

**TRIBUNE  
HARWELL**

# La production de solutions IT à l'heure du DevOps

Septembre 2019

**HARWELL**  
MANAGEMENT

# Toujours plus rapide, plus fédérateur et de meilleure qualité : le DevOps concrétise les promesses de l'Agile.

**Après l'accélération de l'écriture des logiciels via les développements en méthode Agile, c'est-à-dire sur la base d'une granularité très fine des fonctions à coder, le DevOps s'attaque aux goulots d'étranglement qui persistent sur la chaîne de production de solutions IT que sont l'intégration des développements, les tests et la mise en production.**

## **Qu'est-ce que le DevOps ?**

Le sigle DevOps vient de la contraction de « Dev » pour les développeurs informatiques qui écrivent le code des applications et de « Op » pour les Opérations, c'est-à-dire la production informatique qui assure les traitements quotidiens et réalise la mise en production des nouvelles applications ou de leurs évolutions. Il est question dans le DevOps d'une meilleure collaboration entre ces équipes de la Direction des Systèmes d'Information (DSI), qui traditionnellement communiquent peu entre elles, afin d'accélérer les phases en aval de l'écriture de logiciel, c'est-à-dire la phase d'intégration (le continuous integration) qui consiste à créer un programme exécutable à partir des différents bouts de code développés, les phases de tests des programmes obtenus (le continuous testing) et la phase de mise à disposition des utilisateurs ou des clients des

logiciels ayant satisfait aux différents tests (le continuous delivery).

Les développeurs doivent fournir le plus rapidement possible les évolutions du SI demandées par les métiers quand, à l'opposé, les opérations sont responsables de la bonne production du SI et donc de sa stabilité. D'une collaboration constructive entre développeurs et opérateurs de production aux objectifs antinomiques, on est ensuite passé à un état d'esprit de collaboration renforcée tout au long de la chaîne de production de logiciel. Le métier tient compte des contraintes de la DSI pour prioriser ses demandes, lors de la phase de cadrage. La maîtrise d'ouvrage cherche à minimiser les développements nouveaux au profit de l'évolution de l'existant. Les développeurs tiennent compte des contraintes de la production en exploitant les référentiels et les bases de données existantes et facilitent les phases de tests soit en commençant par concevoir les cas de tests avant de coder les applications (c'est le TDD pour Test Driven Development), soit en collaborant avec les testeurs qui devront mettre à jour les outils pour les tests automatiques (les tests de non-régression, en particulier). Inversement, chaque équipe doit faire preuve de pédagogie pour expliquer aux équipes en amont ses contraintes, les facteurs qui facilitent ses tâches et, in fine, accélèrent sa production. Dans une démarche

DevOps mature, chaque acteur est conscient de sa valeur ajoutée mais plus encore de son rôle de facilitateur au bénéfice de l'ensemble de la chaîne de production.

### L'adoption du DevOps est complexe des points de vue technique et gestion du changement mais bénéficie de facilitateurs avec le Cloud computing et les méthodes agiles

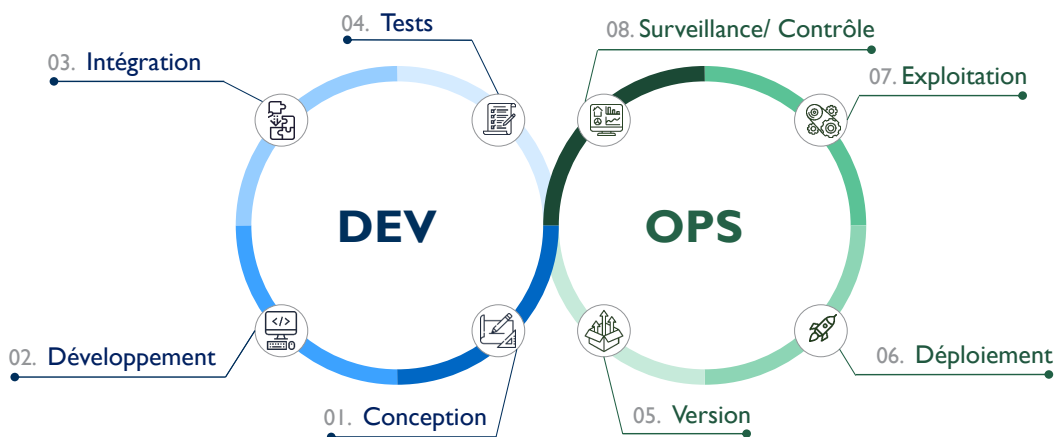
L'adoption du DevOps, au sein d'une DSI et pour tous les acteurs de la production de solutions informatiques plus globalement, n'est pas aisée. Même si, comme nous le verrons, les bénéfices sont importants, il convient de bien mesurer toutes les difficultés et à contrario les facteurs facilitants avant de lancer une démarche DevOps.

En premier lieu il faut garder à l'esprit que l'automatisation des tests et des mises en production sont des sujets complexes techniquement après lesquels les DSI courent depuis longtemps sans l'émergence jusqu'à maintenant d'une solution idéale et exhaustive. Il faut ensuite ne pas sous-estimer les freins à une collaboration étendue entre des équipes disjointes. Selon Gartner, d'ici 2022, 75% des démarches DevOps échoueront à apporter les bénéfices attendus en raison de problèmes relatifs à la conduite du changement. L'impact du DevOps sur les équipes, en particulier les Opérations, doit être anticipé de même que la rareté des experts DevOps sur le marché (toute chose étant égale par ailleurs, le salaire

est 20% supérieur pour une personne ayant une expérience DevOps). A l'inverse, le Cloud computing et les développements en méthode Agile doivent être pris en compte comme facilitateurs d'une démarche DevOps.

Des objectifs de longue date difficiles à réaliser complètement

Aussi, chercher à automatiser les tests ou les mises en production n'est pas nouveau. Depuis plusieurs décennies, les robots de test tentent, avec plus ou moins de bonheur et de pérennité par rapport à la succession des versions d'une application, d'automatiser les cas de test de non-régression en enregistrant les actions du testeur (souris et clavier) pour les rejouer automatiquement plus tard et comparer les résultats avec ceux obtenus lors de l'enregistrement. La première génération de ces outils était très limitée : il suffisait par exemple d'un changement dans le design d'un écran (i.e. le déplacement d'un « bouton ») pour avoir à recommencer l'enregistrement des séquences de saisies impactées. De même les faux positifs, par exemple un résultat avec une date différente entre la reproduction du test et le test initial considéré à tort comme une anomalie, n'étaient pas bien gérés. Les solutions plus modernes interceptent les messages échangés entre le client, c'est-à-dire l'utilisateur, et l'application. Le moteur de test n'est plus perturbé alors par une modification mineure de l'IHM (Interface Homme Machine). Cependant des évolutions de format, ou une



nouvelle version sur les messages nécessitent de revoir le paramétrage de l'outil. La gestion des faux positifs reste souvent un point faible de ces outils et nécessite, dans le meilleur des cas, un paramétrage pour chaque cas de détection erronée.

En résumé, l'automatisation des tests de non-régression est un objectif de longue date des DSI mais reste un sujet pointu techniquement qui nécessite un outillage spécialisé (Ranorex, HP Unified, unctional Test, VS Ultimate, Selenium, Sahi, Sikuli pour n'en citer que quelques-uns) que seul un investissement humain constant permet de garder opérationnel.

L'automatisation des mises en production, elle aussi, n'est pas un objectif nouveau et de nombreuses DSI s'y sont essayées avec des gestionnaires de versions, des outils pour packager les livraisons, des listes d'objets de différentes natures et autre macrolangage de programmation pour gérer l'ensemble : dans une certaine mesure, le DevOps est la formalisation de pratiques qui existaient depuis longtemps dans certaines DSI avec, pour le DevOps, des objectifs précis et un niveau de conceptualisation propres à une application à grande échelle.

### Une collaboration renforcée qui ne se décrète pas

On pourrait naïvement penser que l'esprit de collaboration, « comment, sur une chaîne de production séquentielle, une personne ou une équipe peut faciliter les tâches des équipes en aval » va de soi dans une entreprise. Pour le DevOps, sur la chaîne de production de solutions IT, c'est pourtant l'objectif le plus ambitieux. On observe des équipes métiers qui cherchent, c'est bien légitime, à couvrir leurs besoins pour satisfaire leurs clients, gagner en efficacité, voire en part de marché, etc. Mais elles ne se préoccupent pas de la complexité de leurs demandes ni de la possibilité de capitaliser sur le SI existant. De même les études informatiques cherchent à répondre aux spécifications fonctionnelles sans se soucier de la complexité des tests qu'il faudra mener

pour valider leur production ni sur le degré d'évolution technologique que les équipes de production devront absorber pour mettre en production les nouvelles applications puis assurer leur exploitation au quotidien ou mettre à jour les procédures pour garantir la continuité d'activité. Chaque équipe se concentre sur ses objectifs, dont l'atteinte est souvent associée à une rémunération variable, sans tenir grand compte d'une perspective d'ensemble (souvent le résultat de l'entreprise rentre en jeu pour le calcul de la participation ou de l'intéressement mais cela n'aide pas à appréhender le rôle de facilitateur que chacun doit endosser au-delà de sa propre production) qui pourtant seule fait la différence pour les clients et les utilisateurs finaux. Face à ces constats, très courants, proposer d'emblée une collaboration renforcée n'est pas la bonne stratégie. Elle se heurterait, comme on vient de le constater, à la somme des intérêts particuliers.

### Le devenir des « Ops » à prendre en compte

Aussi parmi les points de difficulté pour la mise en place du DevOps, le devenir des équipes de production, les « Ops », est à anticiper largement. Lorsque dans un niveau de maturité élevé, le DevOps permet de traiter les mises en production de façon automatisée, sans risque, jusqu'à devenir un non-événement, le rôle de l'équipe de production informatique diminue et le besoin en effectif de même. Le reclassement d'une partie de ces équipes doit être envisagé dès le début du processus de mise en place du DevOps si le flux naturel des départs ne permet pas d'absorber le sureffectif à venir. Ces reclassements éventuels sont à bien anticiper dans la mesure où le niveau moyen de qualification des équipes de production IT est moindre que celui des études par exemple et leur reconversion n'est pas forcément simple.

### Le Cloud computing et les méthodes Agile comme facilitateurs du DevOps

Même s'il ne s'agit pas de conditions sine qua none, l'utilisation d'infrastructure en Cloud (public ou privé) et des développements

en méthode Agile facilitent grandement la mise en place du DevOps par opposition aux développements en cycle en V et aux infrastructures traditionnelles gérées unitairement par l'équipe de production. Très compétitif économiquement grâce à une facturation selon l'utilisation (le Pay as you go) et le self-service, le Cloud computing permet en effet de mettre à disposition des développeurs ou des testeurs des infrastructures « iso-prod ». Ce « confort » est un des principes du DevOps dans le sens où il permet de se prémunir d'anomalies liées au dimensionnement des infrastructures. Par ailleurs, les développements en méthode Agile génèrent de nombreuses mises en production (théoriquement à l'issue de chaque release tous les 4 ou 5 sprints) avec souvent des micro-services et des conteneurs dont la mise en production, indépendante pour chaque conteneur, est simplifiée par rapport à celle d'une application classique. Chacune de ces nombreuses mises en production porte sur un nombre limité de fonctionnalités et d'objets informatiques. Le niveau de complexité est bien moindre que pour une mise en production sanctionnant des mois, voire des années de développements et de tests. Cette moindre complexité facilite la recherche de la cause d'un dysfonctionnement dans le processus de mise en production. Cette recherche simplifiée de l'origine des anomalies et la multiplication des mises en production (selon puppet.com, les entreprises ayant un niveau avancé en DevOps délivrent, en moyenne, 200 fois plus souvent que les moins avancées : 1 460 mises en production par an pour les premières contre 7 pour les secondes) permettent d'atteindre un très haut degré de fiabilité pour le processus automatisé : on note une diminution par 5 du taux d'échec des déploiements (les entreprises les plus performantes voient leurs changements échouer 7,5% du temps au lieu de 38,5%).

### **Très souvent le DevOps offre des bénéfices à la hauteur des investissements**

C'est la raison principale de la croissance du marché du DevOps : Selon l'infographie Faits & Prédictions DevOps 2018 de l'éditeur Arcad

software, la taille du marché DevOps devrait passer de 2,90 milliards USD en 2017 à 12,85 milliards USD d'ici 2025. Aucune entreprise n'est connue pour avoir mis en place avec succès le DevOps sur sa chaîne de production de logiciel puis avoir fait marche arrière. IDC prévoit pour les entreprises françaises de 35 à 40% de projets de développements applicatifs en DevOps en 2021 contre 20% aujourd'hui. Les bénéficiaires en question sont nombreux. En réduisant ou en supprimant le goulot d'étranglement pour l'intégration, les tests et les mises en production, le DevOps contribue significativement à réduire le time to market. Pour exemple, Arcad software indique des délais d'exécution jusqu'à 440 fois plus rapides entre développement et déploiement avec des délais de moins d'une heure pour les entreprises les plus performantes. Le DevOps réduit également les risques opérationnels liés à des tests trop parcellaires ou à l'opération de mise en production elle-même. En effet, les délais de récupération des temps d'arrêt suite à un incident sont 96 fois plus courts pour les entreprises appliquant le DevOps avec des délais de récupération jusqu'à une heure ou moins. Le DevOps est aussi un facteur d'Agilité et permet de développer un esprit de collaboration entre les équipes qui interviennent successivement sur la chaîne de production de logiciel.

### **Les bénéfices de l'automatisation**

Les premiers bénéfices du DevOps sont les bénéfices que l'on peut attendre d'une démarche d'automatisation : la réduction des délais et des risques opérationnels. L'automatisation des tests, en particulier les tests de non-régression, permet de « jouer » un nombre très important de cas de test couvrant l'intégralité des fonctionnalités de l'application en un minimum de temps. L'investissement initial lorsque l'on bascule une application dans un outil de test de non-régression est significatif car il faut paramétrer chaque cas de test. Cet investissement est ensuite moindre pour les ajouts de nouvelles fonctionnalités avec de nouveaux cas de test relatifs aux changements uniquement. Même si les outils de test ont leurs



limites, en particulier pour rester opérationnels au fur et à mesure des versions, l'automatisation des tests de non-régression permet une couverture de tests souvent bien plus large, bien plus rapide et bien moins chère qu'une campagne de tests manuels. La qualité des logiciels produits en est grandement améliorée et le risque opérationnel lié aux nouvelles versions diminue d'autant.

La mise en production d'une application est souvent une étape risquée. La grande majorité des incidents de production informatiques sont consécutifs à une mise en production, le plus souvent en raison de bogues non détectés lors de la phase de tests, mais également à cause d'erreurs ou d'oublis lors de la mise en production. Il s'agit, pour cette dernière, de recopier dans l'environnement de production les packages exécutables générés et testés positivement, de migrer des données applicatives automatiquement ou manuellement selon les volumes et la complexité de ces reprises de stock, de recopier des fichiers de configuration, de veiller à la mise à jour des habilitations, de s'assurer que les procédures de sauvegarde et de réplication restent exhaustives et opérationnelles, etc. Il est souvent nécessaire d'interrompre puis de redémarrer certains services (le File Transfert Protocol ou FTP pour les échanges de fichiers par exemple) et la cinématique de ces différentes tâches est parfois déterminante. Dans une démarche DevOps, la multiplication des mises en production permet de fiabiliser l'automatisation de cet exercice pour viser le continuous delivery ou la mise en production en continu au fur et à mesure de la validation des fonctions développées et testées avec succès. En plus de la réduction des risques opérationnels liés aux actions manuelles sur cette phase particulièrement complexe, l'automatisation des mises en production contribue à réduire le délai d'exécution de la mise en production elle-même et par conséquent le délai d'indisponibilité des applications durant cette phase.

Ces bénéfices sur le time to market et la réduction des risques opérationnels se mesurent facilement. D'autres bénéfices du DevOps sont plus délicats à mesurer. C'est le cas par exemple de la prise en compte plus rapide des retours clients, ou la réactivité accrue pour les corrections de bogues. Conséquences directes de la réduction du time to market et de la fréquence élevée des

mises en production que le DevOps autorise, ces atouts particulièrement appréciés des directions marketing et des services qualité permettent de tester le marché et d'adopter une stratégie commerciale offensive.

### Le DevOps contribue à l'agilité et permet à la DSI de se concentrer sur la création de valeur

Prolongement des méthodes Agile sur le pipeline de la production de logiciels, le DevOps contribue à la réduction des délais et donc à la réactivité de la DSI face aux changements. Le DevOps est aussi un levier d'agilité pour la DSI et pour l'entreprise.

De même que le Cloud computing permet aux DSI de se soustraire aux problématiques d'infrastructures, le DevOps permet aux DSI de s'affranchir de tâches risquées pour les mises en production afin de se concentrer sur des tâches à forte valeur ajoutée comme, par exemple, la conception de solutions pour les métiers ou les clients.

### Une collaboration renforcée comme bénéfice ultime du DevOps

Il est un bénéfice du DevOps, difficile à quantifier certes, qui pour l'entreprise dépasse probablement tous les autres. C'est le changement d'état d'esprit sur la chaîne de production de logiciel. La mise en place du DevOps nécessite un partage d'expertises entre les différentes équipes qui se succèdent sur la chaîne de production logiciel. Ces expertises vont de l'expression des besoins par les métiers et l'assistance à maîtrise d'ouvrage jusqu'à l'exploitation par les équipes de production informatique. Elles passent par les spécifications, l'encodage, les tests unitaires par les développeurs, la conception et le paramétrage des tests par les testeurs ainsi que le contrôle qualité. Ces partages d'expertises, d'expériences et de facteurs clef de succès sont l'essence même du DevOps dans le sens où ils contribuent naturellement à l'émergence d'une collaboration renforcée entre équipes disjointes tout au long de la chaîne de production de logiciel. La globalité de cette chaîne de valeur devient bien plus concrète pour tous ses acteurs. Le rôle de facilitateur qu'ils doivent

chacun assumer devient également un réflexe. Il est alors beaucoup plus simple et légitime de fixer des objectifs liés à la performance globale (délais et qualité) sur toute la chaîne de production de solutions IT. En d'autres termes l'esprit d'équipe global ne se décrète pas mais il peut être l'aboutissement d'une démarche DevOps réussie bien au-delà des développeurs et des opérations.

### Faut-il alors adopter le DevOps ?

L'instauration du DevOps est plus simple et plus naturelle dans le contexte de développements en méthode Agile et l'utilisation d'infrastructures en Cloud computing. C'est dans ces conditions que le retour sur investissement est le plus rapide à obtenir. A contrario, l'adoption du DevOps pour le SI legacy, sur de grands mainframes, avec des applications monolithiques maintenues via des développements en cycles en V paraît plus incertaine par rapport à d'autres investissements pour la DSI. Pour une première stratégie de mise en place du DevOps, on pourrait ainsi prioriser l'utilisation des dépenses d'investissement en capital (CAPEX) comme suit : méthodes Agiles et Cloud computing en premier lieu puis DevOps pour les environnements suffisamment matures sur ces deux premiers thèmes. Il faut ensuite prioriser les applications périphériques par rapport aux enjeux principaux des applications cœur de métier et accroître le périmètre, le niveau d'automatisation et l'expertise de l'équipe DevOps au fur et à mesure de l'intégration de nouvelles applications, de nouveaux projets comme autant de cycles d'amélioration continue.

Cette stratégie pour la mise en place du DevOps doit être complétée d'une stratégie spécifique pour constituer et faire monter en compétence l'équipe DevOps. Cette démarche pilote consiste à s'appuyer sur un noyau de volontaires (qu'il conviendra de supporter en toutes circonstances), très motivés par les bénéfices attendus d'une démarche DevOps. Idéalement, cette équipe doit comporter des membres issus de tous les acteurs de la chaîne de production de logiciel (métiers,

Maitrise d'ouvrage (MOA), développeurs, testeurs et contrôle qualité, production, etc.) pour faciliter, à terme, la diffusion du DevOps parmi ces équipes. Au départ, les membres de la première équipe DevOps peuvent conserver leurs responsabilités initiales et ne consacrer qu'une partie de leur temps au DevOps. Cette première équipe doit se fixer un objectif simple, réaliste et mesurable sans grand enjeu : par exemple l'automatisation des tests et de la mise en production d'une application (ou d'une évolution d'une application) jugée peu sensible et peu complexe techniquement. Ces premières itérations du DevOps seront l'occasion d'une étude sur les outils du marché, à acquérir ou déjà en place, par rapport aux technologies utilisées par la DSI. Elles seront également l'occasion de mettre en place ou de revoir des indicateurs clés de performance (KPI) pour mesurer le plus précisément possible la durée des différentes phases sur la chaîne de production logiciel ainsi que le nombre moyen d'itérations nécessaires pour obtenir le niveau de qualité demandé par les donneurs d'ordre. Ces informations serviront au pilotage du DevOps pour investir, via des outils, de la formation ou des processus, sur les goulots d'étranglement identifiés.

Forts de ses premiers résultats, l'équipe DevOps a vocation à se renforcer et à faire tâche d'huile (les employés recommandent leur entreprise comme *great place to work* 2,2 fois plus souvent parmi les entreprises avancées sur la mise en place du DevOps) sur toutes les équipes concernées par la production de logiciels. Pour rester efficaces face aux évolutions technologiques et aux nouvelles versions des outils dont ils ont la responsabilité, les experts de l'équipe DevOps devront bénéficier d'un niveau de formation élevé. Ce mode de constitution interne pour une équipe DevOps est le plus facile tant les ressources expertes restent rares sur le marché. Aussi le nombre d'outils et de solutions disponibles pour traiter tel ou tel aspect du DevOps (gestion des versions, tests de différentes natures, mise en production, suivi des performances, etc...), en l'absence de solution de référence, rend difficile l'émergence d'experts immédiatement opérationnels sur un nouvel environnement.



Enfin, même si l'outillage et la technique sont essentiels pour réussir l'implémentation du DevOps, le facteur humain reste, comme souvent, le facteur de succès clef le plus important et le plus difficile à satisfaire.

Sources: [puppet.com](https://puppet.com), [IDC](https://www.idc.com), [dzone.com](https://www.dzone.com), [gartner.com](https://www.gartner.com), [arc42.com](https://www.arc42.com)

## NOS EXPERTS

### **Sebastien DUPORTAL**

Partner

[sebastien.duportal@harwell-management.com](mailto:sebastien.duportal@harwell-management.com)

### **Pierre-Jean LATOUR**

Senior Manager

### **Anas HAKKAL**

Consultant