



Perspectives **DevOps**

Des personnalités du monde IT abordent les principaux obstacles actuels à l'approche **DevOps**

Perspectives **DevOps**

> **Introduction**

> **Synthèse**

> **Paul Speers**

PDG, Spearhead Group

> **Robert Benefield**

Directeur de la technologie, Evolve Beyond

> **Benjamin Wootton**

Cofondateur, Contino

> **Dave Farley**

Co-auteur de « *Continuous Delivery* », architecte, KCG Ltd



DevOps : deux mots en un au service de la praticité

CA Technologies travaille chaque jour avec des équipes chargées du développement et de l'exploitation d'applications logicielles à tous niveaux. Aussi comprend-elle parfaitement que le nouveau terme « DevOps » nécessite des explications et un examen plus approfondis.

Dans le cadre de notre programme **Perspectives DevOps**, nous nous sommes entretenus des problèmes majeurs ayant un impact sur l'approche DevOps avec de grands noms de la communauté.

Des personnalités au service du changement et de l'évolution IT

Nous sommes heureux de compter parmi les contributeurs de cet ebook certains des acteurs les plus compétents et les plus expérimentés de la communauté DevOps. Nous nous concentrons sur les questions clés, comme la mesure de l'avancement DevOps, le moyen d'obtenir le soutien de la fonction métier ainsi que les principaux points à prendre en compte ou à éviter.

J'espère que cet aperçu vous sera utile et vous permettra d'étoffer votre réflexion. Si vous souhaitez participer à la prochaine édition de **Perspectives DevOps**, contactez-nous sur ca.com/contact.

Justin Vaughan-Brown

Directeur de la stratégie DevOps EMEA
CA Technologies

Ces dernières années, chez CA Technologies, nous sommes parvenus aux conclusions suivantes en matière de DevOps :

Utilisez au maximum l'automatisation : la réalisation manuelle des tâches routinières est source d'erreurs et de défauts de production. Nous devons donc systématiquement considérer l'automatisation comme une opportunité, non comme une menace.

Mieux vaut commencer petit : une entreprise n'a pas besoin d'un comité à grande échelle pour examiner la mise en place de la méthodologie DevOps et formuler des recommandations ou un plan d'implémentation six mois plus tard. Il est souvent plus simple d'établir un projet pilote considéré comme « important », mais pas comme « essentiel ».

Il est important d'obtenir le soutien du sommet de la hiérarchie : vente de la valeur métier et nécessité de devancer la concurrence, mesure de la réussite d'initiatives en amont et preuve de l'efficacité d'une approche orientée DevOps.

Transformez le développement et le déploiement de chaque application du stade de projet à celui de processus : convertissez chaque mise en production d'application et chaque mise à jour ultérieure en un processus automatisé transparent et défini de commun accord. Ne réinventez pas la roue à chaque lancement de produit ou de service sur le marché.

Ne sous-estimez pas le facteur humain : la personnalité des principaux acteurs (et la façon dont ils réagissent face au changement) sera déterminante pour la réussite. La structure de votre organisation permet-elle des collaborations efficaces et l'adoption de nouveaux modes de travail ?





L'approche DevOps est un impératif technique, mais reste un défi culturel humain

Synthèse

Il est aisé de négliger le facteur humain dans l'approche DevOps. Il s'agit pourtant de la concaténation de deux termes professionnels, à savoir « Development » et « Operations », il y est donc bien question de relations humaines. Une entreprise incapable de créer une culture de développement positif au sein de son équipe (ou division) DevOps aura des difficultés à conserver son personnel, ce qui nuira directement à l'approche DevOps.

Nos contributeurs remarquent à juste titre que les équipes « Dev » et « Ops » sont parfois aux antipodes l'une de l'autre : les développeurs persistent à transmettre la version du logiciel aux opérations sans prendre la peine de communiquer et se soucier de la suite. Le principal atout de la méthodologie DevOps est précisément de rassembler tout le monde autour d'une table. Les différents acteurs DevOps doivent trouver un terrain d'entente et s'unir sur une appréciation commune pour éviter que la fonction métier ne prenne pas (assez) en compte l'une des deux disciplines.

Pour en revenir au facteur humain, il est important de veiller à ce qu'il n'existe pas un état d'esprit malsain, avec une certaine forme de rivalité entre développement et exploitation. L'équipe DevOps doit pleinement prendre en compte les deux aspects de la tâche, mais également le client, sans quoi un facteur de risque lié à ce dernier apparaîtra invariablement. Une approche DevOps réussie sera le plus souvent synonyme de restructuration, en vue de créer un environnement qui favorise un engagement continu et un partage optimal de l'information à tous les niveaux du système.

Une restructuration culturelle et organisationnelle sera certes nécessaire, mais l'application de la méthodologie DevOps en ligne ne nécessitera pas pour autant de tout abandonner pour tout reconstruire. Des projets incrémentiels plus restreints peuvent constituer un moyen plus prudent d'introduire l'approche DevOps sur le terrain. De cette façon, l'ensemble du métier peut découvrir ce en quoi consiste (ou ne consiste pas) la méthodologie DevOps et où dessiner les lignes entre responsabilité et propriété.



« DevOps est un impératif technique, mais reste un défi culturel humain. Ce gouffre sera difficile à traverser, mais nous pouvons le franchir sans danger si nous nous y prenons étape par étape, de manière unie et collaborative. »

Des pièges existent évidemment ; il est aisé de mal comprendre la notion de DevOps. Ce serait une erreur de créer une fonction DevOps comme « troisième équipe » indépendante à partir d'une approche descendante ; la division DevOps doit être constituée de personnel qualifié existant et, souvent, des membres les plus expérimentés de l'équipe. Les entreprises manquent clairement de professionnels disposant d'une vaste compréhension à la fois de la compilation ET du déploiement ; c'est ce que la méthodologie DevOps cherche à pallier.

Rappelez-vous : à ce stade, l'approche DevOps consiste à éliminer les anciens silos et à multiplier les expérimentations. Les entreprises qui s'en sortent le mieux à ce niveau sont les sociétés agiles, ce qui n'est pas forcément le cas des organisations de développement d'applications logicielles. Le changement intervient au niveau des fondations de l'entreprise ; les premières étapes peuvent se révéler

douloureuses si vous avez une vaste base de code propriétaire. Une fois encore, commencez doucement et de façon méthodique, avec un objectif stratégique clairement défini.

L'approche DevOps est en effet en pleine expansion en tant que pratique de travail formalisée et élément des systèmes modernes de gestion des données. Cela signifie qu'aujourd'hui, plus que jamais, nous devons comprendre et apprécier le besoin d'intégrer les vastes principes de la culture DevOps (rationalisation, automatisation, mesure et partage) à notre approche globale de l'architecture informatique.

L'approche DevOps est un impératif technique, mais reste un défi culturel humain. Ce gouffre sera difficile à traverser, mais nous pouvons le franchir sans danger si nous nous y prenons étape par étape, de manière unie et collaborative.





Paul Speers,
PDG, Speerhead Group

« N'oubliez jamais le facteur humain ! Le maintien du personnel et des principaux talents est l'une des façons de garantir la réussite des équipes DevOps. »

À quoi ressemble une bonne approche DevOps ?

Une bonne approche DevOps consiste en des mesures simples, utilisables à la fois par l'IT et par le métier. Des processus comme la durée moyenne de détection (MTTD), qui fournissent des mesures concernant un incident et déterminent sa fréquence, sa taille et son impact, peuvent être déterminants. Dans la suite logique vient la durée moyenne pour l'application d'un changement (MTTC). Nous commençons alors à vérifier les taux de répétition afin de déterminer si le même problème revient régulièrement.

Toutes les entreprises peuvent se servir d'un système de gestion des services simple afin de collecter les données, d'en assurer le suivi et de créer des rapports. Ce n'est qu'alors que vous pourrez commencer l'évaluation, puis, au fil du temps, constater une amélioration des services DevOps tangible et mesurable. Mais n'oubliez jamais le facteur humain ! Le maintien du personnel et des principaux talents est l'une des façons de garantir la réussite des équipes DevOps. Un renouvellement élevé entraîne un taux d'échec important du processus DevOps, et, par conséquent, du projet global.

Devons-nous mettre en place un langage commun pour unir les équipes Dev et Ops afin de permettre la réussite de l'approche DevOps ?

Je suis le cofondateur de Fox IT, l'une des entreprises ayant développé et créé ITIL dans le monde des affaires. Plus de 90 % des entreprises dans le monde, jusqu'aux PME, ont implémenté un cadre de gestion des services semblable à ITIL dans leur département IT ; vous comprenez donc l'importance d'une terminologie et d'un langage codifiés.

La clé, c'est de comprendre la façon dont les développeurs communiquent. Les équipes Dev et Ops sont parfois aux antipodes l'une de l'autre ; ITIL doit toutefois découvrir, intégrer et, d'une certaine façon, optimiser et rentabiliser les principes DevOps que sont la culture, la rationalisation, l'automatisation, la mesure et le partage. ITIL n'évolue pas assez rapidement et l'IT fait toujours preuve de réticence ! La base ITIL commence tout juste à s'intéresser à la méthodologie DevOps, tout comme les milliers de départements Ops qui parlent le langage ITIL. Lorsqu'ils auront trouvé la jonction avec le monde ITIL, ils adoreront et adopteront définitivement le « Dev ».



« Choisissez un projet pouvant être démarré et terminé à l'aide d'une nouvelle méthodologie. »

Comment faire en sorte que la prochaine génération soit préparée à la culture DevOps lorsqu'elle arrivera ? Quelles sont les compétences à développer ?

La demande pour des compétences DevOps grandit ; elle a augmenté de 75 % entre janvier 2012 et janvier 2013. Au cours de la même période, la compétence DevOps a augmenté de 50 % sur LinkedIn. Tout comme avec ITIL, un ensemble de modules de formation IP doit être produit par la communauté et mis à la disposition des nouveaux talents.

Selon nous, il serait judicieux de confier des ingénieurs junior au département IT et de les former pendant six mois dans le cadre d'une mission. Au terme des six mois, ils disposeraient des compétences nécessaires pour devenir ingénieur DevOps.

Voici les trois disciplines que nous pensons indispensables à cet effet :

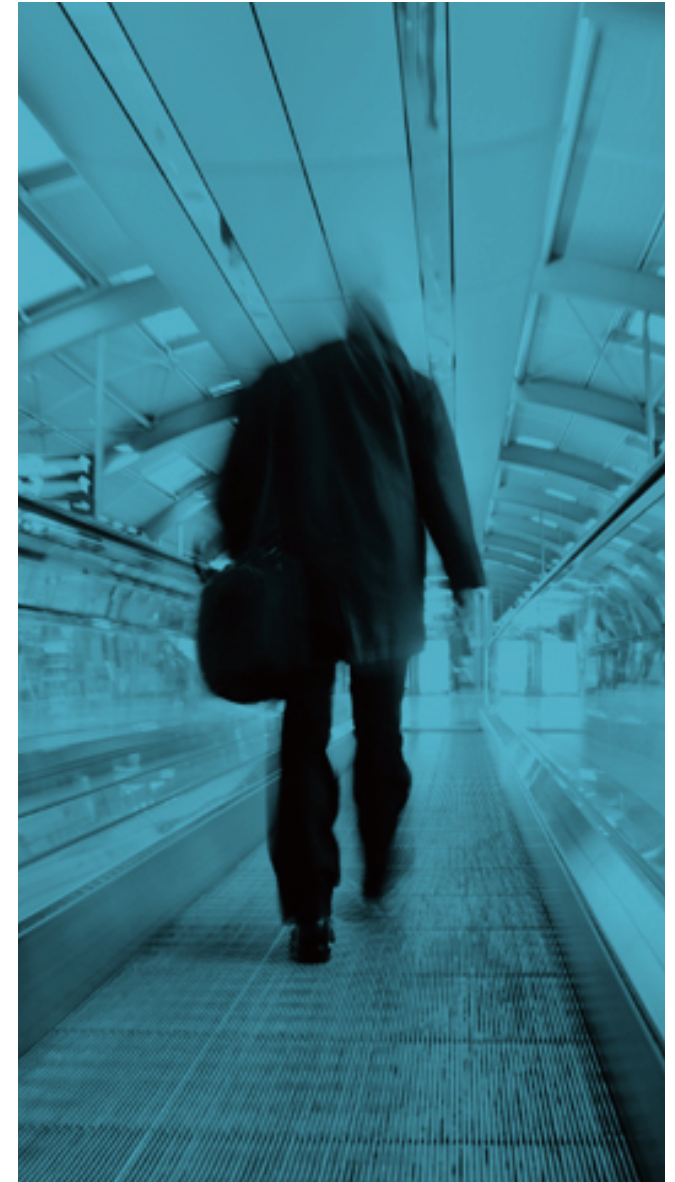
- » Compétences relationnelles et de communication
- » Traitement d'une réorganisation : la compréhension de la façon d'analyser et de réorganiser les processus métier et IT est une des compétences essentielles pour DevOps
- » Expérience des outils d'automatisation.

La méthodologie DevOps doit-elle être vendue au métier ou devons-nous simplement laisser la magie opérer ? Faut-il modifier le métier pour que la stratégie DevOps réussisse ?

L'approche DevOps doit être encouragée à la fois auprès des équipes Dev et Ops. Le métier n'en a que faire ; seuls des déploiements ultra-rapides innovants lui importent. Il faut avant tout se concentrer sur le personnel IT enraciné dans les silos. Choisissez un projet pouvant être démarré et terminé à l'aide d'une nouvelle méthodologie. Définissez les objectifs de l'équipe comme pilote et créez un journal consignnant l'activité, les problèmes et les solutions.

Quels sont les principaux obstacles à l'introduction de la méthodologie DevOps ?

Probablement faire face à la réticence des personnes pour qui il ne s'agit que d'un nouveau système de gestion des services. Citons également d'autres obstacles de taille, tels que les standards non homologués gênants (le cas échéant), les silos informatiques (et la nécessité de les éliminer) ainsi que la réprobation de la part du directeur informatique.





Robert Benefield, directeur de la technologie, Evolve Beyond

« L'approche DevOps nécessite de modifier la manière dont travaille chaque équipe, à la fois seule et avec les autres. »

À quoi ressemble une bonne approche DevOps ?

Je recherche un certain nombre d'éléments pour pouvoir déterminer l'état d'avancement d'une organisation vers la méthodologie DevOps. J'utilise une série de tests pour évaluer ou aider une organisation à faire la transition vers le DevOps, notamment en examinant s'il existe une certaine rivalité entre équipes. Le but est de déterminer si le développement, les opérations et le métier collaborent comme s'ils ne faisaient qu'un ou s'ils forment des équipes bien distinctes avec des objectifs très différents, voire contradictoires.

Nous utilisons également le test « Service Awareness » qui permet de savoir si une personne, au sein de l'entreprise, est capable de vous expliquer qui est un client donné et de quelle façon il se sert du service, ce qui est important à ses yeux et pourquoi, ainsi que la façon dont son travail contribue à améliorer cette expérience. Nous vérifions si les réponses concordent avec la réalité du terrain ; nous accordons des points supplémentaires à la personne si elle occupe un poste technique et si elle a réellement passé du temps avec le client afin d'améliorer le service.

Nous utilisons également le test « Friction » pour déterminer si la durée nécessaire à un concept ou à une idée pour passer au stade de production est bien comprise, et pourquoi. Un autre test « Risk Awareness » détermine si un client a pu aborder plusieurs personnes dans l'entreprise, du métier jusqu'au développement et à l'exploitation, et s'il a obtenu la même réponse quant à la localisation du risque, à sa sévérité et à la façon dont il est contrôlé.

Quelles sont les cinq principaux points à prendre en compte ou à éviter pour implémenter un environnement DevOps ?

1. **Ne croyez pas qu'il existe un outil ou une technologie que vous pouvez acheter pour obtenir instantanément une méthodologie DevOps.** Cette dernière nécessite de modifier la manière dont travaille chaque équipe, à la fois seule et avec les autres.

2. **Prenez en compte l'ensemble de votre cycle de vie...** Le DevOps ne se résume pas à l'agilité. Ce n'est pas parce que DevOps résulte de la fusion des termes « Development » et « Operations » qu'il se limite à ces aspects.

3. **Engagez-vous en continu...** L'engagement entre le domaine technique et le métier doit être efficace et constant afin d'améliorer la compréhension et la réactivité tout en diminuant les ambiguïtés.

4. **Analysez et améliorez les pratiques de développement...** de la structure du travail à l'installation d'un référentiel de code, aux systèmes de compilation et d'intégration, au packaging et aux tests en vue d'accroître la visibilité et le feedback tout en réduisant constamment la complexité, les changements manuels ainsi que les goulots d'étranglement.

5. **Analysez, restructurez et améliorez les pratiques opérationnelles...** de la gestion des infrastructures et de l'environnement à la gestion de la supervision, des changements et des incidents. Le processus est continu ; la visibilité et le feedback doivent s'étendre au-delà de chaque équipe afin d'améliorer la prise en compte de chaque situation et la collaboration au sein de l'organisation.



Comment faire en sorte que la prochaine génération soit préparée à la culture DevOps lorsqu'elle arrivera ? Quelles sont les compétences à développer ?

Le manque d'importance donnée à la réflexion sur le système constitue le plus grand échec qu'a connu le marché des logiciels, à tel point que j'ai l'impression qu'il a totalement perdu de vue le concept du système dans son ensemble. Afin de réduire la complexité, il existe une volonté incessante de mettre en place différentes couches d'abstraction, dans une optique de compartimentation et de simplification, souvent au détriment de la prise en compte de la réalité de l'écosystème et des rôles à y jouer.

Je rencontre de plus en plus de développeurs et d'équipes de production capables d'assembler rapidement des composants logiciels et de les exécuter, comme les rouages d'une machine, sans comprendre ce qui peut arriver ou pas à l'intérieur de la pile logicielle. Lorsque cela arrive, il existe non seulement le risque que ces personnes fassent malgré elles les mauvais choix, mais elles ont également tendance à se dégager de la responsabilité de la résolution de problèmes dont elles sont peut-être elles-mêmes à l'origine.

Le métier est ici tout aussi fautif que l'équipe technique, dans la mesure où il considère souvent l'IT comme un centre de coût qui produit des widgets techniques pouvant être assemblés et exécutés par tout le monde. Il perd de vue le fait que l'IT, comme l'électricité ou la téléphonie, est désormais le cœur du métier, si non le véritable produit procurant sa valeur au métier lui-même. Nous devons trouver le moyen de ramener la réflexion sur les systèmes interconnectés de ce monde.

La méthodologie DevOps doit-elle être vendue au métier ou devons-nous simplement laisser la magie opérer ? Faut-il modifier le métier pour que la stratégie DevOps réussisse ?

En fin de compte, le métier doit faire partie du voyage pour comprendre à la fois ce qu'est la méthodologie DevOps et la valeur qu'elle peut apporter à son organisation, et pour pouvoir en tirer pleinement parti. Les changements qu'il devra finalement effectuer lui permettront de reconnaître plus rapidement les risques inhérents et l'aideront à devancer la concurrence. Néanmoins, le moment auquel les changements doivent être introduits dépend largement de la structure et de la nature de l'organisation.

Le métier doit souvent comprendre qu'il existe une situation à améliorer et qui procurera une vraie valeur à l'entreprise. Les approches proposées autour de DevOps doivent montrer qu'elles ne se limitent pas à être des solutions probables, mais qu'elles ont également de bonnes chances de réussite dans leur environnement particulier. Certaines organisations ont la chance d'avoir en leur sein des responsables visionnaires, désireux d'effectuer des changements en investissant le temps et les efforts nécessaires pour cette transformation. D'autres ont tendance à avoir besoin de plus d'aide et de conviction pour en réaliser la valeur.

Les implémentations les plus réussies auxquelles j'ai pu assister commençaient par un essai à petite échelle, mais néanmoins significatif au sein de l'organisation. Ces essais ont souvent été initiés par des employés passionnés qui ont pu ainsi surmonter des problèmes qui les empêchaient d'améliorer nettement la réactivité et la qualité des solutions livrées.

« Les implémentations les plus réussies auxquelles j'ai pu assister commençaient par un essai à petite échelle, mais néanmoins significatif au sein de l'organisation. »

Ils ont peut-être inclus l'automatisation, mais la volonté a été nettement plus centrée sur les améliorations constantes et continues sur un choix particulier d'outils ou de technologies. La direction leur a souvent procuré un certain niveau de protection ou de soutien, suffisant pour permettre à l'équipe d'expérimenter une meilleure compréhension de la culture DevOps pour l'entreprise et la création d'une feuille de route potentielle à suivre pour les autres. La réussite de ces équipes n'a pas été fonction du nombre de déploiements qu'elles pouvaient réaliser un jour donné, mais de l'intérêt suscité chez d'autres équipes désireuses d'adopter les mêmes changements.



Quels sont les principaux obstacles à l'introduction de la méthodologie DevOps ?

Les obstacles sont nombreux. Parmi ceux que j'ai rencontrés, les plus importants incluent un manque de responsabilisation pour fournir rapidement des services de production de qualité, et procurer de la valeur au client. Ces exemples comprennent en général des développeurs réclamant des outils de déploiement et un accès à la production pour lancer plus rapidement les logiciels, sans toutefois prêter une réelle attention à l'exécution et à la capacité de prise en charge du service. Les équipes opérationnelles mettent en place des processus avec lesquels il devient de plus en plus difficile de modifier des environnements. Le personnel métier considère l'IT comme un centre de coût devant être restreint pour limiter les frais et estime que l'investissement doit être consacré aux fonctionnalités plutôt qu'aux éléments non fonctionnels.

La gestion par commandement et contrôle constitue également un problème majeur. Les équipes disposent de peu de marge pour s'organiser dans ces environnements microgérés, qui préfèrent la communication verticale, à travers la hiérarchie, à celle horizontale, entre les équipes. Il existe également un manque implicite de sens de la propriété et une certaine passivité vis-à-vis des améliorations. Peu importent les talents dont dispose l'équipe, ou les investissements en outils et en formation, le terrain qui en résulte est en grande partie stérile pour la culture DevOps.

Celle-ci pourra difficilement se développer là où de profonds silos organisationnels vont de pair avec des objectifs peu partagés. Nous constatons également qu'une compréhension mal partagée de l'importance et de la valeur des mesures ainsi que de la conscience situationnelle, couplée à un manque d'intérêt pour une amélioration continue, engendre des problèmes.

« Un manque de compréhension ou d'envie d'agilité du service IT se révèle également extrêmement problématique. »

Certaines équipes se satisfont pleinement du statu quo et ne voient pas l'intérêt de comprendre la situation ni de l'améliorer. Elles sont souvent persuadées de connaître la seule et unique façon de procéder et que le moindre problème ou désaccord avec un client ne les concerne pas.

Un manque de compréhension ou d'envie d'agilité du service IT se révèle également extrêmement problématique. Il existe encore un nombre (étonnamment) important de métiers à qui la valeur de l'agilité du service IT échappe totalement, soit parce qu'il existe toujours des éléments monopolistiques ou peu de concurrence dans le secteur à cause d'une évolution relativement lente de celui-ci, ce qui limite le choix des clients, soit parce que la friction est suffisante et génère des coûts élevés de changement de fournisseurs pour les clients. Ce nombre continuera de diminuer au fur et à mesure que l'innovation et la mondialisation supprimeront les barrières à l'entrée des marchés.





Benjamin Wootton, cofondateur, Contino

À quoi ressemble une bonne approche DevOps ?

J'ai tendance à voir une bonne approche DevOps du point de vue du développement étant donné que je me trouve de ce côté du « mur ». Cela suppose essentiellement la conception et le développement de logiciels qui mettent nettement l'accent sur ce qui est traditionnellement considéré comme les exigences et préoccupations de la production.

Par exemple, les développeurs doivent se demander de quelle façon leur logiciel peut être mis en production en toute sécurité, rétabli à un stade antérieur si nécessaire,

« L'approche DevOps doit être une initiative ascendante pour être une réussite. »

supervisé, mis à niveau, géré, mis à échelle et pris en charge au quotidien. Les maquettes et architectures doivent être mises en place pour prendre en charge ces exigences non fonctionnelles et incorporées dans le logiciel à travers toutes les réalisations de l'équipe de développement.

Les développeurs doivent aussi avoir clairement conscience de leur infrastructure de production en comprenant ces environnements et la façon dont ils interagiront avec leur application. Une bonne utilisation de la virtualisation et de l'automatisation des infrastructures permettra aux équipes de développement de gagner en compréhension, en visibilité, en contrôle et en responsabilisation vis-à-vis de l'infrastructure qui prend en charge leur logiciel.

Le problème culturel est ici très important, car les équipes de développement doivent travailler en partenariat avec la production et faire attention au logiciel livré, plutôt que d'envoyer négligemment le code aux testeurs ou à la production. Les développeurs travaillant avec la mentalité DevOps ne verront pas leur travail terminé tant que le logiciel n'en sera pas au stade de production, voire même tant qu'il ne sera pas retiré de la production à la fin de sa vie utile. Ce changement de perspective au sujet de la responsabilisation est subtil, mais d'une importance extrême.

Les équipes de production sont manifestement encore bien loin d'une bonne approche DevOps, mais je pense que nous autres développeurs le sommes encore plus.

La méthodologie DevOps doit-elle être vendue au métier ou devons-nous simplement laisser la magie opérer ? Faut-il modifier le métier pour que la stratégie DevOps réussisse ?

La transition vers la méthodologie DevOps nécessite des changements culturels, organisationnels, de processus et d'outillage technique ; tous sont requis pour pouvoir optimiser les bénéfices.

Je pense que les personnes ou équipes passionnées qui font localement la transition vers un mode de travail DevOps dégagent une grande valeur. La mise en place d'une gestion de la configuration ou la suggestion d'ajustements des processus métier peut certainement délivrer de la valeur et améliorer localement la situation. Tout ceci est formidable, car je pense qu'en fin de compte, l'approche DevOps doit être une initiative ascendante pour être une réussite.

Cela dit, pour pouvoir exceller et obtenir les bénéfices définitifs de l'approche DevOps, nous devons aller encore plus loin, et certains défis ne peuvent être relevés que grâce au soutien de la direction. La culture en matière de responsabilisation et de collaboration doit changer. Les structures d'incitation doivent être modifiées pour que chacun se dirige dans la même direction. Les structures organisationnelles doivent être modifiées pour pouvoir éliminer les silos et permettre une collaboration pragmatique. Seuls les cadres supérieurs disposent d'une telle influence.

En somme, la meilleure situation se résume probablement à la mise en place, par la direction, d'un environnement où la stratégie DevOps peut se développer de bas en haut.



Quelles sont les cinq principaux points à prendre en compte ou à éviter pour implémenter un environnement DevOps ?

L'implémentation d'une équipe DevOps distincte est l'erreur la plus courante que j'ai pu rencontrer. Il s'agit simplement de la mise en place d'un troisième silo qui a tendance à se concentrer sur l'outillage, sans obligation ni tentative d'effectuer des changements culturels ou organisationnels.

J'ai rencontré une équipe DevOps qui, en réalité, nuisait aux performances et à la collaboration au sein de l'organisation. En général, il vaut mieux intégrer directement à vos équipes de développement et de production les personnes disposant de l'approche, de la mentalité et des compétences appropriées.

De la même manière, le fait de donner le nom DevOps à une administration système ou à une équipe de support n'est pas la solution ; la perspective vis-à-vis de l'outillage changera peut-être, mais c'est loin d'être suffisant.

L'intégration d'un mode de travail DevOps à une organisation en adoptant une perspective descendante est également vouée à l'échec. Vous pouvez ordonner à votre personnel d'assister à des réunions, d'utiliser certains outils ou de travailler de telle manière, mais s'il conserve une mentalité de silo, cette initiative est vouée à l'échec.

Je pense que le plus efficace est de laisser le personnel sur le terrain prendre les décisions et ses responsabilités. Le fait de laisser vos équipes choisir les outils qu'elles utilisent et leur façon de travailler vous permettra généralement d'aboutir aux meilleures solutions, les plus pragmatiques et collaboratives.

« Je ne pense pas qu'il y ait une pénurie de personnel aux compétences techniques, mais peu comprennent cette vision d'ensemble »

Comment faire en sorte que la prochaine génération soit préparée à la culture DevOps lorsqu'elle arrivera ? Quelles sont les compétences à développer ?

Il existe actuellement un vaste fossé vis-à-vis des codeurs d'infrastructures. Ce sont des personnes capables d'utiliser des outils d'automatisation d'infrastructures et une approche de développeur pour la configuration de serveurs, d'infrastructures et d'outillages via des API.

Avec l'expansion du modèle Cloud, les batteries de serveurs sont adaptées au fur et à mesure à d'énormes volumes de serveurs Cloud, et les anciennes techniques de gestion sont tout bonnement dépassées. L'ensemble du secteur des opérations logicielles doit simplement avancer dans cette direction.

Il manque également des personnes ayant une très bonne compréhension à la fois de la compilation et du déploiement. Ce n'est pas toujours un domaine très attirant, mais la réussite dans ce dernier permet aux organisations de développement d'avancer plus rapidement et avec une qualité supérieure, engendrant de ce fait un réel avantage concurrentiel. Le mouvement de livraison en continu (« Continuous Delivery ») apporte véritablement cette visibilité.

En fin de compte, les personnes travaillant dans le secteur avec la bonne mentalité DevOps ne sont actuellement pas assez nombreuses. Beaucoup d'ingénieurs viennent me demander conseil en DevOps dans le cadre de Contino, mon cabinet ; je ne pense pas qu'il y ait une pénurie de personnel aux compétences techniques, mais peu comprennent cette vision d'ensemble et sont capables de mener l'organisation à travers les changements culturels, organisationnels, de processus et techniques.



Dave Farley Co-auteur de *Continuous Delivery*, architecte, KCG Ltd

À quoi ressemble une bonne approche DevOps ?

Une bonne approche DevOps nécessite l'application d'une méthode scientifique au développement logiciel. Nous essayons de parvenir à un processus vérifiable et répétable, nous permettant d'établir des boucles de feed-back efficaces. Formuler une hypothèse, la tester par le biais d'une expérience, agir en fonction des résultats et passer à la suite.

La méthode scientifique évoquée précédemment est tout simplement la meilleure approche jamais trouvée pour la résolution de problèmes. C'est la raison pour laquelle ces techniques fonctionnent. Appliquer notre technique de

résolution des problèmes la plus efficace à un problème pour le moins épineux : le développement logiciel. L'une des caractéristiques de cette approche est l'exécution fréquente de ce cycle. Cette itération rapide limite les dégâts que peuvent entraîner de mauvaises suppositions et permet de délivrer très rapidement de la valeur. La durée du cycle est une mesure clé de l'efficacité de votre approche DevOps. Pensez au temps que prendrait la moindre modification dans votre série actuelle de vérifications régulières, et quel qu'il soit, essayez de le réduire.

La brièveté de ce cycle de feed-back est un indicateur clé de réussite. En réalité, les cycles de feed-back se chevauchent. À un niveau affiné, il existe le cycle de feed-back TDD (Test Driven Development) qui doit être court et efficace. Cela signifie que vous devez élaborer de petits tests, assister à leur échec, puis réécrire le code pour leur permettre de réussir, observer le résultat et poursuivre. À l'extérieur se trouve le cycle qui nous intéresse réellement : trouvez une idée, mettez-la entre les mains de vos utilisateurs ou de vos clients, voyez si elle fonctionne et recommencez !

Une bonne stratégie DevOps doit également travailler au renforcement du cycle de feed-back métier, le métier et l'équipe technique devant devenir expérimentateurs. Le développement d'une mentalité et d'un potentiel d'expérimentateur est une étape importante pour les entreprises, pas uniquement pour les équipes de développement. Il s'agit toutefois d'un changement non négligeable, car elles n'ont pas l'habitude de travailler de cette façon. Les difficultés résident dans le changement culturel. Il existe ce que j'appelle une « illusion de rigueur » ; l'obtention d'une réflexion, d'une planification et d'une préparation étendues en amont semble être un processus bien rigoureux. En réalité, nous autres humains sommes tellement peu doués en suppositions que ce n'est pas entièrement vrai. Vous êtes plus rigoureux lorsque vous réalisez une itération affinée.

Devons-nous mettre en place un langage commun pour unir les équipes Dev et Ops afin de permettre la réussite de l'approche DevOps ?

De nombreuses idées évoluent et se répandent lorsqu'une personne ou un groupe de personnes arrive à décrire ce que le personnel effectue déjà et lorsque cette description est adoptée. Toutefois, je doute que les différences de terminologie ou de langage entre les rôles Dev et Ops posent problème. S'il existe un problème de communication, vous devez rassembler tout le monde dans une seule et même pièce pour abattre les murs. De nombreux éléments sont perdus lors de la traduction ; toutefois, si vous travaillez vraiment ensemble, quand bien même vous ne comprenez pas ce que les autres disent, cela se lit dans les expressions lorsque vous êtes face à face.

« La durée du cycle est une mesure clé de l'efficacité de votre approche DevOps. »



Quelles sont les cinq principaux points à prendre en compte ou à éviter pour implémenter un environnement DevOps ?

Il est fondamental d'éliminer les silos et de vous assurer que chacun œuvre en vue d'un objectif commun. Vous devez l'appliquer à l'organisation des équipes, du travail et des objectifs. Quelle que soit sa mission, chacun doit partager l'objectif ultime d'entraîner les changements jusqu'au stade de production ; tout le reste n'a que peu d'importance en comparaison.

Même si la voie vers l'objectif est alors tracée, les processus créés doivent être répétables et fiables, ce qui signifie que leur degré d'automatisation doit être élevé. Les mécanismes de validation, de vérification et de mise en production de votre logiciel ne doivent pas dépendre d'une prise de décision humaine, car les processus ne seront alors ni répétables ni fiables. Utilisez le personnel pour ce qu'il sait bien faire, à savoir la prise de décisions subjectives complexes, et utilisez les machines/l'automatisation pour leur spécialité : la répétabilité et la fiabilité.

Le chemin vers la mise en production doit être automatisé pour pouvoir compter sur le processus afin que le déploiement ne dépende pas d'un jugement humain pour déterminer si votre logiciel est prêt ou non. La décision de déploiement doit être prise lorsque vous avez fait tout ce qui était en votre pouvoir, sur la base de l'automatisation, pour montrer que le code est prêt (que tout indique qu'il n'y a pas la moindre erreur, notre approche scientifique ne permettant pas de prouver que le code fonctionnera au stade de production, vous n'aurez jamais assez de tests pour le savoir, mais si l'un d'eux échoue, vous savez que

le code n'est pas prêt pour passer en production) ; il s'agit déjà là d'un grand pas en avant.

« Il est fondamental d'éliminer les silos et de s'assurer que chacun œuvre en vue d'un objectif commun. »

Comment faire en sorte que la prochaine génération soit préparée à la culture DevOps lorsqu'elle arrivera ? Quelles sont les compétences à développer ?

Nous devons enseigner la science en général, mais également la science informatique. En tant que secteur, le fait de ne pas avoir su tirer des leçons des problèmes déjà résolus est terrible pour nous. Nous échouons souvent à construire sur les épaules de géants ; nous ne disposons pas de la culture appropriée pour ce faire. Au lieu de cela, nous abordons des problèmes résolus il y a des décennies par des personnes très intelligentes comme s'ils étaient nouveaux, souvent à l'aide de méthodes folkloriques et rapides, plutôt que d'essayer de réellement comprendre le problème à résoudre.

Le personnel a besoin de bases non seulement quant aux fondamentaux de la science informatique, mais également quant aux principes de la science et de la méthode scientifique. Nous devons apprendre à itérer, à adopter une approche sceptique et à toujours rester dans le doute (à concevoir des expériences et à les mettre en pratique, même s'il s'agit de votre propre idée).

Richard Feynman, prix Nobel de physique, est l'auteur d'une de mes citations préférées :

« Peu importe votre degré d'intelligence, si vous faites une supposition que vous ne pouvez pas étayer par une preuve expérimentale, cela reste une supposition ! »

Au lieu de cela, nous apprenons trop souvent par cœur. En fin de compte, chaque projet débute de la même manière : établissement de la base de données, installation des couches système, puis décision du langage à utiliser... jusqu'à ce que quelqu'un finisse par demander : « Au fait, quel était le problème ? »

Les projets que j'ai mis en route (notamment dans des environnements à hautes performances) en élargissant la recherche de solutions et en adoptant un état d'esprit ouvert se sont souvent soldés par des solutions plus simples et plus nettes que prévu ; il a suffi de prendre un nouveau départ et de réfléchir au problème fondamental. Que l'écriture de code ou la réalisation de tests soit avérée, cette approche de résolution des problèmes n'est évidemment pas propre à la méthodologie DevOps.



La méthodologie DevOps doit-elle être vendue au métier ou devons-nous simplement laisser la magie opérer ? Faut-il modifier le métier pour que la stratégie DevOps réussisse ?

Absolument, le métier doit changer ! La plupart des organisations bénéficieront d'une approche rationnelle pour tous les types de développements. Il est essentiel de se demander : « Est-ce qu'elle rapportera vraiment de l'argent, nous sera-t-elle utile, ou bien est-il simplement question d'ajouter une fonctionnalité réclamée par tout le monde ? ».

Il s'agit ici d'avoir une approche expérimentale. Celles qui s'en sortent le mieux à ce niveau sont des entreprises agiles, non des organisations de développement agiles. Cette approche intervient à la base de l'entreprise et devient fondamentale à la prise de décision. Pour beaucoup d'organisations, en arriver à ce stade constitue un énorme changement qui est toutefois nécessaire pour pouvoir dégager la valeur la plus élevée possible à partir de cette approche du développement.

« Les organisations qui adoptent cette approche en sont absolument ravies. »

De nombreuses personnes ont passé une grande partie de leur carrière dans des secteurs et des environnements au sein desquels le développement n'était à vrai dire pas très bon et qui ne délivraient pas le maximum de valeur possible ; il s'agit là du principal obstacle. Dans beaucoup de métiers, les logiciels sont devenus un goulot d'étranglement à cause d'une part de son importance, d'autre part de son aspect difficile et créatif. La gestion traditionnelle est un frein à la créativité. Vous ne faites pas un film en établissant les règles à l'avance, il s'agit d'un processus créatif résultant de la fusion d'idées ; il en va de même pour les logiciels.

Quels sont les principaux obstacles à l'introduction de la méthodologie DevOps ?

Les obstacles techniques à l'approche DevOps peuvent être une réalité. Si vous disposez d'une vaste base de code propriétaire, il peut être difficile de réaliser une automatisation de tests suffisante. Ces défis techniques sont toutefois facilement surmontables et presque insignifiants comparés aux changements culturels nécessaires.

Le changement culturel est le plus difficile à réaliser. Il est difficile de modifier l'opinion de personnes ayant construit et réussi leur carrière en agissant de la mauvaise manière. Il est perturbant et réellement problématique de le reconnaître, car c'est autant un défi personnel et émotionnel qu'une remise en cause de ses compétences. La conception d'expériences appropriées est également difficile, c'est une compétence longue à acquérir. Toutefois, toutes les organisations ayant adopté cette approche en sont absolument ravies. Elle est plus productive, de meilleure qualité et bien plus plaisante, malgré les difficultés et les défis qu'elle peut parfois poser.



Biographies

Paul Speers

PDG, Speerhead Group

Paul Speers est le PDG de Speerhead. Il a fondé l'entreprise il y a plus de 5 ans pour mener Speerhead sur la voie de la croissance à long terme, poursuivre sa réussite commerciale sur le marché DevOps et diriger le lancement de DevOps Recruitment, sa franchise révolutionnaire ; il a également joué un rôle prépondérant dans la création de formations DevOps et de Certification IP.

Paul n'est pas uniquement axé sur la domination du marché DevOps ; il se concentre également sur les réussites de la base client globale de l'entreprise ainsi que des solutions de recrutement pour les booms de croissance. Paul est un intermédiaire essentiel entre les clients de Speerhead, ses partenaires de franchises globales ainsi que l'ensemble du secteur DevOps.

Paul apporte à Speerhead plus de 20 ans d'expérience dans la vente et le marketing IT, ayant occupé des postes senior chez Opware, le premier fournisseur d'automatisation informatique de Marc Andersson. Il est également le cofondateur de Fox IT, le fournisseur de Global ITIL.

Robert Benefield

Directeur de la technologie,
Evolve Beyond

Robert Benefield dispose de plus de 20 ans d'expérience en tant que directeur exécutif, années qu'il a passées à construire et à diriger des organisations d'opérations techniques et d'ingénierie fine à hautes performances de classe mondiale, dans des environnements extrêmement exigeants, comme la banque d'investissement, la défense, les télécommunications et les services Internet.

Robert a mené des transformations en utilisant Agile et Lean ; il a développé et implémenté avec brio les meilleures techniques informatiques Cloud et flexibles dans de nombreux environnements complexes. Robert aime résoudre des problèmes complexes, utiliser de façon créative la technologie et les techniques organisationnelles pour apporter un nouveau degré de compréhension et de dynamisme au sein des entreprises et de leurs marchés.

Benjamin Wootton

Cofondateur, Contino

Benjamin est le cofondateur de Contino, un cabinet de consulting basé à Londres, spécialisé dans l'application de pratiques DevOps et Continuous Delivery à des projets de lancement de logiciels.

Il est à la tête du bulletin d'informations populaire « DevOps Friday » et est en pleine écriture d'un ouvrage, « *Flowing Software* » (*Logiciel fluide*), publié en 2014. Il dispose de plus de dix ans d'expérience auprès d'équipes de développement et de production de logiciels flexibles, dans de nombreux secteurs.

Dave Farley

Co-auteur de « *Continuous Delivery* », architecte, KCG Ltd

Dave Farley est le co-auteur de « *Continuous Delivery* », ouvrage lauréat du prix Jolt. Les ordinateurs sont sa passion depuis plus de 30 ans. Durant cette période, il a d'ailleurs travaillé sur la plupart des types de logiciel. Il dispose d'une vaste expérience en matière de direction du développement de logiciels complexes en équipes, grandes ou petites.

Dave a très vite été adepte des techniques de développement agiles : dès le début des années 1990, il a utilisé le développement itératif, l'intégration continue et des niveaux significatifs de tests automatisés sur des projets commerciaux. Plus récemment, Dave a travaillé au développement de logiciels à hautes performances pour le secteur de la finance, dans le domaine de l'informatique à faible latence. Dave travaille actuellement pour KCG Ltd.



Étapes suivantes

La généralisation de la méthodologie DevOps est en marche. Votre organisation est-elle prête à saisir tous les avantages commerciaux et toutes les opportunités que cela représente ? Chez CA Technologies, nous avons conçu un portefeuille de produits et de solutions basés sur notre expertise DevOps.

Rendez-vous sur le site ca.com/contact pour en savoir plus sur la façon dont CA Technologies peut vous aider à combler le fossé entre vos développeurs et votre production, ainsi qu'à conserver votre avantage concurrentiel dans l'économie appliquée.

Pour plus d'informations sur les solutions DevOps de CA Technologies, rendez-vous sur : www.ca.com/insights/devops

#ReWrittenBySoftware

Copyright ©2014 CA. Tous droits réservés. Tous les noms et marques déposées, dénominations commerciales, ainsi que tous les logos référencés dans le présent document demeurent la propriété de leurs détenteurs respectifs. Les informations du présent document sont tirées d'entretiens organisés par CA avec les personnes susmentionnées. CA n'a ni vérifié ni validé les informations et/ou faits présentés, mais offre le présent document à titre d'information. Les informations du présent document ne font l'objet d'aucune garantie et sont susceptibles d'être modifiées.