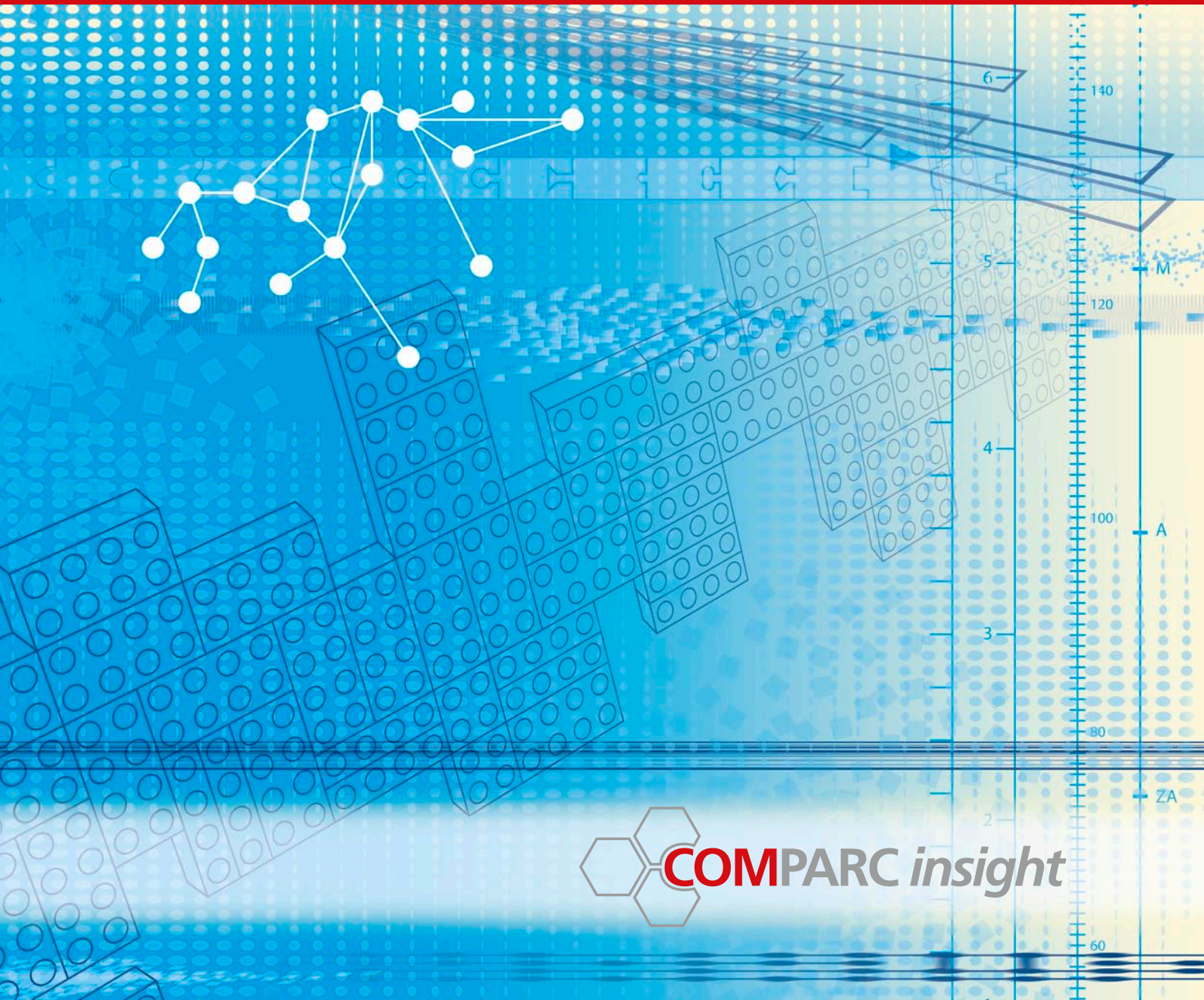


**ANFORDERUNGSMANAGEMENT FÜR
GESCHÄFTSANWENDUNGEN
EIN INTEGRIERTES ANALYSE-SYNTHESE-KONZEPT
WHITE PAPER**



COMPARC *insight*

Im Rahmen der COMPARC *insight*-Publikationen gibt das Fraunhofer ISST Einblicke in aktuelle Forschungsarbeiten und Studien des »Competence Center for Processes and Architectures« (COMPARC), einem Entwicklungs- und Demonstrationslabor für prozessorientierte Architekturen. Insbesondere Personen in leitenden Positionen können sich mithilfe dieser überblicksartigen Publikationen ohne wissenschaftliches Vorwissen über grundlegende Fragen der anwendungsorientierten Forschung informieren.

Fraunhofer-Institut für Software- und Systemtechnik ISST

Institutsteil Berlin
Steinplatz 2
10623 Berlin

Autor

Dipl.-Inform. Oliver Boehm
Telefon 030 24306-347
Fax 030 24306-599
oliver.boehm@isst.fraunhofer.de

www.isst.fraunhofer.de

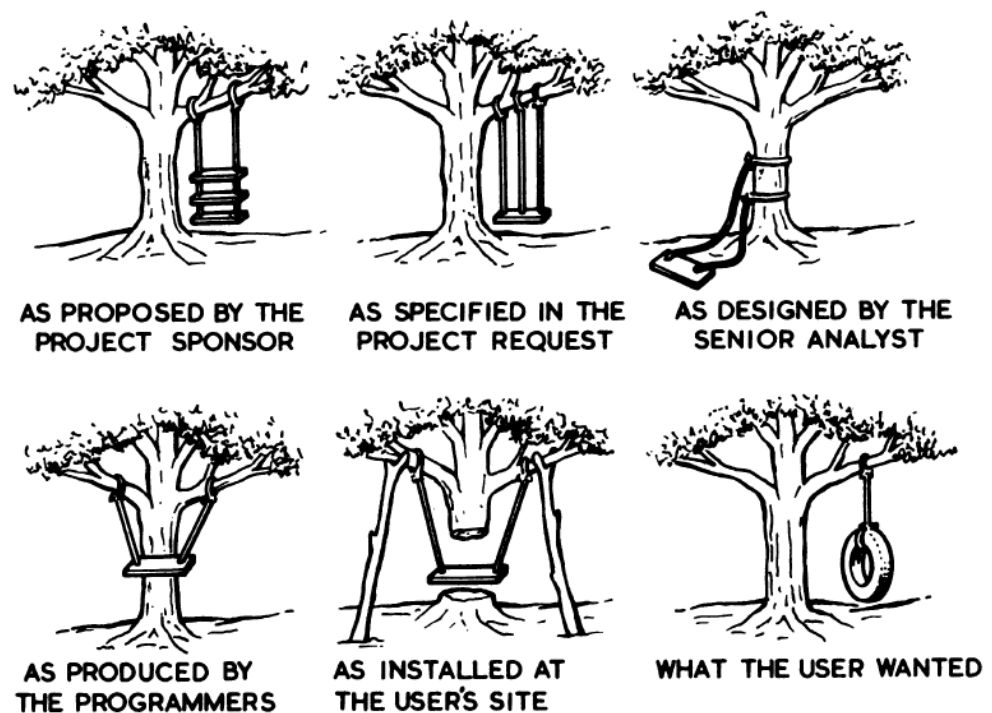
Inhalt

1	Einleitung	3
2	Motivation	5
3	Generelle Sichtweisen	7
4	Ein integriertes Vorgehen	10
4.1	Sichtweise »Forward-Engineering«	11
4.1.1	Geschäftsanalyse	11
4.1.2	Anwendungssynthese bzw. Anwendungsentwurf	13
4.2	Sichtweise »Reverse-Engineering«	14
4.2.1	Anwendungsanalyse	14
4.2.2	Geschäftssynthese	15
5	Funktionale, nicht-funktionale Anforderungen und Rahmenbedingungen	16
6	Aufbereitung der Ergebnisse als Lastenheft	18
7	Model Driven Architecture und V-Modell XT	21
8	Literatur	24

1 Einleitung

In Projekten zur Realisierung von Geschäftsanwendungen ist die Kommunikation zwischen den Beteiligten oft ein Kernproblem während der verschiedenen Projektphasen. Das klassische »Tire Swing Cartoon« in Abbildung 1 [Britt80] verdeutlicht die Notwendigkeit eines gemeinsamen Verständnisses aller Beteiligten, um ein IT-Projekt zum Erfolg zu führen. Es zeigt auch, wie wichtig eine aussagekräftige Spezifikation der Kunden- bzw. Nutzeranforderungen zu Beginn eines jeden IT-Projekts ist.

Abbildung 1 »Tire Swing Cartoon«



Die Abteilung SBI des Fraunhofer-Instituts für Software- und Systemtechnik ISST begleitet bereits seit einigen Jahren zahlreiche Entwicklungsprojekte im privatwirtschaftlichen und öffentlichen Bereich. Aufgabenschwerpunkte sind die Anforderungsanalyse, die Erstellung der notwendigen Ausschreibungsunterlagen wie Lastenheft und Fragenkataloge sowie die Durchführung der Ausschreibung. Dazu gehören auch die Bewertung von Angeboten und Anbietern sowie die Begleitung von Anbietergesprächen. Während der Projektdurchführung übernehmen Mitarbeiter der Abteilung auch

Aufgaben wie Projektkoordination, Entwurf bzw. Konzeption, begleitende Qualitätssicherung, Test und Abnahme.

Das vorliegende Dokument motiviert und beschreibt ein methodisches Vorgehen zur Spezifikation von Anforderungen an Geschäftsanwendungen und zur Erstellung von Lastenheften, das in verschiedenen Projekten erfolgreich zum Einsatz kam. Die Darstellung soll Entscheidern zu einem Einblick in die Philosophie des Vorgehens verhelfen.

Im allgemeinen Sprachgebrauch umreißt der Begriff »Anforderungsanalyse« die Erhebung, Analyse, Spezifikation und Überprüfung von Anforderungen. Anlass einer solchen Analyse ist in der Regel, der Wunsch nach der Entwicklung einer neuen IT-Anwendung oder nach der Anpassung einer existierenden. Ursache hierfür ist meist die mangelnde IT-Unterstützung der Geschäftsprozesse einer Organisation. Ziel der Anforderungsanalyse ist zunächst die nachvollziehbare und verifizierte Spezifikation von Eigenschaften und Funktionen der unterstützenden IT-Anwendung und ihrer Datenstrukturen. Diese Spezifikation ist Ausgangspunkt für den anschließenden Entwurf der Grobarchitektur der Anwendung mit ihren logischen Komponenten. Anforderungen und Grobarchitektur dienen als Grundlage für die Abschätzung des Realisierungsaufwands und als Leistungsbeschreibung für die Ausschreibung. Letzteres gilt insbesondere für den öffentlichen Bereich, da dieser gesetzlich verpflichtet ist, Entwicklungsprojekte ab einem bestimmten finanziellen Umfang öffentlich auszuschreiben. Das dort verbindlich anzuwendende V-Modell [V-Modell] definiert einen allgemeinen Rahmen für die Abwicklung derartiger Entwicklungsprojekte. Dieser legt z. B. die Rollen, Aktivitäten und Produkte im Entwicklungsprozess fest. Die Anforderungen bzw. das Lastenheft dienen im V-Modell als Schnittstellendokument zwischen Auftraggeber und Auftragnehmer bei der Realisierung der IT-Anwendung. Das Lastenheft wird zum vertraglichen Bestandteil und bildet die Grundlage des Pflichtenheftes.

2 Motivation

Die in einem Lastenheft spezifizierten Anforderungen müssen verschiedene Aufgaben erfüllen. Sie dienen generell der Kommunikation zwischen allen Projektbeteiligten. Außerdem bilden sie die Grundlage für Ausschreibungen und Verträge, für die Gestaltung der Systemarchitektur einer IT-Anwendung, für ihre Integration in eine bestehende IT-Infrastruktur sowie bei Änderungen und Erweiterungen. Aus diesen vielfältigen Aufgaben ergeben sich verschiedene Qualitätskriterien für Anforderungen [Rupp02].

Anforderungen müssen

- aktuell gültig,
- vollständig,
- eindeutig,
- korrekt und für alle verständlich,
- in sich konsistent und nachvollziehbar,
- bewert-, überprüf- und testbar die geforderte Funktionalität beschreiben.

Zudem sollten Anforderungen hinsichtlich ihrer juristischen Verbindlichkeit klassifizierbar sein. Eine unstrukturierte prosaische Formulierung der Anforderungen kann diese Qualitätskriterien schwer erfüllen. Zu groß ist in der Regel der Interpretationsspielraum der Formulierungen. Selbst die schematische Formulierung von Anforderungen [Rupp02, S. 233 ff] wie z. B.

»Wenn ein Nutzer innerhalb eines Ausleihvorgangs einen Bibliothekskunden auswählt, soll das Bibliothekssystem diesem Nutzer den Namen des Kunden, die Adresse des Kunden und den augenblicklichen Kontostand des Kunden anzeigen«

fassen oft viele Informationen wie den Kontext bzw. die Bedingung der Anforderung, ihre Verbindlichkeit, den Akteur, den Gegenstand der Anforderung und die geforderte Funktion zusammen. Eine Überprüfung derart formulierter Anforderungen hinsichtlich der genannten Qualitätskriterien ist nur »händisch« möglich. Bei der Komplexität heutiger Geschäftsanwendungen ist dies nicht mehr zu bewältigen. Um die geforderten Qualitätskriterien erfüllen zu können, ist deshalb eine strukturierte und methodische Vorgehensweise notwendig.

Das im Weiteren vorgestellte Vorgehen folgt den Prinzipien eines konsequenten Analyse-Synthese-Konzepts wie es Kosiol in [Kos68] für die Aufgabenanalyse und -synthese in Unternehmen vorstellt. Weiterhin bedient es sich der Ergebnisse aus Arbeiten zur objektorientierten Geschäftsprozessanalyse und

Softwareentwicklung [Oest03, Oest04] sowie zum Einsatz von Anwendungsfällen [Cock00].

Verschiedene Modelle, die in formalen Modellierungssprachen definiert sind, halten durchgängig die Ergebnisse aus den einzelnen Schritten der Analyse und Synthese fest. Diese Modelle beschreiben die für die Realisierung einer Geschäftsanwendung relevanten Aspekte auf einem abstrakten Niveau. Dazu gehören z. B. Akteure, Prozesse, Datenstrukturen, Komponenten und ihre Schnittstellen, Protokolle über diese Schnittstellen und Zustandsänderungen. Modellierungssprachen wie die Unified Modeling Language [UML] unterstützen zwar bestimmte Analyse- und Entwurfparadigmen, sind aber prinzipiell offen für verschiedene Optionen der Programmierung und der Ausführungsplattform.

Formale Modellierungssprachen sind genauso wie Maschinen- und Programmiersprachen künstliche Sprachen für einen dedizierten Zweck. Sie haben gegenüber den natürlichen Sprachen den Vorteil, dass sie nicht vage und mehrdeutig sind. Zu diesen Sprachen existiert eine exakte Definition der zulässigen Ausdrücke und ihrer Bedeutungen. Diese Eigenschaften ermöglichen es, mit diesen Sprachen formulierte »Aussagen« automatisiert zu prüfen und z. B. in Spezifikationsdokumente, Schnittstellendefinitionen, Datenschemata und Programmartefakte zu transformieren. Ein durchgängig modellbasiertes Vorgehen bereits bei der Anforderungsanalyse ermöglicht so einen deutlichen Produktivitäts- und Qualitätsgewinn.

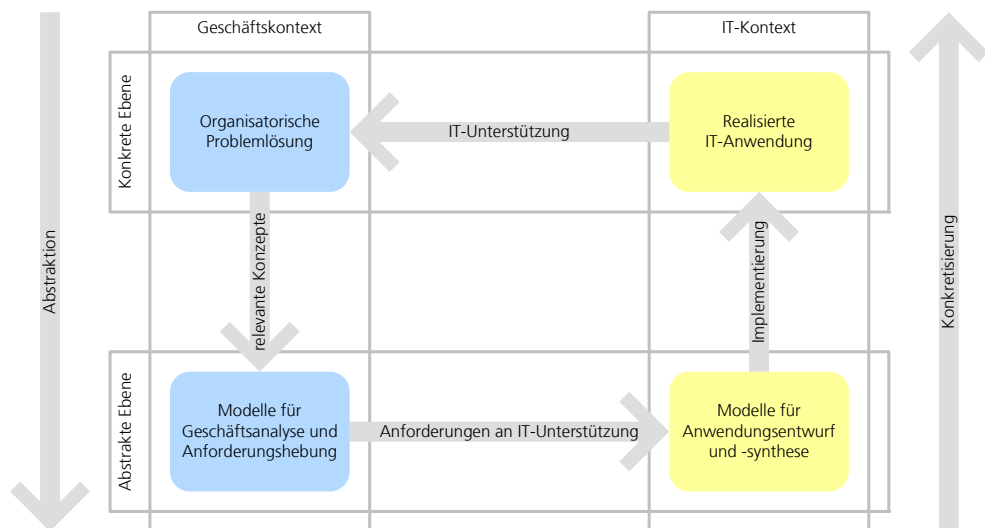
Die folgenden Abschnitte geben einen Überblick zum Vorgehen. Erfahrungen aus allen Entwicklungsphasen verschiedener Projekte, von der Anforderungsanalyse bis zur Abnahme, flossen in dieses Vorgehen ein.

3 Generelle Sichtweisen

In der Regel unterhalten Unternehmen oder Behörden heute eine IT-Infrastruktur, in der sie ihre Geschäftsanwendungen betreiben. Die Entwicklung einer neuen Anwendung »auf der grünen Wiese« ist eher selten. Häufiger ist eine Integration der neuen Anwendung in die bestehende Infrastruktur erforderlich. Dies ist i. d. R. mit der Nutzung existierender Anwendungen oder Systeme verbunden. Dadurch sind Realisierungsoptionen und einzusetzende Technologien von vornherein eingeschränkt. Diese Rahmenbedingungen können zunächst Auswirkungen auf den logischen Entwurf der Anwendung bis hin zu den realisierbaren Anforderungen und der organisatorischen Problemlösung haben. Gleiches gilt beim Einsatz von Standardsoftware, die nur beschränkt anpassbar ist. Aus diesem Grund unterstützt das Vorgehen zwei generelle Sichtweisen: das »Forward Engineering« und das »Reverse Engineering«. Beide Sichtweisen sind auch als SOLL- und IST-Analyse bekannt.

Die folgende Abbildung stellt zunächst die Sichtweise »Forward-Engineering« dar.

Abbildung 2 Sichtweise »Forward-Engineering«



Ausgangspunkt dieser Sichtweise ist die Lösung einer konkreten organisatorischen Problemstellung zur Erfüllung eines bestimmten Geschäftszwecks. Diese Lösung ist in der Aufbau- und Ablauforganisation eines Unternehmens oder einer Behörde umgesetzt. Zur Aufbauorganisation gehören u. a. Organisationseinheiten, deren Struktur, Mitarbeiter und Kunden.

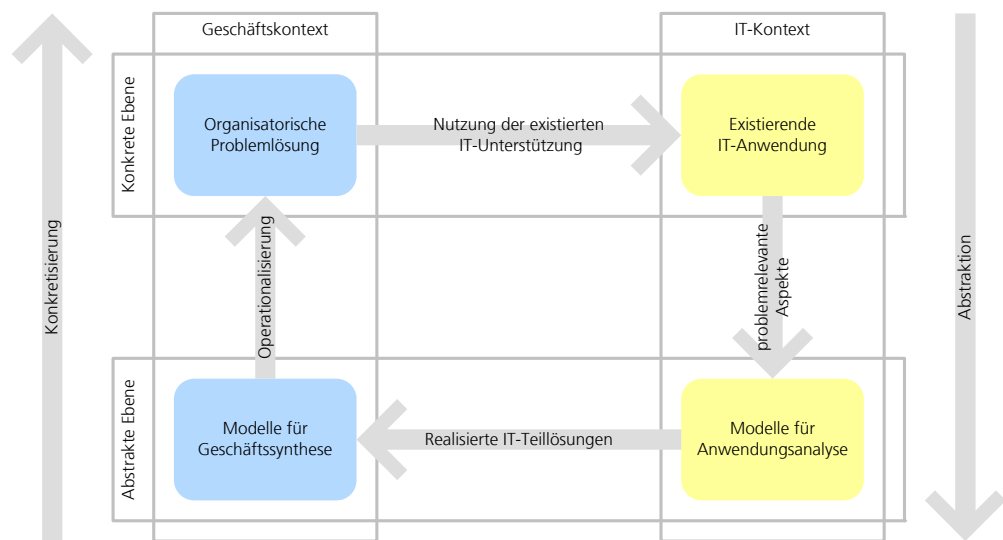
Geschäftsprozesse und Geschäftsanwendungsfälle beschreiben die Ablauforganisation. Gegenstand der Geschäftsanalyse ist die Zerlegung und Modellierung aller Dinge und Sachverhalte, die für die Erhebung der Anforderungen an eine IT-technische Unterstützung relevant sind. Die Modelle der Geschäftsanalyse sind eine abstrakte, strukturierte Beschreibung aller Teile der konkreten organisatorischen Problemlösung. Die strukturierte Beschreibung dieser Teile ermöglicht die Identifizierung und direkte Zuordnung von Anforderungen an eine IT-technische Unterstützung.

Die erhobenen Anforderungen sind nun Grundlage für den logischen Entwurf der IT-Anwendung. Die Modelle des Anwendungsentwurfs beschreiben Daten, Anwendungsfälle, Komponenten und ihre Relationen. Sie ermöglichen die schrittweise Synthese der einzelnen Bestandteile zu einer Gesamtanwendung.

Ergebnis der Implementierung des logischen Entwurfs ist eine konkrete IT-Anwendung, die die geforderte Unterstützung bei der organisatorischen Problemlösung leistet. Bei Entwurf und Implementierung sind Realisierungsoptionen und einzusetzende Technologien zu prüfen.

Abbildung 3

Sichtweise »Reverse-Engineering«



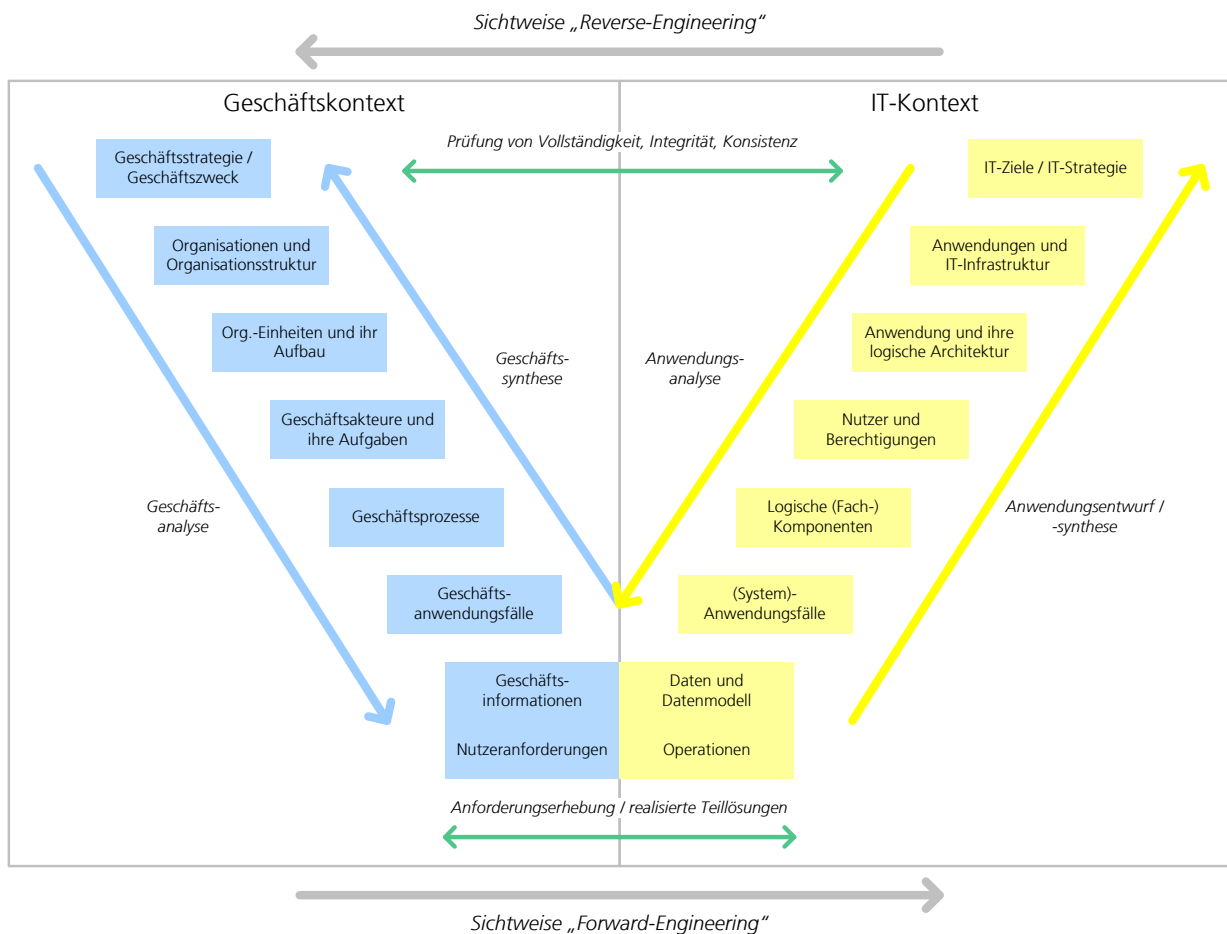
In der Sichtweise »Reverse-Engineering« ist eine vorhandene IT-Infrastruktur mit ihren Anwendungen Ausgangspunkt der Betrachtung. Oft sind diese unzureichend beschrieben, so dass zunächst eine Analyse der Anwendungen notwendig wird. Gegenstand der Anwendungsanalyse sind lediglich die problemrelevanten Aspekte. Das Ergebnis sind Modelle der realisierten Teillösungen der IT-Anwendungen wie z. B. die Möglichkeiten der Verwaltung bestimmter Daten. Die Modelle im Geschäftskontext ermöglichen nun die

Synthese und Operationalisierung der realisierten Teillösungen zu einer möglichen Gesamtlösung einer organisatorischen Problemstellung. Diese Gesamtlösung ist maßgeblich an die Unterstützung durch die realisierte IT-Anwendung angepasst. Ist die organisatorische »Anpassungsfähigkeit« gegeben und erfüllt die unterstützte organisatorische Problemlösung hinreichend den Geschäftszweck kann der Einsatz einer bestehenden IT-Anwendung bzw. einer Standardsoftware für diese Zwecke unter einer Kosten-Nutzen-Abwägung durchaus sinnvoll sein. Ist dies nicht gegeben, ist eine Anpassung oder ein Ersatz der IT-Anwendung erforderlich.

4 Ein integriertes Vorgehen

Die beiden generellen Sichtweisen erfüllen verschiedene Zielstellungen. Bei jeder neu zu entwickelnden IT-Anwendung in einer bestehenden IT-Infrastruktur bzw. IT-Anwendungslandschaft ist es unter dem Kosten-Nutzen-Gesichtspunkt sinnvoll, beide Sichtweisen zu integrieren. Erfüllen die bestehenden IT-Anwendungen unter Berücksichtigung möglicher organisatorischer Anpassungen die geforderten Anforderungen, ist die Identifikation dieser Anforderungen notwendig. Die folgende Abbildung verdeutlicht die Integration beider Sichtweisen.

Abbildung 4 Integration der Sichtweisen Reverse- und Forward-Engineering



Die Abbildung stellt die abstrakten Konzepte in den Modellen zur Analyse und Synthese im Geschäfts- und IT-Kontext auf verschiedenen Granularitätsstufen gegenüber. Grobgranulare eher unspezifische Konzepte wie Ziele und Strategien sind oben angesiedelt und unten die feingranularen Konzepte wie Daten und konkrete Anforderungen bzw. Operationen. Grobgranulare Konzepte sind Ausgangspunkt der Analysen und feingranulare Ausgangspunkt der Synthesen. Der Übergang zwischen Geschäfts- und IT-Kontext ist explizit auf der Basis feingranularer Konzepte wie Geschäftsinformationen und Nutzeranforderungen bzw. Daten und Operationen möglich. Diese feingranularen Konzepte sind in die Konzepte der nächst größeren Granularitätsstufe eingebettet und sind somit immer mit ihrem Kontext verknüpft. Infolgedessen existieren auf jeder größeren Granularitätsstufe implizite Zuordnungen, die jeweils zur Prüfung der Vollständigkeit, Integrität und Konsistenz herangezogen werden können.

4.1 Sichtweise »Forward-Engineering«

In der Sichtweise des »Forward-Engineering« basiert der Anwendungsentwurf bzw. die Anwendungssynthese auf einer Anforderungserhebung, die der Geschäftsanalyse folgt. Die folgenden Abschnitte beschreiben die dazu verwendeten abstrakten Konzepte aus dieser Sichtweise.

4.1.1 Geschäftsanalyse

Dieser Abschnitt erläutert die Konzepte im Geschäftskontext, die Gegenstand der Geschäftsanalyse sind. Ausgehend von den Geschäftsstrategien und dem zu unterstützenden Geschäftszweck bzw. der Aufgabenstellung ist die Definition des Analysefokus zu Beginn einer jeden Geschäftsanalyse von grundlegender Bedeutung. Seine Festlegung hilft eine Grenze um den Kern der Analyse zu ziehen. Er schafft einerseits allen Beteiligten einen Raum des gemeinsamen Verständnisses und verhindert andererseits das Abdriften in Diskussionen, die für das Ergebnis der Analyse keinen Wert beitragen. Die Festlegung des Analysefokus schließt gleichzeitig die Existenz angrenzender Konzepte wie Organisationen, Organisationseinheiten, Akteure und Stakeholder ein, die für die weitere Betrachtung relevant sind. So kann eine Organisationseinheit beispielsweise innerhalb aber auch außerhalb des Analysefokus liegen. Trotzdem muss sie auf Grund ihrer Relevanz in die Analyse einbezogen werden. Die zu unterstützende organisatorische Problemlösung, die relevanten Stakeholder, deren Ziele sowie die übergreifenden Regeln sind im Weiteren Gegenstand der Geschäftsanalyse. Ihre Festlegung ermöglicht die Bestimmung und Strukturierung der relevanten Geschäftsprozesse, Organisationseinheiten und Geschäftsakteure.

Die Geschäftsakteure sind im nächsten Schritt Ausgangspunkt für die Erhebung ihrer Geschäftsanwendungsfälle. Eine grundsätzliche Unterscheidung der Geschäftsakteure in Geschäftspartner und Geschäftsmitarbeiter ist dabei hilfreich. Geschäftspartner sind natürliche oder juristische Personen, die in angrenzenden Organisationseinheiten außerhalb des Analysefokus angesiedelt sind. Ein Geschäftsmitarbeiter ist als Person, in einer Organisationseinheit innerhalb des Analysefokus tätig.

Ein Geschäftsanwendungsfall fasst geschäftliche Aktivitäten der Geschäftsmitarbeiter einer Organisation oder Organisationseinheit zusammen. Ein Ereignis löst einen solchen Geschäftsanwendungsfall aus. Er endet mit einem Ergebnis. Dieses Ergebnis stellt für den Geschäftszweck der Organisation direkt oder indirekt einen Wert dar. Die Aktivitäten eines Geschäftsanwendungsfalles stehen in der Regel in zeitlichen und logischen Abhängigkeiten zueinander. Abläufe zu den Geschäftsanwendungsfällen können diese Abhängigkeiten detailliert darstellen. Die Geschäftsanwendungsfälle beschreiben außerdem, welche Informationen für ihre Abarbeitung erforderlich sind.

Mit der Identifikation und initialen Beschreibung der Geschäftsanwendungsfälle jedes Geschäftspartners ist auch die Bewertung seiner Relevanz für die weitere Analyse möglich. Gleichzeitig erfolgt die Erhebung der Geschäftsmitarbeiter. Diese werden in außen- und innenorientierte Geschäftsmitarbeiter unterschieden. Erstere sind in der Regel für die Kern-Geschäftsanwendungsfälle der Geschäftspartner verantwortlich. Innenorientierte Geschäftsmitarbeiter können anhand der unterstützenden Geschäftsanwendungsfälle identifiziert werden. Die Strukturierung der Geschäftspartner und -mitarbeiter erfolgt im Akteursmodell.

Der Analysefokus beschreibt bereits die anfänglich identifizierten Stakeholder. Die Geschäftsakteure ergänzen diese. Jeder Geschäftsakteur, der unmittelbar oder mittelbar von der IT-Anwendung betroffen ist, ist ein potentieller Stakeholder.

Die beschriebenen Geschäftsanwendungsfälle können die Grundlage für die Überprüfung und Optimierung der Geschäftsprozesse bilden. Hierbei sind insbesondere Abhängigkeiten zwischen den Geschäftsanwendungsfällen und den Geschäftsakteuren Gegenstand der Untersuchung. Diese Abhängigkeiten gilt es zu minimieren, um unnötige Schnittstellen und den damit verbundenen Verbrauch an Zeit und finanziellen Mitteln einzuschränken. Wurden die Geschäftsprozesse zuvor noch nicht beschrieben, können die Geschäftsanwendungsfälle Ausgangspunkt für ihren Entwurf sein.

Die Geschäftsprozesse, ihre Geschäftsanwendungsfälle und Geschäftsakteure sind im nächsten Schritt Basis und Kontext bei der Erhebung der geforderten Unterstützung durch Informationstechnik. Das Geschäftsinformationsmodell

beschreibt die in den Geschäftsanwendungsfällen erforderlichen Informationen und ihre Relationen.

4.1.2 Anwendungssynthese bzw. Anwendungsentwurf

Der Übergang vom Geschäftskontext in den IT-Kontext erfolgt durch eine Strukturierung, Konsolidierung und Formalisierung der Geschäftsinformationen und der erhobenen Nutzeranforderungen. Dieser Abschnitt schließt an den vorherigen an und erläutert die Konzepte im IT-Kontext als Gegenstand des Anwendungsentwurfs bzw. der Anwendungssynthese.

Das Datenmodell und die identifizierten Operationen auf den Daten sind Ausgangspunkt der Synthese. Ein erster Schritt ist die Identifikation und Beschreibung von (System-)Anwendungsfällen. Diese Anwendungsfälle werden durch ein Ereignis z. B. Interaktion des Nutzers ausgelöst, beschreiben Dialoge und Funktionen zur Verarbeitung von Geschäftsinformationen sowie das Ergebnis. Das logische Datenmodell und die Anwendungsfälle stellen die erste konstruktive Aufbereitung der Nutzeranforderungen dar. Sie bilden die Grundlagen für die weitere Synthese der logischen Komponenten bzw. Dienste. Jede Entität im logischen Datenmodell sollte von einer Komponente bzw. einem Dienst »hoheitlich« verwaltet werden. Entsprechend der Abhängigkeiten im Datenmodell und in den Anwendungsfällen sind die Grenzen der Komponenten und Dienste zu ziehen.

Die zu verarbeitenden Daten werden im Weiteren hinsichtlich ihrer Schutzbedarfe und Sicherheitsanforderungen analysiert. Diese können z. B. aus gesetzlichen Regelungen resultieren. Entsprechende Nutzergruppen bzw. Rollen spiegeln in ihren Zugriffsrechten die durchzusetzenden Beschränkungen beim Zugriff auf die Daten wider. Diese Zugriffsbeschränkungen aber auch Anforderungen an das Antwortzeitverhalten und die Verfügbarkeit werden während des gesamten Vorgehens erhoben und jeweils den Konzepten wie Anwendungsfällen, Prozessen oder Komponenten zugeordnet. Die erhobenen Sicherheits- und Datenschutzanforderungen können zu einer Umstrukturierung der zuvor identifizierten Komponenten führen.

Die logische Grobarchitektur beschreibt die Komponenten und Dienste in ihrem Zusammenspiel. Klassisch kann man eine horizontale Unterteilung in die drei Schichten Präsentation, Logik und Persistenz bzw. Datenverwaltung vornehmen. Eine vertikale Aufteilung ist anhand der Komponenten möglich.

Die Einbettung der Anwendung in die existierende IT-Infrastruktur erfolgt als IT-System. Dessen Architektur beschreibt die Verteilung physischer Segmente z. B. in einer Client/Server-Architektur. Insbesondere bei Web-Anwendungen ist die zusätzliche Strukturierung in Tiers wie Client-, Server- oder Enterprise Information Systems (EIS)-Tier sinnvoll. Die Systemarchitektur wird durch ein

Datenschutz- und Sicherheitskonzept von Vorgaben des Bundesamts für Sicherheit in der Informationstechnik (BSI) [IT-Grundschutz] ergänzt, welches mit einer Definition der notwendigen Maßnahmen zur Zugangs- und Zugriffskontrolle, der Sicherheitsdienste und des Sicherheitsmanagements Vorgaben für die Umsetzung der nichtfunktionalen Anforderungen hinsichtlich Datensicherheit festlegt.

Neben der Definition der logischen Architektur der IT-Anwendung ist auch deren Zusammenspiel mit existierenden Anwendungen, Komponenten und Diensten von Bedeutung. Die genutzten oder angebotenen Schnittstellen, Protokolle und Formate der ausgetauschten Daten definieren dieses Zusammenspiel. Potentielle Änderungen an den Diensten, Betriebsprozessen und -mitteln sind für das Service Management [ITIL] der betrieblichen IT-Infrastruktur wichtige Informationen beim Betrieb der neuen Anwendung.

4.2 Sichtweise »Reverse-Engineering«

In der Sichtweise »Reverse-Engineering« ist der IT-Kontext Ausgangspunkt der Anwendungsanalyse. Die im Ergebnis identifizierten Teillösungen bestehender IT-Anwendungen können von anderen verwendet oder auch im Geschäftskontext zu einer neuen Lösung eines bestehenden organisatorischen Problems führen. Die Eröffnung neuer Geschäftsfelder und die Etablierung neuer Geschäftsmodelle durch die Nutzung bestehende Datenbestände sind ebenfalls möglich.

4.2.1 Anwendungsanalyse

Die Anwendungsanalyse verwendet die gleichen Konzepte und Modelle wie der Anwendungsentwurf bzw. die Anwendungssynthese. Hierbei ist eine bestehende IT-Anwendung und ihre Einbettung in die IT-Infrastruktur Gegenstand der Untersuchung. Zunächst ist die Identifikation der problemrelevanten IT-Anwendungen erforderlich. Ihre logische Architektur gibt Auskunft darüber, inwieweit das Herauslösen und die Wiederverwendung einzelner Komponenten möglich sind. Wichtig für die Wiederverwendung sind auch die implementierten Nutzer- und Berechtigungsmodelle. Ausgehend von den logischen Komponenten lassen sich die nutzbaren Funktionen und ihre Schnittstellen zunächst als Anwendungsfälle beschreiben. Ihre Verfeinerung führt zu den konkreten Schnittstellen, den auszutauschenden Daten, deren Formate und den Operationen. Erst auf dieser Granularitätsstufe ist eine Aussage über die Interoperabilität mit einer neuen IT-Anwendung bzw. über den notwendigen Anpassungsbedarf der bestehenden Anwendung möglich.

Neben der Analyse bestehender IT-Anwendungen ist auch die Einordnung der neu zu entwickelnden Anwendung in den IT-Kontext von Beginn an

notwendig. Leitplanken für ihre Konzeption und ihre Realisierung sind die IT-Ziele und die IT-Strategie. Dies kann sich z. B. auf Technologien, Programmiersprachen oder Standardsoftware beziehen. Eine kurze Beschreibung der Idee und der wichtigsten Merkmale der Anwendung gibt einen Rahmen bei ihrem Entwurf. Aus der geplanten Einbettung in die existierende IT-Infrastruktur resultieren ebenfalls Rahmenbedingungen.

4.2.2 Geschäftssynthese

Die Geschäftssynthese hat die Integration der verfügbaren IT-technisch realisierten Teillösungen zur Optimierung oder Erweiterung der organisatorischen Problemlösungen im Geschäftskontext zum Gegenstand. Dabei sind die gleichen Modelle und Konzepte verwendbar wie bei der Geschäftsanalyse. Die Teillösungen können Abläufe in den Geschäftsanwendungsfällen verändern und somit Auswirkungen auf die Gestaltung der Geschäftsprozesse haben. Der Online-Zugriff auf Geschäftsinformationen ermöglicht z. B. die Verlagerung ganzer Geschäftsanwendungsfälle oder einzelner Aktivitäten zum Kunden oder zu einem Subunternehmer. Beispiele dafür sind die Online-Bestellung, das Online-Banking, das E-Payment oder die Online-Sendungsverfolgung. Diese Verlagerung kann die Umstrukturierung der Geschäftsprozesse zur Folge haben oder neue ermöglichen. Dies wiederum wirkt sich auf Geschäftsakteure, die Organisationen und ihren Strukturen aus. Letztlich können die veränderten Geschäftsprozesse neue Geschäftsziele ermöglichen, wie z. B. durch Online-Werbung.

5 Funktionale, nicht-funktionale Anforderungen und Rahmenbedingungen

Die Anforderungen an eine IT-Anwendung lassen sich grundsätzlich in zwei Klassen unterscheiden. Die funktionalen Anforderungen beschreiben auf den verschiedenen Granularitätsstufen, was eine IT-Anwendung leisten soll. Die Grobarchitektur, ihre Komponenten, deren Anwendungsfälle sowie die verarbeiteten Daten stellen den funktionalen Zusammenhang dar. Dadurch ist jede funktionale Anforderung bzw. Operation in einen Kontext eingebettet.

Nicht-funktionale Anforderungen definieren konkrete Eigenschaften einer IT-Anwendung. Sie gliedern sich in

- quantifizierbare und
- qualifizierbare nicht-funktionale Anforderungen.

Zu den quantifizierbaren nicht-funktionalen Anforderungen gehören

- Leistung,
- Skalierbarkeit und
- Verfügbarkeit.

Qualifizierbare nicht-funktionale Anforderungen sind

- Sicherheit,
- Integrierbarkeit,
- Erweiterbarkeit und Zukunftssicherheit sowie
- Wiederverwendbarkeit.

Die nicht-funktionalen Anforderungen stehen nicht losgelöst im Raum, sondern beziehen sich immer auf ein Konzept der funktionalen Anforderungen. So ist es möglich, z. B. die konkrete Antwortzeit zu einem Anwendungsfall zu definieren.

Abbildung 5 Zusammenhang zwischen Anforderungen und Rahmenbedingungen

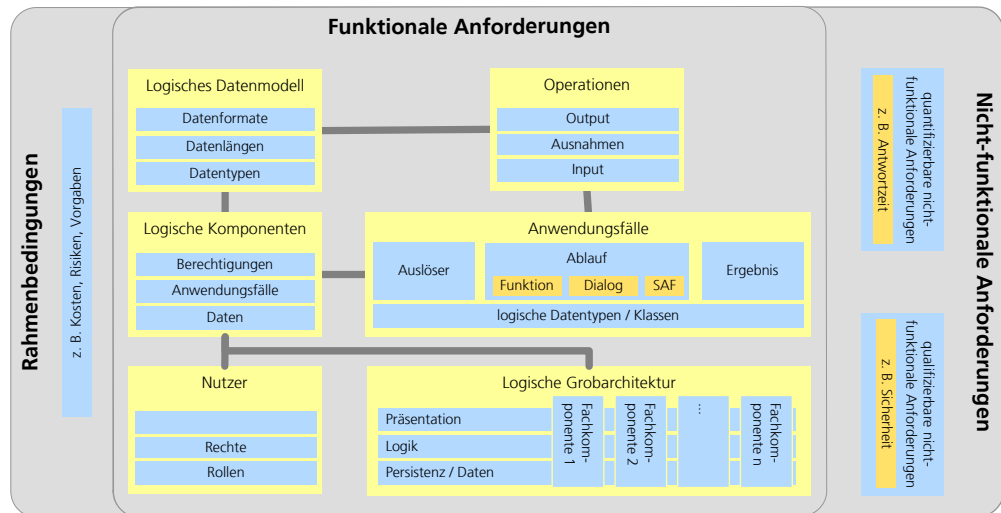


Abbildung 5 verdeutlicht den beschriebenen Zusammenhang. Nicht-funktionale Anforderungen sind eine orthogonale Dimension zu den funktionalen Anforderungen und stehen mit diesen in Beziehung.

Gleiches gilt für die Rahmenbedingungen. Diese adressieren Vorgaben hinsichtlich der einzuhaltenden Kosten oder Termine. Zusätzlich ist auch die Formulierung von technischen und organisatorischen Rahmenbedingungen möglich.

6 Aufbereitung der Ergebnisse als Lastenheft

Das Ergebnis der Analysen und Synthesen des Vorgehens sind verschiedene Modelle, die die untersuchten Konzepte und ihre Zusammenhänge beschreiben. Abhängig von den Rahmenbedingungen im Projekt gibt das Lastenheft diese Inhalte in der geforderten Aufbereitung und Struktur wieder. So gibt z. B. das V-Modell XT [V-Modell] für den öffentlichen Bereich die obersten Punkte der Struktur vor. Diese wurde im Folgenden um Punkte der »IEEE Recommended Practice for Software Requirements Specifications« [IEEE830-1998] ergänzt und auf der Basis der Erfahrungen aus verschiedenen Projekten verfeinert.

Abbildung 6

Mögliche Struktur eines Lastenhefts konform zum V-Modell XT und zu IEEE830-1998

1. Einleitung

- 1.1 Zweck des Dokuments
- 1.2 Verweise auf andere Ressourcen und Quellen
- 1.3 Begriffe und Abkürzungen
- 1.4 Aufbau des Dokuments
- 1.5 Historienführung

2. Ausgangssituation und Zielsetzung

- 2.1 Organisatorische Ausgangssituation
 - 2.1.1 Organisation und Geschäftsakteure
 - 2.1.2 Rechtliche Grundlage
 - 2.1.3 Geschäftsanwendungsfälle und -prozesse
- 2.2 IT-technische Ausgangssituation
 - 2.2.1 Nutzerarbeitsplatz
 - 2.2.2 Schwachstellen
 - 2.2.3 Übernahme von Altdaten
- 2.3 Zielsetzung
 - 2.3.1 Umfang der Anwendung
 - 2.3.2 Ausbaustufen
- 2.4 Organisatorische und Technische Rahmenbedingungen
 - 2.4.1 Einschränkungen
 - 2.4.2 Annahmen und Abhängigkeiten

3. Grobe Beschreibung der Anwendung

- 3.1 Wichtigste Funktionsmerkmale
-

-
- 3.2 Voraussetzungen
 - 3.3 Merkmale der Nutzer
 - 4. Funktionale Anforderungen**
 - 4.1 Grobe Beschreibung der Anwendung
 - 4.2 Logische Anwendungsarchitektur
 - 4.3 Komponenten und Anwendungsfälle
 - 4.4 Nutzer und ihre Rechte
 - 4.5 Datenmodell
 - 4.6 Mensch-Maschine-Schnittstelle
 - 5. Nicht-funktionale Anforderungen**
 - 5.1 Leistung
 - 5.2 Skalierbarkeit
 - 5.3 Verfügbarkeit
 - 5.4 Integrierbarkeit
 - 5.5 Erweiterbarkeit und Zukunftssicherheit
 - 5.6 Wiederverwendbarkeit
 - 6. Skizze des Lebenszyklus und der Systemarchitektur**
 - 6.1 Lebenszyklus
 - 6.2 Systemarchitektur
 - 6.3 Externe Schnittstellen
 - 6.3.1 Schnittstellen zu anderen Systemen und Anwendungen
 - 6.3.2 Import- / Export-Schnittstellen
 - 6.3.3 Datenbereitstellung
 - 6.3.4 Kommunikationsverbindungen
 - 6.4 Betrieb
 - 6.5 Nutzung
 - 6.5.1 Einsatzorte
 - 6.5.2 Einsatzzeiten
 - 7. Risikoakzeptanz**
 - 7.1 Risiken
 - 7.2 Schutzbedarf
 - 7.3 Schutzmaßnahmen
 - 7.3.1 Technische Schutzmaßnahmen
 - 7.3.2 Organisatorische Schutzmaßnahmen
 - 8. Lieferumfang**
 - 9. Abnahmekriterien**
-

Je nach Umfang der erarbeiteten Ergebnisse können detaillierte Informationen z. B. zu jeder Komponente der Anwendung oder dem Datenmodell in separate Anlagen ausgelagert werden.

Das V-Modell selbst gibt keine Vorgaben hinsichtlich der Methodik bei der Modellierung der relevanten Konzepte. Die Aufbereitung der Ergebnisse aus Analyse und Synthese im vorgestellten Vorgehen in verschiedenen UML-Modellen schafft neben den bereits ausgeführten Vorteilen wie der Prüf- und Transformierbarkeit die Voraussetzung für eine durchgängig modellbasierte Entwicklung der IT-Anwendung. Der nächste Abschnitt zeigt dazu eine Verbindung zwischen dem Ansatz des »Model Driven Engineering«, dem V-Modell und dem vorgestellten Vorgehen auf.

7 Model Driven Architecture und V-Modell XT

»Model Driven Architecture« bzw. »Model Driven Engineering« ist ein Ansatz der Object Management Group (OMG), der die durchgängige Verwendung von Modellen und Modellierungssprachen zur Verbesserung der Softwareentwicklung beschreibt [MDA03]. Dieser Ansatz unterscheidet drei Abstraktionsebenen für Modelle:

- Computation Independent Model (CIM),
- Platform Independent Model (PIM) und
- Platform Specific Model (PSM) sowie die Ebene
- Implementierung für den Programmcode.

Das »**Computation Independent Model**« beschreibt die Anforderungen an eine IT-Anwendung und die Situation, in der diese verwendet wird. Es abstrahiert von den Details der IT-technischen Unterstützung und wie diese realisiert ist. Es hilft die Erwartungen an die Funktionen der Anwendung aufzuzeigen.

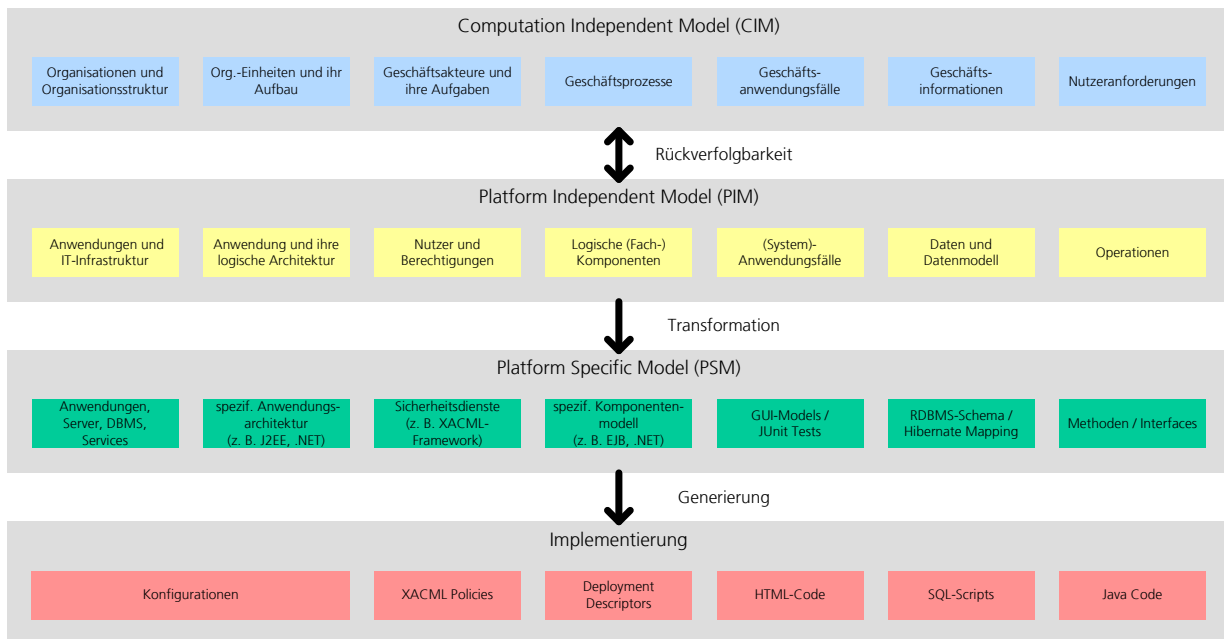
Das »**Platform Independent Model**« beschreibt die logische Architektur einer IT-Anwendung, ihre Komponenten und Operationen unabhängig von den Details ihrer Realisierung auf einer bestimmten Plattform. Es enthält den Teil der Spezifikation einer IT-Anwendung, der sich bei verschiedenen Plattformen nicht ändert.

Das »**Platform Specific Model**« kombiniert die Spezifikationen im PIM mit Details, die die Realisierung der IT-Anwendung auf einer bestimmten Plattform definieren. Auf Basis des PSM erfolgt die **Implementierung** der IT-Anwendung in Programmiersprachen der Plattform.

Ziel des MDA-Ansatzes ist eine (teil-)automatisierte Transformation der Modelle und letztendlich die Generierung des Codes für die Implementierung zu unterstützen. Kritikpunkte an der MDA sind, dass der initiale Aufwand zur Erstellung der Transformationen und Generatoren sehr hoch ist und das MDA eine sehr detaillierte Modellierung voraussetzt. Der dafür erforderliche Aufwand lohnt nicht in jedem Projekt. Unabhängig von der Frage, inwieweit es lohnenswert ist, in einem konkreten Projekt die Implementierung durch eine detaillierte Modellierung vorzubereiten, ist die prinzipielle Unterscheidung der verschiedenen Abstraktionsebenen sinnvoll. Der frühzeitige Verzicht auf natürlich sprachliche Formulierungen und der Einsatz von Modellierungssprachen helfen, die Anforderungen in der notwendigen Qualität zu definieren. Die Verwendung von UML-Modellen im »Computational Independent Model« ist in der MDA-Spezifikation eher vage beschrieben, so dass sich der vorgestellte Ansatz ergänzend einfügt. Abbildung 7 zeigt die

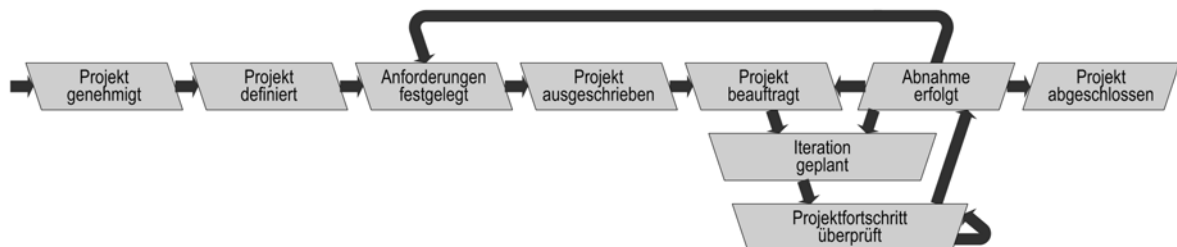
Einordnung der Modelle und Artefakte in die vier Abstraktionsebenen der MDA.

Abbildung 7 Abstraktionsebenen der MDA und Modelle des Vorgehens



Diese Einordnung ist konform zur Projektdurchführungsstrategie des V-Modells in Abbildung 8. Ergebnis der dortigen Aktivität »Anforderungen festlegen« ist das Lastenheft, welches die Anforderungen aus der Sicht des Auftraggebers festlegt.

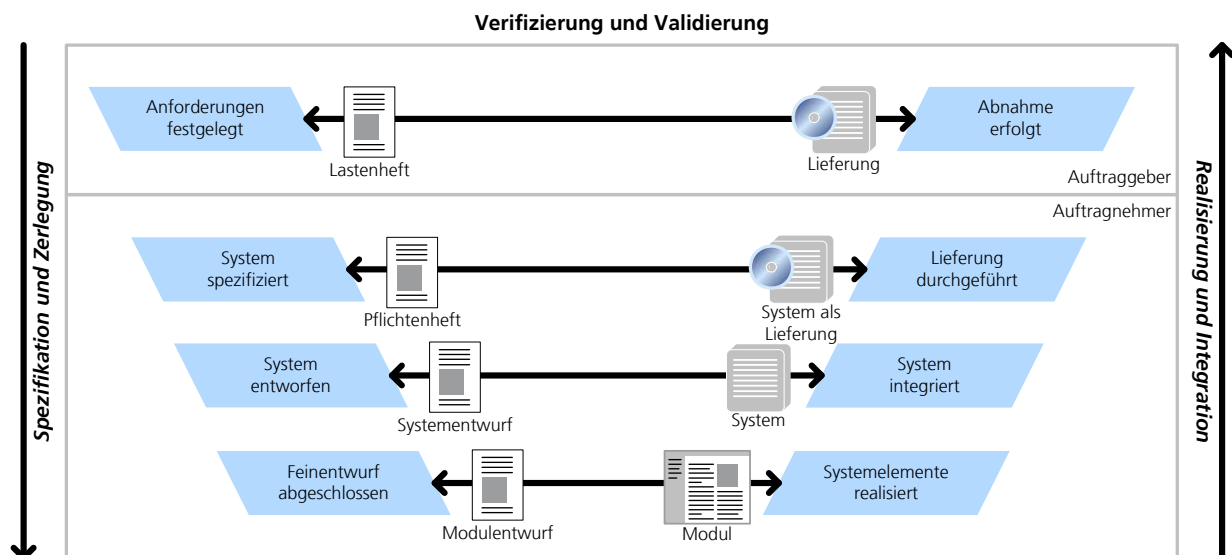
Abbildung 8 Projektdurchführungsstrategie »Vergabe und Durchführung eines Systementwicklungsprojektes« [V-Modell]



Alle Modelle aus dem beschriebenen Vorgehen die in das Lastenheft einfließen erstrecken sich auf die Ebenen »Computation Independent Model« und »Platform Independent Model«. Gleichzeitig enthält das Lastenheft alle

notwendigen Rahmenbedingungen zur Plattform, die sich z. B. aus der vorhandenen Infrastruktur ergeben. Dadurch haben die Anbieter bei der Ausschreibung des Projekts maximalen Spielraum, die plattformunabhängige, logische Anwendungsarchitektur aus dem Lastenheft in ihren Angeboten auf eine für sie passende Plattform abzubilden und das Ergebnis als Entwurfsvorschlag für eine plattformspezifische Realisierung darzustellen. Der Auftraggeber hat so die Möglichkeit, verschiedene Realisierungsoptionen zu vergleichen. Nach der Auftragserteilung erstellt der Auftragnehmer zum Meilenstein »System spezifiziert« (vgl. Abbildung 9) ein Pflichtenheft zu einer konkreten Plattform. Zusammen mit dem System- und Feinentwurf entspricht es dem »Platform Specific Model« des MDA-Ansatzes.

Abbildung 9 Struktur der Systemerstellung [V-Modell]



Nach der Implementierung und Integration erfolgt die Abnahme der IT-Anwendung gegen die plattformunabhängigen Anforderungen aus dem Lastenheft.

Das vorgestellte Vorgehen mit den verschiedenen Modellen aus den Analysen und Synthesen schafft bei einer entsprechenden Werkzeugauswahl die Voraussetzung für eine modellgetriebene Vorgehensweise bei der Realisierung einer IT-Anwendung. Es ist an verschiedenen Stellen skalierbar, so dass die Detaillierung der Modellierung und Spezifikation an die Bedingungen im konkreten Projekt anpassbar ist.

8 Literatur

- [Britt80] J. N. G. Brittan: Design for a Changing Environment, The Computer Journal, Vol. 31, No. 3, Feb. 1980
<http://comjnl.oxfordjournals.org/cgi/reprint/23/1/13.pdf>
- [Cock00] Alistair Cockburn. Writing Effective Use Cases. Addison-Wesley, 2000
- [IEEE830-1998] IEEE Computer Society: IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, 1998
- [IT-Grundschutz] <http://www.bsi.bund.de/gshb/>
- [ITIL] <http://www.itil.org/de/>
- [KOS68] E. Kosiol: Grundlagen der Organisationsforschung, 2. Auflage, Duncker & Humboldt, Berlin, 1968
- [MDA03] OMG: MDA Guide Version 1.0.1, 2003
<http://www.omg.org/cgi-bin/doc?omg/03-06-01>
- [Oest03] B. Oestereich, Ch. Weiss, C. Schröder, T. Weilkiens, A. Lenhard: Objektorientierte Geschäftsprozessmodellierung mit der UML, dpunkt.verlag, 2003
- [Oest04] B. Oestereich: Objektorientierte Softwareentwicklung – Analyse und Design mit der UML 2.0, 6. Auflage, Oldenbourg Verlag München / Wien, 2004
- [RUP02] Chris Rupp: Requirements-Engineering und -Management, 2. Auflage, Carl Hanser Verlag München / Wien, 2002
- [UML] <http://www.uml.org/>
- [V-Modell] <http://v-modell.iabg.de/>



