

# Requirements Engineering und IT Service Management – Ansatzpunkte einer integrierten Sichtweise

Michael Brenner, Markus Garschhammer (Munich Network Management Team)  
michael.brenner@mnm-team.org, markus.garschhammer@mnm-team.org

Friederike Nickl (Sepis GmbH – Ein Unternehmen der Swiss Life)  
friederike.nickl@sepis.de

**Abstract:** Dieser Beitrag untersucht die Zusammenhänge zwischen Requirements Engineering (RE) und IT Service Management (ITSM) und liefert Ansatzpunkte für eine beide Bereiche integrierende Sichtweise. Dazu werden RE-bezogene Grundkonzepte und Produkte aus Sicht der Entwicklung von Softwaresystemen, sowie die im Bereich des ITSM geprägten Begriffe zur Beschreibung des Betriebs eines Softwaresystems als Dienst, an Hand von UML-Modellen dargestellt. Die Zusammenhänge zwischen RE und ITSM werden auf Basis dieser Modelle in zwei grundlegenden Szenarien untersucht und zusammenfassend dargestellt.

## 1 Einführung und Motivation

In größeren Unternehmen werden IT-Systeme und Applikationen, die eine Anpassung an die spezifischen Geschäftsprozesse einer Fachabteilung erfordern, meist von einer spezialisierten IT-Entwicklungsabteilung angefertigt, dann aber von einer organisatorisch getrennten Abteilung<sup>1</sup>, häufig IT-Operations genannt, als Dienst (IT-Service) betrieben (vgl. Abbildung 1).

Die Idee des *Software Engineering* (SE), also des ingenieurmäßigen, methodischen Vorgehens bei der Softwareentwicklung, entstand bereits Ende der 60er Jahre. In der Domäne des IT-Betriebs bzw. des IT-Infrastrukturmanagements wird erst seit Ende der 80er Jahre, jenseits rein technischer Maßnahmen zur Effizienz- und Qualitätssteigerung, an organisatorischen Richtlinien und Rahmenmodellen geforscht, welche die Qualitätssicherung im produktiven Betrieb eines Softwaresystems methodisch stützen sollen. Analog zur Disziplin des *Software Engineering* ist damit im Bereich des IT-Betriebs eine Disziplin entstanden, die meist mit *IT Service Management* (ITSM) bezeichnet wird. Während Konzepte zur Entwicklung von Softwaresystemen, wie z.B. der *Unified Process* [JBR99] oder das *V-Modell* (aktuell zum *V-Modell XT* weiterentwickelt [Koo05]), in der Praxis bereits seit vielen Jahren Anwendung in Entwicklungsprojekten finden, stehen die meisten Unternehmen erst am Anfang der Umsetzung von Richtlinien zur Ablauforganisation des IT-Betriebs, wie beispielsweise der *IT Infrastructure Library* (ITIL) [Off05].

<sup>1</sup>Genauer betrachtet existieren oft mehrere Betriebs- oder auch Entwicklungsabteilungen, die jedoch meist wie dargestellt organisatorisch zusammengefasst werden.

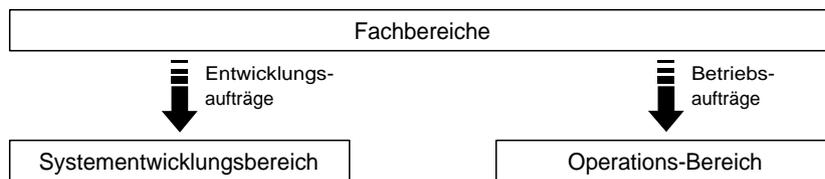


Abbildung 1: Typische organisatorische Strukturierung unternehmensinterner IT

Ein Unternehmen, das beispielsweise seine interne Entwicklung von IT-Systemen am V-Modell XT orientiert, zur gleichen Zeit aber den IT-Betrieb an ITIL ausrichten will, steht allerdings vor einem Problem: Den Rahmenwerken der SE- und ITSM-Domänen ist gemein, dass sie die außerhalb ihres Fokus liegenden Lebenszyklusphasen eines Systems zwar am Rande betrachten, aber kaum eine klare Abbildung ihrer Artefakte (z.B. Rollen, Aktivitäten, Dokumententypen) zu denen der Frameworks der jeweils anderen Domäne leisten (vgl. Abschnitt 4.1).

Eine integrierte Sichtweise ist aber besonders aus Sicht des *Requirements Engineering* (RE) notwendig. Fachliche Anforderungen<sup>2</sup> müssen nicht nur in frühen Phasen von Projekten zur Entwicklung von Softwaresystemen (im Folgenden kurz: Systementwicklung) erhoben und auf technische *Requirements* abgebildet werden – dies ist auch Hauptbestandteil des zentralen ITSM-Konzeptes *Service Level Management* (SLM). Insbesondere stehen die wichtigsten Kategorien der genannten nicht-funktionalen Anforderungen (vgl. Abschnitt 2.1.2) in einem engen Bezug zu den im SLM vereinbarten *Service Levels*.

Dieser Artikel will am Beispiel der repräsentativen Frameworks ITIL (für das IT Service Management) und V-Modell XT (für das Software Engineering) die bislang aus Sicht des RE selten betrachteten Zusammenhänge zwischen diesen beiden Informatik-Disziplinen aufzeigen und damit Ansatzpunkte für eine zukünftige, strukturiertere Betrachtung legen.

Im folgenden Abschnitt 2 werden zentrale RE-relevante Begriffe aus dem Bereich der Anwendungsentwicklung und dem Bereich des ITSM durch Modelle dargestellt und eingeordnet. Abschnitt 3 untersucht für zwei grundlegende Änderungsfälle die Beziehung und Interaktion zwischen den Bestandteilen dieser Modelle und zeigt damit die wechselseitigen Abhängigkeiten zwischen dem Requirements Engineering und dem IT Service Management. Zum Abschluss werden in Abschnitt 4 die Ergebnisse zusammengefasst und Ansatzpunkte für weitere Untersuchungen vorgestellt.

## 2 Systementwicklung und -betrieb

Als Basis für die in Abschnitt 3 folgende Diskussion der Beziehungen zwischen RE und ITSM werden zunächst in Abschnitt 2.1 die RE-bezogenen Grundkonzepte und Produkte aus Sicht der Entwicklung von Softwaresystemen dargestellt. Im anschließenden Abschnitt 2.2 werden die im Bereich des ITSM geprägten Begriffe zur Beschreibung des

<sup>2</sup>Anforderungen der Fachabteilungen

Betriebs eines Softwaresystems als Dienst beschrieben. In beiden Abschnitten erfolgt die, zum Teil bewusst vereinfachte, Darstellung jeweils anhand eines UML-Modells um Bezüge zwischen den Produkten des Entwicklungsprozesses und den Artefakten des ITSM herausarbeiten zu können.

## 2.1 Systementwicklung

Im Bereich der Systementwicklung haben sich sowohl Standards für die Eingliederung des Requirements Engineering in den Entwicklungsprozess etabliert, als auch Standard-Gliederungen für zu definierende Anforderungen. Im folgenden stellen wir zwei solche Modellbildungen in stark vereinfachter Form vor: In Abschnitt 2.1.1 wird ein Ausschnitt der Rollen und Produkte<sup>3</sup> der Systementwicklung in Anlehnung an das V-Modell XT modelliert. Daran anschließend wird in Abschnitt 2.1.2 eine Kategorisierung der Anforderungen im Systementwicklungsprozess nach dem *Volere Requirements Specification Template* [RR04] vorgenommen.

### 2.1.1 Anforderungen im Systementwicklungsprozess

Abbildung 2 stellt in Anlehnung an das V-Modell XT [Koo05] in vereinfachter Form Rollen und Produkte aus dem Entwicklungsprozess eines Software-Systems dar: Vom *Anforderungsanalytiker des Auftraggebers* (Anforderungsanalytiker AG) wird ein *Lastenheft* erstellt. Es enthält die Anforderungen an das zu erstellende Softwaresystem, die Basis für die Abnahme der im Vertrag festgeschriebenen Lieferung sind. Diese Anforderungen werden durch Abnahmekriterien überprüfbar gemacht. Das *Pflichtenheft* (auch Gesamtsystemspezifikation genannt) ist die Fortschreibung des Lastenhefts durch den *Anforderungsanalytiker des Auftragnehmers* (Anforderungsanalytiker AN). Die Anforderungen und Abnahmekriterien aus dem Lastenheft werden übernommen, geeignet aufbereitet und konkretisiert. Zusätzlich wird eine erste Grobarchitektur und Schnittstellenübersicht erstellt.

Aufbauend auf dem Pflichtenheft wird durch den Systemarchitekten die Systemarchitektur entworfen: es wird beschrieben, wie sich das System aus einzelnen Systemelementen zusammensetzen soll. Jedes Systemelement wird durch eine eigene Spezifikation beschrieben<sup>4</sup>.

Die Erstellung der Spezifikationen erfolgt parallel zum Architekturentwurf in der Verantwortung des Systemarchitekten<sup>5</sup>. Durch Anforderungsverfolgung soll sichergestellt werden, dass tatsächlich alle Anforderungen aus der Gesamtsystemspezifikation bei der Verfeinerung berücksichtigt werden. Die Verfeinerung kann sich über mehrere Hierarchie-

---

<sup>3</sup>Als Produkte werden dabei nach dem V-Modell alle im Verlauf eines Entwicklungsprozesses zu erarbeitenden Ergebnisse und Zwischenergebnisse bezeichnet. Dabei kann es sich sowohl um Dokumente als auch um Systemelemente handeln [Koo05]

<sup>4</sup>der Einfachheit halber verwenden wir im Modell den Begriff „Spezifikation Systemelement“ und unterscheiden nicht – wie im V-Modell – zwischen „Systemspezifikation“ und „SW-Spezifikation“

<sup>5</sup>der Einfachheit halber unterscheiden wir hier nicht zwischen „Architektur“ und „SW-Architektur“ und den Rollen „Systemarchitekt“ und „SW-Architekt“

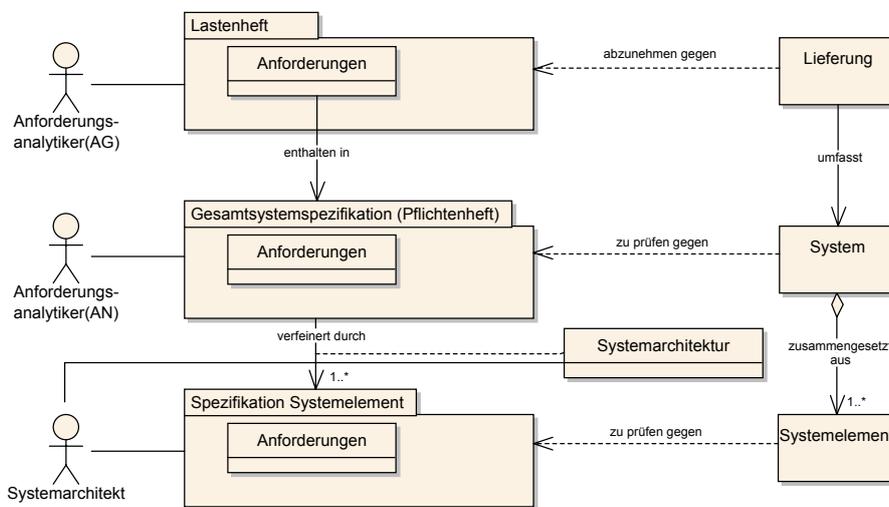


Abbildung 2: Systementwicklung in Anlehnung an das V-Modell XT

ebenen erstrecken (im Diagramm ist nur eine Stufe zu sehen).

Die nicht weiter verfeinerten Systemelemente werden zunächst im Komponententest gegen die Anforderungen in ihrer Spezifikation geprüft (getestet) und dann sukzessive zu komplexeren Systemelementen integriert und wiederum gegen deren Spezifikation getestet. Der abschließende Test beim Auftragnehmer ist der Test des komplett integrierten Systems gegen die Gesamtsystemspezifikation. In diesem Test wird insbesondere gegen die Abnahmekriterien aus dem Lastenheft getestet.

Nach erfolgreichem Test erfolgt die Lieferung des Systems einschließlich aller sonst geforderten Lieferbestandteile (wie beispielsweise Dokumentation und Benutzerhandbuch). Der Abnahmetest des Auftraggebers erfolgt auf Basis der im Lastenheft formulierten Abnahmekriterien.

In Abbildung 2 werden die am Entwicklungsprozess beteiligten Rollen links im Modell mit einer Verbindung zu den jeweils von ihnen verantworteten Produkten des Entwicklungsprozesses dargestellt. Die Beziehungen zwischen den Produkten sollen Produktabhängigkeiten in Anlehnung an das V-Modell XT veranschaulichen.

### 2.1.2 Kategorien von Anforderungen bei der Systementwicklung

Ausgangspunkt für eine Systementwicklung, wie im vorherigen Abschnitt beschrieben, sind die Anforderungen an das System, wie sie von Auftraggeberseite gestellt werden. Im Lauf des Entwicklungsprozesses werden diese Anforderungen immer wieder verfeinert und beeinflussen damit den gesamten Entwicklungsprozess.

Die Anforderungen umfassen auf allen Ebenen der Spezifikation sowohl funktionale als auch nicht-funktionale Anforderungen. Die funktionalen Anforderungen beschreiben die

Fähigkeiten eines Systems, die ein Anwender erwartet, um mit Hilfe des Systems ein fachliches Problem zu lösen [Koo05]. Diese Anforderungen können in Form von Anwendungsfällen (*Use Cases*) beschrieben werden. Die nicht-funktionalen Anforderungen beschreiben Eigenschaften, welche das System haben muss [RR99] – es handelt sich dabei um Charakteristika und Qualitäten, die entscheidend zur Anwendbarkeit und Brauchbarkeit des Systems im produktiven Betrieb beitragen [Koo05]. Da diese Charakteristika und Qualitäten über ein einzelnes Projekt hinaus für Software-Systeme im allgemeinen relevant sind, ist es sinnvoll bei der Beschreibung dieser Anforderungen auf eine Standard-Gliederung zurückzugreifen [Rup04]. Eine Gliederung für Anforderungen an die Dienstqualität liefert z.B. die DIN 66272 [DIN94].

Wir beziehen uns im folgenden auf die Gliederung nicht-funktionaler Anforderungen aus dem Volere Requirements Specification Template [RR04]. In dieser Gliederung werden die folgenden Kategorien nicht-funktionaler Anforderungen unterschieden: Anforderungen an die Oberflächengestaltung (*Look and Feel Requirements*), Benutzbarkeit (*Usability and Humanity Requirements*), Leistungsanforderungen (*Performance Requirements*), betriebliche Anforderungen (*Operational Requirements*), Anforderungen an Wartbarkeit und Anpassbarkeit (*Maintainability and Support Requirements*), Sicherheitsanforderungen (*Security Requirements*), kulturelle und politische Anforderungen (*Cultural and Political Requirements*), juristische Anforderungen (*Legal Requirements*).

Die einzelnen Kategorien werden in dem Template wiederum genauer in Klassen von Anforderungen untergliedert und es werden Beispiele von Testkriterien (*Fit Criteria*) angegeben, mit denen die Erfüllung der Anforderungen überprüft werden kann. Diese Testkriterien beziehen sich auf konkrete Qualitätsmaße, mit denen die Anforderungen messbar gemacht werden können. In der Sprache des IT Service Management (siehe Abschnitt 2.2) werden diese Qualitätsmaße als Dienstgüteparameter bezeichnet.

So untergliedert sich die Klasse *Performance* wiederum in Anforderungen zu folgenden Aspekten: Zeitverhalten (*Speed and Latency Requirements*), Schutz vor möglichem Schaden (*Safety Critical Requirements*), Genauigkeit (*Precision or Accuracy Requirements*), Verfügbarkeit und Zuverlässigkeit (*Reliability and Availability Requirements*), Belastbarkeit und Skalierbarkeit (*Capacity Requirements*), Langlebigkeit (*Longevity Requirements*). Ein Maß zur Überprüfung von Anforderungen zum Zeitverhalten ist beispielsweise die Antwortzeit.

Anforderungen an sonstige Lieferbestandteile wie Dokumentation und Handbücher (siehe [Rup04]) werden im Volere-Template nicht gesondert aufgeführt, sondern den Gliederungspunkten Wartbarkeit und Projektanforderungen (*Project Issues*) zugeordnet. Wir fassen diesen Aspekt im folgenden auch unter die nicht-funktionalen Anforderungen.

## 2.2 Systembetrieb - IT Service Management

Formale Beschreibungen der Vorgehensweisen im ITSM sind erst im Entstehen und keineswegs weitläufig anerkannt. Mit der ITIL [Off05], die *Best Practices* im ITSM beschreibt, haben sich zwar in den letzten Jahren de-facto Standardvorgehen im Bereich



bringung. In einem SLA wird der zur Verfügung zu stellende IT-Dienst<sup>6</sup> beschrieben, d.h. die funktionalen Aspekte der Dienstleistung werden festgelegt. Zusätzlich kann auch der Rahmen für unterstützende Dienstleistungen, wie z.B. telefonischer Support, definiert werden. Schwieriger ist allerdings in der Praxis meist die Festlegung von Service Levels, d.h. einer zu garantierenden Dienstqualität (*Quality of Service*, QoS).

Hierfür müssen Kunde und Provider (vgl. Abbildung 3) zunächst übereinkommen, welche Dienstgüteparameter (*Service Parameter*, *QoS-Parameter*) zur Bestimmung der Dienstqualität herangezogen werden sollen. Die Dienstqualität bestimmt sich grundsätzlich aus nicht-funktionalen Aspekten des Dienstes, wie z.B. der Antwortzeit. Im Idealfall bilden die Dienstgüteparameter ein Kennzahlensystem, das die Qualitätswahrnehmung des Kunden exakt widerspiegelt. Da selbst für die gängigen Parameter, wie z.B. „Verfügbarkeit“, keine standardisierten Mess- und Berechnungsmethoden existieren, müssen auch diese in einem SLA festgelegt werden.

Auf dieser Basis erfolgt zwischen Kunde und Provider die Verhandlung des Dienstgüteniveaus (*Service Level*). Ein Dienstgüteniveau, in von IT-Dienstleistern vorbereiteten Verträgen oft mit Bezeichnungen wie „basic“, „silver“ oder „gold“ versehen, ist durch einzuhaltende Wertebereiche in den Dienstgüteparametern bestimmt (z.B. könnte „gold“ definiert werden mit: Antwortzeit immer kleiner 1 Sekunde und Verfügbarkeit größer 99%). *Service Level* entsprechen also in etwa *Fit Criteria* für nicht-funktionale Requirements. Da das Messen der vereinbarten Dienstqualität an den Service Levels allerdings nicht nur einmalig, sondern kontinuierlich über die Laufzeit des SLA erfolgt, ist hier das rechtzeitige Abwägen zwischen Testaufwand und -nutzen noch wichtiger als bei den *Fit Criteria*.

Welche technischen Maßnahmen auf Providerseite erforderlich sind, um diese Service Levels einzuhalten wird im *Service Specsheat* (Specification Sheet) [Off05] beschrieben<sup>7</sup>. Das Specsheat ist ein Provider-internes, d.h. nicht durch den Kunden einsehbares Dokument, so dass dem Dienstanbieter in der technischen Umsetzung des Dienstes Freiheiten bleiben – solange er sich eben an das SLA hält. Idealerweise würde ein solches Service Specsheat geeignete Berechnungsmethoden enthalten um zwischen Komponenten- und Dienstparametern abbilden zu können, dies ist in der Praxis aufgrund der Komplexität der Abhängigkeiten aber nur selten möglich. Für die Erstellung des Specsheets und der internen Verantwortung für die Bereitstellung des Dienstes ist in der ITIL keine Rolle definiert, in der Praxis ist hierfür häufig ein so genannter *Service Manager* (im Sinne eines technischen Produktmanagers) zuständig. Das Management der Komponenten wird, wie in Abbildung 3 gezeigt, durch den *System Manager* (ebenfalls eine oft in der Praxis angebotene, aber von der ITIL selbst nicht definierte Rolle) kontrolliert.

Analog zur Modellierung in Abschnitt 2.1 werden auch in Abbildung 3 die am ITSM beteiligten Rollen im Bezug zu den von ihnen betreuten Artefakten dargestellt. Die Darstellung richtet diese dabei so aus, dass kundenrelevante Aspekte in der Abbildung oben dargestellt werden und die (für den Kunden nicht sichtbaren) Providerspezifika nach unten hin bis zur Ebene der Komponenten zunehmen.

<sup>6</sup>Wir verstehen Dienst im Kontext des Service Level Managements kundenorientiert, als Enterprise IT Service – abweichend von dem technischen Service-Begriff, wie er z.B. im Kontext von Kommunikationssystemen oder Service Oriented Architectures verwendet wird

<sup>7</sup>Anforderungen an organisatorische Maßnahmen werden in einem *Service Quality Plan* festgehalten

### 2.2.2 Kategorien von QoS-Parametern

Im Bereich des ITSM ist die Etablierung von (Quasi-)Standards zur Kategorisierung von QoS-Parametern weniger fortgeschritten als die Kategorisierung von Software Requirements. Für reine Netzdienste gibt es eine Reihe allgemein anerkannter QoS-Parameter wie *Transfer Delay*, *Jitter* usw. - es gibt aber erst in jüngerer Zeit Bemühungen, Definitionen allgemeinerer QoS-Parameter, wie sie für komplexere Dienste notwendig sind, formal zu spezifizieren (siehe z.B. [OMG04]).

Diese allgemeineren QoS-Parameter lassen sich grob in Kategorien wie *Performance*, *Dependability*, *Security* und *Integrity* (z.B. Datenqualität oder Berechnungsgenauigkeit) einteilen (vgl. [OMG04]). Die damit beschriebenen Qualitätskriterien stimmen weitgehend mit denen überein, die im Volere-Template durch nicht-funktionale Anforderungen (speziell denen aus dem Bereich *Performance*) erfasst werden (vgl. Abschnitt 2.1.2).

## 3 Analyse der Modellbeziehungen anhand von Änderungsfällen

Auf Basis der im vorhergehenden Abschnitt dargestellten Modelle werden in diesem Abschnitt Zusammenhänge zwischen Systementwicklung und ITSM anhand zweier grundlegender Änderungsfälle, Neuentwicklung eines Systems (Abschnitt 3.1) und Anpassung eines bestehenden Systems (Abschnitt 3.2), dargestellt. Die dabei herausgearbeiteten Ansatzpunkte für eine integrierte Sichtweise illustrieren wie das Wissen um die RE-relevanten Zusammenhänge zwischen Systementwicklung und SLM genutzt werden kann, um durch Wiederverwendung von Artefakten und klare Rollenabgrenzung die Gesamtabläufe effizienter zu gestalten.

### 3.1 Änderungsfall „Neuer Dienst“

Um Geschäftsprozesse mittels grundlegend neuer Funktionalitäten zu unterstützen, erteilt die Fachabteilung einen Entwicklungsauftrag für ein neues System. Die Anforderungsanalyse zur Erstellung des Lasten- und Pflichtenheftes geschieht hier in der Regel vor, bestenfalls parallel zur Vereinbarung eines SLA zum Betrieb dieses Systems als Dienst. Die sich in diesem Änderungsfall ergebenden Wechselwirkungen und Zusammenhänge zwischen Systementwicklung und SLM werden im Folgenden diskutiert. Abbildung 4 illustriert die dabei identifizierten Beziehungen zwischen den Produkten und Artefakten aus den Modellen in Abbildungen 2 und 3.

#### 3.1.1 Beziehungen zwischen den Rollen der Systementwicklung und des ITSM

Idealerweise nähme der SL-Manager die Rolle des Anforderungsanalytikers des Auftraggebers wahr und würde die fachlichen Vorgaben des Kunden entgegennehmen. Ist dies, wie oft im Kontext des hier betrachteten Umfeldes, nicht der Fall, so sollte doch eine

enge Zusammenarbeit zwischen dem Anforderungsanalytiker AG und dem SL-Manager, sowie zwischen dem Anforderungsanalytiker AN und dem Service Manager stattfinden. Die Erfüllung vieler nicht-funktionaler Anforderungen kann sowohl durch Maßnahmen in der Entwicklung als auch durch Maßnahmen im Betrieb erreicht werden. So kann Verfügbarkeit durch den Einsatz hochverfügbarer Hardware im Betrieb, oder auch durch einen besonders effektiven technischen Support erhöht werden – oder aber durch ein Software-Design das mittels intelligenter Failover- und Recovery-Mechanismen hohe Verfügbarkeit auch auf Clustern unzuverlässiger Hardware erlaubt. Welche Kombination von Maßnahmen die kosteneffektivste ist, lässt sich nur in einem Dialog zwischen Entwicklung und Betrieb klären.

### **3.1.2 Beziehung zwischen Lastenheft und SLA**

Die Dienstbeschreibung im SLA und die funktionalen Anforderungen im Lastenheft können voneinander abgeleitet werden (vgl. Abbildung 4), indem z.B. entsprechende Anwendungsfälle ins SLA übertragen werden (vgl. Diskussion von ITIL Application Management [Off02] in Abschnitt 4.1). Ein SLA umfasst im allgemeinen darüber hinaus allerdings noch organisatorische Aspekte, wie z.B. die Beschreibung von Support-Diensten im Betrieb (z.B. Kontaktmöglichkeiten und Erreichbarkeitszeiten des Service Desk), die unabhängig von den Systemeigenschaften sind und daher nicht vor Systemfertigstellung festgelegt und in den Systemanforderungen erfasst werden müssen.

Es können sich aber auch aus der Notwendigkeit, Dienstgüteparameter im laufenden Betrieb messen können zu müssen, zusätzliche funktionale Anforderungen an das System bezüglich einer Instrumentierung für Management-Tools ergeben.

Die nicht-funktionalen Qualitätsanforderungen an das neue System entsprechen weitgehend dem im SLA vereinbarten Dienstgüteniveau<sup>8</sup>. Dabei handelt es sich schwerpunktmäßig um Anforderungen aus den Kategorien Leistung, betriebliche Anforderungen, Wartbarkeit und Sicherheit des Volere-Templates (siehe 2.1.2). Wichtige Dienstgüteparameter in der Leistung-Kategorie sind beispielsweise die Verfügbarkeit und die Antwortzeit.

### **3.1.3 Beziehung zwischen Systemelementen aus der Systementwicklung und Komponenten aus dem ITSM**

Im Gesamtarchitekturentwurf im Pflichtenheft und durch die anschließende Festlegung der System- und Software-Architektur erfolgt die sukzessive Dekomposition des Anwendungssystems in Systemelemente. Dabei wird sowohl die fachliche, als auch die technische Architektur festgelegt (siehe dazu beispielsweise [Ten05]). Die technische Struktur beschreibt, aus welchen Schichten und technischen SW-Komponenten das System besteht. Diese technische Architektur muss an die IT-Infrastruktur des Providers angepasst werden, d.h. die in der Systemarchitektur spezifizierten Systemelemente müssen abbildbar sein auf die IT-Komponenten (Hardware, Netzkomponenten, Middleware) des Providers. Dieser Zusammenhang wird in Abbildung 4 durch die Beziehung „realisiert auf“ zwischen

---

<sup>8</sup>eine Ausnahme sind z.B. die Look and Feel Requirements

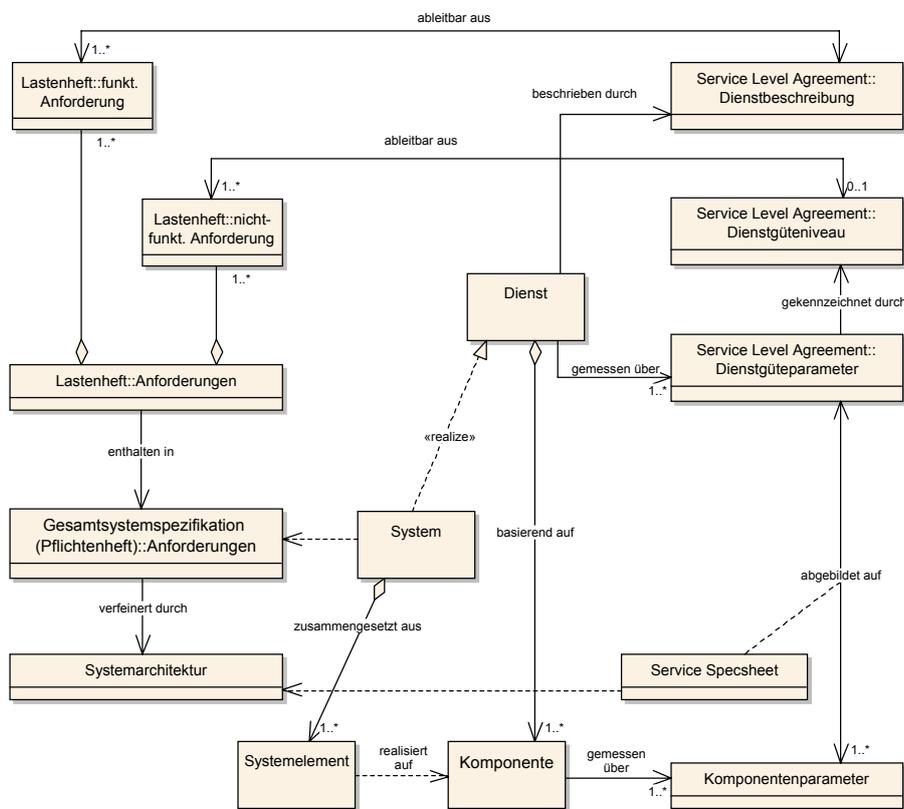


Abbildung 4: Modellbeziehungen aus dem Änderungsfall „neuer Dienst“

den Systemelementen aus der Systementwicklung und den Komponenten des ITSM dargestellt. Die Abbildung zwischen Dienstgüteparameter und Komponentenparameter des Dienstes wiederum basiert wesentlich auf der technischen Architektur des Anwendungssystems. Dies ist dargestellt durch die Abhängigkeitsbeziehung zwischen dem Service Specsheet und der Systemarchitektur.

### 3.2 Änderungsfall „Anpassung eines vorhandenen Dienstes“

Änderungen an einem Dienst im produktiven Betrieb werden im allgemeinen vom Service Level Manager angestoßen. Dieser reagiert entweder auf Trendanalysen, wie z.B. zunehmende Nutzung bislang selten nachgefragter und daher wenig optimierter Funktionen und damit sich verschlechternde QoS-Werte, oder es entwickeln sich im Dialog mit Anwendern oder dem Kunden zusätzliche Anforderungen an den Dienst. Die folgende Betrachtung fokussiert auf den ersten Fall.

Beispielsweise könnte sich bei einem Buchungssystem herausstellen, dass die zur Batchverarbeitung anstehenden Aufgaben stetig wachsen und somit nicht mehr zuverlässig im nächtlichen Batchintervall abgeschlossen werden können – was die Verfügbarkeit des Systems für das Personal der Fachabteilung während der regulären Arbeitszeiten zu beeinträchtigen droht. Eine derartige Gefährdung des vereinbarten Qualitätsniveaus ist oft durch ein erhöhtes, in der Anforderungsanalyse nicht berücksichtigtes Datenvolumen oder verändertes Anwenderverhalten hervorgerufen.

Wie bereits im vorherigen Abschnitt werden im Folgenden die Bezüge zwischen den am Entwicklungsprozess und am Dienstbetrieb beteiligten Rollen, so wie zwischen den entsprechenden Artefakten des jeweiligen Modells dargestellt.

### **3.2.1 Beziehungen zwischen den Rollen des ITSM und der Systementwicklung**

Die Einhaltung eines SLA unter Umständen wie sie im obigen Beispiel beschrieben sind, kann auf effiziente Weise häufig nicht mit Mitteln des Betriebs, wie einem Hardware-Upgrade, sichergestellt werden. Die kosteneffektive Sicherstellung der Qualität des Dienstes, kann in einem solchen Fall Änderungen am ihm zugrunde liegenden System erfordern.

Nach einer nochmaligen Anforderungsanalyse müssen die geänderten Anforderungen an die Entwicklungsabteilung (z.B. an den Anforderungsanalytiker AN) weitergegeben werden. Idealerweise übernehme der Service Level Manager in diesem Fall die Rolle des Anforderungsanalytikers AG gegenüber der Entwicklungsabteilung.

Für die Ermittlung dieser zusätzlichen oder genauer spezifizierten Anforderungen an das System kann der SL-Manager auf im Dienstbetrieb gesammelte Performance-Daten zurückgreifen. Zu diesem Zweck kann der SL-Manager innerhalb des Service Providers auf den Service Manager und den System Manager zurückgreifen und auf diese Weise eventuell auch zusätzliche Requirements, wie z.B. betreffend die Management-Instrumentierung des Systems, identifizieren.

Bei der Umsetzung der neu entstandenen Anforderungen sollten die auf den jeweiligen Detaillierungsebenen beteiligten Rollen in Entwicklung und Betrieb im engen Austausch zusammen arbeiten, um eine korrekte Umsetzung der Anforderungen zu gewährleisten. Zudem haben Service Manager und System Manager detaillierte Erfahrungen mit dem Laufzeitverhalten der Anwendung aus dem Betrieb, die sie an den Anforderungsanalytiker AN bzw. an den Systemarchitekten weitergeben können.

Ergebnis dieser Abstimmung ist auch eine Strategie, in welcher Weise die geänderten Anforderungen effizient umgesetzt werden können. Auf Seiten der Entwicklung kann im Extremfall ein komplett neues System entworfen werden, das auf der vorhandenen Infrastruktur des Providers implementiert wird, so dass auf dessen Seiten keinerlei Änderungen notwendig sind. Im anderen Extrem können alle Anforderungen, besonders wenn sie die Performanz eines Systems betreffen, durch Änderungen an der Infrastruktur des Providers (z.B. Verwendung leistungsfähigerer Server) umgesetzt werden. Natürlich kann auch eine Kombination von Maßnahmen beider Bereiche die beste Lösung darstellen.

### **3.2.2 Beziehung zwischen SLA und Lastenheft**

Aus einem im Detail festgelegten SLA und einer veränderten Betriebssituation (z.B. geänderte Anfrage- oder Datenlast) können so vielfältige Requirements entstehen, die zusätzlich in das Lastenheft aufgenommen werden müssen. Dies kann im Extremfall die Systemspezifikation massiv beeinflussen, wenn beispielsweise auf Grund von Performanzproblemen die Systemarchitektur geändert werden muss.

### **3.2.3 Beziehung zwischen Systemelementen aus der Systementwicklung und Komponenten aus dem ITSM**

Unabhängig von der gewählten Lösung werden aus Sicht der Modellierung Komponenten und Systemelemente und die sie verbindende Assoziation, die das Deployment eines Systems auf eine Infrastruktur modelliert, verändert. Aus Sicht von Abbildung 4 zeigt sich, dass kleine Änderungen in den Eingangsanforderungen in der Implementierung hohen Änderungsaufwand erzeugen können. Eine konsequente Umsetzung von nicht-funktionalen Anforderungen wie Wartbarkeit, Modularität oder Erweiterbarkeit bereits bei der Neuentwicklung einer Anwendung kann hier deutlich zur Aufwandsreduzierung beitragen.

### **3.3 Referenzpunkte in Entwicklung und Betrieb: Lastenheft und SLA**

Die durchgeführte Betrachtung zeigt, dass das über einen Dienst ausgehandelte SLA, das faktisch erst in der Betriebsphase des Dienstes respektive des Systems zum Tragen kommt, massiven Einfluss auf alle Detailstufen der Systementwicklung hat. Besonders deutlich wird dies, wie beschrieben, bei der Durchführung einer Änderung auf Grund gefährdeter Qualitätseigenschaften. Grundsätzlich müssen die möglichen Auswirkungen allerdings auch bei der Neuentwicklung eines Dienstes beachtet werden, um die Basis für eine effiziente Weiterentwicklung zu legen.

In beiden dargestellten Fällen wäre eine zumindest methodisch definierte, wenn nicht vollständig automatisierte Anbindung der beiden Zentraldokumente Lastenheft und SLA aneinander wünschenswert, um die Systementwicklung bzw. das IT Service Management effizienter gestalten zu können und die Kommunikation zwischen den beteiligten Rollen zu erleichtern. Eine derartige Vernetzung der beiden für die jeweiligen Phasen handlungsbestimmenden Dokumente setzt allerdings weitere detaillierte Untersuchungen ihre Abhängigkeiten voraus.

## **4 Zusammenfassung und Ausblick**

Im vorliegenden Beitrag wurde ein Ansatz vorgeschlagen, der eine integrierte Sicht über Entwicklung und Betrieb eines Systems bzw. Dienstes erlaubt und die für das Requirements Engineering relevanten Artefakte des Software Engineering und des IT Service

Management einordnet. Zum Abschluss dieses Beitrags stellt dieser Abschnitt zunächst verwandte Arbeiten sowohl aus dem Bereich des IT Service Managements als auch aus dem Bereich der Systementwicklung vor und gibt schließlich einen Ausblick auf weitere forschungsrelevante Fragestellungen.

#### 4.1 Related Work

Die Modellierung von IT-Diensten wurde neben [Lew99] auch in [GHK<sup>+</sup>01] untersucht. Das dort vorgestellte Dienstmodell deckt sich in den grundsätzlichen Aussagen mit den in Abschnitt 2.2 eingeführten Begrifflichkeiten ist aber besonders in den providerspezifischen Teilen detaillierter, verzichtet aber auf eine umfassende Betrachtung funktionaler Requirements.

Zu der Integration von organisatorischem IT Service Management und Software Engineering existieren aus dem ITSM-Umfeld zahlreiche weitere Vorschläge, wovon allerdings der überwiegende Anteil das Requirements-Engineering nicht näher berücksichtigt.

Die *Application Services Library* der *ASL Foundation* [vdP04] bemüht sich ebenfalls um eine Integration des IT Service Managements mit dem Application Management. In ihrer Definition von Application Management legt sie aber den Schwerpunkt auf die Wartung, Verbesserung und Renovierung bestehender Applikation, d.h. adressiert nicht den Fall einer Dienstneueinführung.

Microsoft strebt eine engere Verzahnung seines ITIL-basierten *Microsoft Operations Framework* (MOF) [Mic04] mit dem *Microsoft Solutions Framework* (MSF). MSF beinhaltet Methoden und Prozesse für die Systementwicklung, ist in diesem Sinne also mit V-Modell oder Unified Process vergleichbar. Zum einen werden allgemeine Aussagen zur Kommunikation zwischen den so genannten *Role Clusters* im Rahmen der *Team Models* von MOF und MSF gemacht – zum anderen werden zwei *Operations Management Reviews* des MOF zu Entscheidungspunkten des MSF abgebildet.

Im Umfeld der Hersteller von IT-Management-Tools befasst man sich ebenfalls zunehmend mit den Konsequenzen, die das ITSM auf die Softwareentwicklung hat [Ent05], allerdings liegt hier der Fokus vorwiegend auf der notwendigen Management-Instrumentierung<sup>9</sup> von zu entwickelnden Applikationen.

Als einziger der in diesem Abschnitt diskutierten Ansätze definiert das ITIL *Application Management*<sup>10</sup> [Off02] ebenfalls grundlegende Requirements-Kategorien (*functional*, *non-functional* und *usability*) und gibt Richtlinien mit welchen Betriebs- und ITSM-Prozessbereichen Requirements allgemein abzustimmen sind. Die Darstellung ist allerdings wenig formal und bleibt damit stellenweise vage. Eine direkte Verbindung zum Service Level Management wird nur in Bezug auf die Verwendbarkeit von Use Cases für sowohl funktionale Requirements, wie auch funktionale Dienstbeschreibungen hergestellt.

Während sich die Software-Engineering Literatur lange vor allem auf die Implementierung

---

<sup>9</sup>Einbettung von Mechanismen und Schnittstellen zur Steuerbarkeit durch IT-Management-Tools

<sup>10</sup>Dieser Band der ITIL wird i.d.R. nicht zum ITSM im engeren Sinne gezählt

funktionaler Aspekte konzentrierte, finden sich in jüngerer Zeit vermehrt Ansätze, die sich mit nicht-funktionalen Qualitätsaspekten befassen: Das *NFR Framework* [CNEM99] liefert einen Rahmen für die systematische Darstellung der logischen Abhängigkeiten und der Nachverfolgung von nicht-funktionalen Anforderungen. In iterativen, architekturzentrierten Entwicklungsprozessen wie dem *Unified Process* [JBR99] wird der Test nicht-funktionaler Anforderungen schon frühzeitig in den Prozess einbezogen, da der in der Elaborationsphase zu entwickelnde Architekturprototyp an Hand der Kernanforderungen überprüft werden muss.

Mit der systematischen Umsetzung nicht-funktionaler Anforderungen im Softwareentwicklungsprozess befasst sich aktuell die *Early Aspect Initiative* (siehe [TMAC04]). Early Aspects sind definiert als Cross Cutting Concerns (Querschnittsthemen) auf der Anforderungs- und Architekturebene. Beispiele solcher Early Aspects sind insbesondere nicht-funktionale Anforderungen wie Performance- und Security-Anforderungen.

## 4.2 Ausblick

Die in diesem Betrag durchgeführte Betrachtung der wechselseitigen Abhängigkeiten zwischen der Systementwicklung und dem Systembetrieb (aus Sicht des Requirements Engineering) hat gezeigt, dass sich auf Basis der Modellierung der Begrifflichkeiten in beiden Bereichen, Wechselbeziehungen klar und deutlich herausstellen lassen.

Für die weitere Forschung erscheint es notwendig, die identifizierten wechselseitigen Abhängigkeiten zu modellieren, dabei zu formalisieren und somit einen Grundstock für eine eventuell automatisierte Weitergabe von Informationen zwischen den entsprechenden Bereichen zu legen. In einem weiteren Schritt könnte auch die partielle automatische Ableitbarkeit der Zentraldokumente Lastenheft und SLA ermöglicht werden.

Nächste Schritte auf diesem Weg wären dabei z.B. der Entwurf eines übergreifenden Modells, das Rollen und Artefakte aus beiden Bereichen (Entwicklung und Betrieb) integriert darstellt. Die in Abschnitt 3 erarbeiteten Ergebnisse können dabei als Grundlage verwendet werden. Um die Integration bzw. Ableitbarkeit der zentralen Dokumente Lastenheft und SLA voranzutreiben, sind allerdings, besonders auf Seiten des ITSM, zusätzliche Formalisierungen und methodische Hilfestellungen notwendig.

Die in dieser Arbeit zum Einsatz gekommene Herangehensweise, die Untersuchung der Beziehungen zwischen diesen beiden Domänen anhand grundlegender Änderungsfälle zu strukturieren, kann auch für ähnliche Fragestellungen angewandt werden. So könnten beispielsweise Wechselbeziehungen zwischen den Rollen, Vorgehensbausteinen und Entscheidungspunkten des V-Modells mit den entsprechenden Elementen des ITIL Service Management analysiert werden.

## Literatur

- [CNEM99] L. Chung, B. Nixon, Yu. E. und J. Mylopoulos. *Non-Functional Requirements in Software Engineering*. International Series in Software Engineering. Springer, 1999.
- [DIN94] DIN Deutsches Institut für Normung e.V. *DIN 66272 – Bewerten von Softwareprodukten: Qualitätsmerkmale und Leitfaden zu ihrer Verwendung*, Oktober 1994.
- [Ent05] Enterprise Management Associates. *Developing Manageable Applications Using ITIL Best Practices – A White Paper Prepared for Hewlett-Packard*, April 2005.
- [GHK<sup>+</sup>01] M. Garschhammer, R. Hauck, B. Kempter, I. Radisic, H. Roelle und H. Schmidt. The MNM Service Model — Refined Views on Generic Service Management. *Journal of Communications and Networks*, 3(4):297–306, Dezember 2001.
- [JBR99] I. Jacobson, G. Booch und J. Rumbaugh, Hrsg. *The Unified Software Development Process*. Addison Wesley, 1999.
- [Koo05] Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung. *V-Modell XT Dokumentation*, 2005. <http://www.v-modell-xt.de/>.
- [Lew99] L. Lewis. *Service Level Management for Enterprise Networks*. Artech House, 1999.
- [Mic04] Microsoft Cooperation. *Microsoft Operations Framework – Version 3.0*, August 2004. <http://www.microsoft.com/mof/>.
- [Off02] Office of Government Commerce (OGC), Hrsg. *Application Management*. IT Infrastructure Library (ITIL). The Stationary Office, Norwich, UK, 2002.
- [Off05] Office of Government Commerce (OGC), Hrsg. *Introduction to ITIL*. IT Infrastructure Library (ITIL). The Stationary Office, Norwich, UK, 2005.
- [OMG04] OMG (Object Management Group). *UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms*, September 2004. <http://www.omg.org/cgi-bin/doc?ptc/2004-09-01>.
- [RR99] J. Robertson und S. Robertson. *Mastering the Requirements Process*. Addison-Wesley, 1999.
- [RR04] J. Robertson und S. Robertson. *Volere Requirements Specification Template*. The Atlantic Systems Guild, 10.1. Auflage, 2004. <http://www.volere.co.uk/template.htm>.
- [Rup04] C. Rupp. *Requirements Engineering und -Management*. Hanser, 3. Auflage, 2004.
- [Ten05] T. Tensi. Anwendungslandschaft: Nachverfolgung der Implementierung von IT-Modellen. In R. Breu, Th. Matzner, F. Nickl und O. Wiegert, Hrsg., *Software-Engineering, Objektorientierte Techniken, Methoden und Prozesse in der Praxis*. Oldenbourg, 2005.
- [TMAC04] B. Tekinerdogan, A. Moreira, J. Araujo und P. Clements, Hrsg. *Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design, Workshop Proceedings, University of TR-CTIT-04-44*. 2004.
- [vdP04] R. van der Pols. *ASL – A Framework for Application Management*. Van Haren Publishing, 2004.