



IT-Service-Kataloge der 2. Generation

Workshop

Frankfurt, 29. November 2005

The Service Architects



Advicio Ingenieurbüro Tonio Grawe

Hauptsitz:	Entwicklungszentrum:
Am Graspoint 15	Peterssteinweg 10
D-83026 Rosenheim	D-04107 Leipzig
+49-8031-2358370	+49-341-9625170
advicio.de	

Datum:	26. November 2005
Autor:	Tonio Grawe
Dateiname:	WS1-Handout-DE-v2.4

Text und Abbildungen wurden mit größter Sorgfalt erarbeitet. Herausgeber und Autoren übernehmen jedoch für eventuell verbliebene fehlerhafte Angaben und deren Folgen keine Haftung.

Copyright © 2005 Advicio Ingenieurbüro Tonio Grawe, Rosenheim, Germany

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Funksendung, der Wiedergabe auf fotomechanischem oder ähnlichem Wege und der Speicherung in Datenverarbeitungsanlagen bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. .

Advicio® ist eine registrierte Marke in Austria, Belgium, China, France, Germany, Italy, Luxembourg, Poland, Russian Federation, Spain, Switzerland, The Netherlands, Turkey, United Kingdom.

Inhaltsverzeichnis

1 Einleitung und Intension.....	4
1.1 Advicio's Service Provider Maturity Model (SPMM).....	6
1.2 Der Service-Katalog in der Kundenbeziehung.....	8
1.3 Interner Nutzen eines Service-Katalogs.....	8
1.4 Der Service-Katalog zur Steuerung von Lieferanten.....	10
2 Strukturierung eines Service-Katalogs.....	12
2.1 Eine Komponentenbasierte Service-Architektur.....	12
2.2 Dekomposition.....	13
2.3 Taxonomie.....	14
3 Paketbildung.....	18
4 Datenmodellierung und Wissensrepräsentation.....	20
4.1 Service Engineering und Configuration Management.....	20
4.2 Wie die Informationen abgebildet werden können.....	22
4.3 Service Specification and Configuration Language.....	22
5 Service Life Cycle.....	24
5.1 Abbildung von Prozessen und Zuständen.....	24
5.2 Lebenszyklus der Spezifikationen.....	24
5.3 Zustandsmodell der Instanzen.....	26
6 Application.....	27
6.1 Eine Anwendungsarchitektur für IT-Dienstleister.....	27
6.2 CMDB.....	28
6.3 Advicio Service Workbench.....	28
7 Summary.....	30

1 Einleitung und Intension

Das IT-Service-Engineering, so nennen wir die systematische Entwicklung von IT-Dienstleistungen, wird zusehends zur Kernkompetenz von IT-Dienstleistern, weil sich nur noch professionell entwickelte Service-Produkte am Markt behaupten können und von den Service-Konsumenten akzeptiert werden. Die Herausforderung vom IT-Service-Engineering ist es, die IT-Services so zu spezifizieren, dass Leistungsnehmer und Leistungsgeber ein unmissverständliches Bild von den IT-Dienstleistungen haben.

„Produktisierung“ ist ein Zeichen von Reife

Professor Hans-Jörg Bullinger, der Präsident der Fraunhofer Gesellschaft, sprach in seiner Eröffnungsrede bei der Konferenz „IT-Service Engineering 2005“¹ von der „Produktisierung“ von IT-Services. IT-Dienstleister müssten die Entwicklungsphase für IT-Services ernst nehmen, sollten hier in die Vorleistung gehen und ausgereifte Dienstleistungen am Markt anbieten, anstatt die IT-Services während der Provisionierung zu entwickeln. In diesem Zusammenhang wächst das Verständnis, dass IT-Services auch Produkte sind, die entwickelt und über den ganzen Lebenszyklus betreut werden müssen.

Diese Entwicklung spiegelt sich auch in dem Reifegradmodell SPMM von Advicio wieder, das in Kapitel 1.1 kurz vorgestellt wird. IT-Dienstleister, die heute individuelle Services per SLA verwalten, gehen dazu über, ein Service-Portfolio zu entwickeln und stellen sich zunehmend dem Wettbewerb. So entsteht ein Markt für IT-Services, die vordefiniert sind und von Kunden abgerufen werden können.

Definition von IT-Service

Als IT-Service bezeichnen wir die Dienstleistung, Informations- und Kommunikationstechnik zu betreiben und für den Anwender nutzbar zu machen. Die Dienstleistung wird durch die Prozesse des Dienstleisters erbracht, denen deshalb eine besondere Beachtung zukommt. Zudem müssen Ressourcen eingesetzt werden: vor allem Personal, aber auch Hardware, Software und Infrastrukturen wie etwa Netzwerke.

Typische Beispiele solcher IT-Services sind der Betrieb einer Geschäftsanwendung (etwa SAP R/3), der Betrieb eines VPN für ein Filialnetz oder die Bereitstellung eines PC-Arbeitsplatzes und eines Telefons für einen Mitarbeiter des Servicenehmers.

Der Markt für IT-Services

Wir kennen zwei Gruppen von IT-Dienstleistern:

1. IT-Dienstleister, die sich am freien Markt behaupten müssen. Diese werden gelegentlich als Externe IT-Dienstleister bezeichnet. Hierzu zählen Anbieter von IT-Outsourcing-Services und Application Service Provider, aber auch Anbieter, welche keine Vollsortimenter sind, sondern sich auf bestimmte Teile (meist Technologien) spezialisieren.
2. IT-Dienstleister, die als Teil einer Organisation (oft ein Konzern) ihre Leistungen innerhalb der Organisation anbieten. Aufgrund dieser Zugehörigkeit werden diese Anbieter auch interne IT-Dienstleister genannt. Sie treten in der Regel als Vollsortimenter auf, wobei häufig Teilleistungen von extern eingekauft werden und der interne IT-Dienstleister die Rolle des Service-Integrators einnimmt.

1 www.it-service-engineering.de

Beide Gruppen von IT-Dienstleistern teilen sich den Markt für IT-Services und stehen einerseits im Wettbewerb und kooperieren andererseits. Nach Untersuchungen des European Information Technology Observatory (EITO) hat der IT-Markt in Deutschland ein Volumen von 63 Milliarden Euro (vgl. Abb. 1). Dies schließt Umsätze für Hardware und Softwarelizenzen ein. Mit dem Verständnis, dass Hardware und Software neben Personal die Ressourcen sind, die für die Erbringung von IT-Services eingesetzt werden, setzen wir den von EITO definierten IT-Markt mit dem Markt für IT-Services gleich.

	2003 Value	2003 %	2003/02 %	2004/03 %	2005/04 %
France	49	17,1	-1,9	2,4	4,3
Germany	63	22,0	-1,7	1,2	3,0
Italy	24	8,4	-2,7	1,7	4,1
Spain	12	4,0	3,3	5,4	7,7
UK	64	22,2	-0,3	2,4	4,3
Other WE	76	26,3	-1,2	2,4	4,4
Western Europe	287	100,0	-1,2	2,4	4,4

Abbildung 1: Das Volumen des IT-Markts in Westeuropa in Milliarden Euro nach EITO.
Quelle: EITO 2004

Das Marktvolumen von 63 Milliarden Euro entspricht etwa drei Prozent vom deutschen Brutto sozialprodukt. Dies stimmt also mit dem Erfahrungswert überein, dass Unternehmen zwischen zwei und fünf Prozent ihres Umsatzes als IT-Budget ansetzen. Der Privatkundenmarkt für IT-Services ist derzeit noch vernachlässigbar.

Nutzen von IT-Service-Katalogen

IT-Services sind Dienstleistungen. Dienstleistungen wurden in der wissenschaftlichen Literatur als immateriell und nicht lagerbar definiert. Salopp gesagt, können Dienstleistungen einem nicht auf den Fuß fallen. Weil Dienstleistungen nicht greifbar sind, tun sich IT-Dienstleister auch so schwer, ihre Leistungen genau zu beschreiben. Bei der Kommunikation mit Kunden, aber auch bei der Zusammenarbeit innerhalb der Organisation und mit Partnern ist die fehlende Leistungsbeschreibung sehr hinderlich.

Kapitel 1.2 stellt dar, welche Rolle der Service-Katalog in der Kundenbeziehung hat. Kapitel 1.3 geht auf den Nutzen des Service-Katalogs für den Dienstleister selbst ein. Aber auch die Lieferantenbeziehungen beim Outsourcing können mit einem Service-Katalog effizient gesteuert werden. Dies ist in Kapitel 1.4 beschrieben.

Einführung und Besetzung neuer Rollen

Mit der Entwicklung im Reifegrad muß bei IT-Dienstleistern eine neue Rolle besetzt werden, die des Produktmanagers. Der Produktmanager vertritt die Interessen des Produktes. Er oder sie verantwortet ein IT-Service-Produkt über den ganzen Lebenszyklus, von der Entwicklung, der Vermarktung, dem Betrieb bis zum Assessment. Der Produktmanager muss dafür sorgen, dass die notwendigen Ressourcen und auch die Prozesse auf das Produkt abgestimmt sind.

Die Aufgabe des Produktmanagers mag zunächst als Overhead und Sisyphus-Aufgabe erscheinen. Bei der großen Menge von IT-Services, welche die meisten IT-Dienstleister im Angebot haben, ist der Aufwand im Produktmanagement entsprechend hoch. Deshalb wird auch das Portfoliomanagement eine wichtige neue Aufgabe. Während bisher praktisch jeder IT-Dienstleister ein Vollsortimenter war, gilt es nun, ein Portfolio strategisch zu definieren. Nicht alle IT-Services sollen selbst erbracht werden, sondern nur diejenigen, mit denen die höchste Wertschöpfung erzielt werden kann. Andere IT-Services sind oftmals günstiger zuzukaufen („selektives Outsourcing“).

1.1 Advicio's Service Provider Maturity Model (SPMM)

Die Veränderungen bei IT-Dienstleistern lassen sich in einem Reifegradmodell abbilden. Aus den Entwicklungsschritten, die wir bei IT-Dienstleistern in den vergangenen fünf bis zehn Jahren beobachtet haben und die wir für die nahe Zukunft erwarten können, ist das von Advicio entwickelte Reifegradmodell Service Provider Maturity Model (SPMM) entstanden. Darin wird die Bandbreite von der internen IT-Abteilung, die als Kostenstelle organisiert ist, bis hin zu unabhängigen, wettbewerbsfähigen, am freien Markt agierenden Anbietern von IT-Services betrachtet.

Advicio hat diese Entwicklung in dem Reifegradmodell Service Provider Maturity Model (SPMM) charakterisiert. Dabei wurden fünf Reifegradstufen identifiziert, die danach benannt sind, worauf der IT-Dienstleister in der jeweiligen Stufe seinen Fokus gerichtet hat:

Infrastruktur: Auf der untersten Reifegradstufe konzentriert sich der Dienstleister völlig auf den Betrieb der IT-Infrastruktur. Definierte IT-Services existieren noch nicht. Die Vorgänge laufen ad hoc und ungesteuert ab.

Prozesse: Der Dienstleister beginnt, seine Prozesse zu definieren und zu dokumentieren. Als Hauptaufgabe wird weiterhin der Betrieb der Infrastruktur gesehen. Dies geschieht nun jedoch gesteuert und nach festgelegten Vorgaben, etwa ITIL. Dienstleistungen sind noch nicht definiert.

Anwender: Auf dieser Reifegradstufe werden Dienstleistungen definiert. Der Dienstleister erkennt in ihnen den Mehrwert, den er dem Kunden anbieten kann. Es werden Service Level Agreements (SLA) abgeschlossen, in denen die Dienstleistungen beschrieben werden und die Dienstgüte festgelegt wird. Der Anbieter versucht dabei jeden Kundenwunsch zu erfüllen und die SLAs genau an die Anforderungen des Kunden anzupassen.

Produkte: Der Schritt in diese Reifegradstufe wird vollzogen, wenn der Anbieter seine Dienstleistungen nicht mehr speziell auf jeden Kunden zuschneidet, sondern im Voraus entwickelt und ein Produktportfolio mit vordefinierten Dienstleistungsprodukten zusammenstellt. Die Standardisierung von Dienstleistungen hat eine deutliche Effizienzsteigerung zur Folge, da die Erbringungsprozesse auf wenige, klar definierte Dienstleistungsprodukte optimiert werden können. Dafür büßt der Anbieter die Anpassungsfähigkeit seiner Dienstleistungen an spezielle Kundenwünsche ein. Um diesem negativen Effekt entgegenzuwirken werden viele Dienstleistungen konfigurierbar gestaltet, d.h. die Dienstleistung selbst und damit auch ihr Erbringungsprozess ist standardisiert, der Kunde kann jedoch die Dienstgüte innerhalb definierter Grenzen anpassen.

Markt: Auf der höchsten Reifegradstufe ist der Dienstleistungsanbieter am freien Markt positioniert. Die Entwicklung von Reifegradstufe 4 auf 5 geschieht durch die konsequente Orientierung an den Bedürfnissen des Marktes sowie dem Wettbewerb.

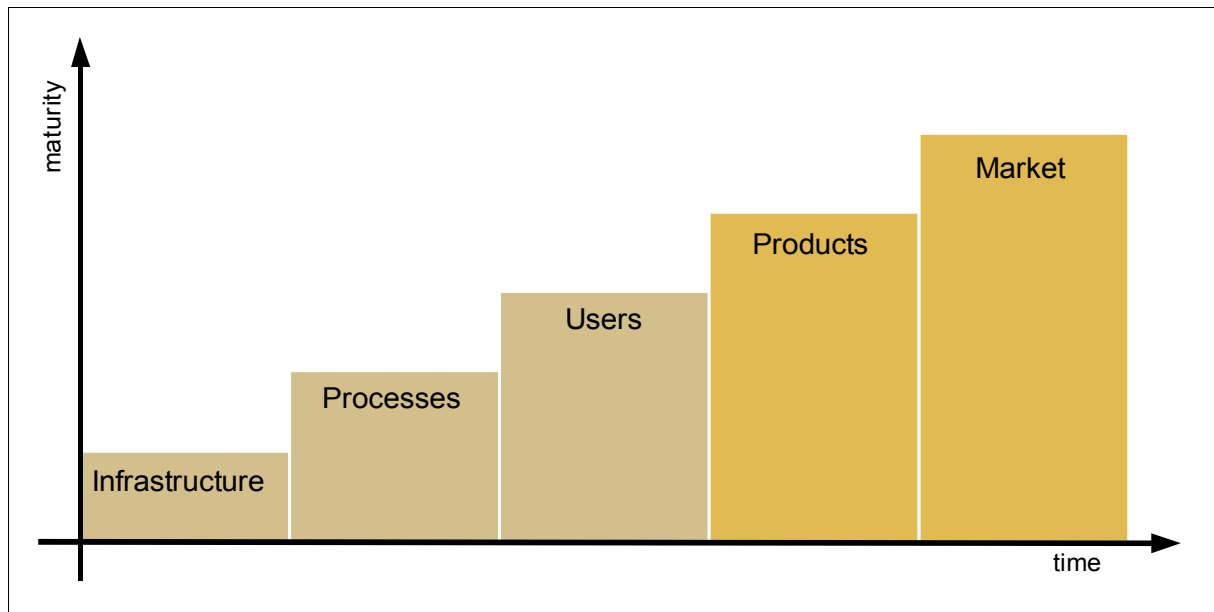


Abbildung 2: Die fünf Entwicklungsstufen im Reifegradmodell SPMM sind danach benannt, was in der jeweiligen Stufe fokussiert wird.

Die in dem Modell abgebildete Entwicklung lässt sich mit einfachen Worten zusammenfassen: Weg vom Einzelfertiger, hin zur Service-Fabrik. Die heute schon stark verbreiteten Service Level Agreements (SLA), das sind Vereinbarungen, welche die Dienstleistung und deren Qualität spezifizieren, führen bei den Dienstleistern zu einem Produktverständnis. Die Dienstleister versuchen, einen einmal spezifizierten und implementierten IT-Service an weitere Kunden zu verkaufen. Damit gehen sie einen wichtigen Schritt: Während früher IT-Services im Kundenauftrag entwickelt wurden, im Rahmen von aufwendigen und riskanten Projekten, gehen die IT-Dienstleister zukünftig in die Vorleistung, die Services zu entwickeln.

Hinweis: Advicio M-Check

M-Check ist eine Dienstleistung von Advicio, mit der das Optimierungspotential von IT-Dienstleistern festgestellt werden kann. Es wird aufgezeigt, mit welchen Massnahmen der Reifegrad des IT-Dienstleisters erhöht werden kann. M-Check basiert auf dem Service Provider Maturity Model (SPMM) von Advicio.

Mit M-Check werden Mitarbeiter in verschiedenen Rollen interviewt. Die Antworten werden statistisch ausgewertet und mit dem Referenzmodell verglichen. Diese Interviews werden webbasiert durchgeführt. Advicio kann den M-Check zu einem sehr attraktiven Preis anbieten, denn durch den Einsatz der Web-Technologie wird der Aufwand für M-Check minimal gehalten. Die Interviews werden mit mehreren Teilnehmern durchgeführt. Dadurch wird einerseits die persönliche Perzeption eines jeden Teilnehmers relativiert, andererseits wird sichergestellt, dass zu allen Fragen auch bei sehr großen IT-Organisationen, kompetente Antworten gefunden werden können.

Im Anschluss an die Interviews wird ein erfahrener Analyst eine individuelle Interpretation vornehmen, sowie Handlungsempfehlungen daraus ableiten. Dieses Ergebnis wird innerhalb einiger Tage nach Abschluss der Interview-Phase dem Kunden als Dokument zugeschickt. Eine Vorstellung und Diskussion im Management-Kreis ist optional möglich.

1.2 Der Service-Katalog in der Kundenbeziehung

In der Kundenbeziehung kommt dem Service-Katalog die Aufgabe zu, die Leistungen zu präsentieren und damit den Verkauf zu unterstützen. Weil jedoch IT-Services eng an die Geschäftsprozesse des Kunden gebunden sind, besteht sehr oft die Notwendigkeit, die IT-Services an die gelebten Prozesse anzupassen. Einige Dienstleister haben etwa bei Application Service Providing (ASP) von SAP R/3 die Erfahrung machen müssen, dass IT-Services nicht von der Stange gekauft werden, sondern stark an die Kundenbedürfnisse angepasst werden müssen.

Der scheinbare Widerspruch von standardisierten Service-Produkten und kundenspezifischen Anforderungen lässt sich im Sinne von Mass Customization durch die Produktkonfiguration lösen. Dabei kann ein Kunde durch eine Vielzahl von Optionen sein Produkt nach den persönlichen Bedürfnissen gestalten. Beim Mass Customization wird Massware hergestellt und doch jedes Stück mit Individualität ausgestattet.

Es besteht jedoch die Gefahr, dass eine Vielzahl von angebotenen Services und Konfigurationsmöglichkeiten den Kunden überfordert, dass aus Mass Configuration „mass confusion“ wird. Ein IT-Dienstleister hat bis zu 1000 Service-Komponenten. Nur ein kleiner Teil der möglichen Permutationen ist sinnvoll und erlaubt. Der Service-Katalog muss den Kunden zielgerichtet zu einem passenden Angebot führen.

1.3 Interner Nutzen eines Service-Katalogs

Mit dem Verständnis, dass die IT-Services die Produkte des IT-Dienstleisters sind, liegt die Frage auf der Hand, was bei IT-Dienstleistern das Produkt-Daten-Management (PDM) ist. Informationen zu den Service-Produkten werden von der gesamten Organisation zur Erbringung der Leistungen ständig benötigt. Ein umfassender IT-Service-Katalog übernimmt die Aufgabe des PDM und hilft damit, für die IT-Organisation wichtige strategische Ziele zu erreichen:

Innovation wird bei einem IT-Dienstleister nicht geschaffen, indem neue IT-Infrastruktur eingeführt wird. Vielmehr entsteht Innovation, indem das Dienstleistungsangebot kontinuierlich an den Anforderungen der Kunden ausgerichtet wird. Neuentwicklungen sind jedoch teuer und langatmig. Durch eine fundierte Dokumentation der bestehenden Leistungen können Teilleistungen (Service-Komponenten) in neuen Dienstleistungen wiederverwendet werden. Dadurch müssen für neue Service tatsächlich nur etwa 20% der Dienstleistung neu entwickelt werden. Eine schneller „time to market“ ist die Folge.

Effizienz. Vor dem Hintergrund einer angespannten Wirtschaftslage und der Wettbewerbssituation kommt den Kosten besondere Aufmerksamkeit zu. Effizienz heißt, den Aufwand minimal zu halten. Aufwand umfasst sowohl die eingesetzten Betriebsmittel, als auch die Arbeitsprozesse. Es gibt also zwei Ansatzpunkte, effizient zu werden: Erstens, die eingesetzte Informationstechnologie möglichst gut auszunutzen, also durch Kapazitätsmanagement die Planung des Ressourceneinsatzes zu optimieren. Und zweitens, die Prozesse des IT-Dienstleisters zu optimieren, integrieren und automatisieren, sowohl in der Bereitstellung als auch im Betrieb.

Kundenorientierung: IT-Services können nicht von der Stange verkauft werden, sondern müssen sich an den Anforderungen des Kunden orientieren. Der bereits erwähnte Ansatz der Mass Customization lässt sich vergleichen mit den Stücklisten, die aus der diskreten Fertigung von Gütern bekannt sind. Damit er sich für IT-Services umsetzen lässt, muss der Dienstleister seine Teilleistungen (Service-Komponenten) genau kennen: Welche Organisationseinheiten involviert sind, welche Prozesse durchlaufen werden und welche Ressourcen eingesetzt werden.

Preisbildung: Der Preis eines Produkts ist ein entscheidendes Merkmal, denn er ist offensichtlich und leicht vergleichbar. Die Preisbildung ist ein Differenzierungsmerkmal, nicht nur die Höhe des Preises, sondern auch dessen Zusammensetzung und Nachvollziehbarkeit. Heute bestehen überwiegend Pauschalbeträge in den Preislisten der IT-Dienstleister. Befürworter von Reformen verlangen Preise, die anhand der Produktionskosten der Dienstleistung kalkuliert sind, die nutzungsabhängige Anteile enthalten. Das PDM, also der IT-Service-Katalog, ist die Kalkulationsgrundlage.

Benchmarking & Controlling: Viele für die Steuerung der IT-Organisation interessante Kennzahlen haben einen Bezug zu den Produkten, beispielsweise die Kundenzufriedenheit, Qualität, Umsätze und andere Betriebskennzahlen. Aufgrund der Immaterialität der Service-Produkte besitzen diese jedoch keine Messpunkte. Die Produktdaten können aber eine Aggregation von Messwerten von Ressourcen ermöglichen, womit Key-Performance-Indikatoren von den Service-Produkten erhoben werden können.

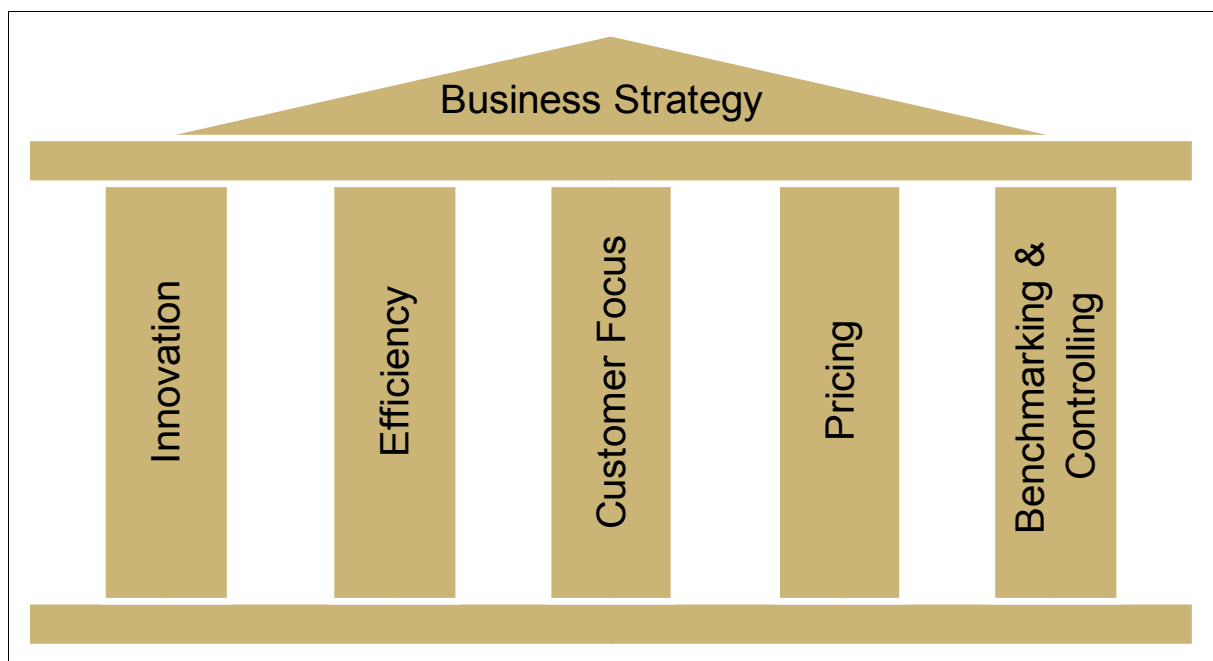


Abbildung 3: Der IT-Service-Katalog ist ein essentielles Werkzeug zur Umsetzung geschäftsstrategischer Ziele von externen IT-Dienstleistern und IT-Profit-Centern.

Schlussfolgerung 1: Produktdatenmanagement

Ein Service-Katalog der 2. Generation beschreibt nicht nur den Service aus Kundensicht, sondern auch die interne Sicht des Dienstleisters. D.h. jegliches Wissen zu dem Service wird darin zusammengeführt. Der Service-Katalog ist das Produkt-Daten-Management des IT-Dienstleisters.

1.4 Der Service-Katalog zur Steuerung von Lieferanten

„Make-or-buy“-Entscheidungen stehen bei IT-Dienstleistern ständig auf der Tagesordnung, um das eigene Dienstleistungsportfolio zu optimieren. Obwohl die Lieferantenbeziehungen einen wichtigen Teil in der Strategie eines jeden IT-Dienstleisters haben, ist das Partner-Management noch eine recht junge und wenig ausgereifte Disziplin bei den IT-Dienstleistern.

IT-Service-Provider hängen in großem Maße von ihren Lieferanten ab. IT-Services, wie Dienstleistungen allgemein, unterscheiden sich wesentlich von Produktionsgütern. IT-Services können nicht auf Vorrat produziert werden, sondern werden in Echtzeit erbracht. Die Erbringer von Teilleistungen arbeiten nicht in einem definierten Prozess mit diskreten Fertigungsschritten nacheinander an dem Produkt, sondern müssen zeitgleich zusammenarbeiten. Die Qualitätssicherung von IT-Services erfolgt nicht als Prozess mit eigenem Zeitfenster, sondern während die Dienstleistung erbracht wird. Die Herausforderung liegt in der Kontinuität, mit der IT-Services betrieben werden müssen.

Der Service-Kunde möchte das Beziehungsnetzwerk, durch das die konsumierte Dienstleistung erbracht wird, nicht kennen müssen. Der Kunde möchte einen einzigen Ansprechpartner haben, der die Verantwortung übernimmt. Einen solchen Generalunternehmer bezeichnen wir als Service-Integrator. Der Service-Integrator hat die Aufgabe, die verschiedenen Teilleistungen aufeinander abzustimmen, sprich: deren Zusammenspiel sicherzustellen.

Bei diesem Geschäftsmodell wird eine höherwertige Dienstleistung dadurch geschaffen, dass mehrere Teilleistungen integriert werden. Dieses Geschäftsmodell wird in der Literatur auch Hub Model genannt und ist in Bild 2 dargestellt.

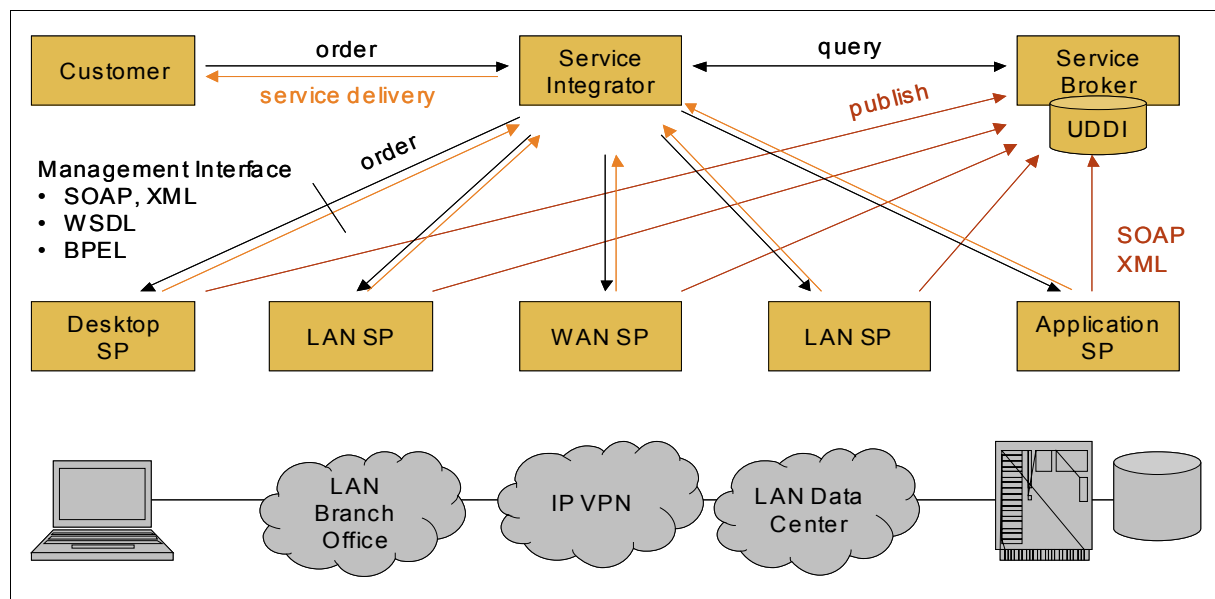


Abbildung 4: Der Service-Integrator erfüllt die Anforderungen des Kunden an den IT-Service durch die geschickte Kombination zugekaufter Teilleistungen.

Ob die Erbringer der einzelnen Teilleistungen externe Lieferanten sind, die ihre Leistungen am freien Markt anbieten, oder ob es sich dabei um konzernangehörige Organisationseinheiten handelt, ist irrelevant. In der Regel werden beide Formen auftreten und für den Service-Integrator sind alles Lieferanten.

Während beim Aufbau einer Lieferantenbeziehung viel „Handarbeit“ gefordert ist, bestehen im operativen Bereich gute Ansätze zur Automatisierung:

1. Beim *Fulfillment*, also dem Beauftragen und einrichten neuer IT-Services im Rahmen einer bestehenden Geschäftsbeziehung.

2. Beim *Assurance*, also der kontinuierlichen Qualitätssicherung. Während das Fulfilment sich nicht wesentlich von Bestellvorgängen anderer Branchen unterscheidet, unterscheidet das Assurance die Dienstleistungen von Produktionsgütern.

Eine besondere Herausforderung ist die Service-Konfiguration. Um die einzelnen Komponenten sinnvoll aufeinander abzustimmen, ist eine detaillierte Kenntnis über deren Funktionalität und Zusammenspiel nötig. Die reine Prozessintegration über mehrere Organisationen hinweg reicht dafür nicht aus. Vielmehr muss ein Wissensmodell dieser Service-Komponenten den beteiligten Partnern Auskunft geben können über das Verhalten, die Voraussetzungen und die Konfigurationsmöglichkeiten der Service-Komponenten.

Technologiestandards helfen, die Lieferkette zu automatisieren. Dazu müssen zweierlei Themen definiert werden:

1. Die Datenformate, die heute typischerweise auf XML basieren und durch eine Document Type Definition (DTD) oder XML-Schemata definiert werden.
2. Die Kommunikationswege, die meist durch IP-Protokolle wie http/s, FTP, SMTP oder SOAP.

Die Thematik gewinnt mit steigendem Kostenbewusstsein an Relevanz. IT-Dienstleister tun gut daran, den Lieferanten einen angemessenen Platz in ihrer Strategie einzuräumen und deren Integration schon heute konsequent voranzutreiben. Automatisierung muss in diesem Bereich mittelfristig umgesetzt werden. Ein freier Handel standardisierter IT-Dienstleistungen hat bereits eingesetzt. Wir erwarten, dass die Supply-Chain-Automation für IT-Dienstleister bis 2009 zur „conditio sine qua non“ wird.

2 Strukturierung eines Service-Katalogs

2.1 Eine Komponentenbasierte Service-Architektur

Heute sind die meisten IT-Dienstleistungen monolithisch aufgebaut. Im Gegensatz dazu verfolgt Advicio einen komponentenbasierten Ansatz. Dienstleistungen setzen sich aus verschiedenen Komponenten zusammen. Werden diese Komponenten spezifiziert und standardisiert, dann lässt sich eine Plattformstrategie fahren, bei der jede Komponente in vielen verschiedenen Produkten eingesetzt wird.

Der Begriff "Plattformstrategie" ist aus der Automobilindustrie bekannt. Gemeint ist damit eine Standardisierung von Komponenten. Die Idee, Komponenten in verschiedenen Modellen wiederzuverwenden, hat Automobilkonzernen geholfen, die Produktion und die Logistik zu optimieren.

Dieser Ansatz kann auch erfolgreich in den IT-Betrieb übernommen werden. Die Produkte (respektive Autos) sind IT-Dienstleistungen, die dem Kunden verkauft werden. Die Komponenten (respektive Motor, Fahrwerk) sind Teilleistungen. Das Ziel ist nun, durch Standardisierung eine Teilleistung möglichst nur einmal zu erbringen und diese in möglichst viele Produkte einfließen zu lassen.

So können schnell und kostengünstig neue Dienstleistungsprodukte entwickelt werden, weil ein Großteil des neuen Produkts sich aus bestehenden Komponenten zusammensetzt.



Abbildung 5: Plattform-Strategien wurden in der Automobilbranche erfolgreich eingeführt.

Dieser Ansatz erfordert eine Differenzierung zwischen Service-Produkten und Service-Komponenten, die wie folgt definiert sind:

Service-Komponenten sind granulare Dienstleistungsmodule, die sich aufgrund ihrer Abstraktion und Konfigurierbarkeit wiederverwenden lassen.

Service-Produkte sind verkaufbare Dienstleistungen, die sich aus einer definierten Menge von Service-Komponenten zusammensetzen.

Schlussfolgerung 2: Produkte und Komponenten

Eine komponentenbasierte Service-Architektur erfordert die Unterscheidung zwischen Service-Produkten und Service-Komponenten. Entsprechend gibt es einen Produktkatalog und einen Komponentenkatalog.

2.2 Dekomposition

Durch eine Dekomposition werden bestehende Service-Angebote in ihre Bestandteile zerlegt. Dazu werden die Services in sinnvolle Funktionsbereiche aufgeteilt, die dann als Service-Komponenten identifiziert sind.

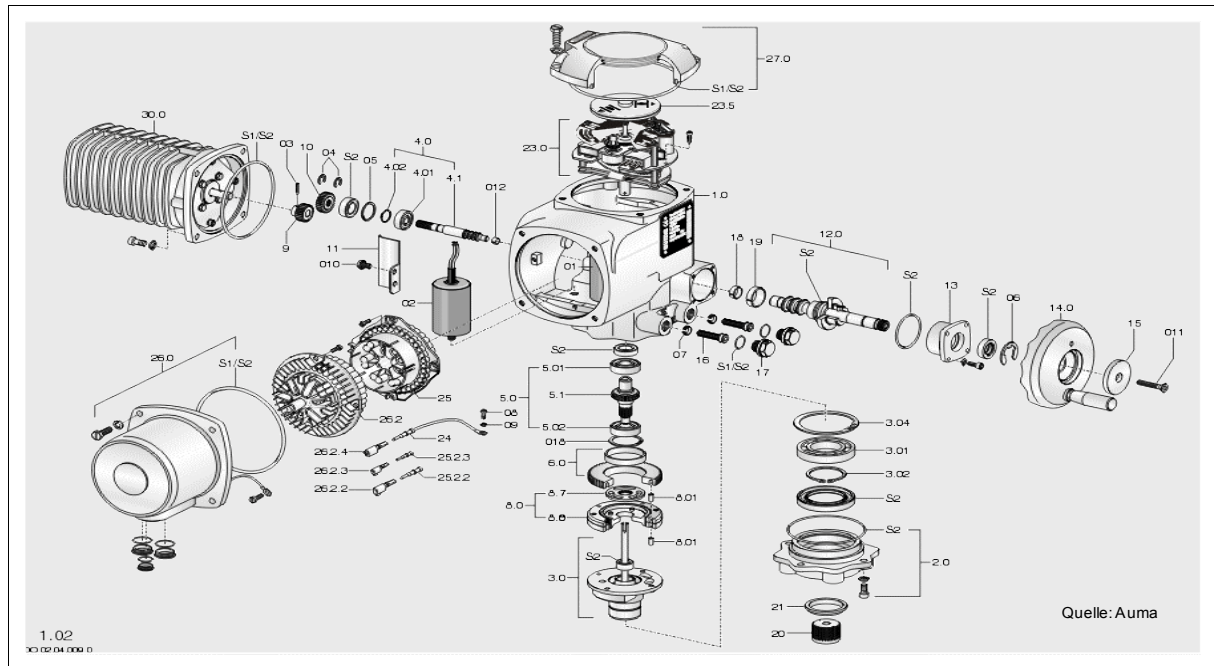


Abbildung 6: Die Dekomposition von IT-Services entspricht dem Zerlegen materieller Güter in ihre Bestandteile.

In mehreren Runden können die Service-Komponenten weiter zerlegt werden. Ziel ist es, die Größe der Komponenten so weit zu reduzieren, dass sie überschaubar sind und leicht dokumentiert werden können.

Je höher die Granularität der Teilleistungen ist, desto besser wird später das Ziel der Wiederverwendbarkeit erreicht. Durch die mehrstufige Dekomposition wird eine komplexe Dienstleistung in atomare Service-Komponenten zerlegt und schließlich einzelnen Ressourcen, sei es Personal oder Infrastruktur, zugeordnet.

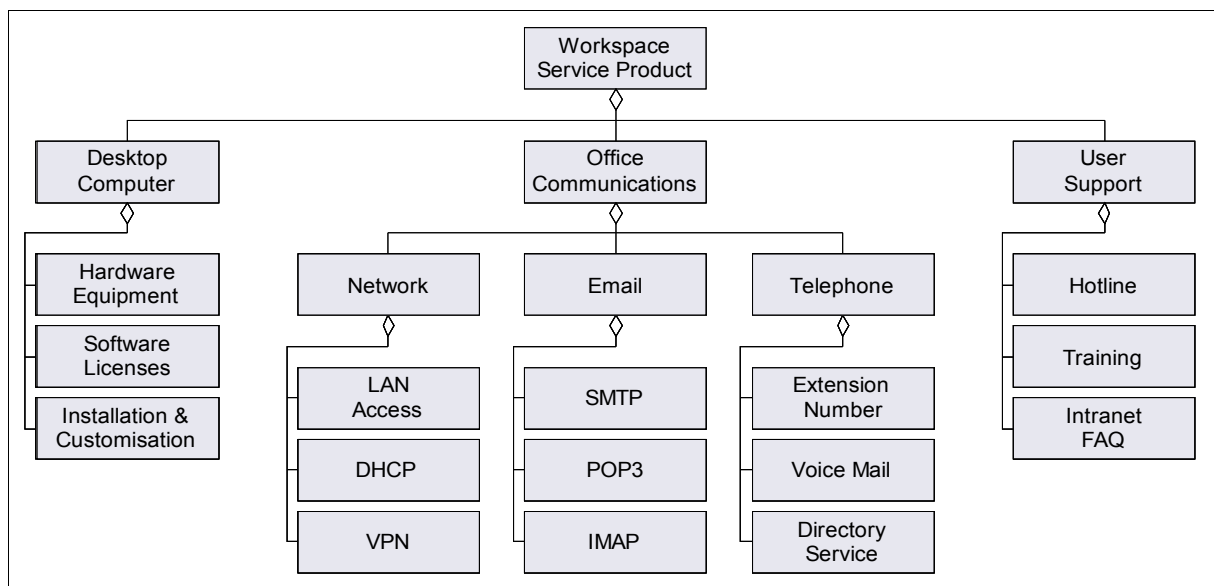


Abbildung 7: Beispielhafte Dekomposition des IT-Services „Workspace“.

Schlussfolgerung 3: Charakteristika von Service-Komponenten

Service-Komponenten sollten einen sehr begrenzten Umfang haben, um sie "greifbar" zu machen. Service-Komponenten können über mehrere Stufen dekomponiert (zerlegt) bzw. komponiert (zusammengebaut) werden. Eine höhere Granularität erhöht die Wahrscheinlichkeit der Wiederverwendung. Typisch ist, dass IT-Dienstleister mehrere hundert Service-Komponenten haben.

2.3 Taxonomie

Durch die Dekomposition alleine wird das erhoffte Ziel, die identifizierten Service-Komponenten als Bausteine bei der Entwicklung neuer Services wiederzuverwenden, noch nicht erreicht. Der Grund ist die Inkompatibilität der Service-Komponenten zueinander. An dieser Stelle muss ein Entwicklungsprozess einsetzen, durch den die Komponenten aneinander angepasst werden. Nur so können die Komponenten in einem Netzwerk gegenseitiger Abhängigkeiten zusammenarbeiten. Die beiden häufigsten Ursachen von Inkompatibilitäten sind:

1. Eine unterschiedliche Terminologie. Beispielsweise wenn für eine Komponente „QoS“ definiert ist, obwohl Verfügbarkeit gemeint ist. Und wie ist eigentlich „Verfügbarkeit“ definiert? Eine gemeinsame Sprache muss durch ein Glossar geregelt sein.
2. Grundsätzliche Design-Aspekte. Dies betrifft einheitliche Regelungen für Verfügbarkeiten, Betriebszeiten et cetera. Sind solche Parameter für jede Service-Komponente individuell festgelegt, so ist es mutig bis riskant, für einen daraus aggregierten Service eine Qualitätsgarantie abzugeben.

Reduktion der Anzahl

Typischerweise sind es zu viele Service-Komponenten. Mehrere hundert sind üblich, in Einzelfällen wurden auch über tausend Service-Komponenten identifiziert. Eine reiche Auswahl ist in einem gut sortierten Service-Baukasten erwünscht. Allerdings wirken sich mehrfach erfasste, identische Service-Komponenten kontraproduktiv aus. Erfahrungsgemäß kann die Anzahl durch Konsolidierung um 15% reduziert werden. Je nach Historie des IT-Dienstleisters – insbesondere Fusionen wirken sich aus – kann die Konsolidierung auch weit größere Effekte haben.

Einführung von Varianten

Desweiteren hilft die Einführung von Varianten, die Anzahl zu minimieren. Allerdings ist die Entscheidung, ob zwei ähnliche Service-Komponenten Varianten voneinander sind, oder ob es sich doch um verschiedene Service-Komponenten handelt, nicht trivial. Advicio empfiehlt diesbezüglich folgende Regelung: Varianten haben eine identische Funktionalität und unterscheiden sich lediglich in der Qualität (Verfügbarkeit, Bereitstellungsdauer, Betriebszeit, etc.), in der Kapazität und im Preis bzw. in den Kosten.

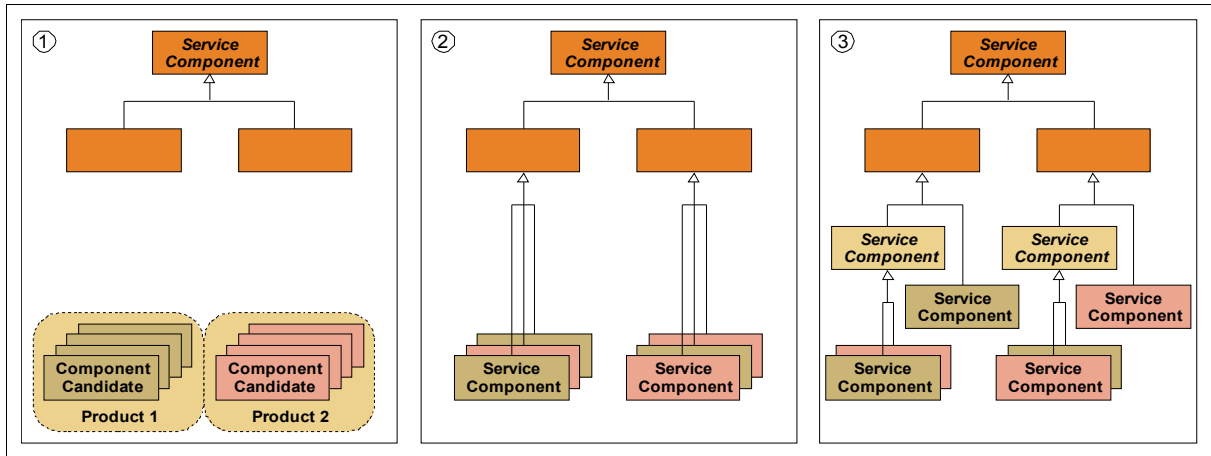


Abbildung 8: Bildung von Komponentenvorschlägen ausgehend von der Basisklassifikation.

Übersicht durch Strukturierung

Im nächsten Schritt soll die Übersichtlichkeit erhöht werden, indem die Service-Komponenten in sinnvolle Gruppen eingeteilt werden. Doch nach welchem Kriterium soll gruppiert werden? Drei Ansätze sind augenscheinlich möglich:

1. Nach der Organisationsstruktur, also welche Einheit die Leistung erbringt.
2. Nach der verwendeten Technologie.
3. Nach der durch den Service bereitgestellten Funktionalität.

Eine Reihe von Gründen spricht für die Funktionalität als Kriterium für die Gruppenbildung. So entsteht aus den Teilleistungen „Betrieb Oracle“, „Betrieb DB2“ und „Betrieb MySQL“ die Gruppe „Betrieb RDBMS“.

Die Gruppe ist eine Oberklasse. Wir bezeichnen diese auch als abstrakte Service-Komponente. Abstrakte Service-Komponenten können nicht instantiiert werden, hingegen können konkrete Service-Komponenten wie „Betrieb MySQL“ tatsächlich umgesetzt werden. Die abstrakten Oberklassen dienen lediglich der Strukturierung des Komponentenkatalogs.

Entsprechend kann nun wiederum für mehrere Gruppen eine gemeinsame Oberklasse gefunden werden, so dass letztlich alle Service-Komponenten in einer übersichtlichen Baumstruktur einsortiert sind. Eine solche Struktur nennen wir eine Service-Taxonomie.

Semantischen Beziehungen der Service-Komponenten

Wie in der Abbildung zu sehen ist, entstehen jede Menge dieser abstrakten Oberklassen, deren Namen gemäß UML-Notation kursiv geschrieben werden. Was ist der praktische Nutzen von diesem Overhead?

Wie eingangs beschrieben, unterscheidet sich eine Service-Architektur von einer Ansammlung unterschiedlicher Service-Komponenten durch die Beziehungen der Service-Komponenten zueinander. Durch die Service-Taxonomie werden die semantischen Beziehungen der Service-Komponenten hergestellt. Dadurch wird semantisches Wissen erfasst. Wenn man beispielsweise nicht wüsste, was *MySQL Operation SC* sein könnte, so verrät die Taxonomie, dass es sich um den Betrieb einer relationalen Datenbank handelt.

Die semantischen Beziehungen helfen auch im IT-Service-Engineering bei der Spezifikation von Service-Komponenten. Soll etwa definiert werden, dass ein Anwendungsservice die Betriebsdienstleistung einer relationalen Datenbank erfordert, ohne festlegen zu wollen, welche dies ist, so wird auch auf eine abstrakte Oberklasse verwiesen. Man erhält dadurch einen Freiheitsgrad, den man ohne Taxonomie nur durch wartungsintensive Konfigurationsregeln erreichen könnte.

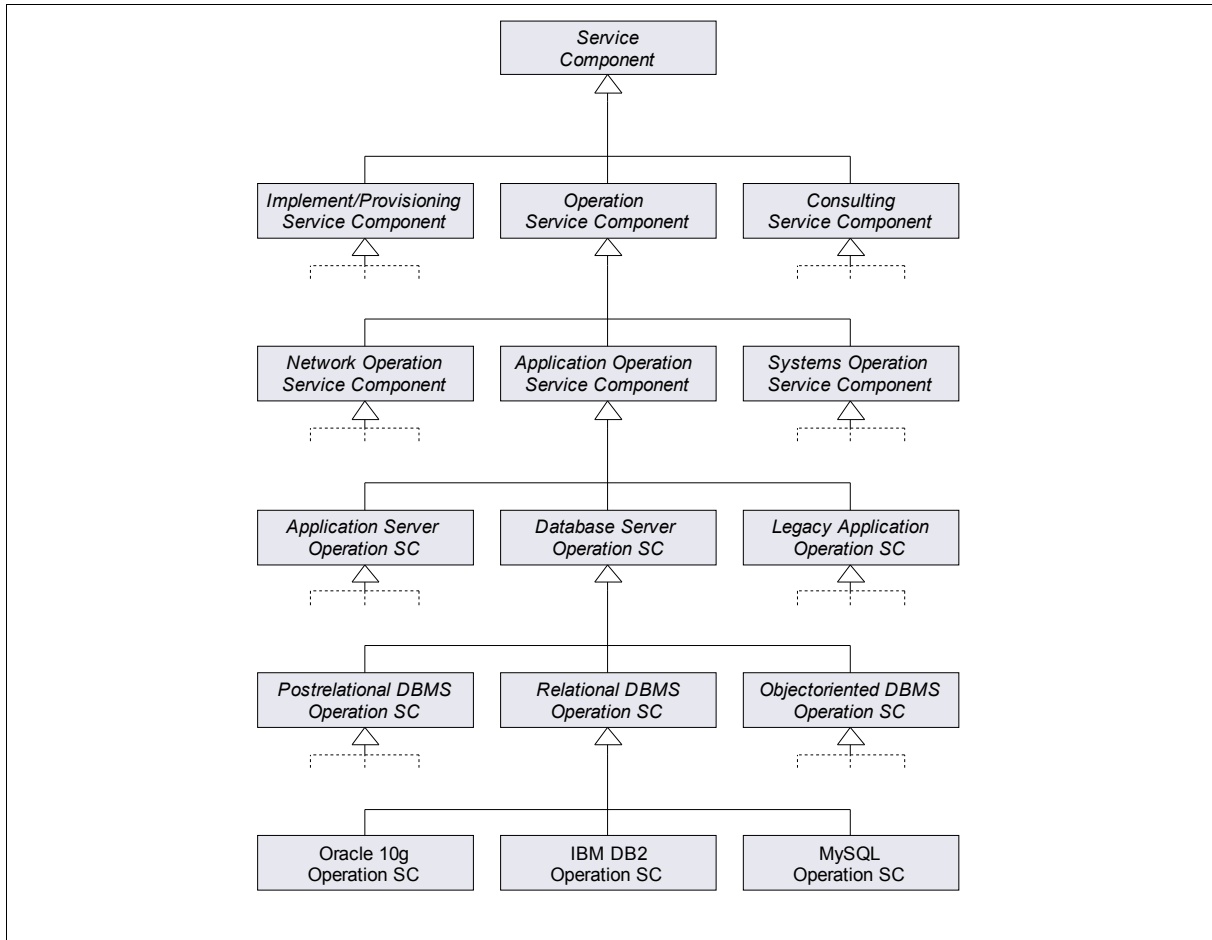


Abbildung 9: Service-Taxonomie.

Vererbungsmechanismen aus der Objektorientierung

Aus der objektorientierten Software-Entwicklung sind die Vererbungsmechanismen bekannt, die sich nun auch beim IT-Service-Engineering bewähren. Damit können Eigenschaften bei den abstrakten Service-Komponenten definiert werden und gelten dann durch die Vererbung für alle davon abgeleiteten Service-Komponenten. Zwei Beispiele dazu:

- Wir definieren ganz oben in der Taxonomie, dass es zwei Qualitätsvarianten geben soll: Standard und Premium. Durch Vererbung gilt dies für alle Service-Komponenten.
- Alle Betriebsdienstleistungen in der Standard-Variante haben Betriebszeiten zu Bürozeiten, die Premium-Varianten 24x7. Dies wird in der *Operation Service Component* definiert und von dort vererbt.

Diese Beispiele verdeutlichen, dass die Vererbung dazu führt, Service-Komponenten gewissen Standards und Vorgaben folgend zu spezifizieren. Wie eingangs festgestellt, ist das die Voraussetzung dafür, aufeinander abgestimmte, integrierbare Service-Komponenten – eben eine Service-Architektur – zu erhalten.

Ein weiterer Nutzen ist, dass die Entwicklung neuer Service-Komponenten wesentlich schneller geht, weil sie auf bereits bestehenden Oberklassen aufsetzen und lediglich in Details angepasst werden müssen.

Schlussfolgerung 4: Service-Taxonomie

Die Service-Komponenten können den Paradigma der Objektorientierung folgend in einer Klassenhierarchie organisiert werden. Eine solche Service-Taxonomie fördert die Standardisierung und vereinfacht die Modellierung durch die Vererbungsmechanismen. Zudem erschließt sie semantisches Wissen.

Note: Service Classification Recommendation (SCR)

Für die industrielle Herstellung dinglicher Produkte, etwa Maschinen, Autos oder Flugzeuge, gibt es zahlreiche Normen und Spezifikationen, die Komponenten dieser Produkte (von der Schraube bis zum Motor) definieren. Für IT-Dienstleistungen fehlen solche Standards gänzlich.

Um diese Lücke auszufüllen, entwickelt Advicio die *Service Classification Recommendation*, eine allgemeingültige Spezifikation von IT-Dienstleistungskomponenten. Dieser Standard wird zunächst als Empfehlung entwickelt. Eine Normierung wird dem folgen.

Die Service Classification Recommendation beinhaltet folgende Definitionen:

- Eine Taxonomie für IT-Dienstleistungskomponenten, d.h. die identifizierten Komponenten werden anhand ihrer Funktionalität klassifiziert und die Klassen werden anhand ihres Verwandtschaftsgrades geordnet.
- Die Spezifikation von Attributen für jede Klasse von Dienstleistungskomponenten. Die Attribute beschreiben die Komponente hinsichtlich der Funktionalität, Leistungsfähigkeit, Qualität und Preis.
- Die Definition von Voraussetzungen und Bedingungen der jeweiligen Komponente, sowie Abhängigkeiten von Ressourcen.
- Einen Produktlebenszyklus für jede Dienstleistungskomponente. Dies ist die Basisspezifikation für eine prozessorientierte Bearbeitung der Komponente.
- Ein Kundeninteraktionsmodell für jede Dienstleistungskomponente. Damit wird festgelegt, in welchem Schritt des Produktlebenszyklus der Leistungsempfänger eingebunden sein soll.

Die Service Classification Recommendation ist somit ein umfangreiches Produktmodell für standardisierte IT-Dienstleistungskomponenten. Grundsätze für die Entwicklung sind, eine Allgemeingültigkeit zu erzielen, sowie Offenheit für anwendungsspezifische Erweiterungen zu bewahren.

3 Paketbildung

Mit Service-Produkten geht der IT-Dienstleister auf den Kunden zu. An die Service-Produkte werden daher andere Anforderungen gestellt, als an die Komponenten:

- Während die Service-Komponenten möglichst granular sein sollten, sollen Service-Produkte häufig möglichst umfangreich gestaltet werden. So werden hochwertige Services zusammengestellt, die entsprechend hoch bepreist werden können. Zudem kann der IT-Dienstleister sich einer direkten Vergleichbarkeit mit Wettbewerbern entziehen, wenn er seine Service-Produkte umfangreicher und anders als der Wettbewerb gestaltet.
- Beim Komponenten-Katalog wurde in Kauf genommen, dass dieser sehr umfangreich wird. Der Produkt-Katalog muss aber sehr überschaubar sein, um den Kunden nicht mit der Auswahl zu überfordern. Statt einer hohen Anzahl von Service-Produkten sollten es nur wenige sein, die aber durch Konfiguration an die Anforderungen des Kunden angepasst werden können.

Service-Produkte sind lediglich Bündelungen von Service-Komponenten, die mit einem Preis und einer Bestellnummer versehen sind. Die eigentliche Leistung wird durch die Komponenten erbracht.

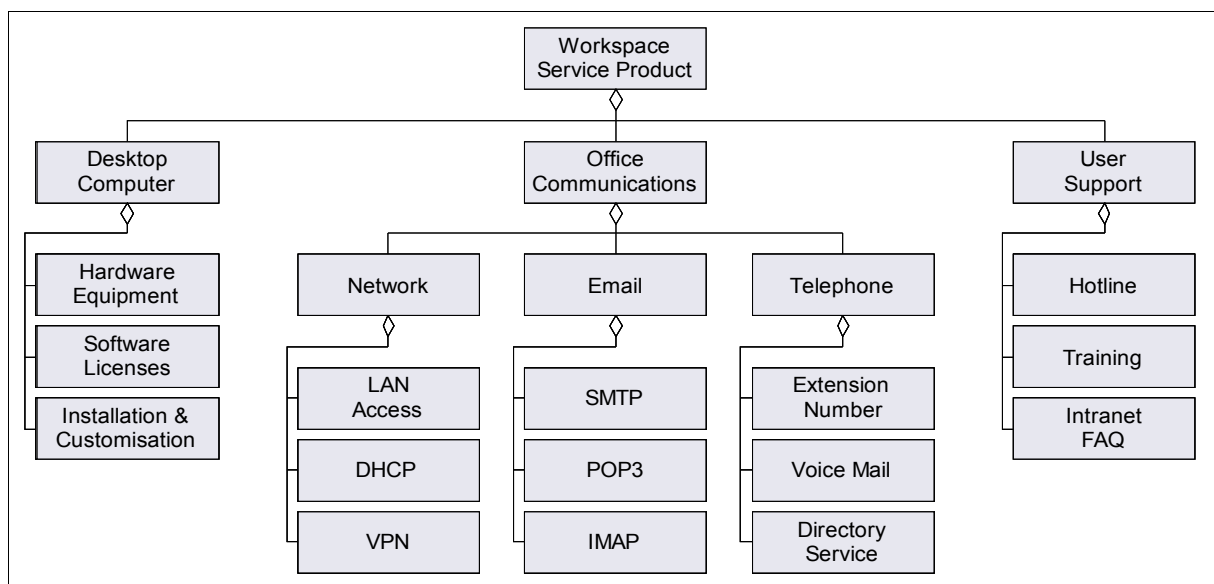


Abbildung 10: Komposition des Service-Produkts „Workspace“ aus Komponenten in der UML-Notation. Service-Komponenten können wiederum aus anderen Service-Komponenten aggregiert werden.

Produktkonfiguration

Natürlich muss das „Zusammenbauen“ von komplexen Dienstleistungen aus einzelnen Modulen bestimmten Regeln folgen. Diese Regeln werden auch als Bauvorschriften bezeichnet, und sie machen einen wesentlichen Teil der Spezifikation von Service-Komponenten aus. Den Vorgang des „Zusammenbauens“ entsprechend der Regeln bezeichnet man als Konfiguration. Das Service-Produkt wird für einen Kunden konfiguriert, indem Service-Komponenten so zusammengesetzt werden, dass die Komposition die Anforderungen des Kunden bestmöglich erfüllt.

Es gibt verschiedene Ansätze zur Lösung des Konfigurationsproblems. Für komponentenbasierte IT-Services bietet sich primär eine ressourcenbasierte Konfiguration an. Bei der ressourcenbasierten Konfiguration wird jede Komponente als eine Ressource angesehen,

welche einerseits ihre Leistung anderen zur Verfügung stellt, und andererseits selbst Bedarfe hat. Ein System gilt dann als vollständig konfiguriert, wenn alle Bedarfe durch Ressourcen abgedeckt sind.

Der Anwender benötigt ein Werkzeug für die Konfiguration der Dienstleistung. Wir schlagen dazu ein wissensbasiertes System vor, das den Anwender durch die Konfiguration führt. Wie bei Produktkonfiguratoren üblich, unterscheiden wir zwischen „high level configuration“, wobei die Konfiguration interaktiv vom Anwender vorgenommen wird, und „low level configuration“, was automatisch vom System oder vom Dienstleister konfiguriert wird.

Bauvorschriften

Bauvorschriften sollen den Freiheitsgrad beim Zusammenstellen von Komponenten einschränken, so dass nur sinnvolle Kombinationen möglich sind. Dazu werden Interfaces definiert. Eine erste Einschränkung geschieht durch die von einer Komponente erforderte Funktionalität. Die Funktionalität ist durch die Semantik aus der Service-Taxonomie zu entnehmen.

Eine weitere Einschränkung ergibt sich für die Konfiguration dadurch, dass es Komponenten gibt, die sich gegenseitig ausschließen, weil charakteristische Eigenschaften nicht zusammen passen. Dafür muss eine Konsistenzprüfung vorgenommen werden. Beispielsweise sollten die Betriebszeiten aller Teilleistungen übereinstimmen. Ein anderes Beispiel ist die Hochverfügbarkeit: Obwohl die Teilleistungen redundant ausgelegt sind, kann eine einzelne Teilleistung, die diesem Anspruch nicht genügt, die Verfügbarkeit der Gesamtleistung stören („single point of failure“). In einem solchen Fall kann die Komponente nicht dem Produkt zugeordnet werden. Derartige Bedingungen können im System durch Constraints, das sind Gleichungen oder Ungleichungen der Prädikatenlogik, hinterlegt werden.

Schlussfolgerung 5: Charakteristika von Service-Produkten

Service-Produkte werden an Endkunden verkauft und sind sehr umfangreiche, wertvolle Dienstleistungen. Es sollte nur wenige Service-Produkte geben, um den Kunden nicht mit der „Qual der Wahl“ zu überfordern. Service-Produkte sollten Varianten enthalten/zulassen, um die individuellen Bedürfnisse der Kunden erfüllen zu können. Der Dienstleister kann sich zudem durch die umfangreiche Gestaltung der Service-Produkte am Markt differenzieren.

4 Datenmodellierung und Wissensrepräsentation

4.1 Service Engineering und Configuration Management

Eine wesentliche Aufgabe vom Produktmanagement ist die umfassende Dokumentation der verantworteten Services. Dafür werden bei der Entwicklung neuer Dienstleistungen, beim sogenannten Service Engineering, jegliche Informationen zusammengetragen. Dies umfasst die Spezifikation der Dienstleistung aus Kundensicht, aber auch interne Prozessdefinitionen und Verfahrensanweisungen, Betriebshandbücher, sowie die erforderlichen Ressourcen, Kostenkalkulationen etc.. Dieses Wissen wird sinnvollerweise strukturiert erfasst, so dass es nicht nur von Menschen, sondern auch von Software verarbeitet werden kann.

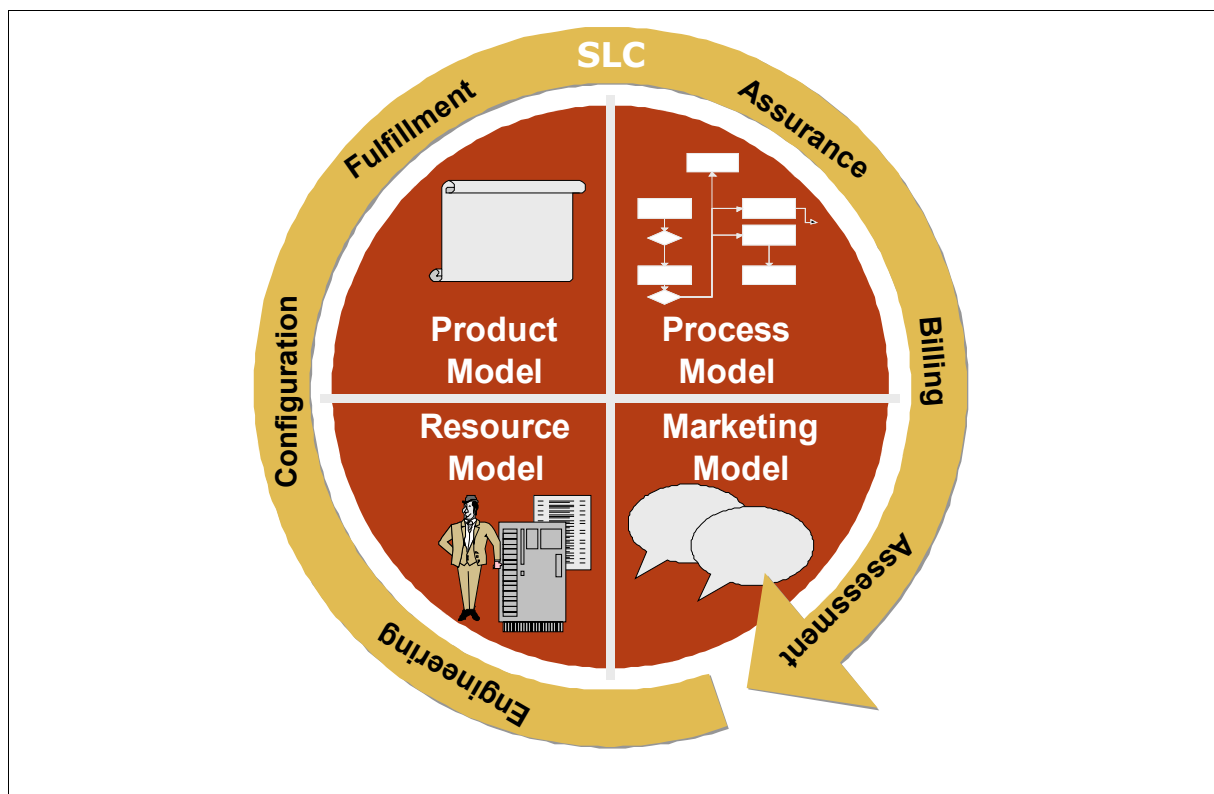


Abbildung 11: Betrachtete Dimensionen bei der Modellierung von IT-Service-Komponenten.

Von den Spezifikationen sind die Instanzen dieser Spezifikationen zu unterscheiden, die entstehen, wenn die Dienstleister zu einer Spezifikation einen Auftrag erhält:

Durch Service Engineering entwickelt der Dienstleister Leistungen, die er erbringen kann. Der Dienstleister definiert damit sein Potenzial. Wir sprechen in diesem Zusammenhang von den Potenzialdaten.

Beim Configuration Management werden dagegen „Assets“ verwaltet (Inventory, Kundenbeziehungen), die gegenwärtig bestehen und einen Wert darstellen. Es wird von Bestandsdaten gesprochen.

Eine Unterscheidung zwischen diesen sehr unterschiedlichen Geschäftsobjekten muß auch gegeben sein, wenn sie in einem gemeinsamen System verwaltet werden. Zudem gibt es auch noch die Bewegungsdaten.

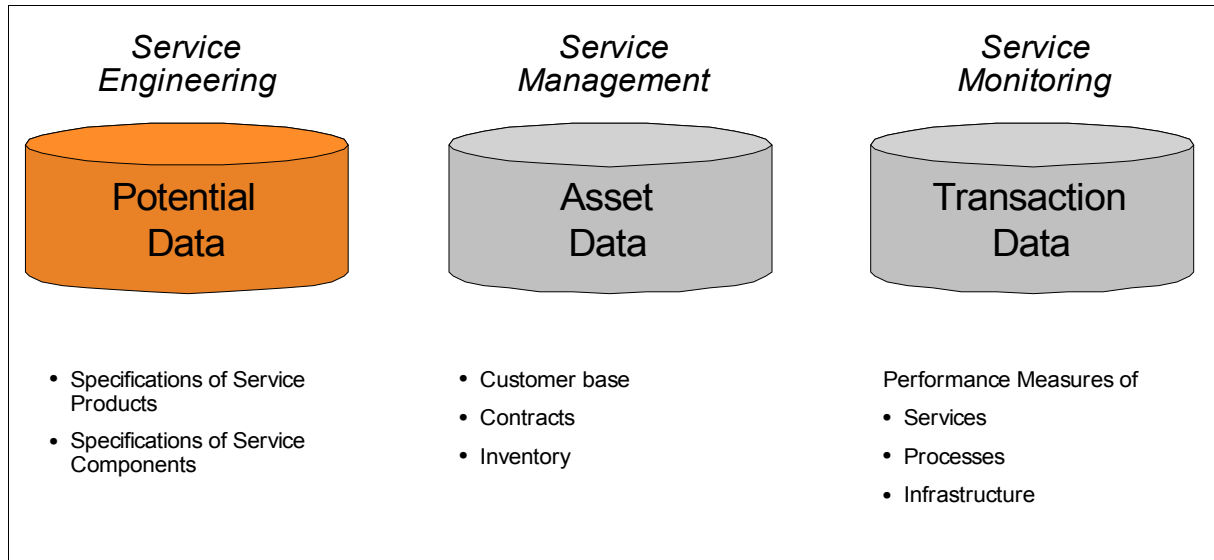


Abbildung 12: Bestandsdaten, beispielsweise Inventar und Verträge (SLAs), sowie Bewegungsdaten, etwa Trouble Tickets und Performance Reports, werden heute umfassend erhoben. Dagegen beginnen IT-Dienstleister erst jetzt, ihr Potenzial in Service-Katalogen zu spezifizieren.

Die vier Objekttypen bei IT-Services

Mit der Unterscheidung zwischen Service-Produkten und Service-Komponenten einerseits, und andererseits der Unterscheidung zwischen Spezifikationen und deren Instanzen, läßt sich eine zweidimensionale Matrix aufspannen, durch welche die vier Objekttypen definiert sind.

	Spezifikation	Instanz
Service-Produkt	Service Produkt Specification	Service Order
Service-Komponente	Service Component Specification	Service Instance

Abbildung 13: Matrix der vier Objekttypen.

Schlussfolgerung 6: Spezifikationen und Instanzen

Es muss jeweils zwischen Spezifikation und Instanz unterschieden werden. Die Instanz der Spezifikation eines Service-Produkts ist ein Auftrag (auch Service-Konfiguration genannt). Die Instanz der Spezifikation einer Service-Komponente ist eine Service-Instanz.

4.2 Wie die Informationen abgebildet werden können

Da die Erbringung von Dienstleistungen insbesondere im IT-Umfeld zunehmend automatisiert wird, reicht es nicht aus, dass IT-Services für den Menschen lesbar beschrieben werden. Auch Software-Systeme werden von uns in den Kreis der „Beteiligten“ eingeschlossen, sollen also die Beschreibungen der IT-Services interpretieren können. Von daher scheidet das klassische Werkzeug zur Dokumentation von Service-Katalogen der ersten Generation, nämlich Word und Excel, aus. Mit diesen Werkzeugen können zwar in einem strukturierten Dokument die charakteristischen Merkmale einer jeden Teilleistung erfasst werden, jedoch lassen sich Relationen zwischen ihnen nicht darstellen. Es müssen zwei-erlei Beziehungen berücksichtigt werden: (1) Die Klassenhierarchie, über die eine Service-Klasse Eigenschaften von ihren Oberklassen erbt. (2) Abhängigkeiten einer Teilleistung von Ressourcen, die dieser Teilleistung zuarbeiten.

Relationale Datenbanken sind primär auch ungeeignet, weil sie die Kollaboration unterschiedlicher Organisationseinheiten erschweren. Service-Spezifikationen erfordern keine zentrale Datenhaltung, sondern sollten zwischen Organisationen ausgetauscht werden können. Die Instanzen müssen natürlich schon zentral verwaltet werden, aber sie müssen sich auch mit Dritten, etwa Kunden, austauschen lassen.

XML bietet sich daher an, um auf XML basierend eine Beschreibungssprache für IT-Services zu erstellen. XML-Dateien können einzeln ausgetauscht werden, beispielsweise über HTTP, und können auch in Webbrowsern für den Menschen lesbar präsentiert werden.

4.3 Service Specification and Configuration Language

Advicio hat mit der Service Specification and Configuration Language (SSCL) eine XML-basierte, offene Beschreibungssprache entwickelt. Die SSCL besteht aus mehreren XML-Schemata. Darunter ist das Schema *Service Component Specification* (SCS) zur Beschreibung von Service-Komponenten der wohl wichtigste Teil der SSCL.

Für jede Service-Komponente wird eine eigene XML-Datei als Spezifikation erstellt. Darin sind unter anderem die nachfolgend beschriebenen Informationen abgelegt.

Name und URI: Es ist wichtig, dass die Spezifikation einer Service-Komponente eindeutig bezeichnet werden kann. Zusätzlich zu einem Namen, der nicht unbedingt eindeutig ist, erhält die Service-Komponente deshalb einen Unified Resource Identifier.

Beschreibung: Die Funktionalität der Service-Komponente wird verbal beschrieben. Weitere textuelle Anmerkungen, die nicht die Funktionalität beschreiben, können als Anmerkung ergänzt werden.

Semantik: Die Komponente erhält eine Semantik durch die Einordnung in eine Taxonomie. Diese kann dazu genutzt werden, dass auch Softwaresysteme die Funktionalität der Service-Komponente verstehen.

Classtype: Service-Komponenten können entweder abstrakt oder konkret sein. Die Bedeutung ist entsprechend der objektorientierten Programmierung.

Usage: Service-Komponenten haben einen oder mehrere Konsumenten. In der Spezifikation einer Service-Komponente wird durch das Element *usage* festgelegt, ob nur ein Konsument möglich ist, oder ob die Kapazität der Service-Komponenten sich von mehreren Konsumenten geteilt werden kann. Usage ist entweder *dedicated* oder *shared*.

Procedure: Service-Komponenten können hinsichtlich ihrer Ausführungszeiten klassifiziert werden. Es werden folgende Schlüsselbegriffe für das vererbte Element *procedure* unterschieden: *continuous*, *one-time*, *periodic*, *on-request*.

Parameter werden verwendet, um die Service-Komponente weiter zu spezifizieren. Es gibt vier Einsatzgebiete für Parameter:

- Qualitätsmerkmale: Parameter können Qualitätsmerkmale sein, wie zum Beispiel der Verfügbarkeit einer Service-Komponente.
- Kapazitätsmerkmale: Parameter können die Kapazität der Service-Komponente festlegen, etwa die Bandbreite eine Standleitung.
- Konfigurationsparameter: Parameter legen fest, welche Angaben zur Konfiguration der Service-Komponenten für die Instantiierung notwendig sind.
- Kostenmerkmale: Parameter legen fest, wie sich die Kosten einer Service-Komponente zusammensetzen bzw. berechnen lassen.

Varianten: Service-Komponenten können in verschiedenen Varianten spezifiziert werden. Varianten haben die selbe Funktionalität und das selbe Prozessmodell, und auch die eingesetzten Ressourcen sind die gleichen. Lediglich hinsichtlich der Qualität, der Kapazität und der Konfigurierbarkeit können sich Varianten unterscheiden.

Bauvorschriften: Die SSCL definiert die Leistung einer Service-Komponente durch das Produktmodell. Welche anderen Service-Komponenten benötigt werden, wird durch interface Elemente spezifiziert. Jedes Interface kann man sich als einen Steckplatz vorstellen, der mit einer anderen Service-Komponente verbunden werden muss, wenn die Dienstleistung implementiert wird.

Lebenszyklus und Prozessmodell: siehe Kapitel 5.3

Versionierung: siehe Kapitel 5.2

Schlussfolgerung 7: Inhalte der Spezifikationen

Service-Produkte und Service-Komponenten werden umfangreich in Spezifikationen beschrieben. Sie enthalten:

- Produktmodell: Spezifikation der Funktionalität, Qualität, Kosten etc.
- Prozessmodell: Lebenszyklus, Zustände, Workflows (als Basis für Automatisierung)
- Ressourcenmodell: Benötigte Service-Komponenten (Bauvorschrift), Personal (Rollen), Infrastruktur.
- Marketing: Kundeninteraktionsmodell, Präsentation

5 Service Life Cycle

5.1 Abbildung von Prozessen und Zuständen

Die Automatisierung von Routinearbeiten ist nicht nur eine Entlastung, sondern beschleunigt auch viele Prozesse in der Service-Provider-Organisation und eliminiert Fehlerquellen. Prozessorientierung ist auch deshalb wichtig, da die Dienstleistungen Lebenszyklen besitzen und durch Prozesse erbracht werden. Die Prozesse im System zu berücksichtigen ist unabdingbar, aber zugleich eine große Herausforderung, weil sie Dynamik in das System bringen: Konfigurationen ändern sich, neue Bestellungen müssen ausgeführt werden, andere Aufträge laufen aus. Dienstleistungen werden neu entwickelt oder eingestellt. Beteiligte Personen haben Rollen, die aber nicht statisch sind, sondern vom behandelten Objekt und dessen Zustand abhängen.

Der Zusammenhang zwischen den zuvor definierten Objekten und den Prozessen wird durch einen endlichen Zustandsautomaten (finite state machine) abgebildet. Die Prozesse sind dabei die Übergänge von einem Zustand zu dem nächsten. Oder in anderen Worten: Der Prozess versetzt das Objekt in einen anderen Zustand.

5.2 Lebenszyklus der Spezifikationen

Der erste Teil des Lebenszyklus von IT-Services betrifft die Spezifikationen von Service-Produkten und Service-Komponenten. Sie haben einen Zustand, der ausdrückt, in welcher Phase des Lebenszyklus sich eine SPS oder SCS befindet. Implizit wird damit festgelegt, wozu das Objekt zu verwenden ist. Folgende vier Stati wurden von Advicio definiert:

DV	Development	SCS befindet sich in Entwicklung.	Die Service-Komponente darf noch nicht in Aufträgen verwendet werden.
AC	Active	SCS ist Bestandteil des Service-Portfolios.	Die Service-Komponente kann verkauft werden.
PO	Phase Out	SCS ist abgekündigt.	Service-Komponenten dürfen nicht mehr neu verkauft werden, aber in bestehenden Verträgen werden sie noch eingesetzt.
EL	End of Life	SCS ist am Ende des Lebenszyklus.	Service-Komponenten werden nicht mehr verkauft und sind nicht mehr in Betrieb.

Version

Die Spezifikationen erhalten eine Versionsnummer, die aus drei durch Punkte getrennte Dezimalzahlen besteht. Beispiel: 1.2.183. Diese drei Zahlen bezeichnen die Hauptversion, Subversion und Build-Nummer.

Best Practice: Umgang mit Versionsnummern

Bei jedem Speichern wird die Versionsnummer erhöht. Wenn der Benutzer der ASW nicht explizit eine höhere Version festlegt, erhöht ASW die letzte Dezimalzahl (Build-Nummer) um eins. Für stabile Versionen sollten gerade Zahlen als Subversionsnummer verwendet werden. Ungerade Subversionsnummer lassen entsprechend auf eine weniger stabile Arbeitsversion schließen.

Zusammenhang von Zustand und Version

Die Angabe des Status bezieht sich immer auf eine Version. Es können also mehrere Versionen einer Spezifikation parallel bestehen, die in unterschiedlichen Zuständen sind. Ein Beispiel dazu:

- Version 1.2.183 ist EL. Diese Version der SCS wird nicht mehr verwendet. Die SCS wird nur noch zu Dokumentationszwecken oder aus regulatorischen Gründen mitgeführt. Nach einiger Zeit wird sie wahrscheinlich gelöscht.
- Version 1.6.5 ist PO. Service-Komponenten dieser Version werden nicht mehr neu verkauft, aber sie wird noch in bestehenden Verträgen eingesetzt. Solange diese Verträge laufen wird auch die Spezifikation noch benötigt und behält diesen Status.
- Version 1.8.0 ist PO. Auch zu dieser Version bestehen noch laufende Verträge.
- Version 2.0.1 ist AC. Dies ist die aktuelle Version, die in neuen Verträgen eingesetzt wird.
- Version 2.1.16 ist DV. Dies ist die nächste Version der Komponente, an der die Service Engineers gerade arbeiten.

Zustandswechsel

Zustandswechsel können immer nur „vorwärts“ gemacht werden, also von DV nach AC, von AC nach PO, von PO nach EL.

Soll ein Zustandswechsel rückwärts gemacht werden, so entsteht formal eine neue, zusätzliche SCS, welche die Historie von der Vorgänger-SCS erhält. Man sagt, es entsteht ein 'branch' von der SCS. Nachfolgendes Bild stellt dies dar.

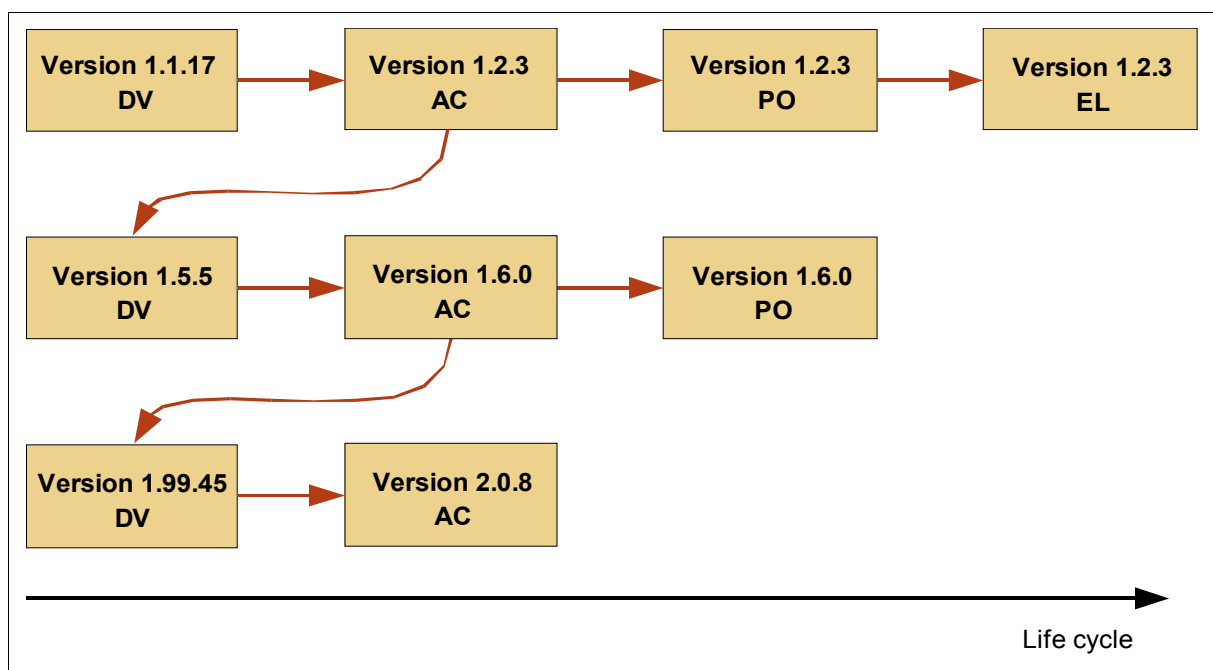


Abbildung 14: Entstehung neuer „branches“. Nur durch eine Abzweigung, bei der die bestehende SCS kopiert wird, kann im Lebenszyklus der SCS scheinbar zurück gegangen werden. Der ältere Zweig bleibt aber bestehen und muss seinen Weg bis „EL“ (End of Life) weiter gehen.

Change Log

Die Spezifikationen protokollieren Veränderungen mit. Zu jedem Change werden die Version, der Status, das Datum, der Benutzer und gegebenenfalls eine vom Benutzer eingepflegte Dokumentation der Änderung protokolliert.

5.3 Zustandsmodell der Instanzen

Service-Instanzen haben sowohl Lebenszyklus-Zustände als auch operative Zustände. Beide sind in dem Zustandsmodell zu berücksichtigen. Es sollte versucht werden, der Einfachheit halber, für alle Service-Komponenten ein einheitliches Zustandsmodell zu definieren. Advicio schlägt dafür das in der folgenden Abbildung gezeigte Modell vor.

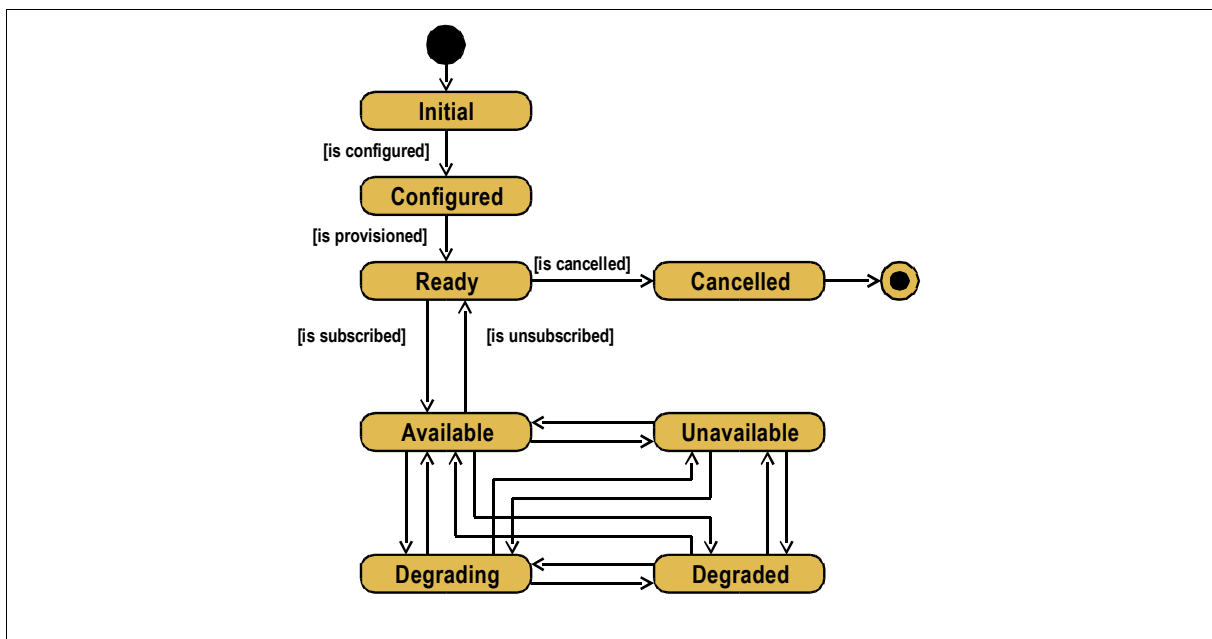


Abbildung 15: Das Zustandsmodell von Service-Instanzen.

Die Prozesse sind die Zustandsübergänge. Natürlich sind die Prozesse speziell für jede Service-Komponente. Schließlich unterscheidet sich beispielsweise der Vorgang zur Einrichtung eines Email-Accounts von der Einrichtung einer WAN-Anbindung einer Filiale.

Die Prozesse selbst können durch BPM- oder Workflow-Systeme implementiert werden. In den Spezifikationen der Services kann dann diese Prozessmodellierung referenziert werden.

6 Der Service-Katalog in Anwendungen

6.1 Eine Anwendungsarchitektur für IT-Dienstleister

Die Anwendungen, mit denen IT-Dienstleister selbst arbeiten, müssen drei Bereiche umfassen: Engineering, Management und Operations. Diese entsprechen den bereits zuvor beschriebenen unterschiedlichen Datentypen, den Potenzialdaten, den Bestandsdaten und den Bewegungsdaten.

Auch wenn die vollständige Datenintegration manchmal gewünscht ist, lässt die betriebliche Praxis dies meist nicht zu. Praktikabler ist es, wie in der Abbildung dargestellt, die Daten getrennt voneinander, meist auch in unterschiedlichen Anwendungssystemen, vorzuhalten.

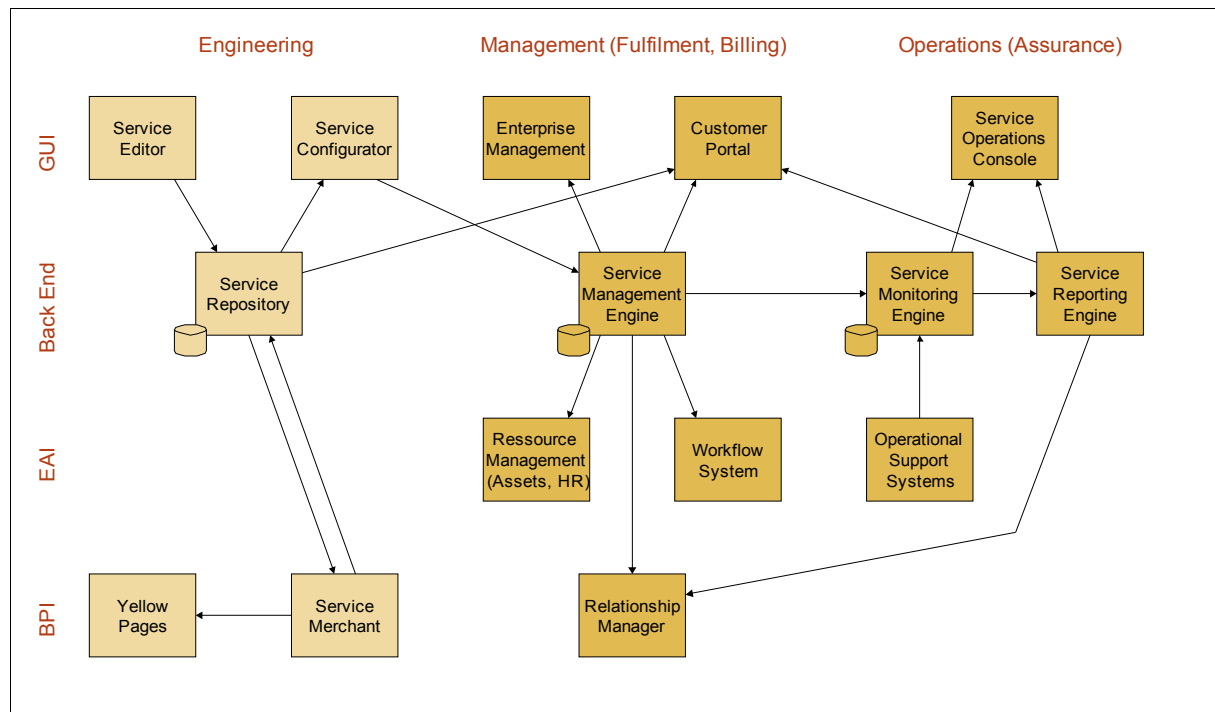


Abbildung 16: Referenz-Anwendungsarchitektur für IT-Dienstleister.

Die Anwendungen im rechten Bereich umfassen operative Systems wie beispielsweise Performance Management und Incident Management. Der mittlere Bereich enthält die kaufmännischen Anwendungen wie ERP-Systeme um Kundendaten zu verwalten oder eine CMDB für IT Assests. In Kapitel 6.2 gehen wir näher auf eine Problematik mit der CMDB ein.

Noch wenig im Einsatz sind Werkzeuge zur Entwicklung von IT-Services, die im linken Bereich dargestellt sind. Zu diesem Bereich zählt die Advicio Service Workbench, die in Kapitel 6.3 vorgestellt wird.

Horizontal ist die Referenzarchitektur in vier Schichten aufgeteilt. Oben, die als „GUI“ bezeichnete Schicht, enthält die Anwendungen, mit denen Mitarbeiter aber auch Kunden interaktiv arbeiten. In den Schichten 2 und 3 sind die Systeme zur Datenhaltung („Back End“) und andere zu integrierende Anwendungen („EAI“). Die unterste Schicht heisst „Business Partner Integration“ (BPI) und ermöglicht den Informationsaustausch in nahezu Echtzeit mit Partnern. Dies ermöglicht es, Wertschöpfungsnetze zu bilden und das eingangs vorgestellte Geschäftsmodell des Service-Integrators umzusetzen.

6.2 CMDB

Eine Art Wissensmanagement-System, welches bei IT-Dienstleistern schon vergleichsweise oft eingesetzt wird, ist die sogenannte Configuration Management Database (CMDB). Die von ITIL definierte Managementdisziplin des Konfigurationsmanagements sieht vor, dass mit einer CMDB die Informationen über alle Configuration Items (CI) verwaltet werden. Was zu den Configuration Items zählt, ist von ITIL nicht definiert, aber typischerweise zählt das Inventar dazu. Auch die bestehenden Verträge mit Kunden können als CI verstanden und mit der CMDB verwaltet werden.

Es liegt die Idee nahe, mit der CMDB auch die Spezifikationen der Dienstleistungen zu verwalten. Grundsätzlich ist dagegen nichts einzuwenden, jedoch liegt damit ein Fehler nahe:

Durch Service Engineering entwickelt der Dienstleister Leistungen, die er erbringen kann. Der Dienstleister definiert damit sein Potenzial. Wir sprechen in diesem Zusammenhang von den Potenzialdaten.

Beim Configuration Management werden dagegen „Assets“ verwaltet (Inventar, Kundenbeziehungen), die gegenwärtig bestehen und einen Wert darstellen. Es wird von Bestandsdaten gesprochen.

Eine Unterscheidung zwischen diesen sehr unterschiedlichen Geschäftsobjekten muß auch gegeben sein, wenn sie in einem gemeinsamen System verwaltet werden. Die SSCL kennt zudem auch die Betriebsdaten.

6.3 Advicio Service Workbench

Speziell für das Service Engineering, also die Entwicklung der Spezifikationen, hat Advicio ein eigenes Werkzeug als Referenzimplementation entwickelt. Dieses Werkzeug heisst Advicio Service Workbench (ASW).

Die Advicio Service Workbench wurde als Plug-in für Eclipse (www.eclipse.org) entwickelt. Eclipse ist eine plattformunabhängige Integrierte Entwicklungsumgebung (IDE: Integrated Development Environment). Solche Entwicklungsumgebungen sind aus dem Bereich der Softwareentwicklung bekannt. Tatsächlich wird Eclipse sehr stark bei der Softwareentwicklung eingesetzt. Da Eclipse aber eine offene Architektur hat, um die Entwicklungsumgebung für verschiedene Programmiersprachen einsetzen zu können, ist die Entwicklungsumgebung nicht auf Software-Entwicklungsprojekte beschränkt.

Eclipse wurde ursprünglich von der IBM entwickelt, ist jedoch seit 2001 als Open-Source-Anwendung in den Händen der Eclipse Foundation.

Advicio hat mit der ASW ein Plug-in entwickelt, mit dem Projekte in der „Programmiersprache“ SSCL durchgeführt werden können. Der Anwender, der neue Services entwickelt, muss jedoch nicht programmieren, sondern benutzt moderne Eingabemasken und eine benutzerfreundliche Menüführung. Die Komplexität der XML-Strukturen, die Mechanismen der Vererbung in der Taxonomie und nicht zuletzt die Sorge um die Konsistenzhaltung der Daten bleibt damit dem Anwender verborgen.

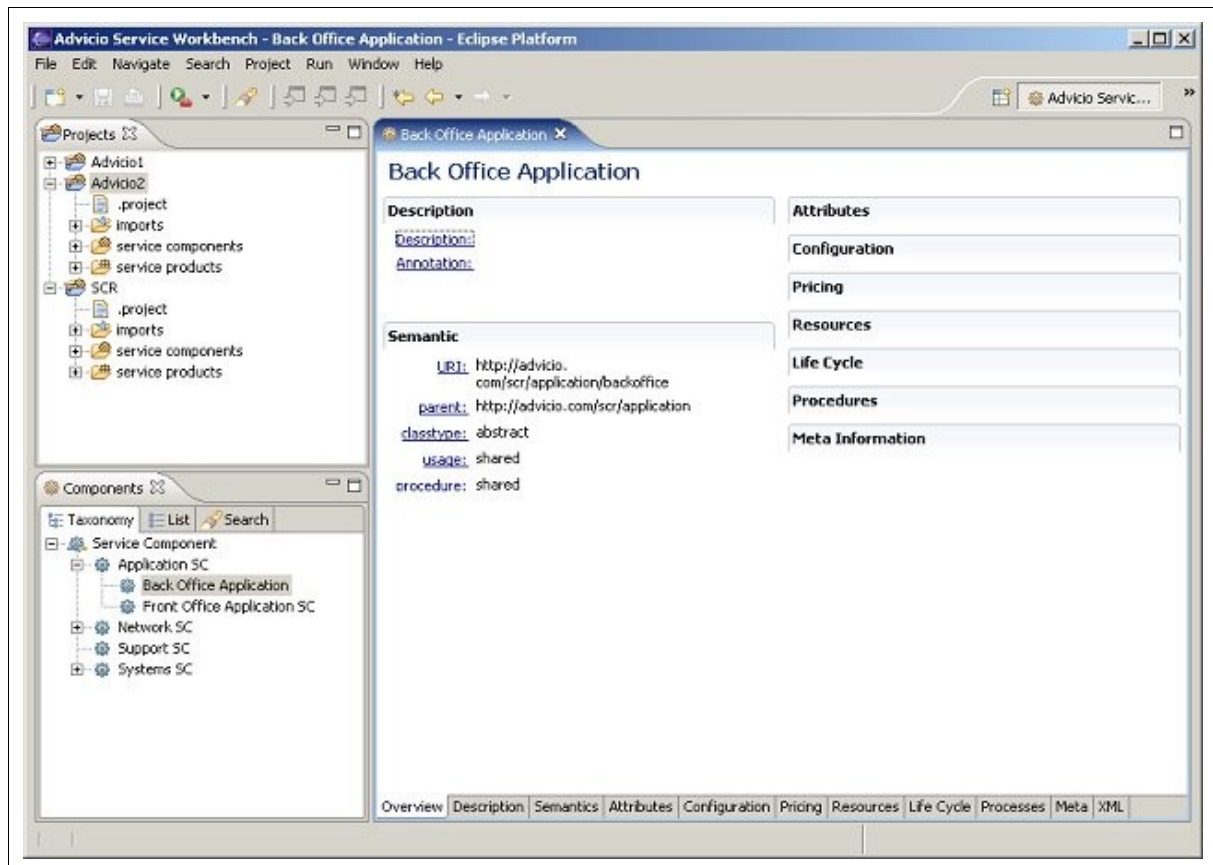


Abbildung 17: Advicio Service Workbench.

7 Zusammenfassung

Die Kernaussagen dieses Workshops lassen sich wie folgt zusammenfassen:

Produktdatenmanagement

1. Ein Service-Katalog der 2. Generation beschreibt nicht nur den Service aus Kundensicht, sondern auch die interne Sicht des Dienstleisters. D.h. jegliches Wissen zu dem Service wird darin zusammengeführt. Der Service-Katalog ist das Produkt-Daten-Management des IT-Dienstleisters.

Produkte und Komponenten

2. Eine komponentenbasierte Service-Architektur erfordert die Unterscheidung zwischen Service-Produkten und Service-Komponenten. Entsprechend gibt es einen Produktkatalog und einen Komponentenkatalog.

Charakteristiken von Service-Komponenten

3. Service-Komponenten sollten einen sehr begrenzten Umfang haben, um sie "greifbar" zu machen. Service-Komponenten können über mehrere Stufen dekomponiert (zerlegt) bzw. komponiert (zusammengebaut) werden. Eine höhere Granularität erhöht die Wahrscheinlichkeit der Wiederverwendung. Typisch ist, dass IT-Dienstleister mehrere hundert Service-Komponenten haben.

Service-Taxonomie

4. Die Service-Komponenten können den Paradigma der Objektorientierung folgend in einer Klassenhierarchie organisiert werden. Eine solche Service-Taxonomie fördert die Standardisierung und vereinfacht die Modellierung durch die Vererbungsmechanismen. Zudem erschließt sie semantisches Wissen.

Charakteristiken von Service-Produkten

5. Service-Produkte werden an Endkunden verkauft und sind sehr umfangreiche, wertvolle Dienstleistungen. Es sollte nur wenige Service-Produkte geben, um den Kunden nicht mit der „Qual der Wahl“ zu überfordern. Service-Produkte sollten Varianten enthalten/zulassen, um die individuellen Bedürfnisse der Kunden erfüllen zu können. Der Dienstleister kann sich zudem durch die umfangreiche Gestaltung der Service-Produkte am Markt differenzieren.

Spezifikationen und Instanzen

6. Es muss jeweils zwischen Spezifikation und Instanz unterschieden werden. Die Instanz der Spezifikation eines Service-Produkts ist ein Auftrag (auch Service-Konfiguration genannt). Die Instanz der Spezifikation einer Service-Komponente ist eine Service-Instanz.

Inhalte der Spezifikationen

7. Service-Produkte und Service-Komponenten werden umfangreich in Spezifikationen beschrieben. Sie enthalten:
 - Produktmodell: Spezifikation der Funktionalität, Qualität, Kosten etc.
 - Prozessmodell: Lebenszyklus, Zustände, Workflows (als Basis für Automatisierung)
 - Ressourcenmodell: Benötigte Service-Komponenten (Bauvorschrift), Personal (Rollen), Infrastruktur.
 - Marketing: Kundeninteraktionsmodell, Präsentation



As a company, **Advicio**[®] is a competence centre, that develops business models for IT service providers. Our recommendations, methods, concepts and tools are applied, with partners, at IT service providers.

Vision

Advicio helps IT service providers to improve their competitive advantage with knowledge-based approaches. The company provides research and development of methodologies, concepts, and recommendations that allow IT service providers to better understand, engineer, document and communicate their business model, in particular their service product portfolio, processes, resources and relationships. By following Advicio's suggestions, IT service providers experience enhanced customer satisfaction, cost reduction, and a greater agility in a dynamic market.

Mission

Advicio continuously develops its domain knowledge in the IT service provider industry. The company maintains its relationships to both industry and scientific communities. Focusing on their domain, vendor independence, and openness are the company's principles.

advicio.de