

---

# Annotations sur le manifeste SOA

Commentaires et réflexions sur le manifeste SOA par Thomas Erl

L'orientation services est un paradigme qui donne un cadre à ce que vous faites. L'architecture orientée services est un type d'architecture qui résulte de l'application de l'orientation services.

Depuis le début, il fût convenu que ceci allait être un manifeste ayant pour sujet deux thèmes, distincts mais étroitement liés: le modèle architectural orienté services, et l'orientation services, le paradigme au moyen duquel l'architecture est définie. Le format de ce manifeste se base sur celui du Manifeste Agile, qui limite son contenu à l'expression concise des ambitions, des valeurs et des principes directeurs pour réaliser ces ambitions et ces valeurs. Un tel manifeste n'est ni une spécification, ni une référence ni même un livre blanc, et, à défaut de pouvoir donner des définitions précises, nous avons donc décidé d'ajouter ce préambule de manière à clarifier comment et pourquoi ces termes sont référencés dans d'autres parties du manifeste.

Nous avons mis en pratique l'orientation services...

Le paradigme orienté services doit être vu comme une méthode ou une démarche permettant d'atteindre un état cible spécifique défini par un ensemble d'objectifs et de bénéfices stratégiques. Quand nous appliquons l'orientation services, nous façonnons les logiciels et l'architecture technologique dans le but d'atteindre cet état cible. C'est ce qui qualifie l'architecture technologique comme étant orientée services.

...afin d'aider les organisations à délivrer de la valeur métier durable, de façon régulière avec une agilité et une rentabilité accrues...

La suite du préambule met en avant certains des bénéfices stratégiques les plus remarquables et les plus fréquemment attendus de l'informatique orientée services. Comprendre ces bénéfices aide à clarifier l'état cible mentionné plus haut, celui que nous pensons atteindre comme résultat de l'application de l'orientation services.

Au niveau métier, l'agilité est comparable à la capacité de réponse d'une organisation. Plus une organisation pourra réagir facilement et efficacement aux variations métier, plus elle sera efficace et pourra s'adapter avec succès aux impacts du changement (et ainsi profiter au mieux des bénéfices que ce changement pourra lui apporter).

L'orientation services positionne les services comme des actifs informatiques dont on attend qu'ils fournissent une valeur répétitive dans le temps, valeur dont on

espère qu'elle excèdera de beaucoup l'investissement initial nécessaire à leur développement. Le terme « rentabilité » fait référence principalement à ce retour sur l'investissement attendu. Sous beaucoup d'aspects, une augmentation de la rentabilité va de pair avec une augmentation en agilité; S'il y a plus de possibilités de réutilisation des services existants, il y aura généralement besoin de moins de dépenses pour construire de nouvelles solutions.

Le terme « Valeur métier durable » fait référence aux objectifs à long terme de l'orientation services qui consistent à mettre en place des logiciels sous forme de services, ayant la flexibilité intrinsèque de pouvoir être constamment composés en de nouvelles configurations de solutions et d'évoluer pour s'adapter à des exigences métiers en constante mutation.

...tout en tenant compte de la nature changeante des besoins métier.

Ces derniers mots du préambule sont la clef pour comprendre la philosophie sous-jacente de l'informatique orientée services. Le besoin de s'adapter constamment aux changements métier, fait partie des fondations de l'orientation services et est considéré comme étant un objectif stratégique fondamental.

À travers notre travail nous avons été amenés à privilégier :

Les affirmations qui vont suivre établissent un ensemble de valeurs clés, Chacune de ces valeurs est exprimée comme une préférence sur quelque chose considéré, aussi, comme étant de valeur. Le but de ce système de valorisation est de confronter les décisions difficiles qu'il faut prendre régulièrement pour que les objectifs stratégiques et les bénéfices de l'informatique orientée services se réalisent de manière permanente.

La valeur métier par rapport à la stratégie technique

Comme établi précédemment, le besoin de s'adapter aux changements métier est un objectif stratégique primordial. En conséquence, la qualité fondamentale de l'architecture orientée services - et de n'importe quel logiciel ou écosystème résultant de l'adoption de l'orientation services d'ailleurs - est qu'il soit orienté métier. Il ne faut pas que la technologie détermine la direction du métier, mais bien que la vision métier dicte l'utilisation de la technologie.

Cette priorité peut avoir un effet de vague profond au sein d'une direction informatique. Cela introduit des changements dans à peu près toutes les phases du cycle de vie des systèmes d'information, depuis la manière dont nous planifions et finançons des solutions automatisées, jusqu'à la façon dont nous les construisons et les gérons. Toutes les autres valeurs et principes du manifeste renforcent, d'une manière ou d'une autre, la réalisation de cette valeur.

Les objectifs stratégiques par rapport aux bénéfices spécifiques de projets particuliers

Historiquement, de nombreux projets informatiques se sont concentrés exclusivement sur la construction d'applications conçues spécifiquement pour automatiser les exigences des processus métier tels qu'ils existaient à ce moment.

Cela satisfaisait les exigences immédiates (tactiques), mais au fur et à mesure que plus de ces applications à but unique se déployaient, on assistait à la construction d'un département informatique couvert d'îlots de logique et de données appelés « applications silos ». Avec l'émergence de nouvelles exigences métiers, il fallut créer de nouveaux silos ou bien établir des canaux d'intégration entre les silos. Et à mesure que de nouveaux changements se présentèrent, il fallut alors augmenter les canaux d'intégration et créer encore plus de silos, et bientôt, le panorama du département informatique devint convoluté et de plus en plus complexe, coûteux et lent à évoluer.

Sous de multiples aspects, l'orientation services est apparue en réponse à ces problèmes. C'est un paradigme qui fournit une alternative au développement d'applications intégrées, basées sur les silos et spécifiques à des projets particuliers, en priorisant catégoriquement l'atteinte d'objectifs stratégiques métier à long terme. L'état cible préconisé par l'orientation services ne contient pas d'application silo traditionnelle. Et même si des ressources et des applications silos existent dans des environnements où l'orientation services a été adoptée, l'état cible est un état où ces ressources et ces applications sont harmonisées au plus haut degré possible.

L'interopérabilité intrinsèque par rapport à l'intégration sur-mesure

Pour que des logiciels partagent des données, ils doivent être interopérables. Si les logiciels ne sont pas conçus pour être compatibles, ils ne seront sans doute pas interopérables. Pour permettre l'interopérabilité entre des logiciels incompatibles, il faut les intégrer. L'intégration est, de ce fait, l'effort nécessaire pour obtenir l'interopérabilité entre des logiciels hétérogènes.

Quoique souvent nécessaire, l'intégration sur mesure peut être coûteuse en temps et en argent, et peut conduire à des architectures fragiles qui sont vraiment fastidieuses à faire évoluer. Un des objectifs de l'orientation services est de minimiser la nécessité d'intégration sur mesure, en concevant les logiciels (dans un domaine particulier), pour qu'ils soient compatibles nativement. Ceci est une qualité décrite comme « interopérabilité intrinsèque ». Le paradigme de l'orientation services couvre un ensemble de principes de conception qui visent à l'établissement de l'interopérabilité intrinsèque à différents niveaux.

L'interopérabilité intrinsèque, en tant que caractéristique des logiciels d'un domaine particulier, est un facteur-clé pour atteindre des bénéfices stratégiques, tels que la rentabilité accrue et l'agilité.

Les services partagés par rapport aux implémentations spécifiques

Comme nous venons de l'expliquer, l'orientation services établit une démarche de conception formée d'un ensemble de principes de conception. Quand on les applique à un degré significatif, ces principes permettent de concevoir un logiciel comme une unité de logique orientée services que l'on peut légitimement nommer un service.

Les services possèdent des caractéristiques concrètes (telle que celles qui permettent l'interopérabilité intrinsèque) qui supportent directement l'état cible décrit plus haut. Une de ces caractéristiques est l'encapsulation de logique polyvalente qui peut être partagée et réutilisée lors de l'automatisation de différents processus métier.

Un service partagé se définit comme étant un actif informatique qui peut fournir de la valeur métier répétitive tout en réduisant la dépense et l'effort nécessaires pour fournir de nouvelles solutions automatisées. Bien qu'il y ait de la valeur dans les applications traditionnelles à objectif unique qui répondent aux exigences métiers tactiques, l'utilisation de services partagés fournit une plus grande valeur en réalisant les objectifs stratégiques de l'informatique orientée services (ce qui comprend, une fois de plus, une augmentation de la rentabilité et de l'agilité).

### La flexibilité par rapport à l'optimisation

Ceci est probablement la déclaration de valeur la plus générale et doit se comprendre comme une philosophie directrice permettant de prioriser de la meilleure manière, différentes considérations au moment de produire et de faire évoluer les services individuels et les inventaires de services.

L'optimisation se réfère surtout à la réalisation de bénéfiques tactiques, en optimisant la conception d'une certaine application ou en accélérant sa livraison pour faire face à des nécessités immédiates. Il n'y a rien de mal à cela, sauf que cela peut conduire à des environnements basés sur des silos, comme ceux mentionnés plus haut, si l'on a pas suffisamment pris en compte l'aspect flexibilité.

Par exemple, la caractéristique de flexibilité peut aller au-delà de la capacité des services de partager des données de manière effective et intrinsèque. Pour avoir une vraie capacité de réponse face aux exigences métier toujours changeantes, les services doivent également être flexibles dans leur capacité à être combinés et agrégés en des solutions composites. À la différence des applications distribuées traditionnelles qui étaient souvent relativement statiques en dépit du fait qu'elles étaient basées sur des composants, les compositions de services doivent être conçues avec un niveau de flexibilité intrinsèque qui permet leur accroissement constant. Ceci signifie que lorsqu'un processus métier existant change ou lorsque un nouveau processus métier est introduit, nous avons besoin de pouvoir ajouter, supprimer et étendre les services à l'intérieur d'une architecture de composition avec un effort minimum d'intégration. C'est pour cela que la composabilité des services est un des principes de conception-clé de l'orientation services.

### L'amélioration évolutive par rapport à la quête de la perfection initiale

Il y a beaucoup de confusion autour du terme « agilité » en relation avec l'orientation services. Certaines méthodes de conception préconisent le déploiement rapide de logiciels pour obtenir des bénéfices immédiats. Ceci peut être vu comme une « agilité tactique », puisque l'on se focalise sur le bénéfice à court terme, tactique. L'orientation services préconise d'atteindre l'agilité au niveau organisationnel ou métier dans le but que l'organisation dans son ensemble puisse avoir la capacité de répondre aux changements. Cette forme d'agilité organisationnelle peut aussi s'appeler « agilité stratégique » parce que l'on se focalise sur la longévité dans la mesure où avec chaque nouveau logiciel que nous livrons, nous visons un état cible qui privilégie l'agilité ainsi que la valeur stratégique à long terme.

Pour qu'un département informatique rende l'agilité organisationnelle possible, celle-ci doit évoluer en parallèle avec le métier. En général, nous ne pouvons pas prédire comment le métier devra évoluer dans le temps et de ce fait, nous ne

pouvons pas construire dès le début des services parfaits. En même temps, il existe généralement déjà une masse de connaissances dans l'intelligence métier existante de l'organisation qui peut être récoltée durant les phases d'analyse et de modélisation des projets SOA.

Cette information, ainsi que les principes de l'orientation services et des méthodologies ayant fait leurs preuves, peuvent nous aider à identifier et définir un ensemble de services qui capturent la manière dont le métier existe et fonctionne aujourd'hui, tout en étant suffisamment flexible pour s'adapter en fonction de comment le métier évoluera dans le temps.

Cela signifie que, bien que nous pensions que tous ces éléments sont importants, les premiers nous semblent primordiaux.

En étudiant comment ces valeurs sont priorisées, nous obtenons un meilleur aperçu de ce qui fait la différence entre l'orientation services et d'autres paradigmes. Les professionnels de l'informatique peuvent bénéficier de ce type d'aperçu de différentes manières. Par exemple, cela peut nous aider à établir des critères fondamentaux que nous pouvons utiliser pour déterminer le degré de compatibilité de l'orientation services avec une certaine organisation ou département informatique. En outre, cela peut également nous aider à déterminer le degré auquel l'orientation services peut ou devrait être adopté. L'appréciation des valeurs-clés peut aussi nous aider à comprendre le défi que peut représenter la mise en place avec succès de projets SOA dans certains environnements. Par exemple, plusieurs de ces priorités peuvent entrer en conflit avec des croyances ou des préférences établies. Dans ce cas-là, les bénéfices de l'orientation services doivent être soupesés face à l'effort et à l'impact possibles de leur adoption (non seulement sur la technologie, mais aussi sur l'organisation et la culture informatique).

Les principes directeurs présentés ci-dessous sont fournis pour nous aider à relever plusieurs de ces types de défi.

Nous suivons les principes suivants :

Jusqu'ici, le manifeste a établi une vision globale ainsi qu'un ensemble de valeurs centrales associées à cette vision. Le reste de cette déclaration consiste en un ensemble de principes qui sont fournis comme un guide pour permettre l'adhésion aux valeurs et la réalisation de la vision.

Il est important de garder en tête qu'il s'agit de principes directeurs propres à ce manifeste. Il existe par ailleurs un ensemble de principes de conception bien établis qui couvrent le paradigme de l'orientation services et également, un grand nombre de motifs et pratiques documentés spécifiques à l'orientation services et à l'architecture orientée services.

Respecter la structure sociale et décisionnelle de l'organisation.

Un des pièges les plus courants de l'Architecture Orientée Services est d'aborder son adoption comme une initiative centrée sur la technologie. Le faire de cette manière conduit presque toujours à l'échec, simplement parce que nous ne sommes pas préparés aux impacts organisationnels inévitables. L'adoption de l'orientation services consiste à transformer la manière dont nous automatisons le métier.

Cependant, indépendamment des plans que nous pouvons avoir pour faire aboutir cet effort de transformation, nous devons toujours commencer par la compréhension et l'appréciation de l'organisation, sa structure, ses objectifs et sa culture.

L'adoption de l'orientation services est sous beaucoup d'aspects une expérience humaine. Elle nécessite le support de ceux qui détiennent l'autorité, et ensuite exige qu'une culture informatique adopte une vision stratégique et centrée sur la communauté. Nous devons absolument reconnaître et planifier ce niveau de changement organisationnel afin de pouvoir recevoir les engagements à long terme nécessaires pour parvenir à l'état cible de l'orientation services.

Ces types de considérations nous aideront non seulement à déterminer comment procéder de la meilleure manière dans le cadre de cette initiative SOA, mais, qui plus est, elles nous aideront à définir le degré et la démarche d'adoption les plus appropriés.

Reconnaître que l'Architecture Orientée Services exige en fin de compte des changements à plusieurs niveaux.

Il existe une maxime selon laquelle « le succès consiste à être prêt pour toute opportunité ». Peut-être que la leçon numéro un que nous puissions tirer des projets SOA exécutés jusqu'à présent est que nous devons parfaitement comprendre et ensuite planifier et nous préparer au volume et à l'éventail de changements qui se produiront en conséquence de l'adoption de l'orientation services. Voici quelques exemples :

L'orientation services change la manière dont nous construisons les solutions automatisées en positionnant les logiciels comme des actifs informatiques fournissant de la valeur métier à long terme et de manière répétitive. Un investissement initial est nécessaire pour créer un environnement composé de tels actifs et un engagement permanent est requis pour maintenir et promouvoir leur valeur. Ainsi, dès le départ, des changements sont nécessaires quant à la façon dont nous finançons, mesurons et maintenons les systèmes dans l'organisation informatique.

De plus, puisque l'orientation services introduit des services qui sont positionnés comme des ressources de l'entreprise, il y aura des changements dans la manière dont nous nous approprions les différentes parties du système et dont nous régulons leur conception et leur utilisation, sans parler des changements de l'infrastructure nécessaires pour garantir une extensibilité et une fiabilité continues.

La portée de l'adoption de l'Architecture Orientée Services peut varier. Les efforts doivent rester gérables et dans des limites significatives.

Un mythe commun veut que pour pouvoir atteindre les objectifs stratégiques de l'informatique orientée services, l'orientation services doit être adoptée au sein de toute l'entreprise. Ceci signifie établir et imposer des standards de conception et industriels, à travers l'organisation informatique, de manière à créer un inventaire global de services intrinsèquement interopérables. Bien qu'il n'y ait rien de mal dans cet idéal, ce n'est pas un objectif réaliste pour de nombreuses organisations, particulièrement celles possédant une large organisation informatique.

Le degré le plus adéquat de tout effort d'adoption SOA doit être déterminé comme le résultat d'une planification et d'une analyse en conjonction, avec des considérations pragmatiques, tels que les impacts mentionnés plus haut sur les structures organisationnelles, les zones d'autorité, et les changements culturels qui en résulteront.

Ces types de facteurs nous aident à déterminer un degré d'adoption qui est gérable. Mais pour qu'un effort d'adoption ait comme résultat un environnement qui fasse progresser l'initiative informatique vers l'état cible stratégique désiré, l'étendue de cet effort doit aussi être significatif. En d'autres termes, l'initiative doit être significativement « inter-silos », pour que des collections de services puissent être livrés en relation les uns avec les autres dans des limites prédéfinies. En d'autres termes, nous voulons créer des « continents de services », et non les redoutables « îlots de services ».

Ce concept consistant à construire des inventaires de services gérés et gouvernés de manière indépendante, dans différents domaines de la même organisation informatique, réduit beaucoup les risques qui sont généralement associés aux projets SOA de type « big-bang » et, de plus, réduit l'impact des changements tant organisationnels que technologiques (puisque l'impact se limite à une étendue segmentée et gérable). C'est aussi une démarche qui permet l'adoption progressive en établissant un seul inventaire de services de domaine à la fois.

Les produits et les standards seuls ne nous livreront jamais l'Architecture Orientée Services ni n'appliqueront les paradigmes orientés services à notre place.

Ce principe adresse deux mythes distincts mais étroitement liés. Le premier selon lequel vous pouvez acheter votre passage à une Architecture Orientée Services avec des produits de technologie moderne, et le second qui présuppose que l'adoption de standards industriels (tels que XML, WSDL, SCA, etc.) résultera naturellement en une architecture technologique orientée services.

Les communautés de vendeurs et de standards industriels passent pour construire des innovations en matière de technologies modernes de services, en se basant sur des frameworks et des plates-formes non propriétaires. Depuis la virtualisation des services, en passant par l'informatique dans le nuage et la grille informatique, tout a aidé à faire avancer le potentiel pour construire des solutions orientées services, sophistiquées et complexes. Cependant, aucune de ces technologies n'est propre à l'Architecture Orientée Services. Vous pouvez tout aussi facilement construire des systèmes basés en silos sur le nuage que vous pouvez le faire sur vos propres serveurs privés.

Il n'existe rien de tel que « la SOA en boîte » parce que pour pouvoir arriver à l'architecture technologique orientée services, l'orientation services doit être appliquée avec succès; ceci, à son tour, exige que tout ce que nous concevons et construisons soit mené par la direction unique, la vision et les exigences du métier.

L'Architecture Orientée Services peut être mis en œuvre à travers une multitude de technologies et standards.

L'orientation services est un paradigme neutre par rapport aux technologies et aux vendeurs. L'architecture orientée services est un modèle architectural neutre par

rapport aux technologies et aux vendeurs. L'informatique orientée services peut être considérée comme une forme d'informatique distribuée. Les solutions orientées services peuvent de ce fait être construites en utilisant pratiquement n'importe quelle technologie et standards industriels adéquats pour l'informatique distribuée.

Bien que certaines technologies (particulièrement celles basées sur les standards industriels) peuvent augmenter le potentiel d'application de certains principes de conception orientés services, c'est réellement la capacité de pouvoir satisfaire aux exigences métier qui en fin de compte déterminera le choix de technologies et de standards industriels le plus approprié..

Etablir un ensemble uniforme de normes et règles d'entreprise, fondées sur des standards industriels, de fait et communautaires.

Les standards industriels représentent des spécifications technologiques non propriétaires, qui aident à établir, entre autres, des caractéristiques de base cohérentes (telles que le transport, l'interface, le format des messages, etc.) de l'architecture technologique. Cependant, l'utilisation seule des standards industriels, ne garantit pas que les services seront intrinsèquement interopérables.

Pour que deux logiciels soient totalement compatibles, il faut qu'ils adhèrent à des conventions supplémentaires (telles que des modèles de données et des polices). C'est pourquoi les entreprises informatiques doivent mettre en place et imposer des standards de conception. Un échec dans la standardisation et régulation adéquate des services à l'intérieur d'un certain domaine commencera à déchirer la trame de l'interopérabilité sur laquelle repose la réalisation de nombreux bénéfices stratégiques.

Ce principe non seulement préconise l'utilisation de standards de conception d'entreprise, mais nous rappelle aussi que, chaque fois que c'est possible et faisable, les standards de conception spécifiques devraient être basés sur - et inclure - des standards déjà en usage dans l'industrie et la communauté en général.

Rechercher l'uniformité dans les échanges avec le monde extérieur, tout en permettant la diversité en interne.

Une fédération peut être définie comme l'unification d'un ensemble de différentes entités. Tout en permettant à chaque entité d'être gouvernée de manière indépendante à l'intérieur, toutes ces entités acceptent d'adhérer à un front unifié et commun. Une partie fondamentale de l'architecture orientée services est l'introduction d'une couche fédérée de points d'entrée qui abstrait les détails d'implémentation des services tout en publiant un ensemble de points d'entrée qui représentent les services individuels dans un certain domaine et de manière unifiée. Obtenir ce résultat implique généralement d'atteindre l'unité sur la base d'une combinaison de standards de conception et de l'industrie. La consistance de cette unité à travers les services est clé pour réaliser l'interopérabilité intrinsèque. Une couche fédérée de points d'entrée permet en plus d'explorer les options de diversification de vendeurs. Par exemple, un service peut avoir besoin d'être construit sur une plate-forme complètement différente d'un autre service. Tant que ces services maintiennent des points d'entrée compatibles, la gouvernance de leur mise en oeuvre respective peut rester indépendante. Ceci met en relief que non seulement les services peuvent être construit en utilisant des moyens de mise en

oeuvre différents, (tels que EJB, .NET, SOAP, REST, etc.) mais également que l'on peut utiliser un ensemble des technologies et des plates-formes intermédiaires différentes, selon la nécessité.

Notez que ce type de diversité a un prix. Ce principe ne préconise pas la diversification en tant que telle, il recommande simplement de permettre la diversification quand elle se justifie, de manière à ce que les technologies et plates-formes les « meilleures de leur classe » puissent être utilisées pour maximiser la satisfaction des exigences métier.

Identifier les services en collaborant avec les intervenants métier et des technologiques.

Pour que les solutions technologiques soient orientées métier, la technologie doit être synchronisée avec le métier. En conséquence, un autre objectif de l'informatique orientée services est d'aligner la technologie et le métier. Le moment au cours duquel cet alignement a lieu se déroule durant les processus d'analyse et de modélisation qui précèdent normalement le développement et la mise en place effective du service.

L'ingrédient critique pour effectuer l'analyse orientée services est d'arriver à ce que les experts métier et technologiques travaillent main dans la main pour identifier et définir les services candidats. Par exemple, les experts métiers peuvent aider à définir de manière précise les contextes fonctionnels relatifs aux services centrés métier, pendant que les experts en technologie fournissent de l'input pragmatique pour assurer que la granularité et la définition des services conceptuels restent réalistes par rapport à leurs plates-formes de mise en oeuvre finales.

Maximiser l'utilisation des services en tenant compte du degré d'utilisation actuel et futur.

L'étendue d'un certain projet SOA peut-être au niveau de l'entreprise entière ou bien peut se limiter à un domaine particulier de l'entreprise. Quelle que soit l'étendue, on établit un périmètre prédéfini pour délimiter un inventaire de services qui doivent être définis et modélisés conceptuellement avant de pouvoir être développés. En modélisant de multiples services en relation les uns les autres, nous pouvons établir essentiellement les plans des services qui seront finalement construits. Cet exercice de modélisation est critique quand on essaye d'identifier et de définir les services qui peuvent être partagés par différentes solutions.

Il existe différentes méthodologies et démarches qui peuvent être utilisées pour exécuter les étapes de l'analyse orientée services. Cependant, un fil conducteur commun à toutes est que les périmètres fonctionnels des services doivent être normalisés pour éviter la redondance. Même ainsi, les services normalisés ne donneront pas nécessairement comme résultat des services hautement réutilisables. D'autres facteurs entrent en jeu, tels que la granularité du service, son autonomie, la gestion des états, l'extensibilité, la composabilité, et le degré dans lequel la logique du service est suffisamment générique pour pouvoir être effectivement réutilisée.

Ces types de considérations, guidées par les expertises technologiques et métier fournissent l'occasion de définir des services qui capturent les exigences actuelles d'utilisation tout en ayant la flexibilité nécessaire pour s'adapter aux changements futurs.

Vérifier que les services satisfont aux exigences et objectifs métier.

Comme pour tout, les services peuvent être mal utilisés. Quand on gère et on fait évoluer un portefeuille de services, leur utilisation et leur efficacité pour satisfaire les exigences métier doivent être vérifiées et mesurées. Des outils modernes fournissent différentes manières de surveiller l'utilisation des services, mais il existe également des facteurs intangibles qui doivent être pris en considération pour s'assurer que les services ne sont pas utilisés uniquement parce qu'ils existent, mais pour vérifier qu'ils satisfont réellement les exigences métier attendues.

Ceci est particulièrement vrai pour les services partagés qui supportent de multiples dépendances. Non seulement les services partagés exigent une infrastructure adéquate pour garantir l'extensibilité et la fiabilité de toutes les solutions qui les réutilisent, mais ils doivent pouvoir aussi être conçus et étendus avec de grandes précautions pour assurer que leurs contextes fonctionnels ne sont jamais biaisés.

Faire évoluer les services et leur organisation en réponse à l'utilisation réelle.

Ce principe directeur est directement lié à la déclaration de valeur décrite plus haut « L'amélioration évolutive par rapport à la quête de la perfection initiale », ainsi qu'à l'objectif global d'assurer l'alignement métier avec la technologie. Nous ne devons jamais nous en remettre au hasard quand il s'agit de déterminer la granularité des services, l'étendue des fonctions qu'ils doivent exécuter, ou la manière dont ils devront être organisés en compositions. Sur la base de l'analyse que nous avons pu faire initialement, et quelle que soit l'étendue de cette analyse, un service recevra un contexte fonctionnel défini, et contiendra une ou plusieurs capacités fonctionnelles qui le feront vraisemblablement participer à une ou plusieurs compositions de services.

Dans la mesure où les exigences métier et les circonstances du monde réel changent, le service peut avoir besoin d'être augmenté, étendu, reconstruit ou peut-être même remplacé. Les principes de conception de l'orientation services permettent de construire une flexibilité native dans les architectures de services de manière à ce que, tout comme les logiciels, les services deviennent résistants et adaptables aux changements et puissent être modifiés en réponse à l'utilisation du monde réel.

Séparer les différents aspects d'un système qui ont des taux de changement différents.

Ce qui rend les systèmes monolithiques et basés sur les silos inflexibles, c'est que les changements peuvent avoir un impact significatif sur leur utilisation existante.

C'est la raison pour laquelle il est souvent plus facile de créer de nouvelles applications silos que d'augmenter ou étendre les applications existantes. Le raisonnement derrière la séparation des problèmes (une théorie courante en informatique) est que un grand problème peut être plus facilement résolu quand on le décompose en une série de problèmes plus petits, qu'on appellerait « souci » ou « préoccupation ». En appliquant l'orientation services à la séparation des problèmes,

nous construisons des unités de logique qui offrent une solution aux soucis individuels, et de cette manière nous permettent de les recombinaison pour résoudre des problèmes plus grands, et qui plus est, nous donnent l'occasion de les recombinaison en différentes configurations pour pouvoir résoudre d'autres problèmes.

En plus de promouvoir la capacité de réutilisation des services, cette approche introduit de nombreuses couches d'abstraction qui aident à protéger des impacts des changements, les systèmes construits sur des services. Cette forme d'abstraction peut exister à différents niveaux. Par exemple, si des ressources obsolètes et encapsulées par un service doivent être remplacées, l'impact de ce changement peut être mitigé tant que le service est capable de conserver son point d'entrée original et son comportement fonctionnel.

Un autre exemple est la séparation entre logique agnostique et non agnostique. Le premier type de logique a un potentiel de réutilisation plus élevé si elle est polyvalente et moins susceptible de changer. La logique non agnostique, par contre, représente typiquement les parties à but unique de la logique du processus métier parent, qui sont souvent plus volatiles. La séparation de ces types de logique respectifs en différentes couches de services, introduit encore plus d'abstraction, qui facilite la réutilisabilité des services tout en protégeant les services et toutes les solutions qui les utilisent, des impacts du changement.

Réduire les dépendances implicites et publions toutes les dépendances externes pour augmenter la robustesse et réduire l'impact du changement.

Un des principes les plus connus de l'orientation services est celui du couplage faible. La manière dont une architecture de services est structurée en interne, et la façon dont les services sont reliés aux programmes qui les consomment (qui peuvent à leur tour inclure d'autres services), tout se résume aux dépendances qui se forment avec des parties dynamiques individuelles qui font partie de l'architecture services.

Les couches d'abstraction aident à faciliter les changements évolutifs en localisant les impacts du changement sur des régions contrôlées. Par exemple, dans les architectures services, les façades de services peuvent être utilisées pour abstraire une partie de l'implémentation de manière à minimiser la portée de ses dépendances.

Par ailleurs, les contrats de services techniques publiés doivent révéler les dépendances que les consommateurs de services devront accepter pour pouvoir interagir avec les services. En réduisant les dépendances internes qui peuvent affecter ces contrats techniques lorsque des changements ont lieu, nous évitons la prolifération de l'impact de ces changements sur les consommateurs de services qui en dépendent.

A tous les niveaux d'abstraction, organiser chaque service autour d'une fonctionnalité cohérente et gérable.

Chaque service exige un contexte fonctionnel bien défini qui détermine la logique qui appartient ou n'appartient pas aux limites fonctionnelles du service. Déterminer l'étendue et la granularité de ces limites fonctionnelles est une des responsabilités les plus critiques durant le cycle de vie de la construction du service.

Les services avec granularité fonctionnelle grossière peuvent être trop inflexibles pour être efficaces, en particulier si on attend d'eux qu'ils soient réutilisables. D'un autre côté, les services de granularité trop fine peuvent pénaliser une infrastructure dans la mesure où les compositions de services seront constituées d'une quantité plus grande de membres.

La détermination de l'équilibre judicieux entre la portée fonctionnelle et la granularité exige une combinaison d'expertise technologique et métier, qui exige à son tour la compréhension de comment les services à l'intérieur d'un périmètre déterminé vont se relier les uns aux autres.

Beaucoup des principes directeurs décrits dans ce manifeste nous aideront dans notre détermination à vouloir positionner chaque service comme un actif informatique capable de faire avancer un département informatique vers un état cible dans lequel les bénéfices stratégiques de l'informatique orientée services peuvent être récoltés.

En fin de compte, cependant, ce sera toujours l'obtention de valeur métier du monde réel qui dictera, de la conception jusqu'à la mise en place et jusqu'à l'utilisation répétée, le chemin évolutif de n'importe quelle unité de fonctionnalité orientée services.

*Remerciements : j'écris le contenu de cette page pendant le week-end du 21 et 22 novembre 2009 pour le livre « Next Generation SOA » qui était encore en développement à ce moment-là. Je voudrais remercier Prentice Hall pour permettre que ce contenu soit ouvertement publié sur ce site comme supplément au manifeste SOA original. J'aimerais aussi remercier les membres du groupe de travail et ceux de la communauté de SOAPatterns.org, pour avoir apporté leurs commentaires et assistance à ces annotations. De nombreux membres du groupe de travail ont publié leurs propres articles, blogs et documents au sujet du manifeste SOA et je vous encourage tous à les rechercher pour obtenir des commentaires et perceptions additionnels.*

- Thomas Erl

---

## L'équipe de traduction

Yves Chaix

---

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

Copyright © 2009-2011