

## Engineering Workflow auf Basis eines objektorientierten Produktmodells

Goltz, M.

*Bedarfsgerechte Kommunikation und Zugang zu entscheidungsbestimmenden Informationen sind Schlüsselfaktoren für eine erfolgreiche Produktentwicklung. Die Koordination komplexer Entwicklungsprojekte erfordert es, Ordnung in das ‚kreative Chaos‘ der Konstruktion zu bringen. Klassische Ansätze des Prozessmanagements versagen hierbei aber aufgrund der Vielzahl von Einflussfaktoren. Der folgende Artikel stellt ein Konzept eines Engineering Workflows auf Basis vernetzter Entwicklungsobjekte innerhalb eines dreistufigen Produktmodells vor. Ein wesentliches Ziel ist dabei die Einbindung von Kunden und Lieferanten in den gesamten Entwicklungsprozess – ausgehend von der Anforderungsermittlung bis hin zu Änderungsprozessen.*

*Communication on demand and access to information decisive for decisions are key success factors in product design. The co-ordination of complex development projects demands control over the ‘creative chaos’ in product design. Conventional approaches of process management fail due to the large number of influencing factors. The following article introduces a concept for an Engineering Workflow based on networked design objects in a three-level object-oriented product model. The concept is focusing on the integration of customers and suppliers all along the product design process – starting from the requirements determination to the management of change processes.*

### 1 Der Konstruktionsprozess – kreatives Chaos ohne Prozessmanagement?

Aus der Sicht einer gezielten Prozesssteuerung, die jederzeit in der Lage ist, den aktuellen Entwicklungsstand sowie noch ausstehende Aktivitäten zu ermitteln, muss diese Frage mit „ja“ beantwortet werden, da die Vielzahl der Einflussfaktoren auf den Entwicklungsprozess die Planbarkeit erheblich erschwert. Verstärkt wird dieses noch durch den Widerspruch der Produktentwicklung, einerseits dem Konstrukteur ein Arbeitsumfeld mit maximaler Freiheit zur Ausarbeitung einer Lösung zur Verfügung zu stellen, und andererseits möglichst genaue

Kenntnisse über Art und Umfang der Aufgaben zu erlangen. Zudem handeln Entwicklungsingenieure häufig auf Basis unscharfer Informationen. Ungenaue Zielvorgaben, konkurrierende Optimierungsziele und nicht abgesicherte Zwischenergebnisse vorangegangener Prozessschritte erfordern vielfach Iterationen, bei denen nicht zwingend die gleichen Arbeitsfolgen durchlaufen werden. Vielmehr wird der nächste Prozessschritt durch die Art und Qualität der verfügbaren Informationen bestimmt, wodurch die Definition standardisierter Prozesse zusätzlich erschwert wird.

Bisherige Lösungsversuche haben gezeigt, dass feste Abläufe entweder auf sehr hohem Abstraktionsniveau oder bei sehr in die Tiefe gehender Detaillierung definiert werden können. Ein hohes Abstraktionsniveau entspricht in den meisten Fällen der Projektmanagementebene. Die Produktentwicklung wird in Teilprojekte und Arbeitspakete zerlegt, wobei jedoch die Verwaltung von Budgets und Ressourcen im Vordergrund steht. Abläufe, die auf dieser Ebene definiert werden können, bieten so gut wie keine Unterstützung für die tägliche Arbeit. Als weiteres Beispiel für einen hohen Abstraktionsgrad sei an dieser Stelle die Zertifizierung nach ISO 9000 angeführt. Während die Prozesse im kaufmännischen und produktiven Bereich sehr gut dokumentiert sind, finden sich im Bereich der Produktentwicklung häufig nur Angaben zu Meilensteindefinitionen oder Freigabeabläufen.

Begibt man sich jedoch auf ein sehr hohes Detaillierungsniveau, ist es durchaus möglich, Standardabläufe für die Entwicklung einzelner Produktkomponenten<sup>1</sup> zu bestimmen. Als Entscheidungskriterium können verschiedene Eigenschaften dienen, wie z.B. Herstellungsverfahren, Baugruppenzugehörigkeit, etc. Die so definierten Standardprozesse dienen dann als Leitfaden für Neu- und

---

<sup>1</sup> Unter einer Produktkomponente werden sowohl einzelne Bauteile als auch Baugruppen, Aggregate oder Systeme zusammengefasst, aus denen sich ein Produkt zusammensetzt.

Anpassungskonstruktionen sowie für Änderungsprozesse. Ein wesentlicher Nachteil dieser Lösung besteht im Generierungs- und Anpassungsaufwand für derartige Regelwerke. Außerdem ist es schwierig, die detaillierten Teilprozesse auf Bauteilebene zu einem Gesamtprozess für das Endprodukt zusammenzuführen. Des Weiteren verhindert die Definition solcher rigider Prozesse die Suche nach neuen innovativen Lösungen, da sie dem Konstrukteur ein bestimmtes Lösungsverfahren vorschreiben.

Trotz der genannten Schwierigkeiten nimmt der Zwang einer plan- und kontrollierbaren Produktentwicklung deutlich zu. Immer komplexere Produkte erzeugen eine Menge an Produktinformationen, die ohne eine gezielte Ablaufsteuerung kaum noch beherrscht werden kann. Outsourcing und die Konzentration auf Kernkompetenzen führen dazu, dass Unternehmen verstärkt in Entwicklungspartnerschaften (sog. „Engineering Networks“) kooperieren. Dies erfordert eine zielgerichtete und bedarfsgerechte Kommunikation und Abstimmung der Entwicklungsschritte der Partner untereinander. Die daraus resultierenden Abhängigkeiten innerhalb des Produkts und innerhalb des Entwicklungsprozesses sind für den einzelnen Konstrukteur nicht mehr überschaubar.

## 2 IT-Unterstützung im Entwicklungsprozess

### 2.1 CAD-Systeme und Wissensmanagement

Die ersten Computersysteme für die Konstruktion waren einfache 2D-CAD-Systeme, deren Aufgabe hauptsächlich darin bestand, das Zeichenbrett zu ersetzen. Heutige 3D-CAD-Systeme übersteigen deren Leistungsfähigkeit natürlich bei weitem. Trotzdem stellen sie immer noch sehr spezialisierte Insellösungen dar, die nur bedingt zur Prozesssteuerung in Engineering Networks geeignet sind. Funktionen zur Erleichterung der Zusammenarbeit beschränken sich in den meisten Fällen auf web-basierte Lösungen, die einen Zugriff auf Daten und die gemeinsame Bearbeitung dieser von unterschiedlichen Standorten aus ermöglichen. Das Wissen, welches sich mittlerweile durch entsprechende Knowledge-Werkzeuge in CAD-Modellen ablegen lässt, ist von Außen nicht zugänglich, und damit für ein integriertes Prozessmanagement nicht verfügbar.

Standardisierte Schnittstellen oder herstellereigene Anpassungen erlauben es zwar, CAD-Daten in nachgelagerten CAx-Systemen (wie z.B. FEM) zu nutzen. Eine Kopplung zu PLM-Systemen ist aber in den meisten Fällen nur auf den Austausch verwaltungsrelevanter Daten (Ersteller, Zeichnungsformat, etc.) und einen Abgleich der Produktstruktur ausgelegt. Dieses führt dazu, dass bei Änderungen, die im PLM-System ablauforganisatorisch geplant werden, gerade geometrische und topologische Abhängigkeiten innerhalb eines Produktmodells übersehen werden. Daraus resultieren zeit- und kostenintensive Iterationen, die sich vermeiden ließen, wenn bereits in der Planungsphase des Änderungsvorhabens ein Zugriff auf diese Informationen möglich wäre.

In Bezug auf das Anforderungsmanagement wäre eine weiterreichende Integration ebenfalls wünschenswert. Viele Anforderungen sind geometrischer Natur, die einerseits über Knowledge-Elemente in das CAD-System eingebracht werden können, und deren Einhaltung andererseits über eine Rückkopplung aus dem CAD-System überwacht werden kann. Dazu sind natürlich auch im PLM-System entsprechende Strukturen zu schaffen, die diese Informationen verwalten können.

### 2.2 PLM-Systeme

Mit zunehmender Nutzung der Computer im Konstruktionsumfeld ergab sich die Notwendigkeit, die anfallenden Datenmengen mit entsprechenden Werkzeugen zu verwalten. Aus einfachen Zeichnungsverwaltungssystemen entstanden so in den letzten Jahren leistungsfähige Produktdaten-Management-Systeme, die zusätzlich zur Dokumentenverwaltung auch Funktionen zum Produktstrukturmanagement, zur Klassifikation und zum Projekt- und Workflowmanagement anbieten.<sup>2</sup>

Mittlerweile verschiebt sich der Fokus vom reinen Datenverwaltungswerkzeug hin zu einem umfassenden System zum Management aller mit dem Produktleben verbundener Daten – zu so genannten PLM-Systemen (Product Lifecycle Management). Die Erweiterung bezieht sich auf die horizontale Integration durch Einbeziehung weiterer Module zur Unterstützung des

<sup>2</sup> PLM-Systeme sind auch Gegenstand des Artikels „PDM/PLM – Verwalten von Produktdaten ohne Grenzen !?“ auf Seite 59

Produktlebenszyklus, wie z.B. Anforderungs- oder Instandhaltungsmanagement, sowie die vertikale Integration zur Unterstützung der Kollaboration vernetzter Unternehmen. Damit entwickeln sich PLM-Systeme immer mehr zum Rückgrat der DV-Landschaft in den entwicklungsbezogenen Bereichen eines Unternehmens.

Allerdings fehlen auch aktuellen PLM-Systemen noch Funktionen zur effizienten Ablaufsteuerung in der unternehmensübergreifenden Produktentwicklung. Dazu zählt beispielsweise auch die Festlegung von Kommunikationsregeln (*Wer ist wann über was in welcher Form zu informieren?*), die mit den verfügbaren Workflow-Funktionen nicht zufrieden stellend definiert werden können. Dieses Defizit wird im nächsten Abschnitt näher untersucht.

## 2.3 Workflowmanagement in der Produktentwicklung

### 2.3.1 ‚Klassischer‘ Workflow

Unter einem Workflow versteht man die (teilweise) Automatisierung eines Geschäftsprozesses zur Übergabe von Dokumenten und Informationen nach vorgegebenen Regeln von einem Teilnehmer zum anderen, um eine bestimmten Aufgabe zu erfüllen (nach /1/). Ihren Ursprung haben Workflow-Management-Systeme (WfMS) in der „Office-Automatisierung“, bei der Dokumente in einer elektronischen Umlaufmappe von einem Bearbeiter zum nächsten geschickt werden.

Für das Management komplexer Entwicklungsprozesse ist diese Art der dokumentenbasierten Workflows aus mehreren Gründen ungeeignet. Zum einen bietet die starre Zuordnung von Personen und Rollen zu Aufgaben nicht die nötige Flexibilität, die bei Konstruktionsprozessen erforderlich ist. Zum anderen lassen sich Workflows nur in Abhängigkeit vom zu bearbeitenden Objekttyp definieren, der Informationsgehalt eines bestimmten Objekts hat keinen Einfluss auf die Rollenzuordnung oder die Art und Abfolge der Aktivitäten. Gerade in der Konstruktion ist die Zuständigkeit einzelner Personen aber stark objektabhängig. Während in einem Fall z. B. ein Bearbeiter für das Erstellen einer 2D-Zeichnung einer Baugruppe A zuständig ist, kann in Bezug auf die Baugruppe B ein ganz anderer Bearbeiter für die Erstellung des gleichen Zeichnungstyps verantwortlich sein. D. h. die Verantwortlichkeit definiert sich in der Konstruktion

vielfach auf Basis der betrachteten Entwicklungsobjekte.

### 2.3.2 Parameterbasierter Engineering Workflow

Im Rahmen des EU-Projekts SIMNET<sup>3</sup> wurde daher ein System für einen parameterbasierten Workflow entwickelt. Im Mittelpunkt stehen dabei Konstruktionsparameter, die konstruktive Eigenschaften einer Produktkomponente definieren. Die Abfolge der Entwicklungsschritte ergibt sich aus der Vernetzung der Parameter untereinander. Personen werden innerhalb einer bestimmten Benutzerkategorie einem Parameter zugeordnet, die die Rolle der Person hinsichtlich des betrachteten Parameters definiert. Die Kommunikation zwischen Projektpartnern erfolgt durch einen Freigabeworkflow, der durch vorgegebene Statusübergänge ausgelöst wird. Obwohl die Vorteile dieser Lösung im Rahmen des Forschungsvorhabens nachgewiesen werden konnten, ist eine Erweiterung des Konzepts auf zusätzliche Produktdatenelemente notwendig, um die Abstimmung zwischen Entwicklungspartnern über den gesamten Produktlebenszyklus unterstützen zu können. Zudem ist bislang keine eindeutige Trennung von Anforderungen und Parametern möglich, was die Nachverfolgung im Rahmen des Änderungsmanagements erschwert.

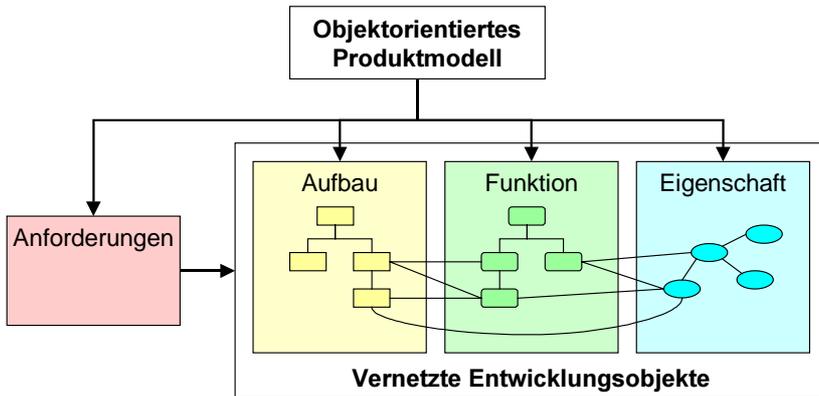
Im folgenden wird daher ein Konzept für einen erweiterten Engineering Workflow vorgestellt, welches es Entwicklungspartnern ermöglicht, auf der Basis unterschiedlicher Produktinformationen gezielt miteinander zu kommunizieren, um Konstruktionsentscheidungen und Entwicklungsaktivitäten untereinander abzustimmen.

## 3 Konzept eines 3-stufigen objektorientierten Produktmodells

### 3.1 Definition „Entwicklungsobjekt“

Der Begriff „Entwicklungsobjekt“ wird von Kruse wie folgt definiert: *„Imaginäres oder reales Objekt eines technischen Systems in beliebiger Konkretisierung, das als Aggregation oder als Einzelobjekt durch seine Eigenschaften und Relationen zu anderen Entwicklungsobjekten beschrieben ist.“ /2/*

<sup>3</sup> SIMNET – Workflow Management for Simultaneous Engineering Networks, gefördert mit Mitteln der Europäischen Kommission (EP-26780), <http://www.imw.tu-clausthal.de/simnet>



**Bild 1:** Objektorientiertes Produktmodell

Gerade in den frühen Phasen der Produktentwicklung greift diese Definition aber zu kurz. Oftmals liegen am Beginn einer Entwicklung nur sehr abstrakte Konzepte für das zu entwickelnde Produkt vor. Bei der Lösungsfindung stehen daher eher zu erfüllende Funktionen oder geforderte Eigenschaften im Mittelpunkt. Aus diesem Grund bietet es sich an, den Begriff des Entwicklungsobjekts auf Funktionen und Eigenschaften auszudehnen, da diese ebenfalls als Auslöser für Diskussionen zwischen Entwicklungspartnern sind. Zudem sind Funktionen und Eigenschaften die Bindeglieder zwischen den einzelnen Produktkomponenten (siehe Bild 1). Verallgemeinert lässt sich damit sagen:

*Ein Entwicklungsobjekt ist Gegenstand einer konstruktiven Tätigkeit. Dabei kann es sich um ein reales oder imaginäres Objekt eines technischen Systems, um eine von diesem System zu erfüllende Funktion oder um eine das System oder eine Funktion beschreibende Eigenschaft handeln. Die*

*Gesamtheit der Entwicklungsobjekte und ihre Relationen untereinander beschreiben den aktuellen Entwicklungsstand eines Produkts.*

### 3.2 Vernetzung mithilfe modellbasierter Kontextbeschreibungen

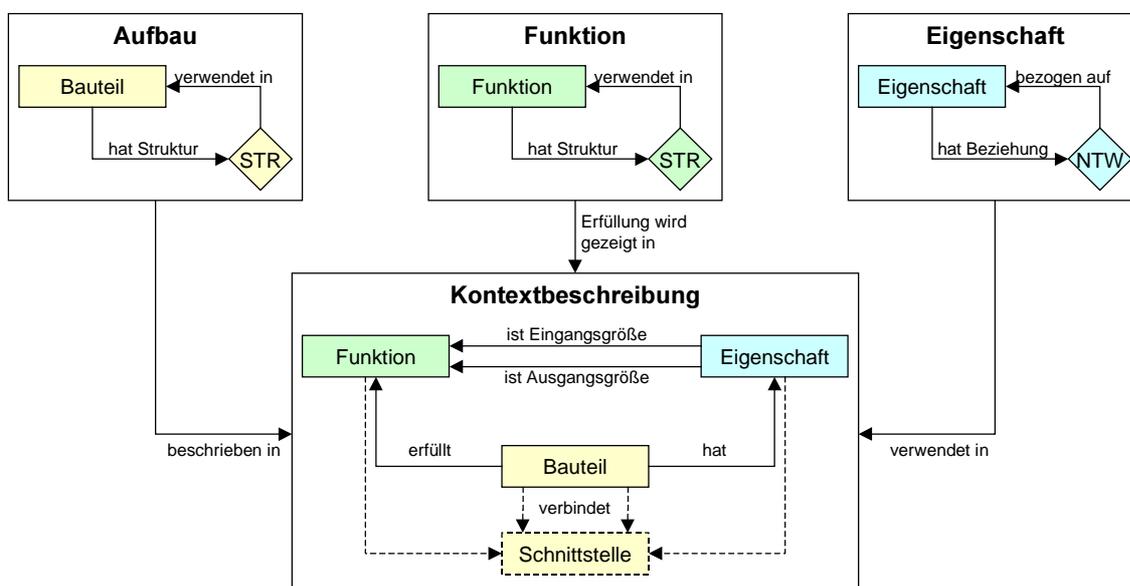
Die Zusammenhänge zwischen verschiedenen Entwicklungsobjekten werden mithilfe einer modellbasierten

Kontextbeschreibung dargestellt (Bild 2), die die Beziehungen zwischen Produktkomponenten, Funktionen und Eigenschaften unter einem bestimmten Gesichtspunkt verdeutlicht. Dieser Ansatz lässt sich mit den Anwendungsfällen (Use Cases) der Unified Modelling Language (UML) vergleichen. Ein Anwendungsfall veranschaulicht eine typische Interaktion zwischen einem Anwender und einem Computersystem [3]. Wesentliche Merkmale eines Anwendungsfalls sind:

- Erfassen einer für den Benutzer sichtbaren Funktion,
- Erreichen eines für den Benutzer abgrenzbaren Ziels.

Darauf basierend lassen sich für die Kontextbeschreibungen die folgenden bestimmenden Merkmale ableiten:

- Erfassen einer für das Produkt wesentlichen Charakteristik (notwendige Komponente, zu erfüllende Funktion oder notwendige



**Bild 2:** Entwicklungsobjekte in der Kontextbeschreibung

Eigenschaft),

- Erfassen der Informationen, die zur Erfüllung einer Anforderung an das Produkt notwendig sind (Kundenforderung, Forderungen zur Abstimmung mit Lieferanten).

Im weitesten Sinne geht es bei der Kontextbeschreibung um die ‚Interaktion‘ der verschiedenen Entwicklungsobjekte im Rahmen von Szenarien. Diese sollen Entwicklungsobjekte passend zur augenblicklichen Aufgabenstellung gruppieren. Dabei kann einerseits die Erfüllung einer bestimmten Funktion durch die jeweiligen Funktionsträger mit den dafür relevanten Eigenschaften im Mittelpunkt stehen. Andererseits kann ein Szenario auch eine detaillierte Beschreibung einer Produktkomponente mit ihren Eigenschaften und Funktionen sowie benachbarten Komponenten darstellen. Dieser Fokus ist vom Anwender frei wählbar.

**3.3 Das 3-stufige Produktmodell**

Der Einsatz der modellbasierten Kontextbeschreibung in der Produktentwicklung erfolgt in drei aufeinander aufbauenden Stufen, die die zunehmende Konkretisierung der Entwicklungsobjekte widerspiegeln (Bild 3):

- *Definitionsebene* – auf dieser Ebene werden allgemeingültige Zusammenhänge für die Produktpalette eines Unternehmens

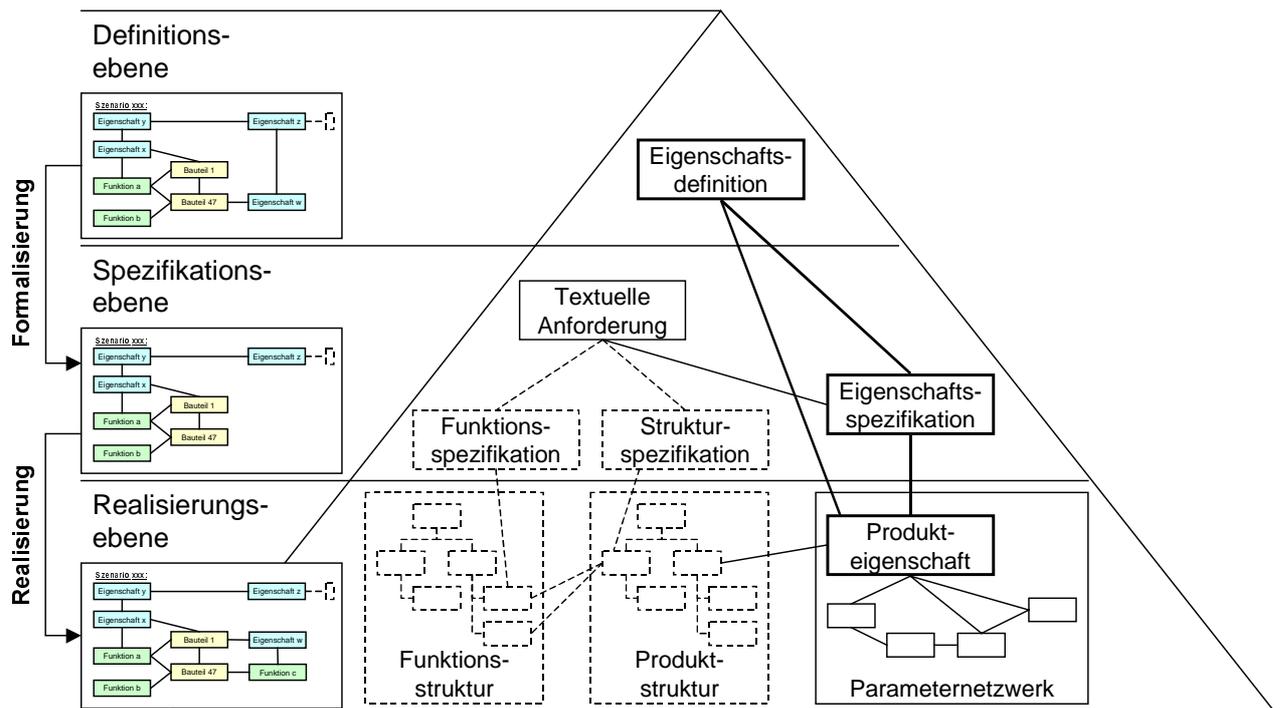
beschrieben

- *Spezifikationsebene* – ausgehend von den Anforderungen wird mithilfe der definierten Szenarien die Spezifikation des Produkts erarbeitet und mit dem Kunden abgestimmt
- *Realisierungsebene* – auf dieser Stufe erfolgt die Verwaltung konkreter Produkte; sie dient außerdem dem Management der Produktentwicklung und der Kommunikation zwischen den Entwicklungspartnern

Zusammen mit der Anforderungsebene, auf der Kundenanforderungen als eigenständige Objekte verwaltet werden können, ergibt sich bei der Anwendung des 3-Stufen-Modells eine lückenlose Dokumentation des Entwicklungsprozesses und der dabei getroffenen Entscheidungen. Zudem lässt sich durch die Aufteilung der unterschiedlichen Ebenen jederzeit nachvollziehen, welche Anforderung des Kunden in welche Spezifikation übergegangen ist, und wie diese dann in eine konstruktive Lösung umgesetzt wurde. Durch Rückkopplungen lässt sich dabei auch nachvollziehen, welchen Einfluß die Erarbeitung unterschiedlicher Lösungskonzepte auf die Produktspezifikation hat.

**3.3.1 Die Definitionsebene**

Auf der Definitionsebene werden allgemeingültige Zusammenhänge innerhalb des Produktspektrums eines Unternehmens beschrieben. Dieses dient der



**Bild 3:** 3-stufiges Produktmodell

Klassifizierung von Entwicklungsobjekten, um damit die Grundlage für das Wissensmanagement über vorhandene (Teil-)Lösungen bereits bearbeiteter Problemstellungen zu schaffen.

Die Kontextbeschreibung hält fest, welche generellen Angaben bei der Bearbeitung eines Problems notwendig sind (Komponenten, Haupteigenschaften, Hauptfunktionen). Diese Angaben dienen in den nachgelagerten Phasen der Kontrolle auf Vollständigkeit, sowie als Leitfaden, welche entwicklungstechnischen Aspekte bei der Ausarbeitung zu berücksichtigen sind. Des Weiteren ergibt sich aus der Zuordnung von Spezifikationen bzw. Realisierungen zu Definitionselementen ein Wissenspool, welche Produkte oder Lösungen bereits entwickelt wurden.

### 3.3.2 Die Spezifikationsebene

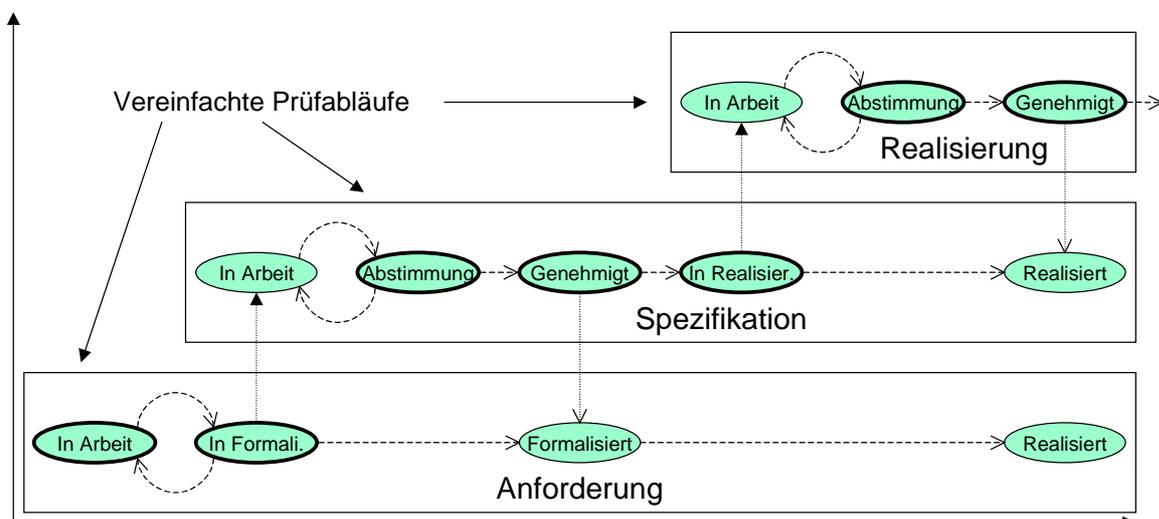
Die projektbezogene Arbeit an Entwicklungsobjekten beginnt mit der Erstellung einer modellbasierten Spezifikation des zu entwickelnden Produkts. Dazu werden Anforderungen, die vom Kunden meistens in textueller Form oder als Listen präsentiert werden, formalisiert. Mit der Formalisierung werden zwei Ziele verfolgt. Zum einen werden natürlichsprachliche Anforderungen, die oft widersprüchlich und mehrdeutig sein können, in eine möglichst eindeutige Beschreibung überführt. Zum anderen beinhalten Kundenanforderungen häufig implizite Informationen (Annahmen oder Voraussetzungen, die nicht offensichtlich formuliert sind), die durch eine Formalisierung explizit aufgezeigt werden sollen (siehe hierzu auch /4/). Während der Formalisierung werden die Kundenanforderungen in funktionale,

komponentendefinierende und eigenschaftsbeschreibende Entwicklungsobjekte unterteilt. Zusammenhänge zwischen den Objekten werden über Kontextbeschreibungen abgebildet. Hilfreich sind dazu die auf der Definitionsebene modellierten Kontextbeschreibungen, da sie Vorgaben über erforderliche Informationen enthalten. Dadurch lassen sich fehlende Anforderungen aufdecken und in einer workflowunterstützten Diskussion mit dem Kunden klären.

Die modellbasierte Spezifikation auf Basis von Entwicklungsobjekten führt zudem dazu, dass Widersprüche in der Anforderungsliste durch die Referenzierung gleicher Objekte aufgedeckt werden können. Jedes definierte Objekt steht dem gesamten Entwicklungsteam zur Verfügung. Dadurch wird bspw. sichergestellt, dass eine einmal spezifizierte Eigenschaft nicht mit unterschiedlichen Werten belegt werden kann. Die notwendige Kontrolle ergibt sich durch die Klassifizierung über die Definitionsebene.

### 3.3.3 Die Realisierungsebene

Während die Spezifikationsebene das Entwicklungsziel festhält, werden auf der Realisierungsebene alle entwicklungsrelevanten Daten verwaltet, d. h. auf dieser Ebene werden die tatsächlich realisierte Produktstruktur, die vom Produkt erfüllten Funktionen und die aktuelle Ausprägung der Eigenschaften festgehalten. Ausgangspunkt dazu sind die Kontextbeschreibungen und Entwicklungsobjekte der Spezifikation, die im Zuge der Entwicklung instanziiert und konkretisiert werden. Bei Bedarf



**Bild 4:** Abgestimmte Statusnetzwerke zur Koordination der Kommunikation

können dem Produktmodell auch weitere Objekte hinzugefügt werden. Das Ergebnis ist eine virtuelle Beschreibung des Produkts.

Die Trennung der Spezifikations- und Realisierungsebene ermöglicht in Bezug auf die Anforderungsverfolgung jederzeit einen Soll/Ist-Vergleich. Außerdem lassen sich in der Phase der Lösungsfindung unterschiedliche Lösungen hinsichtlich des Erfüllungsgrades von Anforderungen miteinander vergleichen.

#### 4 Erweiterter Engineering Workflow

Die Aktivitäten innerhalb des 3-stufigen Produktmodells werden über einen Engineering Workflow gesteuert, der auf der einen Seite die Anforderungsanalyse unterstützt, und auf der anderen Seite die Entwicklungsaktivitäten auf Basis der vernetzten Entwicklungsobjekte kontrolliert. Dadurch lässt sich die gesamte Produktentwicklung ausgehend von der Anforderungsentwicklung über die Erstellung der Spezifikation bis zur Ausarbeitung des Produkts steuern.

Die Koordination zwischen den einzelnen Phasen basiert auf aufeinander abgestimmten Prüfabläufen für die Entwicklungsobjekte in den unterschiedlichen Lebensphasen des Produkts

(siehe Bild 4). Durch die Unterteilung des Produktmodells in drei Ebenen ist es außerdem möglich, Verantwortlichkeiten gezielt für bestimmte Phasen des Produktlebenszyklus zu definieren.

#### 4.1 EWF in der Anforderungsentwicklung

Der Konstruktionsprozess nach Pahl/Beitz oder VDI 2221 stellt zwar die Klärung der Aufgabenstellung mit der Anforderungsliste als Ergebnis an den Anfang der Produktentwicklung. Gleichzeitig wird jedoch darauf hingewiesen, dass diese Liste stets auf dem neuesten Stand zu halten ist, also Änderungen auch während der Entwicklung auftreten können. Zudem ist zu berücksichtigen, dass die Anforderungsentwicklung selbst ein iterativer Prozess ist. Suzanne und James Robertson stellen in /5/ den so genannten Volere Requirements Process zur Anforderungsentwicklung vor (siehe Bild 5). Dieser sehr generisch gehaltene Prozess eignet sich sowohl für die Softwareentwicklung als auch für die Konstruktion. Basierend auf ‚Snow-Cards‘ wird der Kundenwunsch, die Quelle, der Grund, die Prüfkriterien usw. schriftlich festgehalten. Bevor eine Anforderung in die Anforderungssammlung eingeht, muss sie einen ‚Quality Gateway‘ passieren. Dabei wird geprüft, ob die Anforderung formalen Kriterien genügt, wie z. B. Vollständigkeit, Widerspruchsfreiheit, Eindeutigkeit, etc.

In Bezug auf den Engineering Workflow werden im PLM-System eigenständige Anforderungsobjekte definiert, die die DV-technische Umsetzung der ‚Snow-Card‘ darstellen. Jeder Anforderung wird jeweils ein verantwortlicher Mitarbeiter des Lieferanten und des Kunden zugeordnet, die dann bei Änderungen oder Rückfragen als Ansprechpartner zur Verfügung stehen. Des Weiteren muss eine Person festgelegt werden, die die Rolle des ‚Quality Gateways‘ übernimmt. Da diese Zuordnung für eine Vielzahl der Anforderungen gleich ist, genügt für die Ablaufsteuerung ein klassischer Workflow, der für einzelne oder Gruppen von Anforderungen gestartet wird. Am Ende der Anforderungsentwicklung liegt eine strukturierte Anforderungssammlung im System vor, die als Ausgangspunkt für

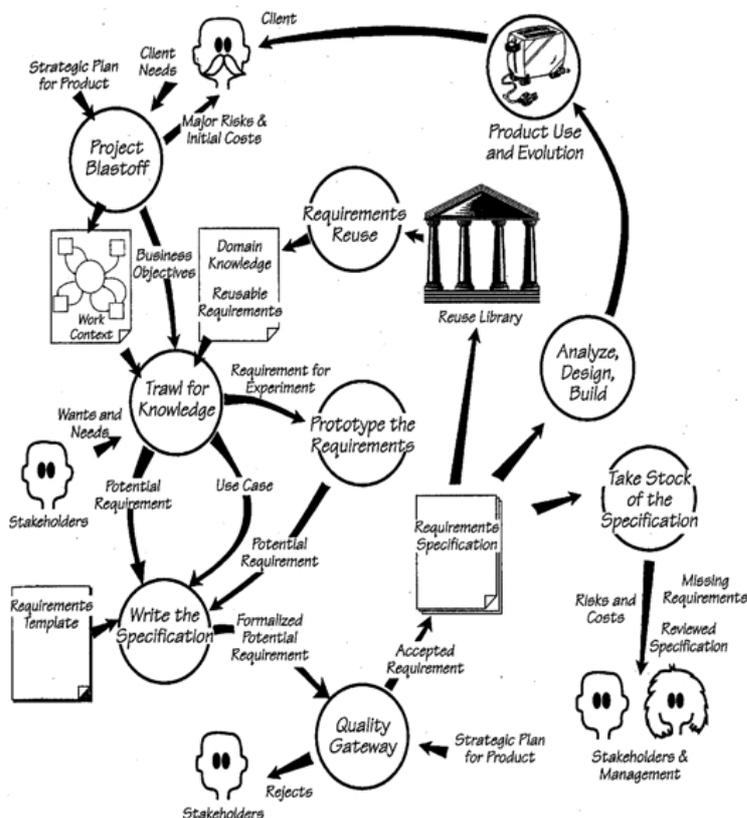


Bild 5: Volere Prozess (aus /5/)

den nachfolgenden Formalisierungsprozess dient.

#### 4.2 Abstimmung der Spezifikation

Mithilfe der Formalisierung werden die natürlichsprachlichen Anforderungen in eine objektorientierte Spezifikationen überführt. Damit werden zwei Ziele verfolgt – die Klärung von Widersprüchen oder zusätzlichen Anforderungen und die formelle Anerkennung der Spezifikation als Vertragsgrundlage für die folgende Entwicklung.

Durch das objektorientierte Spezifikationsmodell auf Basis der Kontextbeschreibungen der Definitionsebene lassen sich Widersprüche und fehlende Angaben leicht aufdecken. Durch eine Rückkopplung zum Anforderungswork-flow lassen sich diese mit den Ansprechpartnern des Kunden klären.

Auf der anderen Seite bilden die Entwicklungsobjekte und Kontextbeschreibungen der Spezifikationsebene die Kommunikationsplattform zwischen Kunden und Lieferanten sowie zwischen Lieferant und Unterauftragnehmer. Dazu werden spezifizierte Entwicklungsobjekte in einem Abstimmungs- und Freigabeworkflow an alle betroffenen Personen geschickt, die dem aktuellen Stand zustimmen oder widersprechen können. Die betroffenen Personen ergeben sich aus ihrer Zuordnung zu einem Entwicklungsobjekt innerhalb einer definierten Benutzerkategorie. Dadurch wird einerseits sichergestellt, dass der Auftragnehmer die Kundenwünsche richtig verstanden hat, andererseits wird die Spezifikation nach erfolgter Genehmigung durch alle Vertragspartner eingefroren und lässt sich nur durch ein entsprechendes Änderungsweisen an neue Anforderungen anpassen.

#### 4.3 Steuerung der Produktentwicklung

Mithilfe der Entwicklungsobjekte auf der Realisierungsebene lassen sich die Entwicklungsaktivitäten und die dazu notwendige Kommunikation zwischen den Entwicklungspartnern gezielt steuern. Dabei kann es sich sowohl um verschiedene Abteilungen innerhalb eines Unternehmens handeln, als auch um externe Engineering Partner. Die durch Kontextbeschreibungen vernetzten Entwicklungsobjekte definieren dabei den Gesamtprozess. Die Ausrichtung des Netzes erfolgt

durch eine schrittweise Bearbeitung der Entwicklungsobjekte in Abhängigkeit von den vorhandenen Beziehungselementen. Der Stand der Entwicklung wird dabei durch den Härtegrad des Objekts wiedergespiegelt (1 – erste Annahme getroffen, 5 – Objekt festgelegt und auskonstruiert, weitere Änderungen sind nur durch einen formelles Änderungsablauf möglich).

Ausgangspunkt der Entwicklung sind vordefinierte Objekte (Härtegrad 5), wie z. B. Bauteile und Baugruppen, die ohne Änderung wieder verwendet werden oder Eigenschaften, die durch Anforderungen vordefiniert sind. Aus diesen lassen sich erste Konzepte für weitere Objekte ableiten, die von den vordefinierten Elementen abhängig sind. Ist dieses nicht möglich, muss einer der verantwortlichen Bearbeiter eine erste Annahme zu einem Objekt treffen, so dass diese als Eingangsinformation für andere Entwicklungsobjekte dienen kann. Die zunehmende Konkretisierung der Konstruktionsergebnisse spiegelt sich in der Erhöhung des Härtegrads wider, und muss durch die zugeordneten Personen genehmigt werden. Dies geschieht über einen Genehmigungs- und Freigabeworkflow auf Basis des betroffenen Entwicklungsobjekts.

### 5 Wissensmanagement mithilfe strukturierter Entwicklungsobjekte

Wie bereits in den vorangegangenen Abschnitten erwähnt, ist eine Grundlage für das Wissensmanagement die Klassifikation der Entwicklungsobjekte über die Definitionsebene. Die Kontextbeschreibung ermöglicht zudem eine Einordnung des Wissens in Bezug auf die Randbedingungen und getroffenen Annahmen.

Eine weitere Möglichkeit ergibt sich aus einer erweiterten Integration zwischen PLM- und CAD-Systemen, die vor allem die Abstimmung geometrischer Eigenschaften und ihrer Beziehungen untereinander betrifft. Dazu werden bereits auf der Definitionsebene Entwicklungsobjekte mit den Wissensselementen des dazugehörigen CAD-Modells verknüpft. In der Realisierungsphase können dann die mit Werten belegten Eigenschaften über eine erweiterte Schnittstelle an das CAD-Modell übergeben werden. Eine bidirektionale Ankopplung ermöglicht außerdem die Aktualisierung der Werte im PLM-System, wenn diese im CAD-Modell geändert wurden. Diese Informationen können dann

wiederum zur Steuerung des Entwicklungsprozesses herangezogen werden. Außerdem lässt sich so die Konsistenz der Informationen auch über Systemgrenzen hinweg sicherstellen.

## 6 Literatur

- /1/ Workflow Management Coalition: Terminology & Glossary: Winchester: 1999
- /2/ Kruse, P. J.: Anforderungen in der Systementwicklung : Erfassung, Aufbereitung und Bereitstellung von Anforderungen in interdisziplinären Entwicklungsprojekten: Fortschritt-Berichte ; 191. Als Ms. gedr. Auflage. Düsseldorf: VDI-Verl., 1996
- /3/ Fowler, M. ; Scott, K.: UML - konzentriert : die Standardobjektmodellierungssprache anwenden: 1. Aufl., 1., korrigierter Nachdr. Auflage. Bonn [u.a.]: Addison-Wesley, 1998
- /4/ Heimannsfeld, K.: Modellbasierte Anforderungen in der Produkt- und Systementwicklung : von Dokumenten zu Modellen: Berichte aus dem Maschinenbau. Aachen: Shaker, 2001
- /5/ Robertson, S. ; Robertson, J.: Mastering the requirements process: Harlow [u.a.]: Addison-Wesley, 1999