

# Top 7 des pratiques d'excellence DevOps

La vitesse est au coeur du DevOps : on veut accélérer le développement d'applications, les modifications, les mises à jour et le développement continu, on veut livrer plus vite et raccourcir ainsi les cycles de développement de systèmes. Selon des études récentes, les équipes de développement qui suivent les bonnes pratiques DevOps et qui sont les plus performantes, restaurent en moyenne 96 fois plus vite un système après son interruption. Le taux d'échec de leurs modifications s'avère aussi cinq fois moins élevé et leurs déploiements de code 46 fois plus fréquents. Les équipes sont capables de déployer du code plusieurs fois par jour plutôt qu'une fois par semaine. En résumé, les équipes DevOps qui réussissent peuvent déployer du code plus rapidement et, en cas d'interruptions, leur reprise d'activité n'est plus qu'une question d'heures et non de jours.

Il n'est pas étonnant que tant d'organisations de toutes tailles, dans toutes les verticales et dans toutes les régions du monde aient adopté DevOps. Une autre étude nous apprend que plus des trois quarts des organisations interrogées dans le monde avaient déjà adopté DevOps. Toutefois, cet enthousiasme et le potentiel de DevOps ne doivent pas occulter certains défis bien réels. La même étude révèle que le tout premier défi auquel se heurte DevOps est l'élimination des goulets d'étranglement afin d'accélérer la publication des versions successives. Rationaliser la collaboration et sélectionner la suite adéquate d'automatisation et les autres outils arrivent immédiatement après. Le rapport énumère les sept grandes règles DevOps que suivent les équipes performantes pour tirer le maximum de leurs investissements financiers et personnels.

## 1 La culture DevOps n'est pas négligeable, surtout lorsqu'elle vient du terrain.

L'auteur du panorama annuel "State of DevOps Survey", actuellement dans sa septième année, écrit : « Pour nous, l'essentiel a toujours été que les équipes aient les moyens de donner toute la mesure de leur talent, en dépassant les différences culturelles entre développement et opérationnel. » Dans cette étude, l'analyse approfondie des données historiques montre que la réussite d'un projet DevOps est le plus souvent la suite logique d'efforts modestes issus de la base. Ensuite, les développeurs ont besoin d'une plateforme légère où relayer en toute sécurité aux autres équipes leurs premiers succès et leurs pratiques éprouvées. Cette façon de faire se répand alors aux autres équipes du même service, puis aux autres services. En d'autres termes : ne vous lancez pas d'entrée de jeu dans des efforts démesurés. Cette pratique est précisément celle que suivent **le plus grand fabricant de logiciels d'entreprise au monde et un leader mondial de la logistique.**

## 2 Intégrer la sécurité au coeur de la démarche DevOps.

Il s'agit là d'automatiser au maximum les configurations de sécurité. Au fur et à mesure que les organisations DevOps s'éloignent de leurs débuts modestes, la politique de sécurité devient inhérente aux opérations et non un élément que le service IT essaie de prouver in extremis pour rentrer dans les critères d'audit. [GitHub](#), première plateforme de développement logiciel au monde, s'inscrit dans l'infrastructure en place de l'organisation. Elle est donc régie par les contrôles locaux de sécurité de l'information, notamment les pare-feux, les réseaux privés virtuels (VPN), les systèmes de gestion des accès et des identités (IAM) et de supervision. Cette solution sur site participe à éviter de nombreux problèmes de conformité légale et réglementaire courants dans les solutions Cloud. Au lieu de s'évertuer à tenir les auditeurs à distance, le service IT peut se concentrer sur la sécurité des données de l'entreprise et de ses clients. Et surtout, les équipes et les **développeurs peuvent créer plus rapidement un code plus sécurisé.**

## 3 Automatiser au maximum.

Les débats sur l'automatisation dans un projet DevOps se concluent habituellement par un accord sur l'automatisation des workflows, de la configuration et du provisioning des systèmes. L'automatisation de l'infrastructure s'attaque à une difficulté courante, à savoir le rythme du développement qui doit rester en phase avec les capacités opérationnelles, directement liées à la capacité à déployer le code. L'automatisation fait disposer d'un autre atout : la création lors des phases de développement ultérieures d'un libre-service plus largement disponible, source d'encore plus d'efficacité. À l'aide d'un ensemble de modèles, GitHub permet aux utilisateurs de **configurer automatiquement des workflows** pour garder le statut des project boards en phase avec les issues et pull cards associées. En programmant des actions automatiques sur la base d'événements déclencheurs, les utilisateurs éliminent des tâches manuelles et chronophages de gestion du project board. Les utilisateurs peuvent également le copier et réutiliser ses adaptations dans des projets similaires.

## 4 Tirer profit des atouts d'une plateforme ouverte qui apporte la transparence.

La collaboration, un des principaux vecteurs d'efficacité de l'approche DevOps, régit le milieu de l'open source où le développement logiciel est le fruit d'un processus extrêmement collaboratif, collégial et ouvert à tous.

Ces dernières années, la hausse des recettes de l'open source dans le monde a été considérable à mesure que les organisations de tout type intégraient en masse l'open source à leur méthodologie DevOps. Avec les plateformes logicielles Web, il est facile de remonter la trace et d'effectuer le suivi des actions d'un très grand nombre d'autres développeurs, où qu'ils se trouvent. Cette ouverture porte en elle le potentiel d'une transparence optimale susceptible d'améliorer drastiquement la collaboration et l'acquisition des connaissances dans le milieu extrêmement complexe du développement logiciel. Une étude montre que, grâce à cette activité en réseau, les utilisateurs établissent des liens et des inférences sociales forts qui s'entretiennent. Par exemple, ils arrivent à inférer la vision et les objectifs techniques de quelqu'un d'autre lorsqu'ils modifient le code, ou ils anticipent mieux les chances de réussite à long terme des projets. Les utilisateurs combinent alors ce type d'inférences en stratégies d'une efficacité accrue sur deux axes : coordination du travail et développement de leurs compétences techniques. Avec plus de 2,1 millions d'organisations et 40 millions de développeurs, la communauté open source des utilisateurs GitHub est la plus grande au monde. Ils mutualisent le code, collaborent étroitement à toutes les phases de la méthodologie DevOps et, au bout du compte, élaborent des logiciels avec un maximum d'efficacité et de détermination, dans le cadre d'une plateforme intuitive et extrêmement conviviale.

## 5 Intégrer avec le plus grand nombre d'outils possible.

Dans la version la plus récente d'une enquête annuelle sur les déploiements DevOps entreprise, les deux barrières principales à l'efficacité accrue des processus DevOps sont humaines : la suppression des goulets d'étranglement dans les modes opératoires et la rationalisation de la collaboration entre développeurs. Mais juste après figurent la sélection et l'implémentation d'outils de supervision, de gestion et d'automatisation de l'application. Le succès d'une démarche DevOps repose en grande partie sur la capacité à intégrer les efforts DevOps à la plus large gamme d'outils possible. Qu'il s'agisse de promouvoir la modification ou la révision continue du code ou encore l'intégration continue, il est indispensable que les développeurs utilisent les bons outils à chaque étape du processus de développement. GitHub ouvre littéralement l'accès à **des centaines d'outils** conçus pour aider les équipes de développement à mieux communiquer, à automatiser leur travail et à développer de meilleurs logiciels. Il arrive parfois que seul un outil fait sur mesure puisse réaliser une tâche spécifique. GitHub donne aux utilisateurs les moyens de créer de tels outils personnalisés par un meilleur accès aux données grâce à l'**API GraphQL** de GitHub, la même que l'entreprise utilise pour élaborer GitHub.

## 6 Inner sourcing : maximiser l'efficacité de l'open source

De plus en plus d'organisations adoptent et suivent la méthodologie de développement dite **inner sourcing**, qui reprend les bonnes pratiques des projets open source à grande échelle. Les développements extrêmement complexes et aux multiples branches de ce type exigent la coordination transversale de centaines sinon de milliers de développeurs et d'équipes. Suivre l'approche de l'inner source se révèle une arme redoutable pour venir à bout des nombreux défis des projets DevOps et un excellent moyen de booster l'efficacité en recyclant le code. Les entreprises peuvent accéder aux bonnes pratiques open source et les mobiliser derrière le pare-feu de l'inner source. GitHub accueille la plus grande communauté open source au monde. Les développeurs et les équipes de la plateforme GitHub peuvent voir, modifier et distribuer un projet dans le but de leur choix, en accord avec les licences open source. Et GitHub met à disposition toute une série de guides en libre-service pour trouver des utilisateurs pour les projets, former des communautés accueillantes, mettre en oeuvre la maintenance et la gouvernance et déterminer des indicateurs open source, pour ne donner que quelques exemples.

## 7 S'inscrire dans une démarche « en continu » : retour d'information, intégration, gestion des changements et déploiement continu.

Il est bon de connaître les pratiques communes et les normes du secteur DevOps, mais **l'important est de les intégrer et de les incorporer** à un flux de processus qui traverse les silos de données et d'informations du développement aux opérations en passant par le déploiement. Ainsi, l'intégration continue va au-delà de l'automatisation des tests lors de l'ajout de code à une application : elle s'intéresse à la qualité au niveau applicatif lors de l'intégration du code nouveau ou modifié. La gestion des changements fait intervenir les opérations qui vont recueillir des informations sur les éventuels autres systèmes impactés et les conséquences ou opportunités qu'un changement peut amener à plus grande échelle. Les stratégies de déploiement incrémentielles sont souvent victimes de contraintes opérationnelles qui restreignent le déploiement continu voire en empêchent la mise en place. Ces entraves sont humaines et non technologiques. Y remédier passe par l'inclusion de tous les partenaires dans le développement des applications, les opérations et l'assistance. Les contributions transversales et continues de ce type vont considérablement renforcer la probabilité qu'une bonne application réponde aussi aux besoins et aux attentes des utilisateurs.

GitHub est le moyen d'élaborer des logiciels. Il aide les équipes DevOps de toutes tailles à collaborer en toute sécurité et à proposer plus vite de meilleures expériences client. Plus de 40 millions de développeurs et la moitié des entreprises du classement Fortune 500 lui font confiance. Pour commencer votre essai gratuit ou en savoir plus sur les solutions GitHub sur site ou à la demande, rendez-vous sur <https://github.com/business>.

<sup>1</sup> "2017 State of DevOps Report," Puppet, 2017.

<sup>2</sup> "2018 State of DevOps Report: Practical guidance for your DevOps evolution," Puppet blog, September 2018.

<sup>3</sup> "Social coding in GitHub: Transparency and Collaboration in an Open Software Repository," jsntsay.com, 2012.