

SOA : LES IMPACTS SUR LE CYCLE DE VIE DU PROJET DE DÉVELOPPEMENT LOGICIEL

Rim Bejaoui, Ivan Maffezzini
Université du Québec à Montréal
CP 8888 Succ. Centre Ville
Montréal H3C 3P8 Canada
Bejaoui.Rim.2@courrier.uqam.ca
Maffezzini.Ivan@uqam.ca

Mots-clés : architecture axée sur les services, développement axé sur les services, cycle de vie du projet de développement logiciel, rôles.

RESUMÉ

Conceptualisée par le Gartner Group, leader mondial dans la fourniture de recherches et d'analyses dans l'industrie des technologies de l'information (TI), la notion d'architecture axée sur les services (notée SOA pour *Service Oriented Architecture*) est présentée comme le bon modèle pour le développement des applications d'entreprise, aujourd'hui et dans le futur (Marks et Bell, 2006). D'après la littérature, le développement d'applications d'entreprise fondées sur SOA présente de nouveaux défis : apparition de nouvelles activités, de nouveaux rôles et de nouvelles tâches au niveau de l'entreprise ; nécessité d'avoir de nouveaux outils de support au développement ; incapacité des méthodes de développement traditionnelles à soutenir le développement axé sur les services ; etc. Des ajustements doivent donc être apportés au cycle de vie du projet de développement logiciel (CVP) pour pouvoir s'adapter à cette nouvelle façon de construire les applications d'entreprise, et permettre, par conséquent, une gestion de projet plus efficace.

L'objectif de notre recherche est de savoir si la mise en œuvre d'une SOA par une organisation apporte des changements aux *rôles* et aux *tâches* du CVP. Pour atteindre son objectif, notre recherche prend en considération les aspects spécifiques soulignés dans la littérature SOA, ainsi que les résultats d'une étude de cas exploratoire menée auprès de deux organisations ayant entamé une initiative SOA.

INTRODUCTION

SOA est un acronyme qui s'est répandu partout depuis l'année 2002. On le retrouve comme mot clé dans un grand nombre d'articles de journaux scientifiques, comme slogan utilisé par les fournisseurs de solutions logicielles (ex. « *Stop Talking. Start Doing. The Smart SOATM Approach* » d'IBM), comme titre pour des dizaines de livres, comme avantage concurrentiel mis en avant par les entreprises et comme approche choisie par de grands projets gouvernementaux, dont le modèle de référence GC SOA (*Government of Canada SOA*) qui guide l'utilisation et l'adoption de SOA par le gouvernement du Canada.

En fait, l'acronyme SOA couvre deux domaines : l'architecture d'entreprise et l'architecture d'application. SOA est fondée sur la construction de services réutilisables et interopérables,

neutres par rapport à la plateforme de communication et qui correspondent aux processus d'affaires de l'entreprise. Elle peut être vue comme « *une approche utilisée pour normaliser l'architecture logicielle à travers l'entreprise* » (Erl, 2005). L'idée sous-jacente est de mettre au centre l'architecture, car elle risque d'avoir une durée de vie plus longue que les exigences et les besoins qui sont au fondement des applications.

L'adoption de SOA par une entreprise constitue une grande initiative qui peut s'étendre sur plusieurs années. Cette initiative, selon certains auteurs, implique l'élaboration de plusieurs projets pilotes qui ont pour objectif de développer l'infrastructure SOA (Bloomberg et Schmelzer, 2006 ; Krafzig et al., 2004).

Actuellement, tous les travaux dans la littérature SOA que nous avons consultés considèrent que l'adoption de SOA n'est pas triviale, qu'elle présente beaucoup de défis et qu'elle nécessite des changements au sein de l'organisation. La question est de savoir, jusqu'à quel point le choix de SOA, peut influencer la façon avec laquelle l'organisation structure et gère ses projets de développement logiciel.

D'une façon générale, un projet logiciel est défini par l'ensemble des activités techniques et managériales, nécessaires à la satisfaction des termes et des conditions d'un accord de projet qui spécifie, entre autres, l'étendue, les objectifs, le budget et l'échéancier du projet (IEEE 1058-1998). Habituellement, lors de la réalisation d'un projet, les organisations divisent ce dernier en plusieurs phases afin d'améliorer le contrôle de gestion¹ et de fournir un lien entre les différentes opérations (IEEE 1490, 2003). L'ensemble des phases constitue le CVP.

Le CVP définit généralement (IEEE 1490-2003) :

- Le travail technique à effectuer durant chaque phase (ex. est-ce que le travail architectural fait partie de la phase de définition ou de la phase d'exécution ?)
- Les individus impliqués durant chaque phase (ex. quels sont les individus qui doivent être impliqués durant la phase de conception ?)

En d'autres termes, le CVP définit les *activités* et les *rôles* nécessaires à l'accomplissement de chaque phase. Nous considérons qu'*activités* et *rôles* sont très corrélés par la notion de *tâche* (voir Figure 1). En effet, une *activité* est définie par un ensemble de *tâches* coordonnées ; d'autre part, un *rôle* est défini par l'ensemble des *tâches* qui lui sont associées. Ainsi, une façon d'identifier les *activités* à accomplir durant le CVP serait d'identifier les *rôles* impliqués et l'ensemble des *tâches* qui leurs sont associées.

¹ D'une façon générale, le contrôle de gestion est un processus permanent qui permet à une organisation de s'assurer que les ressources qui lui sont confiées sont obtenues et utilisées d'une façon efficace (OQLF, 2007). Dans le cas d'un projet logiciel, le contrôle de gestion devrait être imposé via les tests, les révisions, les audits, la vérification, la validation, etc. (IEEE 12207.0, 1996).

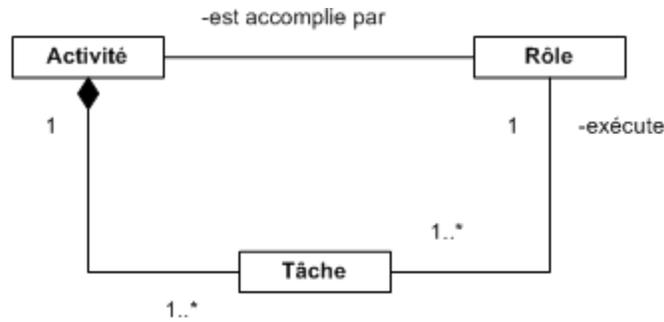


Figure 1 Relation entre *activité*, *tâche* et *rôle*

Le CVP a longtemps été analysé. Au fur et à mesure de l'apparition de nouvelles pratiques de développement, il a été adapté. D'après la littérature SOA, la tendance actuelle vers le développement de systèmes fondés sur SOA nécessite quelques ajustements au CVP. Cependant, on ne présente pas de cas où l'on définit, en détail, les changements qui y sont apportés.

L'objectif de notre recherche est de savoir si le passage vers un environnement axé sur les services pour une organisation demande des changements aux *rôles* et aux *tâches* du CVP. Pour atteindre notre objectif, nous avons pris en considération les aspects spécifiques soulignés dans la littérature SOA, ainsi que les résultats d'une étude de cas par entrevues menée auprès de deux organisations ayant entamé une initiative SOA.

Le reste de l'article est structuré de la façon suivante : la première section présente une revue de la littérature des défis et pratiques du développement logiciel axé sur les services. La deuxième section effectue une synthèse des éléments de la littérature qui sont en relation avec notre objectif de recherche. La troisième section décrit notre étude de cas. La quatrième section présente une discussion concernant les résultats auxquels nous sommes arrivés. Et pour finir, la dernière section constitue une conclusion dans laquelle nous effectuons un survol de notre recherche et nous présentons les possibilités de travaux futurs.

I LE DÉVELOPPEMENT LOGICIEL AXÉ SUR LES SERVICES : DÉFIS ET PRATIQUES

Actuellement, dans la littérature, il est admis que le concept d'« axé sur les services » définit des principes architecturaux qui sont analogues à ceux de la conception logicielle « axée sur les composants » (Ivanyukovich et al., 2005 ; Stojanovic et al., 2005 ; Yukyong et Hongran, 2006). C'est pour cette raison que les méthodes actuellement utilisées pour le développement d'applications SOA sont fondées sur celles utilisées pour le développement de systèmes basés sur les composants. Cependant, certains aspects qui caractérisent les projets de développement dans un environnement axé sur les services (ex. l'aspect des services faiblement couplés, le problème de l'identification des services avec une granularité optimale, etc.) engendrent de nouveaux défis et des différences considérables dans la façon avec laquelle les projets basés sur une SOA sont conduits.

D'après Mittal (2006), les facteurs inhérents au succès de la mise en œuvre d'une SOA sont :

- Un processus de développement clairement défini.
- Une communication améliorée à travers les équipes de projet.
- Des politiques de support et de gouvernance claires.

Nous avons remarqué que le point qui revient le plus souvent dans la littérature en ce qui concerne les défis du développement axé sur les services est la définition d'une méthode ou d'un processus de développement adéquat. Quelles seraient donc les caractéristiques d'une méthode de développement axée sur les services, telles que vues par les recherches actuelles ?

Toutes les méthodes de développement (RUP, XP, dX, etc.) ont leurs forces et faiblesses, et le choix et l'application d'une méthode va dépendre de l'expérience et des meilleures pratiques établies par chaque entreprise. Cependant, en présence d'un environnement axé sur les services, les approches les plus appropriées sont celles qui supportent le développement itératif (Krafzig et al., 2004 ; Stojanovic et Dahanayake, 2005). Les projets de développement de solutions axées sur les services ressemblent, en surface, aux projets de développement d'applications distribuées (Erl, 2005). Cependant, afin de construire et de positionner les services d'une façon convenable dans une solution axée sur les services, les cycles des projets traditionnels nécessitent quelques ajustements. Ces ajustements se reflètent dans les différentes approches de développement axées sur les services proposées par quelques récentes recherches, que nous présentons brièvement selon un ordre chronologique.

- *Service Oriented Analysis and Design* (SOAD) est une approche améliorée et interdisciplinaire de modélisation de service, proposée par O. Zimmermann et al. (2004), partant de processus de développement et de notations existants.
- *Service-Oriented Modeling and Architecture* (SOMA) illustre les activités d'une méthode de modélisation axée sur le service, proposée par Arsanjani (2004). Pour l'identification et la spécification de service, elle combine les trois approches d'analyse ascendante, descendante et *middle-out*.
- Ivanyukovich et al. (2005) proposent l'adoption du processus de développement *Rational Unified Process* (RUP) pour le développement axé sur les services. Mais proposent, par ailleurs, de l'adapter pour qu'il puisse répondre à certaines caractéristiques spécifiques de ce type d'environnement.
- Un modèle conceptuel, présenté par Yukyong et Hongran (2006) et appelé M4SOD (*Method For Service Oriented Development*), a pour but de formaliser le processus de développement SOA. Cette méthode met l'accent sur les phases d'identification et de réalisation des services.
- Rahmani et al. (2006) proposent une approche de modélisation et de conception de systèmes basés sur une SOA qui utilise l'architecture dirigée par les modèles (*Model Driven Architecture*, MDA).

- *Service-Oriented Unified Process* (SOUP), un processus de développement destiné aux systèmes basés sur SOA et proposé par Mittal (2006), reprend les meilleurs éléments de RUP et de XP (*Extreme Programming*).

II ACTIVITÉS, RÔLES ET TÂCHES SOA

Rappelons que l'objectif de notre recherche est de savoir si le passage vers un environnement axé sur les services pour une organisation demande des changements aux *rôles* et aux *tâches* du CVP. La littérature qui a été considérée pour définir les *activités*, *rôles* et *tâches* SOA est la suivante :

- (1) La norme IEEE/EIA std 1074 – 2006 qui donne une méthode pour définir des processus pour le CVP.
- (2) La littérature qui définit les activités, les *rôles* et les *tâches* impliqués dans une initiative SOA et dans le cycle des projets qui sont élaborés dans le cadre de cette initiative (projets SOA).

La norme IEEE/EIA std 1074 classe les activités impliquées dans le CVP de tout projet de développement logiciel sous cinq groupes d'activités : activités de gestion de projet, activités de pré-développement, activités de développement, activités de post-développement et activités de support.

Cette section se présente de la façon suivante. Dans la première sous-section (*Terminologie*), nous définissons la terminologie utilisée dans le reste de l'article. Dans la deuxième sous-section (*Activités SOA*), nous présentons les différentes activités SOA qui ont été identifiées à partir de la littérature, réparties selon la classification proposée par la norme IEEE/EIA std 1074 pour les groupes d'activités du CVP. Dans la troisième sous-section (*Rôles et tâches SOA*), nous décrivons les différents rôles SOA et leurs *tâches* associées, identifiés à partir de la littérature.

II.1 Terminologie

La Figure 2 présente un diagramme UML qui relie les différents éléments de la terminologie.

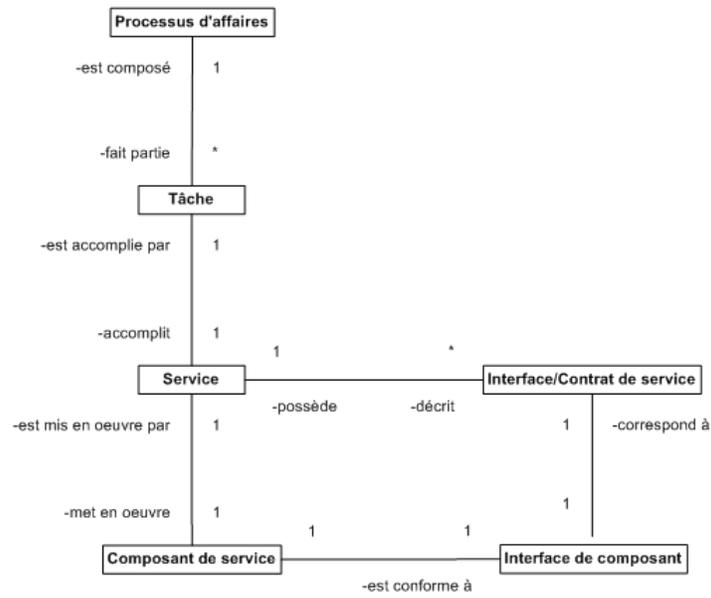


Figure 2 Terminologie

Processus d'affaires : un processus d'affaires peut être vu comme un enchaînement de *tâches* au cours duquel des procédures et des informations sont traitées ou exécutées successivement en vue de réaliser un produit ou de fournir un service (OQLF, 2007).

Tâche : une *tâche* est une unité de travail conceptuelle faisant partie d'un processus d'affaires (Chang et Kim, 2007). Une *tâche* peut être décomposée en des tâches de plus faible granularité (sous-tâches).

Service : il s'agit d'un élément logiciel dont l'exécution accomplit une tâche, de n'importe quelle granularité, faisant partie d'un processus d'affaires (Chang et Kim, 2007). La différence entre une tâche et un service est que la tâche est une unité conceptuelle définie selon une perspective d'affaires, et qu'un service est l'activité qui lui correspond définie selon une perspective d'ingénierie logicielle. Quelques services peuvent être communs à différents processus d'affaires. Ils sont donc réutilisables à travers plusieurs processus d'affaires.

Interface de service/Contrat de service : l'interface (ou contrat) de service décrit le service et les fonctionnalités offertes par ce dernier. Elle contient toute l'information utile à l'utilisation du service (pré-conditions, post-conditions, données en entrée, etc.) et sert à garder la mise en œuvre du service hors de portée des clients du service (Newcomer et Lomow, 2004). Elle peut, par exemple, être décrite avec CORBA IDL, WSDL, ou un format XML *ad hoc* (Krafzig et al., 2004)

Composant de service : élément logiciel qui met en œuvre un service (Chang et Kim, 2007). Dans le paradigme de l'axé sur les services, on distingue clairement les deux niveaux logique et physique : le service est le concept du niveau logique, le composant est le concept physique de mise en œuvre (Bonnet, 2005). Ainsi, à la différence du service, le composant de service représente une entité logicielle physique et exécutable. Certains composants de services sont similaires à ceux du développement basé sur les composants (par exemple, dans l'infrastructure

J2EE, un composant est un EJB, un JavaBeans, une Servlet ou une RMI) ; d'autres composants de service peuvent simplement être des classes enveloppantes (*wrappers*) d'applications héritées (Bonnet, 2005). Ces deux types de composants sont mis en œuvre d'une façon différente, mais tous les deux doivent fournir des interfaces physiques, appelées interfaces de composants.

Interface de composant : les interfaces de composants sont conformes aux interfaces de services (Chang et Kim, 2007). Par exemple, les composants de service peuvent être mis en œuvre en tant qu'EJB et fournissent des interfaces physiques sous la forme d'interfaces EJB *Home* ou *Remote* conformes aux interfaces de services WSDL (Chang et Kim, 2007).

Tel qu'illustré par le diagramme de la Figure 2, un *processus d'affaires* est composé d'une ou plusieurs *tâches*, chacune étant accomplie par un *service*. Chaque *service* est mis en œuvre par un *composant de service* conforme à une *interface de composant*, et est décrit à l'aide d'une *interface/contrat de service*, qui elle-même correspond à une *interface de composant*.

II.2 Activités SOA

La figure 3 présente l'ensemble des activités SOA que nous avons identifié à partir de la littérature. Les activités sont réparties selon les cinq groupes d'activités de la norme IEEE/EIA std 1074.

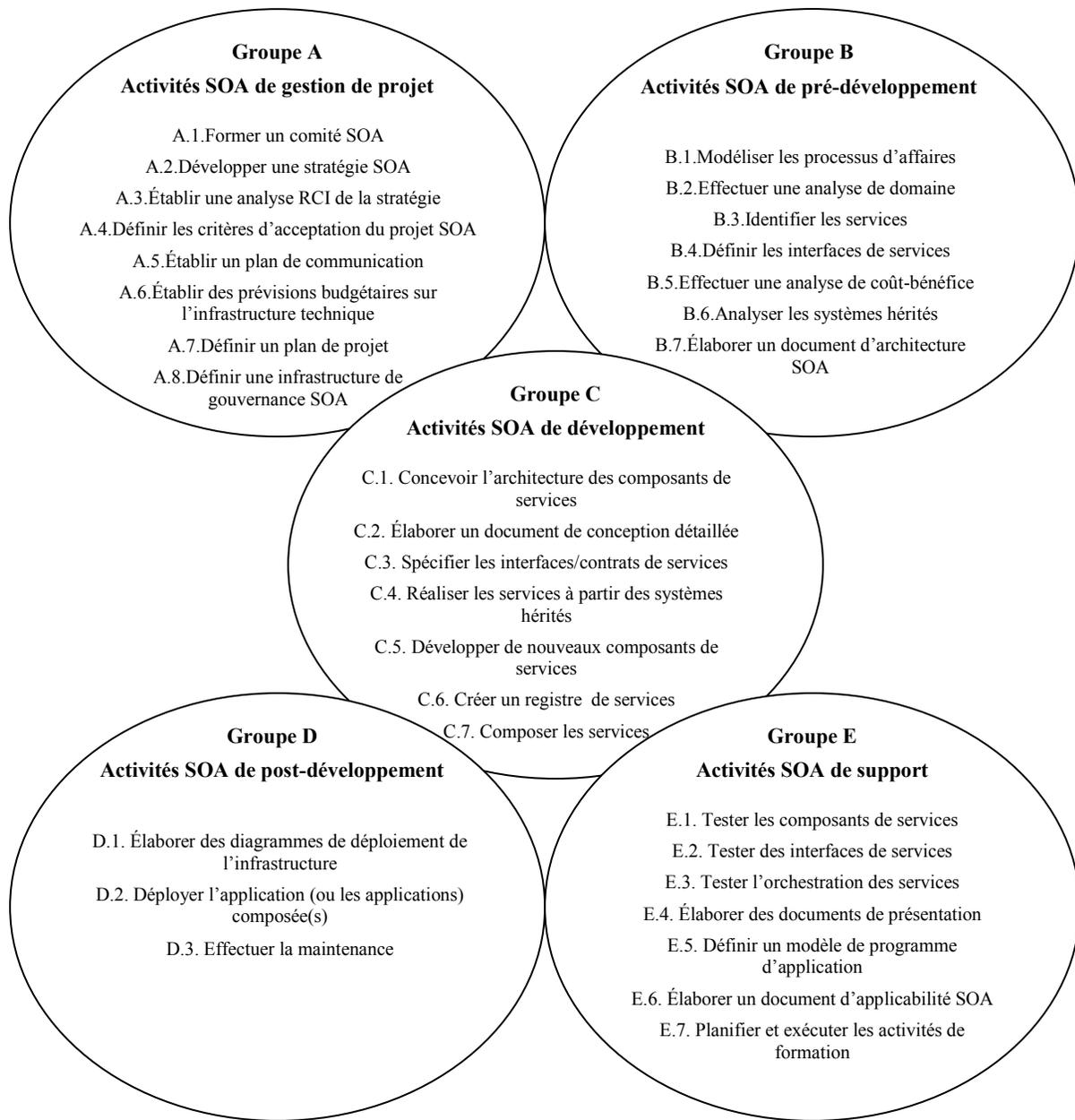


Figure 3 Activités SOA réparties selon les groupes d'activités de la norme IEEE/EIA std 1074

II.3 Rôles et tâches SOA

En partant de la littérature analysée, nous avons effectué une synthèse des *rôles* impliqués dans une initiative SOA et des *tâches* qui leurs sont associées. Une activité étant un ensemble de *tâches*, comprendre les activités à accomplir durant un projet SOA nous a permis de mieux appréhender les *tâches* et de faciliter ainsi leur assignation aux *rôles* identifiés.

Parmi les *rôles* cités dans la littérature, certains sont plutôt des activités camouflées en *rôles* (ex. gestionnaire de gouvernance SOA, gestionnaire de mesures SOA, etc.). Ces *rôles* n'ont pas été retenus dans notre synthèse. Les *rôles* SOA retenus sont au nombre de sept (7) : Gestionnaire de projet, Architecte d'entreprise, Architecte SOA, Analyste d'affaires, Concepteur de services, Développeur de services et Administrateur du registre des services. Un groupe de rôles a de même été retenu en raison de son importance. Il s'agit du Comité SOA.

La figure 4 présente les *tâches* qui sont attribuées à chacun des sept *rôles* retenus, ainsi qu'au groupe de rôles du comité SOA, selon notre analyse de la littérature.

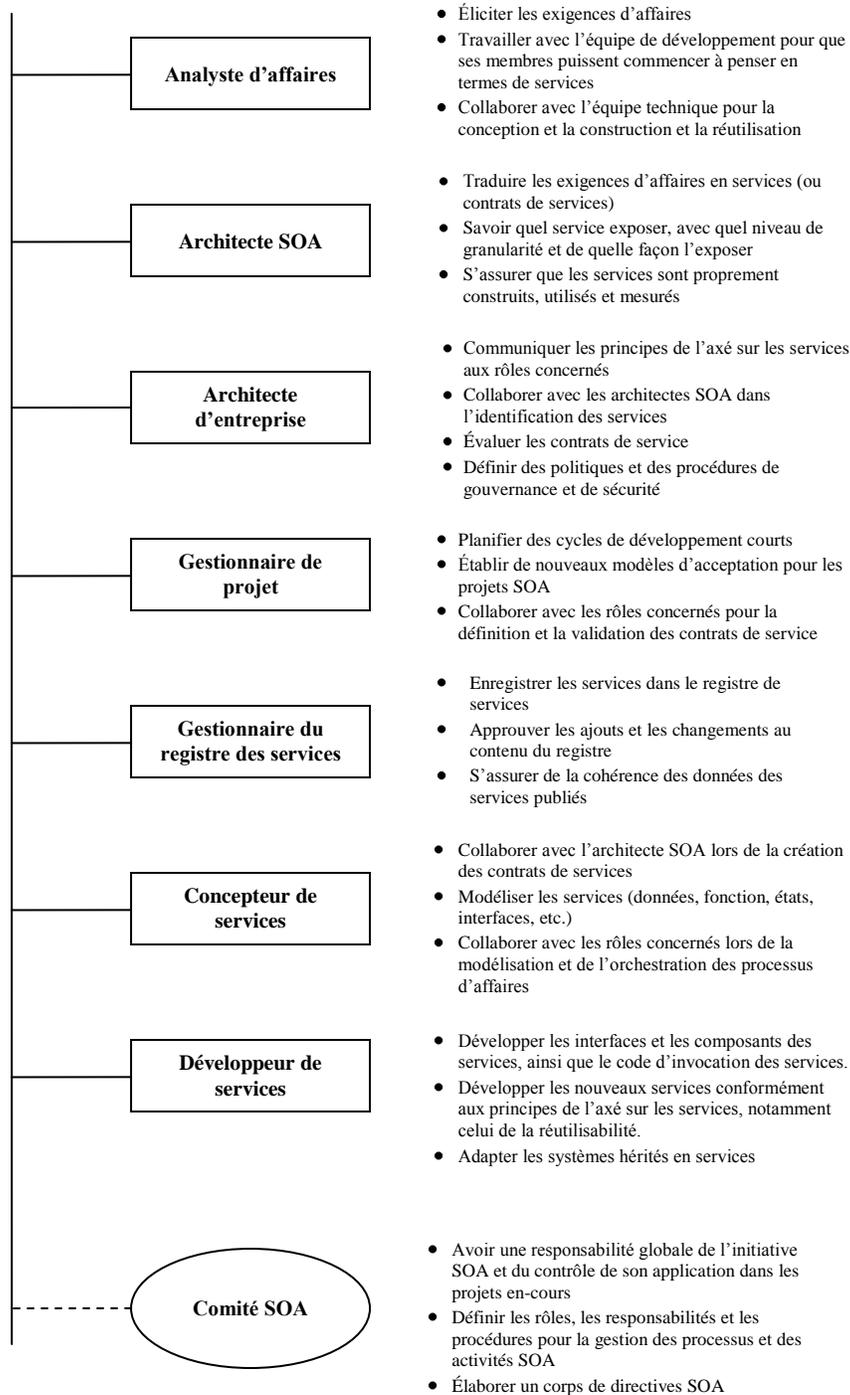


Figure 4 Rôles et tâches SOA

III ÉTUDE DE CAS

Notre recherche suit une approche positiviste de type exploratoire et n'a donc aucune prétention d'apporter des preuves. Notre étude devrait permettre d'approfondir certaines hypothèses (ex. l'importance des nouveaux *rôles* dans le développement axé sur les services), hypothèses qui

devraient être étudiées et insérées dans un contexte théorique plus complet dans des travaux futurs. Qu'elle soit basée sur un ou plusieurs cas, une étude de cas exploratoire a pour objectif de définir des questions, des construits, des propositions ou des hypothèses qui feront l'objet d'une étude empirique subséquente (Yin, 1993). Dans notre recherche nous avons considéré deux cas d'étude.

III.1 Sélection des cas

À présent, beaucoup d'entreprises sont en train d'adopter SOA. Peu importe le domaine d'activité des organisations que nous avons choisi pour mener notre étude – qu'elles soient des entreprises manufacturières, des institutions financières ou des entreprises spécialisées dans un tout autre domaine – l'essentiel c'est que les organisations en question aient déjà entamé une initiative SOA. La sélection de nos cas d'étude a de même dépendu des ressources qui ont pu nous être accordées pour mener notre investigation (nombre de répondants, nombre d'entrevues accordées, nombre de sources de preuves qualitatives, etc.), ainsi que de la grande différence entre les environnements axés sur les services des deux cas choisis.

Le premier cas présente un environnement complètement centralisée, où une seule organisation est responsable du développement des différents composants de l'architecture SOA. Il s'agit d'une institution financière, dénommée organisation X, qui a entamé son initiative SOA depuis trois années. L'objectif de la mise en place d'une SOA dans cette institution financière est principalement de permettre de livrer des fonctionnalités qui pourront être réutilisées de projets en projets.

Le deuxième cas présente une initiative de mise en œuvre d'une SOA dans un environnement axé sur les services décentralisé, où certaines organisations constituent des fournisseurs de services, d'autres sont des demandeurs de services et d'autres sont chargées de la mise en œuvre de l'infrastructure SOA. Ce deuxième cas est celui du Dossier de Santé du Québec (DSQ). Le DSQ est un projet pilote pour l'initiative du Dossier de Santé Électronique (DSÉ) – une initiative d'instauration d'un système de santé interopérable à travers tout le Canada. La solution proposée pour le DSÉ, nommée « infostructure du DSÉ » (iDSÉ) a été conçue en se basant sur une architecture SOA.

III.2 Aperçu du projet de l'étude de cas

Les questions d'investigation de notre étude de cas ont eu pour finalité de :

1. Vérifier si les changements (dans les *rôles* et les *tâches* du CVP) identifiés à partir de la littérature sont présents dans la pratique des entreprises étudiées.
2. Vérifier si des changements introduits dans les entreprises étudiées n'ont pas été considérés dans la littérature que nous avons analysé.
3. Clarifier les aspects qui n'ont pas été assez investigués par les recherches.

III.3 Méthode de collecte des données

La recherche par étude des cas combine plusieurs méthodes de collecte de données. Collecter différents types de données par différentes méthodes, de plusieurs sources, produit une couverture plus importante et peut résulter en une vision plus étendue sur le phénomène étudié (Bonoma, 1985 ; Paré, 2004). Yin (2003) identifie diverses sources de preuves qualitatives, parmi lesquelles la documentation, les documents d'archives, les entrevues, l'observation directe et les artefacts physiques.

Durant notre étude, nous avons essayé d'avoir accès au maximum des sources d'information qui viennent d'être citées. Cependant, les entrevues, la documentation et les questionnaires élaborés pour nos répondants ont constitué notre principale méthode de collecte de données.

Lors de l'analyse de la littérature, nous avons remarqué que ce sont les *rôles* des gestionnaires de projets et des architectes qui subissent le plus de changements et qui se voient accorder les *tâches* les plus importantes lors d'une initiative SOA. De plus, ce sont les individus responsables de ces *rôles* qui possèdent le plus d'autorité et qui prennent les décisions pouvant avoir le plus d'impact sur le déroulement du projet et qui, par conséquent, sont les plus aptes à nous communiquer l'information recherchée. C'est pour cette raison que les répondants sollicités pour nos deux cas d'étude sont responsables de ces *rôles*.

IV DISCUSSION

IV.1 Littérature

L'abondance de la littérature concernant SOA montre l'importance du phénomène et l'intérêt suscité. Elle a constitué une bonne base pour notre travail sur l'étude de cas, en nous introduisant au « jargon » utilisé pour décrire les pratiques reliées au développement axé sur les services et en nous permettant de dégager les *rôles* et les *tâches* à vérifier dans nos deux cas d'étude. Cependant, nous avons été confrontés à beaucoup d'incohérences dans le contenu proposé par certains livres et articles publiés. Ceci pourrait s'expliquer par le manque de normalisation de la terminologie SOA ou par la nouveauté du concept.

Dans la littérature, il existe une conceptualisation cohérente des *rôles* et des *tâches* caractérisant le CVP dans un environnement axé sur les services, mais qui n'a cependant pas été facile à identifier. En effet, dégager un ensemble de *rôles* et de *tâches* SOA que nous avons jugé cohérent a nécessité une analyse des activités à accomplir durant un projet SOA et l'élimination de certaines activités camouflées en *rôles*. Les changements aux *rôles* et aux *tâches* associées aux *rôles* sont présentés dans la section (voir section Activités, rôles et tâches SOA).

IV.2 Étude de cas

a) Cas 1

Le constat le plus important que nous avons fait concernant le cas 1 est qu'il existe une forte correspondance entre les *rôles* et les *tâches* identifiés à partir de la littérature et ceux présents dans le premier cas. Tous les *rôles* SOA identifiés à partir de la littérature avec leurs *tâches* associées ont été retrouvés dans le cas 1. Seuls deux rôles (architecte de solutions et architecte de données) sont présents dans le cas 1 mais n'ont pas été abordés par la littérature.

Nous pouvons affirmer que, dans le cas 1, les changements aux *rôles* et aux *tâches* associées aux *rôles* sont les mêmes que ceux identifiés à partir de la littérature.

Un changement introduit par SOA qui a été constaté dans les pratiques de développement du cas 1, mais qui n'a cependant pas été abordé dans la littérature, se situe au niveau des procédures de tests et d'assurance qualité. Voici comment l'équipe de développement de l'organisation X procède, pour chaque nouveau service identifié :

1. Modélisation détaillée du service.
2. Création de « *stub* » du service².
3. Création d'un jeu de tests unique, qui sera enrichi et approuvé par le bureau d'assurance qualité.
4. Mise en œuvre du service et automatisation des tests.

Cette méthode apporte plusieurs avantages à l'organisation X. D'abord, elle génère moins d'anomalies, étant donné que les tests sont conçus avant le développement et que les impacts d'un changement dans un module sur un autre sont détectés dans l'exécution des tests. Ensuite, grâce à cette méthode, le groupe d'assurance qualité peut orienter ses efforts sur la validation et l'analyse des tests, non sur l'exécution de ces derniers. Tous ces avantages offrent une économie d'effort à plus long terme.

Un autre changement introduit par la mise en œuvre de SOA pour le cas 1 est celui de l'adoption d'un nouveau processus de développement logiciel. Il s'agit de *Scrum* ; une nouvelle méthode qui est supposée offrir plus d'agilité et des livraisons plus rapides.

b) Cas 2

Le projet DSQ inclut une équipe d'intégration chargée d'intégrer la solution de l'iDSÉ avant de la déployer en région. Le groupe d'intégration a de même la responsabilité de définir les règles de gouvernance au niveau de l'architecture. Ici aussi, il y a concordance avec la littérature qui recommande l'instauration d'un tel groupe jouant le rôle du comité SOA, spécialement dans les initiatives SOA à grande envergure

Une des pratiques de la gouvernance SOA intéressantes à citer pour le cas du projet DSQ est l'utilisation d'une méthode de documentation unifiée. En effet, tous les projets du DSQ utilisent *Macroscope*, une méthode de documentation (développée par la firme DMR) qui a été adoptée comme norme par le ministère de la santé du Canada et qui, dans le cadre du DSQ, permet de normaliser la documentation des différents projets.

Un autre constat important pour le cas 2 – qui n'a cependant pas été soulevé par la littérature SOA – est que des méthodes de développement différentes peuvent être utilisées pour les

² En se basant sur la modélisation détaillée du service (en WSDL) et grâce à l'introduction d'un nouvel outil (LISA), l'équipe de développement peut simuler le service, avant sa mise en œuvre, en générant un « Stub de service ».

différents projets qui sont impliqués dans une initiative SOA ; mais, ce qui importe le plus, c'est d'avoir une méthode commune pour la documentation de tous les projets. Tel est le cas des projets du DSQ. Cependant, la méthode de développement la plus utilisée par ces derniers est *Unified Process* (UP) ; ce qui confirme les dires de certains auteurs de la littérature SOA, qui voient en UP un des meilleurs candidats pour le développement axé sur les services.

CONCLUSION

Dans ce travail de recherche, nous nous sommes intéressés à la pratique SOA en adoptant un point de vue théorique. Ce que l'on a remarqué, c'est que les considérations théoriques ne sont pas si loin de la réalité. Le jargon utilisé par les répondants des deux cas d'étude, ainsi que leurs pratiques sont les mêmes que ceux introduits par les auteurs de la littérature SOA.

Nous voyons en SOA une nouvelle façon de nommer les nouvelles pratiques qui sont entrain de façonner les environnements TI actuels, suite à de nombreuses influences, parmi lesquelles :

- La volonté d'intégrer les informations de différentes sources qui crée un souci d'interopérabilité. Interopérabilité qui est de même née de la volonté des entreprises et des organisations de se découpler de vendeurs spécifiques.
- La prédominance d'Internet et de ses technologies (XML, services Web, etc.) dans les échanges inter-organisationnels et entre organisations et clients – qui a aussi engendré le passage du modèle économique de la chaîne de valeurs à celui du réseau de valeurs. Désormais, pour l'entreprise, la fourniture de valeur ajoutée dépend plus que jamais des sous-traitants, fournisseurs et partenaires, rendant ainsi le modèle de la chaîne de valeurs obsolète.

En particulier, dans cette recherche, les nouvelles pratiques auxquelles nous nous sommes intéressées sont celles des *rôles* et des *tâches* associées aux *rôles* dans les environnements axés sur les services. Deux environnements axés sur les services ont été considérés : un environnement centralisé et un environnement décentralisé. Le constat majeur que nous avons fait est que les *rôles* ne sont pas tellement différents de ceux retrouvés dans les autres types d'environnements. Même si les appellations changent de peu, les *rôles* de l'ingénierie logicielle³ (gestionnaire de projet, architecte, analyste, etc.) sont toujours là, à la différence que ces derniers sont introduits à de nouvelles *tâches* qui sont en relation avec les principes SOA (identification des services selon une granularité optimale, définition des contrats de services, etc.).

Étant donné le manque de validation de la conclusion que nous proposons, il serait intéressant, dans un travail futur, d'étendre notre recherche à plus de deux cas afin de pouvoir, éventuellement, tirer des conclusions plus générales concernant les impacts de SOA sur le CVP.

³ Même si, jusque-là aucun consensus n'existe en ce qui concerne les *rôles* et les *tâches* à inclure dans le cycle de vie du projet, nous allons considérer que c'est le cas. En effet, juger de la validité des *rôles* du génie logiciel et des *tâches* qui leurs sont associées n'est pas du ressort de cette recherche.

Références

- Arsanjani A. (2004). « Service-oriented modeling and architecture: how to identify, specify, and realize services for your SOA ». IBM whitepaper. URL : <ftp://www6.software.ibm.com/software/developer/library/ws-soa-design1.pdf>
- Bloomberg J. et R. Schmelzer (2006). « Service Orient or Be Doomed! ». Wiley Editions, 258 pages.
- Bonnet P. (2005). « Cadre de référence Architecture SOA - partie1 ». Orchestra Networks. URL : [http://pie.bonnet.ifrance.com/ON-guideSOA-2005-02-23%20\(part1\).pdf](http://pie.bonnet.ifrance.com/ON-guideSOA-2005-02-23%20(part1).pdf)
- Bonoma T. V. (1985). « Case research in marketing: Opportunities, problems, and a process ». Journal of Marketing Research, vol. 22, 199-208.
- Chang S. H. et Kim S. D. (2007) « A Systematic Approach to Service-Oriented Analysis and Design ». Springer-Verlag Berlin Heidelberg.
- Erl T. (2005). « Service-Oriented Architecture: Concepts Technology and Design ». Prentice Hall.
- IEEE 1058-1998. « IEEE Standard for Software Project Management Plans ». Institute of Electrical and Electronics Engineers.
- IEEE/EIA 12207.0-1996. « Standard for Information Technology—Software life cycle processes ». Institute of Electrical and Electronics Engineers.
- IEEE 1490. « A Guide to the Project Management Body of Knowledge ». Institute of Electrical and Electronics Engineers.
- Ivanyukovich A., G. R. Gangadharan, V. D’Andrea, M. Marchese (2005). « Towards a service-oriented development methodology ». Journal of integrated design & process science. Vol. 9, No. 3, pp. 53-62.
- Krafzig D., K. Banke et D. Slama (2004). « Enterprise SOA: Service-Oriented Architecture Best Practices ». Prentice Hall PTR.
- Marks E. A. et M. Bell (2006). « Service-Oriented Architecture: A planning and Implementation Guide for Business and Technology ». Wiley Editions, 375 pages.
- Mittal K. (2006). « Build your SOA, Part 3: The Service-Oriented Unified Process ». IBM developerWorks. URL : www.ibm.com/developerworks/library/ws-soa-method1.html
- Mittal K. (2006b). « Create the Ideal SOA Team ». IBM developerWorks. URL : <http://www.ibm.com/developerworks/library/ar-soateam/index.html>
- Newcomer E. et G. Lomow (2004). « Understanding SOA with Web Services ». Independent Technology Guides. Addison Wesley, 444 pages.
- Office Québécois de la Langue Française (OQLF). Dernière consultation : 14 Avril 2008. URL : <http://www.granddictionnaire.org>
- Paré G. (2004). « Investigating Information Systems with Positivist Case Study Research ». Communications of the AIS, 13, 2004, pp.231-264.
- Schmelzer R. (2008). « WOA is Me - Another Acronym? WOA and SOA ». ZapFlash Document. URL : <http://www.zapthink.com/report.html?id=ZAPFLASH-2008516>
- Stojanovic Z. et A. Dahanayake (2005). « Service-Oriented Software System Engineering: challenges and practices ». Idea Group Publishing, pp. 27-47.
- Yin R. K. (2003). « Case Study Research: Design and Methods ». SAGE Publications, pp.57-97.

Yukyong K. et Y. Hongran (2006). « An Approach to Modeling Service-Oriented Development Process ». IEEE International Conference on Services Computing (SCC'06), pp. 273-276.

Zimmermann O., P. Krogdahl et C. Gee (2004). « Elements of Service-Oriented Analysis and Design. An interdisciplinary modeling approach for SOA projects ». IBM DeveloperWorks. URL : <http://www.ibm.com/developerworks/library/ws-soad1/>