die Abgrenzung einer Aktivität als „Beschreibung des Erstellens“ von einem Produkt als „Beschreibung des Erstellten“ in der Praxis nicht klar ist. Eine Zusammenfassung der Konzepte Produkt und Aktivität zu einem Konzept würde sich vor diesem Hintergrund anbieten.

Ein Konzept zur Modellierung unterschiedlicher Abstraktionsebenen fehlt im V-Modell XT gänzlich. Aufgrund der Anforderungen der Praxis enthält das V-Modell XT allerdings Produkte offensichtlich unterschiedlicher Abstraktionsebenen, wie beispielsweise eine Produktbibliothek und ein Produkt wie das Projekthandbuch. Abstraktionsebenen sind im V-Modell XT also implizit vermischt und damit einer Werkzeugunterstützung nicht zugänglich.

Am Beispiel der Umsetzung der Disziplin Systemerstellung wird schließlich das Fehlen eines Abstraktionskonzeptes besonders deutlich. Die Modellierung der Systemerstellung weist einige Redundanzen auf, wie sich beispielsweise anhand der Beschreibungen zu den unterschiedlichen Systemelementen wie Segment, SW-Einheit oder SW-Komponente zeigt. Durch ein Spezialisierungskonzept ließen sich diese Redundanzen auflösen.

Redundanz erweist sich aus Sicht des Anwenders aufgrund eines unnötigen Leseaufwands stets als unvorteilhaft. Aus Sicht des Weiterentwicklungsprozesses des V-Modells XT führt Redundanz zu einem erhöhten Aufwand im Hinblick auf die Konsistenzerhaltung.

Wesentliche Ziele im Rahmen der Weiterentwicklung des V-Modells XT waren die einfache Anwendbarkeit der Papier- beziehungsweise HTML-Version, sowie die einfache Handhabbarkeit der Beschreibungsmittel aus Sicht der Prozessingenieure. Um der Aufgabe einer "Architektur" in einem großen und heterogenen Team gerecht zu werden, wurden im Meta-Modell stets einfache Zusammenhänge präferiert, beispielsweise also der Zusammenhang zwischen einer Aktivität und genau einem Produkt. Wie wir in den vorangehenden Kapiteln gezeigt haben, führte dies an einigen Stellen zu einer mangelnden Ausdrucksmächtigkeit der gewählten Beschreibungsmittel in Bezug auf die in einem Vorgehensmodell inhärente Komplexität.

4 Modellierungstechnik für Vorgehensmodelle

In diesem Kapitel legen wir den Grundstein für die Modellierung von Vorgehensmodellen. Wir definieren die in der Arbeit verwendete Modellierungstechnik für Vorgehensmodelle.

In Kapitel 4.1 betrachten wir Grundlegendes zur Definition von Modellierungssprachen. Dabei gehen wir auf die Architektur aufeinander aufbauender Meta-Ebenen ein, wie sie in der Meta Object Facility (MOF) definiert wird.

In Kapitel 4.2 stellen wir die in der Arbeit verwendete Modellierungstechnik dar. Als Modellierungstechnik bedienen wir uns Klassendiagrammen und Constraints in Form prädikatenlogischer Ausdrücke. Da Klassendiagramme in der UML teilweise nicht präzise definiert sind beziehungsweise bestimmte Aspekte der UML in der Arbeit spezifisch eingesetzt werden, definieren wir einige Aspekte der verwendeten Modellierungstechnik in diesem Kapitel. Dabei legen wir Instanziierung sowie die Generalisierung und Spezialisierung von Klassen und Assoziationen fest.

In Kapitel 4.3 stellen wir die Anwendung der definierten Modellierungstechnik für die Modellierung von Vorgehensmodellen dar. Als Modellierungsansatz dient uns in den folgenden Kapiteln eine Einteilung in Meta-Ebenen, die sich durch das Konzept der Spezialisierung abgrenzen.

Inhalt des Kapitels	Seite
4.1 Grundlegendes zur Definition von Modellierungssprachen	87
4.2 Definition der verwendeten Modellierungstechnik	90
4.3 Modellierungsansatz für Vorgehensmodelle und verwandte Ansätze	104

4.1 Grundlegendes zur Definition von Modellierungssprachen

Modelle und Meta-Modelle

Zunächst betrachten wir die Begriffe Modell und Meta-Modell.

Unter einem **Modell** verstehen wir in der Informatik eine vereinfachte Darstellung beziehungsweise eine Abstraktion eines Ausschnitts der realen Welt. Modelle entstehen durch das Bilden von Äquivalenzklassen und durch Strukturierung.

Der Begriff des Meta-Modells leitet sich aus der Sprachstufentheorie ab (vgl. [Strah98]). Entsprechend der Darstellungsfunktion von Sprache werden Aussagen über einen zu untersuchenden Gegenstandsbereich in Sprache formuliert. Wird Sprache selbst zum Gegenstand der Untersuchung, so wird über eine Sprache in einer Sprache gesprochen. Um Verwirrungen zu vermeiden ist es zweckmäßig, Sprachebenen zu unterscheiden – die Objektsprache, die Gegenstand der Betrachtung ist, und die Metasprache, in der die Betrachtung erfolgt.

Die Metasprache kann wiederum selbst zum Gegenstand der Betrachtung werden, wobei die jeweils übergeordnete Sprache als Metametasprache bezeichnet wird. Diese kann auch in Bezug auf die vormalige Metasprache, die nun Objektsprache ist, als Metasprache bezeichnet werden. Das Prinzip der Metasprachen kann also rekursiv angewendet werden, es entsteht eine Hierarchie von Sprachen. Die Eigenschaft einer Sprache, Meta- oder Objekt-

sprache zu sein ist keine absolute Eigenschaft, sondern eine relative Eigenschaft in Bezug auf eine andere Sprache.

Ziel der Modellbildung ist es, einen Gegenstandsbereich durch ein Modell zu beschreiben. Wird die Sprache, in der ein Modell formuliert ist, selbst zum Gegenstandsbereich der Modellbildung, so spricht man in Bezug auf den ursprünglichen Gegenstandsbereich von einem Meta-Modell.

Ein **Meta-Modell** ist ein Modell eines Modells, wobei das übergeordnete Modell ein Beschreibungsmodell ist, das die Sprache, in der das untergeordnete Modell formuliert ist, definiert (vgl. [Strah98] und [OMG03b]).

Die Bildung von Äquivalenzklassen und die Strukturierung erfolgen im Meta-Modell in Bezug auf ein Modell.

Unified Modeling Language (UML) und Meta Object Facility (MOF)

Die Meta Object Facility (MOF) Spezifikation (vgl. [OMG02] und [OMG03b]) der OMG definiert eine Meta-Modell-Architektur in Form einer beliebigen Anzahl aufeinander aufbauender (Meta-)Ebenen. Zudem definiert die MOF ein standardisiertes Meta-Modell der obersten Stufe, das so genannte MOF-Meta-Modell.

(a) Meta-Modell-Architektur der MOF 1.4	(b) Anwendung der MOF im Rahmen der UML	(c) Modellierung von Vorgehensmodellen
Mx MOF-Meta-Modell		
...		
M3 (<i>Meta-Meta-Modell</i>)	MOF-Meta-Modell	Modellierungstechnik: Klassendiagramme mit Generalisierung und Spezialisierung
M2 (<i>Meta-Modell</i>)	UML	Vorgehens-Meta-Modell
M1 (<i>Modell</i>)	User Model	Vorgehensmodell
M0 (<i>Instanzen</i>)	Runtime Instances	Projektebene (inklusive Projektplan)

Abbildung 4–1: Die Meta Object Facility (MOF)

Als Abstraktionsschritt beziehungsweise als Metaisierungsprinzip, also als Prinzip zur Identifizierung von Sprach- beziehungsweise Modellebenen dient in der MOF 1.4 das Konzept von Klasse und Instanz. Unter einer Klasse verstehen wir eine Äquivalenzklasse, unter einer Instanz verstehen wir ein Element dieser Äquivalenzklasse. Jede Ebene ist als eine Instanz

der übergeordneten Ebene zu sehen, das heißt jedes Element einer Ebene ist eine Instanz genau eines Elementes der übergeordneten Ebene. Die Anzahl aufeinander aufbauender Meta-Ebenen kann, wie in Abbildung 4–1 (a) dargestellt, beliebig gewählt werden. In der Nachfolgeversion MOF 2.0 erfolgt dagegen keine Festlegung auf Instanziierung als Metaisierungsprinzip. Stattdessen erlaubt die Spezifikation beliebige nicht genauer festgelegte Beziehungen zwischen (Meta-)Ebenen.

Die Standardisierung im Rahmen der MOF verfolgt das Ziel, die Interoperabilität von Anwendungen beziehungsweise die Austauschbarkeit von Daten unterschiedlicher Anwendungen zu fördern. Meta-Daten dienen der Beschreibung der Syntax von Daten. Die Syntax beinhaltet dabei die Festlegung der Struktur der Meta-Daten. Die MOF standardisiert die Sprache zur Beschreibung der Meta-Daten in Form des MOF-Meta-Modells. Die MOF selbst allerdings ist kein Implementierungsmodell, sondern ein logisches deklarativ beschriebenes Modell. Details möglicher Implementierungen werden bewusst offen gelassen.

Die meisten Anwendungen der MOF nutzen nur eine geringe Anzahl von (Meta-)Ebenen. Anwendungsfälle der MOF sind beispielsweise virtuelle Warenhäuser, Modellierungs- oder Entwicklungswerkzeuge. Die Modellierungssprache UML (vgl. [OMG03a]) wird entsprechend den Konzepten der MOF mit den vier Ebenen „Runtime-Instances“, „User Model“, „UML“ und dem „MOF-Meta-Modell“ als oberster Ebene definiert, wie in Abbildung 4–1 (b) dargestellt.

Die MOF wird in Form von abstrakter Syntax, Wohlgeformtheitsregeln und natürlichsprachig formulierter Semantik definiert. Das in der MOF definierte Meta-Modell wird mit Hilfe der Modellierungstechnik der UML zur Beschreibung der abstrakten Syntax spezifiziert. Dabei werden Klassen als Modellierungstechnik sowie die Instanziierung von Klassen in Form von Objekten als Semantik eingesetzt. Wohlgeformtheitsregeln werden teils natürlichsprachig, teils in einer Sprache für prädikatenlogische Aussagen über gültigen Instanzen eines Klassendiagramms, der so genannten Object Constraint Language (OCL) (vgl. [OMG03c]), angegeben.

In Konsequenz bedeutet dies, dass die UML letztlich durch Konzepte der UML selbst definiert wird.

Anwendung der MOF in der vorliegenden Arbeit

Die vorliegende Arbeit lehnt sich an die Konzepte der MOF an. Wir wenden die Idee der (Meta-)Ebenen für die Definition einer Sprache für Vorgehensmodelle an, wie in Abbildung 4–1 (c) dargestellt. Wir unterscheiden dabei die Ebenen Modellierungstechnik, Vorgehens-Meta-Modell, Vorgehensmodell und Projekt. Die Projektebene beinhaltet den Projektplan.

Für die Definition unseres Vorgehens-Meta-Modells verwenden wir im Rahmen der Arbeit nicht unmittelbar das MOF-Meta-Modell. Vielmehr definieren wir in den folgenden Kapiteln eine Modellierungstechnik. Diese basiert ähnlich dem MOF-Meta-Modell auf Klassen und Assoziationen sowie Constraints in Form von prädikatenlogischen Aussagen.

Im Vergleich zur MOF oder zu Klassendiagrammen, wie sie in der UML definiert werden, verfügt die in der Arbeit verwendete Modellierungstechnik über eine reduzierte Anzahl von Konzepten, die dem Zweck der vorliegenden Arbeit angemessen sind. Da wir keine unmittelbare Abbildung der Konzepte in eine objektorientierte Programmiersprache wie Java anstreben, sind beispielsweise Konzepte für Operationen als Bestandteil von Klassen oder Sichtbarkeitsregeln für Attribute im Kontext der Arbeit nicht erforderlich.

Die verwendete Notation von Klassendiagrammen und Constraints dient der Erstellung eines logischen Modells, das von Details einer möglichen Implementierung weiter abstrahiert, als Klassendiagramme der UML dies tun. Im Kontext einer Werkzeugunterstützung sind die Modelle der Arbeit als logische Datenmodelle zu sehen. Das konzipierte Vorgehens-Meta-Modell kann beispielsweise als logisches Datenmodell eines Editors für Vorgehensmodelle verstanden werden.

Ein Vorteil einer Verwendung des MOF-Meta-Modells als Modellierungstechnik wäre in einer standardisierten Abbildung in andere Datenmodelle, wie beispielsweise in ein XML-Schema zu sehen. Die Abbildung der hier definierten Modellierungstechnik in das standardisierte MOF-Meta-Modell ist allerdings aufgrund der Ähnlichkeit der Konzepte einfach zu erreichen.

Ein Nachteil der MOF besteht in der fehlenden formalen Fundierung der Sprache. Die Semantik der Sprachkonstrukte wird in natürlicher Sprache definiert. Die Beschreibung der Sprachkonstrukte ist zudem teilweise unpräzise (vgl. [OMG02] und [OMG03b]). Das Konzept der einmaligen Instanziierung von Klassen und Assoziationen in der UML erscheint plausibel, das Konzept der Instanziierung über mehrere (Meta-)Ebenen in der MOF 1.4 bleibt allerdings unklar. In der MOF 2.0 dagegen sind beliebige Beziehungen zwischen (Meta-)Ebenen zulässig.

Beispielsweise stellt sich die Frage, ob Einschränkungen eines Modells in Form von Multiplizitäten oder prädikatenlogischen Constraints auch auf der jeweils nächsten (Meta-)Ebene gelten. Diese Frage beantwortet auch die Spezifikation der UML nicht – weder im Hinblick auf das Konzept der Instanziierung noch im Hinblick auf das Spezialisierungskonzept (vgl. [OMG03a]).

Analog zu den Ansätzen in [CEK01] geben wir eine semi-formale Definition der Semantik der Modellierungstechnik an. Im folgenden Kapitel definieren wir die konkrete Syntax der Modellierungstechnik, sowie ihre abstrakte Syntax und darauf aufbauend ihre Semantik.

4.2 Definition der verwendeten Modellierungstechnik

Zunächst definieren wir in Kapitel 4.2.1 die Semantik von Klassen, Assoziationen, Attributen und Constraints im Hinblick auf das Konzept der Instanziierung. Das Kapitel gliedert sich in die Definition der abstrakten Syntax, der Semantik und der konkreten Syntax.

Daraufhin erweitern wir in Kapitel 4.2.2 die Modellierungstechnik um die Konzepte Generalisierung und Spezialisierung von Klassen beziehungsweise Assoziationen. Generalisierung und Spezialisierung dienen uns im Weiteren zur Abgrenzung der (Meta-)Ebenen der Vorgehensmodellierung.

Abschnitte in Kapitel 4.2	Seite
4.2.1 Klassen und Objekte	90
4.2.2 Generalisierung und Spezialisierung	97

4.2.1 Klassen und Objekte

Abbildung 4–2 zeigt das gängige Definitions-Muster für Sprachen in der Informatik. Für eine Sprache wird eine abstrakte Syntax konstruiert, die von den Details der konkreten Syntax abstrahiert. Diese abstrakte Syntax wird durch eine Abbildung in eine semantische Domäne, beispielsweise ein mathematisches Modell, interpretiert. Die Bedeutung der abstrakten Sprachkonstrukte wird in der verwendeten semantischen Domäne definiert.

Unter einer formalen Semantik verstehen wir also eine Interpretationsfunktion, die den in der Syntax der formalen Sprache formulierten Ausdrücken eine Bedeutung in einer semantischen Domäne zuweist. Handelt es sich bei der semantischen Domäne um eine logische oder mathematische Struktur, spricht man von einer denotationalen Semantik.

Die abstrakte Syntax für Klassendiagramme und Constraints geben wir in der Notation von Klassendiagrammen und Constraints an. Constraints der abstrakten Syntax dienen als Re-

geln für die Wohlgeformtheit eines Modells. Wir verwenden also für die Festlegung der abstrakten Syntax dasselbe Beschreibungsmuster wie die MOF beziehungsweise die UML.

Für die Festlegung der Semantik verwenden wir den Ansatz zur Definition der Semantik von Klassendiagrammen aus [CEK01]. Wir geben dabei die Semantik nicht durch eine Abbildung in eine mathematisch fundierte Sprache als semantische Domäne an, sondern bilden die Modellierungssprache der Klassen und Assoziationen in die semantische Domäne der Instanzen ab. Diese Abbildung formulieren wir mit Hilfe prädikatenlogischer Constraints.

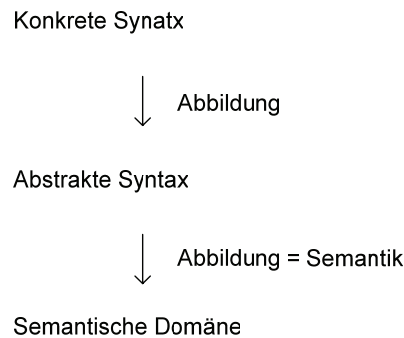


Abbildung 4–2: Muster zur Definition der Semantik einer Sprache

Da Instanzen in einer der Klassenmodellierung ähnlichen Art beschrieben werden, stellt dieses Definitionsmuster wie auch die Definition der MOF oder der UML eine semi-formale Semantik dar. Man bezeichnet das Definitionsmuster als „meta-zyklisch“ oder auch als selbstbeschreibend. Für eine formale Definition wäre als semantische Domäne eine externe Sprache wie beispielsweise die Mathematik erforderlich. Die semi-formale Definition durch Meta-Modellierung in den folgenden Kapiteln ist allerdings hinreichend, um bestimmte semantische Unklarheiten der MOF beziehungsweise UML für die Arbeit geeignet zu präzisieren.

Die Semantik prädikatenlogischer Constraints definieren wir in der Arbeit nicht eigens, sondern stützen uns auf gängige Spezifikationen ab. Auch die konkrete Syntax der Modellierungssprache definieren wir in der Arbeit nicht formal. Klassen, Assoziationen und Attribute notieren wir grafisch, Constraints in Form von prädikatenlogischen Ausdrücken. Beide Notationen setzen wir als bekannt voraus.

Abstrakte Syntax – Klassen

Abbildung 4–3 zeigt die abstrakte Syntax der Modellierungssprache in der Notation eines Klassendiagramms. Die in Abbildung 4–3 nicht dargestellten Multiplizitäten verstehen wir als Multiplizitäten der Form „ * “.

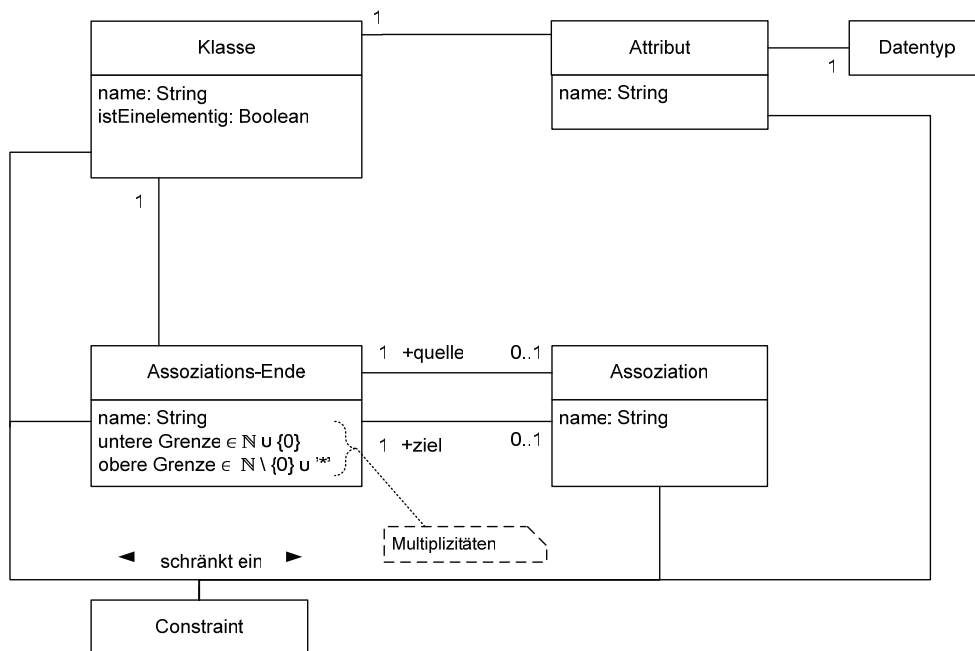


Abbildung 4–3: Abstrakte Syntax der Modellierungssprache

(1)

Zusätzlich gilt die folgende Wohlgeformtheitsregel. Ein *Assoziations-Ende* ist eindeutig einer *Assoziation* zugeordnet und die *Assoziations-Enden* *quelle* und *ziel* einer *Assoziation* sind nicht identisch:

$$\forall ae \in \text{Assoziations-Ende}: |ae.\text{Assoziation}| = 1 \wedge$$

$$\forall a \in \text{Assoziation}: a.\text{quelle} \neq a.\text{ziel}$$

Klassen, *Assoziationen*, *Assoziations-Enden* und *Attribute* sind benannte Elemente. Der Name einer Klasse dient als eindeutiger Identifikator. Ein Namensraumkonzept benötigen wir in der Arbeit nicht. Die Namen von Assoziationen, Assoziations-Enden und Attributen fungieren im Kontext von Klassen als eindeutige Identifikatoren.

Klassen besitzen beliebig viele Attribute, für die jeweils ein Datentyp festgelegt ist. Datentypen können beispielsweise Zahlen, Zeichenketten (Strings) oder Wahrheitswerte (Booleans) sein. Das Konzept der Datentypen betrachten wir hier nicht weiter.

Klassen sind durch binäre, gerichtete Assoziationen verbunden. Die Richtung einer Assoziation wird dabei durch die Assoziations-Enden *quelle* und *ziel* festgelegt. Assoziations-Enden selbst sind benannte Elemente. Assoziations-Enden dienen neben der Benennung und damit Zugänglichkeit für die Formulierung prädikatenlogischer Constraints der Festlegung von Einschränkungen in Form von Multiplizitäten. Eine Multiplizität ist durch eine „untere Grenze“ und durch eine „obere Grenze“ definiert. Der Wertebereich der „unteren Grenze“ besteht aus einer natürlichen Zahl inklusive der Null, der Wertebereich der „oberen Grenze“ aus einer natürlichen Zahl ohne die Null sowie dem Zeichen „*“, wobei „*“ für eine nicht einschränkende Multiplizität steht.

Constraints formulieren Einschränkungen über Geflechten aus Klassen und Assoziationen.

Ein Beispiel für die konkrete Syntax eines Klassendiagramms ist ebenfalls durch Abbildung 4–3 gegeben, ein Beispiel eines Constraints ist obiges Constraint (1). Die Semantik dieser Konzepte definieren wir im folgenden Abschnitt.

Semantik – Klassen und Objekte

Abbildung 4–4 zeigt die verwendete semantische Domäne. Objekte und Links sind benannte Elemente. Links sind binäre und gerichtete Beziehungen zwischen Objekten. Objekte besitzen Attribut-Werte.

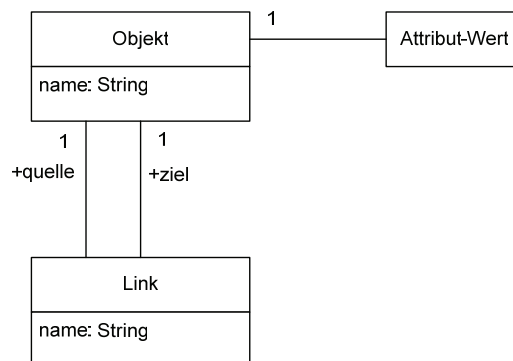


Abbildung 4–4: Semantische Domäne der Modellierungssprache: Instanzen

Abbildung 4–5 definiert zusammen mit den folgenden Constraints die semantische Interpretation der abstrakten Syntax durch die semantische Domäne.

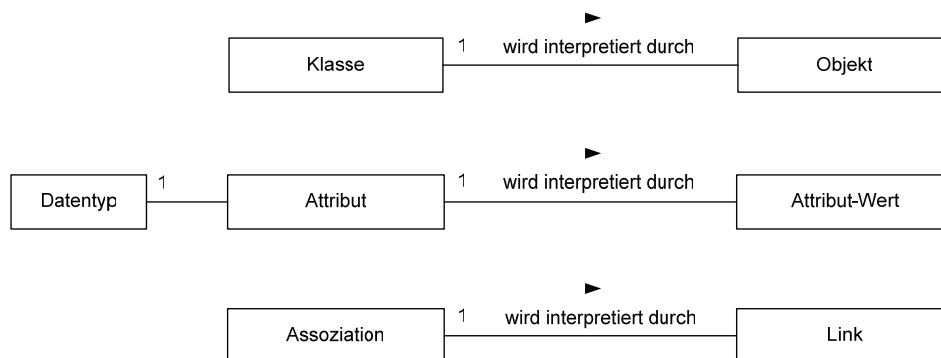


Abbildung 4–5: Semantik der Modellierungssprache

Die Semantik von Klassen wird durch Objekte interpretiert. Ein Objekt ist eine Instanz einer Klasse. Eine Klasse kann durch eine beliebige Anzahl von Objekten ausgeprägt werden, einem Objekt ist dagegen immer genau eine Klasse zugeordnet.

Assoziationen werden durch Links interpretiert. Ein Link ist eine Instanz einer Assoziation. Eine Assoziation kann durch eine beliebige Anzahl von Links ausgeprägt werden, einem Link ist dagegen immer genau eine Assoziation als Typ zugeordnet.

Attribute entsprechen in der semantischen Domäne Attribut-Werten. Der Wert eines Attributs muss dabei konform mit dem Datentyp des Attributs sein. Datentypkonformität formalisieren wir hier allerdings nicht. Die Semantik prädikatenlogischer Constraints definieren wir im Rahmen der Arbeit nicht eigens, sondern stützen uns auf gängige Spezifikationen ab.

(2)

Ein Objekt besitzt zu allen Attributen seiner Klasse einen Attributwert:

$$\forall o \in \text{Objekt}: \forall a \in o.\text{Klasse}.\text{Attribut}: \exists aw \in o.\text{AttributWert}: aw.\text{Attribut} = a$$

(3)

Links zwischen Objekten sind Instanziierungen von Assoziationen zwischen Klassen. Damit müssen die assoziierten Objekte jeweils Instanziierungen der assoziierten Klassen sein:

$$\forall l \in \text{Link}: l.\text{Assoziation}.\text{quelle}.\text{Klasse} \in l.\text{quelle}.\text{Klasse} \wedge \\ l.\text{Assoziation}.\text{ziel}.\text{Klasse} \in l.\text{ziel}.\text{Klasse}$$

(4)

Die im Modell definierten Multiplizitäten auf Assoziations-Enden müssen im instanziierten Modell eingehalten werden:

$$\forall a \in \text{Assoziation}: \forall o \in \text{Objekt}: \\ a.\text{ziel}.\text{untereGrenze} \leq \\ |\{l \in \text{Link} \mid l.\text{Assoziation} = a \wedge l.\text{quelle} = o\}| \\ \leq a.\text{ziel}.\text{obereGrenze} \\ \wedge \\ a.\text{quelle}.\text{untereGrenze} \leq \\ |\{l \in \text{Link} \mid l.\text{Assoziation} = a \wedge l.\text{ziel} = o\}| \\ \leq a.\text{quelle}.\text{obereGrenze}$$

Das Zeichen „ * “ als Wert einer „oberen Grenze“ wird in Constraint (4) als unendliche Zahl interpretiert.

(5)

Ist eine Klasse als einelementig gekennzeichnet, gibt es höchstens ein Objekt dieser Klasse:

$$\forall k \in \text{Klasse}: k.\text{istEinelementig} = \text{True} \Rightarrow |\{o \in \text{Objekt} \mid o.\text{Klasse} = k\}| \leq 1$$

Constraints sind prädikatenlogische Aussagen über Klassen, Assoziationen, Assoziations-Enden und Attributen. Constraints erhalten ihre Semantik durch die übliche prädikatenlogische Interpretation, die wir im Rahmen der Arbeit nicht beleuchten. Wir verwenden für prädikatenlogische Aussagen über Klassendiagrammen die Konzepte der Object Constraint Language (OCL) (vgl. [OMG03c]), wie sie beispielsweise auch in [Breu01] zur Anwendung kommen.

Konkrete Syntax – Klassen und Objekte

Die konkrete Syntax von Klassendiagrammen ist grafisch. Diese Syntax definieren wir nicht formal. In Abbildung 4–6 stellen wir einzig die Aspekte der Notation dar, die spezifisch für die

vorliegende Arbeit sind. Dabei handelt es sich meist um abkürzende Notationen, die in den folgenden Kapiteln der Arbeit zur Anwendung kommen.

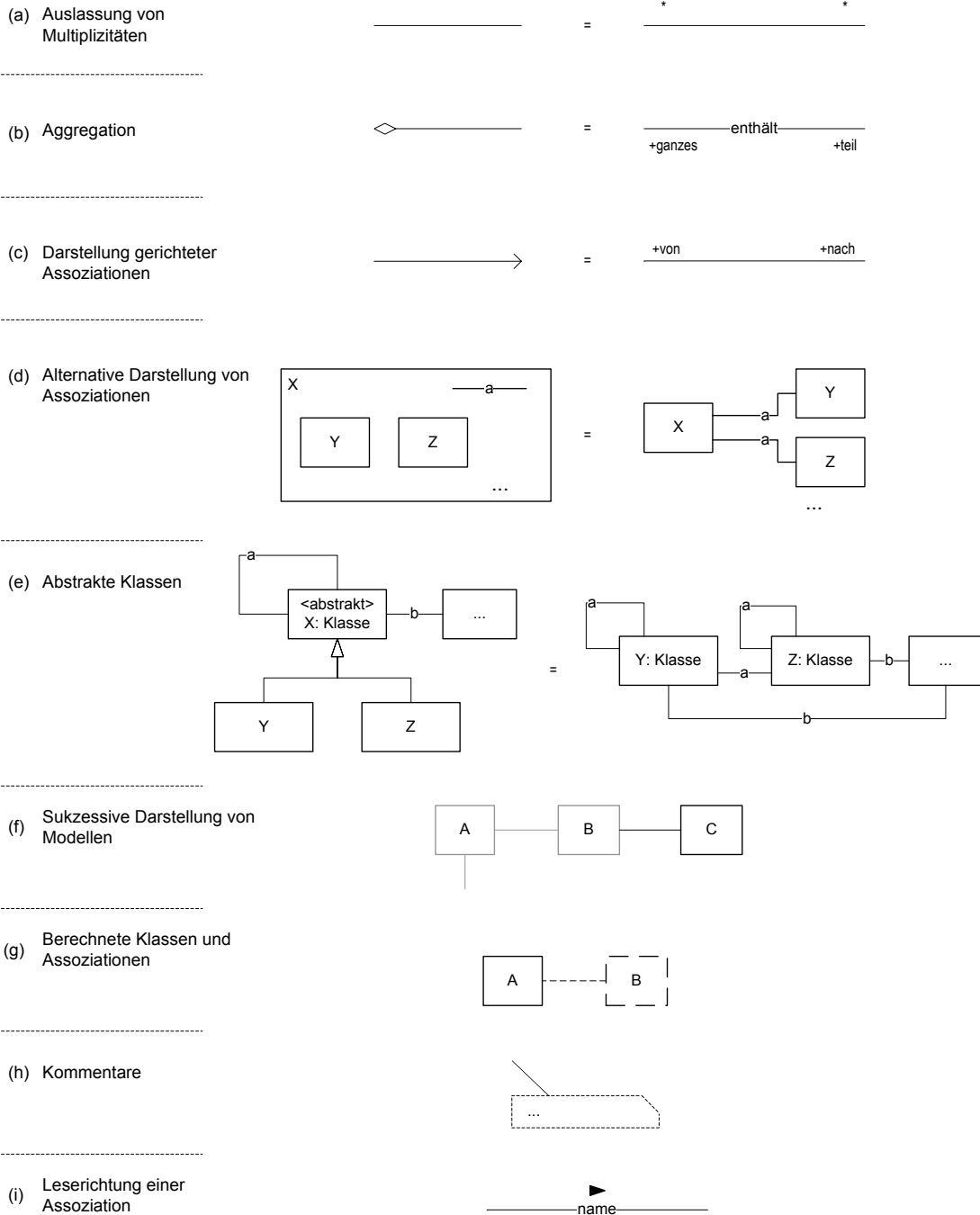


Abbildung 4–6: Konkrete Syntax der Modellierungssprache – Spezifika

- (a) Werden für ein Assoziations-Ende keine Multiplizitäten angegeben, so enthält das Modell für dieses Assoziations-Ende keine Einschränkungen. Dies entspricht einer Multiplizität „ * “.
- (b) Unter Aggregation verstehen wir eine bestimmte Art von Assoziationen, deren Assoziations-Quelle mit einer Raute gekennzeichnet ist. Aggregationen werden als normale Assoziationen interpretiert und sind daher in der abstrakten Syntax nicht eigens modelliert. Eine Aggregation ist lediglich eine abkürzende Notation für eine Assoziation mit dem Namen *enthält* und den Namen der Assoziations-Enden *ganzes* und *teil*. Aggregationen dienen als abkürzende Notationen der übersichtlicheren Darstellung eines Modells. Die UML unterscheidet zudem zwischen Aggregation und Komposition (vgl. [OMG03a]). Diese Unterscheidung ist im Kontext der vorliegenden Arbeit nicht von Bedeutung, da wir keine Abhängigkeiten eines Objektes vom Lebenszyklus eines anderen Objektes modellieren.
- (c) Pfeile können als abkürzende Darstellung für gerichtete Assoziationen verwendet werden. Gerichtete Assoziationen sind durch die Namen „von“ und „nach“ für Assoziations-Enden gekennzeichnet. Die Namen der Assoziations-Enden werden dabei abkürzend nicht notiert.
- (d) Steht eine Klasse mit vielen anderen Klassen über Assoziationen desselben Namens in Beziehung, kann diese Klasse zur Verbesserung der Übersichtlichkeit als ein die in Beziehung stehenden Klassen umschließendes Rechteck dargestellt werden.
- (e) Sollen Assoziationen gleichen Namens eine Menge von Klassen verbinden, kann diese Assoziation als eine Assoziation einer abstrakten Klasse dargestellt werden. Die Klassen, die in Beziehung stehen sollen, werden mit der Darstellung eines nicht ausgefüllten Pfeils als Spezialisierungen der abstrakten Klasse notiert. Diese Darstellung hat keinen Einfluss auf die Semantik und ist auf semantischer Ebene nicht mit dem Spezialisierungskonzept im folgenden Kapitel identisch, auch wenn die Notation für Spezialisierung dieselbe ist. Abstrakte Klassen sind nicht Bestandteil der abstrakten Syntax der Modellierungstechnik, da sie nicht instanziiert werden und somit auch nicht in die semantische Domäne abgebildet werden können. Wir verstehen abstrakte Klassen also lediglich als Darstellungsmittel zur Verbesserung der Übersichtlichkeit.
- (f) Im Rahmen der Arbeit stellen wir Modelle nicht als Ganzes dar, sondern präsentieren sukzessive Erweiterungen bereits eingeführter Modelle. Bereits eingeführte Modellelemente werden grau dargestellt.
- (g) Leitet sich die Existenz bestimmter Klassen oder Assoziationen über ein oder mehrere Constraints aus der Existenz anderer Klassen oder Assoziationen ab, stellen wir diese „berechneten“ Klassen oder Assoziationen mit unterbrochenen Linien dar. Diese Darstellung vermittelt die Intuition, welche Klassen oder Assoziationen sich aus anderen ableiten. Die Richtung der Ableitung ergibt sich dabei nicht immer aus den jeweiligen Constraints. Die Kennzeichnung bestimmter Klassen oder Assoziationen als „berechnet“ ist daher oft mit Freiheitsgraden verbunden.
- (h) Kommentare werden wie in Abbildung 4–6 (h) gezeigt dargestellt und sind ohne Einfluss auf die Semantik.
- (i) Die Leserichtung einer Assoziation kann durch ein ausgefülltes Dreieck notiert werden. Die Leserichtung ist ohne Einfluss auf die semantische Interpretation und nicht mit einer gerichteten Assoziation entsprechend Abbildung 4–6 (c) zu verwechseln.

Konkrete Syntax – Constraints

Wir verwenden in der Arbeit die Konzepte der OCL (vgl. [OMG03c]), allerdings in einer anderen Syntax, die uns für die Formulierung von Constraints übersichtlicher erscheint. Diese Syntax für Constraints erachten wir als selbsterklärend. Einige spezifische Aspekte der

Syntax prädikatenlogischer Constraints, die wir in den folgenden Kapiteln der Arbeit verwenden, geben wir hier an:

- Anders als die OCL verwenden wir die in der Prädikatenlogik üblichen Symbole wie Allquantoren „ \forall “, Existenzquantoren „ \exists “ oder Implikation „ \Rightarrow “.
- Um Ausdrücke zur Navigation über Klassen und Assoziationen übersichtlicher zu gestalten verwenden wir Ausdrücke der Form
 „Name einer Assoziation Name einer Rolle der Assoziation“
 oder
 „Name einer Assoziation Name einer über ein Assoziations-Ende verbundenen Klasse“.
- Ist der Name einer Klasse oder Assoziation nicht eindeutig, die Kombination aus Klassennamen und Generalisierung der Klasse dagegen schon, verwenden wir zur Unterscheidung Ausdrücke der Form
 „Name einer Klasse:Name einer Generalisierung der Klasse“
 oder
 „Name einer Assoziation:Name einer Generalisierung der Assoziation“.
- Als spezifisches Ausdrucksmittel verwenden wir eine Notation, die die mehrfache Navigation über einer Menge von Assoziationen, die einen Zyklus bilden, darstellt. Wir verwenden die Notation
 „ (...) ^{0..*} “
 um anzuzeigen, dass die Assoziationen „ (...) “ einmal, mehrfach oder auch gar nicht entsprechend ihrer Richtung durchlaufen werden können. „ (...) ^{0..*} “ bildet also die transitive Hülle der Assoziationen „ (...) “.

4.2.2 Generalisierung und Spezialisierung

In diesem Kapitel erweitern wir die im vorangehenden Kapitel dargestellte Modellierungstechnik um die Konzepte Generalisierung und Spezialisierung.

Das eingeführte Konzept der Instanziierung kann ebenfalls als eine Form der Spezialisierung eines Modells verstanden werden. Während das Konzept der Instanziierung genau eine Spezialisierung eines Modells erlaubt, ermöglicht das hier definierte Konzept die Spezialisierung eines Modells über eine beliebige Anzahl von Ebenen.

Das vorliegende Kapitel gliedert sich analog dem vorangehenden Kapitel in die Darstellung der abstrakten Syntax, der Semantik sowie der konkreten Syntax im Hinblick auf Generalisierung und Spezialisierung.

Abstrakte Syntax – Generalisierung und Spezialisierung

Abbildung 4–7 zeigt die abstrakte Syntax der Modellierungstechnik. Im Kontext von Generalisierung und Spezialisierung entspricht die abstrakte Syntax zugleich der semantischen Domäne, wie wir im Folgenden erläutern. Entsprechend den Konventionen der konkreten Syntax für die sukzessive Erweiterung von Modellen in Abbildung 4–6 (f) erweitern wir hier die abstrakte Syntax aus Abbildung 4–3 um die Konzepte Generalisierung und Spezialisierung.

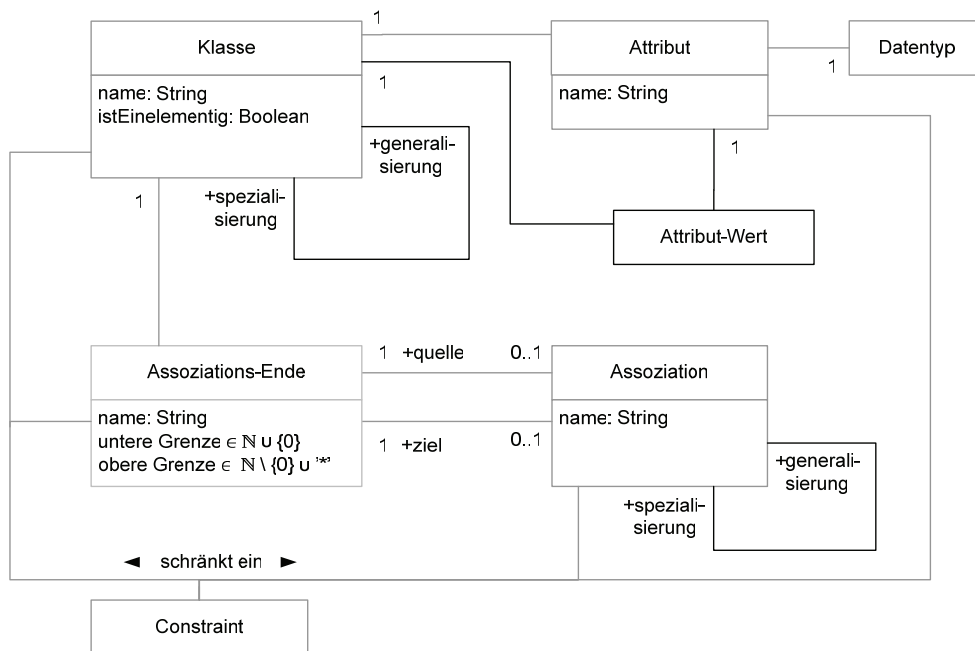


Abbildung 4–7: Abstrakte Syntax und semantische Domäne: Generalisierung und Spezialisierung

Klassen und Assoziationen können generalisiert und spezialisiert werden. Dabei kann es zu einer Klasse mehrere Generalisierungen geben. Im Kontext von Programmiersprachen würde man von „Mehrfachvererbung“ sprechen.

Zudem können Attribut-Werte im Kontext von Generalisierung und Spezialisierung nicht mehr nur als unmittelbare Instanzen von Attributen definiert werden. Vielmehr kann ein Attribut-Wert in jedem Modell gesetzt werden, das eine Spezialisierung der Klasse beinhaltet, der das Attribut zugeordnet ist.

Semantik – Generalisierung und Spezialisierung

Die Semantik der Modellierungstechnik definieren wir im Hinblick auf Generalisierung und Spezialisierung durch eine Abbildung der abstrakten Syntax in die abstrakte Syntax selbst. Die semantische Domäne ist hier also identisch mit der abstrakten Syntax. Die Semantik, die den Zusammenhang zwischen abstrakter Syntax und semantischer Domäne herstellt, ist hier identisch mit der Spezialisierung auf Klassen und Assoziationen.

Die folgenden Constraints formulieren Einschränkungen hinsichtlich der semantischen Interpretation. Die Definition der UML in [OMG03a] ist hinsichtlich dieser Constraints teilweise unpräzise, lässt also unterschiedliche Interpretationen zu.

(6)

Die Generalisierungsbeziehung auf Klassen darf genau wie die Generalisierungsbeziehung auf Assoziationen keine Zyklen enthalten:

**Klassen und
Assoziationen**

$$\forall k \in \text{Klasse}: \neg (k \in k.\text{generalisierung}.(\text{generalisierung})^{0..*}) \wedge$$

$$\forall a \in \text{Assoziation}: \neg (a \in a.\text{generalisierung}.(\text{generalisierung})^{0..*})$$

(7)

Assoziationen können nur dann in einer Generalisierungsbeziehung stehen, wenn die beteiligten Klassen entsprechend der Generalisierungshierarchie der jeweils assoziierten Klassen „kompatibel“ sind, wie in Abbildung 4–8 dargestellt:

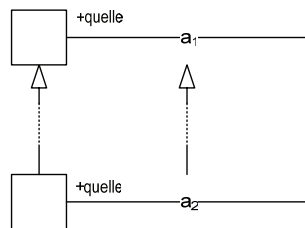


Abbildung 4–8: Visualisierung zu Constraint (7)

$$\forall a_1, a_2 \in \text{Assoziation}: a_2 \in a_1.\text{spezialisierung}:$$

$$a_2.\text{quelle.Klasse} \in a_1.\text{quelle.Klasse}.(\text{spezialisierung})^{0..*} \wedge$$

$$a_2.\text{ziel.Klasse} \in a_1.\text{ziel.Klasse}.(\text{spezialisierung})^{0..*}$$

(8)

Die definierten Multiplizitäten auf Assoziations-Enden gelten in allen spezialisierten Assoziationen, wie in Abbildung 4–9 dargestellt:

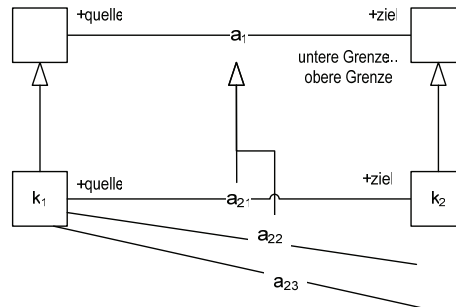


Abbildung 4–9: Visualisierung zu Constraint (8)

$$\forall a_1 \in \text{Assoziation}, k_1 \in a_1.\text{quelle.Klasse.spezialisierung}:$$

$$a_1.\text{ziel.untereGrenze} \leq$$

$$|\{ a_2 \in \text{Assoziation} \mid a_2 \in a_1.\text{spezialisierung} \wedge a_2.\text{quelle.Klasse} = k_1 \}|$$

$$\leq a_1.\text{ziel.obereGrenze}$$

$$\wedge$$

$$\forall k_2 \in a_1.\text{ziel.Klasse.spezialisierung}:$$

$$a_1.\text{quelle.untereGrenze} \leq$$

$$|\{ a_2 \in \text{Assoziation} \mid a_2 \in a_1.\text{spezialisierung} \wedge a_2.\text{ziel.Klasse} = k_2 \}|$$

$$\leq a_1.\text{quelle.obereGrenze}$$

(9)

Spezialisierte Assoziationen können über eingeschränktere Multiplizitäten verfügen. Multiplizitäten können aber durch Spezialisierung nicht abgeschwächt werden:

$$\forall a_1, a_2 \in \text{Assoziation}: a_2 = a_1.\text{spezialisierung}:$$

$$a_2.\text{ziel.untereGrenze} \geq a_1.\text{ziel.untereGrenze} \wedge$$

$$a_2.\text{ziel.obereGrenze} \leq a_1.\text{ziel.obereGrenze} \wedge$$

$$a_2.\text{quelle.untereGrenze} \geq a_1.\text{quelle.untereGrenze} \wedge$$

$$a_2.\text{quelle.obereGrenze} \leq a_1.\text{quelle.obereGrenze}$$

(10)

Ist eine Klasse als einelementig gekennzeichnet, gibt es höchstens eine unmittelbare Spezialisierung dieser Klasse. Gibt es eine unmittelbare Spezialisierung, muss diese ebenfalls als einelementig gekennzeichnet werden, um weitere Spezialisierungen einzuschränken:

$$\forall k_1 \in \text{Klasse}: |\{k_2 \in \text{Klasse} \mid k_2 = k_1.\text{spezialisierung}\}| \leq 1 \wedge$$

$$(k_1.\text{spezialisierung} \neq \emptyset \Rightarrow k_1.\text{spezialisierung.istEinelementig} = \text{True})$$

(11)

Attribut-Werte müssen den Klassen zugeordnet sein, die ein entsprechendes Attribut besitzen. Eine Klasse besitzt dabei alle Attribute ihrer generalisierten Klassen, wie in Abbildung 4–10 dargestellt:

**Attribute
und Attribut-
Werte**

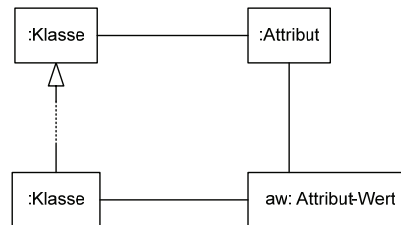


Abbildung 4–10: Visualisierung zu Constraint (11)

$$\forall aw \in \text{Attribut-Wert}: aw.\text{Attribut.Klasse} \in aw.\text{Klasse}.\text{(generalisierung)}^{0..*}$$

In Abbildung 4–10 müssen entsprechend den Vereinbarungen der konkreten Syntax keine Namen für instanziierte beziehungsweise spezialisierte Klassen angegeben werden.

(12)

Attribute können in spezialisierten Klassen nicht spezialisiert beziehungsweise „überschrieben“ werden. Der Name des Attributs fungiert als Identifikator eines Attributes. Die Attribute einer Klasse müssen andere Namen als die Attribute in allen ihren generalisierten Klassen haben:

$$\forall k \in \text{Klasse}: \\ (k.\text{Attribut.name} \cap k.\text{generalisierung}.\text{(generalisierung)}^{0..*}.\text{Attribut.name}) = \emptyset$$

(13)

Definiert eine Klasse den Wert eines Attributes, so kann eine Spezialisierung dieser Klasse den Attribut-Wert nicht neu definieren:

$$\forall aw_1 \in \text{Attribut-Wert}: \neg(\exists aw_2 \in \text{Attribut-Wert}: \\ aw_2.\text{Attribut} = aw_1.\text{Attribut} \wedge aw_2.\text{Klasse} \in aw_1.\text{Klasse}.\text{(spezialisierung)}^{0..*})$$

(14)

Constraints Analog zu Multiplizitäten als Einschränkungen über allen spezialisierten Assoziationen gelten Constraints ebenfalls für alle Spezialisierungen der jeweils eingeschränkten Elemente:

$$\begin{aligned} &\forall c \in \text{Constraint}, \\ &k_1 \in c.\text{schränkt ein}^{\text{Klasse}}, a_1 \in c.\text{schränkt ein}^{\text{Assoziation}}: \\ &\quad \forall k_2 \in k_1.(\text{spezialisierung})^{0..*}: k_2 \in c.\text{schränkt ein}^{\text{Klasse}} \wedge \\ &\quad \forall a_2 \in a_1.(\text{spezialisierung})^{0..*}: a_2 \in c.\text{schränkt ein}^{\text{Assoziation}} \end{aligned}$$

Constraints gelten also stets für alle Spezialisierungen. Entsprechend der hier definierten Modellierungstechnik sind Constraints nicht nur für eine Ebene formulierbar.

Unklarheiten der UML

In der UML bleiben die in den Constraints (8), (9) und (14) formulierten Zusammenhänge, die wir der Modellierungstechnik basierend auf Klassen und Constraints im Rahmen der Arbeit zu Grunde legen, unklar (vgl. [OMG03a]).

Konkrete Syntax – Generalisierung und Spezialisierung

In Abbildung 4–11 definieren wir die Konventionen der konkreten Syntax im Hinblick auf Generalisierung und Spezialisierung, soweit diese spezifisch für die vorliegende Arbeit sind. Die in Kapitel 4.2.1 dargestellte konkrete Syntax gilt im Kontext von Generalisierung und Spezialisierung dieses Kapitels entsprechend.

- Die Notation der UML für Generalisierung in Form eines nicht ausgefüllten Pfeils kann zur Verbesserung der Übersichtlichkeit eines Diagramms durch die Angabe des Namens der generalisierten Klasse, abgetrennt durch einen Doppelpunkt, ersetzt werden.
- Attribut-Werte zu Attributen vom Datentyp Boolean der Form „istXYZ“ werden in einer spezialisierten Klasse für den Wert „True“ als „<XYZ>“ notiert. Wird in einer spezialisierten Klasse kein Wert angegeben und wurde in einer vorangehenden Spezialisierung nicht bereits der Wert „True“ angegeben, so hat das jeweilige Attribut den Wert „False“.
- Wird ein Attribut-Wert in einer Klasse gesetzt, kann dieser Wert abkürzend für Spezialisierungen dieser Klasse nicht dargestellt werden.
- Einmal definierte Multiplizitäten können abkürzend in einer spezialisierten Assoziation ausgelassen werden. Entsprechend den Constraints der Modellierungstechnik gelten für eine spezialisierte Assoziation allerdings immer dieselben oder aber eingeschränktere Multiplizitäten (vgl. Constraint (9)).
- Multiplizitäten, die nicht der Einschränkungen unmittelbar spezialisierter Assoziationen dienen, sondern als Einschränkung von Spezialisierungen spezialisierter Assoziationen werden als umrandete Multiplizitäten notiert.
- Für spezialisierte Klassen wie für spezialisierte Assoziationen müssen in der konkreten Syntax nicht notwendigerweise Namen vergeben werden.

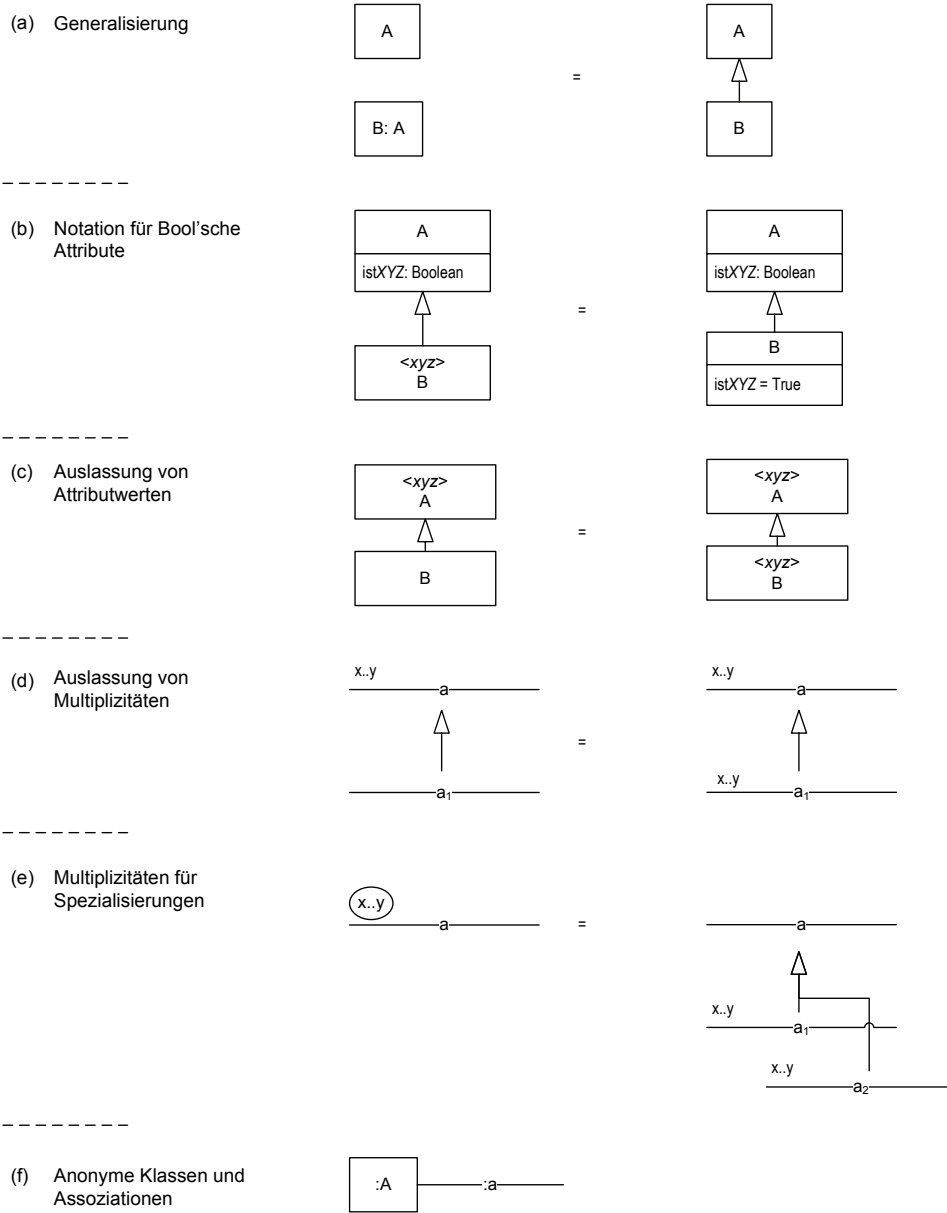


Abbildung 4–11: Konkrete Syntax: Generalisierung und Spezialisierung

4.3 Modellierungsansatz für Vorgehensmodelle und verwandte Ansätze

Im Allgemeinen kann zwischen dem Vorgehens-Meta-Modell als Sprache für die Beschreibung von Vorgehensmodellen, dem Vorgehensmodell und der Projektebene unterschieden werden (vgl. [GMP+03]). Der in der Arbeit verwendete Ansatz zur Modellierung von Vorgehensmodellen stützt sich dabei zur Abgrenzung dieser Meta-Ebenen auf das Konzept der Spezialisierung ab.

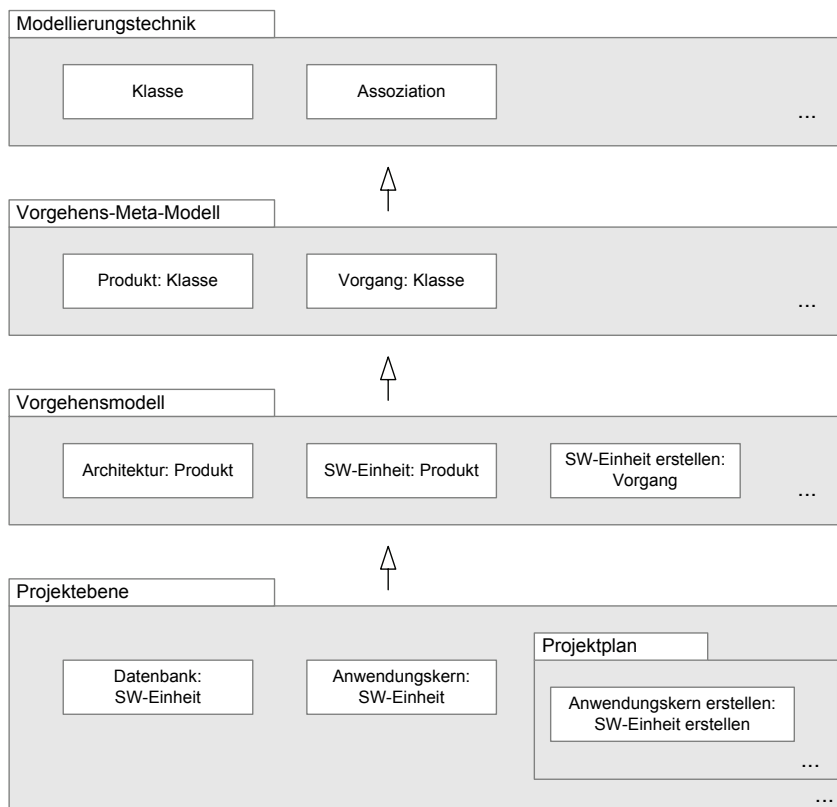


Abbildung 4–12: Modellierungsansatz: Spezialisierung als Metaisierungsprinzip

Wie in Abbildung 4–12 dargestellt definieren wir ein Vorgehens-Meta-Modell als Spezialisierung der abstrakten Syntax der im vorangehenden Kapitel definierten Modellierungstechnik, das heißt als Spezialisierung der Konzepte *Klasse*, *Assoziation* beziehungsweise Attribut.

Das Vorgehens-Meta-Modell dient dem Prozessingenieur als Sprache zur Beschreibung von Vorgehensmodellen. Im Vorgehens-Meta-Modell definieren wir Klassen wie beispielsweise *Produkt* oder *Vorgang*. Die Ebene des Vorgehensmodells enthält Spezialisierungen der Elemente des Vorgehens-Meta-Modells, beispielsweise *Architektur* und *SW-Einheit* als spezialisierte Produkte, sowie „*SW-Einheit erstellen*“ als spezialisierten Vorgang. Auf der Projektebene treten beispielsweise *Datenbank* und *Anwendungskern* als spezialisierte SW-Einheiten auf. Teile der Projektebene sind Bestandteil des Projektplans, wie beispielsweise der Vorgang zur Erstellung einer SW-Einheit „*Anwendungskern erstellen*“.

Da Spezialisierung und Generalisierung die definierte Modellierungstechnik in sich selbst abbilden, verwenden wir zur Beschreibung die im vorangehenden Kapitel definierte konkrete

Syntax auf allen Spezialisierungs-Ebenen. Besonderheiten der Architektur der Meta-Ebenen für die Modellierung von Vorgehensmodellen sowie des Spezialisierungskonzeptes untersuchen wir in Kapitel 5.4.

Die Problematik der Anwendung des Instanzierungs-Konzeptes mit der klassischen 2-Ebenen-Semantik im Kontext einer Meta-Modell-Architektur mit mehr als zwei Ebenen wird auch in [AK01] beleuchtet. Instanzierung bildet Klassen auf Objekte, Assoziationen auf Links und Attribute auf Attribut-Werte ab. Objekte und Links sind für eine weitere Instanzierung auf der jeweils nächsten Meta-Ebene nicht geeignet. Die Anwendung des Instanzierungs-Konzeptes auf mehr als zwei Meta-Ebenen führt wie in [AK01] gezeigt zu einer Reihe von Problemen. Typische Probleme sind eine nicht eindeutige Klassifizierung, das heißt eine "Mehrfach-Instanzierung" von Objekten oder aber die Wiederholung derselben Klassen auf unterschiedlichen Meta-Ebenen.

In [AK01] wird daher ein Konzept zur beliebig tiefen Instanzierung vorgestellt. Dabei werden Klassen einerseits durch die Meta-Ebene, auf der sie angesiedelt sind, und andererseits durch ihre "Potenz" attribuiert. Die Potenz einer Klasse legt die Anzahl der verbleibenden Instanzierungen fest, nimmt also bei jeder Instanzierung um Eins ab. Eine Klasse der Potenz Null entspricht in der klassischen Instanzierungs-Semantik einem Objekt, das nicht weiter instanziiert werden kann, oder aber einer als abstrakt markierten Klasse, die ebenfalls nicht weiter instanziiert werden kann. Für Attribute wird ebenfalls eine Potenz festgelegt, die die Anzahl der Instanzierungen bis zur endgültigen Belegung des Attributes mit einem Attribut-Wert angibt.

Der in der vorliegenden Arbeit verwendete Ansatz, ein Spezialisierungskonzept als Übergang zwischen Meta-Ebenen auszunutzen, ähnelt diesem Ansatz. Spezialisierung erlaubt ebenfalls eine beliebige Anzahl von Meta-Ebenen. Die Potenz einer Klasse beziehungsweise Einschränkungen bezüglich der Anzahl der verbleibenden Spezialisierungen modellieren wir in der vorliegenden Arbeit durch Constraints.

Mit Hilfe von Constraints wie dem folgenden können weitere Spezialisierungen eines beliebigen Konzeptes auf einer bestimmten Meta-Ebene unterbunden werden:

$$\forall k \in \text{Konzept}: k.\text{spezialisierung.spezialisierung} = \emptyset$$

Die Zugehörigkeit einer Klasse beziehungsweise Assoziation zu einer bestimmten Meta-Ebene kennzeichnen wir für die Ebene des Vorgehens-Meta-Modells explizit. Die Spezialisierungen des Vorgehens-Meta-Modells bezeichnen wir entweder als Vorgehensmodell oder als Projektebene. In den folgenden Kapiteln motivieren wir die Festlegung der Projektebene als jeweils "tiefste" Spezialisierung der Konzepte des Vorgehens-Meta-Modells (vgl. dazu Kapitel 5.4).

**Verwandte
Ansätze zur
Meta-Model-
lierung**

5 Modellierung von Vorgehensmodellen

Das Kapitel widmet sich dem Kernthema der Arbeit. Wir entwerfen sukzessive entsprechend der in Kapitel 4 definierten Modellierungstechnik ein Vorgehens-Meta-Modell, das die Anwendung im Rahmen der Projektplanung und die Unterstützbarkeit durch Werkzeuge für die Projektplanung erlaubt (vgl. dazu Kapitel 6). Wir nehmen in diesem Kapitel nicht das Meta-Modell des V-Modells XT als Ausgangspunkt, sondern konzipieren ein Meta-Modell von Grund auf neu. Den möglichen Bezug zum V-Modell XT stellen wir lediglich als Ausblick auf weitere Arbeiten in Kapitel 7 dar.

In Kapitel 5.1 geben wir zunächst einen Überblick über die Unterteilung der gesamten Modellierung in Teil-Modelle. Diese Architektur spiegelt sich in der Aufteilung der folgenden Unterkapitel wider. In Kapitel 5.2 erläutern wir das Strukturmodell, sowie in Kapitel 5.3 das dynamische Modell.

In Kapitel 5.4 beleuchten wir Anwendungsmöglichkeiten des Konzeptes der Spezialisierung, beispielsweise zur Modellierung organisationsspezifischer Ausprägungen oder so genannter generischer Vorgehensweisen.

Wir illustrieren die Anwendung aller Konzepte des Vorgehens-Meta-Modells anhand eines umfangreichen durchgängigen Beispiels auf Ebene des Vorgehensmodells sowie auf Projektebene. Kapitel 5.5 schließlich fasst die wesentlichen Inhalte zusammen.

Inhalt des Kapitels	Seite
5.1 Die Modelle im Überblick	107
5.2 Strukturmodell	110
5.3 Dynamisches Modell	127
5.4 Spezialisierungskonzept	142
5.5 Zusammenfassung	154

5.1 Die Modelle im Überblick

Im diesem Kapitel geben wir einen Überblick über die Aufteilung der gesamten Modellierung in Teil-Modelle. Abbildung 5–1 zeigt die Architektur der Modellierung, die sich auch in der Struktur der folgenden Unterkapitel widerspiegelt.

Wir unterteilen das Vorgehens-Meta-Modell und damit auch die Beispiele auf Ebene des Vorgehensmodells und auf Projektebene in ein *Strukturmodell* und ein *dynamisches Modell*. Strukturmodell und dynamisches Modell unterteilen wir orthogonal dazu in ein *Planungsmodell* und ein *Produktmodell*. Auf Ebene des Vorgehensmodells und der Projektebene geben wir für alle Aspekte des Vorgehens-Meta-Modells jeweils Beispiele zur Illustration der Konzepte an. Während wir im Strukturmodell strukturelle Aspekte beschreiben gehen wir im dynamischen Modell auf die Aspekte ein, die der Durchführung von Vorgängen beziehungsweise der Erstellung der Projektergebnisse eine zeitliche Reihenfolge geben.

Das Strukturmodell beschreiben wir in Kapitel 5.2, das dynamische Modell in Kapitel 5.3, das Planungs- beziehungsweise Produktmodell in den jeweiligen Unterkapiteln. Das *Modell zur inhaltlichen Strukturierung* beschreiben wir im Kontext der Erstellung und Anwendung von Vorgehensmodellen in Kapitel 6.1 sowie das *Tailoring-Modell* in Kapitel 6.2.

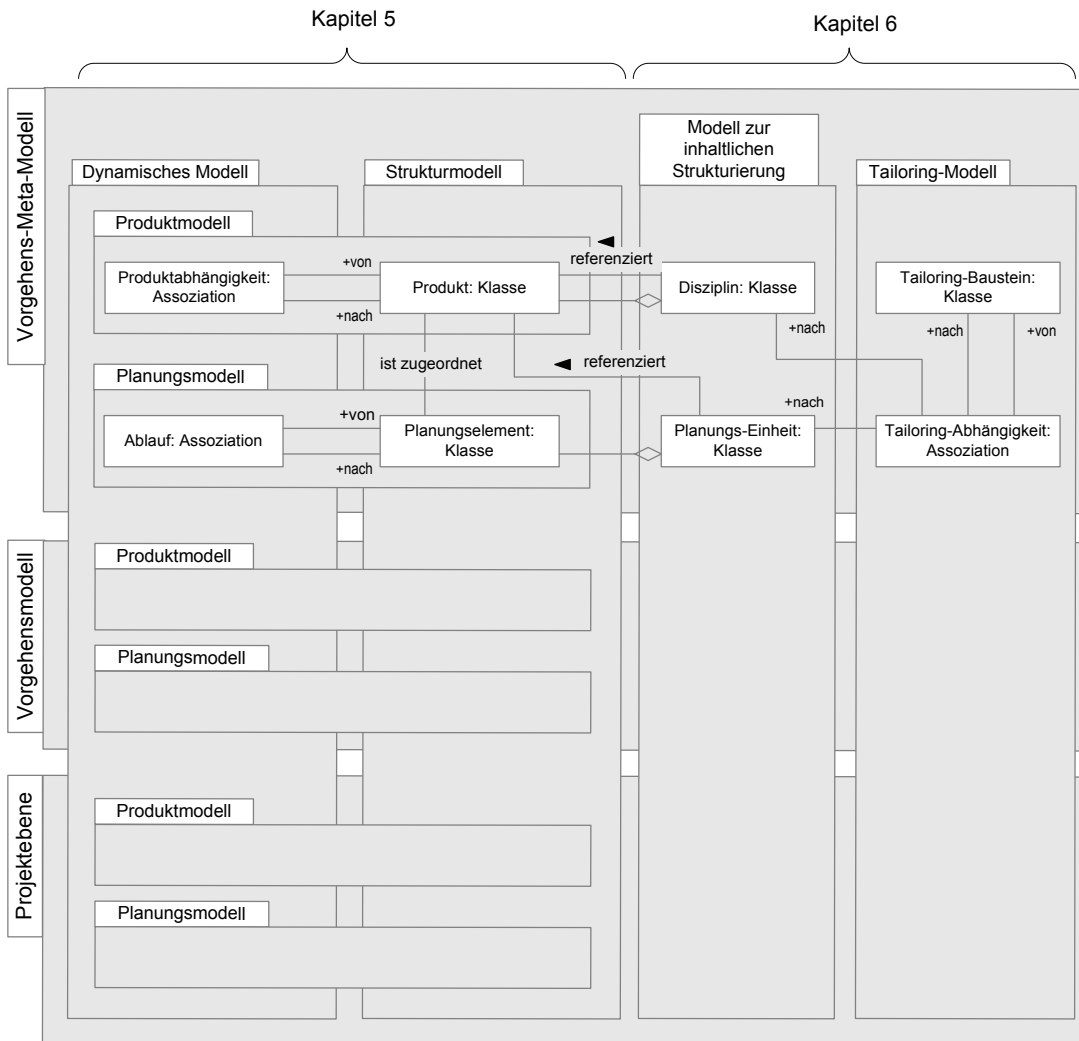


Abbildung 5–1: Architektur der Gesamt-Modellierung

Neben einer Strukturierung in Form von Paketen, die der Kapitelstruktur entspricht, zeigt Abbildung 5–1 eine abstrakte Sicht auf die in den Paketen enthaltenen Klassen. Dabei handelt es sich um abstrakte Klassen, die wir in den folgenden Kapiteln auf Ebene des Vorgehens-Meta-Modells weiter spezialisieren. Das Modell zur inhaltlichen Strukturierung enthält weitere abstrakte Klassen, die der Modellierung von Struktur dienen, in der abstrakten Sicht in Abbildung 5–1 aber nicht dargestellt sind. Eine Darstellung aller Klassen des Vorgehens-Meta-Modells kann dem Anhang entnommen werden (vgl. dazu Anhang A).

Produkte wie beispielsweise Dokumente oder Realisierungselemente sind Planungselementen *zugeordnet*. *Planungselemente* können unter anderem Vorgänge zur Produkterstellung, Meilensteine oder Projektabschnitte sein. Die Beziehung „*ist zugeordnet*“ wird dabei jeweils entsprechend spezialisiert. Das dynamische Modell enthält zum einen *Produktabhängigkeiten* wie Produktfluss oder Konsistenzabhängigkeit als Beziehungen zwischen Produkten. Zum anderen enthält das dynamische Modell *Abläufe* wie beispielsweise Ende-Anfang-Vorgangfolgen, die Planungselemente in Beziehung setzen.

Das Modell zur inhaltlichen Strukturierung beinhaltet Disziplinen und Planungs-Einheiten. *Disziplinen* aggregieren Produkte, *Planungs-Einheiten* aggregieren Planungselemente.

Während Produkte beziehungsweise Planungselemente über Aggregation in genau einer Disziplin beziehungsweise Planungs-Einheit enthalten sind, können Disziplinen beziehungsweise Planungs-Einheiten Produkte ohne Einschränkung *referenzieren*. Referenzierung dient der Formulierung inhaltlicher Überlappungen in der gewählten inhaltlichen Strukturierung. Die Möglichkeit der hierarchischen Strukturierung von Disziplinen und Planungs-Einheiten ist in Abbildung 5–1 nicht dargestellt.

Das Tailoring-Modell besteht aus *Tailoring-Bausteinen* und *Tailoring-Abhängigkeiten*. Disziplinen und Planungs-Einheiten dienen als Einheiten, die im Rahmen des Tailoring projektspezifisch gewählt oder nicht gewählt werden können. Tailoring-Abhängigkeiten setzen Tailoring-Bausteine untereinander oder aber Tailoring-Bausteine mit Disziplinen beziehungsweise Planungs-Einheiten in Beziehung. Mit Hilfe von Tailoring-Abhängigkeiten wie Aggregation, „optionale Auswahl“ oder „exklusive Auswahl“ lassen sich Einschränkungen möglicher projektspezifischer Auswahlmengen von Disziplinen oder Planungs-Einheiten formulieren.

Weiterer zentraler Aspekt ist der Einsatz eines Spezialisierungskonzeptes auf Ebene des Vorgehensmodells. Da dabei keine Erweiterung des Vorgehens-Meta-Modells erforderlich ist, schlägt sich dieser Aspekt nicht in Abbildung 5–1 nieder. Kapitel 5.4 beleuchtet dieses Spezialisierungskonzept im Zusammenhang mit dem Modellierungsansatz der Unterteilung in (Meta-)Ebenen. Die Ebene des Vorgehensmodells kann selbst in mehrere Ebenen zerfallen, da innerhalb eines Vorgehensmodells die Beschreibung von Spezialisierungs-Beziehungen erforderlich werden kann. Spezialisierung im Vorgehensmodell kann der Vermeidung von Redundanz oder der Modellierung organisations- oder domänenspezifischer Ausprägungen dienen. Zum anderen sind Beziehungen von der Ebene des Vorgehensmodells auf Klassen des Vorgehens-Meta-Modells erforderlich, wie wir in Kapitel 5.4 im Kontext generischer Vorgehensweisen wie der Qualitätssicherung zeigen.

Das Vorgehens-Meta-Modell illustrieren wir in den folgenden Abschnitten anhand eines durchgängigen Beispiels. Dieses bewegt sich sowohl auf Ebene des Vorgehensmodells als auch auf Projektebene. Als Beispiel wählen wir passend zu der vorliegenden Arbeit ein Vorgehensmodell für die Erstellung von Vorgehensmodellen beziehungsweise ein „Entwicklungsprojekt“ für Vorgehensmodelle. Wir nennen den Entwicklungsgegenstand dabei abkürzend „VGM“. Das Beispielprojekt weist dabei zum Teil Ähnlichkeit mit dem Projekt WEIT auf, welches die Erstellung des V-Modells XT zum Ziel hatte (vgl. [WEIT04]).

**Durchgängiges Beispiel:
VGM**

Wie die folgenden Kapitel zeigen, ist die Komplexität der Erstellung eines Vorgehensmodells durchaus vergleichbar mit der Komplexität der Entwicklung eines größeren Softwareentwicklungs-Projektes. Das Beispiel stellt sich daher für die Validierung der Ausdrucksmächtigkeit der konzipierten Beschreibungsmittel als durchaus geeignet dar.

Neben der Ausdrucksmächtigkeit ist die Angemessenheit beziehungsweise der Praxisbezug der Beschreibungsmittel sicherzustellen. Meta-Modelle laufen leicht Gefahr zu künstlichen Gebilden ohne Bezug zu einer praktischen Problemstellung zu werden. Die Eleganz eines Meta-Modells wird dabei manchmal höher als seine Angemessenheit eingestuft. Wir verfolgen für die Erstellung des Vorgehens-Meta-Modells einen Bottom-Up-Ansatz. Wir gehen also jeweils von einem Beispiel auf der Projektebene aus und abstrahieren von diesem. Im Vorgehens-Meta-Modell der vorliegenden Arbeit findet sich somit kein Konzept ohne zugehöriges Beispiel. Lediglich die Gliederung der folgenden Kapitel wählen wir anders. In den folgenden Kapiteln stellen wir das Vorgehens-Meta-Modell den Beispielen der Ebene des Vorgehensmodells und der Projektebene voran, da dieser Aufbau für den Leser mehr Übersichtlichkeit mit sich bringt.

Kein Konzept ohne Beispiel

5.2 Strukturmodell

Im Strukturmodell betrachten wir Elemente und Aggregations-Beziehungen des Planungsmodells und des Produktmodells, sowie den Zusammenhang beider Modelle. Wir betrachten also strukturelle Aspekte von Vorgehensmodellen.

Planungs- und Produktmodell werden in den folgenden Kapiteln sukzessive aufgebaut. Kapitel 5.2.1 führt die grundlegenden Aspekte des Planungsmodells ein. Kapitel 5.2.2 definiert das Produktmodell. Kapitel 5.2.3 stellt den Zusammenhang zwischen Planungs- und Produktmodell her. Kapitel 5.2.4 ergänzt das Planungsmodell um Elemente zur Strukturierung des Projektablaufs.

Abschnitte in Kapitel 5.2		Seite
5.2.1	Planungsmodell – Planungselemente	110
5.2.2	Produktmodell – Produkte	112
5.2.3	Planungsmodell – Planung der Produkterstellung	116
5.2.4	Planungsmodell – Strukturierung des Projektablaufs	122

5.2.1 Planungsmodell – Planungselemente

Unter dem Planungsmodell verstehen wir in Abgrenzung zum Produktmodell die Konzepte, die bei Anwendung eines Vorgehensmodells unmittelbar in den Projektplan einfließen. In den meisten verbreiteten Vorgehensmodellen wird das Planungsmodell als Aktivitätenmodell bezeichnet. Zielsetzung dieses Kapitels ist die Definition struktureller Aspekte eines Projektplans auf Ebene des Vorgehens-Meta-Modells sowie eines Beispiels auf Projektebene.

Vorgehens-Meta-Modell

Abbildung 5–2 zeigt die strukturellen Aspekte des Planungsmodells. Dieser Ausschnitt des Vorgehens-Meta-Modells kann ebenso als Modell eines Projektplans gelten. Für ein Vorgehensmodell als Spezialisierung dieses Meta-Modells kann also die Ableitung eines Projektplans als trivial bezeichnet werden. Die Abstraktionsprinzipien, die die Ebene des Vorgehensmodells von der der Projektplanung unterscheiden, sowie weitere Elemente des Vorgehens-Meta-Modells wie Produkte, Produktabhängigkeiten oder Abläufe ergänzen wir in den folgenden Kapiteln.

Planungselemente leiten sich vom Konzept der Klasse der Modellierungstechnik ab (vgl. Kapitel 4). Planungselemente treten in den Spezialisierungen *Vorgang* und *Meilenstein* auf. Abstrakte Klassen stellen entsprechend der definierten Modellierungstechnik für Klassendiagramme eine abkürzende Syntax dar (vgl. Kapitel 4).

Vorgänge sind als zeitlich zu planende Tätigkeiten mit einem Anfangs- und Endtermin zu verstehen (vgl. Kapitel 2.4). Termine als Attribute eines Vorgangs stellen wir in Abbildung 5–2 abkürzend nicht dar. Vorgänge sind über Aggregation mit Planungselementen verbunden. Ein Vorgang kann also selbst wieder Vorgänge oder Meilensteine beinhalten. Anders als in Kapitel 2.4 modellieren wir Sammelvorgänge hier nicht als eigenes Konzept.

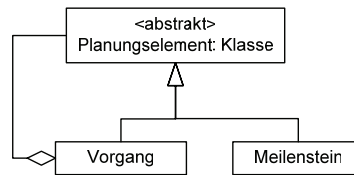


Abbildung 5–2: Planungsmodell – Ebene Vorgehens-Meta-Modell

Ein Meilenstein ist definiert als ein geplanter Prüfzeitpunkt (vgl. Kapitel 2.4). Meilensteine sind also mit genau einem Termin attribuiert. Geprüft wird zu diesem Termin das Erreichen eines gesetzten Ziels in Form eines Projektergebnisses. Für die Termine der Planungselemente gilt das folgende Constraint.

(15)

Alle Planungselemente ordnen sich zeitlich zwischen Anfangs- und End-Termin der Vorgänge ein, in denen sie über Aggregation als Teile enthalten sind.

Die zeitliche Spanne eines Vorgangs umfasst also mindestens die zeitlichen Spannen aller seiner Teil-Vorgänge, kann aber auch länger sein.

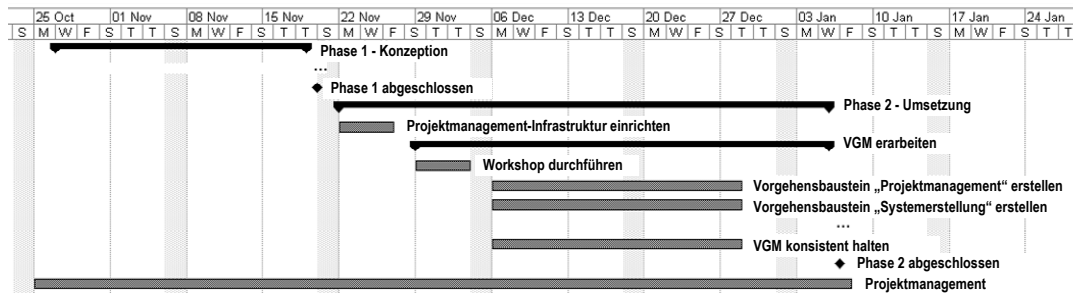
Beispiel: Projektebene

Wir illustrieren das Vorgehens-Meta-Modell anhand eines einfachen Beispiels auf der Projektebene. Dieses Beispiel verwenden wir als durchgängiges Beispiel auch in den folgenden Kapiteln. Abbildung 5–3 (a) zeigt einen Projektplan in der Notation eines Balkendiagramms. Sammelvorgänge sind dabei als Klammern dargestellt. Abbildung 5–3 (b) stellt dieselbe Information als Instanz beziehungsweise Spezialisierung des Vorgehens-Meta-Modells dar. Die zeitlichen Festlegungen in Form von Attribut-Werten für Termine sind nicht dargestellt.

Als „VGM“ bezeichnen wir das entsprechend unserem durchgängigen Beispiel zu erstellende Vorgehensmodell. Das Beispiel-Projekt ist in zwei Phasen aufgeteilt, die jeweils mit einem Meilenstein abgeschlossen werden. Die zweite Phase ist mit den enthaltenen Vorgängen detailliert dargestellt. Dem „*Einrichten der Projektmanagement-Infrastruktur*“ folgt das „*Erarbeiten des VGM*“, das beispielsweise die „*Durchführung von Workshops*“ sowie die „*Erstellung der Vorgehensbausteine Projektmanagement*“ und „*Systemerstellung*“ beinhaltet. Die „*Konsistenzsicherung des VGM*“ ist als eigener Vorgang enthalten. Der Vorgang *Projektmanagement* erstreckt sich als querschnittlicher Vorgang von Projektbeginn bis Projektende.

Das Vorgehens-Meta-Modell bringt in dieser Form bislang einzig den Vorteil der einfachen Ableitbarkeit eines Projektplans mit sich. Vorgehensmodell und Projektplan können hier als identisch gesehen werden. Die Vorteile eines Vorgehensmodells als Abstraktion der Projektebene sind noch nicht gegeben. Die Terminologie der Vorgänge in Abbildung 5–3 ist uneinheitlich, da die Erstellung des Projektplans im Beispiel auf der Intuition des Projektleiters beruht. Der Plan lässt allerdings ein Schema vermuten, da beispielsweise Phasen, Meilensteine zum Abschluss von Phasen, sowie Vorgänge zur Erstellung von Vorgehensbausteinen mehrfach auftreten. Dieses Schema legt die Generalisierung des Projektplans beziehungsweise eine Modellbildung in Form eines Vorgehensmodells nahe. In den folgenden Kapiteln erweitern wir das Vorgehens-Meta-Modell um entsprechende Abstraktionsmöglichkeiten.

(a) Projektplan als Balkendiagramm



(b) Projektplan als Instanz des Vorgehens-Meta-Modells

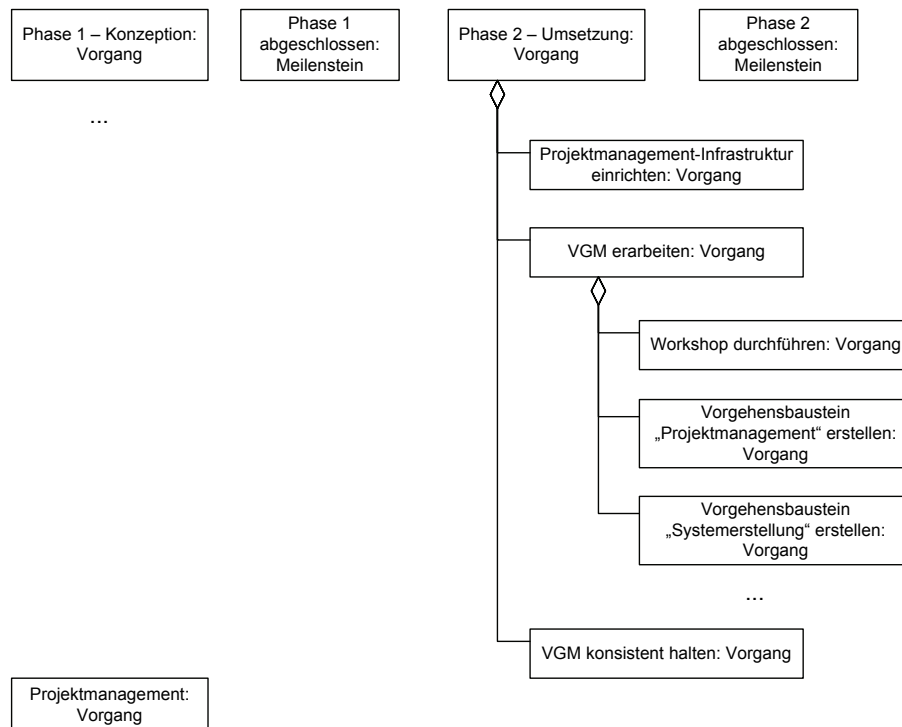


Abbildung 5-3: Planungsmodell – Ebene Projekt (Beispiel)

5.2.2 Produktmodell – Produkte

Das Produktmodell enthält die Konzepte des Vorgehens-Meta-Modells, die der Beschreibung von Projektergebnissen dienen. Planung bezieht sich, von querschnittlichen Aufgaben wie dem Projektmanagement abgesehen, auf die Erstellung von Projektergebnissen. Projektergebnisse werden im Kontext von Vorgehensmodellen Produkte genannt.

Zielsetzung dieses Kapitels ist die Definition unterschiedlicher Arten von Produkten sowie des strukturellen Aufbaus von Produkten auf Ebene des Vorgehens-Meta-Modells. Produkte

und ihr struktureller Aufbau sind entsprechend der Zielsetzung der Arbeit insoweit relevant, als sie die Planung beeinflussen.

Vorgehens-Meta-Modell

Abbildung 5–4 zeigt die strukturellen Aspekte des Produktmodells. Bereits definierte Aspekte des Vorgehens-Meta-Modells werden der Modellierungstechnik der Arbeit entsprechend ausgegraut dargestellt.

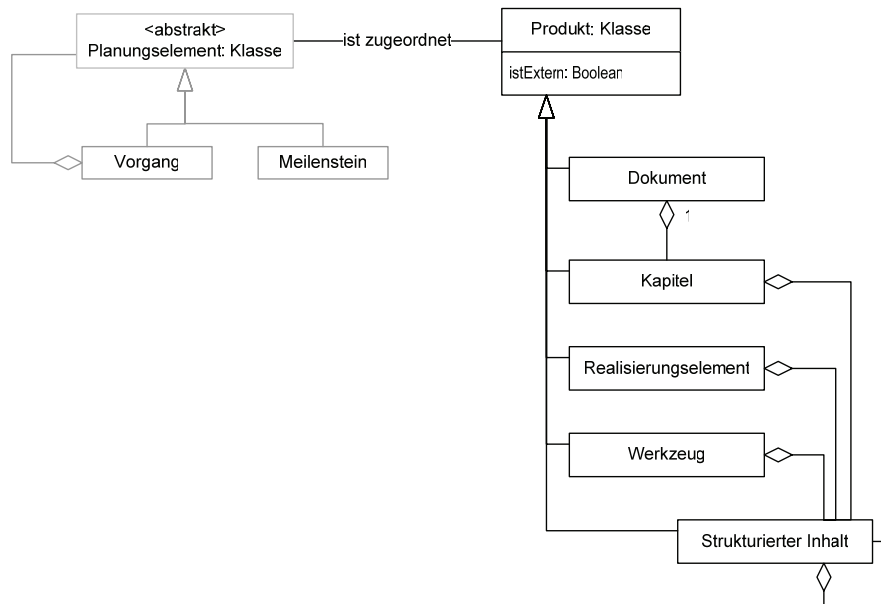


Abbildung 5–4: Produktmodell – Ebene Vorgehens-Meta-Modell

Produkte sind die zu erstellenden Ergebnisse eines Projektes. Die Erstellung eines Projektplans bedeutet größtenteils die Planung der Produkterstellung. Wir modellieren Produkte in unterschiedlichen Spezialisierungen. Ein *Dokument* besteht aus *Kapiteln*. Die Kapitel sind dabei eindeutig einem Dokument zugeordnet. Unter einem *Realisierungselement* verstehen wir den Entwicklungsgegenstand oder einen Teil des Entwicklungsgegenstands eines Projektes, also beispielsweise eine SW-Komponente oder eine HW-Komponente. Ein *Werkzeug* ist eine Beschreibung eines im Projekt einzusetzenden Werkzeugs.

Kapitel, Realisierungselemente und Werkzeuge können „strukturierte Inhalte“ enthalten, die selbst wieder in Form von Aggregation strukturiert sein können. „Strukturierte Inhalte“ modellieren wir als spezialisierte Produkte. Strukturierte Produktinhalte sind im Rahmen der Arbeit insoweit relevant, als sie für die Ableitung eines Projektplans aus einem Vorgehensmodell von Bedeutung sind (vgl. dazu Kapitel 5.4.1). Einen „strukturierten Inhalt“, der Teil eines Werkzeugs ist, verstehen wir dabei selbst als Werkzeug. Aggregation zwischen Dokumenten, Realisierungselementen und Werkzeugen modellieren wir nicht, da wir dies nicht als relevant für typische Planungen erachten.

(16)

„Strukturierte Inhalte“ sind Teil genau eines anderen Produktes:

$$\forall si \in \text{Strukturierter Inhalt: } | si.\text{enthält}^{\text{ganzes}} | = 1$$

Aggregation stellt entsprechend der Vereinbarung der konkreten Syntax der eingesetzten Modellierungstechnik eine Assoziation mit dem Namen „enthält“ und den Assoziations-Enden „Ganzes“ und „Teil“ dar. Aggregation wird abkürzend durch eine Assoziation mit Rauten-Symbol dargestellt (vgl. Kapitel 4).

Produkte können über ein Attribut als *extern* markiert werden. Externe Produkte sind Produkte, die nicht im Projekt zu erstellen sind, sondern wie beispielsweise Fertigprodukte zugekauft werden.

Produkte sind schließlich ohne Einschränkungen in Form von Multiplizitäten Planungselementen *zugeordnet*. Diese zunächst unspezifische Zuordnung zwischen Planungselementen und Produkten konkretisieren wir in den folgenden Kapiteln.

Beispiel: Vorgehensmodell und Projektebene

Wir illustrieren die Konzepte des Produktmodells in Abbildung 5–5 anhand eines Beispiels auf Ebene des Vorgehensmodells und einer möglichen Spezialisierung dieses Vorgehensmodells auf Projektebene. Eine projektspezifische Spezialisierung des Produktmodells eines Vorgehensmodells nennen wir Produktstruktur. Als durchgängiges Beispiel betrachten wir das Entwicklungsprojekt für das Vorgehensmodell VGM des vorangehenden Abschnitts. Abbildung 5–5 (a) zeigt die Ebene des Vorgehensmodells und Abbildung 5–5 (b) die Projektebene.

Das Produkt *Projektmanagement-Infrastruktur* modellieren wir als Werkzeug im Vorgehensmodell. Auf Projektebene kann es mehrere Ausprägungen der Projektmanagement-Infrastruktur wie beispielsweise einen *Emailverteiler* oder ein *Repository* geben. Diese Spezialisierungen auf Projektebene entsprechen installierten Werkzeugen.

Das Dokument *Strategiepapier* enthält beispielsweise eine Beschreibung der Projektorganisation sowie eine Konzeption der Inhalte des Vorgehensmodells. Klassen wie das Strategiepapier können entsprechend der Modellierungstechnik als *einelementig* gekennzeichnet werden. Für die Beschreibung eines Vorgehensmodells steht dieses Sprachmittel, das der Einschränkung der Projektebene dient, ebenfalls zur Verfügung.

Im Falle des Strategiepapiers gibt das Vorgehensmodell keine Kapitelstruktur vor. Diese kann durch den Projektleiter auf Projektebene definiert werden. Die Kapitelstruktur ist beispielsweise dann für die Projektplanung relevant, wenn für die Erstellung der einzelnen Kapitel unterschiedliche Bearbeiter koordiniert werden müssen. Allerdings kann nicht für alle Dokumente eine für alle Projekte gleichermaßen passende Kapitelstruktur angegeben werden. Die Projektebene kann also unmittelbar Spezialisierungen des Konzeptes Kapitel des Vorgehens-Meta-Modells enthalten. Kapitel sind dabei lediglich ein Beispiel.

Kapitel können auf Projektebene mehrfach ausgeprägt werden, wie im Beispiel am Kapitel *Grundlagen* des Dokuments VGM auf Projektebene zu sehen ist. Auf Projektebene planen wir die Grundlagenkapitel *„Erläuterung der Grundkonzepte“* sowie *Managementmechanismen*.

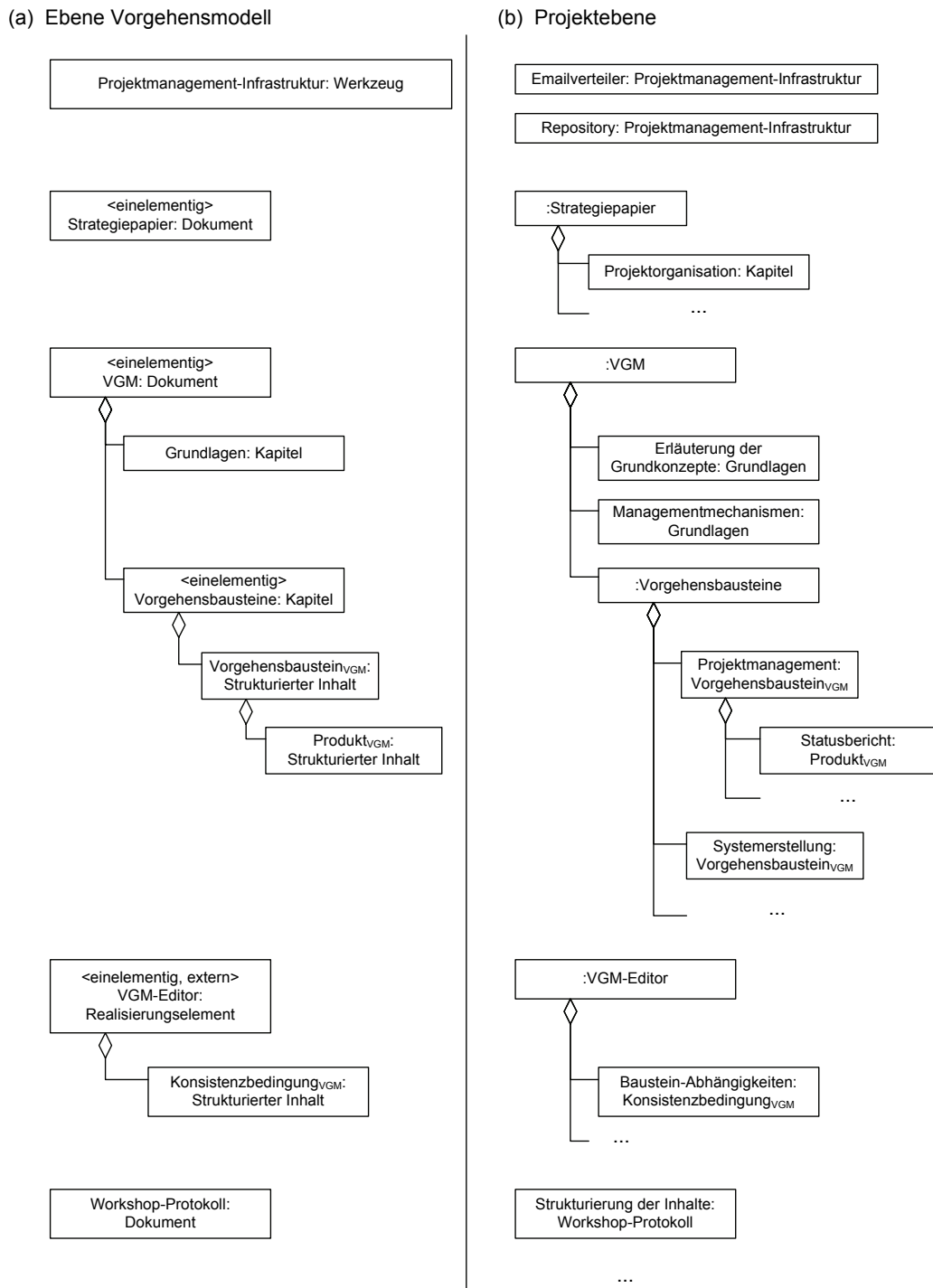


Abbildung 5–5: Produktmodell – Ebene Vorgehensmodell und Projektebene (Beispiel)

Das Dokument *VGM* besitzt neben einer Kapitelstruktur „strukturierte Inhalte“. Das VGM besteht im Beispiel aus *Vorgehensbausteinen*, Vorgehensbausteine wiederum bestehen aus *Produkten*. Zur Kennzeichnung von Begriffen, die dem Beispielprojekt zur Erstellung des Vorgehensmodells *VGM* beziehungsweise dem Beispiel-Meta-Modell des *VGM* entstam-

men, verwenden in Abbildung 5–5 das Suffix VGM. Auf Projektebene sind *Projektmanagement* und *Systemerstellung* Beispiele für Vorgehensbausteine des VGM. Der *Statusbericht* ist ein Beispiel für ein Produkt des VGM, das dem Vorgehensbaustein Projektmanagement zugeordnet ist.

Den Editor zur Erstellung des VGM modellieren wir als Realisierungselement, das extern entwickelt wird. Der *VGM-Editor* beinhaltet zu implementierende *Konsistenzbedingungen*, die nicht extern entwickelt werden sollen, sondern Entwicklungsgegenstand des Beispielprojektes sind. Eine Konsistenzbedingung könnte beispielsweise die Abhängigkeiten der Vorgehensbausteine des VGM einschränken.

Workshop-Protokolle schließlich sind als Dokumente modelliert. Workshop-Protokolle unterstützen als Hilfsprodukte die Erstellung der eigentlichen Projektergebnisse (vgl. dazu Kapitel 5.2.3).

Zusammenfassung

Das Konzept Produkt modellieren wir im Vorgehens-Meta-Modell in den Spezialisierungen Dokument, Kapitel, Realisierungselement und Werkzeug. Produktinhalte in Form „strukturierter Inhalte“ können im Vorgehensmodell ebenfalls beschrieben werden.

Die Projektebene besteht aus Spezialisierungen der Konzepte des Vorgehensmodells. Bei der Erstellung einer Planung auf der Projektebene können allerdings auch Konzepte des Vorgehens-Meta-Modells wie beispielsweise Kapitel unmittelbar spezialisiert werden. Die Projektplanung wird also nicht durch das Vorgehensmodell eingeschränkt, sondern gestaltet sich flexibel. Auch auf Ebene des Vorgehensmodells können alle sprachlichen Mittel eingesetzt werden, die sich aus der Modellierungstechnik ergeben. Beispielsweise können Produkte als einelementig gekennzeichnet werden.

5.2.3 Planungsmodell – Planung der Produkterstellung

In diesem Kapitel erweitern wir das Planungsmodell um Spezialisierungen des Konzeptes Vorgang, die sich für die präzise Beschreibung der Erstellung von Produkten eignen. Zielsetzung des Kapitels ist die Präzisierung der Schnittstelle „ist zugeordnet“ zwischen Produkten und Planungselementen des vorangehenden Kapitels. Dabei sind im Vorgehens-Meta-Modell spezialisierte Vorgänge zu definieren, die die Produkterstellung und die Produktbearbeitung im Planungsmodell geeignet strukturieren.

Vorgehens-Meta-Modell

Abbildung 5–6 zeigt die Erweiterungen des Planungsmodells auf Ebene des Vorgehens-Meta-Modells, die der Planung von Produkterstellung und -bearbeitung dienen. Klassen beziehungsweise Assoziationen, deren Existenz sich durch Constraints aus anderen Klassen beziehungsweise Assoziationen ableitet, werden der verwendeten Modellierungstechnik entsprechend mit unterbrochenen Linien dargestellt.

Anders als im Fall der Spezialisierung von abstrakten Klassen wie *Planungselement* handelt es sich bei der Spezialisierung der Klasse Vorgang nicht um eine Konvention der konkreten Syntax, sondern um eine Anwendung des eigentlichen Spezialisierungskonzeptes.

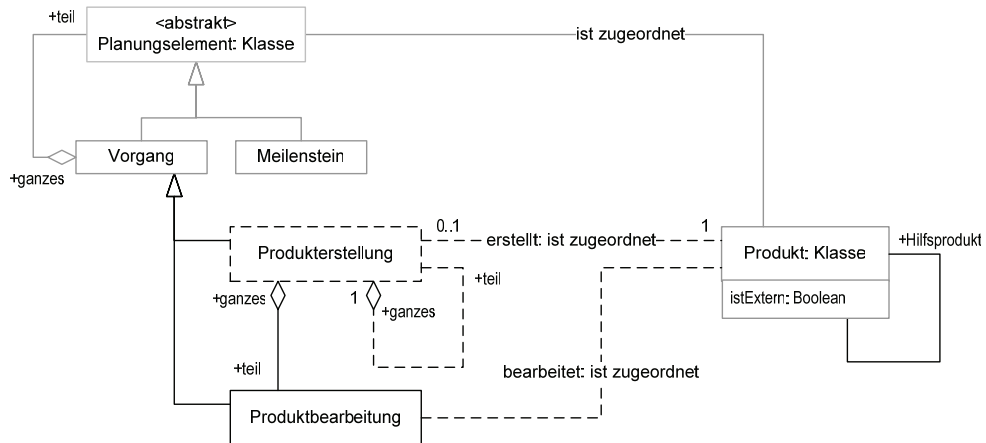


Abbildung 5–6: Planung der Produkterstellung – Ebene Vorgehens-Meta-Modell

Vorgänge können in den Spezialisierungen *Produkterstellung* und *Produktbearbeitung* auftreten. Für diese Ausprägungen von Vorgängen wird in Form der Assoziationen *erstellt* beziehungsweise *bearbeitet* jeweils der Bezug zu Produkten angegeben. Die Assoziationen *erstellt* beziehungsweise *bearbeitet* sind Spezialisierungen der Assoziation „*ist zugeordnet*“ zwischen *Planungselementen* und *Produkten*. Eine *Produkterstellung* erstellt entsprechend der angegebenen Multiplizität immer genau ein Produkt.

(17)

Allen Produkten, die nicht als extern markiert sind, ist ein Vorgang *Produkterstellung* zugeordnet. Diese *Produkterstellung* folgt einem festen Namensschema:

$$\forall p \in \text{Produkt}: p.\text{istExtern} = \text{False} \Rightarrow \exists v \in \text{Produkterstellung}: v.\text{erstellt} = p \wedge v.\text{name} = p.\text{name} + \text{„erstellen“}$$

Für externe Produkte ließen sich ebenfalls spezialisierte Vorgänge angeben, die nicht der Erstellung, sondern beispielsweise der Bereitstellung oder Beschaffung dienen.

Produkterstellungen sind über Aggregation assoziiert, *Produkterstellungen* und *Produktbearbeitungen* sind ebenfalls über Aggregation assoziiert. Dabei handelt es sich um Spezialisierungen der Aggregation zwischen Vorgängen. Dieser Spezialisierungs-Zusammenhang ist in Abbildung 5–6 vereinfachend nicht dargestellt.

(18)

Spezialisierungen des Konzeptes *Produkt* wie *Dokumente* und *Kapitel* sind über Aggregation assoziiert. Enthält ein Produkt ein anderes, existiert für die jeweiligen *Produkterstellungen* eine entsprechende Aggregation:

$$\forall p_1, p_2 \in \text{Produkt}: p_1.\text{istExtern} = p_2.\text{istExtern} = \text{False} \wedge p_2 \in p_1.\text{enthält}^{\text{teil}} \Rightarrow p_2.\text{erstellt} \in p_1.\text{erstellt}.\text{enthält}^{\text{teil}}$$

Die Struktur der *Produkterstellungen* spiegelt also genau die Struktur der *Produkte* wider. Für *Produktbearbeitungen* gilt keine Einschränkung bezüglich der Anzahl bearbeiteter *Produkte*. *Produktbearbeitungen* können im Vorgehensmodell und auf Projektebene mehrere *Produkte* bearbeiten und *Produkte* können von mehreren *Produktbearbeitungen* bearbeitet

werden. Produktbearbeitungen dienen der Planung der Zusammenarbeit mehrerer Bearbeiter bei der Erstellung eines Produktes.

(19)

Ein Vorgang Produktbearbeitung bearbeitet die Produkte, in deren Produkterstellungsvorgängen er über Aggregation enthalten ist.

$$\forall b \in \text{Produktbearbeitung}: \forall e \in b.\text{Produkterstellung}: e.\text{erstellt} \in b.\text{bearbeitet}$$

Eine Produktbearbeitung kann dabei von mehreren Produkterstellungen aggregiert werden, wie wir im folgenden Abschnitt am Beispiel zeigen.

Hilfsprodukte tragen zur Erstellung der eigentlichen Projektergebnisse bei. Die Assoziation *Hilfsprodukte* zwischen Produkten ordnet bestimmte Produkte als Hilfsprodukte anderen Produkten zu. Die Assoziation Hilfsprodukt ist nicht hierarchisch, da ein Hilfsprodukt unter mehreren anderen Produkten eingeordnet werden kann, muss aber natürlich frei von Zyklen sein. Die Kennzeichnung eines Produktes als Hilfsprodukt eines anderen Produktes spiegelt eine Entscheidung des Prozessingenieurs oder des Projektleiters wider, die der Übersichtlichkeit des Vorgehensmodells beziehungsweise des Projektplans dient.

(20)

Vereinfachend gilt, dass nur ein „ganzes“ Produkt Hilfsprodukt sein kann, nicht aber der aggregierte Teil eines Produktes:

$$\forall p_1, p_2 \in \text{Produkt}: p_1.\text{Hilfsprodukt} = p_2 \Rightarrow p_2.\text{enthält}^{\text{ganzes}} = \emptyset$$

(21)

Ist ein Produkt als Hilfsprodukt eines anderen Produktes gekennzeichnet, sind die Produkterstellungsvorgänge entsprechend aggregiert:

$$\forall p_1, p_2 \in \text{Produkt}: p_1.\text{istExtern} = p_2.\text{istExtern} = \text{False} \wedge \\ p_1.\text{Hilfsprodukt} = p_2 \Rightarrow p_2.\text{erstellt} \in p_1.\text{erstellt.enthält}^{\text{teil}}$$

In der Struktur der Produkterstellungen spiegelt sich also auch die Kennzeichnung von Produkten als Hilfsprodukten wider. Im folgenden Abschnitt verdeutlichen wir die Konzepte des Vorgehens-Meta-Modells anhand von Beispielen.

Beispiel: Vorgehensmodell

Abbildung 5–7 zeigt die Konzepte des Meta-Modells anhand des durchgängigen Beispiels der vorangehenden Kapitel auf Ebene des Vorgehensmodells. Jedem Produkt, mit Ausnahme dem als extern markierten *VGM-Editor*, ist ein Vorgang zur Produkterstellung zugeordnet (vgl. Constraint (17)). Gemäß der Modellierungstechnik gilt dieses Constraint auch auf Ebene des Vorgehensmodells als Einschränkung der Projektebene (vgl. Kapitel 4.2.2, Constraint (14)). Auch die im Meta-Modell definierten Multiplizitäten zwischen Produkterstellung und Produkt gelten auf Ebene des Vorgehensmodells entsprechend (vgl. Kapitel 4.2.1, Constraint (8)). Multiplizitäten spezialisierter Assoziationen werden allerdings gemäß den Vereinbarungen der konkreten Syntax der Modellierungstechnik in Abbildung 5–7 nicht dargestellt.

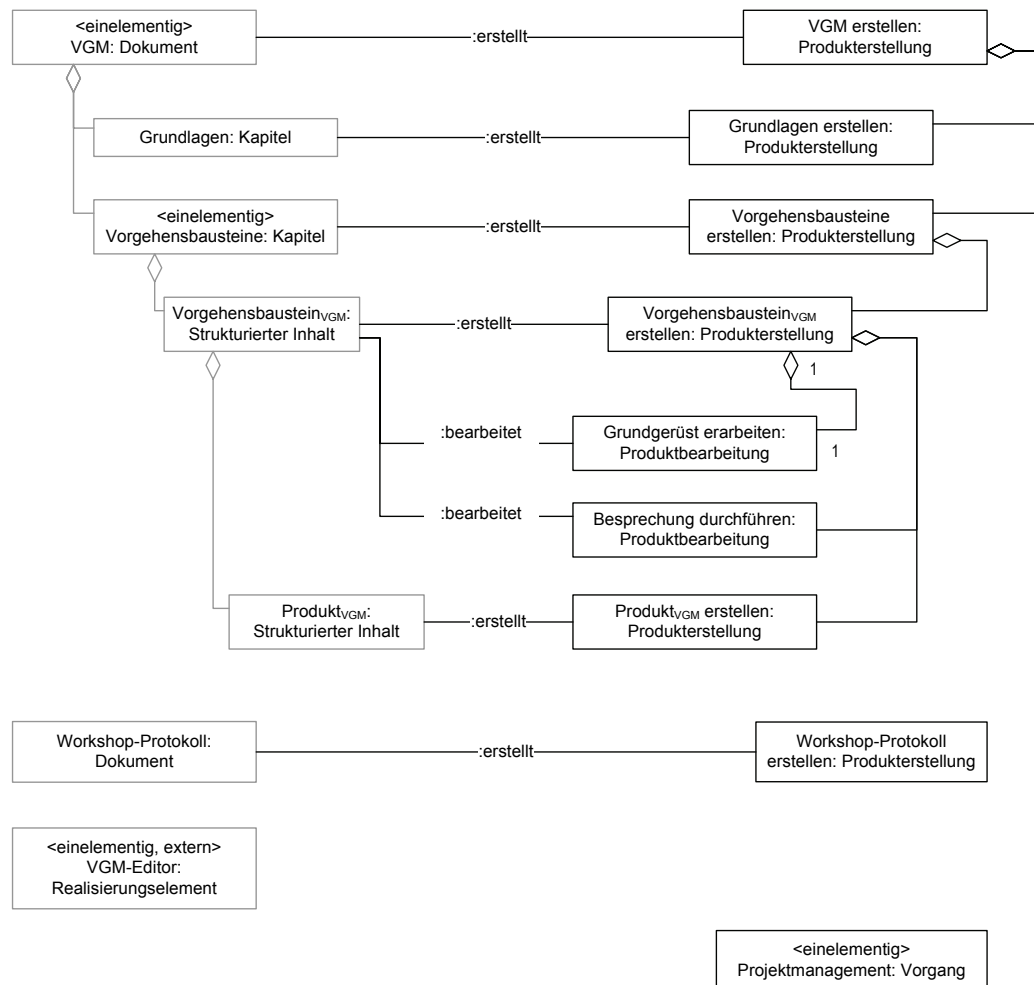


Abbildung 5–7: Planung der Produkterstellung – Ebene Vorgehensmodell (Beispiel)

Die Aggregation zwischen Produkterstellungen entspricht im Vorgehensmodell der Aggregation zwischen Produkten (vgl. Constraint (18)). Dieses Constraint gilt ebenfalls wie alle Constraints des Meta-Modells auch auf Ebene des Vorgehensmodells als Einschränkung der Projektebene.

Dem *Vorgehensbaustein* als Element des VGMs im Beispiel sind zwei Vorgänge zur Produktbearbeitung zugeordnet. Genau ein Vorgang zur „*Erarbeitung des Grundgerüsts*“ eines Vorgehensbausteins sowie eine beliebige Anzahl von Vorgängen zur „*Durchführung von Besprechungen*“. Für beide Vorgänge fordert das Vorgehensmodell im Beispiel kein eigenes dokumentiertes Projektergebnis. Vielmehr fließen die Ergebnisse beider Vorgänge unmittelbar in das Produkt Vorgehensbaustein ein. Hierin unterscheiden sich Produktbearbeitungen von Produkterstellungen.

Die Produktbearbeitung „*Grundgerüst erarbeiten*“ ist genau einer Produkterstellung „*Vorgehensbaustein erstellen*“ zugeordnet und tritt für jede Produkterstellung „*Vorgehensbaustein erstellen*“ genau einmal auf. Diese Multiplizitäten leiten sich nicht aus dem Vorgehens-Meta-Modell ab. Vielmehr steht dem Prozessingenieur auf der Ebene des Vorgehensmodells das Konzept der Multiplizitäten als Sprachmittel zur Einschränkung der Projektebene zur Verfügung.

Projektmanagement ist als spezialisierter, einelementiger Vorgang im Vorgehensmodell enthalten. Da das Projektmanagement eine querschnittliche Aufgabe darstellt, die nicht unmittelbar mit der Erstellung eines Produktes verbunden ist, aber doch Teil vieler Projektpläne ist, bietet sich hier die Modellierung als Vorgang an.

Beispiel: Projektebene

Abbildung 5–8 zeigt eine mögliche Spezialisierung des Vorgehensmodells in Abbildung 5–7 auf Projektebene. Die Namen der Produkterstellungen, die sich schematisch aus den Namen der jeweiligen Produkte ergeben (vgl. Constraint (17)), lassen wir in Abbildung 5–8 abkürzend aus. Die spezialisierten Vorgänge zur Produktbearbeitung des VGMs versehen wir dagegen in Abbildung 5–8 jeweils mit Namen. Die Namen der Produktbearbeitungen sind dabei nicht vorgegeben. Die Erarbeitung des Grundgerüsts eines Vorgehensbausteins „*PM strukturieren*“ wird entsprechend den im Vorgehensmodell definierten Multiplizitäten spezifisch für den Vorgehensbaustein Projektmanagement eingeplant.

Die Produktbearbeitung „*Überarbeitung Herr Maier*“ ist als eine direkte Spezialisierung des Konzeptes Produktbearbeitung des Vorgehens-Meta-Modells eingefügt. Auf Ebene des Vorgehensmodells lässt sich keine sinnvolle Abstraktion dieser Produktbearbeitung beschreiben. Somit ist diese Produktbearbeitung auch nicht Bestandteil des Vorgehensmodells. Produktbearbeitungen auf Projektebene sind erforderlich, um die Zusammenarbeit von mehreren Bearbeitern bei der Erstellung desselben Produktes zeitlich planen zu können.

Die „*Besprechung PM und SE*“ dient als Beispiel einer Produktbearbeitung, die zur Erstellung zweier Produktexemplare beiträgt, also sowohl den Vorgehensbaustein *Projektmanagement* als auch den Vorgehensbaustein *Systemerstellung* des VGM bearbeitet. Produktbearbeitungen sind nicht eindeutig Produkterstellungen zugeordnet. Die Aggregations-Beziehung zwischen Produkterstellung und -bearbeitung stellt also keine Hierarchie dar. Die Einschränkung auf eine Hierarchie würde die Ausdrucksmächtigkeit unerwünscht reduzieren.

Weiterhin zeigt Abbildung 5–8 zwei Beispiele für die Anwendung des Konzeptes Hilfsprodukt. Ein Workshop „*Strukturierung der Inhalte*“ wird unterstützend zur Erstellung des VGM als eigentlichem Projektergebnis eingeplant. Die Kennzeichnung des *Workshop-Protokolls* als Hilfsprodukt erfolgt erst auf der Projektebene. Da Workshop-Protokolle sehr unterschiedlich eingesetzt werden können, erscheint eine Festlegung der Assoziation Hilfsprodukt auf Ebene des Vorgehensmodells nicht sinnvoll. Die Kennzeichnung von Hilfsprodukten kann allerdings im Allgemeinen auch auf Ebene des Vorgehensmodells erfolgen.

Die Erstellung einer Einladungs-Email als Vorbereitung beziehungsweise als Hilfsprodukt für den Workshop wird ebenfalls geplant. Das Dokument Einladungs-Email leitet sich dabei nicht aus dem Vorgehensmodell ab, sondern unmittelbar aus der Klasse Dokument des Vorgehens-Meta-Modells. Eine Abstraktion einer Einladungs-Email auf Ebene des Vorgehensmodells erachten wir nur bedingt für sinnvoll.

Die unmittelbare Spezialisierung der Konzepte des Meta-Modells bringt ein hohes Maß an Flexibilität bei der Erstellung einer Planung auf Projektebene mit sich. Planung als Spezialisierung der Konzepte des Vorgehensmodells kann in einem ersten Schritt erfolgen, während Ergänzungen durch unmittelbare Spezialisierung der Konzepte des Vorgehens-Meta-Modells in einem zweiten Schritt angebracht erscheinen.

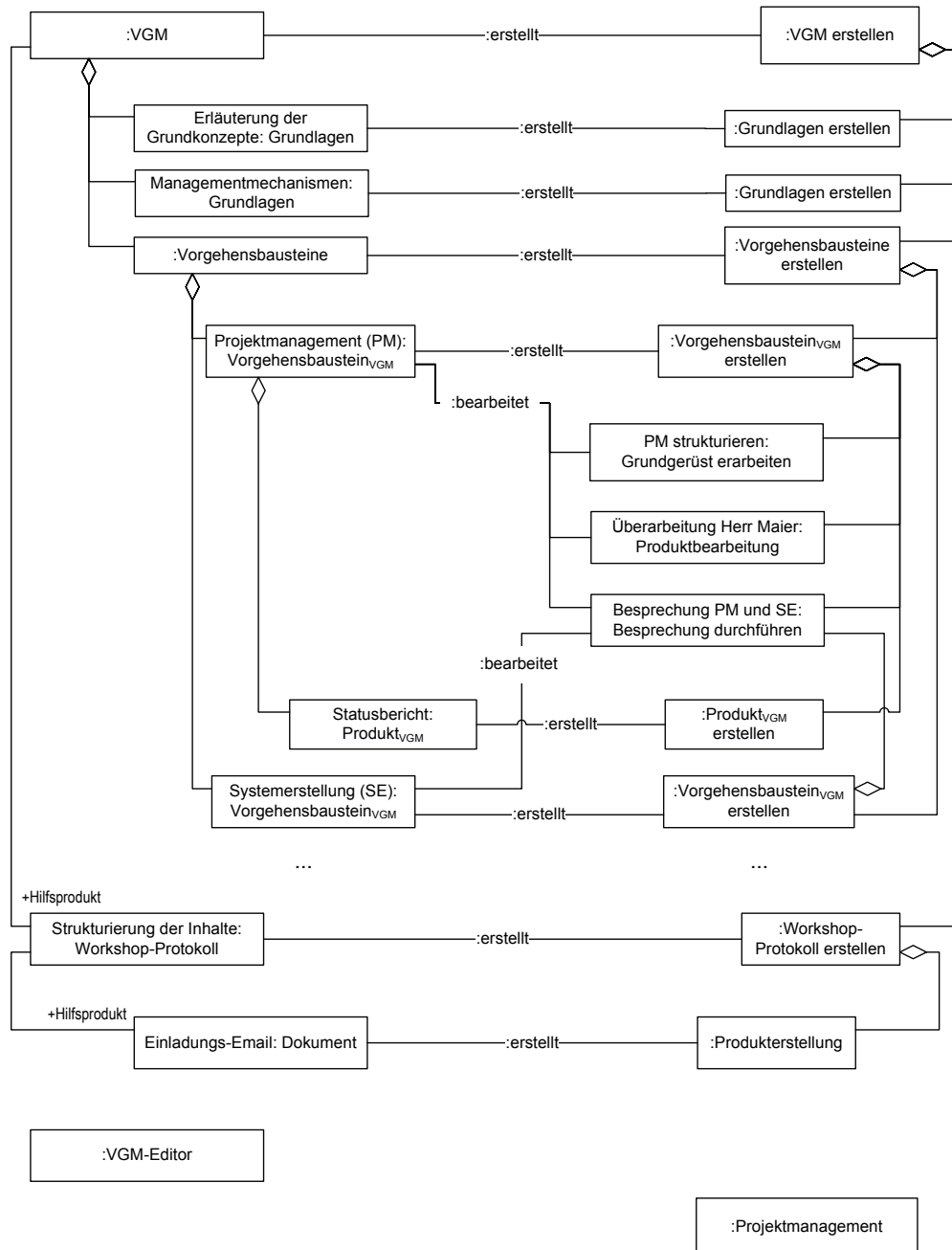


Abbildung 5–8: Planung der Produkterstellung – Projektebene (Beispiel)

Entsprechend den Vorgaben des Vorgehens-Meta-Modells existiert zu jedem Produktexemplar genau ein Exemplar eines Vorgangs zur Produkterstellung. Die Erstellung eines Produktes im Rahmen eines Vorgangs entspricht der initialen Planung für ein Projekt. Im Projektverlauf kann beispielsweise aufgrund veränderter Anforderungen oder Fehler im Entwicklungsgegenstand eine Überarbeitung eines Produktexemplars, dessen Produkterstellung bereits abgeschlossen ist, erforderlich werden.

Eine Produkterstellung pro Produktexemplar

Ein Vorgehensmodell kann beispielsweise durch die Formulierung von Produktabhängigkeiten die Notwendigkeit einer zeitlichen Überlappung bestimmter Produkterstellungen festle-

gen (vgl. dazu Kapitel 5.3.2). Für den Fall einer Inkonsistenz eines Produktexemplars als Folge der Inhalte eines anderen Produktexemplars kann im Vorgehensmodell also ein zu frühes Beenden einer Produkterstellung im Projekt verhindert werden. Eine Produkterstellung kann auch bei geringem Aufwand über einen längeren Zeitraum geplant werden, da der Aufwand nicht mit der Dauer der Produkterstellung, sondern mit der Dauer der enthaltenen Produktbearbeitungen korreliert.

Nichtsdestotrotz kann ein unvorhergesehenes „Wiederbeleben“ eines Vorgangs zur Bearbeitung eines Produktexemplars im Projekt erforderlich werden. Der Fall eines geringfügigen Aufwands für die Beseitigung einer Inkonsistenz eines Produktexemplars ist dabei nicht planungsrelevant. Im Falle unvorhergesehener, planungsrelevanter Überarbeitungen eines Produktexemplars enthält das Vorgehens-Meta-Modell keine Vorgaben. Auf Projektebene kann in diesem Fall ohne Einschränkung beispielsweise in Form unmittelbarer Spezialisierungen des Konzeptes Vorgang des Meta-Modells geplant werden.

Zusammenfassung

Den Produkten als zentralen Projektergebnissen ist jeweils genau ein Vorgang zur Produkterstellung zugeordnet. Die Planung orientiert sich also stets an den zu erbringenden Ergebnissen. Produktbearbeitungen sind für die zeitliche Planung der Zusammenarbeit mehrerer Bearbeiter bei der Erstellung desselben Produktes erforderlich. Ein Vorgang zur Produktbearbeitung kann unterschiedliche Produkte bearbeiten und ordnet sich damit nicht nur an einer Stelle in die Hierarchie der Vorgänge ein. Das V-Modell XT dagegen beinhaltet beispielsweise kein der Produktbearbeitung entsprechendes Konzept, sondern vielmehr eine sehr eingeschränkte Produkt-Aktivitäts-Schnittstelle (vgl. Kapitel 3.4.2).

Produkte sind von unterschiedlicher „Wichtigkeit“. Hilfsprodukte tragen zur Erstellung der eigentlichen Projektergebnisse bei. Die Erstellung von Hilfsprodukten wird im Projektplan auf Projektebene unter den jeweiligen Produkterstellungen der eigentlichen Projektergebnisse eingeordnet.

Dem Prozessingenieur stehen bei der Beschreibung des Vorgehensmodells Multiplizitäten der Modellierungstechnik zur Verfügung. Diese dienen der Einschränkung der projektspezifischen Produktstruktur und beeinflussen damit auch die Projektplanung in Form von Produkterstellungen und Produktbearbeitungen. Auch bei der Erstellung einer Projektplanung stehen die Konzepte des Vorgehens-Meta-Modells zur Verfügung. Beispielsweise kann die Projektebene Produktbearbeitungen, die im Vorgehensmodell nicht vorgesehen sind, als unmittelbare Spezialisierungen des Vorgehens-Meta-Modells beinhalten. Produkte können auch projektspezifisch als Hilfsprodukte gekennzeichnet werden. Diese Flexibilität im Rahmen der Planung ist wünschenswert, da Vorgehensmodelle als Abstraktionen der Projektebene in der Praxis nicht hinsichtlich aller einzuplanenden Vorgänge vollständig sein können.

5.2.4 Planungsmodell – Strukturierung des Projektablaufs

In diesem Kapitel erweitern wir das Planungsmodell um spezialisierte Vorgänge als Strukturierungsmöglichkeit für den Projektablauf auf grobgranularer Ebene. Zielsetzung ist zudem die Klärung des Zusammenhangs zwischen dem Projektablauf auf grobgranularer Ebene und dem Produktmodell.

Vorgehens-Meta-Modell

Abbildung 5–9 zeigt eine Erweiterung des Planungsmodells auf Ebene des Vorgehens-Meta-Modells. Neben *Produkterstellung* und *-bearbeitung* sind *Projektabschnitte* als Spezialisierung des Konzeptes *Vorgang* Bestandteil des Vorgehens-Meta-Modells.

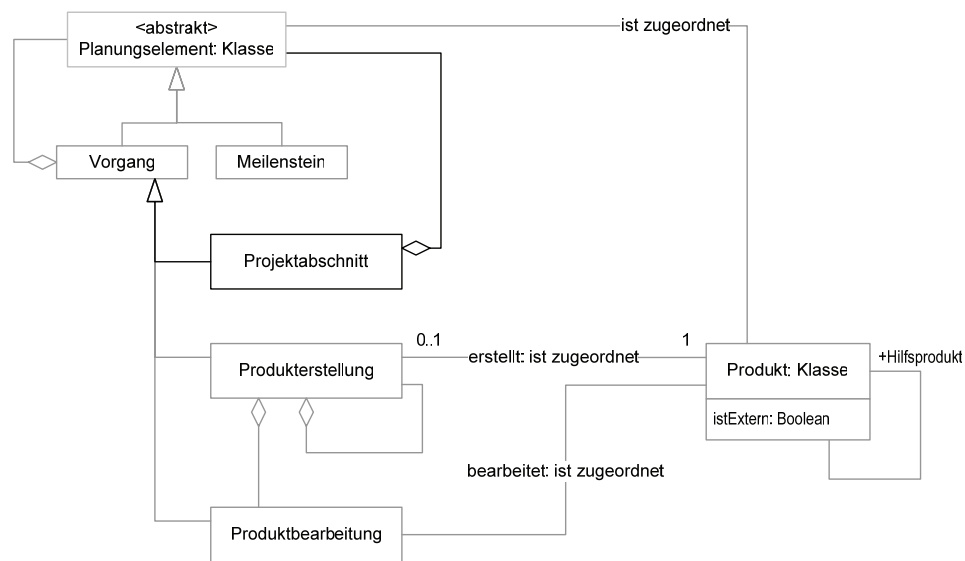


Abbildung 5–9: Strukturierung des Projektablaufs – Ebene Vorgehens-Meta-Modell

Projektabschnitte als Strukturierung des grobgranularen Ablaufs eines Projektes aggregieren Planungselemente, können also sowohl *Vorgänge*, *Meilensteine*, *Produkterstellungen* und *-bearbeitungen* als auch andere Projektabschnitte enthalten. Projektabschnitte dienen der Strukturierung des Projektablaufs. Planungselemente sind nicht eindeutig in Projektabschnitten eingeordnet. Dies motivieren wir in den folgenden Abschnitten anhand von Beispielen.

Projektabschnitten sind wie allen Planungselementen Produkte *zugeordnet*. Eine Spezialisierung der Assoziation „ist zugeordnet“ wie im Falle der Assoziationen *erstellt* und *bearbeitet* modellieren wir hier nicht, da mit dieser Zuordnung abhängig von dem im Vorgehensmodell modellierten Projektablauf unterschiedliche Semantiken verbunden sein können. Auch Constraints in Bezug auf Projektabschnitte geben wir im Vorgehens-Meta-Modell nicht an, da für unterschiedliche im Vorgehensmodell modellierte Projektabläufe unterschiedliche Constraints benötigt werden.

Beispiel: Vorgehensmodell

Abbildung 5–10 zeigt eine Strukturierung des Projektablaufs anhand eines Beispiels auf der Ebene des Vorgehensmodells. Wir beziehen uns dabei auf die Erstellung des Vorgehensmodells VGM. Das Beispiel lehnt sich an dieser Stelle eng an die Planung im Projekt WEIT an (vgl. [WEIT04]). Die Klassen α -Version und β -Version sind in Abbildung 5–10 entsprechend den Vereinbarungen der konkreten Syntax ohne eine durch Doppelpunkt abgetrennte Generalisierung dargestellt, da ihre Generalisierungen bereits in Form von Spezialisierungspfeilen angegeben sind.

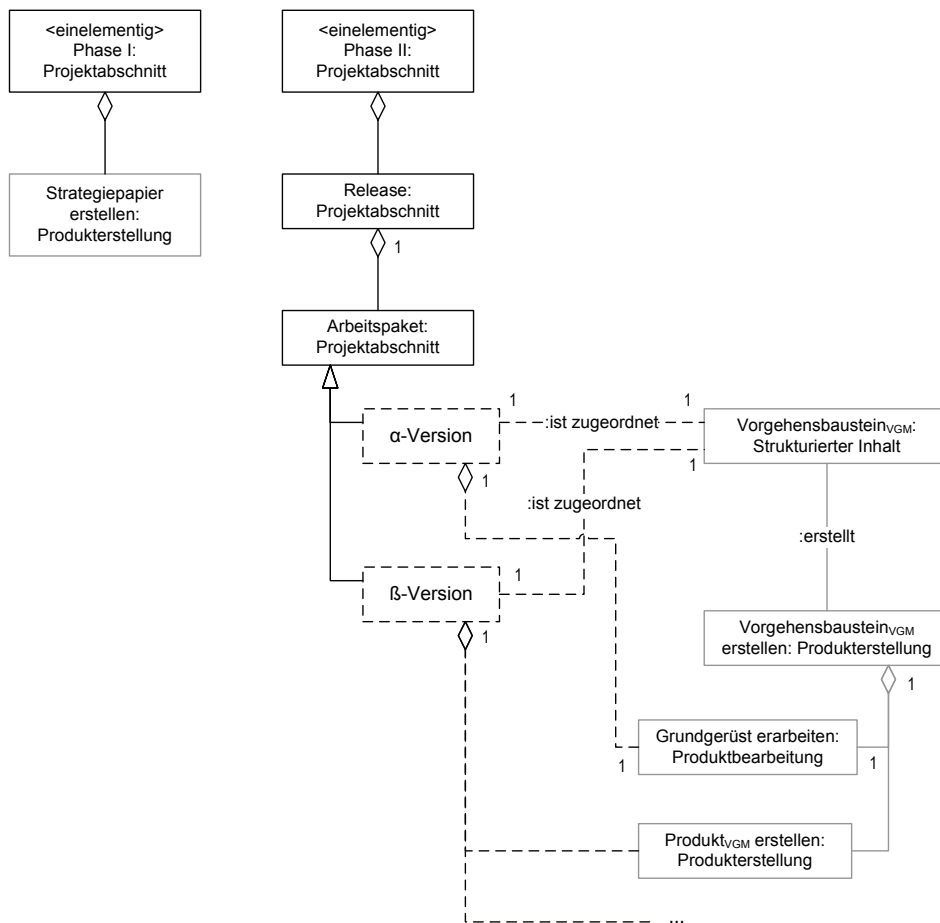


Abbildung 5–10: Strukturierung des Projektablaufs – Ebene Vorgehensmodell (Beispiel)

Den Projektablauf teilen wir im Beispiel in zwei Projektabschnitte ein. „Phase I“ aggregiert im Beispiel den Vorgang zur Produkterstellung des *Strategiepapiers*. Der Projektabschnitt „Phase II“ ist durch weitere Projektabschnitte unterstrukturiert. So genannte *Releases* markieren Abschnitte, mit deren Abschluss ein Ergebnis des Projektes fertig gestellt werden soll.

Releases entsprechen der Idee des „Time Boxing“, die unter anderem dem Extreme Programming entstammt (vgl. [Beck99]). Releases umfassen dabei Zeitabschnitte gleicher Dauer. Der Umfang des in einem Release zu erbringenden Ergebnisses orientiert sich an der für das Release zur Verfügung stehenden Zeit.

Releases enthalten *Arbeitspakete*, die ebenfalls als Projektabschnitte modelliert sind. Arbeitspakete treten in den Ausprägungen *α-Version* und *β-Version* auf. Für die Arbeitspakete *α-Version* und *β-Version* ist dabei im Beispiel definiert, welche Planungselemente diese jeweils beinhalten. *α-Versionen* beinhalten die Produktbearbeitungen „*Grundgerüst erarbeiten*“. *β-Versionen* beinhalten unter anderem die Produkterstellungen zu den Produkten des VGM.

Für Projektabschnitte wie für alle Planungselemente gilt, dass sich diese zeitlich zwischen Anfangs- und End-Termin der Vorgänge einordnen müssen, in denen sie enthalten sind (vgl. Kapitel 5.2.1, Constraint (15)). Damit wird das Vorgehensmodell im Beispiel dem Prinzip des „Time Boxing“ gerecht.

Arbeitspakete beziehen sich auf Vorgehensbausteine des VGM. Eine Spezialisierung der Beziehung „ist zugeordnet“ des Vorgehens-Meta-Modells ordnet ein Arbeitspaket α -Version beziehungsweise ein Arbeitspaket β -Version jeweils eindeutig einem Vorgehensbaustein des VGM zu.

(22)

Einem Vorgehensbaustein_{VGM} ist ein Projektabschnitt α -Version zugeordnet, der eine Produktbearbeitung „Grundgerüst erarbeiten“ enthält, die in der jeweiligen Produkterstellung „Vorgehensbaustein_{VGM} erstellen“ enthalten ist. Der Name der α -Version ist dabei identisch mit dem Namen des Vorgehensbaustein_{VGM}:

$$\forall v \in \text{Vorgehensbaustein}_{\text{VGM}}:$$

$$v.\alpha\text{-Version.Grundgerüst erarbeiten.Vorgehensbaustein erstellen.} \quad \text{Vorgehensbaustein}_{\text{VGM}} = v \wedge$$

$$v.\alpha\text{-Version.name} = v.\text{name}$$

Für die Projektabschnitte β -Version und Vorgehensbausteine gelten analoge Constraints, die wir hier abkürzend nicht angeben. Die Projektabschnitte β -Version enthalten dabei jeweils alle Produkterstellungen, die von der jeweiligen Produkterstellung „Vorgehensbaustein_{VGM} erstellen“ aggregiert werden, also beispielsweise die Produkterstellungen zu Produkten des VGM.

Im Gegensatz zu den Beispielen der vorangehenden Kapitel obliegt dem Prozessingenieur in diesem Beispiel die Bewältigung eines höheren Komplexitätsgrades. Das Vorgehens-Meta-Modell bietet an dieser Stelle keine vorgefertigten Constraints, da diese je nach modelliertem Projektablauf unterschiedlich sind. Neben der Formulierung von Einschränkungen in Form von Multiplizitäten müssen Constraints spezifiziert werden. Die Erstellung eines Vorgehensmodells ist also keine triviale Aufgabe.

Eine Verallgemeinerung des Beispiels oder einzelner Aspekte auf der Ebene des Vorgehens-Meta-Modells bietet sich mangels Allgemeingültigkeit des Beispiels nicht an. Auch auf Ebene des Vorgehensmodells ist das Beispiel für unterschiedliche Projekttypen nicht uneingeschränkt anwendbar, sondern muss Gegenstand des Tailorings sein (vgl. Kapitel 6.2).

Beispiel: Projektebene

Abbildung 5–11 zeigt ein Beispiel einer möglichen Spezialisierung des Vorgehensmodells auf Projektebene. Im Beispiel werden für das Erstellungsprojekt des VGM drei *Releases* geplant, die jeweils Arbeitspakete *α -Version* beziehungsweise *β -Version* der Vorgehensbausteine *Projektmanagement* beziehungsweise *Systemerstellung* des VGM beinhalten. Die Produktbearbeitungen „Grundgerüst erarbeiten“ beziehungsweise die Produkterstellungen zu den Produkten des VGM sind entsprechend den Vorgaben des Vorgehensmodells den Arbeitspaketen zugeordnet.

Im Beispiel wird deutlich, dass Aggregations-Beziehungen zwischen Vorgängen keine Hierarchie bilden. Produkterstellungen beziehungsweise -bearbeitungen sind im Beispiel sowohl unter Produkterstellungen als auch unter Projektabschnitten eingeordnet. Dies entspricht den Erfordernissen der Praxis, wie sich beispielsweise am Projektplan des Projektes WEIT ablesen lässt (vgl. [WEIT04]). Da Standardlösungen für Planungswerkzeuge wie [MSProj03] allein hierarchische Strukturierungen der Vorgänge erlauben, behilft sich beispielsweise dieser Projektplan einer redundanten Modellierung mit Hilfe von Vorgangsfolgen. Der Plan enthält eine Strukturierung von Vorgängen, die sich am zu erstellenden Produkt orientiert, sowie eine weitere Strukturierung von Vorgängen, die sich an der zeitlichen Planung in Form von Releases orientiert. Da es sich bei den Vorgängen beider Strukturierungen jeweils eigentlich um denselben Vorgang handelt, sind diese redundanten Vorgänge mit Hilfe von

Vorgangsfolgen so gekoppelt, dass die planbaren Termine für beide Vorgänge jeweils dieselben sind. Nicht-hierarchische Vorgangsstrukturen, wie in diesem Kapitel am Beispiel gezeigt, sind geeignet, die Zahl der Vorgänge und damit die Komplexität des WEIT-Projektplans in etwa zu halbieren.

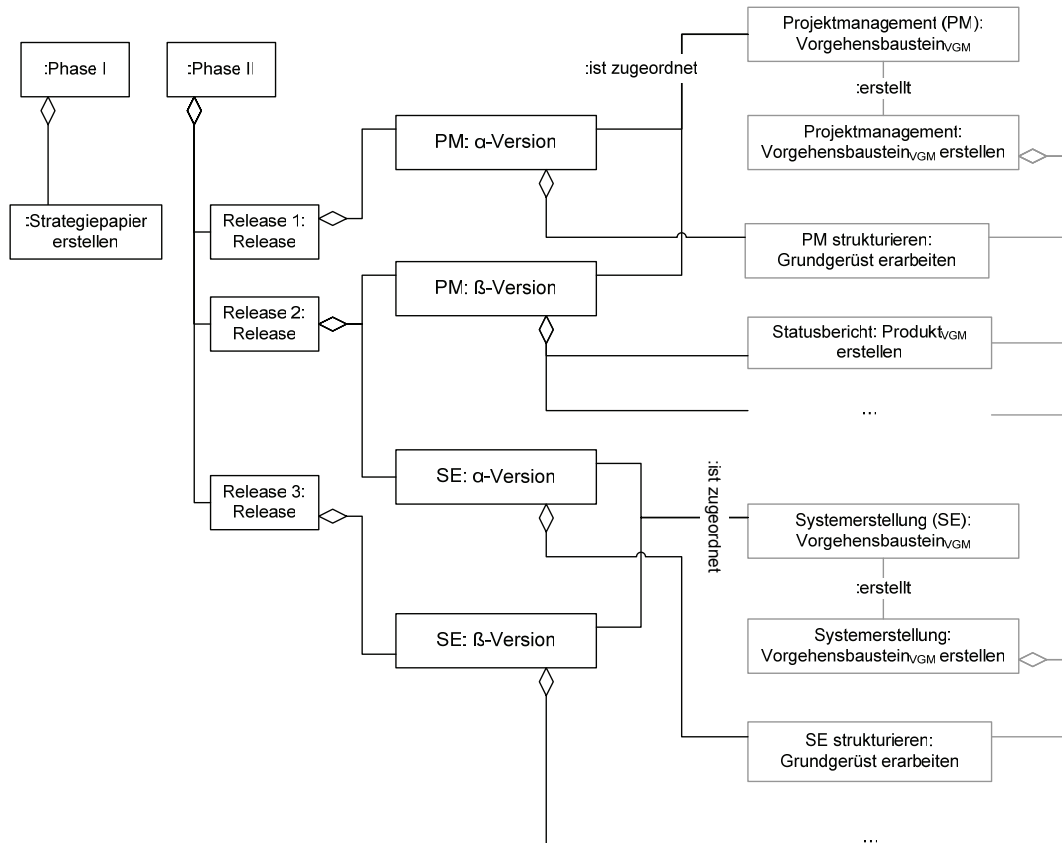


Abbildung 5–11: Strukturierung des Projektablaufs – Projektebene (Beispiel)

Zusammenfassung

Vorgänge zur Produkterstellung und Produktbearbeitung ordnen wir in Projektabschnitte ein. Vorgänge sind dabei nicht in genau einem anderen Vorgang enthalten. Beispielsweise Produktbearbeitungen können sowohl von Projektabschnitten als auch von Produkterstellungen aggregiert werden, wie wir anhand eines Beispiels gezeigt haben.

Projektabschnitte können vielfach unterschiedlich strukturiert werden. In diesem Kapitel haben wir auf Ebene des Vorgehensmodells ein Beispiel einer Strukturierung in Phasen, Releases und Arbeitspaketen gezeigt. Arbeitspakete sind definiert als Gruppierungen zu erstellender Produkte. Wir verstehen Arbeitspakete als Projektabschnitte. Auf Ebene des Vorgehensmodells müssen im Beispiel neben Multiplizitäten auch Constraints als Einschränkungen der Projektebene formuliert werden. Dem Prozessingenieur steht das gesamte Spektrum der Modellierungstechnik zur Verfügung. Die Erstellung von Vorgehensmodellen ist also keine triviale Aufgabe.

5.3 Dynamisches Modell

Im dynamischen Modell gehen wir auf die Aspekte des Planungsmodells und des Produktmodells ein, die der Durchführung von Vorgängen beziehungsweise der Erstellung der Projektergebnisse eine zeitliche Reihenfolge geben.

In Kapitel 5.3.1 betrachten wir im Planungsmodell Reihenfolge-Beziehungen zwischen Planungselementen sowie iterative Abläufe. In Kapitel 5.3.2 zeigen wir unterschiedliche Arten von Abhängigkeitsbeziehungen zwischen Produkten, die sich auf die Vorgänge zur Produkterstellung und -bearbeitung beziehungsweise deren Reihenfolge auswirken.

Abschnitte in Kapitel 5.3		Seite
5.3.1	Planungsmodell – Ablaufbeziehungen und Iterationen	127
5.3.2	Produktmodell – Abhängigkeitsbeziehungen	132

5.3.1 Planungsmodell – Ablaufbeziehungen und Iterationen

In diesem Kapitel ergänzen wir das Planungsmodell der vorangehenden Kapitel um Konzepte für Abläufe und Iterationen. Zielsetzung dieses Kapitel ist die Definition von Ablaufbeziehungen für Projektpläne zur Festlegung der logischen beziehungsweise zeitlichen Reihenfolge der Planungselemente. Weiterhin sollen die Parameter beziehungsweise Planungsentscheidungen auf Ebene des Vorgehensmodells definiert werden, deren projektspezifische Belegung zu iterativen Abläufen führt.

Vorgehens-Meta-Modell

Abbildung 5–12 zeigt Ablaufbeziehungen und Iterationen auf Ebene des Vorgehens-Meta-Modells. Die Modellierung dabei entspricht der Ontologie eines Projektplans (vgl. Abbildung 2–25 in Kapitel 2.4). Lediglich die Unterscheidung in Sammelvorgänge und Vorgänge modellieren wir hier nicht. Zudem ist das Konzept der Iteration nicht Bestandteil eines Projektplans entsprechend dem State-Of-The-Art.

Planungselemente können über *Abläufe* assoziiert sein. Abläufe treten in den dargestellten Spezialisierungen auf. Dabei handelt es sich um die aus der Netzplantechnik bekannten Vorgangfolgen Ende-Anfang, Ende-Ende, Anfang-Anfang sowie Anfang-Ende (vgl. Kapitel 2.3.2). Die Auswirkungen der Vorgangfolgen auf die zeitliche Planung lassen sich wie folgt formal fassen:

(23)

EA-Vorgangfolgen schränken die Termine der Planungselemente derart ein, dass der Endtermin des Vorgangs, der die Quelle der Vorgangfolge bildet, vor dem Anfangstermin des Vorgangs, der das Ziel der Vorgangfolge bildet, liegen muss:

$$\forall vf \in \text{EA-Vorgangfolge: } vf.von.Endtermin \leq vf.nach.Anfangstermin$$

Für die übrigen Vorgangfolgen Ende-Ende, Anfang-Anfang und Anfang-Ende ließen sich Constraints analog formulieren.

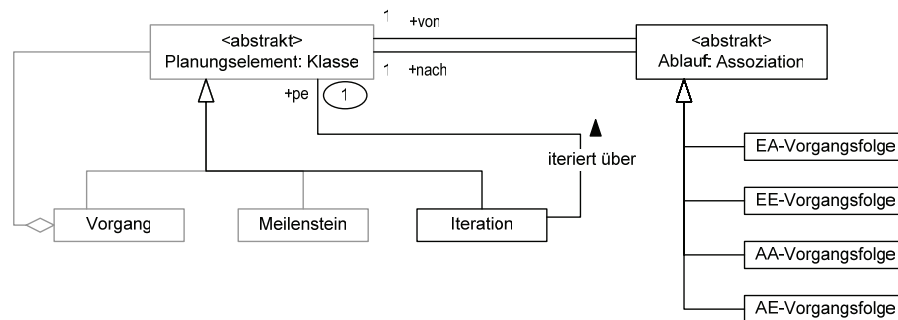


Abbildung 5–12: Ablaufbeziehungen und Iterationen – Ebene Vorgehens-Meta-Modell

Zudem ist die Modellierung zyklischer Vorgangsfolgen im Vorgehensmodell nicht sinnvoll, da sich in diesem Fall keine Termine zur Erfüllung dieser Constraints finden ließen. Als weitere Einschränkung der zeitlichen Planung gilt das folgende Constraint des Kapitels 5.2.1. Alle Planungselemente ordnen sich zeitlich zwischen Anfangs- und End-Termin der Vorgänge ein, in denen sie enthalten sind.

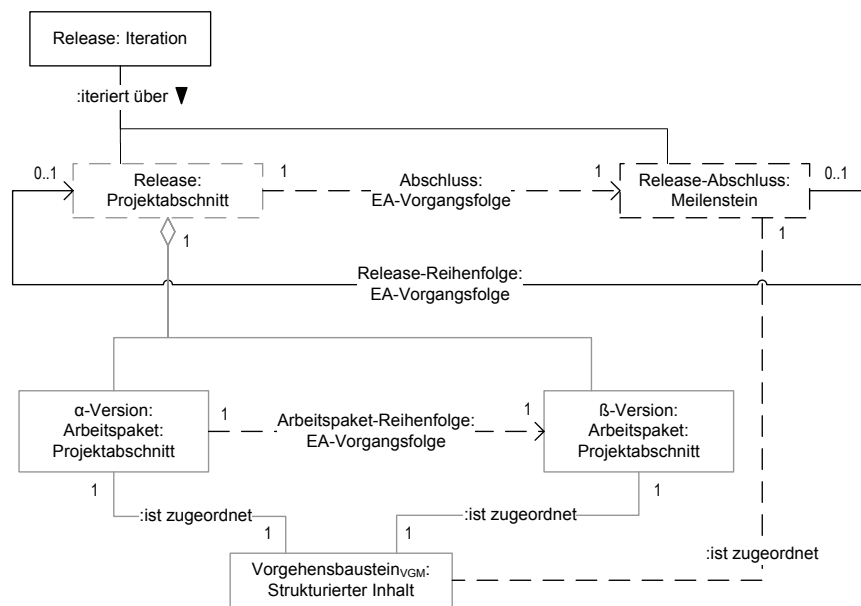
Unter *Iterationen* auf Ebene des Vorgehensmodells sind genau genommen Iterationstypen zu verstehen, wie beispielsweise auch unter Produkten Produkttypen zu verstehen sind. *Iterationen* „iterieren über“ Planungselementen. Die jeweiligen Planungselemente treten auf Projektebene für jede auf Projektebene spezialisierte Iteration jeweils einmal auf. Entsprechend den Vereinbarungen der konkreten Syntax der Modellierungstechnik wird dies in Abbildung 5–12 durch eine eingekreiste Multiplizität notiert. Eine entsprechende Multiplizität gilt nicht unmittelbar als Einschränkung der möglichen Spezialisierungen einer Assoziation. Vielmehr schränkt die eingekreiste Multiplizität die Spezialisierungen aller Spezialisierungen einer Assoziation ein.

Wir unterscheiden das Konzept der Iterationen vom Konzept der Projektabschnitte, da Iterationen der expliziten Kennzeichnung von Planungsentscheidungen dienen. Projektabschnitte verfügen über keine entsprechende einschränkende Multiplizität bezüglich der enthaltenen Elemente auf Projektebene. Im folgenden Abschnitt geben wir Beispiele für Abläufe und Iterationen an.

Beispiel: Vorgehensmodell und Projektebene – Strukturierung des Projektablaufs

Abbildung 5–13 zeigt Beispiele für Ablaufbeziehungen und Iterationen auf Ebene des Vorgehensmodells sowie auf Projektebene. Wir führen dabei das vorangehende Beispiel zur groben Strukturierung des Projektablaufs durch Projektabschnitte aus Kapitel 5.2.4 fort. Zur Verbesserung der Übersichtlichkeit verwenden wir in Abbildung 5–13 Pfeile zur Darstellung von spezialisierten EA-Vorgangsfolgen.

(a) Ebene Vorgehensmodell



(b) Projektebene

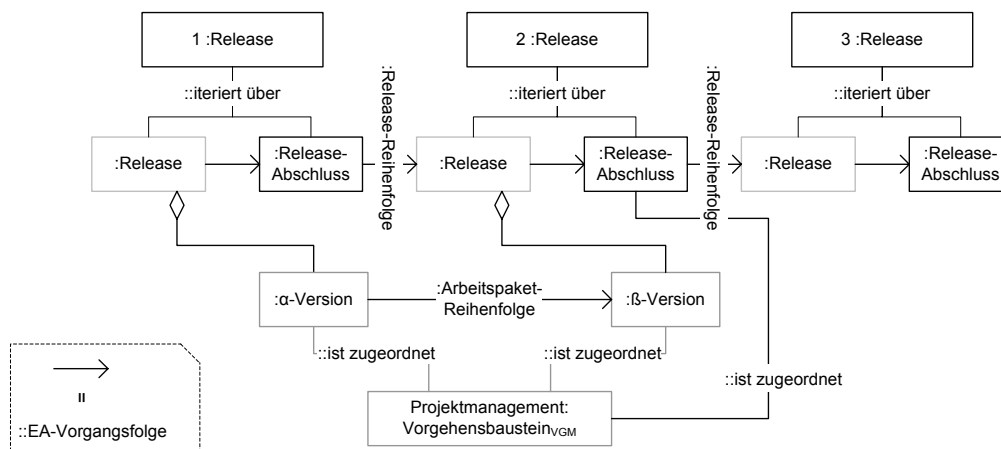


Abbildung 5–13: Ablaufbeziehungen und Iterationen – Ebene Vorgehensmodell und Projektebene (Beispiel: Strukturierung des Projektablaufs)

EA-Vorgangsfolgen als Ablaufbeziehungen setzen ein *Release* mit einem Meilenstein in Beziehung, der den *Release-Abschluss* markiert. Die EA-Vorgangsfolgen *Release-Reihenfolge* definieren zudem die Reihenfolge der Projektabschnitte *Release*.

(24)

Die Projektabschnitte Release und die Meilensteine Release-Abschluss sind über die EA-Vorgangfolgen Abschluss beziehungsweise Release-Reihenfolge verbunden und bilden auf Projektebene eine Kette.

$$\exists r_1 \in \text{Release:Projektabschnitt} : \forall r_2 \in \text{Release:Projektabschnitt} : \\ r_2 \in r_1.(\text{Abschluss}^{\text{Release-Abschluss}}.\text{Release-Reihenfolge}^{\text{Release:Projektabschnitt}})^{0..*}$$

Wir verwenden in Constraint (24) entsprechend den Vereinbarungen der konkreten Syntax der Modellierungstechnik die Notation „ (...) ^{0..*} “ um anzuzeigen, dass die jeweiligen Assoziationen einmal, mehrfach oder auch gar nicht entsprechend ihrer Richtung durchlaufen werden können.

Die Anzahl der Releases wird im Beispiel über eine Iteration festgelegt. Entsprechend den im Vorgehens-Meta-Modell definierten Multiplizitäten werden Releases und Release-Abschlüsse auf Projektebene je Iteration jeweils einmal spezialisiert.

Die Iteration *Release* dokumentiert auf Projektebene die Planungsentscheidung bezüglich der Anzahl der Projektabschnitte Release und Meilensteine Release-Abschluss im Projekt. Es handelt sich dabei um eine Entscheidung, die sich anders als in den Beispielen des folgenden Kapitels nicht aus anderen Produkten der Produktstruktur ableitet (vgl. dazu Kapitel 5.3.2), sondern um eine Entscheidung, die Bestandteil des Projektplans selbst ist.

Die Modellierung von Iterationen ist aus Sicht eines einfachen und redundanzfreien Vorgehens-Meta-Modells nicht zwingend erforderlich, da die Projektabschnitte Release auch ohne die Iteration Release bereits mehrfach eingeplant beziehungsweise spezialisiert werden können. Iterationen dienen also allein dem Zweck, Planungsentscheidungen explizit hervorzuheben.

(25)

Einem Meilenstein Release-Abschluss sind die Vorgehensbausteine als Ergebnisse zugeordnet, die im jeweiligen Release in *β-Version* erstellt werden:

$$\forall ra \in \text{Release-Abschluss} : \\ v \in ra.\text{Abschluss}^{\text{Release:Projektabschnitt}}.\beta\text{-Version.Vorgehensbaustein}_{\text{VGM}} \Leftrightarrow \\ v \in ra.\text{Vorgehensbaustein}_{\text{VGM}}$$

Zwischen den Arbeitspaketen *α-Version* und *β-Version* eines Vorgehensbausteins des VGM definieren EA-Vorgangfolgen eine Reihenfolge.

(26)

Die *α*- und *β*-Versionen, die demselben Vorgehensbaustein zugeordnet sind, sind über eine EA-Vorgangfolge Arbeitspaket-Reihenfolge verbunden:

$$\forall \alpha \in \alpha\text{-Version} : \\ \alpha.\text{Arbeitspaket-Reihenfolge}^{\beta\text{-Version}}.\text{Vorgehensbaustein} = \alpha.\text{Vorgehensbaustein}$$

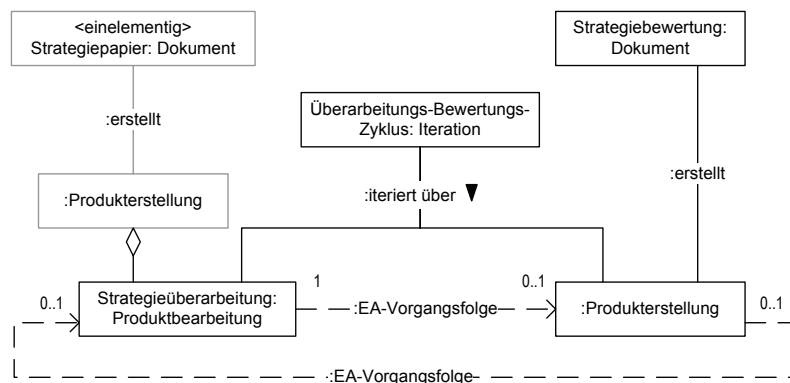
Die Arbeitspaket-Reihenfolge schränkt damit auch die Möglichkeiten der Zuordnung von Arbeitspaketen zu Releases ein, da Releases ebenfalls eine Reihenfolge haben.

Beispiel: Vorgehensmodell und Projektebene – Planung der Produkterstellung

Abbildung 5–14 zeigt ein weiteres Beispiel für den Einsatz von Ablaufbeziehungen und Iterationen. Iterationen als Planungsentscheidungen bewegen sich nicht allein auf der Ebene

der Projektstrukturierung. Iterationen treten auch auf der Ebene der Produktbearbeitung auf, also im Rahmen der Planung der Zusammenarbeit mehrerer Bearbeiter bei der Erstellung eines Produktes. Ein Constraint analog dem Constraint (24) des vorangehenden Abschnitts geben wir hier abkürzend nicht an.

(a) Ebene Vorgehensmodell



(b) Projektebene

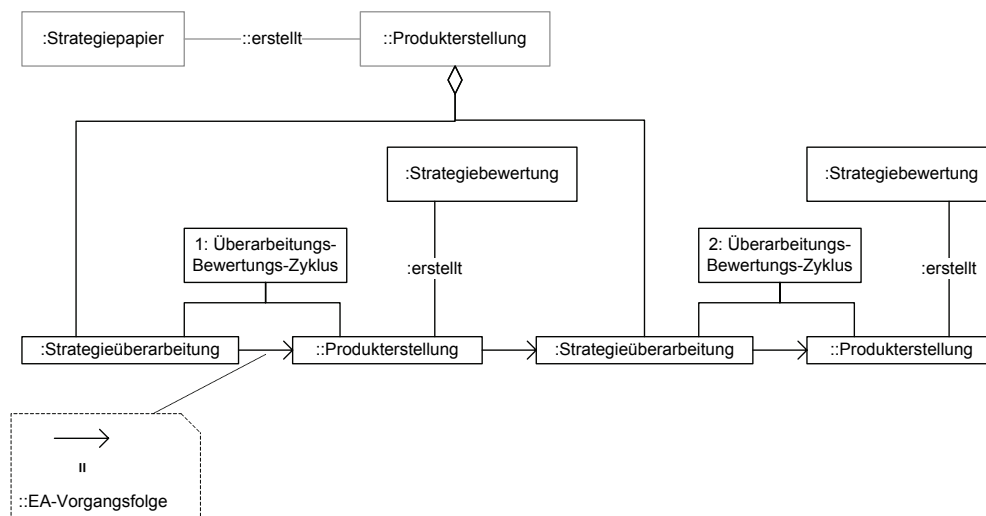


Abbildung 5–14: Ablaufbeziehungen und Iterationen – Ebene Vorgehensmodell und Projektebene (Beispiel: Planung der Produkterstellung)

Im Beispiel wird das *Strategiepapier* durch die Produktbearbeitung *Strategieüberarbeitung* erstellt. Strategieüberarbeitungen wechseln sich im Projekt mit *Produkterstellungen* des Dokumentes *Strategiebewertung* ab. Die Iteration *Überarbeitungs-Bewertungs-Zyklus* dient also der Planung der Zusammenarbeit bei der Produkterstellung. Diese Iteration spiegelt eine Planungsentscheidung wider, also eine Entscheidung, die sich nicht aus einem anderen Produkt des Projektes ableitet. Das Beispiel kann für die Planung der Erstellung eines Produktes als typisch gelten.

Im Beispiel stellen wir die Iteration Überarbeitungs-Bewertungs-Zyklus in Bezug auf das Strategiepapier dar. Die damit verbundene Vorgehensweise ist aber natürlich für die Erstellung der meisten Produktexemplare auf Projektebene anwendbar. Eine entsprechende Abstraktion geben wir in Kapitel 6.1 in Abbildung 6–4 (c) an.

Zusammenfassung

Abläufe definieren wir in Form von Vorgangfolgen der Netzplantechnik. Dieses Ablaufkonzept ist im Vergleich zum Ablaufkonzept des V-Modells XT mächtiger, da Vorgangfolgen auch Festlegungen bezüglich der möglichen zeitlichen Überlappung von Vorgängen auf Modellebene erlauben (vgl. Kapitel 3.2.3).

Iterationen dienen der expliziten Kennzeichnung von Planungsentscheidungen. Iterationen sind Parameter, die bei der Ableitung eines Projektplans auf Projektebene zu belegen sind. Anders als im Kontext der Beispiele des folgenden Kapitels leiten sich Planungselemente, die im Kontext von Iterationen auftreten, nicht aus den Inhalten anderer Produktexemplare ab. Iterationen sind als Planungsentscheidungen Bestandteil des Projektplans selbst. Iterationen treten sowohl auf der Ebene der Projektstrukturierung durch Projektabschnitte als auch auf der Ebene der Produktbearbeitung auf.

5.3.2 Produktmodell – Abhängigkeitsbeziehungen

In diesem Kapitel ergänzen wir das Produktmodell um ein Konzept für Produktabhängigkeiten. Produktabhängigkeiten sind im Kontext der Arbeit in soweit von Interesse als sie die Planung beeinflussen. Produktabhängigkeiten sehen wir als typische Muster für die Planung der Zusammenarbeit mehrerer Bearbeiter bei der Erstellung mehrerer Produkte an. Wir erheben hierbei also nicht den Anspruch auf Vollständigkeit des Vorgehens-Meta-Modells, sondern geben lediglich typische Muster an.

Zudem handelt es sich bei Produktabhängigkeiten um ein eigenständiges Konzept, das von der Korrespondenz der Namen abgesehen keinen Bezug zu den Produktabhängigkeiten des V-Modells XT aufweist. Zielsetzung des Kapitels ist die Definition unterschiedlicher Arten von Produktabhängigkeiten sowie die Definition der Auswirkungen dieser Produktabhängigkeiten auf den Projektplan.

Vorgehens-Meta-Modell

Abbildung 5–15 zeigt die Assoziation *Produktabhängigkeit* auf Ebene des Vorgehens-Meta-Modells. Wir modellieren Produktabhängigkeiten in den Formen *Erzeugung*, *Konsistenzabhängigkeit*, *Produktfluss* und *Bezug*, die wir jeweils mit unterschiedlicher Semantik versehen. Im Allgemeinen handelt es sich bei Produktabhängigkeiten um n-stellige Assoziationen, die Spezialisierungen verfügen allerdings teilweise über eingeschränktere Multiplizitäten.

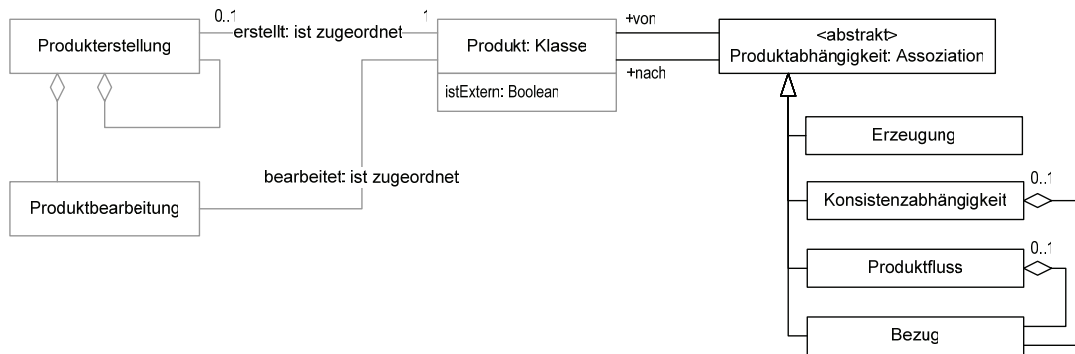


Abbildung 5–15: Produktabhängigkeiten – Ebene Vorgehens-Meta-Modell

Im Gegensatz zu Konsistenzabhängigkeiten und Produktflüssen konzipieren wir Erzeugung als Hinweis für den Projektleiter bei der Erstellung einer Planung. Dabei formalisieren wir also nicht die Auswirkungen auf die Projektebene in Form von Constraints. Erzeugung erfordert das Lesen und Interpretieren der Inhalte der Produktexemplare durch den Projektleiter. Da es in der Praxis nicht lohnenswert ist, jeden Produktzusammenhang und Produktinhalt als Vorgabe für einen Projektplan präzise zu formulieren und damit einer Werkzeugunterstützung zugänglich zu machen, ist ein Konzept wie die Erzeugung als Produktabhängigkeit unabdingbar. Im Gegensatz zu den übrigen Produktabhängigkeiten leitet sich im Kontext einer Werkzeugunterstützung aus einer Erzeugung die Existenz weiterer Produktexemplare also nicht automatisch aus der projektspezifischen Produktstruktur ab.

Erzeugung

Die Produktabhängigkeit *Erzeugung* modellieren wir als gerichtete n-stellige Assoziation, die eine Menge von Quellprodukten mit einer Menge von Zielprodukten in Beziehung setzt. Die Quellprodukte enthalten dabei eine Spezifikation, ob beziehungsweise wie viele Exemplare der Zielprodukte auf Projektebene existieren sollen. Die Erzeugung muss sich nicht auf die Zielprodukte selbst beziehen, sondern kann auch die mehrfache Existenz aggregierter Teilprodukte spezifizieren. Dies erläutern wir im folgenden Abschnitt anhand eines Beispiels. Die Definition der Erzeugung entspricht dem Verständnis der erzeugenden Produktabhängigkeiten im V-Modell XT. Erzeugende Produktabhängigkeiten sind also hinsichtlich ihrer Konsequenzen auf die Projektebene nicht weiter präzisiert. Durch die Modellierung von Produktabhängigkeiten Bezug kann eine entsprechende Präzisierung erfolgen, wie wir im Folgenden noch darstellen.

(27)

Die Erzeugung als n-stellige Assoziation muss über mindestens eine Quelle (Assoziations-Ende *von*) und mindestens ein Ziel (Assoziations-Ende *nach*) verfügen:

$$\forall e \in \text{Erzeugung: } | e.nach^{\text{Produkt}} | \geq 1 \wedge | e.von^{\text{Produkt}} | \geq 1$$

Anders als im V-Modell XT folgt aus der Modellierung einer Erzeugung keine Konsistenzsicherung der jeweiligen Produktexemplare über die Dauer der Erstellung dieser Produktexemplare im Projekt. Erzeugung bedeutet aus Sicht des Bearbeiters eines Produktexemplars den Hinweis, die Inhalte anderer Produktexemplare zu berücksichtigen. Das zuletzt erstellte Produktexemplar darf dabei allerdings durchaus von den Inhalten des zuerst erstellten Produktexemplars abweichen. Wir verstehen Erzeugung im Vorgehensmodell als Hilfestellung für die Erstellung eines Projektplans. Für den Projektleiter stellt Erzeugung also einen Hinweis für die Planung dar, welche Produktexemplare die Existenz anderer Produktexemplare spezifizieren können, also gegebenenfalls einzuplanen sind.

Aus einer Erzeugung lassen sich keine Reihenfolge-Beziehungen beziehungsweise Abläufe ableiten. Die Spezifikation der Existenz eines Produktexemplars kann im Projektablauf auch nach der Erstellung dieses Produktexemplars erstellt werden, wie beispielsweise im Kontext von Bottom-Up-Vorgehensweisen.

Konsistenzabhängigkeit

Anders als Erzeugung modellieren wir *Konsistenzabhängigkeiten* als binäre und ungerichtete Assoziationen. Diese Einschränkung der Stelligkeit auf binäre Assoziationen wäre im Vorgehens-Meta-Modell nicht zwingend erforderlich, motiviert sich aber an dieser Stelle aus der Beschränkung des Vorgehens-Meta-Modells auf typische Planungsmuster. Soll auf Projektebene beispielsweise die Konsistenz dreier Produktexemplare hergestellt werden, wird die Planung häufig zunächst die Konsistenzsicherung zweier dieser Produktexemplare vorsehen.

(28)

Konsistenzabhängigkeiten verfügen also nur und genau über zwei Zielprodukte:

$$\forall k \in \text{Konsistenzabhängigkeit: } |k.\text{nach}_{\text{Produkt}}| = 2 \wedge |k.\text{von}_{\text{Produkt}}| = 0$$

Aus Konsistenzabhängigkeiten im Vorgehensmodell leitet sich auf der Projektebene eine Konsistenzsicherung der Produktexemplare über die Erstellungsdauer der jeweiligen Produktexemplare ab. Diese Konsistenzsicherung zweier Produktexemplare verstehen wir als einen zeitlich zu planenden Vorgang beziehungsweise als Vorgang zur Produktbearbeitung beider Produkte. Aus einer Konsistenzabhängigkeit leitet sich zwar keine Reihenfolge der Produkterstellung ab, allerdings eine notwendige Überlappung der jeweiligen Produkterstellungen. Konsistenzabhängigkeiten machen eine Zusammenarbeit der Bearbeiter der Produkte zumindest im Rahmen der Konsistenzsicherung der jeweiligen Produkte erforderlich. Dies formulieren wir in folgendem Constraint.

(29)

Konsistenzabhängigkeiten implizieren eine Überlappung der Vorgänge zur Produkterstellung der in Beziehung stehenden Produkte. Diese Überlappung ergibt sich aus einer gemeinsamen Produktbearbeitung zur Konsistenzsicherung, die von den beiden jeweiligen Produkterstellungen aggregiert wird. Der Ausdruck „k.nach“ in Constraint (29) bezieht sich dabei entsprechend Constraint (28) auf zwei Produkte:

$$\begin{aligned} \forall k \in \text{Konsistenzabhängigkeit: } \exists b \in \text{Produktbearbeitung:} \\ b.\text{name} = \text{„Abgleich von“} + k.\text{nach}_{\text{Produkt}}.\text{name} \wedge \\ k.\text{nach}_{\text{Produkt}}.\text{erstellt}_{\text{Produkterstellung}}.\text{Produktbearbeitung} = b \end{aligned}$$

Produktfluss

Produktflüsse modellieren wir anders als Konsistenzabhängigkeiten als binäre, gerichtete Assoziationen dar. Die Einschränkung der Stelligkeit auf binäre Assoziationen motiviert sich hier ebenfalls aus der Beschränkung des Vorgehens-Meta-Modells auf typische Planungsmuster. Unter dem Gesichtspunkt der Modellierungstechnik wäre eine Erweiterung des Konzeptes auf n-stellige Produktflüsse nicht mit Problemen verbunden. Da wir auf Projektebene allerdings kein typisches Beispiel für diesen Fall angeben können, schränken wir das Vorgehens-Meta-Modell an dieser Stelle ein.

(30)

Produktflüsse verfügen über jeweils genau ein Produkt als Quelle und genau ein Produkt als Ziel:

$$\forall f \in \text{Produktfluss}: | f.\text{von}^{\text{Produkt}} | = | f.\text{nach}^{\text{Produkt}} | = 1$$

Ein Produktfluss zeichnet das Exemplar des Quellprodukts als verbindliche Vorgabe für die Erstellung eines anderen Produktexemplars aus. Das Quellprodukt dient als Eingangsprodukt für die Erstellung des Zielproduktes, das heißt das Quellprodukt muss vor dem Zielprodukt fertig gestellt werden. Produktflüsse implizieren wie Konsistenzabhängigkeiten einen zeitlich zu planenden Vorgang zur Konsistenzsicherung der Produktexemplare. Im Falle von Produktflüssen verstehen wir das Quellprodukt allerdings als Vorgabe. Damit obliegt die Konsistenzsicherung allein dem Bearbeiter des Zielproduktes. Eine Überarbeitung des Quellproduktes ist bei Bedarf natürlich nicht ausgeschlossen, entspricht aber zumindest nicht dem geplanten Fall.

(31)

Produktflüsse implizieren

- eine Ende-Ende-Vorgangsfolge der jeweiligen Vorgänge zur Produkterstellung,
- eine Produktbearbeitung, die Teil der Produkterstellung des Zielproduktes ist und der Konsistenzsicherung dient,
- sowie eine Ende-Anfang-Vorgangsfolge der Produkterstellung des Quellproduktes und der Produktbearbeitung zur Konsistenzsicherung:

$$\begin{aligned} \forall pf \in \text{Produktfluss}: & \exists ee \in \text{EE-Vorgangsfolge}: \\ & ee.\text{von}^{\text{Produkterstellung}} = pf.\text{von}^{\text{Produkt}}.\text{erstellt}^{\text{Produkterstellung}} \wedge \\ & ee.\text{nach}^{\text{Produkterstellung}} = pf.\text{nach}^{\text{Produkt}}.\text{erstellt}^{\text{Produkterstellung}} \wedge \\ & \exists b \in \text{Produktbearbeitung}: \\ & b.\text{enthält}^{\text{Produkterstellung}} = pf.\text{nach}^{\text{Produkt}}.\text{erstellt}^{\text{Produkterstellung}} \wedge \\ & b.\text{name} = \text{„Abgleich“} + pf.\text{von}^{\text{Produkt}}.\text{name} + \text{„und“} + pf.\text{nach}^{\text{Produkt}}.\text{name} \wedge \\ & \exists ea \in \text{EA-Vorgangsfolge}: \\ & ea.\text{von}^{\text{Produkterstellung}} = pf.\text{von}^{\text{Produkt}}.\text{erstellt}^{\text{Produkterstellung}} \wedge \\ & ea.\text{nach}^{\text{Produktbearbeitung}} = b \end{aligned}$$

In Constraint (31) wählen wir eine Ende-Ende-Vorgangsfolge als den aus einem Produktfluss resultierenden Ablauf. Dabei handelt es sich nur um eine mögliche Interpretation der Semantik eines Produktflusses. Im Kontext eines reinen Top-Down-Vorgehens können hier alternativ auch Ende-Anfang-Vorgangsfolgen für Produktflüsse gewählt werden, das heißt sequentielle Abläufe ohne Überlappung der Vorgänge. Im Rahmen des Tailorings eines Vorgehensmodells kann in Bezug auf die mit Produktflüssen verbundenen Abläufe eine projektspezifische Auswahlmöglichkeit vorgesehen werden (vgl. dazu Abbildung 6–5 und Abbildung 6–7 in Kapitel 6).

Produktabhängigkeiten *Bezug* konzipieren wir für die Modellierung feingranularer Zusammenhänge des Produktmodells. Bezüge dienen der Modellierung von Zusammenhängen der Produkte, die nicht in Form von Erzeugung, Konsistentabhängigkeit oder Produktfluss dargestellt werden können. Bezüge implizieren weder einen Ablauf noch eine Konsistenzsicherung. Mit Hilfe von Bezügen kann die projektspezifische Produktstruktur durch die Angabe von Multiplizitäten auf Ebene des Vorgehensmodells eingeschränkt werden, wie wir im folgenden Abschnitt sowie in Kapitel 5.4.1 anhand von Beispielen zeigen.

Bezug

Bezüge können von Konsistenzabhängigkeiten oder Produktflüssen aggregiert werden und damit Details dieser Produktabhängigkeiten widerspiegeln, wie wir ebenfalls im Folgenden anhand von Beispielen zeigen. Erzeugungen dagegen aggregieren keine Bezüge, da Erzeugungen als Hinweise für die Projektplanung nur temporär gültig sind und im Projektverlauf nicht weiter verfolgt werden.

Beispiel: Vorgehensmodell und Projektebene

Abbildung 5–16 zeigt Produktabhängigkeiten auf Ebene des Vorgehensmodells am Beispiel. Das Beispiel lehnt sich eng an das Vorgehen im Projekt WEIT an (vgl. [WEIT04]). Wir erläutern zunächst die dargestellten Produkte und dann die jeweiligen Produktabhängigkeiten.

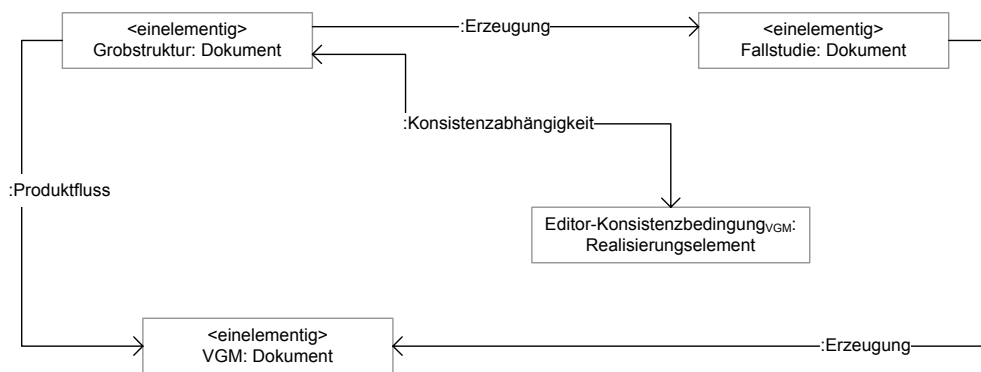


Abbildung 5–16: Produktabhängigkeiten – Ebene Vorgehensmodell (Beispiel)

Ebene Vorgehensmodell

Das Dokument *Grobstruktur* dient der Festlegung des Vorgehens-Meta-Modells des in unserem Beispiel zu erstellenden Vorgehensmodells *VGM*. In der *Grobstruktur* werden im Beispiel – analog zum Ziel der vorliegenden Arbeit – die Beschreibungsmittel definiert, die zur Beschreibung des *VGM* eingesetzt werden sollen. Zusätzlich spezifiziert die *Grobstruktur* Konsistenzbedingungen für die Anwendung der Beschreibungsmittel. Diese spezifizierten Konsistenzbedingungen müssen als *Editor-Konsistenzbedingungen* realisiert und zur Unterstützung der Bearbeiter des *VGM* in den *VGM*-Editor integriert werden.

Die *Fallstudie* beinhaltet eine prototypische Konzeption der Inhalte des *VGM* entsprechend den Beschreibungsmitteln des *VGM*. Die *Fallstudie* evaluiert die *Grobstruktur* und dient als Vorbereitung für die inhaltliche Umsetzung des *VGM*.

Grobstruktur und *Editor-Konsistenzbedingungen* stehen über eine *Konsistenzabhängigkeit* in Beziehung. Beide Produktexemplare werden also zeitlich überlappend erstellt. Die Konsistenz beider Produktexemplare wird von den jeweiligen Bearbeitern sichergestellt. Die *Grobstruktur* verstehen wir also nicht im Sinne eines Produktflusses als Vorgabe. Dies bringt den Vorteil mit sich, dass vor der endgültigen Spezifikation einer Konsistenzbedingung des *VGM* in der *Grobstruktur* bereits die Realisierbarkeit überprüft werden kann und entsprechendes Feedback in die *Grobstruktur* einfließen kann. Die Produktbearbeitungen zur Konsistenzsicherung, die sich aus Konsistenzabhängigkeit nach Constraint (29) und Produktfluss nach Constraint (31) auf der Ebene des Vorgehensmodells ergeben, stellen wir in Abbildung 5–16 nicht dar. Diese zeigen wir im Folgenden auf Projektebene.

Die *Grobstruktur* enthält die Spezifikation der Beschreibungsmittel und wirkt sich daher auf die prototypische Konzeption der Inhalte in der *Fallstudie* aus. Dieser Zusammenhang ist über eine Produktabhängigkeit der Kategorie *Erzeugung* modelliert. Die hier modellierte Erzeugung dient als Hinweis für den Projektleiter, bei der Planung der Erstellung der Inhalte

der Fallstudie die Grobstruktur zu berücksichtigen beziehungsweise bei der Planung der Erstellung der Grobstruktur die Fallstudie zu berücksichtigen. Die Planung beider Produkte unterliegt keinen zeitlichen Einschränkungen hinsichtlich einer Reihenfolge der Produkt-erstellungen. Auch die Planung einer Konsistenzsicherung durch die Bearbeiter auf Projekt-ebene leitet sich aus Produktabhängigkeiten der Kategorie Erzeugung nicht ab. Im Rahmen der Erstellung eines Prototyps ist dies nicht erforderlich.

Die Fallstudie trägt zudem zur Erzeugung der Inhalte des VGM bei, da in der Fallstudie bereits erste Erfahrungen mit der inhaltlichen Umsetzung des VGM gemacht werden. Für die erste Planung der Erstellung des VGM kann der Projektleiter also die Fallstudie zu Rate ziehen. Im weiteren Projektverlauf werden sich die Inhalte des VGM allerdings erwartungs-gemäß weiterentwickeln. Die Fallstudie muss nicht im gesamten Projektverlauf mit dem VGM konsistent gehalten werden und erfüllt also den Zweck eines „Wegwerf-Prototypen“. Die Reihenfolge der Erstellung von Fallstudie und VGM leitet sich nicht aus der Erzeugung ab. Allerdings stellt die Einteilung des Projektes in Phasen die Fertigstellung der Fallstudie vor dem Beginn der Erstellung des VGM bereits sicher (vgl. Kapitel 5.2.4, Abbildung 5–10).

Grobstruktur und VGM sind über einen *Produktfluss* verbunden. Grobstruktur und VGM sollen also für die Dauer ihrer Erstellung konsistent gehalten. Die Grobstruktur fungiert dabei als Vorgabe für das VGM. Die Produktbearbeitung zur Konsistenzsicherung ist allein dem Produkterstellungsvorgang des VGM und damit den Bearbeitern des VGM zugeordnet. Im Falle der Grobstruktur ist anders als bei der Fallstudie die Aktualität der Inhalte von Bedeu-tung, da eine Dokumentation der Beschreibungsmittel sowohl für den Erstellungsprozess als auch den anschließenden Weiterentwicklungsprozess des VGM benötigt wird.

Ein Produktfluss kann wie im Falle des Produktflusses zwischen Grobstruktur und VGM der Idee der Erzeugung dahingehend entsprechen, dass die Exemplare der Quellprodukte eine Spezifikation der Existenz der Zielproduktexemplare enthalten. Die zusätzliche Kennzeich-nung eines Produktflusses als Erzeugung sehen wir in Abbildung 5–16 allerdings nicht vor, da dieser Zusammenhang der Produkte bereits in Form des Produktflusses zum Ausdruck kommt. Produktabhängigkeiten der Kategorie Erzeugung verwenden wir allein dann, wenn die Zusammenhänge der Produkte nicht bereits durch andere Produktabhängigkeiten model-liert werden.

Abbildung 5–17 und Abbildung 5–18 führen obiges Beispiel auf Projektebene fort. Abbildung 5–17 zeigt auf Projektebene *Grobstruktur* und *Fallstudie*, die über eine Produktabhängigkeit der Kategorie *Erzeugung* verbunden sind. Zusammenhänge der Kategorie Erzeugung sind im Vorgehensmodell hinsichtlich ihrer Auswirkungen auf die Projektebene nicht präzise modelliert. Die Strukturierung und Planung der jeweiligen Produktinhalte obliegt also dem Projektleiter.

Projektebene

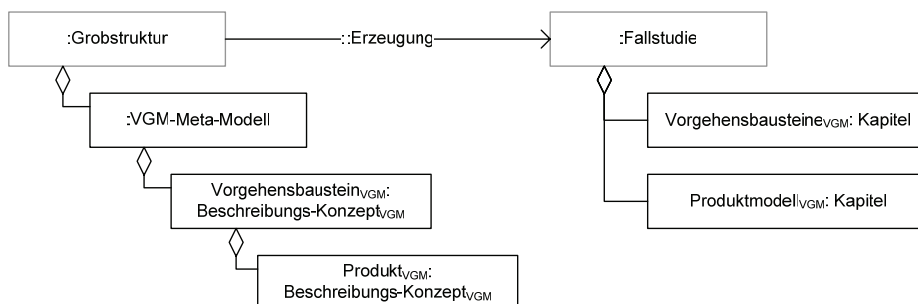


Abbildung 5–17: Produktabhängigkeiten der Kategorie Erzeugung – Projektebene (Bei-spiel)

In der Grobstruktur werden *Vorgehensbausteine* und *Produkte* als Beschreibungsmittel des VGM konzipiert. Die Fallstudie wird entsprechend in Form von zwei Kapiteln *Vorgehensbausteine* und *Produktmodell* strukturiert. Vorgänge zur Produkterstellung werden entsprechend geplant, sind in Abbildung 5–17 aber nicht dargestellt. Eine Reihenfolge der Produkterstellung ist durch die Produktabhängigkeit nicht gegeben. Die Fallstudie könnte also im Projekt auch vor der Grobstruktur erstellt werden.

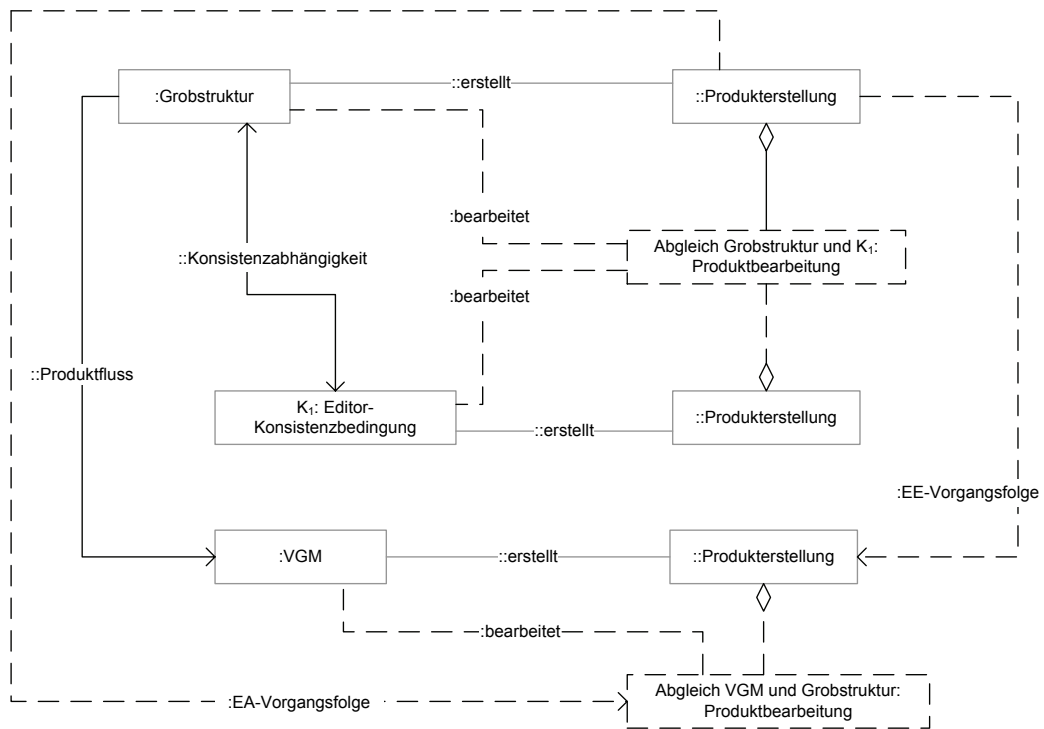
Produktabhängigkeiten der Kategorie Erzeugung sind n-stellige Beziehungen. Beispielsweise könnte zur Erzeugung der Kapitel der Fallstudie neben der Grobstruktur auch das Produkt Anforderungspapier herangezogen werden. Das Anforderungspapier enthielte eine Beschreibung, welche Inhalte in der Fallstudie prototypisch zu evaluieren sind. Die Erzeugung in Abbildung 5–17 hätte dann zwei Produkte als Quelle. Die Erzeugung von Fallstudie und VGM aus der Grobstruktur könnte ebenfalls in einer Erzeugung mit zwei Zielprodukten zusammengefasst sein. Da Erzeugung nur als Hinweis dient und keine Präzision erfordert, sind viele mögliche Modellierungen vorstellbar.

Abbildung 5–18 verdeutlicht auf Projektebene die Implikation von Abläufen und Produktbearbeitungen durch Produktabhängigkeiten der Kategorie Konsistenzabhängigkeit beziehungsweise Produktfluss. Abbildung 5–18 (a) ist dabei als Klassendiagramm notiert, Abbildung 5–18 (b) als Balkendiagramm. In der Balkendiagramm-Darstellung ist die Aggregation der Produktbearbeitung „*Abgleich VGM und Grobstruktur*“ sowohl durch die Produkterstellung „*Grobstruktur erstellen*“ als auch die Produkterstellung „*Konsistenzbedingung K1 erstellen*“ in Form gestrichelter Linien dargestellt.

Die *Konsistenzabhängigkeit* zwischen Grobstruktur und der Spezialisierung K_1 der Editor-Konsistenzbedingung führt gemäß Constraint (29) zu einer gemeinsamen Produktbearbeitung „*Abgleich Grobstruktur und K_1* “ der jeweiligen Produkterstellungen. Aufgrund der zeitlichen Einschränkungen als Folge der Aggregation von Vorgängen überlappen sich die Produkterstellungen mindestens für die Dauer dieser gemeinsamen Produktbearbeitung.

Der *Produktfluss* von der Grobstruktur zum VGM impliziert eine Produktbearbeitung zur Konsistenzsicherung „*Abgleich VGM und Grobstruktur*“, die allein der Produkterstellung des VGM über Aggregation zugeordnet ist. Zudem sind die Produkterstellungen der Grobstruktur und des VGM durch eine Ende-Ende-Vorgangsfolge verbunden. Die Produkterstellung der Grobstruktur und der „*Abgleich zwischen VGM und Grobstruktur*“ sind durch eine Ende-Anfang-Vorgangsfolge verbunden. Die Erstellung des VGM kann also erst abgeschlossen werden, wenn die Grobstruktur fertig gestellt ist. Die Produktbearbeitung zur Konsistenzsicherung kann erst begonnen werden, wenn die Grobstruktur fertig gestellt ist. Ein Beispiel einer Terminplanung passend zu diesen zeitlichen Rahmenbedingungen lässt sich der Abbildung 5–18 (b) entnehmen.

(a) als Klassendiagramm



(b) als Balkendiagramm

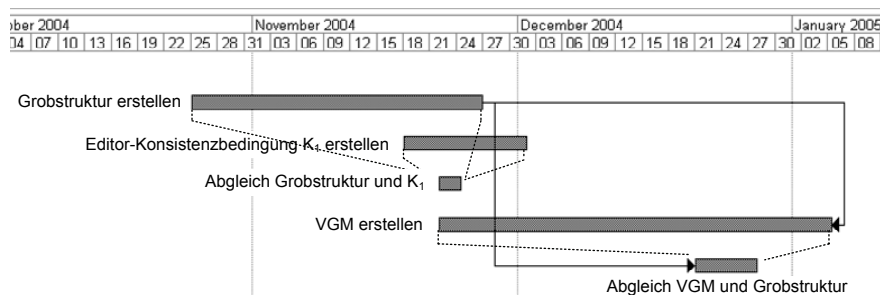
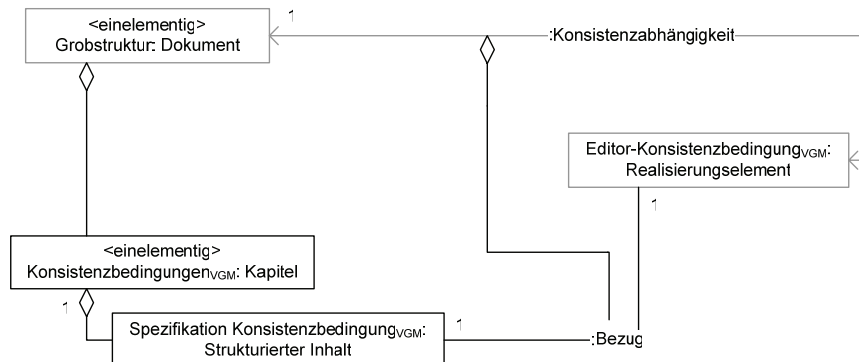


Abbildung 5–18: Produktabhängigkeiten: Implikation von Vorgängen – Projektebene (Beispiel)

Abbildung 5–19 (a) zeigt ein Beispiel für Produktabhängigkeiten der Kategorie *Bezug* auf Ebene des Vorgehensmodells. Abbildung 5–19 (b) führt dieses Beispiel auf Projektebene fort. Bezüge können durch Konsistenzabhängigkeiten und Produktflüsse aggregiert werden. Bezüge dienen der Modellierung von Zusammenhängen der Produkte, die nicht in Form von Erzeugung, Konsistentabhängigkeit oder Produktfluss modelliert werden können. Bezüge implizieren weder Abläufe noch Produktbearbeitungen. Für die Planung eines eigenen Vorgangs zur Konsistenzsicherung ist die Ebene, auf der Bezüge modelliert werden, oft zu feingranular.

(a) Ebene Vorgehensmodell



(b) Projektebene

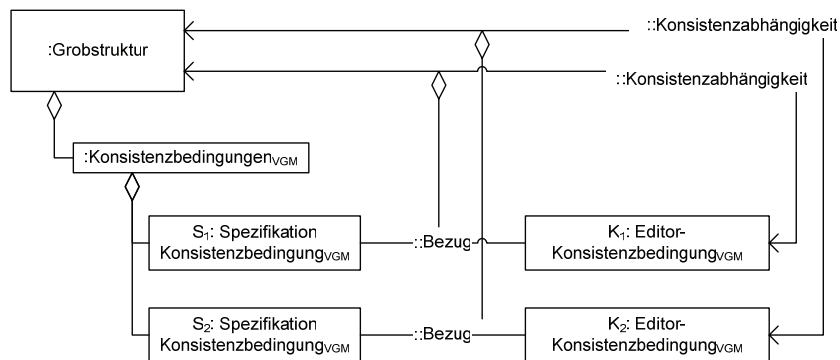


Abbildung 5–19: Produktabhängigkeiten: Aggregation von Bezügen – Ebene Vorgehensmodell und Projektebene (Beispiel)

Im Beispiel enthält das Dokument *Grobstruktur* ein Kapitel *Konsistenzbedingungen*, das Spezifikationen der geltenden Konsistenzbedingungen in Form „strukturierter Inhalte“ enthält. Jede in der Grobstruktur spezifizierte Konsistenzbedingung muss durch eine *Editor-Konsistenzbedingung* realisiert werden. Dieser Zusammenhang wird durch eine Produktabhängigkeit *Bezug* mit geeigneten Multiplizitäten ausgedrückt. Jeder Spezifikation einer Konsistenzbedingung des VGM ist genau eine Editor-Konsistenzbedingung als Realisierung zugeordnet.

Da es sich dabei um Details der Konsistenzabhängigkeit zwischen Grobstruktur und Editor-Konsistenzbedingung handelt, modellieren wir den Bezug als Aggregation dieser Konsistenzabhängigkeit. Die Modellierung einer Konsistenzabhängigkeit auf feingranularer Ebene bietet sich in diesem Beispiel nicht an, da dies auf Projektebene die Planung eigener Konsistenzsicherungsvorgänge für jede Konsistenzbedingung des VGM nach sich ziehen würde. Abbildung 5–19 (b) führt das Beispiel auf Projektebene fort. Weitere Beispiele für Produktabhängigkeiten der Kategorie *Bezug* finden sich im folgenden Kapitel 5.4.

Am Beispiel in Abbildung 5–19 zeigt sich zudem die mögliche Planungsrelevanz von Produkten der Kategorie „strukturierter Inhalt“. Neben der Planung der Realisierung einer Konsistenzbedingung kann auch die Spezifikation einer Konsistenzbedingung bei der Planung eines Projektes relevant sein. Entsprechend Kapitel 5.2.3 sind „strukturierte Inhalte“ wie alle

Produkte mit einem Vorgang zur Produkterstellung verbunden, der auch zeitlich geplant werden kann.

Produktabhängigkeiten der Kategorie **Erzeugung** dienen als Hinweise für die Erstellung des Projektplans. Produktexemplare spezifizieren dabei die Existenz von Produktexemplaren eines anderen Produktes. Erzeugung erfordert das Lesen und Interpretieren der Inhalte der jeweiligen Produktexemplare durch den Projektleiter. Da es in der Praxis nicht lohnenswert ist, jeden Produktinhalt beziehungsweise Zusammenhang mit anderen Produkten als Vorgabe für einen Projektplan präzise zu modellieren und damit einer Werkzeugunterstützung zugänglich zu machen, ist ein Konzept wie die Erzeugung unabdingbar. Mit einer Erzeugung sind keine Konsistenzsicherung der Produktexemplare und keine Reihenfolge der Produkterstellung verbunden.

Zusammenfassung

Produktabhängigkeiten der Kategorie Konsistenzabhängigkeit und Produktfluss sehen wir als typische Muster für die Planung der Zusammenarbeit mehrerer Bearbeiter bei der Erstellung mehrerer in Beziehung stehender Produkte an. Anders als im V-Modell XT stellen Produktabhängigkeiten hier keine Anforderungen an Konfigurationsmanagement beziehungsweise Zustände bestimmter Produktexemplare. Das Vorgehens-Meta-Modell haben wir in Bezug auf Produktabhängigkeiten bewusst nicht mit dem Anspruch auf Vollständigkeit modelliert, sondern lediglich als Hilfsmittel für typische Planungen. Andere Arten von Abläufen und Vorgängen zur Konsistenzsicherung können allerdings durch unmittelbare Spezialisierung der Konzepte Ablauf oder Produktbearbeitung im Vorgehensmodell modelliert werden.

Eine **Konsistenzabhängigkeit** zweier Produkte impliziert eine Produktbearbeitung zur Prüfung beziehungsweise Herstellung der Konsistenz der entsprechenden Produktexemplare. Diese Produktbearbeitung ist den jeweiligen Produkterstellungen gemeinsam, das heißt sie wird von beiden Produkterstellungen aggregiert. Die Konsistenzsicherung erfordert also eine Zusammenarbeit der Bearbeiter dieser Produktexemplare. Die Produkterstellungen über Konsistenzabhängigkeiten verbundener Produkte überlappen sich zeitlich mindestens für die Dauer dieser gemeinsamen Konsistenzsicherung.

Produktflüsse kennzeichnen ein Produktexemplar als Vorgabe für ein anderes Exemplar eines Produktes. Die Herstellung der Konsistenz beider Produktexemplare obliegt allein dem Bearbeiter des Zielproduktexemplars. Produktflüsse implizieren also eine Produktbearbeitung zur Konsistenzsicherung, die von der Produkterstellung des Zielproduktes aggregiert wird. Ein Produktfluss hat zudem Auswirkungen auf die Reihenfolge der Produkterstellungen. Diese Auswirkungen können im Rahmen des Tailorings beispielsweise hinsichtlich Ende-Ende- oder Ende-Anfang-Vorgangfolgen ausgeprägt werden (vgl. dazu Kapitel 6.2).

Verglichen mit dem Konzept des V-Modells XT sind Produktabhängigkeiten hier als konkreter einzustufen, da wir mit Produktabhängigkeiten Reihenfolgen beziehungsweise zeitliche Überlappungen der Produkterstellung verbinden. Produktabhängigkeiten können also der Beschreibung konkreter beispielsweise organisationsspezifischer Vorgehensmodelle dienen. Allerdings können die hier konzipierten Produktabhängigkeiten auch für ein im Abstraktionsgrad dem V-Modell XT vergleichbares Vorgehensmodell Anwendung finden. Beispielsweise erscheint es sinnvoll, zwischen Produkten wie der „Gefährdungs- und Systemsicherheitsanalyse“ und der „Systemarchitektur“ des V-Modells XT eine Konsistenzabhängigkeit zu modellieren, da sich die Erstellung beider Produkte im Projekt aufgrund starker gegenseitiger Abhängigkeiten überlappen sollte.

Produktabhängigkeiten der Kategorie **Bezug** dienen der Modellierung feingranularer Zusammenhänge des Produktmodells, die nicht durch die übrigen Arten von Produktabhängigkeiten modelliert werden können. Bezüge und Multiplizitäten dienen dem Prozessingenieur als Sprachmittel zur Einschränkung der projektspezifischen Produktstruktur.

5.4 Spezialisierungskonzept

Wir betrachten in diesem Kapitel das Spezialisierungskonzept im Zusammenhang mit dem Modellierungsansatz in Form von (Meta-)Ebenen. Spezialisierung dient der Abgrenzung des Vorgehens-Meta-Modell, des Vorgehensmodell und der Projektebene (vgl. Kapitel 4.3, Abbildung 4–12). Das vorliegende Kapitel erläutert den Einsatz von Spezialisierung anhand von Beispielen. Eine Erweiterung des Vorgehens-Meta-Modells ist also anders als in den vorangehenden Kapiteln im vorliegenden Kapitel nicht erforderlich.

In Kapitel 5.4.1 stellen wir dar, dass ein Spezialisierungskonzept auf der Ebene des Vorgehensmodells erforderlich ist. In jedem Entwicklungsprojekt ist Modellbildung von Bedeutung. Modellbildung beziehungsweise Generalisierung und damit Spezialisierung müssen also auch im Vorgehensmodell beschreibbar sein. Modellbildung stellt insbesondere einen Einflussfaktor auf die Projektplanung dar.

In Kapitel 5.4.2 stellen wir die Notwendigkeit eines Spezialisierungskonzeptes im Vorgehensmodell aus dem Blickwinkel der Vermeidung von Redundanzen dar. Dabei nutzen wir als Beispiel das V-Modell XT, das insbesondere bei der Beschreibung der Entwicklung von Systemen zahlreiche Redundanzen aufweist.

Zudem zeigen wir in Kapitel 5.4.3, dass Spezialisierung als alleinige Beziehung zwischen den Ebenen Vorgehens-Meta-Modell und Vorgehensmodell nicht ausreichend ist. Dies verdeutlichen wir am Beispiel der Qualitätssicherung als so genannter generischer Vorgehensweise.

Abschnitte in Kapitel 5.4	Seite
5.4.1 Spezialisierung auf Ebene des Vorgehensmodells	142
5.4.2 Studie: Spezifische Ausprägungen des V-Modells XT	147
5.4.3 Beziehungen zwischen Spezialisierungs-Ebenen: Generische Vorgehensweisen	148

5.4.1 Spezialisierung auf Ebene des Vorgehensmodells

Soll aus einem Vorgehensmodell ein Projektplan abgeleitet werden, ist ein Spezialisierungskonzept auf der Ebene des Vorgehensmodells beziehungsweise auf Projektebene unumgänglich. Modellbildung durch Generalisierung ist für Entwicklungsprojekte eine sehr grundlegende Technik. Ein Vorgehensmodell muss also Generalisierung und Spezialisierung als Arbeitstechnik auf Projektebene zur Verfügung stellen. Aus Sicht der Projektplanung hat Modellbildung oftmals Auswirkungen auf den weiteren Ablauf eines Projektes. Beispielsweise leiten sich aus einem Modell wie der Software-Architektur die zu entwickelnden und damit einzuplanenden SW-Komponenten sowie deren Realisierungs-Reihenfolge ab.

Zielsetzung dieses Kapitels ist die Darstellung der Notwendigkeit eines Spezialisierungskonzeptes im Vorgehensmodell beziehungsweise auf Projektebene anhand von Beispielen. Wir verwenden dabei das bereits definierte Spezialisierungskonzept der Modellierungstechnik (vgl. Kapitel 4.2.2).

Beispiel: Vorgehensmodell und Projektebene

Abbildung 5–20 zeigt ein Beispiel eines Vorgehensmodells, das Spezialisierungsbeziehungen beinhaltet. Wir verwenden als durchgängiges Beispiel weiterhin ein Projekt zur Erstel-

lung des Vorgehensmodells VGM. In Abbildung 5–20 und auch in Abbildung 5–21 stellen wir vereinfachend Bezüge und Spezialisierung in Form einer gemeinsamen Linie dar. Es handelt sich allerdings genau genommen um zwei Konzepte, da entsprechend der Definition der Modellierungstechnik Spezialisierung keine Assoziation ist, sondern ein eigenständiges Konzept.

In diesem Beispiel gehen wir davon aus, dass ein Meta-Modell für das VGM im Projekt entwickelt werden muss. Modellbildung ist nicht allein Thema der vorliegenden Arbeit, sondern vielmehr für jedes Entwicklungsprojekt eine gängige Technik. Die Betrachtung eines Software-Entwicklungsprojektes wäre also als Beispiel in diesem Kapitel ebenfalls geeignet. Eine dem Meta-Modell des VGM vergleichbare Modellbildung erfolgt in einem Software-Entwicklungsprojekte beispielsweise in einer technischen Architektur, die Konzepte für die Beschreibung von Komponenten und Schnittstellen definiert.

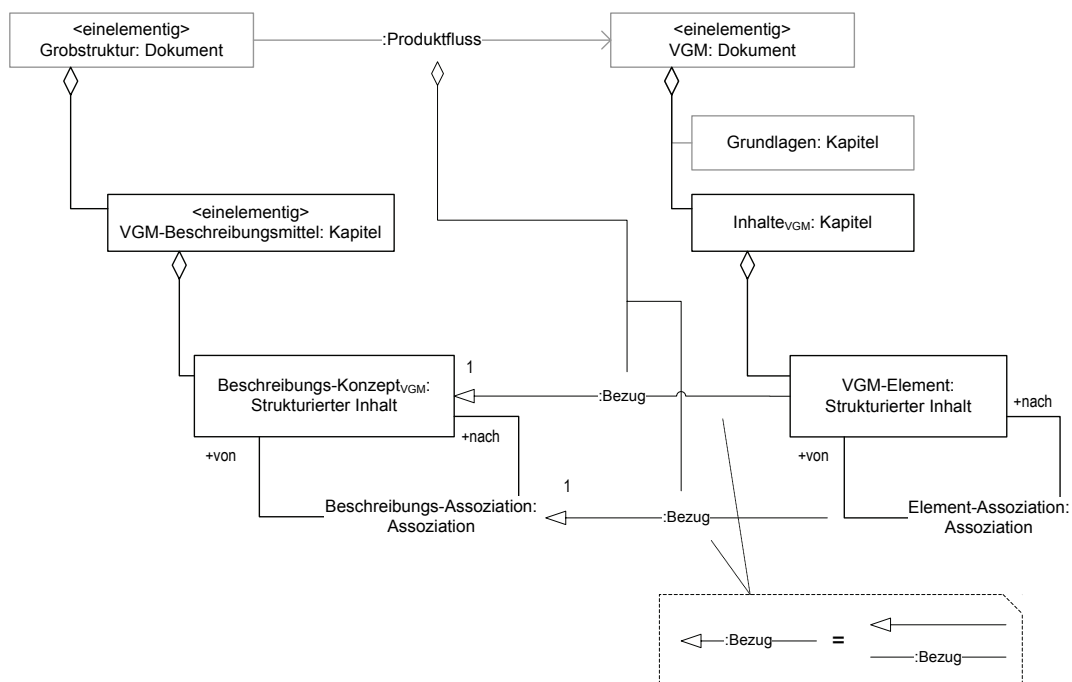


Abbildung 5–20: Spezialisierung auf Ebene des Vorgehensmodells (Beispiel)

Die *Grobstruktur* beinhaltet im Beispiel ein Kapitel *VGM-Beschreibungsmittel*. Die Beschreibungsmittel werden in Form „*strukturierter Inhalte*“ und *Assoziationen* beschrieben. Die *Grobstruktur* besteht aus *Beschreibungs-Konzepten* und *Beschreibungs-Assoziationen*. Dabei handelt es sich um eine sehr generische Vorgabe zur Beschreibung eines Meta-Modells, die in ihrer Ausdrucksmächtigkeit der Modellierungstechnik basierend auf Klassen und Assoziationen entspricht. Das Beispiel dieses Kapitels erachten wir daher nicht für praxisrelevant, aber doch für die Verdeutlichung der Idee des vorliegenden Kapitels als hinreichend.

Das *VGM* besteht neben den *Grundlagen*-Kapiteln aus Kapiteln, die die eigentlichen *Inhalte* des *VGM* beschreiben. Das Kapitel *Inhalte* enthält *VGM-Elemente*, die über *Element-Assoziationen* verbunden sind. *VGM-Elemente* *beziehen* sich eindeutig auf die in der *Grobstruktur* definierten *Beschreibungs-Konzepte*. *Element-Assoziationen* *beziehen* sich eindeutig auf die *Beschreibungs-Assoziationen* der *Grobstruktur*.

Der zwischen Grobstruktur und VGM modellierte *Produktfluss* aggregiert die Bezüge zwischen Beschreibungs-Konzepten und VGM-Elementen sowie zwischen Beschreibungs-Assoziationen und Element-Assoziationen. Entsprechend dem Vorgehens-Meta-Modell können Produktabhängigkeiten der Kategorie Bezug von Produktabhängigkeiten der Kategorie Produktfluss aggregiert werden (vgl. Kapitel 5.3.2).

Abbildung 5–21 führt das Beispiel auf Projektebene fort. Dabei können Teile der Abbildung auch als Vorgehensmodell verstanden werden, wie wir im Folgenden noch erläutern.

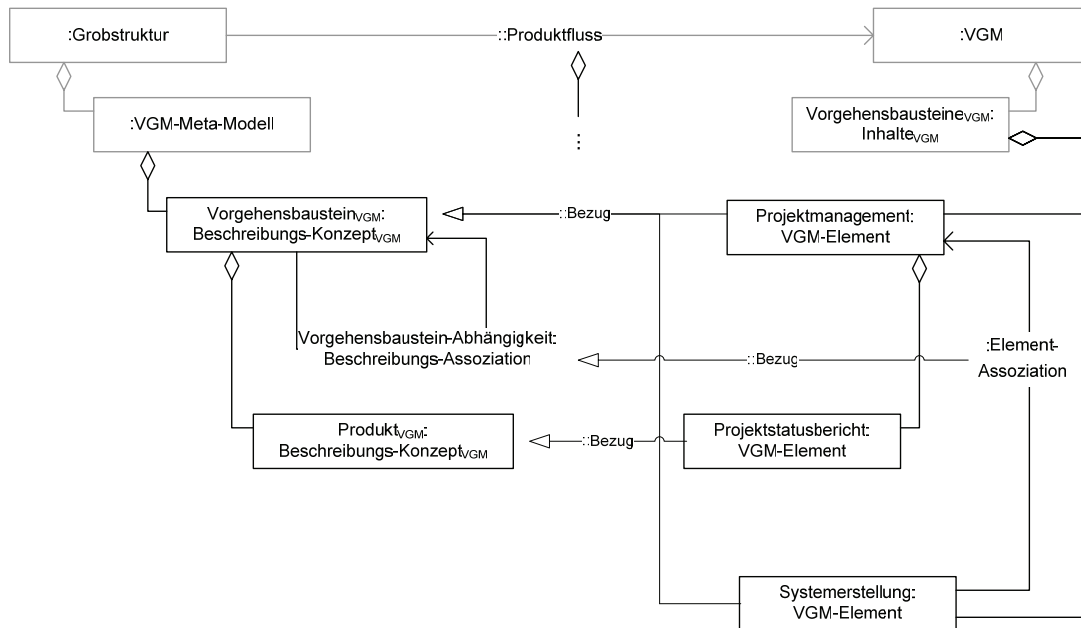


Abbildung 5–21: Spezialisierung auf Projektebene beziehungsweise auf Ebene des Vorgehensmodells (Beispiel)

Die *Grobstruktur* definiert im Beispiel auf Projektebene *Vorgehensbausteine* und *Produkte* als *Beschreibungs-Konzepte* des VGM. Vorgehensbausteine können über *Vorgehensbaustein-Abhängigkeit* assoziiert werden.

Das *VGM* enthält entsprechend dem Vorgehensmodell spezialisierte *VGM-Elemente* und *Element-Assoziationen*. VGM-Elemente sind im Beispiel *Projektmanagement*, *Projektstatusbericht* und *Systemerstellung*. Diese sind über Produktabhängigkeiten der Kategorie *Bezug* entsprechend dem Vorgehensmodell den Beschreibungs-Konzepten Vorgehensbaustein beziehungsweise Produkt der Grobstruktur zugeordnet. Projektmanagement und Systemerstellung sind zudem Spezialisierungen des Beschreibungs-Konzeptes Vorgehensbaustein. Der Projektstatusbericht ist eine Spezialisierung des Beschreibungs-Konzeptes Produkt. Im Kontext von Programmiersprachen würde man von Mehrfach-Vererbung sprechen.

Verschachtelte Spezialisierungsebenen

Abbildung 5–22 zeigt die Spezialisierungs-Ebenen des Beispiels im Überblick. Die Ebene des Vorgehensmodells besteht aus dem *VGM*, dem die Rolle eines Meta-Modells in Bezug auf das VGM auf Projektebene zukommt, sowie den „*Beschreibungsmittel VGM*“, die als Meta-Meta-Modell in Bezug auf die „*Beschreibungsmittel VGM*“ auf Projektebene verstanden werden können.

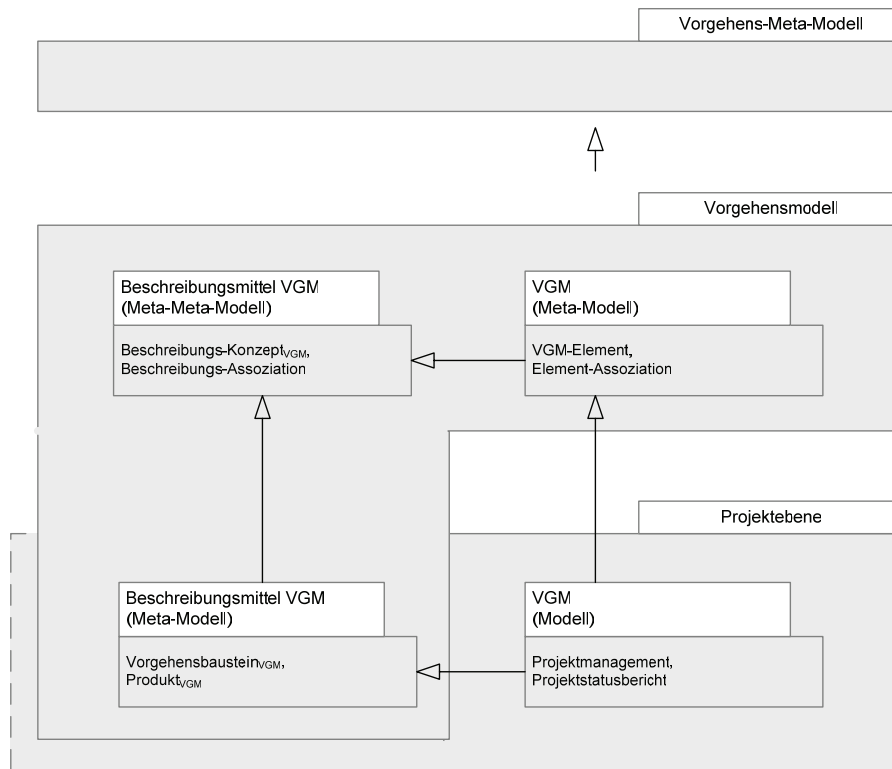


Abbildung 5–22: Spezialisierungs-Beziehungen aus Sicht des Beispiel-Projektes im Überblick

Die „Beschreibungsmittel VGM“, die im Projekt erarbeitet und in Form von Vorgehensbausteinen und Produkten definiert werden, können sowohl der Projektebene als auch der Ebene des Vorgehensmodells zugerechnet werden. Einerseits werden die Konzepte Vorgehensbaustein und Produkt im Projekt erarbeitet. Andererseits dienen sie im weiteren Projektverlauf als Vorgabe und können somit auch als Teil des Vorgehensmodells für dieses Projekt verstanden werden.

Das Beispiel dieses Kapitels sieht also die Konzeption eines Vorgehens-Meta-Modells in einem Projekt vor. Das Beispiel ist allein vor dem Hintergrund sinnvoll, dass Vorgehensmodelle und auch Vorgehens-Meta-Modelle in der Praxis nicht vollständig sein können. Als Beispiel für die Unvollständigkeit des in der Arbeit konzipierten Vorgehens-Meta-Modells können die in Kapitel 5.3.2 modellierten Produktabhängigkeiten gelten, die eben lediglich typische Planungs-Muster widerspiegeln. Zudem hat das Beispiel seine Berechtigung, da mit Vorgehensmodellen und damit auch mit Vorgehens-Meta-Modellen unterschiedliche Zielsetzungen verfolgt werden. Das VGM mag spezielleren oder anderen Anforderungen als das in der Arbeit konzipierte Vorgehens-Meta-Modell genügen. Beispielsweise könnte das VGM analog dem V-Modell XT eine übersichtlich strukturierte Beschreibung für alle Projektbeteiligten fokussieren, während das in der vorliegenden Arbeit konzipierte Vorgehens-Meta-Modell auf die Projektplanung abzielt. Um ein künstliches Beispiel handelt es sich allerdings dennoch, da das Meta-Meta-Modell auf Vorgehensmodell-Ebene vereinfachend weitgehend dem Konzept des Meta-Meta-Modells der vorliegenden Arbeit entspricht, also der Modellierungstechnik basierend auf Klassen.

Abbildung 5–22 zeigt lediglich die temporäre Sichtweise des Beispielprojektes zur Erstellung des VGM. Wir gehen allerdings von der Notwendigkeit eines lebendigen Vorgehensmodells und damit auch Vorgehens-Meta-Modells aus. Damit stellt sich diese Sichtweise nicht als

ungewöhnlich dar. Die im Beispielprojekt erarbeiteten „Beschreibungsmittel VGM“ können natürlich mit der Ebene des Vorgehens-Meta-Modells in Abbildung 5–22 integriert werden. Zudem würde das Meta-Modell auf Ebene des Vorgehensmodells durch das Meta-Modell auf Projektebene abgelöst werden. Eine Notwendigkeit zweier Generalisierungen des VGM bestünde bei einer Integration der Modelle letztlich nicht. Ein Konzept wie „Mehrfach-Spezialisierung“ ist also nur temporär erforderlich.

Auswirkungen auf die Planung

Im Beispiel hat die Konzeption der Beschreibungsmittel des VGM Auswirkungen auf die weitere Planung des Projektes. Im Beispiel sind Vorgehensbausteine des VGM analog zum V-Modell XT über Vorgehensbaustein-Abhängigkeiten verbunden. Bestimmte Vorgehensbausteine stellen also eine erweiterbare Basis für andere Vorgehensbausteine dar (vgl. Kapitel 3.1.1). Dieser Zusammenhang impliziert Produktflüsse und damit Abläufe bezüglich der Erstellung der Vorgehensbausteine des VGM. Das folgende Constraint wird in unserem Beispiel auf Projektebene definiert, kann allerdings auch als Teil des Vorgehensmodells verstanden werden. Abbildung 5–23 verdeutlicht die Auswirkung des Constraints (32) für unser Beispiel.

(32)

Die auf Projektebene definierten Element-Assoziationen, die zugleich spezialisierte Vorgehensbaustein-Abhängigkeiten sind, implizieren einen Produktfluss in der der jeweiligen Element-Assoziation entgegen gesetzten Richtung:

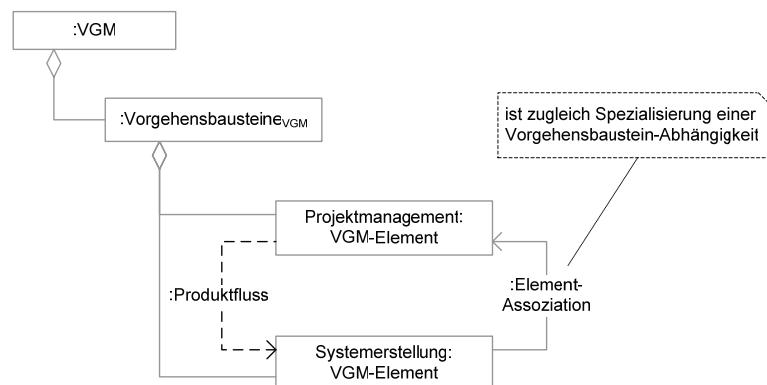
$$\forall ea \in \text{Element-Assoziation}: \exists pf \in \text{Produktfluss}: \\ pf.von = ea.nach \wedge pf.nach = ea.von$$


Abbildung 5–23: Auswirkungen von Beschreibungsmitteln auf die Planung gemäß Constraint (32) (Beispiel)

Beliebige Anzahl von Meta-Ebenen

Die bislang auf dem Gebiet der Modellierung von Vorgehensmodellen verfolgte Architektur sieht eine feste Einteilung in drei (Meta-)Ebenen vor: Vorgehens-Meta-Modell, Vorgehensmodell als Instanz des Meta-Modells, sowie eine Ebene die mit dem Projekt assoziiert ist (vgl. [GMP+01a], [GMP+01b], [GMP+03], [GDM+04] und [VM-XT05]).

Im Rahmen der Arbeit wählen wir einen offeneren Ansatz. Wir verabschieden uns von der Idee der festen Einteilung in drei (Meta-)Ebenen. Als festgelegt betrachten wir die Ebene des Vorgehens-Meta-Modells. Diese Ebene kann mit dem Konzept der Spezialisierung in einen Projektplan überführt werden. Die Anzahl der erforderlichen Spezialisierungen bis hin zum Projektplan ist dabei nicht definiert. Die aktuell jeweils „tiefste“ Spezialisierung nennen wir Projektebene. Dies entspricht den Gegebenheiten der Praxis, da Projektpläne im Projektver-

lauf unterschiedliche Spezialisierungsgrade durchlaufen. Eine formale Definition des Projektplans als Teil der Projektebene geben wir in Kapitel 6.4 an.

Im Rahmen des Beispiels zur Erstellung des VGM liegt es beispielsweise nahe, zunächst die Beschreibungsmittel des VGM zu konzipieren und im Anschluss daran auf Basis der Beschreibungsmittel eine detailliertere Planung für die Erstellung des VGM zu erstellen. Diese Detaillierung der Planung verstehen wir als Spezialisierung. Nach der Definition der Beschreibungsmittel im Projekt wird die Vorgehensweise, die sich aus den Beschreibungsmitteln ableitet, sozusagen auf die Ebene des Vorgehensmodells aus Sicht des VGM-Projektes gehoben. Ein weiteres Beispiel mehrfacher Spezialisierung auf Ebene des Vorgehensmodells findet sich in Kapitel 6.4 (vgl. dazu Abbildung 6–10).

Diese Idee an ihre Grenzen getrieben, kann auch das Vorgehens-Meta-Modell selbst, solange keine Spezialisierungen existieren, als Projektplan verstanden werden. Das Vorgehens-Meta-Modell als Projektplan enthält nichts weiter als die Aussage, dass in der Projektlaufzeit Projektergebnisse in Form von Produkten durch Vorgänge erstellt werden müssen. Die Erstellung von Ergebnissen in einem zeitlich begrenzten Rahmen entspricht der Definition des Begriffs Projekt. Das Vorgehens-Meta-Modell als Projektplan ist also in jedem Fall korrekt. In der Praxis muss das Vorgehens-Meta-Modell als Projektplan aufgrund seines Detaillierungs-Grades natürlich sogar zu Projektbeginn als ungenügend bezeichnet werden. Theoretisch ist der Grad der Detaillierung eines Projektplans allerdings relativ und nicht absolut definierbar.

Das Konzept der Spezialisierung dient der Abgrenzung des Vorgehens-Meta-Modell, des Vorgehensmodell und der Projektebene. Wie wir am Beispiel zeigen, ist ein Spezialisierungskonzept allerdings auch auf der Ebene des Vorgehensmodells erforderlich. Spezialisierung als Mittel der Modellbildung ist eine verbreitete Technik. Als Beispiel haben wir die Konzeption eines Meta-Modells für die Beschreibung eines Vorgehensmodells betrachtet. Dieses auf Projektebene konzipierte Meta-Modell impliziert bestimmte Abläufe für die weitere Planung des VGM-Projektes. Nach Fertigstellung kann dieses Meta-Modell aus Sicht des VGM-Projektes also als Teil des Vorgehensmodells verstanden werden.

Planung verstehen wir als gleichbedeutend mit dem Konzept der Spezialisierung. Die Projektebene und damit den Projektplan definieren wir als die aktuell „tiefste“ Spezialisierungsebene des Vorgehens-Meta-Modells. Eine formale Definition des Projektplans als Teil der Projektebene geben wir in Kapitel 6.4 an.

Zusammenfassung

5.4.2 Studie: Spezifische Ausprägungen des V-Modells XT

In diesem Kapitel zeigen wir eine weitere Anwendung des Spezialisierungskonzeptes auf Ebene des Vorgehensmodells. Gute Modelle zeichnen sich durch Redundanzfreiheit aus. Das V-Modell XT weist in der Umsetzung der Systemerstellung eine Reihe von Redundanzen auf (vgl. Kapitel 3.4.4), da mit den Beschreibungsmitteln des V-Modells XT Spezialisierung nicht zur Verfügung steht. Zudem ist das Spezialisierungskonzept für die Modellierung organisations- oder domänenspezifischer Ausprägungen anwendbar.

Wir stellen anhand eines Beispiels aus dem V-Modell XT einen Lösungsansatz für die Vermeidung von Redundanzen vor. Zielsetzung ist kein Lösungskonzept zu den Schwachstellen des V-Modells XT, sondern die Demonstration der Notwendigkeit eines Spezialisierungskonzeptes auf Ebene des Vorgehensmodells.

Abbildung 5–24 zeigt eine Modellierung eines Ausschnitts der Disziplin Systemerstellung im V-Modell XT mit den Beschreibungsmitteln der vorliegenden Arbeit. Bei der in Abbildung 5–24 dargestellten Erzeugung handelt es sich um eine n-stellige Assoziation mit zwei Quellen und einem Ziel. Bei den Assoziationen Bezug handelt es sich allerdings jeweils um binäre Assoziationen.

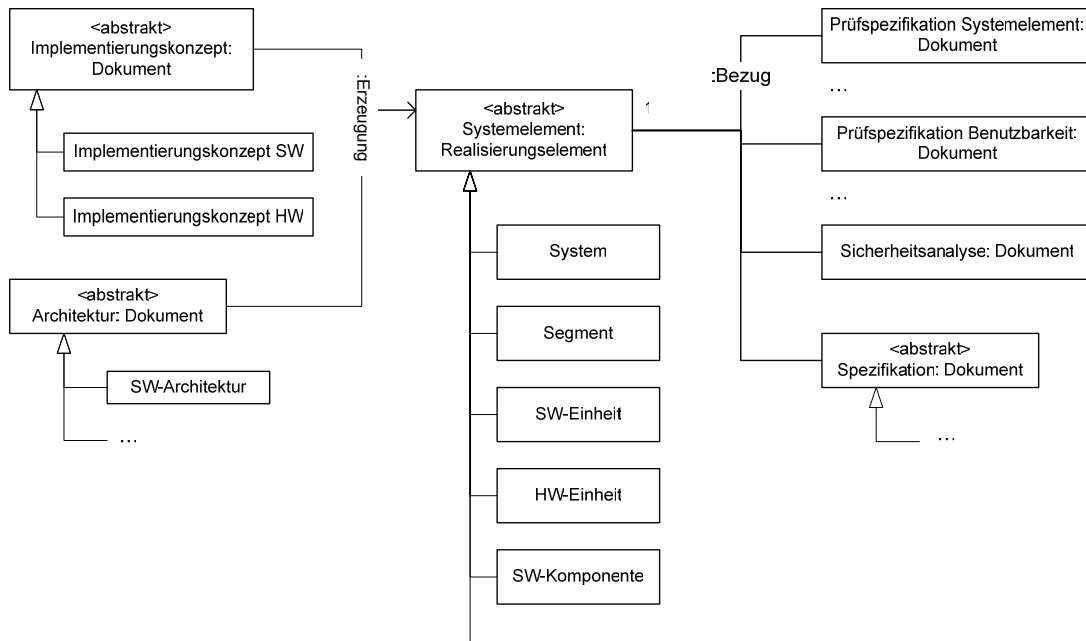


Abbildung 5–24: Spezifische Ausprägungen des V-Modells XT – Ebene Vorgehensmodell

Das V-Modell XT weist hinsichtlich der Beschreibungen zu SW und Hardware beispielsweise in Bezug auf die Systemelemente einige Redundanzen auf (vgl. Kapitel 3.4.4). Wir modellieren daher eine abstrakte Klasse *Systemelement*, die Gemeinsamkeiten der Systemelemente kapselt. Gemeinsam ist den Systemelementen beispielsweise das Schema der erzeugenden Produktabhängigkeiten. Wir modellieren die *Erzeugung* ausgehend von den abstrakten Klassen *Implementierungskonzept* und *Architektur*, die in Spezialisierungen wie „*Implementierungskonzept SW*“ existieren. Gemeinsam ist den Systemelementen auch der *Bezug* zu einer Reihe von Dokumenten wie einer „*Prüfspezifikationen Systemelement*“, einer *Sicherheitsanalyse* oder einem abstrakten Dokument *Spezifikation*.

Diese Modellierung bietet eine Reihe von Vorteilen. Im Rahmen des Erstellungsprozesses eines Vorgehensmodells (vgl. dazu Kapitel 6) können abstrakte Klassen als Schema für die Ausarbeitung der Details dienen. Die Vermeidung von Redundanzen in einer Modellierung dient also nicht zuletzt dem Ziel der einheitlichen Beschreibung eines Vorgehensmodells. Aus Sicht des Anwenders eines Vorgehensmodells ist eine Beschreibung des wesentlichen Prinzips in Form abstrakter Klassen, sowie spezifischer Ausprägungen dieses Prinzips, ebenfalls vorzuziehen.

Das Beispiel in Abbildung 5–24 zeigt zudem die Anwendung des Spezialisierungskonzeptes für die Modellierung organisations- oder domänenspezifischer Ausprägungen. Die Beschreibung der Entwicklung von Software oder Hardware könnte auch als domänenspezifische Ausprägung des allgemeinen V-Modells XT in einem eigenen Vorgehensmodell beschrieben werden.

5.4.3 Beziehungen zwischen Spezialisierungs-Ebenen: Generische Vorgehensweisen

In diesem Kapitel betrachten wir so genannte generische Vorgehensweisen. Generische Vorgehensweisen beziehen sich auf eine bestimmte Menge von Spezialisierungen von Konzepten des Vorgehens-Meta-Modells. Beispielsweise bezieht sich die Qualitätssicherung

als generische Vorgehensweise auf alle Produktexemplare eines Projektes. Die Menge aller Produktexemplare wird dabei projektspezifisch auf bestimmte qualitätszusichernde Produkte beziehungsweise Produktexemplare eingeschränkt.

Generische Vorgehensweisen erfordern eine Modellierung, bei der Elemente des Vorgehensmodells Elemente des Vorgehens-Meta-Modells referenzieren. Zielsetzung dieses Kapitels ist die beispielhafte Darstellung von Beziehungen zwischen Spezialisierungsebenen. Beziehungen zwischen Spezialisierungsebenen erfordern keine Erweiterung des Vorgehens-Meta-Modells.

Beispiel: Vorgehensmodell und Projektebene

Abbildung 5–25 zeigt ein Beispiel eines Vorgehensmodells mit Referenzierung des Vorgehens-Meta-Modells. Dabei handelt es sich um die Qualitätssicherung als generische Vorgehensweise.

Im Beispiel gehen wir von zwei unterschiedlichen Methoden für die Durchführung von Prüfungen als Maßnahmen der Qualitätssicherung aus. Wir unterscheiden schriftliche Reviews und Prüfungen im Rahmen einer Sitzung, deren Ergebnis in einem Sitzungsprotokoll festgehalten wird. Das Beispiel ist planungsrelevant, da für diese Prüfmethode Dauer und zugeordnete Personen variieren.

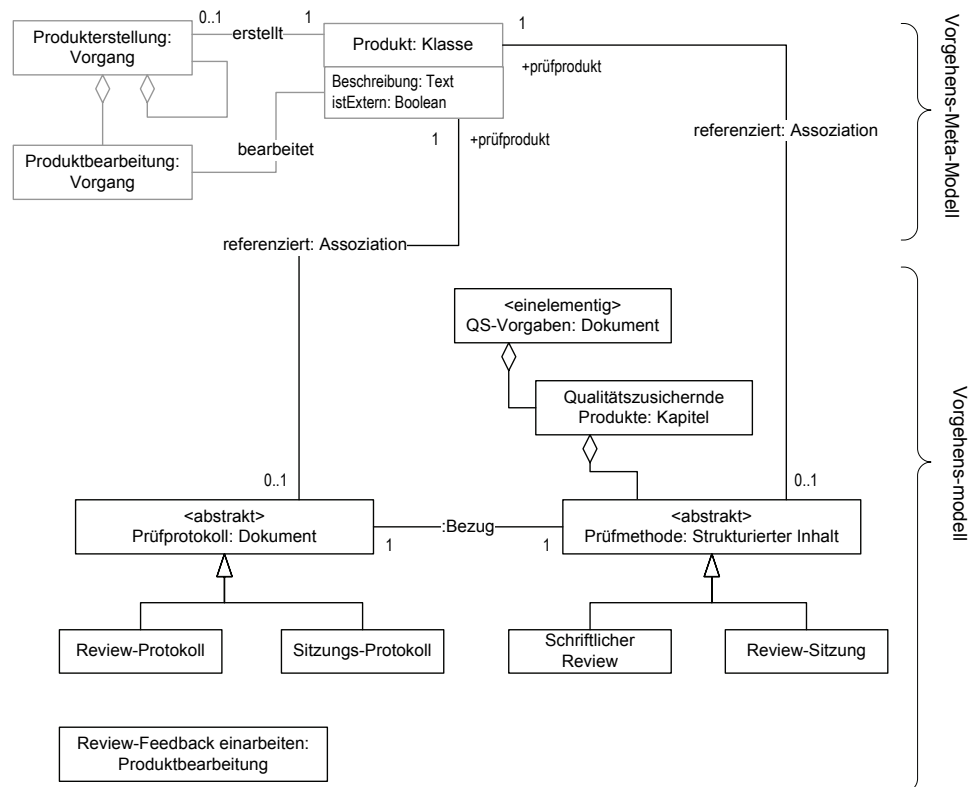


Abbildung 5–25: Beziehungen zwischen Spezialisierungsebenen – Ebene Vorgehensmodell mit Referenzierung des Vorgehens-Meta-Modells (Beispiel)

Prüfmethode und *Prüfprotokoll* dienen als Abstraktionen der konkreten Prüfmethode „*Schriftlicher Review*“ und *Review-Sitzung* sowie der konkreten Prüfprotokolle *Review-Protokoll* und *Sitzungs-Protokoll*.

(33)

Der Prüfmethode „*Schriftlicher Review*“ ist als Prüfprotokoll ein *Review-Protokoll* zugeordnet. Der *Review-Sitzung* ist ein *Sitzungs-Protokoll* zugeordnet:

$$\begin{aligned} \forall sr \in \text{Schriftlicher Review}: \exists rp \in \text{Review-Protokoll}: sr.\text{Prüfprotokoll} = rp \wedge \\ \forall rs \in \text{Review-Sitzung}: \exists sp \in \text{Sitzungs-Protokoll}: rs.\text{Prüfprotokoll} = sp \end{aligned}$$

Die jeweiligen Prüfmethode werden im Kapitel „*Qualitätszusichernde Produkte*“ des Dokumentes *QS-Vorgaben* auf Projektebene vereinbart. Jede in den *QS-Vorgaben* vereinbarte Prüfmethode *referenziert* genau ein *Produkt* als *Prüfprodukt*. Das jeweilige Prüfprotokoll muss entsprechend dasselbe Produkt *referenzieren*. Wir gehen im Beispiel vereinfachend davon aus, dass Produktexemplare jeweils nur einer Prüfung unterzogen werden.

(34)

Jede Prüfmethode bezieht sich auf ein Prüfprotokoll, welches einem Prüfprodukt zugeordnet ist, das wiederum der Prüfmethode selbst zugeordnet ist:

$$\forall pm \in \text{Prüfmethode}: pm.\text{Prüfprotokoll}.\text{Prüfprodukt}.\text{Prüfmethode} = pm$$

Die in Constraint (34) als Prüfprodukte referenzierten Produkte sind dabei Teil des Vorgehens-Meta-Modells.

Der Prüfung von Produktexemplaren folgt nach der Erstellung eines Prüfprotokolls die Berücksichtigung des Feedbacks der Prüfung im Rahmen der Erstellung des jeweiligen Produktexemplars. Die Berücksichtigung des Feedbacks modellieren wir als Produktbearbeitung „*Review-Feedback einarbeiten*“, die der jeweiligen Produkterstellung untergeordnet ist. Die Einarbeitung des Feedbacks ist durch Ablauf-Beziehungen mit der Produkterstellung des Prüfproduktes und der Produkterstellung des Prüfprotokolls verbunden.

(35)

- Für jede Prüfmethode gibt es eine Produktbearbeitung „Review-Feedback einarbeiten“, die Teil der Produkterstellung des Prüfproduktes ist.
- Die Produkterstellung des Prüfproduktes ist über eine Ende-Ende-Vorgangsfolge mit dieser Produktbearbeitung verbunden.
- Die Produkterstellung des Prüfprotokolls ist über eine Ende-Anfang-Vorgangsfolge mit dieser Produktbearbeitung verbunden:

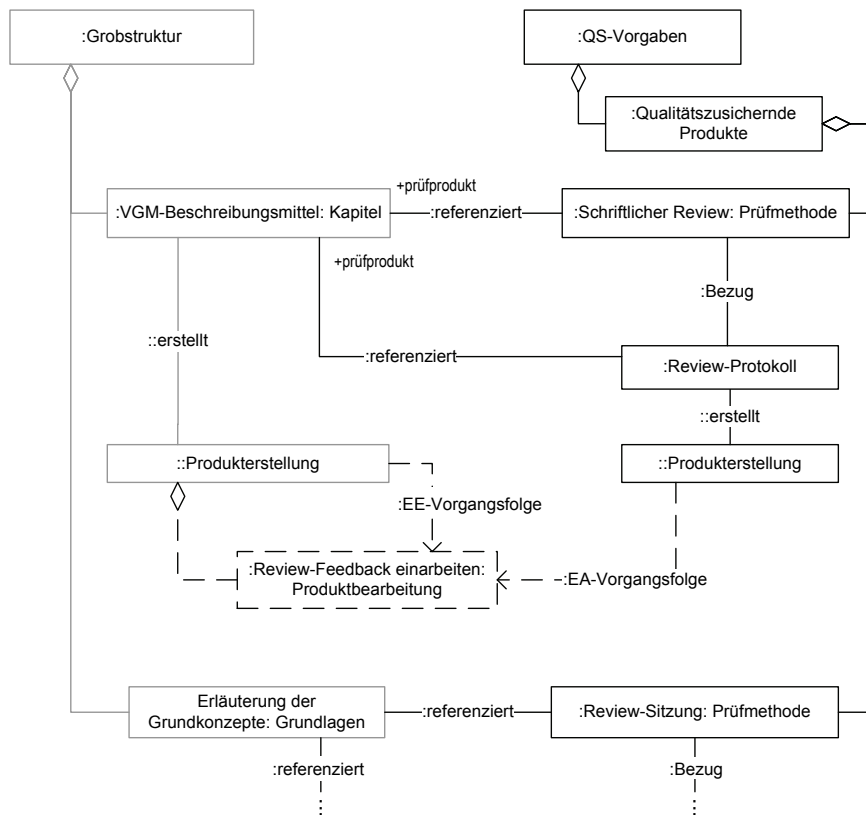
$$\begin{aligned}
&\forall pm \in \text{Prüfmethode}: \exists b \in \text{Produktbearbeitung}: \\
&\quad b.name = \text{„Review-Feedback einarbeiten“} \wedge \\
&\quad b \in pm.prüfprodukt.Produkterstellung.enthält \text{Produktbearbeitung} \\
&\quad \wedge \\
&\quad \exists vf_1 \in \text{EE-Vorgangsfolge}: \\
&\quad \quad vf_1.von = b.Produkterstellung \wedge vf_1.nach = b \wedge \\
&\quad \exists vf_2 \in \text{EA-Vorgangsfolge}: \\
&\quad \quad vf_2.von = pm.Prüfprotokoll.Produkterstellung \wedge vf_2.nach = b
\end{aligned}$$

Eine Spezialisierung der Qualitätssicherung auf Projektebene stellt sich wie in Abbildung 5–26 am Beispiel gezeigt dar.

Wir betrachten als Beispiel eine Prüfung des Kapitels *VGM-Beschreibungsmittel* des Dokuments *Grobstruktur*. Das Kapitel wird im Kapitel „Qualitätszusichernde Produkte“ der *QS-Vorgaben* als nach der *Prüfmethode* „Schriftlicher Review“ zu prüfen vereinbart.

Dies impliziert entsprechend den Vorgaben des Vorgehensmodells in Constraint (35) die Spezialisierung eines *Review-Protokolls* als Form eines Prüfprotokolls mit zugehöriger Produkterstellung, eine Produktbearbeitung zur Einarbeitung des Feedbacks, sowie entsprechende Vorgangsfolgen. Diese Vorgänge und Vorgangsfolgen sind in Abbildung 5–26 (b) ergänzend als Balkenplan dargestellt.

(a) als Klassendiagramm



(b) als Balkendiagramm

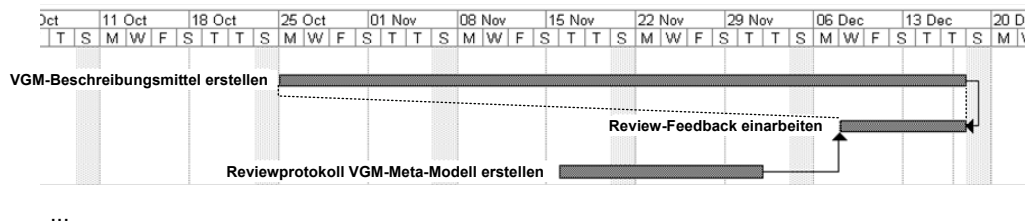


Abbildung 5–26: Beziehungen zwischen Spezialisierungs-Ebenen – Projektebene (Beispiel)

Die Assoziation *referenziert*, die in Abbildung 5–25 (a) die Ebene des Vorgehensmodells mit dem Vorgehens-Meta-Modell verbindet, kann gemäß den Eigenschaften des Spezialisierungskonzeptes der Modellierungstechnik auf Spezialisierungen des Vorgehens-Meta-Modells zeigen. In Abbildung 5–26 (a) verbindet die Assoziation *referenziert* die VGM-Beschreibungsmittel als Spezialisierung eines Produktes auf der Projektebene mit einer Prüfmethode „Schriftlicher Review“ beziehungsweise mit einem Prüfprotokoll Review-Protokoll auf Projektebene.

Zusammenfassung

Generische Vorgehensweisen auf Ebene des Vorgehensmodells referenzieren Elemente des Vorgehens-Meta-Modells. Die Qualitätssicherung als generische Vorgehensweise stellt beispielsweise einen Bezug zu allen Produktexemplaren her. Generische Vorgehensweisen und damit Assoziationen zwischen Spezialisierungs-Ebenen stellen eine Besonderheit dar, die in Vorgehensmodellen öfters auftritt. Weitere Beispiele für generische Vorgehensweisen sind das Konfigurationsmanagement aller Produkte oder der Einkauf aller als extern markierten Produkte. In diesem Kapitel haben wir am Beispiel gezeigt, wie generische Vorgehensweisen mit Hilfe von Assoziationen zwischen Meta-Ebenen und Constraints präzise modelliert werden können.

5.5 Zusammenfassung

In diesem Kapitel fassen wir die wesentlichen Konzepte des Vorgehens-Meta-Modells überblicksartig zusammen. Eine generelle Positionierung dieser Konzepte in Bezug auf das V-Modell XT geben wir als Ausblick in Kapitel 7 an. Wir bedienen uns in diesem Kapitel teilweise unmittelbar der Texte der Zusammenfassungen zu den einzelnen Unterkapiteln des Kapitels 5.

Das Konzept des *Produktes* modellieren wir im Vorgehens-Meta-Modell in den Spezialisierungen *Dokument*, *Kapitel*, *Realisierungselement* und *Werkzeug*. Produktinhalte können in Form des Konzeptes der „*strukturierten Inhalte*“ beschrieben werden. Produktinhalte können als Parameter die Ableitung eines Projektplans beeinflussen. *Vorgänge* modellieren wir in den Spezialisierungen *Produkterstellung*, *Produktbearbeitung* und *Projektabschnitt*. Genau ein Vorgang zur *Produkterstellung* ist jeweils den Produkten als zentralen Projektergebnissen zugeordnet. Die Projektplanung orientiert sich also stets an den zu erbringenden Ergebnissen. Produkte sind dabei von unterschiedlicher „Wichtigkeit“. *Hilfsprodukte* dienen der Erstellung der eigentlichen Projektergebnisse. Die Erstellung von Hilfsprodukten wird im Projektplan unter den jeweiligen Produkterstellungen der eigentlichen Projektergebnisse eingeordnet.

Produktbearbeitungen dienen der zeitlichen Planung der Zusammenarbeit mehrerer Bearbeiter bei der Erstellung desselben Produktes. Ein Vorgang zur Produktbearbeitung kann allerdings auch mehrere Produkte zugleich bearbeiten. Vorgänge zur Produkterstellung und Produktbearbeitung können in *Projektabschnitte* eingeordnet werden. Projektabschnitte können im Vorgehensmodell beziehungsweise Projekt sehr unterschiedlich strukturiert werden. Am Beispiel haben wir eine Strukturierung in Phasen, Releases und Arbeitspaketen gezeigt. Die unterschiedlichen Arten von Vorgängen bilden dabei anders als in vielen Planungswerkzeugen realisiert keine Hierarchie, da sich beispielsweise Produktbearbeitungen in mehrere Produkterstellungen und zugleich in Projektabschnitte einordnen können sollten.

Abläufe modellieren wir in Form von Vorgangsfolgen der Netzplantechnik. Vorgangsfolgen wie „Ende-Ende“ erlauben die Festlegung möglicher zeitlicher Überlappungen von Vorgängen. *Iterationen* kennzeichnen Planungsentscheidungen hinsichtlich iterativer Abläufe explizit. Als Parameter sind Iterationen bei der Ableitung eines Projektplans zu belegen. Anders als im Falle von Produktabhängigkeiten leiten sich Planungselemente, die im Kontext von Iterationen auftreten, nicht aus den Inhalten anderer Produktexemplare ab, sondern sind Bestandteil des Projektplans selbst. Iterationen treten sowohl auf der Ebene der Projektstrukturierung durch Projektabschnitte als auch auf der Ebene der Produktbearbeitung auf.

Produktabhängigkeiten der Kategorie *Erzeugung* dienen als Hinweise für die Erstellung des Projektplans. Produktexemplare spezifizieren dabei die Existenz von Produktexemplaren eines anderen Produktes. Erzeugung erfordert das Lesen und Interpretieren der Inhalte der jeweiligen Produktexemplare durch den Projektleiter. Da es in der Praxis nicht lohnenswert ist, jeden Produktinhalt beziehungsweise Zusammenhang mit anderen Produkten als Vorgabe für einen Projektplan präzise zu modellieren und damit einer Werkzeugunterstützung zugänglich zu machen, ist ein Konzept wie die Erzeugung unabdingbar. Mit einer Erzeugung sind keine Konsistenzsicherung der Produktexemplare und keine Reihenfolge der Produkterstellung verbunden.

Produktabhängigkeiten der Kategorie *Konsistenzabhängigkeit* und *Produktfluss* sehen wir als typische Muster für die Planung der Zusammenarbeit mehrerer Bearbeiter bei der Erstellung mehrerer in Beziehung stehender Produkte an. Anders als im V-Modell XT stellen Produktabhängigkeiten hier keine Anforderungen an Konfigurationsmanagement beziehungsweise Zustände bestimmter Produktexemplare. Das Vorgehens-Meta-Modell haben wir in Bezug auf Produktabhängigkeiten bewusst nicht mit dem Anspruch auf Vollständigkeit modelliert, sondern lediglich als Hilfsmittel für typische Planungen. Andere Arten von Abläufen

und Vorgängen zur Konsistenzsicherung können allerdings durch unmittelbare Spezialisierung der Konzepte Ablauf oder Produktbearbeitung im Vorgehensmodell modelliert werden.

Eine *Konsistenzabhängigkeit* zweier Produkte impliziert eine Produktbearbeitung zur Prüfung beziehungsweise Herstellung der Konsistenz der entsprechenden Produktexemplare. Diese Produktbearbeitung ist den jeweiligen Produkterstellungen gemeinsam, das heißt sie wird von beiden Produkterstellungen aggregiert. Die Konsistenzsicherung erfordert also eine Zusammenarbeit der Bearbeiter dieser Produktexemplare. Die Produkterstellungen über Konsistenzabhängigkeiten verbundener Produkte überlappen sich zeitlich mindestens für die Dauer dieser gemeinsamen Konsistenzsicherung.

Produktflüsse kennzeichnen ein Produktexemplar als Vorgabe für ein anderes Exemplar eines Produktes. Die Herstellung der Konsistenz beider Produktexemplare obliegt allein dem Bearbeiter des Zielproduktexemplars. Produktflüsse implizieren also eine Produktbearbeitung zur Konsistenzsicherung, die von der Produkterstellung des Zielproduktes aggregiert wird. Ein Produktfluss hat zudem Auswirkungen auf die Reihenfolge der Produkterstellungen. Diese Auswirkungen können im Rahmen des Tailorings beispielsweise hinsichtlich Ende-Ende- oder Ende-Anfang-Vorgangfolgen ausgeprägt werden (vgl. dazu Kapitel 6.2).

Verglichen mit dem Konzept des V-Modells XT sind Produktabhängigkeiten hier als konkreter einzustufen, da wir mit Produktabhängigkeiten Reihenfolgen beziehungsweise zeitliche Überlappungen der Produkterstellung verbinden. Produktabhängigkeiten können also der Beschreibung konkreter beispielsweise organisationspezifischer Vorgehensmodelle dienen. Allerdings können die hier konzipierten Produktabhängigkeiten auch für ein im Abstraktionsgrad dem V-Modell XT vergleichbares Vorgehensmodell Anwendung finden. Beispielsweise erscheint es sinnvoll, zwischen Produkten wie der „Gefährdungs- und Systemsicherheitsanalyse“ und der „Systemarchitektur“ des V-Modells XT eine Konsistenzabhängigkeit zu modellieren, da sich die Erstellung beider Produkte im Projekt aufgrund starker gegenseitiger Abhängigkeiten überlappen sollte.

Produktabhängigkeiten der Kategorie *Bezug* dienen der Modellierung feingranularer Zusammenhänge des Produktmodells, die nicht durch die übrigen Arten von Produktabhängigkeiten modelliert werden können. Bezüge und Multiplizitäten dienen dem Prozessingenieur als Sprachmittel zur Einschränkung der projektspezifischen Produktstruktur.

Das Vorgehens-Meta-Modell dient dem Prozessingenieur als Sprache für die Beschreibung des Vorgehensmodells. Das Vorgehens-Meta-Modell zeichnet sich durch Einfachheit aus, beispielsweise durch die Zuordnung genau eines Vorgangs zur Produkterstellung zu jedem modellierten Produkt. Daneben können aber auch unmittelbar alle sprachlichen Mittel eingesetzt werden, die sich aus der definierten Modellierungstechnik ergeben. Dem Prozessingenieur stehen bei der Beschreibung des Vorgehensmodells beispielsweise Multiplizitäten zur Einschränkung der projektspezifischen Produktstruktur zur Verfügung. Auf Ebene des Vorgehensmodells können auch Constraints als Einschränkungen der Projektebene formuliert werden. Die Beschreibung von Vorgehensmodellen wird also nicht durch das Vorgehens-Meta-Modell eingeschränkt, sondern kann sich der Ausdrucksmächtigkeit der Modellierungstechnik basierend auf Klassen und prädikatenlogischen Constraints bedienen. Die Beschreibung komplexer Zusammenhänge als Vorgabe für die Planung ist möglich, wie wir anhand von Beispielen gezeigt haben, wenn auch im Allgemeinen keine triviale Aufgabe.

Die Projektebene enthält Spezialisierungen der Konzepte des Vorgehensmodells. Bei der Erstellung einer Projektplanung können allerdings auch Konzepte des Vorgehens-Meta-Modells unmittelbar spezialisiert werden. Beispielsweise kann die Projektebene Produktbearbeitungen als unmittelbare Spezialisierungen des Vorgehens-Meta-Modells beinhalten, die im Vorgehensmodell nicht vorgesehen sind oder auch bestimmte Produkte projektspezifisch als Hilfsprodukte kennzeichnen. Diese Flexibilität im Rahmen der Planung ist wünschenswert, da Vorgehensmodelle als Abstraktionen der Projektebene hinsichtlich aller einzuplanenden Vorgänge in der Praxis nicht vollständig sein werden.

Der verwendete Modellierungsansatz der Arbeit sieht die Aufteilung in die Ebenen Vorgehens-Meta-Modell, Vorgehensmodell und Projekt vor, die sich durch das Konzept der Spezi-

alisierung abgrenzen. Ein Spezialisierungskonzept ist allerdings auch auf der Ebene des Vorgehensmodells erforderlich. In einem Entwicklungsprojekt stellt Modellbildung eine gängige Technik dar. Modellbildung, das heißt Generalisierung muss also auch im Vorgehensmodell beschreibbar sein, da Modellbildung insbesondere einen Einflussfaktor auf die Projektplanung darstellt. Als Beispiel haben wir die Konzeption eines Meta-Modells für die Beschreibung des Vorgehensmodells VGM betrachtet. Dieses auf Projektebene konzipierte Meta-Modell impliziert bestimmte Abläufe für die weitere Planung des VGM-Projektes. Nach Fertigstellung kann dieses Meta-Modell aus Sicht des VGM-Projektes also als Teil des Vorgehensmodells verstanden werden.

Planung verstehen wir als gleichbedeutend mit dem Konzept der Spezialisierung. Die Projektebene und damit den Projektplan definieren wir als die aktuell „tiefste“ Spezialisierungsebene des Vorgehens-Meta-Modells. Als festgelegt betrachten wir die Ebene des Vorgehens-Meta-Modells. Diese Ebene kann mit dem Konzept der Spezialisierung in einen Projektplan überführt werden. Die Anzahl der erforderlichen Spezialisierungen bis hin zum Projektplan ist dabei nicht fix. Dies entspricht den Gegebenheiten der Praxis, da Projektpläne im Projektverlauf unterschiedliche Spezialisierungsgrade durchlaufen. Treibt man diese Idee an ihre Grenzen, kann auch das Vorgehens-Meta-Modell selbst, solange keine Spezialisierungen existieren, als Projektplan verstanden werden. Das Vorgehens-Meta-Modell als Projektplan enthält dabei die Aussage, dass in der Projektlaufzeit Projektergebnisse in Form von Produkten durch Vorgänge erstellt werden müssen. Die Erstellung von Ergebnissen in einem zeitlich begrenzten Rahmen entspricht der Definition des Begriffs Projekt. Das Vorgehens-Meta-Modell als Projektplan ist also in jedem Fall korrekt. In der Praxis muss das Vorgehens-Meta-Modell als Projektplan aufgrund seines Detaillierungs-Grades natürlich sogar zu Projektbeginn als ungenügend bezeichnet werden. Theoretisch ist der Grad der Detaillierung eines Projektplans allerdings relativ und nicht absolut definierbar.

Eine bestimmte Art der Modellierung ist für so genannte generische Vorgehensweisen wie die Qualitätssicherung erforderlich. Generische Vorgehensweisen auf Ebene des Vorgehensmodells referenzieren Elemente des Vorgehens-Meta-Modells. Die Qualitätssicherung als generische Vorgehensweise stellt beispielsweise einen Bezug zu allen Produktexemplaren her. Generische Vorgehensweisen und damit Assoziationen zwischen Spezialisierungsebenen treten in Vorgehensmodellen häufiger auf. Mit Hilfe von Assoziationen zwischen Meta-Ebenen und Constraints lassen sich generische Vorgehensweisen präzise modellieren. Dies haben wir am Beispiel der Qualitätssicherung gezeigt.

6 Erstellung und Anwendung von Vorgehensmodellen

In diesem Kapitel betrachten wir die Anwendung der im vorangehenden Kapitel erarbeiteten Konzepte zur Modellierung von Vorgehensmodellen. Unter Erstellung verstehen wir die Anwendung der konzipierten Beschreibungsmittel im Erstellungsprozess eines Vorgehensmodells. Anwendung bedeutet die Anwendung des Vorgehensmodells in einem Projekt, also im Kontext der Arbeit die Ableitung eines Projektplans.

In Kapitel 6.1 entwerfen wir zunächst ein Modell zur Strukturierung der Inhalte eines Vorgehensmodells. Neben einer übersichtlichen Aufteilung des Vorgehensmodells bildet diese Strukturierung der Inhalte die Grundlage für das Tailoring. In Kapitel 6.2 stellen wir ein Modell zur Festlegung von Tailoring-Einheiten vor, also für die projektspezifische Anpassung eines Vorgehensmodells. Das Tailoring-Modell erproben wir in Kapitel 6.3 anhand von Beispielen aus dem V-Modell XT.

Die Anwendung eines Vorgehensmodells im Sinne des Tailorings und der Ableitung eines Projektplans definieren wir in Kapitel 6.4. Schließlich liefert uns die inhaltliche Strukturierung des Vorgehensmodells die Basis für die Definition von Bearbeitungs-Einheiten im Erstellungsprozess eines Vorgehensmodells. Ein methodisches Vorgehen für diesen Erstellungsprozess geben wir in Kapitel 6.5 an. In Kapitel 6.6 zeigen wir die „Modellierung von Texten“ zur Beschreibung der Elemente eines Vorgehensmodells im Zusammenhang mit dem Aspekt des Tailorings. In Kapitel 6.7 stellen wir schließlich ein Werkzeugkonzept für die Erstellung und Anwendung von Vorgehensmodellen vor, das die in der Arbeit definierten Konzepte zur Anwendung bringt.

Inhalt des Kapitels	Seite
6.1 Modell zur inhaltlichen Strukturierung	157
6.2 Tailoring-Modell	169
6.3 Studie: Tailoring anhand von Beispielen des V-Modells XT	174
6.4 Anwendung von Vorgehensmodellen: Tailoring und Ableitung eines Projektplans	177
6.5 Methodisches Vorgehen der Erstellung von Vorgehensmodellen	184
6.6 Modellierung von Texten	191
6.7 Werkzeugkonzept	194

6.1 Modell zur inhaltlichen Strukturierung

Mit der Strukturierung der Inhalte eines Vorgehensmodells verfolgen wir mehrere Ziele. Aus Sicht des Lesers eines Vorgehensmodells ist die Einteilung von Produkten und Planungselementen in inhaltliche Kategorien erforderlich, um Übersichtlichkeit zu gewährleisten. Diesen Aspekt stellen wir in diesem Kapitel in den Vordergrund. Zudem bietet die inhaltliche Strukturierung den Ausgangspunkt für die Definition sowohl von Tailoring-Einheiten (vgl. dazu Kapitel 6.2) als auch von Bearbeitungs-Einheiten (vgl. dazu Kapitel 6.5).

Zielsetzung dieses Kapitels ist die Konzeption von Beschreibungsmitteln auf Ebene des Vorgehens-Meta-Modells für die Strukturierung der Inhalte eines Vorgehensmodells. Wir erläutern das Modell zur inhaltlichen Strukturierung anhand von Beispielen auf der Ebene des Vorgehensmodells.

Abbildung 6–1 zeigt das Modell zur inhaltlichen Strukturierung auf Ebene des Vorgehens-Meta-Modells. Strukturiert werden dabei die Inhalte des Produktmodells und des Planungsmodells, die in den vorangehenden Kapiteln entworfen wurden. Die abstrakte Klasse *Vorgehensmodell-Inhalt* modellieren wir in den Spezialisierungen *Strukturierung* und *Strukturierungs-Element*. Strukturierungen aggregieren weitere Vorgehensmodell-Inhalte. Strukturierungs-Elemente bilden die unterste Ebene der Hierarchie. Strukturierungs-Elemente selbst sind abstrakt und treten in Form der Spezialisierungen *Disziplin* beziehungsweise *Planungs-Einheit* auf. Die inhaltliche Strukturierung stellt eine hierarchische Strukturierung dar, das heißt jedes Strukturierungs-Element beziehungsweise jede Strukturierung wird eindeutig von höchstens einer anderen Strukturierung aggregiert.

Disziplinen aggregieren *Produkte*. Produkte sind genau einer Disziplin zugeordnet. Planungs-Einheiten aggregieren *Planungselemente* und zudem *Constraints*. Constraints sind Elemente der verwendeten Modellierungstechnik (vgl. Kapitel 4). Planungs-Einheiten enthalten dabei allein Constraints, die dem Vorgehensmodell zuzurechnen sind, also die Projektebene einschränken.

(36)

Planungselemente sind abgesehen von Produkterstellungen genau einer Planungs-Einheit zugeordnet:

$$\forall pe \in \text{Planungselement}: pe \neq \text{Produkterstellung} \Rightarrow | p.\text{Planungs-Einheit} | = 1$$

Die Zuordnung von Produkterstellungen zu Planungs-Einheiten wäre nicht sinnvoll, da sich Produkterstellungen aus der Existenz von Produkten ableiten (vgl. Kapitel 5.2.3, Constraint (17)) und im Vergleich zu Produkten keine zusätzliche Information beinhalten. Eine inhaltliche Strukturierung mit Produkten und Produkterstellungen in unterschiedlichen Strukturierungs-Elementen ist daher nicht wünschenswert. Die Aufteilung von Produkten und Projektabschnitten beziehungsweise von Produkten und Produktbearbeitungen in unterschiedliche Strukturierungs-Elemente ist dagegen sinnvoll, da sowohl Projektabschnitte als auch Produktbearbeitungen oft unabhängig von Produkten sind. Wie die Beispiele des folgenden Kapitels zeigen sind Produktbearbeitungen oft nicht spezifisch auf ein Produkt ausgelegt.

Den Sinn der Zuordnung von Elementen des Vorgehensmodells zu genau einem Strukturierungs-Element verdeutlichen wir im Zusammenhang mit dem Tailoring beziehungsweise dem Konzept für Bearbeitungs-Einheiten (vgl. dazu Kapitel 6.2 und 6.5).

Disziplinen und Produkte können zusätzlich zur Aggregation von Produkten beliebige weitere Produkte *referenzieren*. Für den Leser sind im Hinblick auf eine Disziplin beziehungsweise Planungs-Einheit alle Elemente des Vorgehensmodells sichtbar, die von dieser aggregiert oder referenziert werden.

Vorgehensmodell-Inhalte, also sowohl Strukturierungen als auch Disziplinen und Planungs-Einheiten, können sich inhaltlich überlappen. Dies kennzeichnen wir durch die Assoziation „*inhaltliche Überlappung*“. Inhaltliche Überlappungen kennzeichnen Vorgehensmodell-Inhalte, die eng aneinander gekoppelt sind und dienen damit als Hinweis für den Leser auf verwandte Themen. Planungs-Einheiten können als *querschnittlich* gekennzeichnet werden, wenn sie potentiell mit Produkten aus allen Disziplinen in Beziehung stehen. Eine Kennzeichnung konkreter inhaltlicher Überlappungen entfällt in diesem Fall. Im Kontext der Methodik der Erstellung eines Vorgehensmodells wirken sich „inhaltliche Überlappungen“ und querschnittliche Planungs-Einheiten zudem auf die Planung eines Erstellungsprojektes für ein Vorgehensmodell aus (vgl. dazu Kapitel 6.5).

Referenzierung gibt den genauen Bezug eines Strukturierungs-Elementes zu einem Produkt einer inhaltlich überlappenden Disziplin an. Während die „inhaltliche Überlappung“ zwei Strukturierungs-Elemente in Beziehung setzt, zeigt Referenzierung den Bezug zweier Strukturierungs-Elemente im Detail auf Ebene einzelner Elemente des Vorgehensmodells. Für

den Zusammenhang zwischen Referenzierung und „inhaltlicher Überlappung“ formulieren wir im Folgenden eine Reihe von Constraints.

(37)

Strukturierungen überlappen sich, wenn sich die enthaltenen Disziplinen, Planungseinheiten beziehungsweise Strukturierungen überlappen. Dies drücken wir über die abstrakte Klasse Vorgehensmodell-Inhalt aus:

$$i_1, i_2, i_3, i_4 \in \text{Vorgehensmodell-Inhalt:}$$

$$i_1 \in i_2.\text{enthält}^{\text{teil}} \wedge i_3 \in i_4.\text{enthält}^{\text{teil}} \wedge i_1 \in i_3.\text{inhaltliche Überlappung} \Rightarrow$$

$$i_2 \in i_4.\text{inhaltliche Überlappung}$$

(38)

Disziplinen überlappen sich genau dann, wenn ein Produkt in einer Disziplin enthalten ist und in einer anderen Disziplin referenziert wird:

$$p \in \text{Produkt}, d_1, d_2 \in \text{Disziplin:}$$

$$p \in d_1.\text{enthält} \wedge p \in d_2.\text{referenziert} \Leftrightarrow d_1 \in d_2.\text{inhaltliche Überlappung}$$

(39)

Planungs-Einheiten und Disziplinen überlappen sich genau dann, wenn eine Planungseinheit ein Produkt einer Disziplin referenziert:

$$p \in \text{Produkt}, d \in \text{Disziplin}, pe \in \text{Planungs-Einheit:}$$

$$p \in pe.\text{referenziert} \wedge p \in d.\text{enthält} \Leftrightarrow d \in pe.\text{inhaltliche Überlappung}$$

Eine inhaltliche Überlappung von Planungs-Einheiten erlauben wir in der Modellierung mangels eines passenden Beispiels auf Ebene des Vorgehensmodells nicht.

(40)

Disziplinen referenzieren ein Produkt einer anderen Disziplin in jedem Fall dann, wenn dieses über eine Aggregations-Beziehung als Teil eines Produktes der Disziplin gekennzeichnet ist:

$$p_1, p_2 \in \text{Produkt}, d \in \text{Disziplin:}$$

$$p_2 \in d.\text{enthält} \wedge p_1 \in p_2.\text{enthält}^{\text{teil}} \wedge p_1.\text{Disziplin} \neq d \Rightarrow p_1 \in d.\text{referenziert}$$

Referenzierung tritt allerdings auch unabhängig von Aggregation zwischen Produkten auf. Weitere Fälle einer Referenzierung leiten wir allerdings nicht über Constraints aus Beziehungen des Vorgehens-Meta-Modells ab, da sie für eine umfassende Formalisierung nicht geeignet erscheinen. Beispiele dazu erläutern wir im folgenden Kapitel.

Assoziationen ordnen wir nicht explizit Strukturierungs-Elementen zu. Assoziationen treten im Vorgehens-Meta-Modell in Form von *Produktabhängigkeiten*, *Abläufen* oder auch Aggregationen entweder zwischen Produkten oder zwischen Planungselementen auf. Für den Betrachter einer Disziplin beziehungsweise einer Planungseinheit sind die Assoziationen des Vorgehensmodells sichtbar, die mit einem Element des Vorgehensmodells verbunden sind, das Bestandteil der jeweiligen Disziplin beziehungsweise Planungseinheit ist.

Beispiel: Vorgehensmodell

In Abbildung 6–2 geben wir ein Beispiel für eine inhaltliche Strukturierung auf Ebene des Vorgehensmodells an. Das Beispiel entspricht dem durchgängigen Beispiel der vorangehenden Kapitel zur Erstellung des Vorgehensmodells VGM. Anders als in den vorangehenden Kapiteln gestalten wir das Beispiel hier nicht als minimales Beispiel, sondern sammeln vielmehr alle Beispiele der vorangehenden Kapitel auf und strukturieren diese.

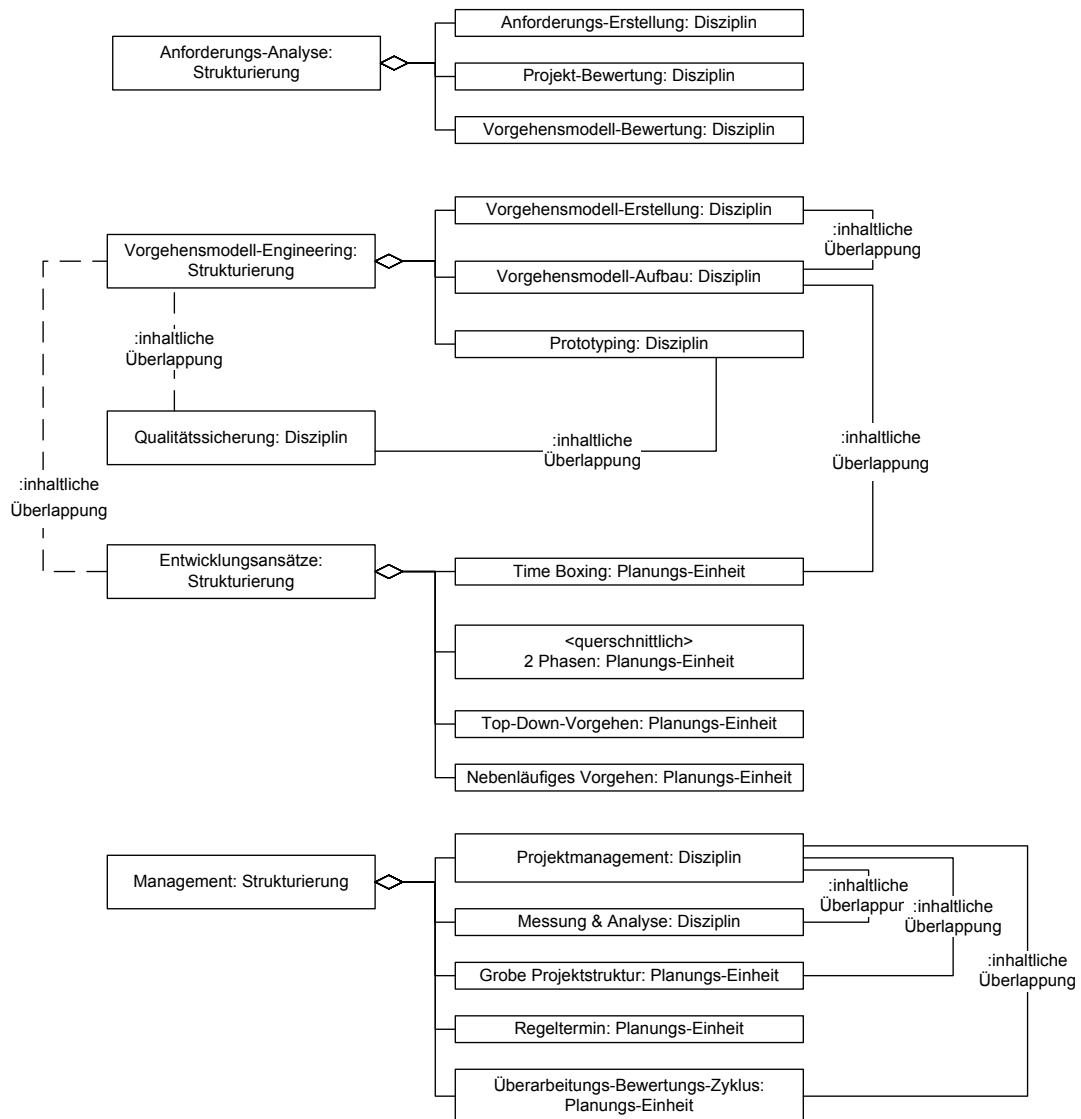


Abbildung 6–2: Modell zur inhaltlichen Strukturierung: Strukturierungen und Strukturierungselemente – Ebene Vorgehensmodell (Beispiel)

Wir gliedern das Vorgehensmodell in die Strukturierungen *Anforderungs-Analyse*, *Vorgehensmodell-Engineering*, *Qualitätssicherung*, *Entwicklungsansätze* und *Management*. Mit Ausnahme der *Qualitätssicherung* sind diese Strukturierungen weiter untergliedert. Die *Anforderungs-Analyse* besteht aus den Disziplinen *Anforderungs-Erstellung*, *Projekt-Bewertung* und *Vorgehensmodell-Bewertung*. *Projekt-Bewertung* und *Vorgehensmodell-*

Bewertung sind der Strukturierung Anforderungs-Analyse zugeordnet, da beide vor dem Hintergrund eines Erstellungsprojektes für ein Vorgehensmodell dem Zweck dienen Anforderungen zu sammeln.

In Abbildung 6–2 sind eine Reihe „*inhaltlicher Überlappungen*“ dargestellt. Die „inhaltlichen Überlappungen“, die als berechnete Assoziationen dargestellt sind, ergeben sich aus den Constraints des vorangehenden Abschnitts. Die Details dieser „inhaltlichen Überlappungen“ erläutern wir im Folgenden. Zunächst gehen wir dabei auf Disziplinen ein.

Disziplinen

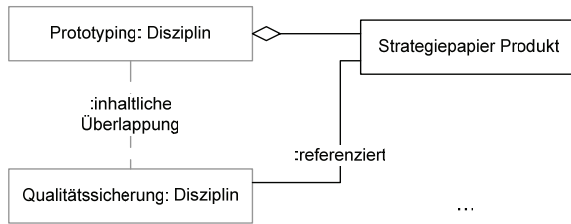
Abbildung 6–3 (a) zeigt eine inhaltliche Überlappung, die sich aus einem gemeinsamen Produkt ergibt, das von zwei Disziplinen aus sichtbar ist. Das *Strategiepapier* dient der prototypischen Umsetzung des VGMs in der Sprache der Grobstruktur (vgl. Kapitel 5.4.1) und ist damit Bestandteil der Disziplin *Prototyping*. Das Strategiepapier wird zudem von der Disziplin Qualitätssicherung referenziert, da eine prototypische Umsetzung des VGMs als qualitätssichernde Maßnahme einzustufen ist. Die Entscheidung, das Strategiepapier als Teil des Prototyping und als Referenz der Qualitätssicherung zu modellieren, ist aus Sicht einer inhaltlich getriebenen Strukturierung willkürlich. Wesentlich ist dieser Unterschied allerdings im Kontext von Tailoring-Einheiten beziehungsweise Bearbeitungs-Einheiten im Erstellungsprozess eines Vorgehensmodells, auf die wir in den folgenden Kapiteln eingehen (vgl. dazu Kapitel 6.2 und 6.5).

Abbildung 6–3 (b) zeigt eine inhaltliche Überlappung, die sich aus den Aggregations-Beziehungen der enthaltenen Produkte ableitet. Die Disziplin *Vorgehensmodell-Erstellung* enthält neben anderen Produkten das VGM als zu erstellendes Dokument. Die Disziplin *Vorgehensmodell-Aufbau* beinhaltet den *Vorgehensbaustein* als „strukturierten Inhalt“ des Dokumentes VGM, den das VGM aggregiert. Aus dieser Aggregations-Beziehung leitet sich gemäß Constraint (40) die dargestellte „inhaltliche Überlappung“ ab.

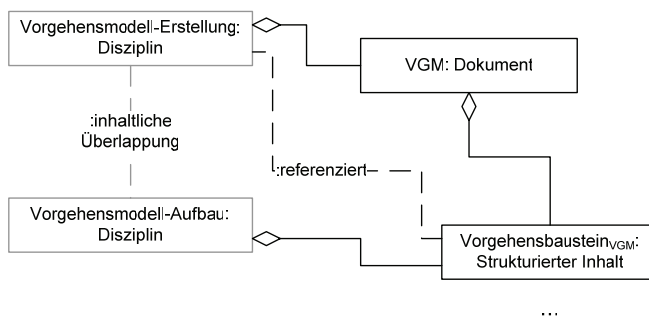
Produktabhängigkeiten haben im Gegensatz zu Aggregationen zwischen Produkten keinen unmittelbaren Einfluss auf Referenzierung und inhaltliche Überlappung. Abbildung 6–3 (c) verdeutlicht dies an einem Beispiel. Die Disziplin „*Messung & Analyse*“ enthält das Dokument *Metrikkatalog*. Der Metrikkatalog dient als eine Vorgabe für die *Projektmanagement-Infrastruktur*, die Bestandteil der Disziplin *Projektmanagement* ist. Dieser Zusammenhang der Produkte ist im Beispiel über eine Produktabhängigkeit der Kategorie Produktfluss modelliert. Da wir die Erfassung von Metriken als wesentlichen Bestandteil der Projektmanagement-Infrastruktur erachten, modellieren wir eine Referenzierung des Metrikkatalogs ausgehend von der Disziplin Projektmanagement. Diese Referenzierung leitet sich allerdings nicht über ein Constraint des Vorgehens-Meta-Modells aus der Existenz der Produktabhängigkeit ab. Die Existenz einer Produktabhängigkeit dient lediglich als ein Indiz für die Modellierung eines Produktes als gemeinsames Produkt zweier Disziplinen.

Die Dokumente *Projektbewertung* und *Anforderungspapier* der Disziplinen *Projekt-Bewertung* beziehungsweise *Anforderungs-Erstellung* in Abbildung 6–3 (c) erachten wir im Beispiel trotz der Existenz eines Produktflusses zwischen beiden als weitgehend unabhängig. Die Projektbewertung kann auch in anderen Kontexten als der Erstellung von Anforderungen an das Vorgehensmodell VGM auftreten. Die Anforderungserstellung kann sich ebenfalls anderer Quellen als der Bewertung eines Projektes bedienen. Die beiden Produkte verstehen wir also nicht als eng gekoppelt und modellieren daher im Beispiel keine Referenzierung. Für den Leser einer Disziplin ist das jeweils andere Produkt also nicht sichtbar. Die Auswirkungen auf den Erstellungsprozess zeigen wir in Kapitel 6.5.

(a) Gemeinsame Produkte



(b) Inhaltliche Überlappung als Folge der Aggregation von Produkten



(c) Inhaltliche Überlappung und Produktabhängigkeiten

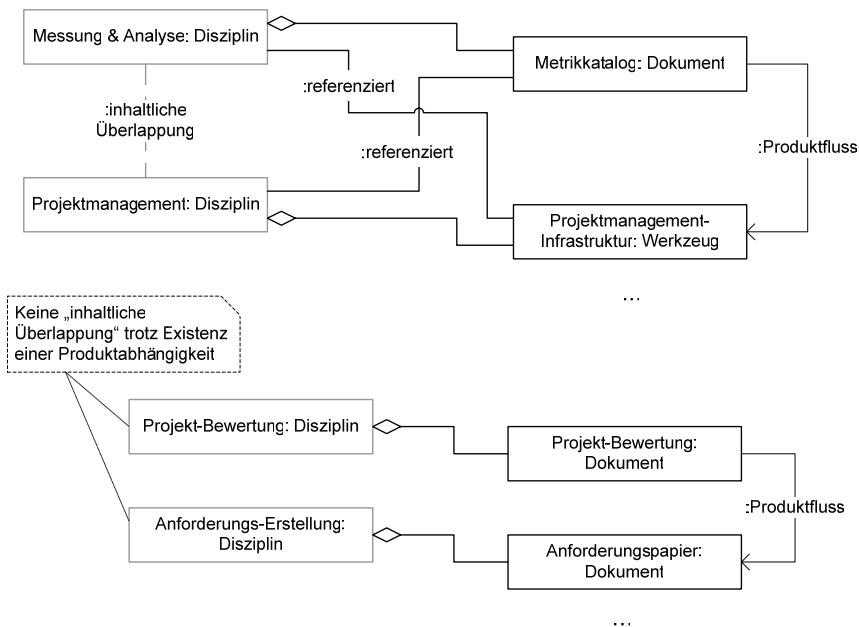


Abbildung 6–3: Modell zur inhaltlichen Strukturierung: Disziplinen – Ebene Vorgehensmodell (Beispiel)

Planungs-Einheiten und Disziplinen

In Abbildung 6–4 betrachten wir Beispiele für Planungs-Einheiten sowie die inhaltliche Überlappung von Planungs-Einheiten und Disziplinen. Assoziationen wie Aggregationen oder Abläufe stellen wir in Abbildung 6–4 nicht dar, da diese nicht Bestandteil der Planungs-Einheiten beziehungsweise Disziplinen sind. Eine Zuordnung ist nicht erforderlich. Unter dem Gesichtspunkt der inhaltlichen Gliederung sind für den Leser alle Assoziationen sichtbar, die mit einem Element der jeweiligen Disziplin beziehungsweise Planungs-Einheit in Beziehung stehen. Unter dem Gesichtspunkt des Tailorings und des Erstellungsprozesses ist ebenfalls keine Zuordnung von Assoziationen erforderlich (vgl. dazu Kapitel 6.2 und 6.5)

Abbildung 6–4 (a) zeigt die Planungs-Einheit „*Grobe Projektstruktur*“. Diese überlappt sich inhaltlich mit der Disziplin *Projektmanagement*. Die „Grobe Projektstruktur“ spiegelt die Einteilung eines Projektes in die Projektabschnitte *Initialisierung*, *Projektdurchführung* und *Abschluss* wider. Diese Projektabschnitte aggregieren im Vorgehensmodell Produkterstellungen. Diese sind in Abbildung 6–4 (a) nicht dargestellt, da Produkterstellungen gemäß dem Vorgehens-Meta-Modell nicht in Strukturierungs-Elemente eingeordnet werden. Die „Grobe Projektstruktur“ *referenziert* die Produkte, die von diesen Produkterstellungen erstellt werden. Die referenzierten Produkte befinden sich in der Disziplin *Projektmanagement*. Daher leitet sich aus Constraint (39) die „inhaltliche Überlappung“ zur Disziplin *Projektmanagement* ab.

In Abbildung 6–4 (b) zeigen wir ein Beispiel einer *querschnittlichen* Planungs-Einheit. Die Planungs-Einheit „*2 Phasen*“ als Bestandteil der Strukturierung Entwicklungsansätze teilt den Projektlauf in zwei Projektabschnitte ein. Die erste Phase dient dabei der Konzeption des VGM, während die zweite Phase die Umsetzung des VGM beschreibt (vgl. Kapitel 5.2.4). Diese Projektabschnitte können potenziell jede Produkterstellung aggregieren, ohne sich dabei auf die Produkte bestimmter Disziplinen zu beschränken. Die Kennzeichnung inhaltlicher Überlappungen würde also die Planungs-Einheit „*2 Phasen*“ mit den meisten Disziplinen des Vorgehensmodells in Beziehung setzen. Wir kennzeichnen die Planungs-Einheit aus diesem Grund als Hinweis für den Leser des Vorgehensmodells im Beispiel als *querschnittlich*. Eine Kennzeichnung konkreter inhaltlicher Überlappungen entfällt damit.

Abbildung 6–4 (c) zeigt ein Beispiel einer Planungs-Einheit im Zusammenhang mit Produktbearbeitungen. Anders als in den übrigen Beispielen in Abbildung 6–4 (a) und (b) stellen wir hier neben „inhaltlicher Überlappung“ und Referenzierung weitere Assoziationen dar, auch wenn diese nicht Bestandteil der Planungs-Einheiten sind. Das Beispiel zeigt eine generische Variante des *Überarbeitungs-Bewertungs-Zyklus*, den wir bereits in einer nicht-generischen Variante mit Bezug zum Strategiepapier als Beispiel herangezogen haben (vgl. Kapitel 5.3.1). Da der Bewertungs-Überarbeitungs-Zyklus allgemein im Kontext der Produkterstellung anwendbar ist, beziehen wir diese Planungs-Einheit allgemein auf Dokumente mit *Bewertungen* als Ergebnis. Die Planungs-Einheit stellt eine Anwendung des Prinzips generischer Vorgehensweisen dar (vgl. Kapitel 5.4.3). Die Referenzierung des Konzeptes Dokument des Vorgehens-Meta-Modells ist in Abbildung 6–4 (c) ausgelassen.

Entsprechend diesem Beispiel scheint es sich bei den meisten Planungs-Einheiten auf Ebene der Produktbearbeitung um generische Vorgehensweisen ohne Bezug zu einem bestimmten Produkt des Vorgehensmodells zu handeln. Die Ebene der Produktbearbeitung ist weder so feingranular, dass einzelne Arbeitsschritte angegeben würden, da diese nicht planungsrelevant sind. Noch werden auf Ebene der Produktbearbeitungen Arbeitsschritte formuliert, deren Ergebnis sich in Produkten niederschläge, da diese als Produkterstellungen geplant würden. Produktbearbeitungen tragen aber lediglich zur Produkterstellung bei.

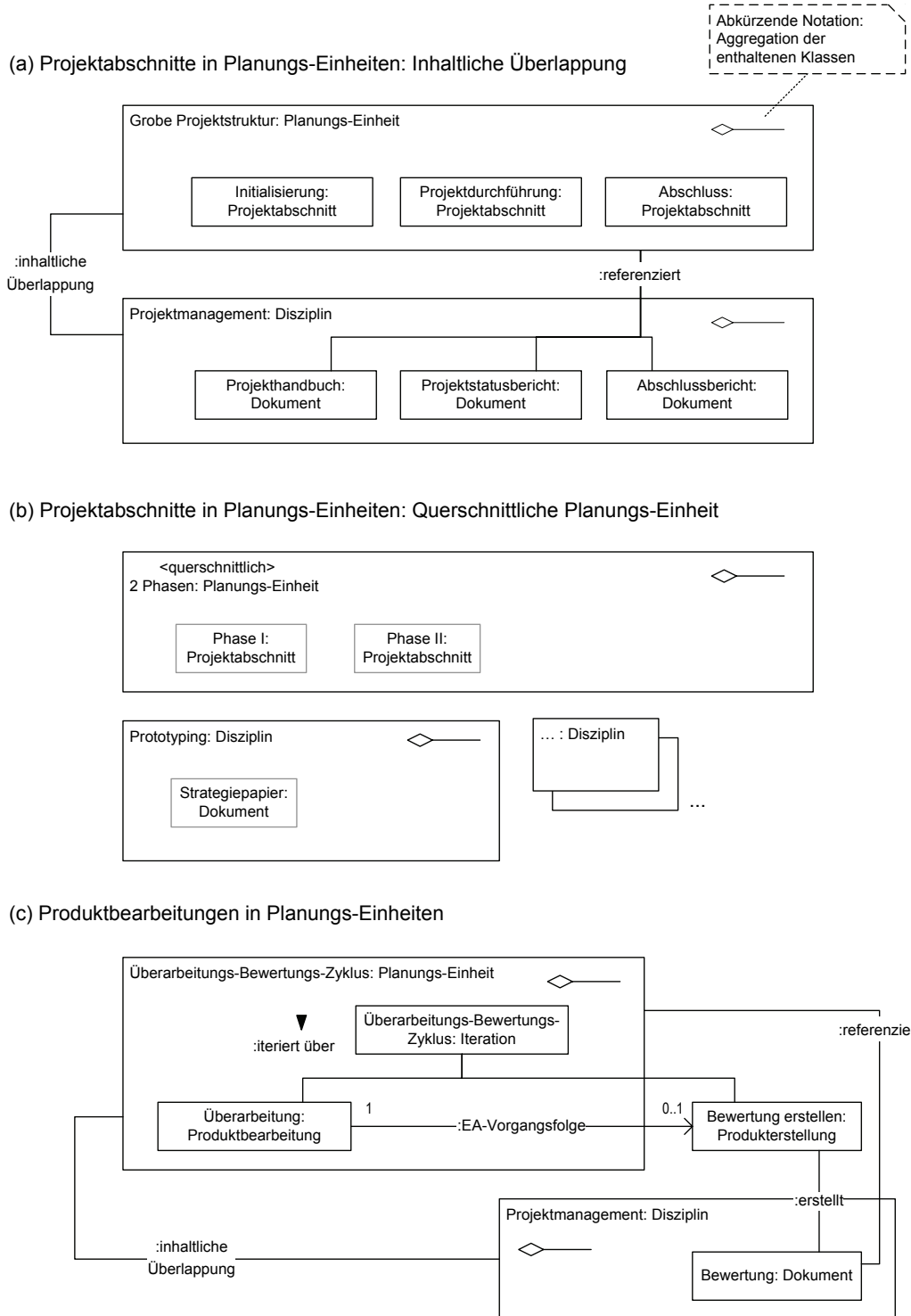


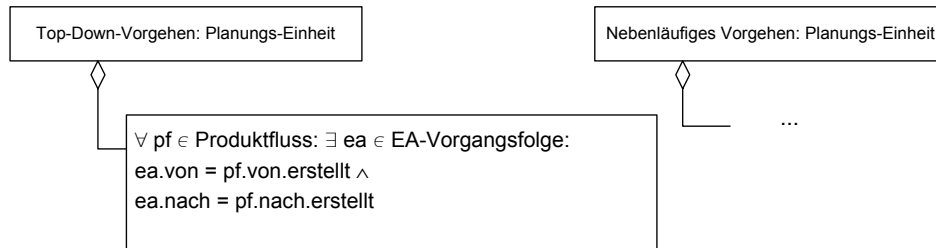
Abbildung 6–4: Modell zur inhaltlichen Strukturierung: Planungs-Einheiten – Ebene Vorgehensmodell (Beispiel)

Als weiteres Beispiel betrachten wir die in Abbildung 6–5 (a) dargestellten Planungseinheiten *Top-Down-Vorgehen* und „*Nebenläufiges Vorgehen*“, die Teil der Strukturierung Entwicklungsansätze sind. Dabei handelt es sich um Planungseinheiten, die ein Constraint auf Ebene des Vorgehensmodells als Einschränkung der Projektebene beinhalten. Beide Planungseinheiten legen die Auswirkungen der im Vorgehensmodell enthaltenen Produktflüsse auf die Abläufe im Projekt fest. Im Kontext eines Top-Down-Vorgehens impliziert ein Produktfluss eine Ende-Anfang-Vorgangsfolge, im Kontext eines nebenläufigen Vorgehens beispielsweise eine Ende-Ende-Vorgangsfolge (vgl. Kapitel 5.3.2). Während wir in Kapitel 5.3.2 das Constraint allgemeingültig auf Ebene des Vorgehens-Meta-Modells formuliert haben, besteht nun die Möglichkeit einer projektspezifischen Anpassung. Top-Down-Vorgehen und Nebenläufiges Vorgehen sind im Rahmen des Tailorings als Alternativen zu sehen (vgl. dazu Kapitel 6.2).

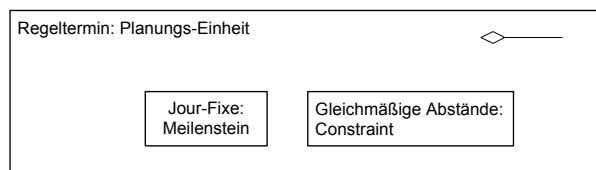
Die Planungseinheit *Regeltermin* in Abbildung 6–5 (b) stellt eine Planungseinheit der Strukturierung Management dar. Die Durchführung eines Jour-Fixe modellieren wir als ein Planungselement der Kategorie Meilenstein mit einer leeren Menge vorzulegender Produkte. Den Meilenstein Jour-Fixe versehen wir zusätzlich mit dem Constraint „*Gleichmäßige Abstände*“, das ein zeitliches Raster für die Einplanung von Jour-Fixes vorgibt.

Abbildung 6–5 (c) zeigt ein komplexeres Beispiel einer Planungseinheit. Die Planungseinheit *Time-Boxing* weist im Beispiel eine inhaltliche Überlappung mit der Disziplin *Vorgehensmodell-Aufbau* auf. Unter Time-Boxing verstehen wir die Unterteilung der Projektlaufzeit in Releases, die mit dem Erreichen eines Meilensteins abgeschlossen werden. Im Kontext der Erstellung des Vorgehensmodells VGM sind den Releases Arbeitspakete α -Version und β -Version zugeordnet, die jeweils mit dem Erreichen des Release-Abschlusses fertig gestellt werden sollen (vgl. Kapitel 5.2.4). Der Inhalt dieser Arbeitspakete wird jeweils in Abhängigkeit von dem Vorgehensbaustein des VGM definiert, dem das jeweilige Arbeitspaket zugeordnet ist. Die Auswirkungen des Time-Boxing auf weitere Planungselemente sind in Kapitel 5.2.4 in Constraint (22) festgelegt, das wir ebenfalls der Planungseinheit Time-Boxing zuordnen.

(a) Constraints in Planungs-Einheiten



(b) Meilensteine in Planungs-Einheiten



(c) Beispiel: Time Boxing

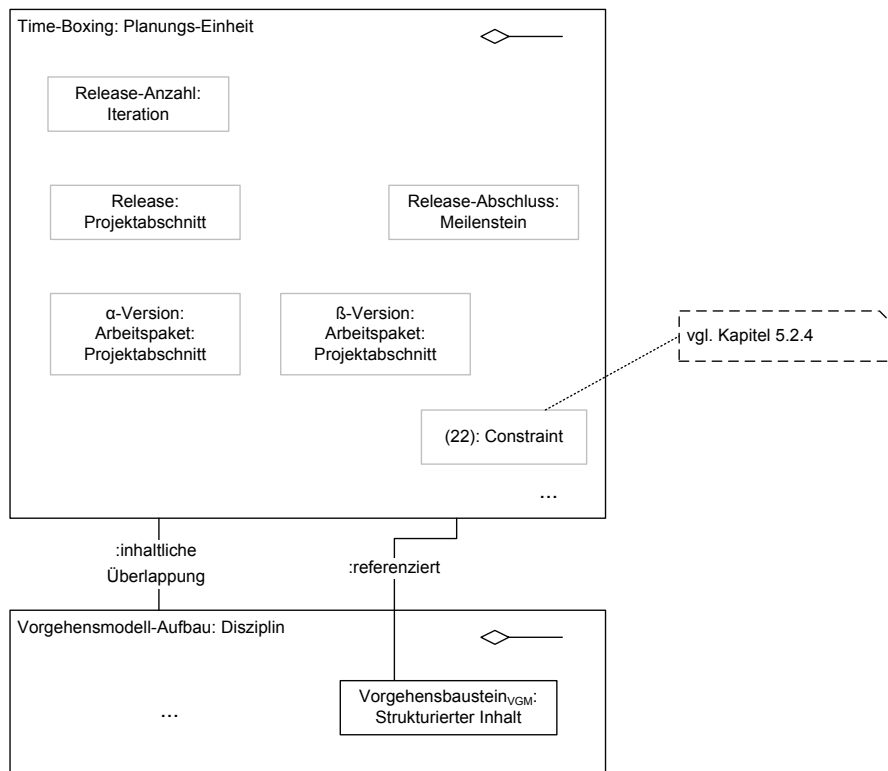


Abbildung 6–5: Modell zur inhaltlichen Strukturierung: Weitere Planungs-Einheiten – Ebene Vorgehensmodell (Beispiel)

Zusammenfassung

Die inhaltliche Strukturierung dient der Übersichtlichkeit eines Vorgehensmodells und ist zudem, wie wir in den folgenden Kapiteln zeigen, dem Tailoring und der Methodik der Erstellung von Vorgehensmodellen dienlich. Die Inhalte des Vorgehensmodells lassen sich in Form von Disziplinen, Planungs-Einheiten und Strukturierungen hierarchisch gliedern. Disziplinen enthalten dabei Produkte, Planungs-Einheiten enthalten Planungselemente und Constraints des Vorgehensmodells. Jedes Element des Vorgehensmodells mit Ausnahme der im Vorgehens-Meta-Modell definierten Assoziationen und Produkterstellungen wird eindeutig genau einer Disziplin beziehungsweise einer Planungs-Einheit zugeordnet. Referenzierung erlaubt den Verweis auf zusätzliche Produkte, die in Konsequenz für den Betrachter einer Disziplin beziehungsweise Planungs-Einheit ebenfalls sichtbar werden.

Inhaltliche Überlappungen kennzeichnen Vorgehensmodell-Inhalte, die eng aneinander gekoppelt sind und dienen damit für den Leser als Hinweis auf verwandte Themen. Referenzierung gibt den Bezug eines Strukturierungs-Elementes zu einem Produkt einer inhaltlich überlappenden Disziplin im Detail an. Die Referenzierung eines Produktes durch eine Disziplin ist in jedem Fall erforderlich, falls das Produkt in einer Aggregations-Beziehung zu einem Produkt der Disziplin steht. In diesem Fall ist die Kopplung der Produkte eng und die Sichtbarkeit dieser in beiden Disziplinen erforderlich. Weitere Regeln für die Ableitung von Referenzierungen aus den Zusammenhängen eines Vorgehensmodells geben wir mangels Allgemeingültigkeit nicht an. Im Falle der Existenz von Produktabhängigkeiten zwischen Produkten unterschiedlicher Disziplinen ist beispielsweise jeweils zu entscheiden, ob eine Referenzierung erforderlich ist.

Aus einer Referenzierung leitet sich eine Kennzeichnung der jeweiligen Disziplinen beziehungsweise Planungs-Einheiten als inhaltlich überlappend ab. Überlappt sich eine Planungs-Einheit mit den meisten beziehungsweise potentiell mit allen Disziplinen des Vorgehensmodells, kann sie als querschnittlich gekennzeichnet werden. Die Notwendigkeit einer Kennzeichnung der konkreten inhaltlichen Überlappungen entfällt damit.

6.2 Tailoring-Modell

In diesem Kapitel konzipieren wir ein Modell zur projektspezifischen Anpassung eines Vorgehensmodells. Unter dem Begriff des Tailorings verstehen wir analog zum V-Modell XT die projektspezifische Auswahl bestimmter Teile des Gesamtmodells für die Anwendung im Projekt. Die eigentliche Anwendung des Vorgehensmodells, also Tailoring und Ableitung eines Projektplans auf Basis der in diesem Kapitel definierten Konzepte beschreiben wir in Kapitel 6.4. Zielsetzung dieses Kapitels ist die Konzeption eines Modells für Tailoring-Einheiten auf Ebene des Vorgehens-Meta-Modells, die eine Strukturierung des Vorgehensmodells unter dem Gesichtspunkt des Tailorings erlauben. Gültige projektspezifische Auswahlmengen von Tailoring-Einheiten sollen auf Ebene des Vorgehensmodells eingeschränkt werden können.

Die Konzepte erläutern wir in diesem Kapitel anhand von Beispielen. Um die Anwendbarkeit des Konzeptes anhand eines umfangreicheren Beispiels zu zeigen, betrachten wir neben dem durchgängigen Beispiel in Kapitel 6.3 die Anwendung des Tailoring-Modells im Kontext des V-Modells XT.

Vorgehens-Meta-Modell

Abbildung 6–6 zeigt die Konzepte des Tailoring-Modells auf der Ebene des Vorgehens-Meta-Modells. Das Tailoring-Modell umfasst *Tailoring-Bausteine* und *Tailoring-Abhängigkeiten*. Tailoring-Bausteine sind Einheiten, die im Rahmen der projektspezifischen Anpassung eines Vorgehensmodells für die Anwendung im Projekt gewählt oder nicht gewählt werden können. Tailoring-Abhängigkeiten schränken die gültigen projektspezifischen Auswahlmengen bezüglich bestimmter Inhalte des Vorgehensmodells ein.

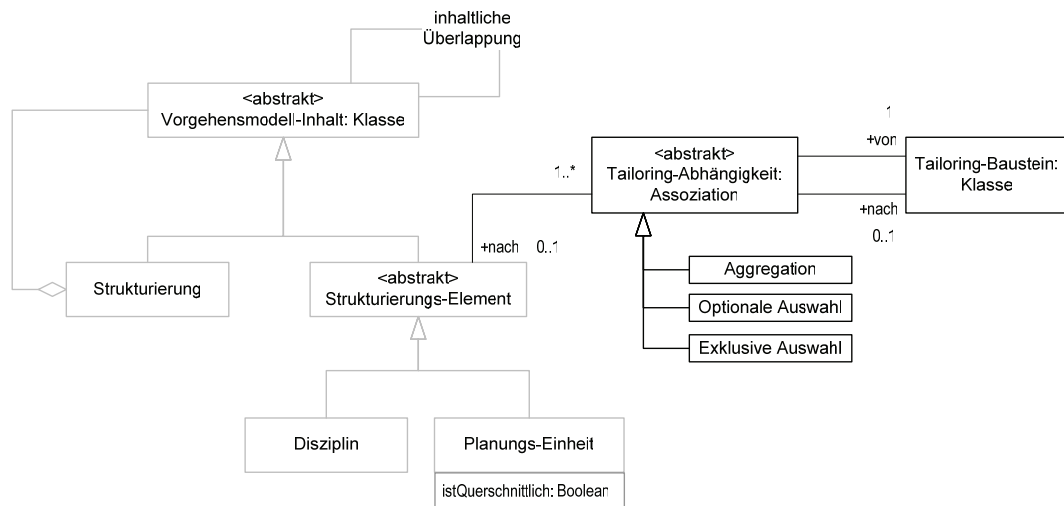


Abbildung 6–6: Tailoring-Modell – Ebene Vorgehens-Meta-Modell

(41)

Tailoring-Abhängigkeiten sind gerichtete binäre Assoziationen. Eine Tailoring-Abhängigkeit setzt entweder zwei Tailoring-Bausteine in Beziehung oder aber einen Tailoring-Baustein und eine Disziplin beziehungsweise eine Planungs-Einheit:

$$\forall ta \in \text{Tailoring-Abhängigkeit:} \\ | ta.nach^{\text{Strukturierungs-Element}} \cup ta.nach^{\text{Tailoring-Baustein}} | = 1$$

Disziplinen und Planungs-Einheiten und damit indirekt alle Elemente eines Vorgehensmodells werden über Tailoring-Abhängigkeiten den Tailoring-Bausteinen zugeordnet. Disziplinen und Planungs-Einheiten, die die tiefste Ebene der inhaltlichen Strukturierung eines Vorgehensmodells darstellen, bilden also zugleich die Basis-Einheiten des Tailorings. Der Zusammenhang zwischen inhaltlicher Strukturierung und Tailoring ist aus methodischen Gründen wichtig, wie wir im Folgenden noch darstellen (vgl. dazu Kapitel 6.5).

Tailoring-Abhängigkeiten schränken die Menge der gültigen projektspezifischen Auswahlmengen von Tailoring-Bausteinen beziehungsweise Strukturierungs-Elementen ein. Tailoring-Abhängigkeiten treten in den Ausprägungen *Aggregation*, „*Optionale Auswahl*“ und „*Exklusive Auswahl*“ auf.

Wird auf Projektebene ein Tailoring-Baustein gewählt, so bedeutet

- Aggregation, dass der abhängige Tailoring-Baustein (Assoziations-Ende *nach*) beziehungsweise das abhängige Strukturierungs-Element ebenfalls gewählt werden **muss**,
- „Optionale Auswahl“, dass der abhängige Tailoring-Baustein beziehungsweise das abhängige Strukturierungs-Element ebenfalls gewählt werden **kann**,
- „Exklusive Auswahl“, dass von allen Tailoring-Bausteinen beziehungsweise Strukturierungs-Elementen, die über eine derartige Beziehung mit dem jeweiligen Tailoring-Baustein in Beziehung stehen (Assoziations-Ende *nach*), genau einer gewählt werden muss.

Wir formalisieren die Semantik der Tailoring-Abhängigkeiten im Kontext des Kapitels zur Anwendung von Vorgehensmodellen (vgl. dazu Kapitel 6.4).

Die Zusammenhänge der Inhalte des Vorgehensmodells erfordern zum Teil eine geeignete Strukturierung der Tailoring-Abhängigkeiten. Beispielsweise Aggregations-Beziehungen zwischen Produkten unterschiedlicher Disziplinen wirken sich auf das Tailoring-Modell aus. Die alleinige projektspezifische Auswahl eines Produktes, das von einem anderen Produkt aggregiert wird, ist nicht sinnvoll ohne dabei das aggregierende Produkt auszuwählen. Die Auswahl eines Produktes ohne bestimmte aggregierte Produkte als Bestandteile kann dagegen sinnvoll sein.

(42)

Wird ein Tailoring-Baustein gewählt, der ein Produkt enthält, das aggregierter Teil eines anderen Produktes ist, so muss auch der Tailoring-Baustein gewählt werden, der das aggregierende Produkt enthält. Dieses Constraint formulieren wir indirekt. Letztlich müssen beide Produkte im Projektplan enthalten sein. Den Projektplan definieren wir formal in Kapitel 6.4:

$$\forall p_1, p_2 \in \text{Produkt: } p_2 \in p_1.\text{enthält}^{\text{teil}} \wedge p_2 \in \text{Projektplan} \Rightarrow p_1 \in \text{Projektplan}$$

Weitere Auswirkungen der Struktur der Elemente des Vorgehensmodells auf das Tailoring-Modell formalisieren wir nicht. Die Ausgestaltung des Tailoring-Modells obliegt weitgehend dem Prozessingenieur.

Ebene Vorgehensmodell: Beispiel

Abbildung 6–7 zeigt ein Beispiel eines Tailoring-Modells auf Ebene des Vorgehensmodells. Dabei setzen wir das Beispiel des vorangehenden Kapitels fort. Wir ergänzen das Modell zur inhaltlichen Strukturierung um Tailoring-Bausteine und Tailoring-Abhängigkeiten. Die Tailoring-Abhängigkeiten Aggregation stellen wir in Abbildung 6–7 in der üblichen Notation für Aggregation mit einem Rauten-Symbol dar.

Der Tailoring-Baustein *Projekt* ist im Beispiel der einzige Baustein ohne eingehende Abhängigkeiten und dient daher als Ausgangspunkt für die projektspezifische Anpassung (vgl. dazu Kapitel 6.4). Der Baustein *Projekt* aggregiert den Tailoring-Baustein *Vorgehens-Kern*. Für jedes Projekt ist also die Auswahl des Vorgehens-Kerns ebenfalls erforderlich. Über Tailoring-Abhängigkeiten „*exklusive Auswahl*“ steht der Tailoring-Baustein *Projekt* mit unterschiedlichen Projekttypen in Beziehung, das heißt genau ein Projekttyp muss gewählt werden. Wir stellen dabei im Beispiel allein den Tailoring-Baustein „*Erstellung eines Vorgehensmodells*“ als Projekttyp dar. Eine für ein Projekt „*optionale Auswahl*“ stellt der Tailoring-Baustein *Ablauf* dar, der für ein Projekt gewählt werden kann, aber nicht gewählt werden muss.

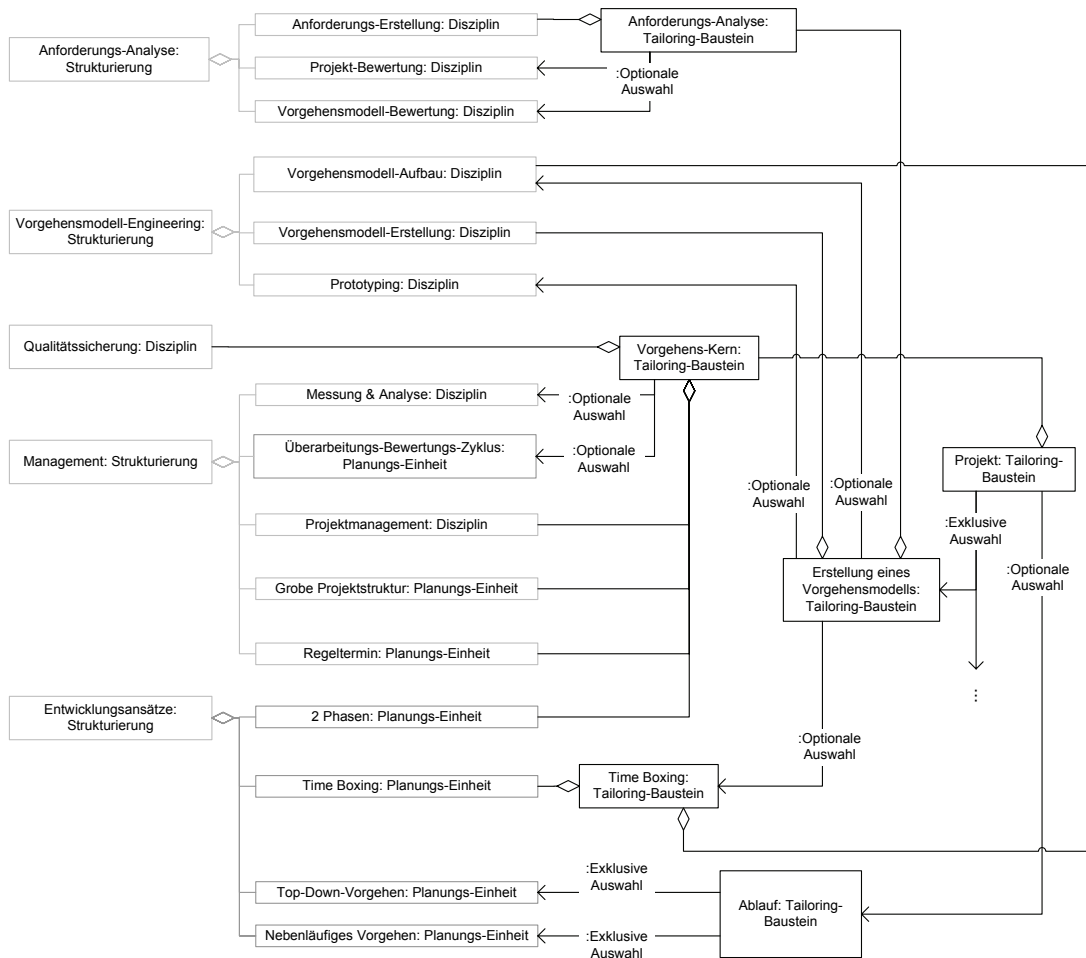


Abbildung 6–7: Tailoring-Modell – Ebene Vorgehensmodell (Beispiel)

Den Tailoring-Baustein Vorgehens-Kern verstehen wir als Baustein, der für alle Projekte relevant ist. Im Beispiel aggregiert der Vorgehens-Kern die Disziplinen *Qualitätssicherung* und *Projektmanagement* sowie die Planungs-Einheiten „*Grobe Projektstruktur*“, *Regeltermin* und „*2 Phasen*“. Die Disziplin „*Messung & Analyse*“ und die Planungs-Einheit „*Überarbeitungs-Bewertungs-Zyklus*“ sind dagegen als optional modelliert.

Der Tailoring-Baustein „Erstellung eines Vorgehensmodells“ beinhaltet verpflichtend den Tailoring-Baustein *Anforderungs-Analyse* sowie die Disziplin *Vorgehensmodell-Erstellung*. Die Disziplin *Vorgehensmodell-Aufbau*, die zusätzlich in Form von Vorgehensbausteinen beziehungsweise Produkten des VGM den spezifischen inhaltlichen Aufbau des zu erstellenden Vorgehensmodells VGM vorgibt, ist dagegen für die „Erstellung eines Vorgehensmodells“ als optional gekennzeichnet. Die Disziplin Prototyping ist für die „Erstellung eines Vorgehensmodells“ ebenfalls optional.

Weder der Vorgehensmodell-Aufbau noch das Prototyping können in einem Projekt ohne die Disziplin Vorgehensmodell-Erstellung auftreten, da beide allein ausgehend vom Tailoring-Baustein „Erstellung eines Vorgehensmodells“ gewählt werden können. Diese Einschränkung bezüglich gültiger Auswahlmengen von Tailoring-Bausteinen formulieren wir allgemein im folgenden Kapitel (vgl. dazu Kapitel 6.4). Für die Disziplin Vorgehensmodell-Aufbau muss die gemeinsame Auswahl mit der Disziplin Vorgehensmodell-Erstellung aufgrund von Constraint (42) gewährleistet sein. Die Disziplin Vorgehensmodell-Aufbau definiert beispielsweise das Produkt Vorgehensbaustein als aggregierten Teil des Produktes VGM der Disziplin Vorgehensmodell-Erstellung.

Wird der Tailoring-Baustein „*Time Boxing*“ als optionale Ergänzung der „Erstellung eines Vorgehensmodells“ gewählt, führt dies neben der Anwendung der Planungs-Einheit „*Time Boxing*“ zur verpflichtenden Auswahl des Tailoring-Bausteins Vorgehensmodell-Aufbau. Die Planungs-Einheit „*Time Boxing*“ referenziert das Produkt Vorgehensbaustein des VGM (vgl. Kapitel 6.1, Abbildung 6–4), da Arbeitspakete in Abhängigkeit von Vorgehensbausteinen definiert werden. Die Anwendung der Planungs-Einheit „*Time Boxing*“ setzt also die Verwendung der Strukturierung eines Vorgehensmodells voraus, die in der Disziplin Vorgehensmodell-Aufbau beschrieben ist. Dieser Zusammenhang leitet sich nicht aus einem Constraint ab. Die geeignete Modellierung obliegt dem Prozessingenieur.

Die Disziplinen Projekt-Bewertung und Vorgehensmodell-Bewertung treten auf Projektebene nur gemeinsam mit der Disziplin Anforderungs-Erstellung auf. Projekt-Bewertung und Vorgehensmodell-Bewertung sind ausgehend vom Tailoring-Baustein Anforderungs-Analyse als optional gekennzeichnet. Die Planungs-Einheiten Top-Down-Vorgehen und „Nebenläufiges Vorgehen“ sind Alternativen. Die exklusive Auswahl beider Planungs-Einheiten ausgehend vom Tailoring-Baustein Ablauf stellt sicher, dass beide nicht gemeinsam in einem Projekt auftreten.

Inhaltliche Überlappungen und Tailoring

Im Vorgehens-Meta-Modell geben wir keinen formalen Zusammenhang zwischen inhaltlicher Überlappung der Disziplinen beziehungsweise Planungs-Einheiten und dem Tailoring-Modell an. Beispielsweise erscheint die gemeinsame Auswahl der Planungs-Einheit „*grobe Projektstruktur*“ und der Disziplin „*Projektmanagement*“, die sich inhaltlich überlappen, sinnvoll und notwendig, da die Projektabschnitte der „*groben Projektstruktur*“ allein Produkte des Projektmanagements beinhalten (vgl. Abbildung 6–4 (a)).

Dagegen lässt sich aus dem Beispiel der Planungs-Einheit „*Überarbeitungs-Bewertungs-Zyklus*“, die sich ebenfalls mit der Disziplin Projektmanagement überlappt, keine Auswirkung auf das Tailoring-Modell ableiten (vgl. Abbildung 6–4 (c)). Eine Anwendung des „*Überarbeitungs-Bewertungs-Zyklus*“ ist auch ohne die Erstellung einer schriftlich dokumentierten Bewertung, die Bestandteil der Disziplin Projektmanagement ist, vorstellbar. Inhaltliche Überlappung betrachten wir also lediglich als Indiz für die Zusammengehörigkeit von Disziplinen beziehungsweise Planungs-Einheiten im Rahmen des Tailoring, ein präziser Zusammenhang besteht allerdings nicht.

Zusammenfassung

Das Tailoring-Modell dient der Formulierung der Möglichkeiten zur projektspezifischen Anpassung eines Vorgehensmodells. Tailoring-Bausteine fassen Elemente des Vorgehensmo-

dells zusammen, die auf Projektebene gewählt oder nicht gewählt werden können. Tailoring-Abhängigkeiten dienen der Einschränkung gültiger projektspezifischer Auswahlmengen von Tailoring-Bausteinen. Tailoring-Abhängigkeiten treten in den Ausprägungen Aggregation, optionale Auswahl und exklusive Auswahl auf.

Die Basis des Tailoring-Modells bilden Disziplinen beziehungsweise Planungs-Einheiten, die zugleich die tiefste Ebene des Modells der inhaltlichen Strukturierung des Vorgehensmodells darstellen. Die Granularität des Tailorings entspricht also der Granularität der inhaltlichen Strukturierung. Dies ist aus methodischen Gründen von Bedeutung, die wir in Kapitel 6.5 angeben.

Die Zusammenhänge der Elemente der Disziplinen beziehungsweise Planungs-Einheiten erfordern eine geeignete Festlegung des Tailoring-Modells. Beispielsweise Aggregations-Beziehungen zwischen Produkten unterschiedlicher Disziplinen müssen bei einer Auswahl eines Teilproduktes stets zu einer Auswahl des „Gesamt-Produktes“ führen. Dies formulieren wir als Constraint für Spezialisierungen des Vorgehens-Meta-Modells. Dieses Constraint lässt sich durch die geeignete Modellierung von Tailoring-Abhängigkeiten im Vorgehensmodell erfüllen. Anders als im V-Modell XT sind die Auswirkungen der Struktur des Vorgehensmodells auf die Einheiten des Tailorings abgesehen von Aggregation zwischen Produkten unterschiedlicher Disziplinen nicht präzise festgelegt. Aus inhaltlichen Überlappungen der Disziplinen beziehungsweise Tailoring-Einheiten leiten sich keine Konsequenzen für das Tailoring ab, da trotz inhaltlicher Überlappung die Auswahl nur einer Disziplin beziehungsweise Planungs-Einheit im Rahmen des Tailorings sinnvoll sein kann. Dies haben wir anhand von Beispielen gezeigt.

Im V-Modell XT bedingt die Existenz einer Produktabhängigkeit zwischen Produkten zweier Vorgehensbausteine die Notwendigkeit einer Vorgehensbaustein-Abhängigkeit (vgl. Kapitel 3.1.1). Vorgehensbaustein-Abhängigkeiten schränken die gültigen Auswahlmengen von Vorgehensbausteinen im Rahmen des Tailorings zum Teil unnötig ein. Das Tailoring-Modell der vorliegenden Arbeit weist mehr Freiheitsgrade auf. Die geeignete Modellierung des Tailorings obliegt hier dem Prozessingenieur.

6.3 Studie: Tailoring anhand von Beispielen des V-Modells XT

Im diesem Kapitel skizzieren wir die Anwendung des Modells zur inhaltlichen Strukturierung sowie des Tailoring-Modells anhand von Beispielen aus dem V-Modell XT. Dabei beziehen wir uns auf die in Kapitel 3.1 identifizierten Schwachstellen des V-Modells im Zusammenhang mit dem Konzept des Vorgehensbausteins.

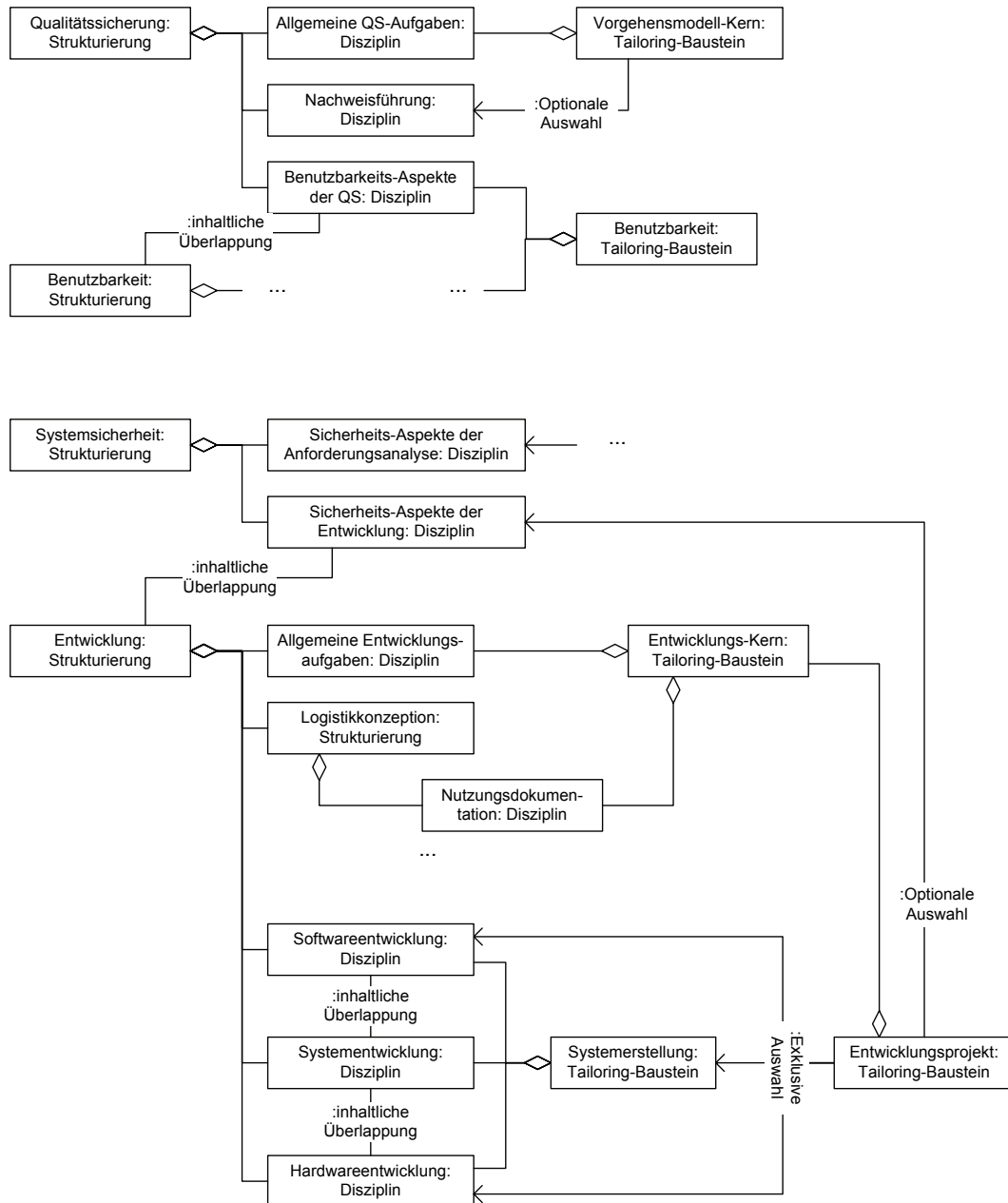


Abbildung 6–8: Modellierung von inhaltlicher Strukturierung und Tailoring am Beispiel des V-Modells XT

Abbildung 6–8 zeigt einen Lösungsansatz zu den Problemen, die wir in Kapitel 3.1 in Abbildung 3–3 im Zusammenhang mit dem Konzept des Vorgehensbausteins des V-Modells XT dargestellt haben. Den in Abbildung 3–3 dargestellten Vorgehensbausteinen entsprechen in Abbildung 6–8 die Strukturierungen *Qualitätssicherung*, *Benutzbarkeit*, *Systemsicherheit* und *Entwicklung*.

Strukturierungen sind Gruppierungen der Elemente eines Vorgehensmodells nach inhaltlichen Gesichtspunkten, die entsprechend den Vorgehensbausteinen des V-Modells XT im Erstellungsprozess als modulare Bearbeitungs-Einheiten dienen können (vgl. dazu Kapitel 6.5). Anders als die Vorgehensbausteine des V-Modells XT können diese Strukturierungen in Form von Strukturierungen, Disziplinen und Planungs-Einheiten weiter untergliedert werden. Da Disziplinen und Planungs-Einheiten zudem die Basis des Tailorings bilden, wird ein feingranulares Tailoring möglich.

Der Tailoring-Baustein *Vorgehensmodell-Kern* beinhaltet die Disziplin „*Allgemeine QS-Aufgaben*“, die für jedes Projekt grundlegende Aspekte der Qualitätssicherung enthält. Aspekte wie die *Nachweisführung*, die wir inhaltlich ebenfalls als Bestandteil der Qualitätssicherung sehen, sind nicht aggregierter Bestandteil des Tailoring-Bausteins Vorgehensmodell-Kern, sondern stellen ausgehend vom Vorgehensmodell-Kern eine optionale Auswahl dar.

Ein weiterer Bestandteil der Strukturierung Qualitätssicherung sind Aspekte der Benutzbarkeit, die wir in der Disziplin „*Benutzbarkeits-Aspekte der QS*“ sammeln. Aus Sicht des Tailorings ist die Benutzbarkeit allein bei Auswahl des Tailoring-Bausteins *Benutzbarkeit* relevant. Anders als der Vorgehensbaustein Qualitätssicherung im V-Modell XT ist die Strukturierung Qualitätssicherung im Beispiel als Bearbeitungs-Einheit geeignet, da sich alle inhaltlichen Aspekte der Qualitätssicherung in dieser Strukturierung befinden. Die dargestellten Strukturierungen sind nicht frei von Überlappungen. Benutzbarkeit und Qualitätssicherung überlappen sich inhaltlich beispielsweise hinsichtlich der Durchführung von Prüfungen. Die dargestellte inhaltliche Überlappung erfordert im Erstellungsprozess eine Zusammenarbeit der Bearbeiter bei der Erstellung der Strukturierungen Benutzbarkeit und Qualitätssicherung (vgl. dazu Kapitel 6.5).

Während sich der Vorgehensbaustein Systemsicherheit im V-Modell XT aus Sicht des Tailorings ungünstig darstellt, unterteilen wir die Strukturierung Systemsicherheit in die Disziplinen „*Sicherheits-Aspekte der Anforderungsanalyse*“ und „*Sicherheits-Aspekte der Entwicklung*“. Diese Disziplinen ermöglichen ein feingranulares Tailoring. Durch die Zusammenfassung beider Disziplinen in der Strukturierung Systemsicherheit stellt sich die Strukturierung Systemsicherheit analog dem Vorgehensbaustein Systemsicherheit im V-Modell XT als geeignete Bearbeitungs-Einheit dar.

In der Strukturierung *Entwicklung* fassen wir die Inhalte mehrerer Vorgehensbausteine des V-Modells XT zusammen. Die Disziplin „*allgemeine Entwicklungsaufgaben*“ beinhaltet grundlegende Aufgaben wie beispielsweise die Erstellung eines Pflichtenheftes, die für jedes Entwicklungsprojekt anwendbar sind. Die „*allgemeinen Entwicklungsaufgaben*“ sind als Bestandteil des Tailoring-Bausteins *Entwicklungs-Kern*.

Die Disziplinen *Softwareentwicklung* und *Hardwareentwicklung* beinhalten jeweils spezifische Aspekte für die Entwicklung von Software beziehungsweise Hardware wie beispielsweise die Beschreibung eines Produktes SW-Einheit. Die Disziplin *Systementwicklung* beinhaltet Aspekte für die Entwicklung eines Systems mit Software- und Hardware-Anteilen zugleich, beispielsweise also die Beschreibung eines Produktes Segment als Zusammenfassung von SW- und HW-Einheiten. Anders als im V-Modell XT modellieren wir das Tailoring als exklusive Auswahl zwischen den Disziplinen Softwareentwicklung, Hardwareentwicklung und dem Tailoring-Baustein *Systemerstellung*. Der Tailoring-Baustein Systemerstellung setzt sich aus Software-, Hardware- und Systementwicklung zusammen. Im Gegensatz zum V-Modell XT ist die Systementwicklung also allein für Projekte relevant, die sich mit der Entwicklung von Software und Hardware zugleich beschäftigen.

Trotzdem kann die Disziplin Systementwicklung in Abbildung 6–8 im Rahmen des Erstellungsprozesses eines Vorgehensmodells als gemeinsame Basis dienen. Sowohl Softwareentwicklung als auch Hardwareentwicklung überlappen sich inhaltlich mit der Systementwicklung. Im Erstellungsprozess des Vorgehensmodells führt dies zu einer Zusammenarbeit bei der Erarbeitung der Inhalte (vgl. dazu Kapitel 6.5).

Im V-Modell XT stellt sich schließlich der Vorgehensbaustein Logistikkonzeption als ungeeignete Tailoring-Einheit dar, da bestimmte Aspekte der Logistik durchaus im Rahmen des Tailorings wählbar sein sollten, andere Aspekte wie die Nutzungsdokumentation aber für alle Entwicklungs-Projekte verbindlich sein sollten. Wir modellieren daher die Disziplin *Nutzungsdokumentation* als Teil der Strukturierung *Logistikkonzeption*. Die Disziplin *Nutzungsdokumentation* modellieren wir als Bestandteil des Tailoring-Bausteins Entwicklungs-Kern, während wir andere Teile der Logistikkonzeption vom Entwicklungs-Kern ausnehmen.

Durch eine Trennung der Konzepte Tailoring-Einheit und Bearbeitungs-Einheit entsprechend dem hier konzipierten Vorgehens-Meta-Modell lassen sich also die Probleme vermeiden, die sich aus dem Konzept des Vorgehensbausteins im V-Modell XT ergeben.

6.4 Anwendung von Vorgehensmodellen: Tailoring und Ableitung eines Projektplans

In diesem Kapitel definieren wir auf Basis der Konzepte des Vorgehens-Meta-Modells die Anwendung von Vorgehensmodellen. Dabei betrachten wir das Tailoring sowie die Ableitung eines Projektplans. Zielsetzung dieses Kapitels ist die präzise Definition des Tailoring-Mechanismus sowie des Mechanismus zur Ableitung eines Projektplans aus einem entsprechend dem Vorgehens-Meta-Modell beschriebenen Vorgehensmodell. Wir illustrieren das Konzept der Anwendung von Vorgehensmodellen anhand eines einfachen Beispiels.

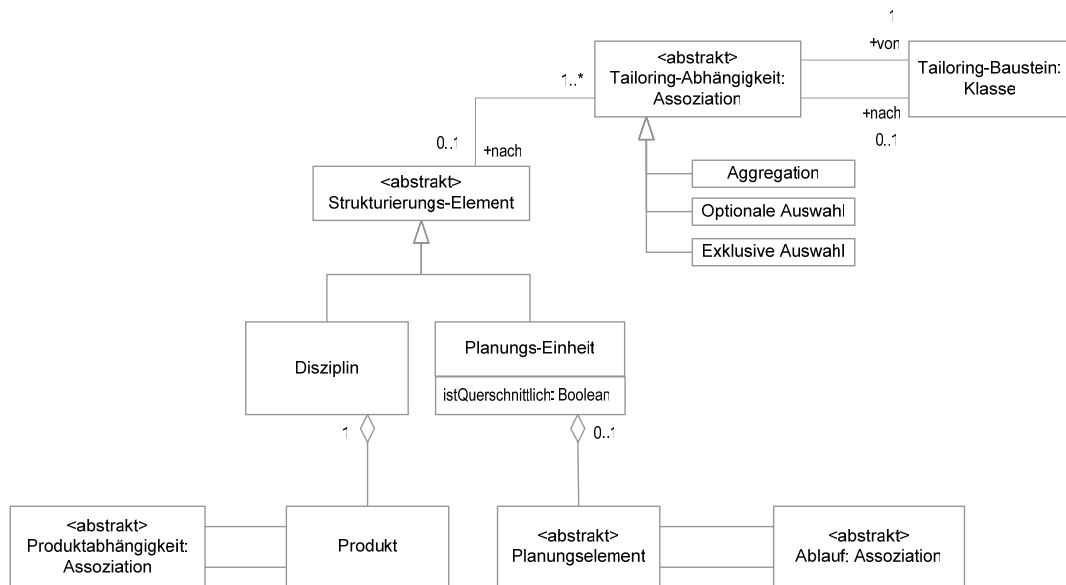


Abbildung 6–9: Anwendung von Vorgehensmodellen: Ebene Vorgehens-Meta-Modell

Wir definieren projektspezifische Anpassung und Ableitung des Projektplans mit Hilfe von Constraints, die die Projektebene einschränken. Abbildung 6–9 zeigt den bereits in den vorangehenden Kapiteln dargestellten Teil des Vorgehens-Meta-Modells, auf den sich diese Constraints beziehen.

Zunächst geben wir Regeln für die projektspezifische Anpassung an. Unter der projektspezifischen Anpassung verstehen wir die Spezialisierung von Tailoring-Bausteinen, Disziplinen und Planungs-Einheiten des Vorgehensmodells auf Projektebene. Diese werden auf Projektebene entweder für die Projektdurchführung gewählt, das heißt auf Projektebene einmal spezialisiert, oder aber finden keine Anwendung im Projekt.

Tailoring

(43)

Tailoring-Bausteine, Disziplinen und Planungs-Einheiten werden auf Projektebene höchstens einfach spezialisiert. Dies legen wir über die Eigenschaft *Einelementig* (vgl. Kapitel 4.2.2) auf Ebene des Vorgehensmodells als Einschränkung der Projektebene fest:

$$\begin{aligned} \forall tb \in \text{Tailoring-Baustein}: & \text{tb.istEinelementig} = \text{True} \wedge \\ \forall vi \in \text{Strukturierungs-Element}: & \text{vi.istEinelementig} = \text{True} \end{aligned}$$

Die mit den unterschiedlichen Tailoring-Abhängigkeiten Aggregation, Exklusive Auswahl und Optionale Auswahl verbundenen Semantiken lassen sich mit Hilfe von Multiplizitäten als Einschränkungen der Projektebene präzise beschreiben.

(44)

Wird auf Projektebene ein Tailoring-Baustein spezialisiert, der andere Tailoring-Bausteine beziehungsweise Disziplinen oder Planungs-Einheiten über eine Tailoring-Abhängigkeit aggregiert, müssen diese auf Projektebene ebenfalls spezialisiert werden. Dies formulieren wir als Multiplizität des Assoziations-Endes nach der Tailoring-Abhängigkeit Aggregation:

$$\begin{aligned} \forall a \in \text{Aggregation:Tailoring-Abhängigkeit}: \\ a.\text{nach}^{\text{Assoziations-Ende}}.\text{untere Grenze} = 1 \wedge a.\text{nach}^{\text{Assoziations-Ende}}.\text{obere Grenze} = 1 \end{aligned}$$

(45)

Wird auf Projektebene ein Tailoring-Baustein spezialisiert, der über eine Tailoring-Abhängigkeit eine exklusive Auswahl anderer Tailoring-Bausteine beziehungsweise Disziplinen oder Planungs-Einheiten vorsieht, muss genau ein Element aus dieser Menge auf Projektebene spezialisiert werden. Der Ausdruck „Tailoring-Baustein.spezialisierung“ bezieht sich dabei auf Elemente der Projektebene:

$$\begin{aligned} \forall ea \in \text{Exklusive Auswahl}: \\ \exists tb \in \text{Tailoring-Baustein}: & \text{tb} \in \text{ea.von}^{\text{Tailoring-Baustein}}.\text{spezialisierung} \Rightarrow \\ & | \text{tb.exklusive Auswahl.nach}^{\text{Tailoring-Baustein}} \cup \\ & \text{tb.exklusive Auswahl.nach}^{\text{Strukturierungs-Element}} | = 1 \end{aligned}$$

Für Tailoring-Abhängigkeiten der Kategorie „optionale Auswahl“ benötigen wir kein Constraint, da die „optionale Auswahl“ nicht mit Einschränkungen verbunden ist.

Um sicherzustellen, dass Tailoring-Bausteine beziehungsweise Disziplinen oder Planungs-Einheiten auf Projektebene nicht ohne mindestens einen Tailoring-Baustein spezialisiert werden, in den sie eingeordnet sind, formulieren wir das folgende Constraint.

(46)

Gibt es zu einer Disziplin, einer Planungs-Einheit oder einem Tailoring-Baustein auf Projektebene auf Ebene des Vorgehensmodells eingehende Tailoring-Abhängigkeiten, muss mindestens einer der über eine eingehende Tailoring-Abhängigkeit verbundenen Tailoring-Bausteine auf Projektebene spezialisiert werden. Der Ausdruck „x.spezialisierung“ bezieht sich auf die Projektebene:

$$\begin{aligned} \forall x \in (\text{Disziplin} \cup \text{Planungs-Einheit} \cup \text{Tailoring-Baustein}): \\ \exists ta \in \text{Tailoring-Abhängigkeit}: x = ta.nach \wedge x.spezialisierung \neq \emptyset \Rightarrow \\ | x.spezialisierung.Tailoring-Abhängigkeit.von^{\text{Tailoring-Baustein}} | \geq 1 \end{aligned}$$

Ohne diese Einschränkung könnten beliebige Kombinationen von Strukturierungselementen auf Projektebene spezialisiert werden. Mit Hilfe von Tailoring-Abhängigkeiten ließe sich dann nicht sicherstellen, dass bestimmte Elemente des Vorgehensmodells in einem Projekt nur gemeinsam mit bestimmten anderen verwendet werden. Beispielsweise Aggregations-Beziehungen zwischen Produkten unterschiedlicher Disziplinen erfordern allerdings eine entsprechende Einschränkung der projektspezifischen Auswahlmöglichkeiten (vgl. Kapitel 6.2, Constraint (42)).

Auf Projektebene werden allein die Elemente des Vorgehensmodells spezialisiert, die in den projektspezifisch gewählten Disziplinen beziehungsweise Planungs-Einheiten enthalten sind. Dies lässt sich auch wie folgt ausdrücken. Wird ein Element des Vorgehensmodells auf Projektebene spezialisiert, muss auch die zugehörige Disziplin beziehungsweise Planungs-Einheit spezialisiert werden. Dies leitet sich aus Constraint (36) in Kapitel 6.1 ab, das auch als Einschränkung der Projektebene gilt:

$$\forall pe \in \text{Planungselement}: pe \neq \text{Produkterstellung} \Rightarrow | p.\text{Planungs-Einheit} | = 1$$

Elemente des Vorgehensmodells, die in projektspezifisch nicht gewählten Tailoring-Bausteinen enthalten sind, werden also auch nicht projektspezifisch spezialisiert.

Eine Verpflichtung zur Spezialisierung der Elemente des Vorgehensmodells, die in den projektspezifisch gewählten Disziplinen beziehungsweise Planungs-Einheiten enthalten sind, formulieren wir allerdings nicht. Voraussetzung für die sinnvolle Existenz bestimmter Produktexemplare auf Projektebene können neben der Einordnung des jeweiligen Produktes in eine Disziplin auch die Inhalte anderer Produktexemplare sein.

**Definition
Projektplan**

Den Projektplan definieren wir nun wie folgt. Der Projektplan enthält alle Planungselemente und alle Abläufe, die im Rahmen des Tailorings gewählt wurden auf der aktuell jeweils tiefsten Spezialisierungsebene. Die Definition der Projektebene als tiefste Spezialisierung begründet sich aus der möglichen Existenz mehrfacher Spezialisierungsebenen auf Ebene des Vorgehensmodells (vgl. Kapitel 5.4.1).

(47)

Der Projektplan enthält alle Planungselemente und Abläufe, die nicht weiter spezialisiert sind und

- die Teil einer auf Projektebene spezialisierten Planungs-Einheit sind,
- oder die eine Produkterstellung eines Produktes darstellen, das Bestandteil einer auf Projektebene spezialisierten Disziplin ist,
- oder die eine Assoziation zwischen zwei Elementen des Projektplans darstellen wie beispielsweise einen Ablauf oder eine Aggregations-Beziehung zwischen zwei Planungselementen:

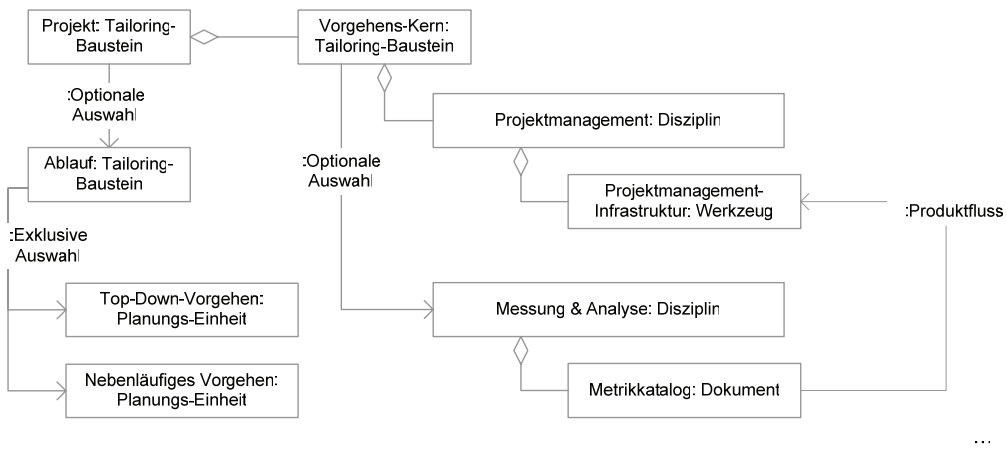
$$\begin{aligned} \forall x \in (\text{Planungselement} \cup \text{Ablauf}): x \in \text{Projektplan.enthält} &\Leftrightarrow \\ x.\text{spezialisierung} = \emptyset \wedge & \\ (x \in \text{Planungs-Einheit.spezialisierung.enthält} \vee & \\ x \in \text{Disziplin.spezialisierung.enthält} \overset{\text{Produkt}}{\text{erstellt}} \overset{\text{Produkterstellung}}{\vee} & \\ (\exists e_1, e_2 \in \text{Projektplan.enthält}: x \in \text{Assoziation} \wedge x.\text{quelle} = e_1 \wedge x.\text{ziel} = e_2)) & \end{aligned}$$

**Beispiel der
Anwendung
eines Vor-
gehensmo-
dells**

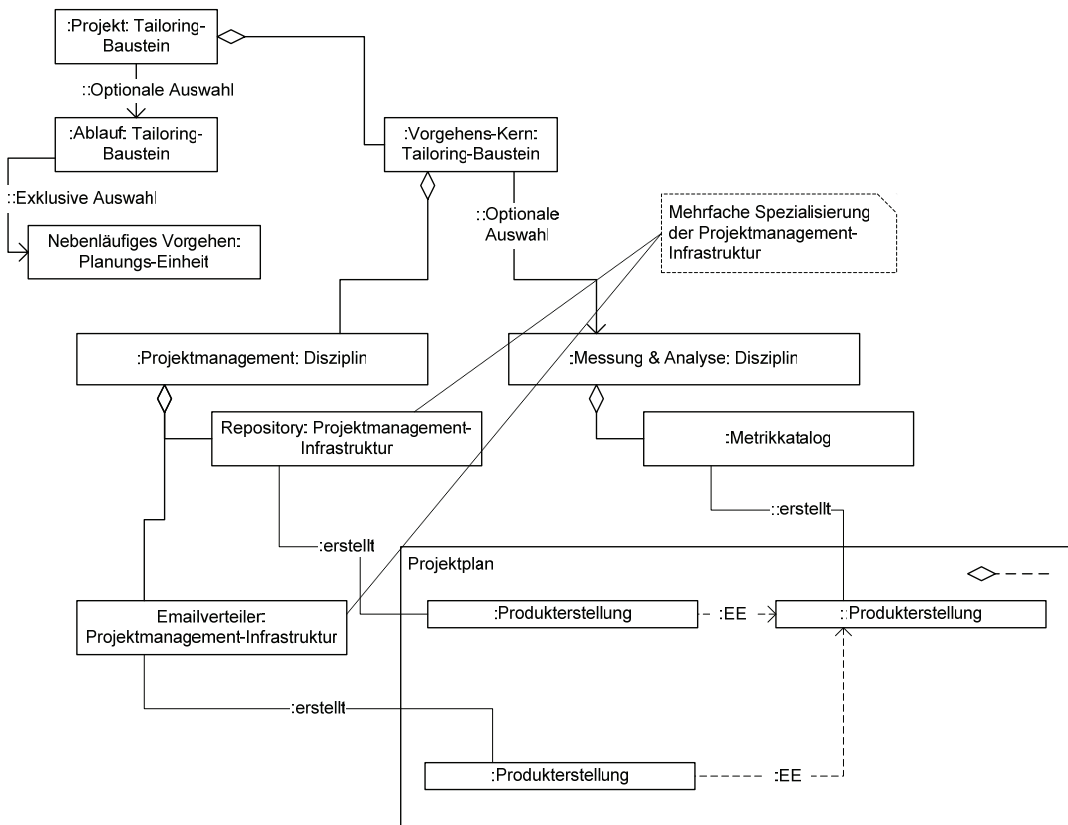
In Abbildung 6–10 zeigen wir das Konzept der Anwendung von Vorgehensmodellen anhand eines Beispiels. Wir beschränken uns dabei auf einen Ausschnitt des in der Arbeit verwendeten durchgängigen Beispiels, um den Umfang der Darstellung nicht ausufern zu lassen. Wir verdeutlichen am folgenden Beispiel insbesondere den Aspekt der mehrfachen Spezialisierung auf Ebene des Vorgehensmodells. Weitere Beispiele für Projektpläne können dem Kapitel 5 entnommen werden.

Spezialisierungen des Produktflusses aus Abbildung 6–10 (a) stellen wir in Abbildung 6–10 (b) und (c) vereinfachend nicht dar. In Abbildung 6–10 (c) stellen wir zudem keine Vorgangsfolgen dar.

(a) Ebene Vorgehensmodell



(b) Projektebene



(c) Projektebene: Weitere Spezialisierungen

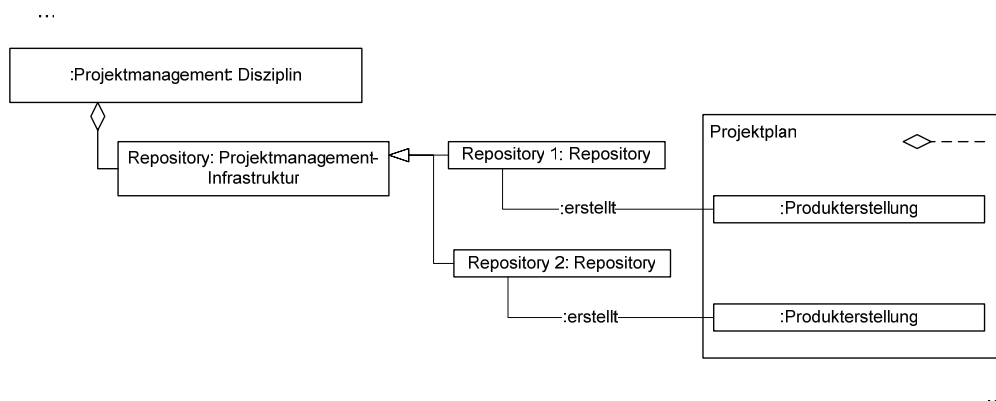


Abbildung 6–10: Anwendung von Vorgehensmodellen: Beispiel

Das Beispiel zeigt die Disziplinen *Projektmanagement* und „*Messung & Analyse*“. In Abbildung 6–10 (a) sind auf Ebene des Vorgehensmodells die Tailoring-Bausteine *Projekt* und *Vorgehens-Kern* dargestellt, die zur Anwendung der Disziplinen *Projektmanagement* und „*Messung & Analyse*“ führen, sowie die in diesen Disziplinen enthaltenen Produkte *Projektmanagement-Infrastruktur* und *Metrikkatalog*. Die *Projektmanagement-Infrastruktur* ist über einen *Produktfluss* mit dem *Metrikkatalog* verbunden. Zudem ist der Tailoring-Baustein *Ablauf* dargestellt, der eine exklusive Auswahl der Planungs-Einheiten *Top-Down-Vorgehen* und „*Nebenläufiges Vorgehen*“ fordert.

Abbildung 6–10 (b) zeigt eine mögliche Spezialisierung auf Projektebene. Dabei werden ausgehend vom Tailoring-Baustein *Projekt* der *Vorgehens-Kern* und damit die Disziplin *Projektmanagement* verpflichtend gewählt. Die optionalen Tailoring-Bausteine „*Messung & Analyse*“ und *Ablauf* werden ebenfalls gewählt. Das „*Nebenläufige Vorgehen*“ wird als Teil einer exklusiven Auswahl gewählt. Entsprechend den Constraints (43), (44) und (45) stellt dies eine gültige projektspezifische Auswahl von Tailoring-Einheiten dar.

Während im Beispiel der *Metrikkatalog* auf Projektebene einfach spezialisiert wird, tritt die *Projektmanagement-Infrastruktur* in den beiden Spezialisierungen *Repository* und *Emailverteiler* auf (vgl. auch Kapitel 5.2.2). Als Folge der Planungs-Einheit „*Nebenläufiges Vorgehen*“ (vgl. auch Abbildung 6–5 (a)) und den Produktflüssen zwischen *Repository* und *Metrikkatalog* beziehungsweise *Emailverteiler* und *Metrikkatalog* stehen die zugehörigen *Produktionserstellungen* über Ende-Ende-Vorgangfolgen in Beziehung. Entsprechend der Definition des *Projektplans* in Constraint (47) sind diese *Produktionserstellungen* und Vorgangfolgen Teile des *Projektplans*. In Abbildung 6–10 (b) notieren wir den *Projektplan* als Klasse.

Abbildung 6–10 (c) zeigt ein Beispiel der Projektebene zu einem späteren Zeitpunkt. Wie wir bereits in Kapitel 5.4.1 gezeigt haben, können die Konzepte des Vorgehensmodells weiter spezialisiert werden. Das *Repository* als Teil der *Projektmanagement-Infrastruktur* wird nun im Beispiel auf Projektebene als „*Repository 1*“ und als „*Repository 2*“ spezialisiert. Damit wird das *Repository* selbst aus Sicht des Projektes zu einem Teil des Vorgehensmodells. Als Projektebene betrachten wir die jeweils tiefste Spezialisierung der Konzepte des Vorgehens-Meta-Modells. Entsprechend der Definition des *Projektplans* in Constraint (47) sind nun die *Produktionserstellungen* „*Repository 1*“ und „*Repository 2*“, sowie die zugehörigen Vorgangfolgen Teil des *Projektplans*. Die *Produktionserstellung* des Produktes *Repository* ist dagegen anders als in Abbildung 6–10 (b) kein Teil des *Projektplans* mehr.

Der Projektplan enthält die jeweils tiefste Spezialisierung der Elemente des Vorgehens-Meta-Modells. Enthalten sind dabei die Elemente des Vorgehensmodells, die projektspezifisch im Rahmen des Tailorings ausgewählt wurden. **Zusammenfassung**

6.5 Methodisches Vorgehen der Erstellung von Vorgehensmodellen

In diesem Kapitel geben wir ein methodisches Vorgehen für die Erstellung von Vorgehensmodellen an, die die Ableitung von Projektplänen entsprechend dem Konzept der vorliegenden Arbeit ermöglichen. Unter einem methodischen Vorgehen beziehungsweise einer Methodik verstehen wir nach [Chroust92] eine systematische Handlungsvorschrift, die folgerichtig und zweckmäßig ist, zur Lösung einer bestimmten Klasse von Problemen. Dabei beziehen wir uns auf die Konzepte des Vorgehens-Meta-Modells zur Beschreibung von Vorgehensmodellen. Wir beschreiben also den Einsatz des Vorgehens-Meta-Modells.

Ziel ist nicht die Beschreibung einer „Methodik im Kleinen“. Methodische Aspekte des Einsatzes der einzelnen Konzepte haben wir in den vorangehenden Kapiteln bereits anhand von Beispielen betrachtet. Zielsetzung dieses Kapitels ist vielmehr eine Methodik, die sich auf der Granularität einer Projektplanung eines Projektes für die Erstellung eines Vorgehensmodells bewegt. Im Fokus steht also die Planung der arbeitsteiligen Erstellung eines Vorgehensmodells. Die Erstellung eines Vorgehensmodells im Team ist wünschenswert, um einen breiten Erfahrungsschatz einzubeziehen und die Akzeptanz des Vorgehensmodells zu fördern.

Im Kontext der Arbeit betrachtet bedeutet Methodik die Formulierung eines Vorgehensmodells als Abstraktion eines Projektplans. Es soll also ein Vorgehensmodell für die Erstellung eines Vorgehensmodells angegeben werden, das sich an den Konzepten des in der Arbeit definierten Vorgehens-Meta-Modells orientiert. Dieses Vorgehensmodell ließe sich in der Sprache des konzipierten Vorgehens-Meta-Modells selbst beschreiben. Wir nehmen allerdings in diesem Kapitel von der Anwendung des Vorgehens-Meta-Modells zur Beschreibung der Methodik Abstand, da in diesem Kapitel das Ziel keine präzise Modellierung ist. Die präzise Modellierung in der Sprache des Vorgehens-Meta-Modells würde sich für die Einspeisung in ein geeignetes Werkzeug eignen (vgl. Kapitel 6.7). Hier bemühen wir uns allerdings um eine hinsichtlich „Lesefreundlichkeit“ optimierte Darstellung.

Modulare Bearbeitungs- Einheiten

In einem arbeitsteiligen Erstellungsprozess eines Vorgehensmodells ist wie in jedem Projekt die Festlegung klarer Zuständigkeiten von Bedeutung. Arbeitspakete müssen geeignet definiert werden. Erforderlich ist ein Konzept für eigenständig verantwortete Bearbeitungseinheiten. Bearbeitungseinheiten sollten dabei ein hohes Maß an inhaltlicher Zusammengehörigkeit und Unabhängigkeit untereinander aufweisen, um einem möglichst effizienten Bearbeitungsprozess dienlich zu sein. Wir stützen die Definition modularer Bearbeitungseinheiten auf die Konzepte zur inhaltlichen Strukturierung des Vorgehensmodells ab.

Disziplinen, Planungs-Einheiten und Strukturierungen (vgl. Kapitel 6.1) können als Bearbeitungseinheiten definiert werden. Disziplinen kapseln Produkte, Planungs-Einheiten kapseln Planungselemente. Strukturierungen sind Zusammenfassungen von Disziplinen, Planungs-Einheiten beziehungsweise anderen Strukturierungen unter inhaltlichen Gesichtspunkten. Die Ebene der Disziplinen beziehungsweise Planungs-Einheiten stellt neben der Strukturierung der Inhalte die Basis für die Definition von Tailoring-Einheiten dar. Damit kann diese Ebene für die Definition von Bearbeitungseinheiten zu feingranular sein. Als Bearbeitungseinheiten können also auch Strukturierungen dienen. Die Granularität der Bearbeitungseinheiten muss im Erstellungsprozess geeignet gewählt werden.

Trotz Referenzierung von Produkten beziehungsweise Planungselementen in mehreren Disziplinen beziehungsweise Planungs-Einheiten, sind Produkte und Planungselemente über eine Aggregations-Beziehung genau einer Disziplin beziehungsweise Planungs-Einheit zugeordnet. Disziplinen und Planungs-Einheiten wiederum werden höchstens von einer Strukturierung, und Strukturierungen selbst ebenfalls höchstens von einer Strukturierung aggregiert. Produkte und Planungselemente finden sich also über Aggregations-Beziehungen eindeutig in einer Bearbeitungseinheit wieder.

Die Idee einer Aufteilung in modulare Bearbeitungs-Einheiten, die eine entkoppelte Bearbeitung ermöglichen, lässt sich im Kontext von Vorgehensmodellen nur bedingt verwirklichen. Die Zusammenhänge der Elemente eines Vorgehensmodells sind mannigfaltig, wie sich unter anderem am Beispiel des V-Modells XT ablesen lässt. Das Finden einer geeigneten inhaltlichen Strukturierung, die ein möglichst hohes Maß an Unabhängigkeit aufweist, ist nicht trivial.

Die inhaltliche Strukturierung eines Vorgehensmodells weist meist zahlreiche Überschneidungen auf. Diese ergeben sich in Form von Elementen, die mehreren Elementen zur Strukturierung zugeordnet werden können. Das Modell der inhaltlichen Strukturierung sieht die Kennzeichnung „inhaltlicher Überlappungen“ vor. Die Details dieser inhaltlichen Überlappungen definieren die Sichtbarkeit bestimmter Elemente bei der Betrachtung durch den Anwender. Im Kontext des Erstellungsprozesses leitet sich aus einer inhaltlichen Überlappung die Notwendigkeit der Zusammenarbeit bei der Erstellung bestimmter Bearbeitungs-Einheiten ab. Lokalisieren sich beispielsweise Produkte, die in einer Aggregations-Beziehung stehen, in unterschiedlichen Disziplinen (vgl. beispielsweise Abbildung 6–3 (b)) und damit potentiell in unterschiedlichen Bearbeitungs-Einheiten, so ist im Rahmen des Erstellungsprozesses bezüglich dieser Produkte eine Abstimmung der jeweiligen Bearbeiter erforderlich. Zusammenarbeit ist für alle Produkte erforderlich, die Teil einer inhaltlichen Überlappung sind.

Planungs-Einheiten können als querschnittlich gekennzeichnet werden, wenn sie potentiell mit allen Disziplinen des Vorgehensmodells in Beziehung stehen. Beispielsweise Planungs-Einheiten zur Strukturierung des Projektablaufs können querschnittlich sein (vgl. beispielsweise Abbildung 6–4 (b)). Ist eine Planungs-Einheit als querschnittlich gekennzeichnet, entfällt die explizite Kennzeichnung inhaltlicher Überlappungen. Aus Sicht des Erstellungsprozesses bedeutet dies, dass die Erstellung einer querschnittlichen Bearbeitungs-Einheit eine Zusammenarbeit beziehungsweise Abstimmung mit allen Bearbeitungs-Einheiten erforderlich macht, die Disziplinen beinhalten.

Die Nutzung der Konzepte des Modells zur inhaltlichen Strukturierung für die Definition modularer Bearbeitungs-Einheiten eignet sich für die eindeutige Zuordnung von Klassen des Produkt- und des Planungsmodells zu Bearbeitern. Die Elemente des Vorgehensmodells, die im Vorgehens-Meta-Modell als Assoziationen modelliert sind, sind dagegen nicht unmittelbar den Bearbeitungs-Einheiten zugeordnet. Im Vorgehens-Meta-Modell sind als Assoziationen unter anderem die Konzepte Produktabhängigkeit, Ablauf, Aggregation zwischen Produkten sowie Aggregation zwischen Planungselementen definiert. Zudem können im Vorgehensmodell unmittelbare Spezialisierungen des Konzeptes Assoziation der Modellierungstechnik auftreten (vgl. Kapitel 5.4.3).

Im Kontext des Erstellungsprozesses verstehen wir die Assoziationen, die Produkte beziehungsweise Planungselemente derselben Bearbeitungs-Einheit in Beziehung setzen, als dieser Bearbeitungs-Einheit zugehörig. Damit liegen diese Assoziationen im Verantwortungsbereich des jeweiligen Bearbeiters der Bearbeitungs-Einheit. Übergreifende Assoziationen ordnen wir nicht den Bearbeitungs-Einheiten zu. Vielmehr definieren wir für den Erstellungsprozess zusätzliche Arbeitspakete, die die übergreifende Strukturierung des Vorgehensmodells zum Ziel haben, die sich nicht in Bearbeitungs-Einheiten lokalisiert. Alternativ ließe sich entsprechend dem V-Modell XT eine Assoziation zwischen Bearbeitungs-Einheiten einer dieser Bearbeitungs-Einheiten zuordnen. Der Nachteil der Bearbeitung durch einen Bearbeiter bestünde bei ungenügender Abstimmung in einer mangelnden Einheitlichkeit dieser Strukturierungen. Durch ein „querschnittliches“ Arbeitspaket in der Hand eines Bearbeiters kann diese Einheitlichkeit erreicht werden.

In Abbildung 6–11 geben wir eine Methodik auf der Granularität eines Projektplans für die Erstellung eines Vorgehensmodells in Form einzuplanender Schritte mit einer kausalen Ordnung an. Während sich die hier beschriebene Methodik auf die Erstellung eines Vorgehensmodells von Grund auf bezieht, gehen wir auf die Weiterentwicklung eines bestehenden Vorgehensmodells im Anschluss ein.

**Methodik der
Erstellung**

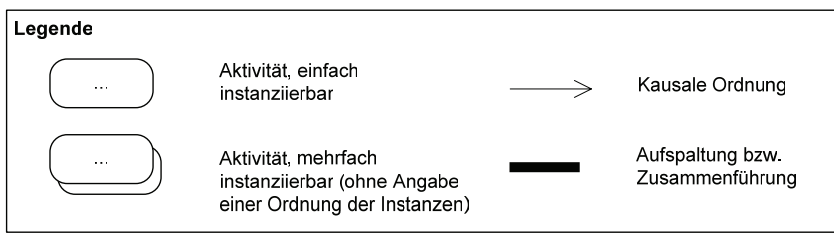
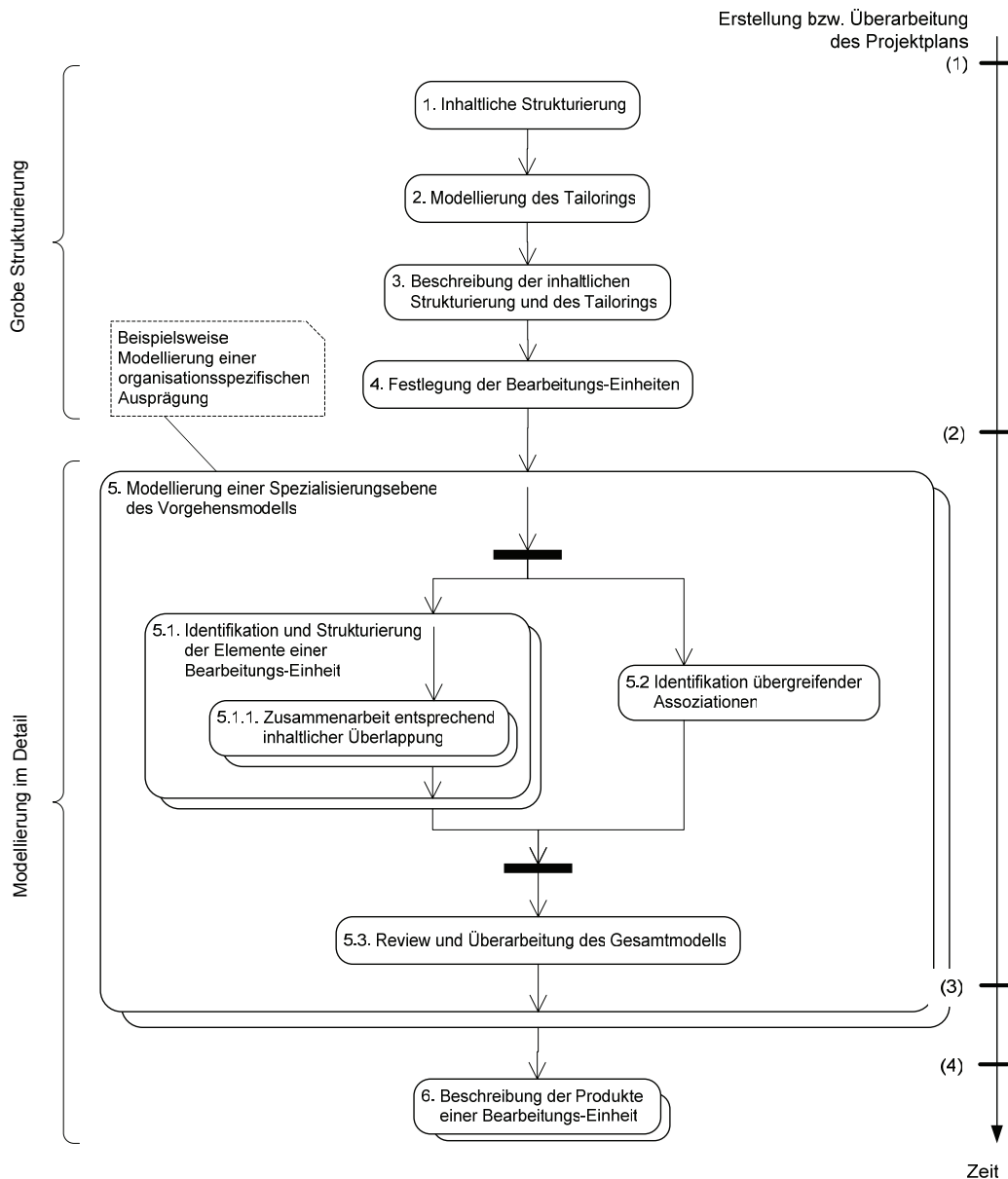


Abbildung 6–11: Methodik der Erstellung von Vorgehensmodellen

Die kausale Ordnung in Abbildung 6–11 spiegelt die Notwendigkeit wider, die Ergebnisse der jeweiligen Quellaktivität zumindest teilweise vor dem Beginn der Zielaktivität zu erarbeiten. Zudem macht die kausale Ordnung zweier Aktivitäten die Konsistenz der jeweiligen

Ergebnisse erforderlich. Die kausale Ordnung der Aktivitäten ist also vergleichbar mit dem Konzept des Produktflusses des Vorgehens-Meta-Modells (vgl. Kapitel 5.3.2).

Die Notation der Abbildung 6–11 lehnt sich an die Notation eines Aktivitätsdiagramms der UML an. Nicht der UML entstammen die Schachtelung von Aktivitäten sowie die Kennzeichnung einfacher beziehungsweise mehrfacher Instanziierung von Aktivitäten. Zudem verstehen wir die Pfeile der Abbildung nicht als Aktivitätsfluss, sondern als kausale Ordnung.

Das methodische Vorgehen gliedert sich in die beiden Phasen „Grobe Strukturierung“ und „Modellierung im Detail“. Diese Aufteilung legt ein top-down-orientiertes Vorgehen nahe. Um eine Planung des Erstellungsprojektes zu ermöglichen, sind beide Phasen sequentiell zu durchlaufen, können sich dabei aber auch überlappen, iterativ ausgeführt werden oder aber beispielsweise um die Erstellung einer prototypischen Umsetzung des Vorgehensmodells erweitert werden. Ein reines Bottom-Up-Vorgehen, das mit der Ausarbeitung der Details startet, erscheint allerdings im Kontext eines zu planenden Projektes nicht geeignet. Die Planung eines arbeitsteiligen Projektes erfordert stets die Festlegung von Bearbeitungseinheiten, also eine grobe Strukturierung des Projektergebnisses. Top-Down-Vorgehen kann als Synonym zum Begriff Planung verstanden werden, während bei einem reinen Bottom-Up-Vorgehen Planung nicht möglich ist.

Ein erster Projektplan, der die Phase der groben Strukturierung zeitlich begrenzt, kann bereits zu Projektbeginn erstellt werden.

Phase: Grobe Strukturierung

1. Inhaltliche Strukturierung

Die Inhalte des zu erstellenden Vorgehensmodells werden auf grober Ebene nach inhaltlichen Gesichtspunkten strukturiert. Dabei werden auch inhaltliche Überlappungen zwischen den strukturierten Inhalten identifiziert. Die inhaltliche Strukturierung ist die Voraussetzung für die Definition von Bearbeitungs-Einheiten und damit für die Erstellung einer detaillierten Planung.

Im Weiteren dient die inhaltliche Strukturierung als vereinbarte Architektur des zu erstellenden Vorgehensmodells. Bei der sich anschließenden detaillierten Modellierung des Vorgehensmodells sind inhaltliche Überlappungen als einschränkende Vorgabe zu verstehen. Die potentiellen Abhängigkeiten der Elemente eines Vorgehensmodells sind mannigfaltig. Die Festlegung der inhaltlichen Überlappungen dient der Eingrenzung der möglichen Inhalte der Disziplinen beziehungsweise Planungs-Einheiten. Die getroffenen Festlegungen sind dabei im Weiteren natürlich nicht unumstößlich, sondern bei Bedarf anpassbar.

2. Modellierung des Tailorings

Das Tailoring-Modell definiert die Möglichkeiten der projektspezifischen Anpassung des Vorgehensmodells. Die Konzeption des Tailoring-Modells kann eine feinere Unterteilung der rein inhaltlich getriebenen Strukturierung des vorangehenden Schrittes erforderlich machen, da sich die Definition von Tailoring-Bausteinen auf diese inhaltliche Strukturierung abstützt. Die feingranularsten Einheiten des Tailorings sind identisch mit den feingranularsten Einheiten der inhaltlichen Strukturierung.

3. Beschreibung der inhaltlichen Strukturierung und des Tailorings

Als Grundlage eines gemeinsamen Verständnisses im weiteren Erstellungsprozess sind die Elemente der inhaltlichen Strukturierung sowie des Tailorings textuell zu beschreiben. Wir erweitern das Vorgehens-Meta-Modell um ein Konzept für beschreibende Texte im folgenden Kapitel (vgl. dazu Kapitel 6.6).

4. Festlegung der Bearbeitungs-Einheiten

Auf Basis der inhaltlichen Strukturierung können Bearbeitungs-Einheiten definiert werden. Dabei handelt es sich um eigenständig verantwortete und bearbeitbare Teile des Vorgehensmodells. Modulare Bearbeitungs-Einheiten können isoliert bearbeitet werden, soweit inhaltliche Überlappungen keine Abstimmung beziehungsweise Zusammenarbeit erforderlich machen.

Die Beschreibung der inhaltlichen Strukturierung dient dabei als Beschreibung der mit einer Bearbeitungs-Einheit verbundenen Aufgabe. Das bereits modellierte und beschriebene Tailoring dient demselben Zweck, nämlich der Positionierung beziehungsweise Abgrenzung der Inhalte der Bearbeitungs-Einheiten. Den Bearbeitern einer inhaltlichen Strukturierung muss die Rolle dieser Strukturierung im Rahmen des Tailorings bekannt sein. Der Zusammenhang zwischen dem Modell zur inhaltlichen Strukturierung und dem Tailoring-Modell, dessen feinste Einheiten den feinsten Einheiten der inhaltlichen Strukturierung entsprechen (vgl. Kapitel 6.2), ist also aus methodischen Gründen wichtig. Neben der inhaltlichen Strukturierung und den definierten inhaltlichen Überlappungen dient also auch das Tailoring-Modell als Vorgabe für den weiteren Erstellungsprozess.

Auf Grundlage festgelegter Bearbeitungs-Einheiten kann der ursprüngliche Projektplan überarbeitet und die folgenden Aktivitäten können detailliert geplant werden.

Phase: Modellierung im Detail

An die „grobe Strukturierung“ schließt sich die „Modellierung im Detail“ an, die auch die Beschreibung der Produkte beinhaltet und damit die eigentliche Erstellung des Vorgehensmodells darstellt.

5. Modellierung einer Spezialisierungsebene des Vorgehensmodells

Das methodische Vorgehen zur Erstellung eines Vorgehensmodells kann die Modellierung mehrerer Spezialisierungs-Ebenen beinhalten. Ein Vorgehensmodell kann mit Hilfe des Spezialisierungskonzeptes in unterschiedlichen Abstraktionen erstellt werden. Neben einem allgemeinen Vorgehensmodell bietet sich die Modellierung domänenspezifischer Ausprägungen beispielsweise für die Entwicklung von Hardware oder Software an. Zudem können organisationsspezifische Ausprägungen des allgemeinen Modells erstellt werden, die die allgemeinen Vorgehensweisen des Modells hinsichtlich der Abläufe in bestimmten Organisationen detaillieren (vgl. Kapitel 2.1.5).

Sind mehrere Spezialisierungs-Ebenen für ein Vorgehensmodell identifiziert, kann die Modellierung dieser Ebenen sequentiell, überlappend oder iterativ geplant und durchgeführt werden. Die Modellierung der abstrakten Ebene vor der detaillierteren Ebene kann als Vorgabe beziehungsweise als Schema für die Modellierung der Details dienen. Dies bietet sich beispielsweise für den Bereich der Systemerstellung des V-Modells XT an, bei der die erzeugenden Produktabhängigkeiten beispielsweise auch in der Softwareentwicklung als Erweiterung der Systemerstellung einem festen Schema folgen (vgl. Kapitel 5.4.2). Andererseits kann auch die Modellierung der Details mit einer anschließenden Generalisierung als sinnvolle Vorgehensweise gelten, da dabei nicht die Gefahr einer abstrakten Modellierung besteht, die nicht geeignet spezialisierbar ist.

5.1. Identifikation und Strukturierung der Elemente einer Bearbeitungs-Einheit

Innerhalb der festgelegten Bearbeitungs-Einheiten werden Produkte beziehungsweise Planungselemente entsprechend den Vorgaben der groben Strukturierung durch die jeweiligen Bearbeiter identifiziert. Assoziationen wie Abläufe und Produktabhängigkeiten werden durch die Bearbeiter modelliert, soweit sie innerhalb der Grenzen der Bearbeitungs-Einheiten angesiedelt sind.

5.1.1. Zusammenarbeit entsprechend inhaltlicher Überlappung

Die festgelegten inhaltlichen Überlappungen zwischen Bearbeitungs-Einheiten machen eine Abstimmung beziehungsweise Zusammenarbeit der jeweiligen Bearbeiter erforderlich. Bearbeitungs-Einheiten, die querschnittliche Planungs-Einheiten beinhalten, erfordern die Abstimmung mit den Bearbeitern aller Disziplinen des Vorgehensmodells. In der Planung des Projektes sind für inhaltlich überlappende und querschnittliche Bearbeitungs-Einheiten beispielsweise Workshops vorzusehen.

Entsprechend dem Konzept der inhaltlichen Produktabhängigkeiten des Vorgehens-Meta-Modells (vgl. Kapitel 5.3.2) können inhaltliche Überlappungen zu zeitlichen Überlappungen der Erstellung der jeweiligen Bearbeitungs-Einheiten führen. Eine Reihenfolge der „Aktivitäten 5.1“ leitet sich daraus im Allgemeinen allerdings nicht ab.

5.2. Identifikation übergreifender Assoziationen

Die Identifikation übergreifender Assoziationen wie Produktabhängigkeiten oder Abläufe, die sich nicht in Bearbeitungs-Einheiten lokalisieren, muss als querschnittliche Aufgabe bewältigt werden. Während die Modellierung von Assoziationen zwischen inhaltlich überlappenden Bearbeitungs-Einheiten auch bilateral durch die jeweiligen Bearbeiter erfolgen kann, muss die Modellierung aller weiteren Assoziationen im Rahmen eines eigenständigen Arbeitspaketes erfolgen.

Die Aktivität zur Identifikation übergreifender Assoziationen ordnen wir parallel zur Erstellung der Bearbeitungs-Einheiten an, da eine laufende Interaktion mit den Bearbeitern der Bearbeitungs-Einheiten erfolgen soll. Die Aktivität ist zudem der Einheitlichkeit der Modellierung zuträglich.

5.3. Review und Überarbeitung des Gesamtmodells

Im Anschluss an die Identifikation und Strukturierung der Elemente des Vorgehensmodells einer Spezialisierungsebene schließt sich eine Überprüfung hinsichtlich Stimmigkeit und Einheitlichkeit der gesamten Modellierung an. Dabei wird auch die Zugehörigkeit zur jeweiligen Spezialisierungs-Ebene überprüft. Redundanzen sind ein Indiz für eine erforderliche Generalisierung (vgl. Kapitel 5.4.2) beziehungsweise die Modellierung einer weiteren Abstraktionsebene und damit einen weiteren Durchlauf durch „Aktivität 5“.

Nach Abschluss der Modellierung einer Spezialisierungsebene erfolgt eine Überarbeitung des Projektplans. Ist die Strukturierung des Vorgehensmodells in allen Spezialisierungsebenen abgeschlossen, kann die detaillierte Planung der sich anschließenden Aktivitäten zur Erstellung von Beschreibungen der Produkte erfolgen.

6. Beschreibung der Produkte einer Bearbeitungs-Einheit

Im Anschluss an die detaillierte Modellierung des Vorgehensmodells werden beschreibende Texte zu den Produkten erstellt (vgl. dazu Kapitel 6.6). Die Referenzierung von Produkten im Vorgehens-Meta-Modell gibt dabei die Notwendigkeit zur Zusammenarbeit zwischen Bearbeitern unterschiedlicher Bearbeitungs-Einheiten an.

Wir positionieren „Aktivität 6“ nach der Strukturierung des gesamten Vorgehensmodells, da Beschreibungen in Form von Text stets die Struktur reflektieren, also bei einer Überarbeitung der Struktur in der Regel angepasst werden müssen. Umgekehrt sollte die Erstellung von Texten dagegen in der Regel keine Anpassungen der Struktur erforderlich machen.

Vergleichbar Software-Systemen sind Vorgehensmodelle oft langlebige Systeme, da eine komplette Neuerstellung mit einem hohen Aufwand verbunden ist. Die Weiterentwicklung beziehungsweise Anpassung eines Vorgehensmodells ist also verglichen mit der Neuerstellung der häufigere Fall. Für die Akzeptanz eines Vorgehensmodells ist zudem die regelmäßige Berücksichtigung und Einarbeitung von Feedback aus der Anwendung des Vorge-

Weiterentwicklungsprozess

hensmodells von Bedeutung. Entsprechend der Idee des Prozessmuster-Ansatzes sollte ein Vorgehensmodell „lebendig“ sein (vgl. Kapitel 2.1.3, Abbildung 2–7).

Der Ansatz der Arbeit ermöglicht die Aufnahme von Vorgängen in den Projektplan, die nicht dem Vorgehensmodell entstammen. Die Projektebene kann unmittelbare Spezialisierungen des Konzeptes Vorgang des Vorgehens-Meta-Modells enthalten. Diese Vorgänge können als Kandidaten für die Erweiterung des Vorgehensmodells im Rahmen des Weiterentwicklungsprozesses dienen.

Überarbeitungen eines Vorgehensmodells können sich auch im Weiterentwicklungsprozess an der inhaltlichen Strukturierung des Vorgehensmodell und definierten Abhängigkeiten orientieren. Änderungen lokalisieren sich also entweder in den Bearbeitungs-Einheiten des Erstellungsprozesses oder aber inhaltliche Überlappungen beziehungsweise Querschnittlichkeit von Planungs-Einheiten ermöglichen die Identifizierung zusätzlicher Auswirkungen einer Änderung auf andere Teile des Vorgehensmodells.

6.6 Modellierung von Texten

In diesem Kapitel beschäftigen wir uns mit der „Modellierung“ von Texten. Ein Vorgehensmodell dient als Anleitung für die Durchführung von Projekten. Es wird dabei nicht so präzise modelliert, dass eine Anwendung ohne beschreibenden Text möglich wäre. Textuelle Beschreibungen dienen im Kontext der Arbeit als Orientierungshilfe für den Projektleiter bei der Erstellung einer Planung. Interessant sind Texte insbesondere im Kontext des Tailorings, da dieser Aspekt die „Modellierung“ von Texten erforderlich macht.

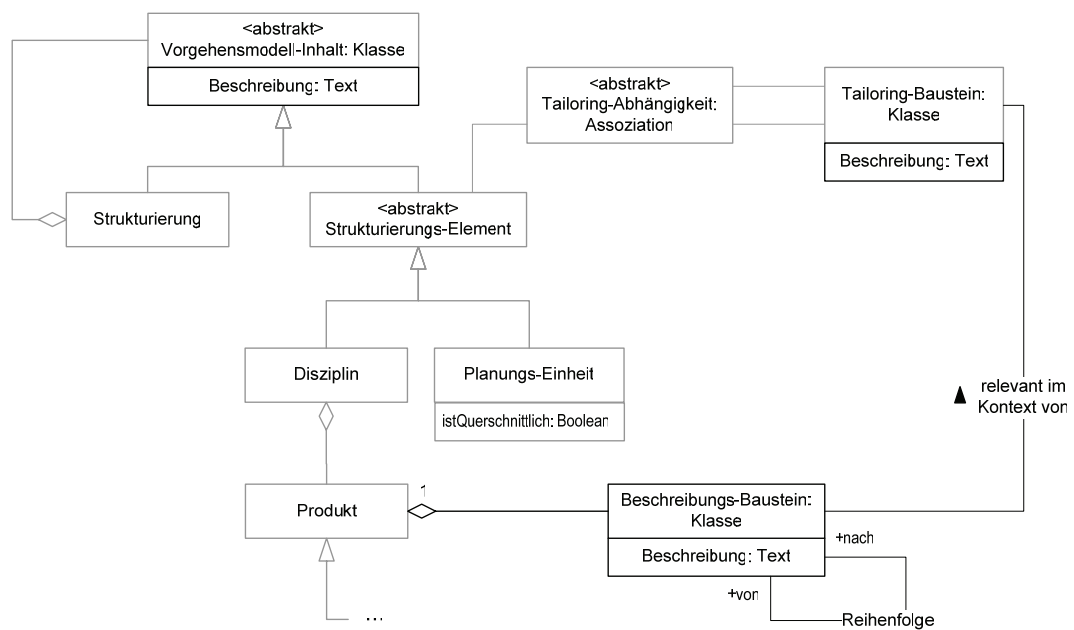


Abbildung 6–12: Modellierung von Texten – Ebene Vorgehens-Meta-Modell

Wir stellen in Abbildung 6–12 eine Erweiterung des Vorgehens-Meta-Modells um ein Modell für Texte dar. Beschreibende Texte sind im Vorgehensmodell in Bezug auf die Elemente *Tailoring-Baustein*, *Strukturierung*, *Disziplin*, *Planungs-Einheit* und *Produkt* relevant. Für Tailoring-Bausteine muss anhand des Textes entschieden werden können, ob ein Tailoring-Baustein im Projekt gewählt werden soll oder nicht. Die Elemente Strukturierung, Disziplin und Planungs-Einheit zur inhaltlichen Strukturierung des Vorgehensmodells ermöglichen durch eine abstrakte Beschreibung ihrer Bestandteile die „Navigation“ des Lesers im Vorgehensmodell. Für Produkte schließlich muss ein Vorgehensmodell eine klare Beschreibung des Inhalts eines Produktexemplars liefern. Eine Beschreibung ist dabei für alle Ausprägungen von Produkten vorgesehen, also für Dokumente, Kapitel, Realisierungselemente, Werkzeuge und „strukturierte Inhalte“.

Unter der Beschreibung eines Produktes verstehen wir eine Beschreibung des Inhalts eines Produktexemplars. Die Beschreibung der Produkte beschränken wir dabei auf eine Granularität, die für die Erstellung einer Projektplanung hinreichend ist. Beispielsweise ist die Beschreibung einer Methodik zur szenarienbasierten Architekturbewertung nicht als planungsrelevant anzusehen. Aus Sicht der Planung ist die Existenz eines entsprechenden Vorgangs zur Bewertung der Architektur ohne weitergehende Beschreibung ausreichend. Im Allgemeinen ist es natürlich möglich, in einem Vorgehensmodell Beschreibungen für die Zielgruppen Projektleiter und beispielsweise Entwickler zu integrieren. Allerdings halten wir eine klare

Kennzeichnung der jeweiligen Zielgruppe eines Textes für sinnvoll, da ein Projektleiter eine andere Sicht beispielsweise auf ein Architekturdokument als ein Entwickler hat.

Wie im Vorgehens-Meta-Modell in Abbildung 6–12 dargestellt sind Beschreibungen für Tailoring-Bausteine, Strukturierungen, Disziplinen und Planungs-Einheiten Attribute des Typs Text. Für Produkte erweist sich die Modellierung eines einfachen Text-Attributes als nicht ausreichend. Die Beschreibung zu einem Produkt kann über die Grenzen des Produktes hinaus Querverweise auf andere Produkte enthalten und sollte entsprechende Querverweise auch enthalten, um die Zusammenhänge der Produkte zu beschreiben.

In einem projektspezifisch angepassten Vorgehensmodell können sich Querverweise auf Produkte beziehen, die projektspezifisch nicht enthalten sind. Aus diesem Grund modellieren wir *Beschreibungs-Bausteine*, die jeweils ein Attribut *Beschreibung* vom Typ *Text* enthalten. Ein Beschreibungs-Baustein ist eindeutig einem Produkt zugeordnet. Ein Produkt kann mehrere Beschreibungs-Bausteine enthalten. Um den Lesefluss des beschreibenden Textes eines Produktes zu gewährleisten, ist die Angabe einer Reihenfolge der Beschreibungs-Bausteine eines Produktes erforderlich. Den Lesefluss modellieren wir in Form der Assoziation *Reihenfolge* zwischen Beschreibungs-Bausteinen.

Stellt ein Beschreibungs-Baustein eines Produktes den Zusammenhang zu einem anderen Produkt her, ist dieser Beschreibungs-Baustein nur in einem projektspezifisch angepassten Vorgehensmodell relevant, das beide Produkte beinhaltet. Für Beschreibungs-Bausteine kennzeichnen wir also ihre „*Relevanz im Kontext von*“ Tailoring-Bausteinen. Je nach projektspezifischer Auswahl von Tailoring-Bausteinen kann die Beschreibung eines Produktes also variieren.

Im V-Modell XT ist für ein Produkt eine abstrakte Beschreibung der enthaltenen Themen erforderlich, da die projektspezifisch enthaltenen Themen je nach Auswahl von Vorgehensbausteinen variieren können (vgl. Kapitel 3.1.4). Entsprechend der Modellierung in Abbildung 6–12 kann sich die zusammengesetzte Beschreibung eines Dokumentes direkt auf die Kapitel dieses Dokumentes beziehen, auch wenn diese projektspezifisch entfallen können. Abstrakte Beschreibungen sind damit allein für das Modell zur inhaltlichen Strukturierung erforderlich, in diesem Kontext allerdings nicht von Nachteil.

Beschreibung weiterer Elemente

Anders als im V-Modell XT knüpfen wir Beschreibungen nicht an ein Aktivitätenmodell beziehungsweise an die Vorgänge zur Produkterstellung des Vorgehens-Meta-Modells. Dies würde aufgrund des unmittelbaren Zusammenhangs zwischen genau einem Produkt und genau einer Produkterstellung analog der Umsetzung des V-Modells XT zu redundanten Texten führen (vgl. Kapitel 3.4.2).

Sinnvoll wäre die Beschreibung durchzuführender Aktivitäten allein in einem Vorgehensmodell, das auf Basis eines Produktmodells unterschiedliche methodische Vorgehensweisen enthielte, also die Zuordnung unterschiedlicher Aktivitäten zu einem Produkt erlauben würde. Im Kontext des Aufbaus einer Prozessmuster-Bibliothek haben sich alternative methodische Vorgehensweisen für die Erstellung einer bestimmten Menge von Produkten nicht als praxisrelevant erwiesen (vgl. auch [GMP+03] und [BBE+03]). Alternative Vorgehensweisen bewegen sich meist auf die Ebene von Projektdurchführungsstrategien. Entwicklung nach Wasserfall oder inkrementelle Entwicklung sind Beispiele alternativer Projektdurchführungsstrategien. Auf der Ebene der Beschreibung von Aktivitäten zur Erstellung einer festgelegten Menge von Produkten treten alternative Vorgehensweisen dagegen kaum auf.

Wir verfolgen in der Arbeit analog dem V-Modell XT einen produktzentrierten Ansatz. Dabei werden Produkte detailliert beschrieben. Das methodische Vorgehen zur Erstellung eines Produktes ist damit festgelegt und kann unmittelbar in die Produktbeschreibung integriert werden. Beschreibende Texte für Produkterstellungen sind damit nicht erforderlich. Eine Beschreibung alternativer Vorgehensweisen erfordert entsprechend diesem Ansatz also die Modellierung unterschiedlicher Produkte.

Auch für Projektabschnitte zur Strukturierung des Projektablaufs und Produktbearbeitungen sehen wir keine beschreibenden Texte vor, da diese nicht als Anleitung dienen, sondern allein der Strukturierung des Projektplans. Ist eine Beschreibung einer Produktbearbeitung

für das Verständnis bezüglich der Einplanung der Produktbearbeitung erforderlich, kann diese Beschreibung im Kontext der Beschreibung der Planungs-Einheit gegeben werden, die die Produktbearbeitung beinhaltet. Mit der Assoziation Produktabhängigkeit verbinden wir ebenfalls keine textuelle Beschreibung. Die Produktabhängigkeiten des Vorgehens-Meta-Modells verwenden wir zur Modellierung der Konsistenzsicherung zweier Produkte sowie der damit verbundenen Abläufe aus Sicht der Projektplanung (vgl. Kapitel 5.3.2). Beschreibungen der Zusammenhänge mehrerer Produkte platzieren wir dagegen in den Beschreibungen zu den jeweiligen Produkten.

Spezialisierung dient der Abgrenzung der Ebenen Vorgehens-Meta-Modell, Vorgehensmodell und Projektebene. Das Vorgehensmodell kann zudem Spezialisierung beinhalten. Für Produkte legen wir fest, dass Spezialisierungen die Beschreibung eines Produktes erweitern, nicht aber „überschreiben“ können. Dies erscheint für Konkretisierungen auf Ebene des Vorgehensmodells beispielsweise im Rahmen einer organisationsspezifischen Anpassung sinnvoll zu sein, weil die Anforderungen des abstrakten Modells dabei im Allgemeinen nicht reduziert werden dürfen. Beschreibungen auf abstrakter Ebene sollten zudem so abstrakt sein, dass eine Spezialisierung nicht zu Widersprüchen führt. Die Beschreibung eines Produktes besteht also aus der Konkatenation der Beschreibung der Generalisierung des Produktes mit den konkatenierten Beschreibungs-Bausteinen des Produktes selbst in der modellierten Reihenfolge. Auf eine Formalisierung verzichten wir an dieser Stelle.

**Texte und
Spezialisierung**

6.7 Werkzeugkonzept

In diesem Kapitel geben wir ein Konzept einer Werkzeugunterstützung für die Erstellung und Anwendung von Vorgehensmodellen im Projekt an.

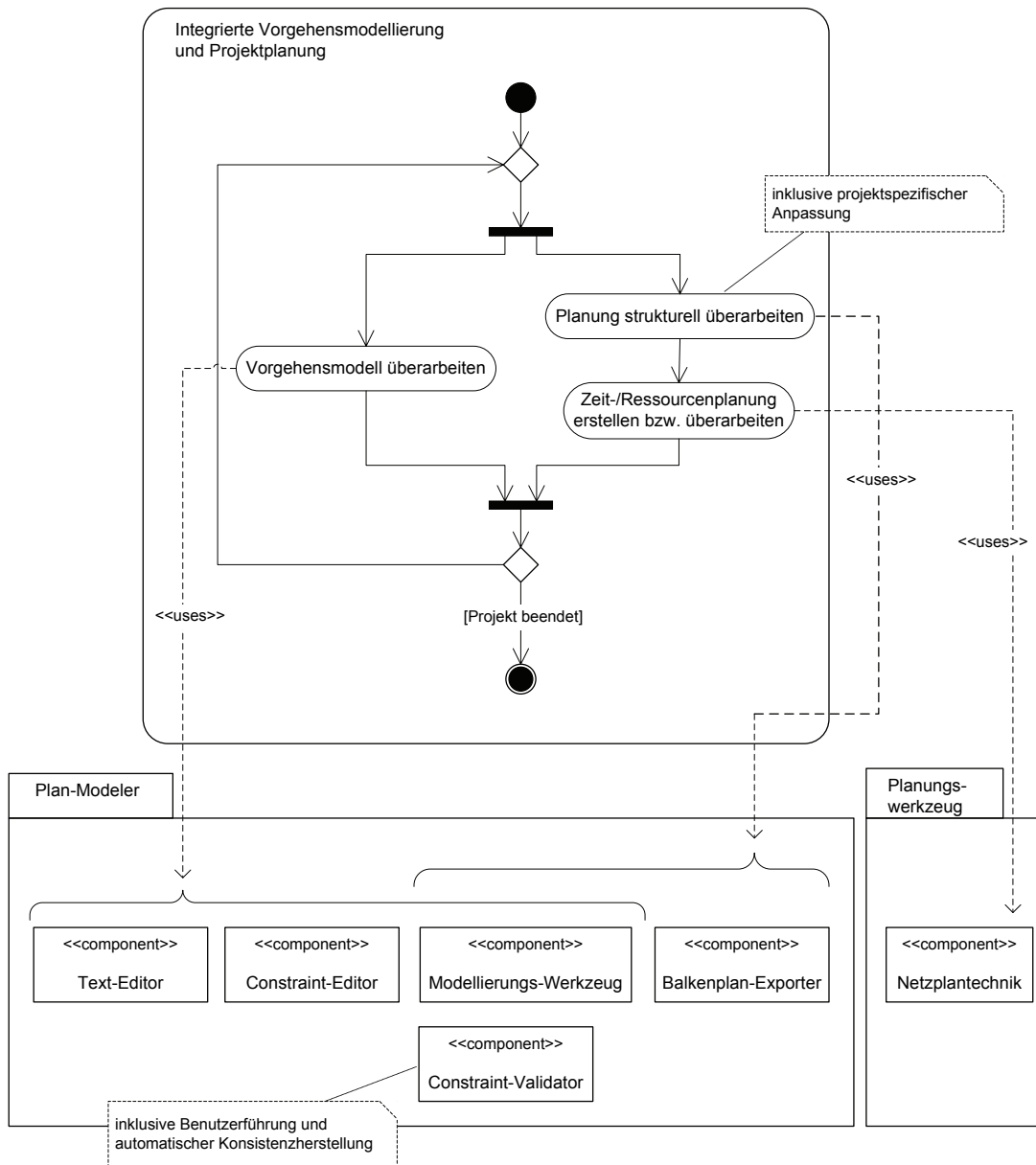


Abbildung 6–13: Konzept einer Werkzeugunterstützung

Die Konzepte der vorliegenden Arbeit eignen sich nicht zur Abbildung in einen struktur-basierten Editor wie dem V-Modell-Editor (vgl. [4EverEdit]). Während dieser Editor für die Beschreibung einer Instanz eines Vorgehens-Meta-Modells geeignet ist, unterstützt er nicht die Beschreibung mehrerer Spezialisierungs-Ebenen. Eine Werkzeugunterstützung für die

Projektplanung entsprechend dem Konzept der vorliegenden Arbeit muss allerdings die Beschreibung mehrerer Spezialisierungs-Ebenen im Vorgehensmodell unterstützen.

Abbildung 6–13 zeigt ein Werkzeug-Konzept in Form eines Aktivitätsdiagramms für die integrierte Vorgehensmodellierung und Projektplanung. Zudem ist eine konzeptionelle Aufteilung der Werkzeugunterstützung *Plan-Modeler* in Komponenten dargestellt.

Um der Idee eines lebendigen Vorgehensmodells (vgl. auch Kapitel 6.5 beziehungsweise Kapitel 2.1.3, Abbildung 2–7) gerecht zu werden integrieren wir die Erstellung eines Vorgehensmodells und die Planung auf Basis dieses Vorgehensmodells in der Werkzeugumgebung *Plan-Modeler*. Beide Vorgänge werden in Abbildung 6–13 iterativ und nebenläufig dargestellt. Ein Vorgehensmodell kann diesem Ablauf entsprechend projektbegleitend angepasst beziehungsweise erweitert werden.

Die initiale Erstellung des Vorgehensmodells beziehungsweise des Projektplans bezeichnen wir in Abbildung 6–13 auch als Überarbeitung, da mit dem Vorgehens-Meta-Modell bereits ein wenn auch sehr abstraktes Vorgehensmodell beziehungsweise ein sehr abstrakter Projektplan zur Verfügung stehen (vgl. Kapitel 6.4).

Während sich Zeit- und Ressourcenplanung nach den Techniken der Netzplantechnik auf ein gängiges Werkzeug zur Projektplanung abstützen können, werden Vorgehensmodellierung und strukturelle Planung durch den *Plan-Modeler* realisiert. Die Komponenten des *Plan-Modelers* spiegeln dessen wesentliche Funktionalitäten wider.

Zentrale Komponente sowohl für die Erstellung des Vorgehensmodells als auch die Planung ist ein *Modellierungs-Werkzeug*, das die Modellierung entsprechend der in Kapitel 4 definierten Modellierungstechnik basierend auf Klassen unterstützt. Anders als beispielsweise mit dem V-Modell-Editor (vgl. [4EverEdit]) kann mit dem *Plan-Modeler* mehr als eine Spezialisierungsebene modelliert werden. Für die Realisierung bietet sich die Verwendung eines graphischen UML-Editors an, der um spezifische Aspekte der in Kapitel 4 definierten Modellierungstechnik zu erweitern ist. Modellierungswerkzeuge wie [ArgoUML] unterstützen beispielsweise nicht die Spezialisierung von Assoziationen, obwohl dieses Konzept Bestandteil der UML ist (vgl. [OMG03a]).

Ebenfalls zentral ist die Integration eines *Constraint-Validators*, der bei der Erstellung von Vorgehensmodell beziehungsweise Planung Constraints überprüft und Inkonsistenzen aufdeckt. Der *Constraint-Validator* kann um eine geeignete Benutzerführung beziehungsweise einen Mechanismus zur automatischen Herstellung bestimmter Konsistenzen erweitert werden. Beispielsweise die Existenz eines Vorgangs zur Produkterstellung zu jedem Produktexemplar erfordert keine Benutzerinteraktion, kann also automatisiert werden (vgl. Kapitel 5.2.3, Constraint (17)).

Für die Modellierung von Vorgehensmodellen ist ein *Constraint-Editor* erforderlich, da dem Prozessingenieur auch die Modellierung von Constraints obliegt (vgl. beispielsweise Kapitel 5.2.4 und 5.3.1). Ein *Text-Editor* schließlich dient der Beschreibung der inhaltlichen Strukturierung, des Tailorings und der Produkte des Vorgehensmodells.

Planung bedeutet im *Plan-Modeler* die Spezialisierung der Konzepte des Vorgehensmodells. Die projektspezifische Anpassung stellt sich ebenfalls als Spezialisierung dar. Entsprechend den Vorgaben der Tailoring-Abhängigkeiten werden bestimmte Tailoring-Bausteine spezialisiert (vgl. Kapitel 6.4). Eine Benutzerführung als Bestandteil des *Constraint-Validators* kann dabei die für ein Projekt nicht benötigten Produkte beziehungsweise Planungselemente des Vorgehensmodells ausblenden.

Die Kopplung des *Plan-Modelers* an das Planungswerkzeug zur Zeit- und Ressourcenplanung realisiert die Komponente *Balkenplan-Exporter*. Diese extrahiert die modellierten Vorgänge der Projektebene. Ein mehrfacher Export von Vorgängen in das verwendete Planungswerkzeug sollte dabei nicht zum Verlust bereits geplanter Termine führen („round-tripping“).

Nicht jedes Planungswerkzeug eignet sich zur Unterstützung der Konzepte der Arbeit. In Microsoft Project (vgl. [MSProj03]) können beispielsweise allein hierarchische Strukturierung

gen von Vorgängen angelegt werden. Die Aggregation eines Vorgangs durch zwei übergeordnete Vorgänge, die wir als sinnvoll ansehen (vgl. Kapitel 5.2), ist in vielen gängigen Planungswerkzeugen nicht möglich.

7 Zusammenfassung und Ergebnisse

Studien zufolge müssen viele Softwareentwicklungs-Projekte heute in Bezug auf Zeit, Kosten und Qualität als nur eingeschränkt erfolgreich eingestuft werden. Der Einsatz geeigneter Vorgehensmodelle kann wesentlich zum Erfolg eines Projektes beitragen. Kernthema der Arbeit ist die Konzeption eines Vorgehensmodells als Abstraktion bewährter Projektpläne, mit dem Ziel, aus einem entsprechenden Vorgehensmodell auf einfache Weise einen Projektplan ableiten zu können. Im Zentrum steht die präzise Definition von Mechanismen und damit die Spezifikation einer Werkzeugunterstützung für die projektspezifische Anpassung eines Vorgehensmodells sowie die Ableitung eines Projektplans aus einem Vorgehensmodell. Vorgehensmodelle werden also eigens für den Anwendungsfall Projektplanung ausgelegt.

Den Ausgangspunkt unserer Betrachtungen bildet die Analyse der Eignung des V-Modells XT für diesen Anwendungsfall. Die aufgedeckten Schwachstellen sind dabei vor dem Hintergrund unterschiedlicher Zielsetzungen des V-Modells XT und der vorliegenden Arbeit zu sehen. Während das V-Modell vorrangig den Anspruch eines übersichtlich strukturierten Nachschlagewerkes hat, steht in der vorliegenden Arbeit die Spezifikation einer Werkzeugunterstützung für die Projektplanung im Vordergrund.

**V-Modell XT
als Ausgangspunkt**

Das Ablaufkonzept des V-Modells ist im Vergleich zu dem hier konzipierten Ablaufkonzept als abstrakter einzustufen. Aus Sicht der Erstellung des V-Modells stellt die abstrakte Beschreibung von Abläufen eine bewusste Entscheidung dar. Ziel ist dabei, durch Abstraktion eine breite Anwendbarkeit des V-Modells sicherzustellen, sowie die Erstellung und Pflege des V-Modells mit vertretbarem Aufwand betreiben zu können. Nicht alle Arten von Abläufen sind also mit dem Meta-Modell des V-Modells beschreibbar, wie wir anhand von Beispielen gezeigt haben. Das Ablaufkonzept des V-Modells stellt sich somit als unvorteilhaft für die einfache Anwendung im Rahmen der Projektplanung beziehungsweise die Unterstützbarkeit durch ein Planungswerkzeug dar.

Das V-Modell XT verfolgt die Idee eines Produktmodells mit explizit formulierten Produktabhängigkeiten. Produktabhängigkeiten werden allerdings nicht präzise formuliert, sondern ebenfalls gemäß einer Reihe von Abstraktionsprinzipien. Produktabhängigkeiten im V-Modell können als Aggregationen der genauen Zusammenhänge der Produkte verstanden werden. Einschränkungen der projektspezifischen Produktstruktur, beispielsweise in Form von Multiplizitäten, können im V-Modell nicht formuliert werden. Für den Leser erscheinen diese Abstraktionen angebracht, da sie der Übersichtlichkeit dienen. Aus Sicht einer Werkzeugunterstützung für die Projektplanung ist die geringe Präzision des Produktmodells allerdings nicht hilfreich beziehungsweise setzt den Möglichkeiten einer Werkzeugunterstützung enge Grenzen.

Das V-Modell XT ist auf genau einer Abstraktionsebene angesiedelt, da ein Konzept zur Modellierung von Spezialisierung fehlt. Aufgrund der Anforderungen der Praxis enthält das V-Modell allerdings Produkte offensichtlich unterschiedlicher Abstraktionsebenen, beispielsweise ein Produkt wie die Produktbibliothek und ein Produkt wie das Projekthandbuch. Produktbibliothek und Projekthandbuch sind gleichermaßen als Produkte modelliert. Eine Beziehung, die das Enthaltensein aller Produktexemplare eines Projektes in der Produktbibliothek formuliert, ist nicht explizit modelliert, sondern einzig aus der textuellen Beschreibung ersichtlich. Abstraktionsebenen sind im V-Modell also implizit vermischt und damit einer Werkzeugunterstützung nicht zugänglich. Insbesondere am Beispiel der Umsetzung der Disziplin Systemerstellung wird das Fehlen eines Spezialisierungskonzeptes deutlich, da die Modellierung der Systemerstellung einige Redundanzen aufweist.

Das Konzept des Vorgehensbausteins sowie dessen inhaltliche Umsetzung sind nicht immer konform mit der Zielsetzung des V-Modells XT. Vorgehensbausteine wurden als modulare

Bearbeitungs- und Tailoring-Einheiten konzipiert. Die aufgezeigten Probleme rühren aus der Tatsache, dass die tatsächlichen Abhängigkeiten der Vorgehensbausteine komplex und vielfältig sind. Die Konsistenzregeln, die aus der Konzeption des Vorgehensbausteins sowohl als Tailoring-Einheit als auch als Bearbeitungs-Einheit resultieren, sind verglichen mit der Komplexität der Abhängigkeiten in einem Vorgehensmodell zu eng gefasst. Themen wie die Qualitätssicherung oder Systemsicherheit, die einen querschnittlichen Charakter aufweisen, lassen sich daher nicht befriedigend auf das Konzept des Vorgehensbausteins abbilden.

Konzepte des Vorgehens-Meta- Modells

Kernthema der Arbeit ist also die Modellierung von Vorgehensmodellen, die die Ableitung von Projektplänen präzise vorgeben. In der Arbeit erstellen wir ein Konzept für Vorgehensmodelle in Form eines Vorgehens-Meta-Modells. Dieses konzipieren wir in der Arbeit nicht als Erweiterung des Meta-Modells des V-Modells XT, sondern von Grund auf neu.

Den Grundstein für die Modellierung von Vorgehensmodellen bildet die Definition einer der UML ähnlichen Modellierungstechnik auf Basis von Klassendiagrammen und prädikatenlogischen Constraints. Wir vereinbaren Konventionen der konkreten Syntax, die im Rahmen der Arbeit spezifisch sind. Im Zusammenhang mit dem Konzept der Spezialisierung definieren wir folgende Einschränkungen, die sich der UML nicht entnehmen lassen. Einmal für eine Assoziation definierte Multiplizitäten gelten entsprechend unserer Modellierungstechnik auch für alle Spezialisierungen dieser Assoziation. Eine spezialisierte Assoziation kann Multiplizitäten dabei allenfalls weiter einschränken. Entsprechend gelten einmal formulierte Constraints auch für alle Spezialisierungen der jeweils eingeschränkten Elemente.

Der Modellierungsansatz der Arbeit sieht die Aufteilung in Meta-Ebenen vor. Das Konzept der Spezialisierung verwenden wir als Metaisierungsprinzip. Spezialisierung dient der Abgrenzung der Ebenen Vorgehens-Meta-Modell, Vorgehensmodell und Projekt. Ein Überblick über die Konzepte des Vorgehens-Meta-Modells findet sich im Anhang der Arbeit. In diesem Kapitel geben wir einen knappen Überblick über wesentliche Aspekte des Vorgehens-Meta-Modells.

Da sich die Planung eines Projektes stets an den zu erstellenden Projektergebnissen ausrichtet, ist das Konzept des Produktes im Vorgehens-Meta-Modell zentral. Da Produktinhalte die Ableitung eines Projektplans beeinflussen, sind Produkte des Vorgehensmodells feingranular beschreibbar. Vorgänge als unmittelbaren Bestandteil eines Projektplans modellieren wir in unterschiedlichen Ausprägungen. Ausprägungen des Konzeptes Vorgang dienen der zeitlichen Planung der Erstellung von Produkten, der Planung der Zusammenarbeit mehrerer Bearbeiter bei der Bearbeitung eines Produktes sowie der Strukturierung und Planung eines Projektes in Form von Projektabschnitten.

Projektabschnitte können im Vorgehensmodell beziehungsweise Projekt sehr unterschiedlich strukturiert werden. Die Vorgabe einer geeigneten Strukturierung für Projektpläne auf Ebene der Projektabschnitte kann also nicht Aufgabe eines Meta-Modells sein, sondern ist Aufgabe des Vorgehensmodells. Am Beispiel haben wir eine Strukturierung in Form von Phasen, Releases und Arbeitspaketen gezeigt. Die unterschiedlichen Arten von Vorgängen müssen dabei, anders als in vielen Werkzeugen für die Projektplanung implementiert, keine Hierarchie bilden. Dies haben wir anhand von Beispielen motiviert.

Abläufe modellieren wir im Vorgehens-Meta-Modell in Form von Vorgangsfolgen der Netzplantechnik. Vorgangsfolgen wie „Ende-Ende“ ermöglichen die Festlegung möglicher zeitlicher Überlappungen von Vorgängen. Im Vergleich zum Ablaufkonzept des V-Modells XT erlauben Abläufe die Modellierung präziserer Vorgaben für Projektpläne.

Produktabhängigkeiten der Kategorie Konsistenzabhängigkeit und Produktfluss stellen typische Muster für die Planung der Zusammenarbeit mehrerer Bearbeiter bei der Erstellung der in Beziehung stehenden Produkte dar. Im Gegensatz zum Konzept der Vorgangsfolgen ist dieses Konzept also nicht als vollständig hinsichtlich der Abdeckung aller möglichen Abläufe in einem Projekt anzusehen. Produktabhängigkeiten dienen lediglich als Hilfsmittel für typische Planungen. Andere Arten von Abläufen und Vorgängen zur Konsistenzsicherung können allerdings durch unmittelbare Spezialisierung der Konzepte Ablauf oder Produktbearbei-

tung des Vorgehens-Meta-Modells modelliert werden. Zudem handelt es sich bei Produktabhängigkeiten um ein eigenständiges Konzept, das abgesehen von der Korrespondenz der Namen keinen Bezug zu den Produktabhängigkeiten des V-Modells XT aufweist. Die in der Arbeit konzipierten Produktabhängigkeiten beeinflussen im Gegensatz zu den Produktabhängigkeiten des V-Modells stets den Projektplan.

Eine Produktabhängigkeit der Kategorie Konsistenzabhängigkeit impliziert eine Produktbearbeitung zur Prüfung beziehungsweise Herstellung der Konsistenz der entsprechenden Produktexemplare. Diese Produktbearbeitung ist den jeweiligen Produkterstellungen gemeinsam, das heißt, die Konsistenzsicherung dieser Produktexemplare erfordert eine Zusammenarbeit der jeweiligen Bearbeiter im Projekt. Derart verbundene Produkterstellungen überlappen sich zeitlich mindestens für die Dauer dieser gemeinsamen Konsistenzsicherung.

Produktabhängigkeiten der Kategorie Produktfluss kennzeichnen ein Produktexemplar als Vorgabe für ein Exemplar eines anderen Produktes. Die Herstellung der Konsistenz beider Produktexemplare obliegt dabei allein dem Bearbeiter des Zielproduktexemplars. Produktflüsse implizieren also eine Produktbearbeitung zur Konsistenzsicherung, die dem Bearbeiter des Zielproduktexemplars zugeordnet ist. Produktflüsse wirken sich zudem auf die Reihenfolge der jeweiligen Produkterstellungen aus.

Produktabhängigkeiten der Kategorie Bezug dienen der Modellierung feingranularer Zusammenhänge des Produktmodells, die nicht durch die übrigen Arten von Produktabhängigkeiten modelliert werden können. Bezüge kommen im Vorgehensmodell auf einer Ebene zur Anwendung, die für die Planung eines eigenen Vorgangs zur Konsistenzsicherung zu feingranular ist. Bezüge dienen dabei der Einschränkung der projektspezifischen Produktstruktur mit Hilfe von Multiplizitäten.

Das Vorgehens-Meta-Modell dient dem Prozessingenieur als Sprache für die Beschreibung des Vorgehensmodells. Das Vorgehens-Meta-Modell zeichnet sich durch Einfachheit aus, beispielsweise durch die Zuordnung genau eines Vorgangs zur Produkterstellung zu jedem modellierten Produkt. Daneben können aber für die Beschreibung eines Vorgehensmodells auch unmittelbar alle sprachlichen Mittel der verwendeten Modellierungstechnik eingesetzt werden. Dem Prozessingenieur stehen bei der Beschreibung des Vorgehensmodells beispielsweise Multiplizitäten zur Verfügung, die der Einschränkung der projektspezifischen Produktstruktur dienen. Auf Ebene des Vorgehensmodells können zudem Constraints als Einschränkungen der Projektebene formuliert werden. Die Beschreibung von Vorgehensmodellen wird also nicht durch das Vorgehens-Meta-Modell eingeschränkt, sondern kann sich der Ausdrucksmächtigkeit der Modellierungstechnik basierend auf Klassen und prädikatenlogischen Constraints bedienen. Die Beschreibung komplexer Zusammenhänge als Vorgabe für die Planung ist auf diese Art möglich, wie wir anhand von Beispielen gezeigt haben, wenn auch im Allgemeinen keine triviale Aufgabe.

Modellierungsansatz

Die Modellierung so genannter generischer Vorgehensweisen wie der Qualitätssicherung erfordert komplexe Modellierungen auf Ebene des Vorgehensmodells. Generische Vorgehensweisen auf Ebene des Vorgehensmodells referenzieren Elemente des Vorgehens-Meta-Modells. Die Qualitätssicherung stellt beispielsweise einen Bezug zu allen Produktexemplaren her. Generische Vorgehensweisen treten in Vorgehensmodellen häufiger auf. Mit Hilfe von Assoziationen zwischen Meta-Ebenen und mit Hilfe von Constraints lassen sich generische Vorgehensweisen präzise modellieren, wie wir am Beispiel gezeigt haben.

Entsprechend dem Modellierungsansatz der Arbeit enthält die Projektebene Spezialisierungen der Konzepte des Vorgehensmodells. Bei der Erstellung einer Projektplanung können allerdings auch Konzepte des Vorgehens-Meta-Modells unmittelbar spezialisiert werden. Beispielsweise kann die Projektebene Vorgänge, die im Vorgehensmodell nicht vorgesehen, sind als unmittelbare Spezialisierungen eines Konzeptes des Vorgehens-Meta-Modells beinhalten. Diese Flexibilität im Rahmen der Planung ist wünschenswert, da Vorgehensmodelle als Abstraktionen der Projektebene hinsichtlich aller einzuplanenden Vorgänge in der Praxis nicht vollständig sein können.

Der Modellierungsansatz sieht also die Aufteilung in die Ebenen Vorgehens-Meta-Modell, Vorgehensmodell und Projekt vor, die sich durch das Konzept der Spezialisierung abgrenzen. Ein Spezialisierungskonzept ist allerdings auch auf der Ebene des Vorgehensmodells erforderlich. In einem Entwicklungsprojekt stellt Modellbildung eine gängige Technik dar. Modellbildung, beziehungsweise Generalisierung und Spezialisierung, müssen also auch im Vorgehensmodell beschreibbar sein, da Modellbildung insbesondere einen Einflussfaktor auf die Projektplanung darstellt. Als Beispiel haben wir die Konzeption eines Meta-Modells für die Beschreibung eines Vorgehensmodells betrachtet. Dieses auf Projektebene konzipierte Meta-Modell impliziert bestimmte Abläufe für die weitere Planung des Projektes zur Erstellung des Beispiel-Vorgehensmodells. Nach Fertigstellung kann dieses Beispiel-Meta-Modell aus Sicht des Projektes also als Teil des Vorgehensmodells verstanden werden.

Planung verstehen wir als gleichbedeutend mit dem Konzept der Spezialisierung. Die Projektebene und damit den Projektplan definieren wir als die aktuell „tiefste“ Spezialisierungsebene des Vorgehens-Meta-Modells. Als festgelegt betrachten wir die Ebene des Vorgehens-Meta-Modells. Diese Ebene kann mit dem Konzept der Spezialisierung in einen Projektplan überführt werden. Die Anzahl der erforderlichen Spezialisierungen bis hin zum Projektplan ist dabei nicht fix. Dies entspricht den Gegebenheiten der Praxis, da Projektpläne im Projektverlauf unterschiedliche Spezialisierungsgrade durchlaufen.

Treibt man diese Idee an ihre Grenzen, kann auch das Vorgehens-Meta-Modell selbst, solange keine Spezialisierungen existieren, als Projektplan verstanden werden. Das Vorgehens-Meta-Modell als Projektplan enthält dabei die Aussage, dass in der Projektlaufzeit Projektergebnisse in Form von Produkten durch Vorgänge erstellt werden müssen. Die Erstellung von Ergebnissen in einem zeitlich begrenzten Rahmen entspricht der Definition des Begriffs Projekt. Das Vorgehens-Meta-Modell als Projektplan ist also in jedem Fall korrekt. In der Praxis muss das Vorgehens-Meta-Modell als Projektplan aufgrund seines Detaillierungs-Grades natürlich sogar zu Projektbeginn als ungenügend bezeichnet werden. Theoretisch ist der Grad der Detaillierung eines Projektplans allerdings relativ und nicht absolut definierbar.

Methodik der Erstellung von Vorgehensmodellen

Aufbauend auf den Konzepten des Vorgehens-Meta-Modells geben wir eine Methodik für die arbeitsteilige Erstellung von Vorgehensmodellen an. Diese formulieren wir entsprechend dem Konzept der vorliegenden Arbeit auf der Granularität eines Projektplans in Form einzuplanender Vorgänge. Wir geben also ein Vorgehensmodell für die Erstellung von Vorgehensmodellen an. Zentral dabei ist die Festlegung von Zuständigkeiten und Arbeitspaketen. Bearbeitungs-Einheiten sollten ein hohes Maß an inhaltlicher Zusammengehörigkeit und Unabhängigkeit untereinander aufweisen, um einen möglichst effizienten Bearbeitungsprozess zu ermöglichen.

Modulare Bearbeitungs-Einheiten lassen sich auf Basis der inhaltlichen Strukturierung eines Vorgehensmodells definieren. Die Inhalte des Vorgehensmodells lassen sich in Form von Disziplinen und Planungs-Einheiten hierarchisch strukturieren. Disziplinen kapseln Produkte, während Planungs-Einheiten Planungselemente wie Vorgänge oder Meilensteine kapseln.

Die Idee einer Aufteilung in modulare Bearbeitungs-Einheiten, die eine entkoppelte Bearbeitung ermöglichen, lässt sich im Kontext von Vorgehensmodellen allerdings nur bedingt verwirklichen. Die Zusammenhänge der Elemente eines Vorgehensmodells sind mannigfaltig. Inhaltliche Überlappungen zwischen Disziplinen und Planungs-Einheiten kennzeichnen verwandte Vorgehensmodell-Inhalte. Überlappt sich eine Bearbeitungs-Einheit mit den meisten beziehungsweise potentiell mit allen Disziplinen des Vorgehensmodells, kann sie als querschnittliche Bearbeitungs-Einheit gekennzeichnet werden. Im Kontext des Erstellungsprozesses leitet sich aus einer inhaltlichen Überlappung die Notwendigkeit zur Zusammenarbeit bei der Erstellung bestimmter Bearbeitungs-Einheiten ab. Assoziationen, die sich nicht innerhalb von Bearbeitungs-Einheiten lokalisieren, erfordern ebenfalls eine übergreifende Bearbeitung.

Entsprechend der Methodik zur Erstellung von Vorgehensmodellen sind die inhaltliche Strukturierung und das Tailoring-Modell zuerst zu erarbeiten. Das Tailoring-Modell dient der For-

mulierung der Möglichkeiten zur projektspezifischen Anpassung eines Vorgehensmodells. Tailoring-Bausteine fassen Elemente des Vorgehensmodells zusammen, die auf Projektebene gewählt oder nicht gewählt werden können. Tailoring-Abhängigkeiten dienen der Einschränkung gültiger projektspezifischer Auswahlmengen von Tailoring-Bausteinen. Die Basis des Tailoring-Modells bilden Disziplinen beziehungsweise Planungs-Einheiten.

Inhaltliche Strukturierung und das Tailoring-Modell gelten im Erstellungsprozess als vereinbarte Architektur. Bei der anschließenden detaillierten Modellierung des Vorgehensmodells sind inhaltliche Überlappungen als einschränkende Vorgabe zu verstehen. Da die potentiellen Abhängigkeiten der Elemente eines Vorgehensmodells mannigfaltig sind, dient die Festlegung der inhaltlichen Überlappungen der Eingrenzung der möglichen konkreten Inhalte der Disziplinen beziehungsweise Planungs-Einheiten. Das bereits modellierte Tailoring dient dem gleichen Zweck, nämlich der Positionierung beziehungsweise Abgrenzung der Inhalte der Bearbeitungs-Einheiten. Den Bearbeitern einer Bearbeitungs-Einheit muss jeweils die Rolle der Elemente dieser Bearbeitungs-Einheit im Rahmen des Tailorings bekannt sein, um die Inhalte dieser Bearbeitungs-Einheit geeignet festlegen zu können. Der Zusammenhang zwischen dem Modell zur inhaltlichen Strukturierung und dem Tailoring-Modell, die sich beide auf Disziplinen und Planungs-Einheiten abstützen, ist also aus methodischen Gründen wichtig.

Die Überlappungen der Disziplinen beziehungsweise Planungs-Einheiten erfordern eine geeignete Festlegung des Tailorings in Form von Tailoring-Bausteinen und -Abhängigkeiten. Beispielsweise Aggregations-Beziehungen zwischen Produkten unterschiedlicher Disziplinen müssen bei einer Auswahl eines Teilproduktes stets zu einer Auswahl des Gesamtproduktes führen. Abgesehen von diesem Constraint legen wir, anders als im V-Modell XT, die Auswirkungen der Struktur des Vorgehensmodells auf die Einheiten des Tailorings nicht auf Ebene des Vorgehens-Meta-Modells fest. Aus inhaltlichen Überlappungen der Disziplinen beziehungsweise Tailoring-Einheiten leiten sich keine Konsequenzen für das Tailoring ab, da trotz inhaltlicher Überlappung die Auswahl nur einer Disziplin beziehungsweise Planungs-Einheit im Rahmen des Tailorings sinnvoll sein kann. Dies haben wir anhand von Beispielen gezeigt. Das Tailoring-Modell der vorliegenden Arbeit weist im Vergleich zum V-Modell XT mehr Freiheitsgrade auf. Die geeignete Modellierung des Tailorings obliegt hier dem Prozessingenieur.

Eine Integration der Konzepte der vorliegenden Arbeit mit den Konzepten des V-Modells XT wäre wünschenswert. Die unterschiedlichen Zielsetzungen beider Modelle, nämlich einerseits die übersichtliche Gestaltung eines Vorgehensmodells als Nachschlagewerks für alle Projektbeteiligten und zum anderen die präzise Vorgabe von Projektplänen, ließen sich zweifellos in einem Modell vereinen. Im Rahmen der Arbeit haben wir keine Integration der Meta-Konzepte angegeben. Eine triviale Abbildung ist größtenteils nicht möglich.

Bezug zum V-Modell XT

Die Konzepte des Meta-Modells entsprechen sich weitgehend in Bezug auf Produkte. Produkte können dem Ansatz der vorliegenden Arbeit entsprechend lediglich feingranularer modelliert werden. Auch Vorgänge zur Produkterstellung haben wir analog zum V-Modell XT modelliert. Vorgänge zur Produktbearbeitung sind dagegen nicht mit den Teilaktivitäten des V-Modells vergleichbar. Teilaktivitäten im V-Modell werden nicht im Hinblick auf die Planung modelliert, sondern mit dem Ziel der strukturierten Darstellung einer methodischen Beschreibung. Die Konzepte Produktabhängigkeit und Ablauf finden keine Entsprechung im V-Modell. Diese Konzepte sind auf einer verglichen mit dem V-Modell konkreteren Abstraktionsebene angesiedelt, wie sie für die einfache Ableitbarkeit von Projektplänen erforderlich ist. Inhaltliche Strukturierung und Tailoring-Modell schließlich sind als alternative Modellierungen zu den Vorgehensbausteinen beziehungsweise Produktgruppen des V-Modells zu sehen. Eine schematische Abbildung dieser Konzepte ist ebenfalls nicht möglich.

Die Anwendbarkeit der Konzepte der vorliegenden Arbeit haben wir anhand von zahlreichen Beispielen untersucht, die auch Bestandteil der Arbeit sind. Dennoch können diese nicht die Evaluierung der Konzepte im größeren Rahmen ersetzen. Eine Evaluierung sollte dabei sowohl auf der Ebene des Vorgehensmodells als auch auf Projektebene erfolgen. Auch die prototypische Realisierung einer Werkzeugunterstützung entsprechend dem in Kapitel 6.7

Bewertung

vorgestellten Konzept wäre einer weitergehenden Bewertung der Anwendbarkeit der Konzepte der Arbeit dienlich.

Im Vergleich zum V-Modell XT kann mit Hilfe der hier konzipierten Beschreibungsmittel ein konkreteres Vorgehensmodell beschrieben werden. Die Gefahr besteht dabei in einem zu hohen Detaillierungsgrad eines Vorgehensmodells. Eine hohe Detaillierung kann in Relation zur Zahl der Projekte, die im Anwendungsspektrum eines entsprechenden Vorgehensmodells liegen, in einem zu großen Aufwand für die Erstellung des Vorgehensmodells resultieren. Ausschlaggebend für die Rentabilität in der Praxis werden also zum einen die geeignete Anwendung der Konzepte durch die Rolle des Prozessingenieurs und zum anderen der Nutzen der Projektbeteiligten aus einem entsprechenden Vorgehensmodell sein. Eine Messung des konkreten Nutzens eines Vorgehensmodells in der Praxis ist allerdings nur schwer möglich.

Abbildungsverzeichnis

Abbildung 1–1:	Thema der Arbeit.....	5
Abbildung 1–2:	Adressierte Anwendungsfälle.....	6
Abbildung 1–3:	Präzise Definition der Semantik von Vorgehensmodellen.....	9
Abbildung 2–1:	Merkmale zur Klassifikation von Vorgehensmodellen.....	18
Abbildung 2–2:	Das Wasserfallmodell nach [Royce70].....	19
Abbildung 2–3:	Evolutionäre Entwicklung versus inkrementelle Entwicklung.....	21
Abbildung 2–4:	Agile Entwicklung.....	22
Abbildung 2–5:	Nebenläufige Entwicklung nach [Balzert98] (vereinfacht).....	23
Abbildung 2–6:	Das Spiralmodell aus [Boehm86].....	25
Abbildung 2–7:	Lebendiger Entwicklungsprozess basierend auf Prozessmustern aus [GMP+03].....	26
Abbildung 2–8:	Tailoring im V-Modell XT – Entscheidungspunkte aus [VM-XT05].....	28
Abbildung 2–9:	Tailoring im V-Modell XT – Vorgehensbausteine aus [VM-XT05].....	29
Abbildung 2–10:	Inhaltliches Spektrum im V-Modell XT – Produktgruppen aus [VM-XT05].....	32
Abbildung 2–11:	Inhaltliches Spektrum von Vorgehensmodellen.....	33
Abbildung 2–12:	Abstraktionsgrad von Vorgehensmodellen.....	34
Abbildung 2–13:	Abstraktionsgrad in Bezug auf die Beschreibungsmitteln von Vorgehensmodellen.....	35
Abbildung 2–14:	Anwendungsfälle von Vorgehensmodellen.....	36
Abbildung 2–15:	Allgemeiner Aufbau von Vorgehensmodellen (Ontologie).....	39
Abbildung 2–16:	Produktgruppen des erweiterten Projektmanagements.....	43
Abbildung 2–17:	Projektmanagement-Regelkreisprinzip.....	44
Abbildung 2–18:	Vorgehensbaustein Projektmanagement.....	45
Abbildung 2–19:	Produktstrukturplan: Beispiel.....	47
Abbildung 2–20:	Projektstrukturplan: Beispiel.....	48
Abbildung 2–21:	Vorgangsfolgen.....	49
Abbildung 2–22:	Netzplan mit Vorgangsfolgen und Terminrechnung: Beispiel.....	51
Abbildung 2–23:	Balkendiagramm: Beispiel.....	53
Abbildung 2–24:	Balkendiagramm mit Sammelvorgängen und Meilensteinen: Beispiel ...	54
Abbildung 2–25:	Allgemeiner Aufbau von Projektplänen (Ontologie).....	56
Abbildung 3–1:	Vorgehensbausteinkonzept des V-Modells XT.....	60
Abbildung 3–2:	Instanzdiagramm zum Vorgehensbausteinkonzept (Beispiel): Rollenbeziehungen und Produktabhängigkeitserweiterungen.....	62
Abbildung 3–3:	Beispiele zu Schwachstellen des V-Modells XT: Vorgehensbausteinkonzept (Instanzdiagramm).....	65
Abbildung 3–4:	Ablaufkonzept des V-Modells XT.....	69
Abbildung 3–5:	UML-Aktivitätsdiagramme und Netzplantechnik.....	72
Abbildung 3–6:	Produktmodellkonzept des V-Modells XT.....	75

Abbildung 3–7:	Beispiele zu Schwachstellen des V-Modells XT: Produktmodellkonzept (Instanzdiagramm).....	77
Abbildung 3–8:	Die Konzepte Aktivität, Produkt und Rolle im V-Modell XT.....	81
Abbildung 3–9:	Abstraktionsebenen im V-Modell XT (Beispiele).....	83
Abbildung 3–10:	Systemelemente im V-Modell XT.....	84
Abbildung 4–1:	Die Meta Object Facility (MOF).....	88
Abbildung 4–2:	Muster zur Definition der Semantik einer Sprache.....	91
Abbildung 4–3:	Abstrakte Syntax der Modellierungssprache.....	92
Abbildung 4–4:	Semantische Domäne der Modellierungssprache: Instanzen.....	93
Abbildung 4–5:	Semantik der Modellierungssprache.....	93
Abbildung 4–6:	Konkrete Syntax der Modellierungssprache - Spezifika.....	95
Abbildung 4–7:	Abstrakte Syntax und semantische Domäne: Generalisierung und Spezialisierung.....	98
Abbildung 4–8:	Visualisierung zu Constraint (7).....	99
Abbildung 4–9:	Visualisierung zu Constraint (8).....	100
Abbildung 4–10:	Visualisierung zu Constraint (11).....	101
Abbildung 4–11:	Konkrete Syntax: Generalisierung und Spezialisierung.....	103
Abbildung 4–12:	Modellierungsansatz: Spezialisierung als Metaisierungsprinzip.....	104
Abbildung 5–1:	Architektur der Gesamt-Modellierung.....	108
Abbildung 5–2:	Planungsmodell – Ebene Vorgehens-Meta-Modell.....	111
Abbildung 5–3:	Planungsmodell – Ebene Projekt (Beispiel).....	112
Abbildung 5–4:	Produktmodell – Ebene Vorgehens-Meta-Modell.....	113
Abbildung 5–5:	Produktmodell – Ebene Vorgehensmodell und Projektebene (Beispiel).....	115
Abbildung 5–6:	Planung der Produkterstellung – Ebene Vorgehens-Meta-Modell.....	117
Abbildung 5–7:	Planung der Produkterstellung – Ebene Vorgehensmodell (Beispiel) ..	119
Abbildung 5–8:	Planung der Produkterstellung – Projektebene (Beispiel).....	121
Abbildung 5–9:	Strukturierung des Projektablaufs – Ebene Vorgehens-Meta-Modell ...	123
Abbildung 5–10:	Strukturierung des Projektablaufs – Ebene Vorgehensmodell (Beispiel).....	124
Abbildung 5–11:	Strukturierung des Projektablaufs – Projektebene (Beispiel).....	126
Abbildung 5–12:	Ablaufbeziehungen und Iterationen – Ebene Vorgehens-Meta-Modell	128
Abbildung 5–13:	Ablaufbeziehungen und Iterationen – Ebene Vorgehensmodell und Projektebene (Beispiel: Strukturierung des Projektablaufs).....	129
Abbildung 5–14:	Ablaufbeziehungen und Iterationen – Ebene Vorgehensmodell und Projektebene (Beispiel: Planung der Produkterstellung).....	131
Abbildung 5–15:	Produktabhängigkeiten – Ebene Vorgehens-Meta-Modell.....	133
Abbildung 5–16:	Produktabhängigkeiten – Ebene Vorgehensmodell (Beispiel).....	136
Abbildung 5–17:	Produktabhängigkeiten der Kategorie Erzeugung – Projektebene (Beispiel).....	137

Abbildung 5–18:	Produktabhängigkeiten: Implikation von Vorgängen – Projektebene (Beispiel).....	139
Abbildung 5–19:	Produktabhängigkeiten: Aggregation von Bezügen – Ebene Vorgehensmodell und Projektebene (Beispiel)	140
Abbildung 5–20:	Spezialisierung auf Ebene des Vorgehensmodells (Beispiel).....	143
Abbildung 5–21:	Spezialisierung auf Projektebene beziehungsweise auf Ebene des Vorgehensmodells (Beispiel).....	144
Abbildung 5–22:	Spezialisierungs-Beziehungen aus Sicht des Beispiel-Projektes im Überblick.....	145
Abbildung 5–23:	Auswirkungen von Beschreibungsmitteln auf die Planung gemäß Constraint (32) (Beispiel).....	146
Abbildung 5–24:	Spezifische Ausprägungen des V-Modells XT – Ebene Vorgehensmodell.....	148
Abbildung 5–25:	Beziehungen zwischen Spezialisierungs-Ebenen – Ebene Vorgehensmodell mit Referenzierung des Vorgehens-Meta-Modells (Beispiel).....	149
Abbildung 5–26:	Beziehungen zwischen Spezialisierungs-Ebenen – Projektebene (Beispiel).....	152
Abbildung 6–1:	Modell zur inhaltlichen Strukturierung – Ebene Vorgehens-Meta-Modell	158
Abbildung 6–2:	Modell zur inhaltlichen Strukturierung: Strukturierungen und Strukturierungs-Elemente – Ebene Vorgehensmodell (Beispiel).....	161
Abbildung 6–3:	Modell zur inhaltlichen Strukturierung: Disziplinen – Ebene Vorgehensmodell (Beispiel)	163
Abbildung 6–4:	Modell zur inhaltlichen Strukturierung: Planungs-Einheiten – Ebene Vorgehensmodell (Beispiel)	165
Abbildung 6–5:	Modell zur inhaltlichen Strukturierung: Weitere Planungs-Einheiten – Ebene Vorgehensmodell (Beispiel).....	167
Abbildung 6–6:	Tailoring-Modell – Ebene Vorgehens-Meta-Modell.....	169
Abbildung 6–7:	Tailoring-Modell – Ebene Vorgehensmodell (Beispiel)	171
Abbildung 6–8:	Modellierung von inhaltlicher Strukturierung und Tailoring am Beispiel des V-Modells XT	174
Abbildung 6–9:	Anwendung von Vorgehensmodellen: Ebene Vorgehens-Meta-Modell	177
Abbildung 6–10:	Anwendung von Vorgehensmodellen: Beispiel.....	182
Abbildung 6–11:	Methodik der Erstellung von Vorgehensmodellen.....	186
Abbildung 6–12:	Modellierung von Texten – Ebene Vorgehens-Meta-Modell.....	191
Abbildung 6–13:	Konzept einer Werkzeugunterstützung	194

Quellenverzeichnis

- [4EverEdit] 4Ever XML Framework. <http://sourceforge.net/projects/fouever/>. 2005.
- [AAMI03] Association for the Advancement of Medical Instrumentation. Medical device software – Software life cycle processes. Committee Draft for Vote. 2003.
- [AK01] C. Aktinson, T. Kühne. The Essence of Multilevel Metamodeling. Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools. Lecture Notes In Computer Science. Vol. 2185. 2001.
- [Amb198] S. W. Ambler. Process Patterns: Building Large-Scale Systems Using Object Technology. Cambridge University Press. 1998.
- [Amb199] S. W. Ambler. More Process Patterns: Delivering Large-Scale Systems Using Object Technology. Cambridge University Press. 1999.
- [ArgoUML] ArgoUML. <http://argouml.tigris.org/>. 2005.
- [Balzert98] H. Balzert. Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung. Spektrum Verlag. 1998.
- [Bauer93] F. L. Bauer. Software Engineering – wie alles begann. Informatik-Spektrum, 16(5):259-260, Oktober 1993.
- [Beck99] K. Beck. Extreme Programming Explained: Embrace Change. Addison-Wesley. 1999.
- [BFG91] S. Bandinelli, A. Fuggetta, C. Ghezzi, A. Morzenti. A multi-paradigm Petri net based approach to process description. Proceedings of the 7th International Software Process Workshop. 1991.
- [BMI04] Bundesministerium des Innern. KBSt-Brief „IT-Projektmanagement“. Vorversion vom 20.1.2004. 2004.
- [BBE+03] M. Broy, B. Brügge, J. Eberspächer, G. Färber, G. Reinhart, H. Wildemann. Bayerischer Forschungsverband Software Engineering FORSOFT II. Abschlussbericht 2003. <http://www.forsoft.de>. 2003.
- [BT75] V. R. Basili, A. J. Turner. Iterative Enhancement: A Practical Technique for Software Development. IEEE Transactions on Software Engineering. 1975.
- [Bening56] H. D. Benington. Production of Large Computer Programs. Reprinted in: Annals of the History of Computing, October 1983. 1956.
- [Boehm86] B. W. Boehm. A Spiral Model of Software Development and Enhancement. ACM Software Engineering Notes. 1986.
- [Boehm83] B. W. Boehm. Software Process Management: Lessons Learned from History. Annals of the History of Computing, October 1983.

- [BB00] E. Breton, J. Bezivin. An Overview of Industrial Process Meta-Models. Proceedings of the 13th International Conference Software & System Engineering and their Applications. 2000.
- [BB01] E. Breton, J. Bezivin. Process-Centered Model Engineering. Proceedings of the 5th IEEE International Enterprise Distributed Object Computing Conference. 2001.
- [BHK+98] U. Becker-Kornstaedt, D. Hamann, R. Kempkens, P. Rösch, M. Verlage, R. Webby, J. Zettel. The SPEARMINT Approach to Software Process Definition and Process Guidance. IESE-Report 035.98/E. 1998.
- [Breu01] R. Breu. Objektorientierter Softwareentwurf - Integration mit UML. Springer. 2001.
- [BRSV98a] K. Bergner, A. Rausch, M. Sihling, Alexander Vilbig. A Componentware Development Methodology based on Process Patterns. Proceedings of the 5th Annual Conference on the Pattern Languages of Programs. 1998.
- [BRSV98b] K. Bergner, A. Rausch, M. Sihling, A. Vilbig. A Componentware Methodology based on Process Patterns. Technical Report TUMI9823, Technische Universität München. 1998.
- [CEK01] T. Clark, A. Evans, S. Kent. Engineering Modeling Languages: A Precise Meta-Modeling Approach. The precise UML group. <http://www.puml.org/>. 2001.
- [CG98] G. Cugola, C. Ghezzi. Software Processes: A Retrospective and a Path to the Future. Proceedings of the 5th International Conference on Software Process. 1998.
- [CHL+94] R. Conradi, M. Hagaseth, J.-O. Larsen, M. N. Nguyen, B. P. Munch, P. H. Westby, W. Zhu. Object-Oriented and Cooperative Process Modelling in EPOS. Research Studies Press Advanced Software Development Series. Software process modelling and technology. 1994.
- [Chroust92] G. Chroust. Modelle der Software-Entwicklung. Oldenbourg Verlag München Wien. 1992.
- [Chroust94] G. Chroust. Partial Process Models. Software Systems in Engineering. The American Society of Mechanical Engineers. New Orleans. 1994.
- [Chroust00] G. Chroust. Software Process Models: Structure and Challenges. Proceedings of the IFIP Congress 2000. Software: Theory and Practice. 2000.
- [CMMI] CMMI® - Capability Maturity Model Integration. Carnegie Mellon, Software Engineering Institute. Pittsburgh, USA. <http://www.sei.cmu.edu/CMMI>. 2004.
- [DeMarco78] T. DeMarco. Structured Analysis and System Specification. Yourdon Press. 1978.

- [DGR03] M. Deubler, M. Gnatz, A. Rausch. Analyse-Workshop Ergebnisse. Workshops mit V-Modell 97 Anwendern. <http://www.v-modell-200x.de/>. 2003.
- [DIN69900-1] DIN 69900 Teil 1. „Projektwirtschaft: Netzplantechnik, Begriffe“. 1987.
- [DIN69901] DIN 69901. „Projektwirtschaft: Projektmanagement, Begriffe“. 1987.
- [DGH02] T. Dittmann, V. Gruhn, M. Hagen. Improved Support for the Description and Usage of Process Patterns. Proceedings of the 1st Workshop on Software Development Process Patterns. 2002.
- [Dröschel00] Das V-Modell 97. Der Standard für die Entwicklung von IT-Systemen mit Anleitung für den Praxiseinsatz. Oldenbourg. 2000.
- [DW98] D. F. D'Souza, A. C. Wills. Objects, Components, and Frameworks With UML: The Catalysis Approach. Addison Wesley Publishing Company. 1998.
- [EG90] W. Emmerich, V. Gruhn. Software Process Modelling with FUNSOFT Nets. Technical Report 47, Dept. of Computer Science, University of Dortmund. 1990.
- [GDM+04] M. Gnatz, M. Deubler, M. Meisinger, A. Rausch. Towards an Integration of Process Modeling and Project Planning. The 5th International Workshop on Software Process Simulation and Modeling. 2004.
- [GMH00] S. Goldmann, J. Münch, H. Holz. Distributed Process Planning Support with MILOS. International Journal of Software Engineering and Knowledge Engineering. August 2000.
- [GMP+01a] M. Gnatz, F. Marschall, G. Popp, A. Rausch, W. Schwerin. Towards a Living Software Development Process based on Process Patterns. Proceedings of the Eighth European Workshop on Software Process Technology. 2001.
- [GMP+01b] M. Gnatz, F. Marschall, G. Popp, A. Rausch, W. Schwerin. Modular Process Patterns supporting an Evolutionary Software Development Process. Proceedings of the Third International Conference on Product Focused Software Process Improvement. 2001.
- [GMP+02] M. Gnatz, F. Marschall, G. Popp, A. Rausch, W. Schwerin. Towards a Tool Support for a Living Software Development Process. Proceedings of the Hawai'i International Conference on System Sciences, Big Island, Hawaii. 2002.
- [GMP+03] M. Gnatz, F. Marschall, G. Popp, A. Rausch, W. Schwerin. The Living Software Development Process. Software Quality Professional, Volume 5, Issue 3, June 2003.
- [HJK+96] P. Heimann, G. Joeris, C.-A. Krapp, B. Westfechtel. DYNAMITE: Dynamic Task Nets for Software Process Management. Proceedings of the 18th international conference on Software engineering. 1996.

- [HERMES03] Hermes. Führen und Abwickeln von Projekten der Informations- und Kommunikationstechnik (IKT). Informationsstrategieorgan Bund. Bern. Entwurf Ausgabe 2003.
- [Humph90] W. S. Humphrey. Managing the Software Process. Addison-Wesley. 1990.
- [HSW03] M. Heller, A. Schleicher, B. Westfechtel. A Management System for Evolving Development Processes. Proceedings of the 7th International Conference on Integrated Design and Process Technology. 2003.
- [in-Step] microTool in-Step 2.0 – The Workflow Management System for IT projects. <http://www.microtool.de/instep/en/>
- [ISO15288] ISO 15288. Life Cycle Management – System Life Cycle Processes. 2000.
- [Jacob92] I. Jacobson. Object-Oriented Software Engineering: A Use Case Driven Approach. Addison Wesley Publishing Company. 1992.
- [JBR99] I. Jacobson, G. Booch, J. Rumbaugh: The Unified Software Development Process. Addison Wesley Longman. 1999.
- [JPS+94] G. Junkermann, B. Peuschel, W. Schäfer, S. Wolf. MERLIN: Supporting cooperation in software development through a knowledge-based environment. Software Process Modelling and Technology. Research Studies Limited. 1994.
- [KK03] P. Kroll, P. Kruchten. The Rational Unified Process Made Easy. An Practitioner's Guide to the RUP. Addison-Wesley. 2003.
- [LC93] C. Liu, R. Conradi. Automatic Replanning of Task Networks for Process Model Evolution in EPOS. Lecture Notes In Computer Science, Vol. 717. 1993.
- [LGJ+98] J. Lee, M. Gruninger, Y. Jin, T. Malone, A. Tate, G. Yost. The PIF Process Interchange Format and Framework Version 1.2. The Knowledge Engineering Review. March 1998.
- [MDBG+00] F. Maurer, B. Dellen, F. Bendeck, S. Goldmann, H. Holz, B. Kötting, M. Schaaf. Merging Project Planning and Web-Enabled Dynamic Workflow Technologies. IEEE Internet Computing. Vol. 4, No. 3 2000.
- [ME99] G. Müller-Ettrich. Objektorientierte Prozessmodelle. UML einsetzen mit OOTC, V-Modell, Objectory. Addison-Wesley. 1999.
- [Merbeth93] G. Merbeth. Maestro II – das integrierte CASE-System von Softlab. In: CASE – Systeme und Werkzeuge (Hrsg. H. Balzert). 1993.
- [MSProj03] Microsoft Corporation. Microsoft Project 2003.
- [NR69] P. Naur, B. Randell (Eds.). NATO Konferenz: Working Conference on Software Engineering. Garmisch, 7-11 Oct. 1968. 1969.

- [Oester98] B. Oestereich. Objektorientierte Softwareentwicklung. Analyse und Design mit der Unified Modeling Language. Oldenbourg. 1998.
- [OMG02] Object Management Group. Meta Object Facility (MOF) Specification. Version 1.4. April 2002.
- [OMG03a] Object Management Group. Unified Modelling Language (UML) Specification: Infrastructure. Version 2.0 – 15.09.2003. Working Document. 2003.
- [OMG03b] Object Management Group. Meta Object Facility (MOF) 2.0 Core Specification. 04.10.2003, Final Adopted Specification. 2003.
- [OMG03c] Object Management Group. Object Constraint Language (OCL). Response to the UML 2.0 OCL Request for Proposal. 01.06.2003.
- [Osterw87] L. Osterweil. Software processes are software too. Proceedings of the 9th international conference on Software Engineering. 1987.
- [Plögert02] K. Plögert. Erfahrungen bei der Einführung von angepassten Vorgehensmodellen auf der Basis des V-Modells. In: R. Kneuper, R. Petrasch, M. Wiemers (Hrsg.). Angepasste Vorgehensmodelle. 9.ter Workshop der Fachgruppe 5.11 der Gesellschaft für Informatik e.V.. Shaker Verlag. 2002.
- [PM04] Projektmagazin. <http://www.projektmagazin.de>. 2004.
- [Primav04] Primavera. Primavera Project Planner Version 4.1. <http://www.primavera.com>. 2004.
- [RF00] J. M. Ribo, X. Franch. PROMENADE: A PML Intended to Enhance Standardization, Expressiveness and Modularity in Software Process Modelling. Research Report. Departament de Llenguatges i Sistemes Informàtics. 2000.
- [Royce70] W. W. Royce. Managing the Development of Large Software Systems. Reprinted in: Proceeding of the 9th International Conference of Software Engineering. 1970.
- [RPW04] IBM. Rational Process Workbench (RPW). <http://www-128.ibm.com/developerworks/rational>. 2004.
- [Scheer98] A.-W. Scheer. ARIS: Modellierungsmethoden, Metamodelle, Anwendungen. Springer Verlag, 1998.
- [SHO95] S. M. Sutton, D. Heimbigner, L. J. Osterweil. APPL/A: A Language for Software Process Programming. ACM Transactions on Software Engineering and Methodology. Juli 1995.
- [SKR98] C. Schlenoff, A. Knutilla, S. Ray. A Robust Process Ontology for Manufacturing Systems Integration. Proceedings of the 2nd International Conference on Engineering Design and Automation. 1998.

- [SGI01] Standish Group International, Inc. 2001. Collaborating on Project Success. Software Magazine, February/March 2001. Wiesner Publishing. 2001.
- [Spect91] Special Report: Concurrent Engineering. IEEE Spectrum. 1991.
- [SPEM01] Object Management Group. The Software Process Engineering Metamodel (SPEM). OMG document number ad/2001-03-08. 2001.
- [Strah98] S. Strahringer. Ein sprachbasierter Metamodellbegriff und seine Verallgemeinerung durch das Konzept des Metaisierungsprinzips. Proceedings des GI-Workshops in Münster. 1998.
- [SWZ95] A. Schürr, A. J. Winter, A. Zündorf. Graph Grammar Engineering with PROGRES. Proceedings of the European Software Engineering Conference. LNCS 989. 1995.
- [Taa04] The Agile Alliance. <http://www.agilealliance.org/>. 2004.
- [Thaller00] G. E. Thaller. ISO 9001. Software-Entwicklung in der Praxis. Heise. 2000.
- [VM97a] Allgemeiner Umdruck Nr. 250. Entwicklungsstandard für IT-Systeme des Bundes, Vorgehensmodell. Teil 1: Regelungsteil. Juni 1997.
- [VM97b] Allgemeiner Umdruck Nr. 250. Entwicklungsstandard für IT-Systeme des Bundes, Vorgehensmodell. Teil 3: Handbuchsammlung: Tailoring und Projektspezifisches V-Modell. Juni 1997.
- [VM-XT05] V-Modell XT – Entwicklungsstandard des Bundes für IT-Systeme. <http://www.v-modell-xt.de>. Stand: 04.02.2005.
- [WEIT04] Projekt WEIT. Weiterentwicklung des Entwicklungsstandards für IT-Systeme des Bundes auf Basis des V-Modells 97. <http://weit.informatik.tu-muenchen.de>. August 2004.
- [Westfe01] B. Westfechtel. Ein graphbasiertes Managementsystem für dynamische Entwicklungsprozesse. Informatik Forschung und Entwicklung, Vol. 16. Springer-Verlag. 2001.
- [WfMC95] Workflow Management Coalition. Workflow Reference Model. WfMC-TC-1003. <http://www.wfmc.org/>. 1995.
- [WfMC99] Workflow Management Coalition. Terminology & Glossary. WfMC-TC-1011. <http://www.wfmc.org/>. 1999.
- [XP03] Extreme Programming: A gentle introduction. <http://www.extremeprogramming.org/>. 2003.
- [YC79] E. Yourdon, L. Constantine. Structured Design. Prentice Hall. 1979.

Anhang

A. Vorgehens-Meta-Modell im Überblick

