

Das Kommentierte SOA-Manifest

Kommentare und Einsichten ueber das SOA-Manifest von Thomas Erl

Service-Orientierung ist ein Paradigma, das den Rahmen fuer unser Handeln vorgibt, Service-orientierte Architektur (SOA) ist ein Architekturtyp, der aus der Anwendung von Service-Orientierung entsteht.

Von Anfang an war es selbstverstaendlich, dass dies ein Manifest ueber zwei verschiedene, aber nahe verwandte Themen ist: das Service-orientierte Architektur-Modell und die Service-Orientierung, die durch das Paradigma der Architektur definiert werden sollten. Das Format dieses Manifestes wurde nach dem "Agile Manifesto" modelliert, welches Inhalte auf Aussagen beschraenkt, dass die Ambitionen, die Werte auszudruecken praegnant modellierten und Leitprinzipien fuer die Realisierung dieser Ziele und Werte. Ein solches Manifest ist keine Spezifikation, ein Referenzmodell oder sogar ein weisses Papier, und ohne eine Option eigentliche Definitionen zu geben, haben wir beschlossen, diese Praeambel hinzuzufuegen, um klarzustellen, wie and warum diese Begriffe in den anderen Teilen des Manifest-Dokumentes referenziert werden.

Wir haben Service-Orientierung angewendet ..

Das Service-Orientierungs Paradigma ist am besten als eine Methode oder ein Ansatz zur Realisierung einer bestimmten Zielsetzung, die zudem durch eine Reihe von strategischen Handlungen und Leistungen definiert ist, anzusehen. Wenn wir Service-Orientierung anwenden, gestalten wir Software-Programme und Technologie-Architektur zur Unterstuetzung der Realisierung des Zieles. Das ist, was die Technologie-Architektur als Service orientiert qualifiziert.

..um Organisationen zu helfen, kontinuierlich nachhaltigen Geschaefstwert zu liefern mit hoeherer Agilitaet und Kosteneffizienz..

Die Fortsetzung der Praeambel hebt einige der bekanntesten und am haeufigsten erwarteten strategischen Vorteile der Service-Oriented Computing heraus. Das Verstehen dieser Vorteile hilft etwas Licht in die genannte Zielsetzung zu bringen und als Ergebnis der Anwendung von Service-Orientierung zu realisieren. Die Beweglichkeit auf kommerzieller Ebene ist vergleichbar mit der Reaktionsfaehigkeit einer Organisation. Je einfacher und effektiver eine Organisation auf geschaefftliche Veraenderungen reagieren kann, desto effizienter und erfolgreicher wird es zur Anpassung an die Auswirkungen der Veraenderung kommen (und weiter nutzen, was die Vorteile der Veraenderung herbeifuehren koennen). Service orientierte Positionen Dienstleistungen als IT-Werte, die voraussichtlich wiederholte Werte im

Laufe der Zeit bieten, die weit ueber die anfaengliche Investition fuer die Lieferung erforderlich sind. Die Auswirkung der Kosten bezieht sich in erster Linie auf die zuerwartenden Gewinne der Anlagen. In vielerlei Hinsicht geht eine Zunahme der Kosten-Wirksamkeit Hand-in-Hand mit einer Steigerung von Beweglichkeit, wenn es mehr Gelegenheit zur Wiederverwendung bestehender Dienste gibt, dann ist in der Regel weniger Aufwand benoetigt, um neue Loesungen zu bauen. "Dauerhafte" Geschaeftswerte beziehen sich auf die langfristigen Ziele der Service-Orientierung um Software-Programme zu gruenden, die als Dienste mit der inhaerenten Flexibilitaet staendig neue Loesungs-Konfigurationen zu schaffen und sich staendig aendernden Geschaeftsanforderungen anzupassen.

..im Einklang mit den sich aendernden fachlichen Beduerfnissen

Diese letzten Worte der Praeambel sind der Schluessel zum Verstaendnis der zugrunde liegenden Philosophie der Service-Oriented Computing. Die Notwendigkeit geschaeftliche Veraenderungen auf einer kontinuierlichen Basis unterzubringen, ist grundlegend zur Service-Orientierung und als ein grundlegendes, uebergeordnetes, strategisches Ziel.

Im Rahmen unserer Arbeit sind wir zu folgender Priorisierung gekommen:

Die nachfolgenden Aussagen bilden die Kernwerte von denen jeder Ausdruck einer Priorisierung ueber etwas ist, dass auch als Wert angesehen wird. Die Absicht dieses Wertsystems ist mit den harten Entscheidungen, die auf einer regelmaessigen Basis gemacht werden muessen, die strategischen Ziele und Vorteile der Service-Oriented Computing konsequent zu realisieren.

Geschaeftswert ueber technische Strategie

Wie bereits erwaeht, ist die Notwendigkeit geschaeftliche Veraenderungen unterzubringen, ein uebergreifendes strategisches Ziel. Daher wird die grundlegende Qualitaet der Service orientierten Architektur und alle Softwareprogramme, Loesungen und Oeko-Systeme, die durch die Annahme der Service-Orientierung vom Geschaeft angetrieben werden. Die Technik bestimmt nicht die Richtung, es ist die Geschaeftsvision die die Nutzung der Technologie diktiert. Diese Prioritaet kann einen tiefen Welleneffekt innerhalb der Regionen eines IT-Unternehmens hervor rufen. Er fuehrt Aenderungen zu fast allen Teilen des IT Delivery Lifecycles aus, von Automatisierungsloesungen, planen und finanzieren, bis zu wie wir bauen und regieren. Alle anderen Werte und Prinzipien in dem Manifest unterstuetzen die Verwirklichung dieses Wertes in gewisser Weise.

Strategische Ziele ueber projektspezifischen Nutzen

Historisch gesehen konzentrieren sich viele IT-Projekte ausschliesslich auf den Aufbau von Anwendungen, die speziell fuer Business Process Anforderungen entwickelt wurden, die damals aktuell waren. Diese erfuellten die sofortige (taktische) Notwendigkeit, aber als mehr von diesen Einzweck-Anwendungen geliefert wurden, fuehrten sie in ein IT-Unternehmen, dass gefuellt mit Inseln von Logik und Daten war, und bezeichneten die Anwendung als "Silos". Als neue geschaeftliche Anforderungen auftauchten, entstanden entweder neue Silos, oder

es wurden Integrations-Kanaele zwischen Silos errichtet. Als noch mehr Geschaeftsveraenderungen entstanden, mussten die Integrations-Kanaele erweitert werden, noch mehr Silos geschaffen werden, und schon bald wurde die IT-Unternehmungs-landschaft zunehmend verdreht und beschwerlich, teuer und langsam in der Entwicklung. In vieler Hinsicht entstand Service-Orientierung als Reaktion auf diese Probleme. Es ist ein Paradigma, das eine Alternative zum Projekt-spezifischen, Silo-basierten und integrierter Anwendungsentwicklung durch stricte Priorisierung der Erreichung der langfristigen, strategischen Unternehmensziele zu realisieren. Das Ziel der Service-Orientierung befuerwortet keine traditionellen Anwendungs-Silos. Und selbst wenn Legacy-Ressourcen und Anwendungs-Silos in Umgebungen existieren, in denen Silos Service-Orientierung angenommen wurden, ist das Ziel, dass sie in dem Umfang harmonisiert sind, der durchfuehrbar ist.

Immanente Interoperabilitaet ueber massgeschneiderte Integration

Damit Software-Programme Datenaustausch vornehmen koennen, muessen sie interoperabel sein. Wenn Software-Programme nicht kompatibel ausgelegt sind, werden sie wahrscheinlich nicht interoperabel sein. Um die Interoperabilitaet zwischen inkompatiblen Software-Programmen zu ermoeeglichen, ist es erforderlich, dass sie integriert werden. Integration ist also der Aufwand, um die Interoperabilitaet zwischen verschiedenen Software-Programmen zu erreichen. Obwohl oft notwendig, kann kundenspezifische Integration teuer und zeitaufwendig sein, und kann zu schwachen Architekturen fuehren, die schwer zu entwickeln sind. Eines der Ziele der Service-Orientierung ist die Notwendigkeit fuer kundenspezifische Integration durch Gestaltung von Software-Programmen (innerhalb einer bestimmten Domaene zu minimieren), so dass sie nativ kompatibel sind. Diese Qualitaet wird als intrinsische Interoperabilitaet bezeichnet. Das service-orientierte Paradigma umfasst eine Reihe von spezifischen Design-Prinzipien, die in Richtung Aufbau einer intrinsischen Interoperabilitaet auf mehreren Ebenen ausgerichtet sind. Intrinsic Interoperabilitaet als ein Merkmal von Software-Programmen, das sich innerhalb einer bestimmten Domaene befindet, ist der Schluessel zur Realisierung strategischer Vorteile, wie z.B. erhoelte Kosten-Effektivitaet und Agilitaet.

Gemeinsam verwendete Service ueber zweckgebundene Implementierungen

Wie soeben ausgefuehrt, gruendet Service-Orientierung ein Design Konzept, bestehend aus einer Reihe von Design Prinzipien. Wenn in einem sinnvollen Umfang angewandt, formen diese Grundsaeetze ein Software-Programm zu einer Einheit von Service-orientierter Logik, die rechtsmaessig als Service bezeichnet werden kann. Services sind mit konkreten Merkmalen (z.B. diejenigen, die intrinsische Interoperabilitaet ermoeeglichen) ausgestattet, die direkte Unterstuetzung des zuvor beschriebenen Zieles. Eines dieser Merkmale ist die Verkapselung von Mehrzweck-Logik, die geteilt und wiederverwendet werden kann zur Unterstuetzung der Automatisierung verschiedener Geschaeftsprozesse. Ein Shared Service etabliert sich als ein IT-Wert, der wiederholten geschaeftlichen Nutzen bei gleichzeitiger Verringerung der Kosten und Aufwand bieten kann, um neue Automatisierungs- Loesungen zu liefern. Zwar gibt es traditionellen Wert in den Single-Purpose-Anwendungen, die taktischen geschaeftlichen Anforderungen loesen. Die Verwendung von Shared Services bieten mehr Wert bei der

Realisierung der strategischen Ziele des Service-Oriented Computing (die wiederum auch eine Erhoehung der Wirtschaftlichkeit und Agilitaet einschliessen)

Flexibilitaet ueber Optimierung

Dies sind vielleicht die allgemeinen Wert Priorisierungs-Aussagen und sie werden am besten als Leitprinzip fuer die Philosophie angesehen, wie man bessere Prioritaeten verschiedener Erwaegungen bei der Auslieferung und sich entwickelnden individuellen Serviceleistungen und Vorraete von Dienstleistungen liefert. Optimierung bezieht sich in erster Linie auf die Erfuellung der taktischen Gewinne durch Einstellen eines bestimmten Anwendungs-Designs oder Beschleunigung der Belieferung von unmittelbarem Bedarf. Da gibt es nichts Unerwueschtes ausser, dass es zu den genannten Silo-basierten Umgebungen fuehren kann, wenn sie nicht richtig in Bezug auf die Foederung priorisiert sind, um Flexibilitaet zu foedern. Zum Beispiel gehen die Eigenschaften der Flexibilitaet jenseits der Faehigkeiten fuer Dienstleistungen um effektiv (und eigensicher) Daten auszutauschen. Um wirklich auf die sich staendig aendernden geschaeftlichen Anforderungen reagieren zu koennen, muessen Leistungen auch flexibel sein, wie sie in Composite-Loesungen kombiniert und aggregiert werden. Im Gegensatz zu traditionellen verteilten Anwendungen, die oft relativ statisch waren trotz der Tatsache, dass sie componentisiert wurden, muessen Service-Kompositionen mit einem Grad von inhaerenter Flexibilitaet ausgestattet werden, die fuer konstante Veraenderung ermoeglicht. Dies bedeutet, dass wenn ein bestehender Geschaeftsprozess sich aendert, oder wenn ein neuer Gesschaeftsvorgang eingefuehrt wird, muessen wir hinzufuegen, entfernen und Dienstleistungen innerhalb der Komposition Architektur mit minialem (Integration) Aufwand erweitern koennen. Deshalb ist Service Zusammensetzbarkeit eine der wichtigsten Design-Prinzipien der Service-Orientierung.

Evolutionaere Vervollkommnung ueber Streben nach anfaenglicher Perfektion

Es gibt einen gemeinsamen Punkt der Verwirrung, wenn es um den Begriff "Agility" im Verhaeltnis Zur Service-Orientierung geht. Einige Design-Ansaetze befuerworten die rasche Bereitstellung von Software-Programmen fuer die sofortigen Gewinne. Dies kann als "taktische Beweglichkeit" ausgelegt werden, da der Fokus auf dem taktischen, kurzfristigen Nutzen liegt. Service-Orientierung setzt fuer die Erreichung der Wendigkeit auf organisatorischer oder geschaeftlicher Ebene mit der Absicht der Staerkung der Organisation als Ganzes, um auf Veraenderungen reagieren zu koennen. Diese Form der organisatorischen Agilitaet kann auch als "strategische Agilitaet" bezeichnet werden, weil die Betonung auf Langlebigkeit liegt, denn mit jedem Software-Programm das wir liefern, wollen wir auf ein Ziel hinarbeiten, das Agilitaet mit langfristigem strategischen Wert foedert. Um einem IT-Unternehmen organisatorische Flexibilitaet zu ermoeglichen, muss es sich im Tandem mit dem Unternehmen weiterentwickeln. Wir koennen in der Regel nicht vorhersagen, wie ein Unternehmen sich im Laufe der Zeit weiterentwickelt; und deshalb koennen wir nicht von Anfang an perfekte Services aufbauen. Zur gleichen Zeit gibt es in der Regel eine Fuelle von Wissen, das bereits innerhalb einer Organisation vorhandene Business Intelligence existiert, das waehrend der Analyse und Modellierungs-Stadien von SOA Projecten geerntet werden kann. Diese Informationen, zusammen mit Service-Orientierungsgrundsuetzen und bewaehrten Methoden, koennen uns

helfen eine Reihe von Diensten zu indentifizieren und definieren, die erfassen, wie das Geschaeft heute betrieben wird und existiert, waehrend es flexibel genug sein muss, um sich anzupassen, wie das Geschaeft sich im Laufe der Zeit aendert.

Das heisst, obwohl wir die Werte auf der rechten Seite schaetzen, sind uns die Werte auf der linken Seite wichtiger

Durch die Untersuchung dieser Werte gewinnen wir einen Einblick, wie sie priorisiert werden und was Service-Orientierung von anderen Paradigmen unterscheidet. Von dieser Einsicht koennen IT-Fachleute in vielfacher Hinsicht profitieren. Zum Beispiel kann es grundlegende Kriterien festzulegen helfen, die wir verwenden um festzustellen, wie kompatibel Service-Orientierung fuer eine bestimmte Organisation oder ein IT-Unternehmen sein kann. Es kann weiterhin zu bestimmen helfen, inwieweit die Service-Orientierung sollte oder kann angenommen werden. Eine Anerkennung des zentralen Wertes kann uns auch zu verstehen helfen, wie schwierig es sein kann, SOA Projekte innerhalb bestimmter Umgebungen erfolgreich durchzufuehren. Zum Beispiel koennen mehrere dieser Priorisierungen zum frontalen Zusammenstoss mit etablierten Ueberzeugungen und Praeferenzen fuehren. In diesem Fall muessen die Vorteile der Service-Orientierung gegen den Aufwand und die Auswirkungen ihrer Annahme abgewogen werden (nicht nur auf die Technologie, sondern auch auf die Organisation und die IT-Kultur). Die bevorstehenden Leitprinzipien wurden entworfen, um viele Arten von Herausforderungen anzusprechen.

Wir folgen den folgenden Prinzipien:

Bisher hat das Manifest eine globale Vision gegruendet, sowie eine Reihe von Grundwerten, die mit der Vision verbunden sind. Der Rest der Erklaerung besteht aus einer Reihe von Grundsuetzen, die als Orientierungshilfe fuer die Einhaltung der Werte und die Vision vorgesehen sind. Es ist wichtig zu beachten, dass sich diese Leitgrundsuetze speziell auf die Werte und Verwirklichung der Manifestsvision beziehen. Es gibt einen separaten Satz von etablierten Design-Prinzipien, die die Service-Orientierungs -Design Paradigma enthalten; und es gibt viele weitere dokumentierte Verfahren und Muster, die spezifisch zu Service-Orientierung und Service-orientierter Architektur sind.

Respektieren der Sozial-und Machtstruktur der Organisation

Eine der haeufigsten Fallen der SOA ist die Adoption als Technologie-centric Initiative. Dies fuehrt fast immer zum Scheitern, weil wir einfach nicht fuer die unvermeidlichen organisatorischen Auswirkungen bereit sind. Die Annahme der Service-Orientierung dreht sich um die Umwandlung der Art, wie wir Geschaeftsprozesse automatisieren. Doch unabhaengig davon, welche Plaene wir fuer die Herstellung dieser Transformations-Anstrengung machen, muessen wir immer mit einem Verstaendnis und einer Wertschaetzung der Organisation beginnen, seiner Struktur, seiner Ziele und seiner Kultur. Die Annahme der Service-Orientierung ist hauptsaechlich eine menschliche Erfahrung. Es erfordert Unterstuetzung durch die Leitung und fragt, ob die Kultur eine strategische, Gemeinschafts-centriche Denkweise adoptiert. Wir muessen dieses Niveau der organisatorischen Veraenderungen voll anerkennen und planen, um die notwendigen, langfristigen Verpflichtungen, die erforderlich sind, um das Ziel der

Service-Orientierung zu erreichen. Die Art von Erwaegungen, die nicht nur uns feststellen helfen, wie man am besten mit einer SOA-Initiative weitermacht; sie unterstuetzen uns auch bei der Festlegung des geeigneten Umfangs und Ansatzes fuer die Adoption.

Erkenne an, dass SOA letztlich Veraenderung auf vielen Ebenen bedeutet

Es gibt ein Sprichwort das lautet: "Erfolg ist fuer die Gelegenheit vorbereitet". Vielleicht ist die erste Lektion die von SOA Projektion gelernt und durchgefuehrt wurde, dass wir das Volumen und die Reichweite des Wandels vollstaendig verstehen, dann planen und vorbereiten als Folge der Annahme der Service-Orientierung. Hier sind einige Beispiele: Service-Orientierung aendert wie wir Automatisierungsloesungen bauen durch die Positionierung von Software-Programmen als IT-Werte mit langfristigen Vermoegenswerte und wiederholbaren Geschaeftwert. Eine Vorab-Investion ist erforderlich, um ein Umfeld solcher Vermoegenswerte zu schaffen, und ein kontinuierliches Engagement ist erforderlich, um ihre Werte zu erhalten und zu nutzen. Also von Anfang an sind Aenderungen erforderlich, wie wir die IT-Unternehmen finanzieren, messen und unterhalten. Ausserdem, weil Service-Orientierung Services einfuehrt, die als Ressourcen des Unternehmens positioniert werden, wird es Aenderungen geben, wie wir in verschiedenen Teilen der Systeme ihre Gestaltung und Nutzen regulieren. Die Aenderungen an der Infrastruktur werden hier nicht erwaeht, um kontinuierliche Skalierbarkeit und Zuverlaessigkeit zu garantieren.

Der Bereich, in dem SOA eingefuehrt wird, kann unterschiedlich ausfallen. Halte die Aufwaende in einem ueberschaubaren und sinnvollen Rahmen

Ein allgemeiner Mythos war, um die strategischen Ziele der Service-Oriented Computing zu realisieren, muss Service-Orientierung auf eine unternehmensweite Basis angenommen werden. Dies bedeutet, Festlegung und Durchfuehrung der Design- und Industriestandards der gesamten IT Unternehmen, um eine unternehmensweite Bestandsaufnahme der eigensicheren interoperablen Dienste zu schaffen. Zwar ist an diesem Ideal nichts falsch, dennoch ist es nicht ein realistisches Ziel fuer viele Organisationen, insbesondere diejenigen mit groesseren IT-Unternehmen. Der am besten geeignete Rahmen fuer eine bestimmte SOA-Einfuehrungsadoption muss als Ergebnis der Planung und Analyse in Verbindung mit pragmatischen Ueberlegungen, wie die bereits erwaehten Auswirkungen auf die organisatorischen Strukturen bestimmt werden, Bereiche der Behoerde und kulturellen Veraenderungen herbeigefuehrt werden. Diese verschiedenen Faktoren helfen uns, einen Anwendungsbereich der Annahme, der ueberschaubar ist zu erzielen. Aber fuer alle Fortschrittsbemuehungen in einem Milieu, dass die IT-Unternehmen dem gewuenschten, strategischen Ziel naeher bringt, muss der Geltungsbereich auch sinnvoll sein. Mit anderen Worten, es muss sinnvolles Cross-Silo sein, so dass Sammlungen von Dienstleistungen in Bezug auf einander innerhalb einer vordefinierten Grenze geliefert werden. Mit anderen Worten, wir wollen "Kontinente von Dienstleistungen" schaffen, nicht die gefuerchteten "Inseln der Dienstleistungen". Dieses Konzept des Bauens von unabhagigen und privaten Service Vorrtaeten innerhalb Domains des gleichen IT-Unternehmens reduziert viele der Risiken, die allgemein "big bang" SOA-Projekte sind; und darueber hinaus die

Auswirkungen der organisatorischen und technologischen Veraenderungen mildern (weil die Wirkung sich auf einen segmentierten und verwalteten Bereich beschaenkt). Es ist auch ein Ansatz, der eine schrittweise Adoption erlaubt, wo ein Domain Service Inventar gegrundet werden kann.

Produkte und Standards alleine werden weder SOA liefern noch das Service-orientierte Paradigma umsetzen

Dieser Grundsatz hat zwei getrennte, aber sehr verwandte Mythen. Die erste ist, dass Sie sich in die SOA mit modernen Technologie-Produkten einkaufen koennen; und die zweite ist die Voraussetzung, dass die Annahme von Industriestandards (z.B. XML, WSDL, SCA, etc.) natuerlich in Service-orientierter IT-Architektur resultiert. Dem Hersteller und den Industriestandards-Gemeinden wurden den Bau moderner Dienstleister technologische Innovation auf nicht-proprietare Plattformen und Frameworks gutgeschrieben. Alles von Service-Virtualisierung zu Cloud Computing und Grid-Computing hat dazu beigetragen, das Potenzial fuer den Aufbau anspruchsvoller und komplexer Service-orientierter Loesungen zu foerdern. Allerdings sind keine dieser Technologien exklusiv fuer SOA. Man kann ebenso leicht Silo-basierte Systeme in der Wolke bauen, wie auf dem eigenen Server. Es gibt nicht so etwas wie "SOA in a box", denn um Service-orientierte IT-Architektur zu erreichen, muss Service-Orientierung erfolgreich angewandt werden; dieses wiederum setzt voraus, dass alles was wir planen und bauen durch die einzigartige Direrktion angetrieben wird, Vision und Anforderungen des Unternehmens.

SOA kann mit unterschiedlichen Technologien und Standards umgesetzt werden

Service-Orientierung ist ein technologie-neutrales und herstellerunabhaengiges Paradigma. Service-Orientierte Architektur ist ein technologie-neutrales und herstellernerneutrales Architekturmodell. Service-oriented Computing kann als eine spezielle Form des verteilten Computing betrachtet werden. Service-orientierte Loesungen koennen also mit Hilfe von fast jeden Technologien- und Industriestandards die fuer verteiltes Computing geeignet sind, gebaut werden. Waehrend einige Technologien (insbesondere solche auf Basis von Industriestandards) das Potenzial der Anwendung einiger Service-Orientierungs-Gestaltungsprinzipien erhoehen kann, ist es wirklich das Potenzial, das geschaeftliche Anforderungen erfuellt, die letztlich die am besten geeignete Wahl der Technologien und Industriestandards entscheiden.

Etabliere einheitliche Unternehmensstandards und-richtlinien auf der Basis von Industrie- und De-facto Standards sowie Standards der SOA-Gemeinde

Industriestandarde stellen nicht-proprietare Technologie Spezifikationen vor, die unter anderem helfen, gleichbleibende Baseline-Charakteristika (wie Verkehr, Schnittstelle, Nachrichtenformat usw.) der IT-Architektur zu gruenden. Allerdings garantiert die Verwendung von Industriestandards allein nicht, dass Dienstleistungen intrinsisch interoperabel werden. Das zwei Software-Programme voll kompatibel sind, muessen zusaetzhliche Konventionen (wie Datenmodelle und Politik) eingehalten werden. Deshalb muessen IT-Unternehmen die Design-Standards gruenden und durchsetzen. Eine nicht fachgerechte standardisierte und regulierte

Standardisierung von Dienstleistungen innerhalb einer bestimmten Domäne beginnt an dem Gewebe der Interoperabilität, auf denen sich die Realisierung der vielen strategischen Vorteile stützt, zu reißen. Dieses Prinzip befürwortet nicht nur den Einsatz von Enterprise-Design-Standards, es erinnert uns auch daran, dass, wann immer es möglich und machbar ist, individuelle Design-Standards sollten auf der Grundlage basieren, und Standards enthalten, die bereits allgemeine Verwendung durch die Industrie und die Gemeinschaft finden.

Strebe nach außen Einheitlichkeit an, aber lasse nach innen Vielfalt zu

Eine Federation kann als die Vereinigung von völlig verschiedenen Gebilden definiert werden. Während sich jedes Unternehmen unabhängig voneinander intern verwalten kann, befolgen alle eine gemeinsame, einheitliche Front. Ein wesentlicher Bestandteil der Service-orientierten Architektur ist die Einführung einer föderalen Endpunkt-Schicht, die Service Details der Implementierung abstrahiert, während der Veröffentlichung einer Reihe von Endpunkten, die individuellen Service innerhalb einer bestimmten Domäne in einer einheitlichen Art und Weise geben. Vervollständigung dieser Regel erfordert die Verwirklichung von Einheit aus einer Kombination von Industrie und Design-Standards. Die Konsistenz dieser Einheit über Dienstleistungen ist der Schlüssel zur Verwirklichung der intrinsischen Interoperabilität. Eine föderierte Endpunkt-Schicht trägt zur weiteren Erhöhung der Möglichkeiten bei, die Anbieter-Vielfalt Optionen zu erkunden. Beispielsweise kann ein Service auf einer völlig anderen Plattform gebaut werden. Solange dieser Service kompatible Endpunkte aufrecht erhalten, können die Governance der jeweiligen Implementierungen unabhängig bleiben. Dies zeigt nicht nur, dass Dienstleistungen mit Hilfe verschiedener Umsetzungs-Medien wie (EJB, NET, SOAP, REST, etc.) betont werden, die aber auch erfolgreich verschiedene Plattformen und Technologien gemeinsam benutzen können. Beachten Sie, dass diese Art von Vielfalt mit einem Preis kommt. Dieser Grundsatz gilt jedoch nicht für eine Diversifizierung selbst - es empfiehlt lediglich, dass wir die Diversifizierung ermöglichen, wenn dies gerechtfertigt ist, so dass "best-of-breed"-Technologien und -Plattformen genutzt werden können, um geschäftliche Anforderungen optimal zu erfüllen.

Identifizierte Services durch Zusammenarbeit zwischen fachlichen und technischen Interessenvertretern

Die Technologie muss im Gleichtakt mit dem Business sein, so dass Technologie-Lösungen business-driven sind. Deshalb ist ein weiteres Ziel der Service-Oriented Computing, Technologie und Wirtschaft auszurichten. Die Phase, in der diese Ausrichtung durchgeführt wird, ist während der Analyse und Modellierung von Prozessen, die meistens tatsächlicher Service-Entwicklung und Lieferung vorausgehen. Der kritische Inhaltsstoff, um die Durchführung der Service-orientierten Analyse auszuführen, ist, das Geschäfts- und Technologie-Experten Hand in Hand arbeiten, um Kandidat-Services zu ermitteln und festzulegen. Zum Beispiel können Business-Experten funktionale Zusammenhänge in Bezug auf Geschäfts-centric Dienste genau zu definieren helfen, während die Technologie-Experten pragmatischen Input liefern können, um sicherzustellen, dass die Granularität und Definition der konzeptionellen Dienstleistungen realistisch in Bezug auf ihre eventuelle Umsetzung bleibt.

Maximiere die Anwendbarkeit von Services durch Beruecksichtigung der derzeitigen und zukuenftigen Anwendungsgebiete

Das Ausmass eines bestimmten SOA-Projekts kann unternehmensweit sein, oder es kann zu einer Domaene des Unternehmens beschraenkt werden. Unabhaengig von der Tragweite, eine vordefinierte Grenze ist gegruendet, wenn sie eine Bestandaufnahme der Dienstleistungen einschliesst, die konzeptionell modelliert werden, bevor sie entwickelt werden koennen. Durch die Modellierung mehrerer Dienste in Beziehung zu einander haben wir im Wesentlichen einen Entwurf der Dienstleistungen geschaffen, die wir eventuell bauen werden. Diese Modellierungs-Uebung ist entscheidend, wenn man versucht Dienstleistungen zu ermitteln und festzulegen, die von verschiedenen Loesungen gemeinsam genutzt werden koennen. Es gibt verschiedene Methoden und Ansaetze, die zur Durchfuehrung von Service-orientierten Analyse-Stadien gebraucht werden koennen. Allerdings haben sie eins gemeinsam, und zwar, dass die funktionalen Grenzen von Dienstleistungen normiert werden, um Redundanz zu vermeiden. Selbst dann sind normalisierte Leistungen nicht unbedingt wieder hoch verwendbare Services. Andere Faktoren kommen ins Spiel, z.B. Service-Granularitaet, Autonomie, staatliche Verwaltung, Skalierbarkeit, Kombinierbarkeit, und in welchem Umfang Service Logik ausreichend generisch ist, so dass sie effektiv wiederverwendet werden. Diese Art von Ueberlegungen, die vom Geschaefts- und Technologie-Know-how geleitet werden, bieten die Moeglichkeit, Dienste zu definieren, die aktuelle Auslastung zu erfassen und gleichzeitig die Flexibilitaet haben, um sich zukuenftigen Veraenderungen anzupassen.

Stelle sicher, dass Services fachlichen Anforderungen und Zielen dienen

Wie mit allem, koennen Dienstleistungen missbraucht werden. Beim Aufbau und beim Management eines Portfolios von Dienstleistungen muessen ihr Verbrauch und ihre Effektivitaet bei der Erfuellung geschaeftlicher Anforderungen ueberprueft und gemessen werden. Moderne Tools bieten verschiedene Moeglichkeiten der Ueberwachung der Service-Nutzung, aber es gibt immaterielle Werte, die auch beruecksichtigt werden muessen, um sicherzustellen, dass Dienstleistungen nicht nur verwendet werden weil sie verfuegbar sind, sondern zu pruefen, ob sie wirklich in Erfuellung geschaeftlicher Anforderungen und Erwartungen sind. Dies gilt insbesondere fuer Shared Services, die mehrere Abhaengigkeiten tragen. Shared Services erfordern nicht nur adaequate Infrastruktur um die Skalierbarkeit und Zuverlaessigkeit fuer alle Loesungen zu garantieren, die ihnen die Wiederverwendung gewaehrleisten - sie muessen auch mit grosser Sorgfalt gestaltet und erweitert werden, um den funktionsfaehigen Zusammenhang zu gewaehrleisten.

Services und deren Ausgestaltung sollten sich anhand der Art und Weise, wie sie wirklich genutzt werden, entwickeln

Dieser Leitsatz verbindet direkt zurueck zu der "evolutionaeren Weiterentwicklung ueber Ausuebung der urspruenglichen Vollkommenheit" value-Anweisung, sowie das allgemeine Ziel der Aufrechterhaltung einer Ausrichtung von Business und

Technologie. Wir koennen uns nie auf Vermutungen verlassen, wenn es um die Bestimmung der Service-Granularitaet geht, die zahlreichen Funktionen, die Services brauchen um Leistungen zu bringen; oder wie Services in Kompositionen organisiert werden. Basierend auf dem Umfang der Analyse sind wir in der Lage, einen bestimmten Dienst auszufuehren, der einem definierten funktiOnalen Kotext zugeordnet wird, und eine oder mehrere funktionale Faehigkeiten enthaelt, die eine oder mehrere Service-Kompositionen beinhalten. Da sich reale, geschaeftliche Anforderungen und Umstaende aendern, muss das Service ergaenzt, erweitert, umgestaltet oder vielleicht sogar ersetzt werden. Service-orientierungs-Gestaltungsprinzipien bauen native Flexibilitaet in Service-Architekturen, so dass Software-Programme und Dienste belastbar und anpassungsfaehig auf Veraenderungen sind, als Antwort auf echten Weltverbrauch.

Trenne die verschiedenen Aspekte eines Systems, die sich unterschiedlich haeufig aendern

Was monolitische und Silo-basierte Systeme unflexibel macht, ist, dass Aenderungen erhebliche Auswirkungen auf ihre Nutzung haben koennen. Deshalb ist es oft einfacher, neue Silo-basierte Anwendungen zu erstellen, als bestehende zu erweitern. Die Logik hinter der Trennung von Anliegen (eine allgemein bekannte Software-Engineering-Theorie) ist, dass ein groesseres Problem effektiver geloest werden kann, wenn es in eine Reihe von kleineren Problemen oder Bedenken zerlegt wird. Bei der Anwendung von Service-Orientierung an die Trennung von Bedenken, bauen wir die entsprechenden Einheiten der Loesungs-Logik, die individuelle Anliegen loesen, wodurch es uns ermoeoglicht, die Einheiten zu aggregieren um die groesseren Probleme zu loesen; darueber hinaus geben sie uns die Moeglichkeit, sie in verschiedene Konfigurationen zu aggregieren, um andere Probleme zu loesen. Neben der Foerderung der Service Wiederverwendbarkeit, fuehrt dieser Ansatz viele Schichten der Abstraktion ein, die dazu beitragen, Service kompromittierte Systeme vor den Auswirkungen der Veraenderungen zu schuetzen. Diese Form der Abstraktion kann auf verschiedenen Ebenen existieren. Zum Beispiel, wenn Legacy-Ressourcen, die von einem Service gekapselt ersetzt werden muessen, koennen die Auswirkungen dieser Veraenderung gemildert werden, solange der Service in der Lage ist, seinen urspruenglichen Endpunkt und sein funktionales Verhalten beizubehalten. Ein weiteres Beispiel ist die Trennung von Agnostik zu nicht-Agnostik. Die erstgenannte Art von Logik hat hohes Wiederverwendbarkeits-Potenzial, wenn es Mehrzweckverwendung hat und sich wahrscheinlich wenig aendern wird. Die nicht-agnostische Logik auf der anderen Seite, in der Regel stellt die Einzweckteile der uebergeordneten Geschaeftsprozess-Logik, die oft mehr volatil sind. Die Trennung dieser jeweiligen Logik Typen in unterschiedliche Service Schichten stellt weitere Abstraktion vor, die Service-Wiederverwendbarkeit ermoeoglicht, waehrend sie Dienstleistungen sowie alle Loesungen, die sie benutzen, von den Auswirkungen des Wandels abschirmt.

Reduziere implizite Abhaengigkeiten und publiziere alle externen Abhaengigkeiten, um Robustheit zu foerdern und die Auswirkungen von Veraenderungen zu reduzieren

Einer der bekanntesten Design-Prinzipien der Service-Orientierung ist, dass der Service loose coupling. Wie eine Service-Architektur intern strukturiert ist, und wie Dienstleistungen sich auf den Verbrauch von Programmen beziehen (die andere

Dienstleistungen umfassen koennen), welche hauptsaechlich von den individuellen beweglichen Teilen abhaengig sind und ein Teil der Service-Architektur bilden. Abstraktionsebenen lindern evolutionaere Veraenderungen bei der Lokalisierung der Auswirkungen der Umstellung auf kontrollierte Regionen. Zum Beispiel koennen innerhalb Service-Architekturen Service-Fassaden benutzt werden und abstrakte Teile der Implementierung verwendet werden, um die Reichweite der Umsetzungs-Abhaengigkeit zu minimieren. Andererseits muessen veroeffentliche technische Servicevertraege die Abhaengigkeit bekannt machen, die die Service-Verbraucher entwickeln, um mit Services zu interagieren. Um interne Abhaengigkeiten zu reduzieren, die diese technischen Vertraege beeinflussen wenn sich Veraenderungen ergeben, vermeiden wir die Proliferation dieser Veraenderungen fuer die abhaengigen Service Verbraucher.

Organisiere jeden Service auf jeder Abstraktionsebene in oder anhand einer zusammenhaengenden und ueberschaubaren Funktionseinheit

Jeder Dienst erfordert einen gut definierten, funktionalen Kontext, der bestimmt, welche Logik nicht in den Dienst der funktionalen Grenzen gehoert. Festlegung des Umfangs und Granularitaet dieser funktionalen Service Grenzen ist einer der wichtigsten Aufgaben waehrend des Service Delivery Lifecycles. Services mit grober funktionaler Granularitaet sind moeglicherweise zu unflexibel, um wirksam zu sein, besonders wenn sie wiederverwendbar sein sollen. Andererseits koennen allzu feinkoernige Services die Infrastruktur belasten, da die Service-Kompositionen aus einer groesseren Menge von Kompositions-Mitgliedern bestehen. Das Ermitteln der richtigen Balance vom funktionalen Umfang und der Granularitaet, erfordert eine Kombination von Business und Technologie know-how und weiterhin erfordert ein Verstaendnis, wie sich Dienstleistungen innerhalb einer bestimmten Grenze zueinander beziehen. Viele der Leitprinzipien, die in diesem Manifest beschrieben sind, werden diese Bestimmung zur Unterstuetzung der Positionierung jedes Dienstes als ein IT-Wert sehen, der faehig ist, ein IT-Unternehmen zum Zielzustand zu bringen, wobei die strategischen Vorteile der Service-oriented Computing realisiert werden.

Danksagung: Ich schrieb den Inhalt dieser Seiten am Wochenende von November 21-22, 2009 fuer das "Next Generation SOA" Buch, dass zu dieser Zeit noch in der Entwicklung war. Ich moechte mich bei Prentice Hall fuer die Erlaubnis bedanken, dass der SOA Inhalt dieser Website als Ergaenzung zum Original veroeffentlicht wurde. Ich moechte mich auch bei den Mitgliedern der Arbeitsgruppe bedanken und den Mitgliedern der Gemeinschaft des SOA Patterns.org Annotationen fuer feedback und Hilfe. Viele Mitglieder der Arbeitsgruppe haben blogs, ihre eigenen Artikel und Papiere ueber das SOA Manifest veroeffentlicht; und ich ermutige Sie alle, weiterhin Kommentare abzugeben und nach neuen Einblicken zu suchen.

- Thomas Erl

[Original SOA Manifesto](#) [PDF](#) [About the Translators](#)

