



Le génie pour l'industrie

RAPPORT TECHNIQUE
PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
DANS LE CADRE DU COURS LOG792 PROJET DE FIN D'ÉTUDES EN GÉNIE
LOGICIEL

DÉPLOIEMENT AUTOMATISÉ POUR LA MAINTENANCE DE PROJET

LOUIS BORO
BORL01018905

DÉPARTEMENT DE GÉNIE LOGICIEL ET DES TI

Professeur-superviseur

Alain April

MONTRÉAL, 23 AVRIL 2014
HIVER 2014

DÉPLOIEMENT AUTOMATISÉ POUR LA MAINTENANCE DE PROJET

**LOUIS BORO
BORL01018905**

RÉSUMÉ

Ce rapport présente un projet entrepris dans le but d'automatiser les déploiements d'artéfacts logiciel issus de la maintenance de projets. Le déploiement de changement logiciel est un processus à risque pour les erreurs humaines. L'automatisation permet d'éliminer ces erreurs et d'améliorer globalement la qualité des logiciels livrés.

Ce rapport présente le contexte de l'entreprise dans lequel le projet a eu lieu afin d'aider le lecteur à comprendre les décisions prises. Ce rapport présente aussi le processus de sélection d'un outil commercial pour aider à automatiser les déploiements.

InRelease est un produit de déploiement automatisé qui a été choisi par l'entreprise. Un survol des fonctionnalités du produit est effectué afin de démontrer les avantages et les inconvénients d'utiliser un logiciel commercial de déploiement automatisé. Par la suite, ce rapport présente l'implémentation et l'intégration du produit aux procédures existantes de l'entreprise.

Ce rapport présente aussi la méthodologie utilisée afin de standardiser le déploiement des différentes technologies de l'entreprise. Cet élément est un aspect clé de la réussite du projet.

Finalement, ce rapport offre une rétrospective dans le but de déterminer les facteurs de succès et les défis rencontrés lors du projet.

TABLE DES MATIÈRES

INTRODUCTION	1
CHAPITRE 1 ANALYSE DES DÉPLOIEMENTS MANUELS	2
1.1 Éléments importants.....	2
1.2 Particularités de l'entreprise	2
1.3 Vue globale du processus.....	3
1.4 Problèmes rencontrés	5
CHAPITRE 2 PROPOSITIONS DE SOLUTIONS	7
2.1 Viabilité d'une solution.....	7
2.2 Les déploiements automatisés.....	7
2.3 Les outils de déploiement automatisés	9
2.4 Sélection d'un outil de déploiement automatisé	9
CHAPITRE 3 INRELEASE	11
3.1 Aperçu du produit	11
3.2 Environnement virtuels.....	13
3.3 Principe d'utilisation.....	14
3.4 Intégration aux processus de gestion de changements.....	15
CHAPITRE 4 STANDARDISATION DES DÉPLOIEMENTS	18
4.1 Sources d'informations	18
4.2 Approche par package de déploiement	19
4.3 Approche agile	21
4.4 Modèle de permissions	22
CHAPITRE 5 RÉTROSPECTIVE	24
5.1 Facteurs de succès.....	24
5.2 Défis.....	24
CONCLUSION	25
BIBLIOGRAPHIE.....	26

LISTE DES TABLEAUX

Page

Tableau 1 : Décomposition d'une activité de déploiement	19
---	----

LISTE DES FIGURES

	Page
Figure 1 : Environnements de l'entreprise	3
Figure 2 : Cycle de vie des artéfacts logiciel	4
Figure 3 : Création d'un « Release Template »	11
Figure 4: Création d'un build à partir de Visual Studio	12
Figure 5 : Environnement virtuel isolé	13
Figure 6 : Création d'un « Release Path »	17

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

DBA : Database Administrator.

SQL : Structured Query Language.

ITIL : Information Technology Infrastructure Library.

CDA : Continuous Delivery Automation.

INTRODUCTION

L'automatisation est un sujet populaire dans les départements d'informatique. En effet, l'automatisation permet d'éliminer les erreurs qui découlent des manipulations humaines. De plus, l'automatisation permet d'alléger la charge de travail des employés, ce qui leur permet de travailler sur des activités qui ont une valeur ajoutée pour l'entreprise. Pour ces raisons, les grandes entreprises sont prêtes à investir beaucoup d'effort à l'automatisation.

Dans plusieurs entreprises, le déploiement de maintenances applicatives est encore effectué manuellement à l'aide d'un plan de déploiement. Ce plan contient la marche à suivre pour déployer et configurer les artefacts logiciels issus d'une demande de changement.

Ce processus de déploiement est rigoureux et les risques d'erreurs humaines sont élevés. Lorsqu'une erreur survient lors d'un déploiement, il faut mobiliser plusieurs experts pour identifier la cause de l'erreur et régler le problème.

Le présent projet présente l'initiative d'automatisation d'une firme d'investissements dans le but d'automatiser le déploiement de leurs changements logiciels. Ce projet a été entrepris à l'été 2013 et a permis de répondre au besoin de l'entreprise depuis sa mise en production. Ce rapport présente le projet dans son contexte ainsi que les différentes étapes qui ont mené le projet au succès.

Afin de préserver le secret professionnel de l'entreprise instigatrice du projet, elle n'est pas nommée et elle est référencée formellement en tant que « l'entreprise ».

CHAPITRE 1

ANALYSE DES DÉPLOIEMENTS MANUELS

1.1 Éléments importants

Avant d'automatiser un processus, il est important de bien le comprendre. L'étude des éléments suivants aide à mieux comprendre le processus de déploiement manuel :

- Particularités de l'entreprise
- Vue globale du processus
- Acteurs impliqués dans le processus
- Problèmes rencontrés

1.2 Particularités de l'entreprise

L'entreprise compte plus de 450 employés et opère dans le secteur de l'investissement. Le département d'informatique contient, à lui seul, plus de la moitié des employés. Les clients des services informatiques sont tous des clients internes. Les applications sont donc développées et exploitées à l'interne.

Le parc informatique de l'entreprise utilise les technologies Microsoft. Windows Server, SharePoint, BizTalk, SQL Server, C# et Power Shell. Ce sont des produits critiques pour l'entreprise. Ces produits constituent la majorité des produits commerciaux utilisés au niveau opérationnel.

L'entreprise contient plusieurs équipes de développement qui travaillent sur des produits différents. Toutes les équipes doivent passer par le même processus lors de la livraison de nouveaux logiciels ou lors de la livraison de correctifs de maintenance. Ce processus consiste à soumettre les artefacts logiciels et un plan de déploiement à l'équipe de production.

L'équipe de production est mandatée au bon fonctionnement des opérations et c'est elle qui effectue le déploiement des nouveaux logiciels ainsi que le déploiement de correctifs applicatifs.

Chaque équipe de développement a un intégrateur qui est responsable de la livraison. Les intégrateurs sont les points d'interaction entre les équipes de développement et l'équipe de production.

Chaque livraison en production doit obligatoirement passer par la chaîne suivante :

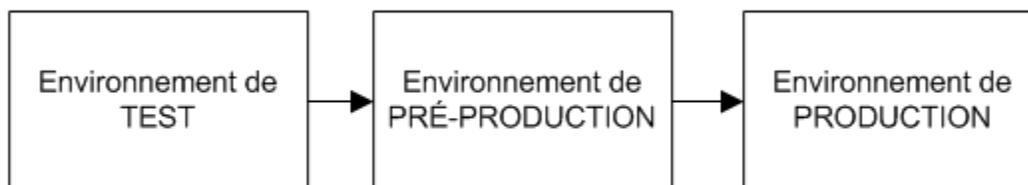


Figure 1 : Environnements de l'entreprise

L'environnement de test et l'environnement de pré-production sont virtuels et ils représentent chacun une copie de la production.

Chaque équipe de développement possède son propre environnement de test. L'intégrateur de l'équipe est responsable de le maintenir semblable à la production. Il existe un seul environnement de pré-production et il est sous la responsabilité de l'équipe de production.

Dû à cette réalité, chaque livraison comprend, au minimum, trois déploiements selon la même procédure.

1.3 Vue globale du processus

Voici une vue à haut niveau des activités impliquées dans la mise en production de correctifs d'application suite à une opération de maintenance sur le logiciel. Cette vue représente un chemin optimiste (sans erreurs).

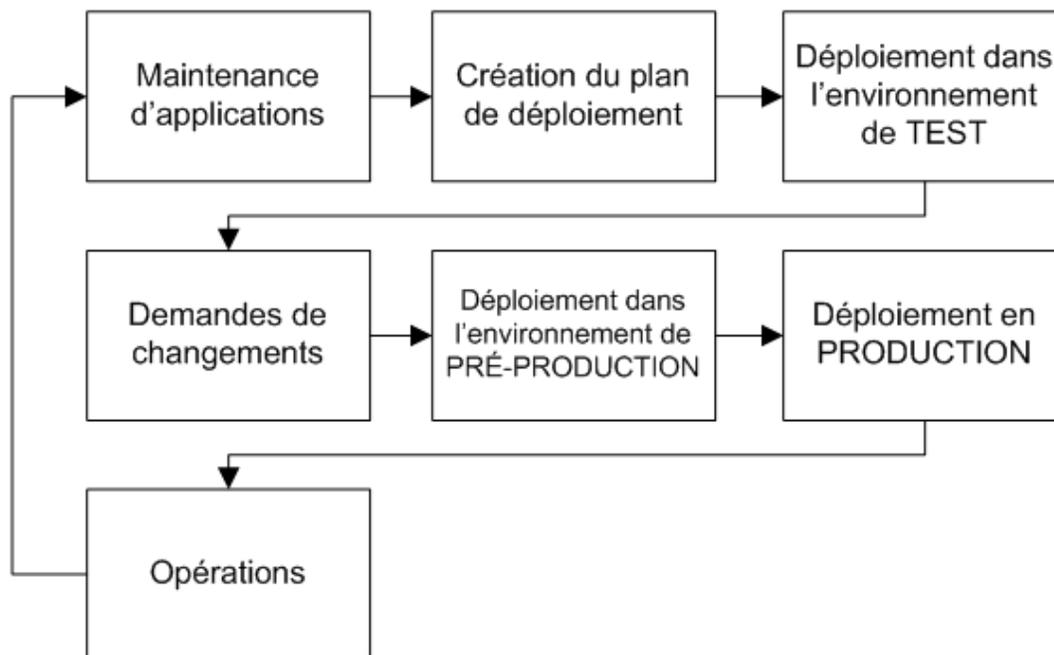


Figure 2 : Cycle de vie des artefacts logiciel

Développement / Maintenance d'applications : Cette activité est effectuée par les développeurs et consiste à développer les artefacts logiciels nécessaires à la création ou à la maintenance d'applications.

Création du plan de déploiement : L'intégrateur est chargé de créer un document qui détaille les étapes du déploiement. Les manipulations consistent principalement à déplacer des fichiers entre les serveurs, à rouler des scripts et à changer des configurations.

Déploiement dans l'environnement de test : L'intégrateur suit le plan de déploiement pour déployer les correctifs dans l'environnement de test. Cette étape sert principalement à valider le bon fonctionnement du plan de déploiement. Aussi, cette étape permet de tester l'intégration des nouveaux artefacts avec les autres systèmes de l'entreprise.

Demandes de changements : L'intégrateur crée les billets de demande de changements pour le déploiement en pré-production et en production. Le plan de déploiement est soumis pour révision et pour approbation par l'équipe de production. Cette étape permet de planifier la date des changements et de réserver les ressources de l'équipe de production.

Déploiement dans l'environnement de pré-production : L'équipe de production suit le plan de déploiement et déploie les correctifs dans un environnement semblable à la production. Cette étape est la validation finale du plan de déploiement et permet de tester la fonctionnalité des nouveaux artefacts logiciels. En théorie, un déploiement avec succès en pré-production assure un déploiement avec succès en production.

Déploiement en production : L'équipe de production suit le plan de déploiement et déploie les correctifs en production.

Opérations : Le logiciel est maintenant utilisé par l'entreprise. L'utilisation à long terme du logiciel mène à la détection de nouveaux problèmes et/ou de nouveaux besoins. Ceux-ci sont transmis à l'équipe de développement et le cycle recommence.

1.4 Problèmes rencontrés

Le déploiement manuel des changements est un processus rigoureux qui est hautement sujet aux erreurs. Voici les problèmes les plus fréquemment rencontrés par l'entreprise :

- L'équipe de production n'utilise pas la dernière version du plan de déploiement
- L'équipe de production n'utilise pas la dernière version des artefacts logiciels
- Une étape est sautée dans le plan de déploiement
- Le déploiement est effectué sur le mauvais serveur

La source d'erreur principale est l'erreur humaine. De plus, en déployant plusieurs fois, le risque de commettre des erreurs est multiplié. Par conséquent, les déploiements sont

inconsistants et il est impossible d'assurer que tout est déployé de manière identique d'un environnement à l'autre.

Le plan de déploiement est conçu par l'équipe de développement et il est exécuté par l'équipe de production. Ainsi, lors d'une erreur dans la procédure de déploiement, la personne qui fait le déploiement n'a souvent pas l'information requise pour régler le problème. Elle doit entrer en contact, dans les délais les plus brefs possible, avec l'intégrateur et/ou les développeurs afin de compléter le déploiement. Les déploiements sont donc une source de risque et de stress important pour l'entreprise.

Une étude de cas sur le « Release Management » est proposée par [LAH 11]. Parmi les points soulevés, il y en a un qui est aussi un problème dans l'entreprise, soit un rythme trop élevé de déploiement. En effet, avec l'avènement de la méthodologie agile, les déploiements sont de plus en plus fréquents et les activités de « Release Management » deviennent le point de contention dans la livraison de nouvelles fonctionnalités. Les artefacts sont développés plus rapidement qu'ils sont déployés et ceci entraîne une désynchronisation entre les équipes de développement et les équipes de production. Cette désynchronisation rend le support à la production plus difficile car les artefacts à problème ont été développés il y a longtemps.

CHAPITRE 2

PROPOSITIONS DE SOLUTIONS

2.1 Viabilité d'une solution

Les problèmes issus des déploiements manuels sont clairement identifiés. Il existe plusieurs outils commerciaux ainsi que plusieurs pratiques de développement pour améliorer la situation.

L'amélioration du processus de déploiement peut être perçue comme une activité d'amélioration au niveau de l'assurance qualité. Le coût de la solution fait donc partie du coût de qualité pour l'entreprise tel que décrit dans [AQL 11].

L'entreprise œuvre dans le secteur de l'investissement et son rendement est déterminé par sa rentabilité. L'entreprise ne travaille ni sur des logiciels critiques, ni sur des logiciels utilisés par le public. Ses efforts de développement sont entièrement destinés aux logiciels internes. Ainsi, les bénéficiaires d'une augmentation de l'assurance qualité ne sont nuls autres que l'entreprise elle-même. Dans cet ordre d'idée, pour qu'un projet d'amélioration d'assurance qualité soit accepté, il doit nécessairement promettre d'améliorer la rentabilité de l'entreprise.

En d'autres mots, pour qu'une solution au problème des déploiements manuels soit viable, elle doit, à long terme, réduire les coûts liés à ces activités. Dans cet ordre d'idée, une solution trop complexe et trop coûteuse peut être moins avantageuse pour l'entreprise que de continuer de déployer manuellement.

2.2 Les déploiements automatisés

Parmi les solutions envisageables pour améliorer les problèmes issus des déploiements manuels, la solution des déploiements automatisés est la plus populaire. Voici les principales raisons.

Premièrement, en automatisant le processus de déploiement, l'entreprise est forcée de standardiser son processus de déploiement. La standardisation permet de détecter et d'adresser les problèmes récurrents liés au déploiement. Une fois une erreur corrigée, elle n'arrivera plus jamais.

Deuxièmement, l'automatisation permet de certifier que les artefacts déployés d'un environnement à l'autre proviennent de la même source et qu'ils ont été déployés de la même manière et dans le bon ordre. Par exemple, on peut certifier que le déploiement dans l'environnement de test a été fait de la même manière que dans l'environnement de pré-production et que le déploiement sera identique en production. En conséquence, le niveau de confiance de succès du déploiement en production augmente.

Cependant, les efforts initiaux nécessaires pour standardiser et pour automatiser les déploiements sont significatifs. Il faut prouver que le coût de ces efforts sera compensé par une réduction du coût d'exploitation de l'entreprise à long terme.

La compagnie américaine XebiaLabs s'est penchée sur le sujet de la rentabilité de l'automatisation des déploiements. Cette entreprise se spécialise dans les déploiements automatisés. Elle offre des outils et des services aux entreprises afin de les aider à automatiser leurs déploiements. On retrouve dans [XEB 11] une étude de cas effectué auprès de trois entreprises qui ont automatisé leurs déploiements.

Cette étude avance que les trois entreprises ont atteint un retour sur leur investissement en aussi peu que six mois. En effet, le coût des efforts initiaux d'automatisation ainsi que le coût des licences d'outils de déploiement ont été absorbés plus rapidement que prévu. Plus ces entreprises ont raffiné leur processus de déploiement, moins ils ont eu d'erreurs reliée au déploiement. La réduction du nombre d'erreurs de déploiement entraîne une diminution directe du nombre d'heures consacrées au support de déploiement. C'est en réduisant ces efforts humains que les entreprises tirent profit de l'automatisation.

2.3 Les outils de déploiement automatisés

La liste des outils disponibles pour automatiser les déploiements est large et la majorité de ces outils offrent des fonctionnalités semblables. Certains outils se démarquent par rapport à leur facilité d'intégration à une plateforme spécifique et d'autres par rapport à leur modèle de licence.

L'entreprise a commencé les démarches officielle de recherche d'un produit à l'hiver 2013 dans le but de pouvoir commencer à l'exploiter l'été suivant. Parmi les produits disponibles, l'entreprise a favorisé les produits développés et supportés par des grandes entreprises. L'entreprise a donc retenu deux produits commerciaux.

Le premier se nomme CDA (Continuous Delivery Automation). Ce produit est développé et supporté par HP (Hewlett Packard). L'avantage principal de ce produit est son niveau d'intégration avancé avec les autres produits HP existants.

Le second est InRelease, un logiciel développé par la firme montréalaise InCycle Software. Ce produit a été directement recommandé par Microsoft au moment des recherches à l'hiver 2013. Il est important de mentionner qu'InRelease a été acheté par Microsoft vers la fin de l'été 2013. Ce produit est maintenant connu sous le nom de « Release Management for Visual Studio » et il est inclus avec la version 2013 de Visual Studio (environnement de développement pour les produits Microsoft).

2.4 Sélection d'un outil de déploiement automatisé

Les deux candidats retenus ont été invités à présenter leurs produits à l'entreprise à l'aide d'une démonstration.

La présentation donnée par InCycle Software s'est bien passée. Il a été possible d'installer l'application et de créer un exemple de déploiement fonctionnel en quelques heures. La

présentation a permis de ressortir certains problèmes avec l'application, comme l'impossibilité de déployer à travers plusieurs environnements virtuels différents. Toutefois, InCycle a effectué le suivi sur ces problèmes et a offert une version corrigé du logiciel peu de temps après leur présentation.

La présentation offerte par HP a été plus laborieuse. En deux jours, il a été impossible d'installer correctement l'application sur l'infrastructure Microsoft de l'entreprise. L'application, codée en Java, nécessite l'installation et la configuration de plusieurs composantes additionnelles. Selon HP, le produit est testé et supporté sur Windows Server 2008 R2. Ironiquement, les présentateurs n'avaient pas accès à un plan de déploiement complet; Il manquait des éléments pour installer correctement l'application sur une plateforme Microsoft. Après plusieurs efforts en vain, la présentation a pris une tangente plus théorique que pratique.

Ces présentations ont mené aux conclusions suivantes :

- Les besoins de l'entreprise exerce une influence sur le « backlog » InRelease
- InRelease s'intègre mieux sur la plateforme Windows que CDA
- InRelease a une maturité plus grande que CDA

Au niveau du prix, InRelease offre 10 licences pour 4 950\$ tandis que CDA offre 5 licences pour 18 149\$. Une licence permet d'installer un déployeur sur un serveur. L'entreprise avait besoin de 4 déployeurs pour la preuve de concept initiale et prévoyait avoir besoin du double pour déployer la majorité des applications.

Pour son prix et pour ses avantages sur son compétiteur, l'entreprise a décidé de choisir InRelease comme outil commercial pour le déploiement automatisé des changements.

CHAPITRE 3

INRELEASE

3.1 Aperçu du produit

InRelease utilise l'architecture client-serveur. Un dépoyeur (client) est en réalité un service Windows installé sur une machine. La relation entre le serveur et les dépoyeurs est une relation un à plusieurs. Chaque dépoyeur interroge le serveur de déploiement à un intervalle de 30 secondes. Lorsqu'un déploiement est assigné à un dépoyeur, il télécharge les artefacts logiciels par HTTP et il exécute la séquence de déploiement associé.

Une vue globale de l'infrastructure associé au logiciel est disponible à l'annexe I.

Le produit permet de créer des séquences de déploiements à travers plusieurs environnements. Pour ce faire, le produit permet de créer une séquence de déploiement à l'aide d'activités prédéfinies. Pour construire un déploiement, il suffit de glisser des activités sur le canevas et de configurer les paramètres.

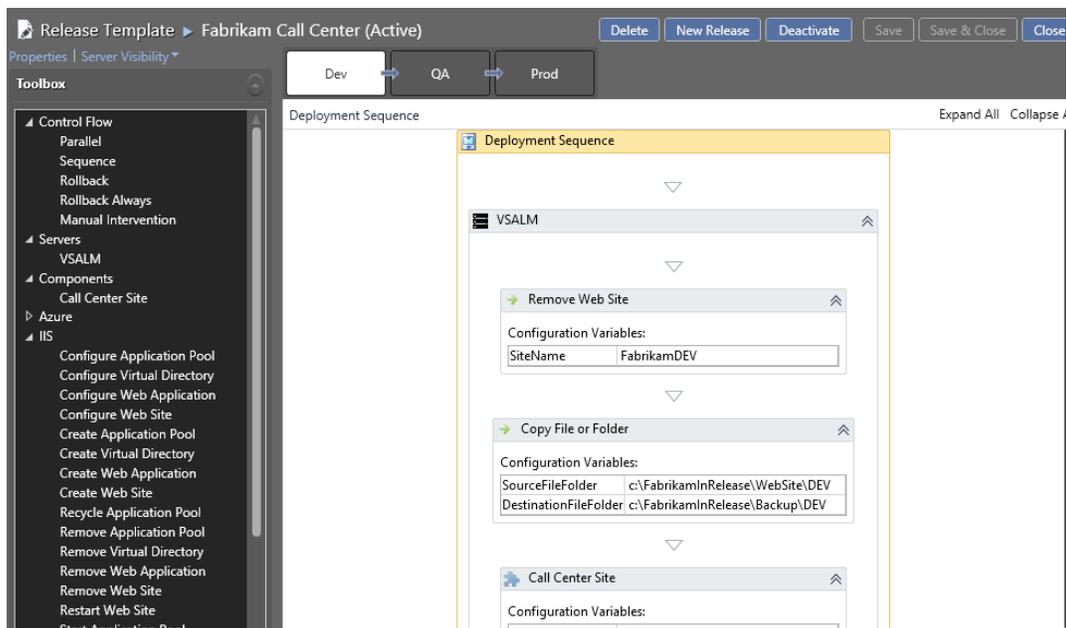


Figure 3 : Création d'un « Release Template »

Cette méthode de création de « workflow » est virtuellement semblable à celle proposé par Microsoft pour l'édition des fichiers XAML, à l'exception qu'InRelease ne permet pas de traitement basé sur des conditions. Ces fichiers sont utilisés pour gérer la séquence et la complexité du processus de compilation associé aux technologies .NET.

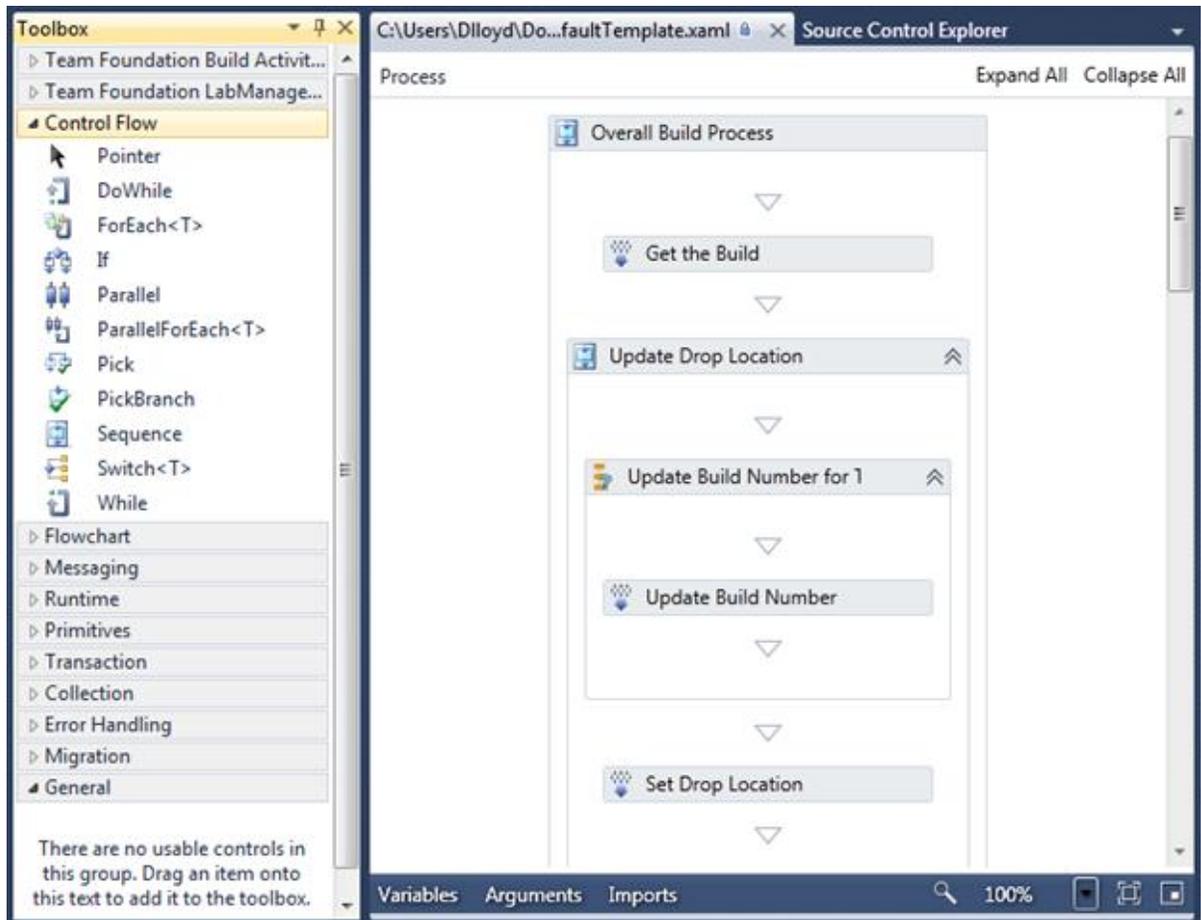


Figure 4: Création d'un build à partir de Visual Studio

3.2 Environnement virtuels

Much of the complexity in the deployment process [...] stems from interactions between the product being installed, the environment and the execution policy constraints. One approach to avoiding this complexity is to abstract over these interdependencies; this is the approach followed in .Net, Java Beans and in the Web-Services approach. Another approach is to obviate these problems entirely by creating perfect custom environments into which applications and components may be installed. Whilst in the past this might have seemed a Utopian dream, it may now be achieved using virtualisation.¹

La virtualisation permet de cloner un ensemble de machine et de les isoler dans un réseau privé. Par exemple, il est possible de recréer le réseau de production dans un environnement séparé et de couper la communication à l'extérieur de cet environnement. Chaque environnement possède son propre contrôleur de domaine qui contient, entre autre, sa propre version de « Active Directory » et son propre registre d'entrées DNS.

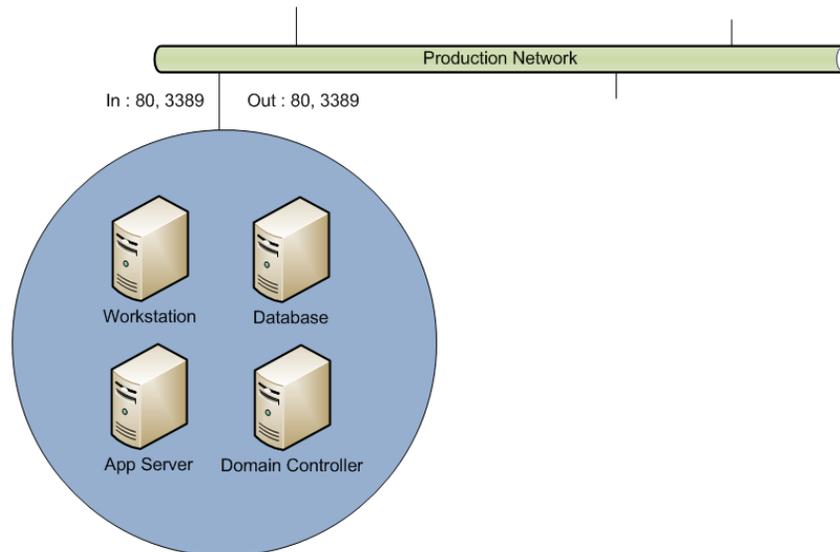


Figure 5 : Environnement virtuel isolé

¹ Passage tiré de [DEA 07]

Il est ainsi possible de recréer exactement les mêmes conditions de déploiement que la production. Lorsqu'un déploiement fonctionne dans un environnement virtuel de test, le niveau de confiance de succès du déploiement est assez élevé.

Avec la version originale d'InRelease, il n'était pas possible d'installer des agents de déploiement dans des environnements virtuels. Le serveur enregistrait les agents avec une clé unique composée du nom de la machine et de son adresse. Cependant, avec les environnements virtuels, ces configurations sont volontairement identiques d'un environnement à l'autre.

L'entreprise a adressé ce problème à InCycle lors de la recherche d'un outil de déploiement automatisé. InCycle, en coopération avec l'entreprise, a développé une nouvelle fonctionnalité permettant d'enregistrer les agents de déploiement avec une clé composée du nom de la machine et de l'adresse IP de la passerelle (gateway) utilisée par l'environnement virtuel. Cette adresse est unique à chaque environnement.

Cet exercice a non seulement rendu possible l'utilisation du produit, mais a aussi démontré que les besoins de l'entreprise seront adressés adéquatement à l'avenir. En d'autres mots, l'entreprise considère que ses besoins futurs auront une influence directe sur la direction du produit. Ce facteur a été pris en considération lors du choix du produit.

3.3 Principe d'utilisation

Le logiciel InRelease est intimement lié à la plateforme de build Microsoft. Un « workflow » de déploiement InRelease peut être lié à un ou plusieurs modèles de build. Au moment de lancer un déploiement, il est possible de choisir le ou les builds spécifiques (en fonction du numéro de build) à déployer. Les artefacts logiciels associés à ces builds seront utilisés lors

du déploiement. En cas de problème avec un déploiement, il est facile de trouver le build fautif et le code source utilisé pour compiler.

Lors de la création d'une séquence de déploiement pour un logiciel particulier, les efforts initiaux sont effectués dans le design de la séquence de déploiement. Une fois cette étape complétée, il suffit de lancer un build et de déployer à partir de ce build. Toutes les demandes subséquentes de changement au produit déployé se résumeront à relancer un build et redéployer avec InRelease. Conséquemment, il est pratique d'automatiser les produits qui sont déployés le plus souvent.

3.4 Intégration aux processus de gestion de changements

L'entreprise suit de près la méthode ITIL de gestion de changements. Voici les grandes lignes de cette méthode :

1. La demande de changement est complétée et communiquées au « Service Desk », cette demande comprend le plan de déploiement.
2. Le plan de déploiement est revu par les experts des différents systèmes concernés.
3. La priorité, l'impact et le risque du changement sont évalué.
4. Les ressources nécessaires au changement sont réservées selon la cédule de changements.
5. Le changement est autorisé.
6. Le changement est effectué à la date et à l'heure prévue.
7. Le succès du changement est évalué.

L'application InRelease vient avec une gestion de changement de base qui permet, entre autre, de planifier un horaire de déploiement et de gérer l'autorisation des changements (point 4, 5 et 6). Ces fonctionnalités sont excellentes pour une entreprise peu mature dans la gestion de changement.

Or, la procédure ITIL actuelle ne peut pas être remplacée ni modifiée tant que tous les déploiements de changement de l'entreprise ne seront pas automatisés. Il a donc fallu adapter le produit en fonction de la procédure existante.

Pour ce faire, la fonctionnalité de cédule un déploiement à l'avance a été abandonnée pour deux raisons. Premièrement, l'équipe de production a manifesté l'intérêt de vouloir garder le contrôle sur le moment exact du déploiement. En effet, lorsqu'un incident majeur survient, les ressources assignées au déploiement des changements sont assignées à régler le problème de production. Ils pensent oublier de canceler un déploiement prévu à l'avance lorsqu'une telle situation se produit. Deuxièmement, certains déploiements ont une partie manuelle à effectuer avant la partie automatisée. Il est difficile d'estimer le temps exact où la partie manuelle se termine.

La fonctionnalité d'autorisation de déploiement a été conservée, elle est utilisée de la manière suivante. Dans le plan de déploiement, au moment où la partie automatisée du changement doit être effectuée, il est indiqué d'autoriser et de surveiller le déploiement. Ceci permet de déclencher le déploiement automatisé au bon moment. Dans le cadre d'un déploiement complètement automatisé, le plan de déploiement contient une seule étape : Accepter la demande d'autorisation InRelease.

La gestion des autorisations passe par le module de « Release Path ». Ce module permet d'attribuer une équipe de personnes responsables d'accepter et de valider les déploiements dans un environnement spécifique. Les déploiements suivent le « Release Path ». En temps normal, un déploiement prend un peu plus qu'une semaine pour terminer son « Release Path ». L'entreprise exige un délai minimum de deux jours après avoir déployé dans un environnement pour tester et valider l'intégration avec les autres programmes. Dans le cas d'une correction urgente, le « Release Path » peut être complété en quelques minutes.

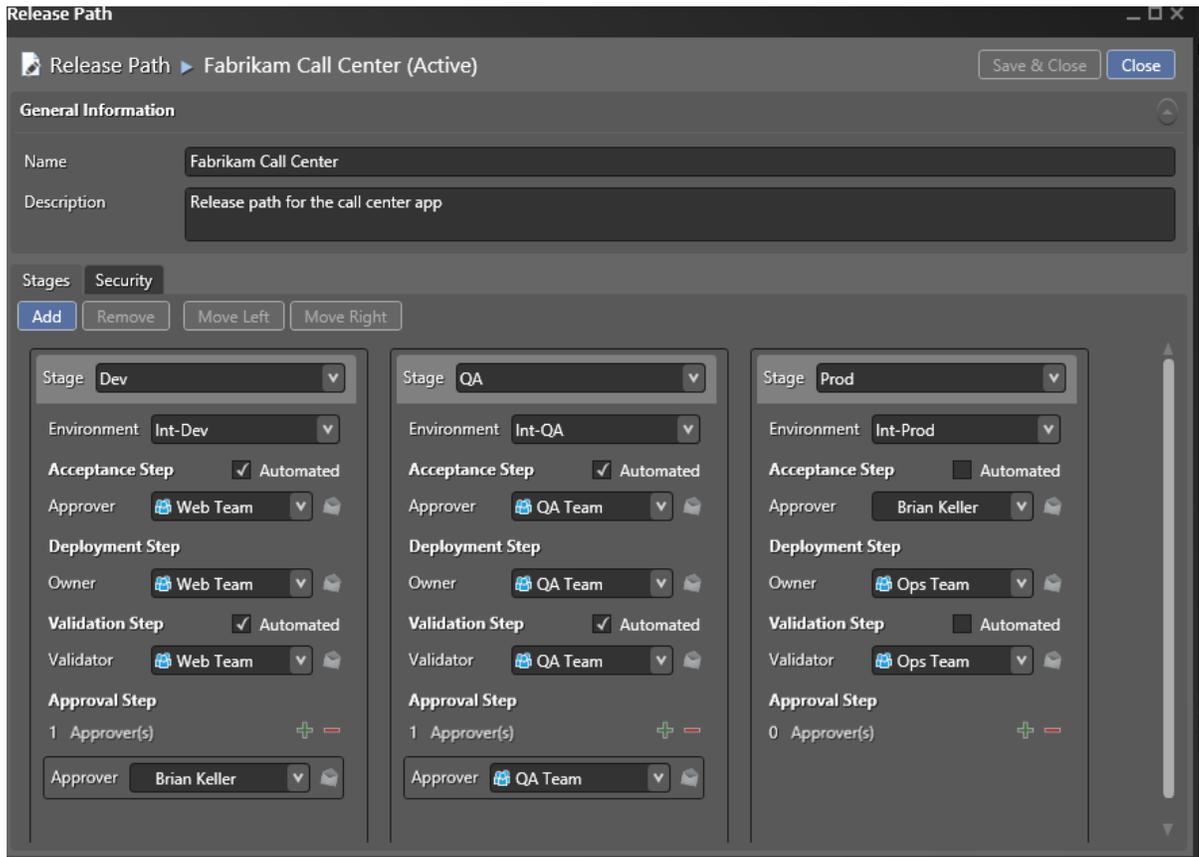


Figure 6 : Création d'un « Release Path »

En respectant la procédure actuelle, le projet a été plus facile à intégrer dans l'entreprise. De plus, cette approche permet la flexibilité de déployer manuellement et automatiquement côte à côte. Il est donc possible de commencer à utiliser l'outil progressivement, sans avoir à automatiser le déploiement de toutes les technologies que l'entreprise supporte.

CHAPITRE 4

STANDARDISATION DES DÉPLOIEMENTS

4.1 Sources d'informations

Avant d'entreprendre l'automatisation d'une procédure de déploiement, il faut d'abord standardiser la procédure. Pour ce faire, il faut comprendre en détail le déroulement de la procédure et il faut aussi connaître les acteurs qui l'exécutent dans le quotidien.

Pour comprendre la procédure, il est possible d'étudier l'historique des plans de déploiement. En comparant tous les plans de déploiement, il est possible de trouver les étapes critiques de la procédure, soit celles qui se répètent lors de chaque déploiement. Inversement, l'étude des plans de déploiement permet aussi de ressortir les étapes exceptionnelles, soit celles liées à des cas particuliers. Une fois les étapes d'un déploiement bien découpées, il suffit de les placer dans le bon ordre afin d'obtenir une procédure standardisée.

Pour avoir une vision complète de la procédure de déploiement, il faut aussi parler à ceux qui déploient les applications, par exemple, les groupes de production et/ou les administrateurs de bases de données (DBA). Ces acteurs effectuent souvent des activités additionnelles, au cours du déploiement, pour assurer un déploiement sans problème. Par exemple, lors du déploiement d'un projet de base de données, l'outil de déploiement Microsoft génère un script SQL en comparant un schéma source avec la base de donnée cible. Une fois le script généré, les DBA vérifient la présence d'opérations dangereuses dans le contenu du script. Si une opération dangereuse est détectée, le changement est annulé. Cette opération, non spécifié dans le plan de déploiement, est une partie importante de la procédure de déploiement de cette technologie.

En étudiant les plans de déploiement précédents et en interrogeant les personnes qui déploient les applications, il est possible de décomposer une procédure de déploiement en un nombre fini d'étapes. Par la suite, il faut analyser chacune des étapes en terme des aspects

suivants : Les entrées, les sorties et la stratégie en cas d'erreur. Reprenons l'exemple invoqué précédemment.

Tableau 1 : Décomposition d'une activité de déploiement

Tâche	Vérification d'opérations dangereuses dans le script SQL.
Entrée	Script SQL généré automatiquement par l'outil Microsoft.
Sortie	Rapport sur les différentes opérations présentes dans le script.
Stratégie en cas d'erreur	Si une opération dangereuse est détectée, le déploiement est annulé.

En décomposant chacune des tâches de cette manière, il est possible d'obtenir un standard défini. Il ne reste qu'à traduire ces tâches en activités logicielles afin d'obtenir un déploiement automatisé.

4.2 Approche par package de déploiement

Alors que la logique de déploiement peut être contenue dans le logiciel InRelease, cette approche est déconseillée.

Conserver la logique de déploiement dans l'application crée une dépendance envers le produit. Il sera plus difficile de se séparer du logiciel ou de migrer vers un autre produit de déploiement automatisé dans le futur. De plus, il est difficile de trouver des experts InRelease à Montréal pour le développement et la maintenance des séquences de déploiement.

De plus, en cas de panne ou de problème avec un déployeur, il n'est plus possible de déployer à partir de l'application InRelease. Aussi, entretenir, déboguer et tester une séquence de déploiement est une opération longue et disponible à réaliser.

Finalement, tel que mentionné précédemment, les séquences de déploiement InRelease ne permettent pas de traitement conditionnel. Ceci impose des limitations importantes à la logique de déploiement.

Pour palier à ces problèmes, l'entreprise a favorisé d'approcher les déploiements par package. Un package est un ensemble d'artéfacts logiciels et de scripts Power Shell qui contiennent la logique de déploiement. Le package doit être complet et posséder le moins de dépendances possible à des configurations externes. Power Shell est l'outil de déploiement favorisé par Microsoft et il est facilement utilisable par les programmeurs .NET. De plus, Microsoft offre des modules Power Shell avec tous leurs nouveaux produits pour automatiser les tâches d'administration et de configuration.

Le package est généré automatiquement lors du build de l'application. Les artéfacts logiciels obtenus par la compilation du programme sont agencés au script de déploiement et au manifeste de déploiement afin de former une seule archive zippée. Le manifeste du package contient des métadonnées tels que le nom et la date du build utilisé pour créer le package.

En cas de problème avec l'application ou un déployeur, il suffit de copier manuellement le package et d'exécuter le script de déploiement. Il est aussi possible de partager les packages avec d'autres équipes sans avoir à passer par le formalisme de déploiement imposé par l'utilisation d'InRelease. Inversement, une autre équipe peut développer un package qui respecte les règles et ce dernier sera facilement déployable à partir d'InRelease.

4.3 Approche agile

La méthodologie agile propose de séparer la réalisation d'un travail en plusieurs itérations. Chaque itération doit offrir une valeur ajoutée au projet. En effet, le premier principe du manifeste agile [AGI 01] est de livrer tôt et fréquemment du logiciel à valeur ajoutée au client. Cette méthode de travail est directement applicable dans un projet d'automatisation et de standardisation.

Considérant la vaste étendue de technologies utilisées et supportées par l'entreprise, il est inconcevable de tout automatiser en même temps. Il faut donc sélectionner les morceaux à automatiser en fonction de leur valeur ajoutée pour l'entreprise. Voici les critères de sélection pour déterminer la valeur de l'automatisation :

Risque : Ce critère exprime la sévérité de l'impact qu'un déploiement peut avoir sur les applications existantes. Par exemple, un déploiement fautif dans SharePoint pourrait rendre l'intranet inutilisable. Une technologie à faible risque est favorisée.

Fréquence de déploiement : Il est important de viser les technologies qui sont déployées le plus souvent. Ce sont elles qui gagnent le plus à être automatisées. Les technologies déployées le plus souvent sont donc favorisées.

Simplicité du déploiement : Représente le nombre et la complexité des opérations nécessaires lors d'un déploiement. Un processus de déploiement complexe peut être difficile à standardiser et, conséquemment, difficile à implanter dans InRelease. Les technologies faciles à déployer sont favorisées.

Facilité de « roll-back » : En cas de problèmes lors du déploiement automatique, il faut soit corriger le problème manuellement et poursuivre le déploiement (Fix forward), soit défaire ce qui a été déployé (Roll back). Une technologie facile à défaire est favorisée.

Maîtrise actuelle des packages : L'utilisation d'InRelease nécessite l'utilisation de packages de déploiement standardisés; Le package doit toujours avoir la même structure et son déploiement doit dépendre des mêmes scripts. Les technologies qui possèdent déjà des « packages » standardisés sont favorisées.

Le projet a suivi la méthodologie agile en commençant par une technologie simple à déployer. Ceci a permis de faire un bout en bout et de compléter la preuve de concept de l'utilisation d'InRelease. Par la suite, les technologies restantes ont été priorisées selon l'évaluation des critères énoncés précédemment. Au moment de la rédaction de ce rapport, il reste encore des technologies dont le déploiement n'a pas été automatisé. Par exemple, les déploiements SharePoint et BizTalk sont encore manuels. Ceux-ci seront adressés prochainement.

4.4 Modèle de permissions

Étant donné que l'entreprise opère sous les technologies Microsoft, elle utilise Active Directory pour gérer les comptes des utilisateurs et les permissions. Le déploiement de changements nécessite des permissions spéciales au niveau des applications. Voici quelques exemples :

- Pour effacer un fichier sous Windows, le compte de l'utilisateur doit avoir accès en lecture et en écriture sur le fichier ou sur son dossier parent.
- Pour déployer des changements sur une base de données SQL Server, le compte de l'utilisateur doit être administrateur sur la base de données spécifique.
- Pour arrêter ou partir des services Windows, le compte de l'utilisateur doit être administrateur local sur la machine.

Un des aspects important du projet a été de déterminer les accès spécifiques nécessaires pour chaque type de déploiement. Un effort particulier a été effectué pour essayer de trouver le minimum possible de permissions pour réaliser chaque type de déploiement. Les permissions ont été séparées en groupes et chaque groupe contient la liste des permissions minimales pour un type de déploiement en particulier.

Dans l'infrastructure actuelle, un seul déployeur est utilisé par environnement. Ceux-ci sont configurés sous l'identité d'un compte de service qui est membre de chacun des groupes de déploiement.

Dans le futur, lorsqu'il y aura plus de déployeurs, il sera possible de créer d'autres comptes de services et de séparer les permissions correctement. Par exemple, il serait possible de créer un déployeur qui effectue les déploiements traditionnellement réservés au DBA et un autre déployeur qui effectue les déploiement du groupe de support applicatif. Ainsi, la séparation des droits limitera les impacts d'une brèche de sécurité.

CHAPITRE 5

RÉTROSPECTIVE

5.1 Facteurs de succès

Le projet d'automatisation des déploiements est un projet reconnu et encouragé par la haute direction. Les directeurs n'ont pas hésité pour faire avancer les éléments bloquant du projet. Ceci a permis d'effectuer le projet sans embûche administrative importante.

L'utilisation d'un environnement virtuel a permis de simplifier la logique de déploiement en éliminant la gestion de configuration de déploiement. Par exemple, l'utilisation d'entrées DNS lors des déploiements évite d'avoir à maintenir des fichiers de configuration pour chacun des environnements virtuels.

5.2 Défis

Au niveau de l'architecture, l'utilisation d'un environnement virtuel s'est entraîné certains problèmes techniques. En effet, le serveur InRelease est installé dans le domaine de production tandis que la plupart des dépoyeurs sont installés à l'intérieur d'environnements virtuels qui ont leurs propres domaines. Pour établir une communication entre les dépoyeurs et le serveur, il a fallu faire part d'ingéniosité et combiner des techniques de délégation Kerberos entre des domaines différents. Ce défi technique a pris une majeure partie du temps alloué à la preuve de concept du produit.

La résistance au changement a aussi été un défi important. Certains acteurs, impliqués dans le processus de déploiement de changement, ne faisaient pas confiance à la solution d'automatisation proposée. Il a été nécessaire de les rencontrer plusieurs fois afin de leur présenter le projet et de prendre leur feedback. Leurs points d'inquiétudes ont été adressés et le projet a gagné en popularité auprès de l'entreprise. En perspective, certains voulaient avoir un droit de regard sur la solution avant de donner leur confiance au projet.

CONCLUSION

Les problèmes issus des déploiements manuels permettent de sensibiliser les entreprises vis-à-vis la complexité de leur parc informatique. En prenant conscience de cette complexité, elles ouvrent le chemin vers la standardisation et l'automatisation. Ces processus permettent d'améliorer la vitesse et l'efficacité des équipes de développement et de maintenance.

L'automatisation d'un processus requiert une compréhension exhaustive du mode de fonctionnement et du contexte dans lequel il prend place. Ce rapport démontre l'importance et l'ampleur de l'analyse nécessaire pour automatiser un processus.

L'automatisation des déploiements est une pratique jeune et en évolution. Les standards ne sont pas encore bien définis et il existe une variété de produits différents pour adresser le problème. Cependant, les fournisseurs logiciels investissent de plus en plus d'efforts pour faciliter le déploiement de correctifs sur leur plateforme. En effet, le potentiel d'automatisation est un critère qui commence à avoir un poids important lors du processus de sélection d'un logiciel.

BIBLIOGRAPHIE

- [XEB 11] Baart, Coert. 2011. « ROI Study on Automated Application Deployments for the Enterprise ». En ligne. 13 p. < http://go.xebialabs.com/rs/xebialabs/images/WP_2012-08_ROI.pdf >. Consulté le 4 février 2014.
- [LAH 11] Lahtela, A.; Jantti, M., « Challenges and problems in release management process: A case study », Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on, vol., no., pp.10,13, 15-17 July 2011
- [DEA 07] Dearle, A., « Software Deployment, Past, Present and Future », Future of Software Engineering, 2007. FOSE '07, vol., no., pp.269,284, 23-25 May 2007
- [AGI 01] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. (2001). « Principles behind the Agile Manifesto ». En ligne. < <http://agilemanifesto.org/principles.html> >. Consulté le 10 mars 2014.
- [AQL 11] Alain April et Claude Y. Laporte. 2011. L'assurance qualité logicielle 1 : concepts de base, ISBN 978-2-7462-3147-4. Paris : Lavoisier. 385 pages.

ANNEXE I

