

# DECRYP'TECH

## LE DEVOPS

### PAR CAROLE, CONSULTANTE IT



*DevOps est un ensemble de pratiques et d'outils, ainsi qu'une philosophie culturelle. Son but est d'automatiser et d'intégrer les processus entre les équipes de développement et informatiques opérationnelles. Une démarche DevOps met l'accent sur l'autonomisation des équipes, la communication et la collaboration transverses ainsi que l'automatisation technologique.*

#### HISTORIQUE

Le mouvement DevOps a débuté vers **2007** lorsque les équipes informatiques opérationnelles et de développement ont exprimé leurs inquiétudes quant au modèle traditionnel de développement, dans lequel les programmeurs travaillaient à l'écart des équipes opérationnelles chargées de déployer le code et d'en assurer le support. Le terme DevOps, qui combine les mots « développement » et « opérations », reflète le processus d'intégration de ces disciplines en un processus continu.

#### LES AVANTAGES THEORIQUES

Dans l'enquête sur les tendances DevOps de 2020 d'Atlassian, 99 % des personnes interrogées ont déclaré que DevOps avait un impact positif sur leur organisation. Parmi les avantages de DevOps, citons l'accélération/la simplification des livraisons, l'efficacité des équipes, la sécurité accrue, les produits de meilleure qualité et une plus grande satisfaction des équipes et des clients qui en découle.

- ★ Vitesse
- ★ Collaboration améliorée
- ★ Déploiement rapide
- ★ Qualité
- ★ Fiabilité
- ★ Sécurité

### LES 5 GRANDS PRINCIPES DE LA DEMARCHE DEVOPS

**La culture de collaboration** : Elle permet de donner aux équipes DEV et OPS une vision globale du Système d'Information et donc de fluidifier et de faciliter leur communication.



**La création de valeur**: elle se fait avec la mise en place de cycles d'amélioration continue, le rapprochement des différentes équipes et leur concentration sur la valeur business et la satisfaction des utilisateurs de l'application.



**La mise en place de métrique** : Elle est nécessaire pour observer l'amélioration en mesurant l'évolution entre un état initial et un état final. Les métriques métier permettent la mesure de la vitalité du business et les métriques d'équipe leur performance. La mesure de la satisfaction client permet ensuite l'amélioration des produits.



**Le partage** : Les différentes équipes travaillent ensemble en partageant l'objectif commun de la création de valeur mais aussi en partageant l'information dans le travail quotidien : les connaissances respectives, les retours d'expérience, les bonnes pratiques, la communication sur les éventuels incidents, problèmes. Ceci permet de développer l'entraide, l'esprit inter-équipes et d'inspirer de nouvelles idées grâce aux points de vue différents des DEV et OPS.



**L'automatisation des processus** : elle se fait au niveau du provisioning (paramétrage) d'environnement, des tests pour valider la qualité attendue du nouveau code et du déploiement de l'application. L'automatisation de ces trois processus doit permettre de converger vers le déploiement continu. Un ensemble d'outillage permet l'automatisation à ces différents niveaux comme par exemple Docker pour le provisioning, Selenium pour les tests fonctionnels et Puppet pour le déploiement.



# Fonctionnement d'une entreprise de développement et de mise en exploitation / maintenance d'une application

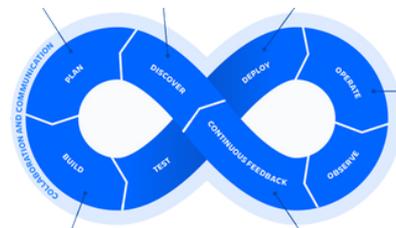
## SANS DEMARCHE DEVOPS

Les Développeurs (Dev) et les Opérationnels (Ops) ont des objectifs différents. Les développeurs doivent faire évoluer rapidement l'application pour répondre aux demandes du marché en cherchant à réduire le Time To Market (le TTM = délai entre l'idée initiale et sa concrétisation en production). Dans le contexte économique concurrentiel actuel un TTM plus court permet à une entreprise de s'adapter et de répondre rapidement à la demande du marché qui ne cesse d'évoluer et aussi à proposer de l'innovation au client avant les concurrents. Les opérationnels ont pour objectif de garantir la stabilité du système par un contrôle des changements apportés au système d'information, sachant qu'une panne de production peut coûter cher à l'entreprise (business perdu, contraintes légales non respectées, atteinte de l'image de l'entreprise). **Dans cette configuration les Dev et les Ops ont un conflit d'intérêt dû à l'antagonisme de leur objectif.** Par absence de communication les Dev peuvent blâmer les Ops pour des retards et problèmes de livraison et les Ops peuvent blâmer les Dev pour les incidents de production liés à une mauvaise qualité du code.



## AVEC DEMARCHE DEVOPS

L'approche DevOps cherche à réconcilier les mondes Dev et Ops. Le but est l'aligner leurs objectifs à court terme autour d'un objectif commun sur le long terme c'est-à-dire la création de valeur pour l'entreprise qui repose sur trois indicateurs : un Time To Market court, des produits de qualité et des équipes plus efficaces.



En raison de la nature continue du DevOps, les experts utilisent **la boucle infinie pour indiquer comment les phases du cycle de vie DevOps sont liées les unes aux autres.** Bien qu'elle semble se dérouler de manière séquentielle, la boucle symbolise la nécessité de collaboration constante et d'amélioration itérative tout au long du cycle de vie.

**Le cycle de vie DevOps comprend huit phases représentant les processus,** les capacités et les outils nécessaires au développement (sur la partie gauche de la boucle) et aux opérations (sur la partie droite de la boucle). Tout au long de chaque phase, les équipes collaborent et communiquent pour maintenir l'alignement, la vélocité et la qualité.

## QUELQUES NOTIONS DU DEVOPS

### LE DEPLOIEMENT CONTINU

C'est une approche qui considère qu'une application doit être construite de telle manière à ce qu'elle puisse être envoyée en production à tout moment c'est-à-dire qu'à partir du moment où les applications réussissent les tests automatisés elles sont déployées automatiquement en production. L'idée c'est d'avoir un rythme en production plus régulier, rapide et léger ce qui est moins risqué qu'une mise à jour importante en mode « big bang ». L'objectif du déploiement continu est une accélération du Time To Market. Avec un tel type de cycle de livraison il est plus facile pour l'entreprise en exploitant les données d'avoir des retours des clients sur le produit livré et d'aboutir par itérations à celui qui serait le plus proche du besoin du marché. Ainsi à chaque nouvelle livraison le produit répond mieux aux attentes des clients permettant ainsi d'obtenir des produits de qualité correspondant aux besoins des clients

### CI INTÉGRATION CONTINUE / CD DÉPLOIEMENT CONTINU

L'intégration continue désigne la pratique qui consiste à automatiser l'intégration des changements de code dans un projet de développement. Elle permet aux développeurs de logiciels de merger fréquemment des changements de code dans un dépôt central où les builds et les tests s'exécutent. Cela permet aux équipes DevOps de corriger les bugs plus rapidement, d'améliorer la qualité des logiciels, et de réduire le temps nécessaire à la validation et à la livraison des nouvelles mises à jour logicielles. La livraison continue s'étend à l'intégration continue en déployant automatiquement les changements de code dans un environnement de test/production. Elle suit un pipeline de livraison continue, où les builds, les tests et les déploiements automatisés sont orchestrés en un workflow de livraison unique.

### INFRASTRUCTURE AS CODE

Que votre organisation dispose d'un data center sur site ou qu'elle soit intégralement dans le cloud, la capacité de provisionner, de configurer et de gérer rapidement et systématiquement l'infrastructure est la clé d'une adoption fructueuse de DevOps. L'IaC va au-delà de la simple utilisation de scripts pour configurer une infrastructure : il traite votre infrastructure comme du code réel en ayant recours au contrôle de version, aux revues de code, aux tests, et bien plus encore.

### DEVSCOPS

L'approche DevSecOps (Development - Security - Operations) est une approche qui permet d'intégrer la sécurité des données dès le début d'un projet. La sécurité de celles-ci est considérée comme une condition préalable avant de commencer. L'objectif est de pouvoir l'intégrer dans l'ensemble du cycle de vie du projet, du développement à la mise en oeuvre, en utilisant des méthodes flexibles et l'approche DevOps. DevSecOps vous permet de configurer l'automatisation des tests de sécurité à chaque fois que votre application, logiciel, site internet est testé et déployé. Cela permet de détecter et de corriger rapidement les anomalies.

## EXEMPLES D'OUTILS

### LA PHASE DE DEVELOPPEMENT CI/CD

Lors de la phase de test de CI «Intégration Continue / CD « Déploiement Continu», qui challenge le code, celui-ci n'est pas livré par les développeurs aux opérationnels en un seul bloc et à un seul moment, mais est testé tout au long de son écriture ou de ses modifications. Ce qui permet de réduire le temps nécessaire habituel pour créer de nouvelles fonctions. Le code, fonctionnel puisque vérifié, servira de base aux opérationnels. Comme outils de CI et de CD on peut trouver :

**GitHub** : En enregistrant toutes les modifications du code source, GitHub permet de suivre l'avancement des travaux et d'avoir un aperçu sur l'historique de travail. Très avantageux pour une équipe, GitHub fusionne les codes sources.

**SourceForge** : l'adversaire de GitHub permet de créer des projets en open source grâce à une palette d'outils très complète : wikis de projets, accès à une BDD MySQL, url de sous-domaines uniques, métriques et analyses.

Les outils CI/CD alternatifs : si l'on souhaite éviter les deux premières sources, on peut se tourner vers **GitLab** et son accès facilité à **Azure DevOps**; **Bitbucket** et son environnement complet intégrant un puissant outil de suivi des problèmes ou **Beanstalk** qui permet de coder, tester et déployer directement depuis un navigateur.



### GESTION DES CONFIGURATIONS

La gestion des configurations c'est l'ensemble des processus qui valide la conformité d'un produit face aux diverses exigences durant tout son cycle de vie. Les outils de la gestion des configurations sont par exemple :

**Ansible** : il automatise des tâches courantes telles que le provisioning dans le cloud, le déploiement d'application et l'orchestration IT. Il facilite l'installation, la configuration et la modification d'un ou plusieurs serveurs de manière prévisible et déclarative.

**Jenkins** : il permet aux équipes de surveiller la réalisation des tâches répétées. L'outil s'appuie sur ses nombreux plugins pour délivrer une solution simple d'utilisation pratique dans les configurations et les tests de régression d'automatisation.





## EXEMPLES D'OUTILS

### INFRASTRUCTURE AS CODE

Il s'agit de chiffrer l'ensemble des processus afin d'automatiser les tâches reproductibles telles que le provisioning, la configuration ou les déploiements. Les plateformes répondent, pour la plupart d'entre elles, à la philosophie UNIX c'est-à-dire qu'elles ne font qu'une chose, mais qu'elles le font bien. L'outil **Terraform** est un environnement logiciel qui se concentre principalement sur le provisioning d'infrastructure. Il automatise la construction des ressources et facilite la définition des typologies cloud pour les principaux fournisseurs d'infrastructure.



### LA PRODUCTION

Grâce aux conteneurs aujourd'hui, les applications sont empaquetées avec leur contexte d'exécution. Elles sont livrées avec tous leurs logiciels, bibliothèques et documents nécessaires. Ces paquets immuables traversent les équipes du développement à la production.

**Docker** : c'est un logiciel libre en open source se concentre sur la normalisation et la spécification des containers afin de construire un écosystème sain. Son objectif? Ne faire tourner qu'une seule application. Son système repose sur une utilisation minimaliste des fichiers, une gestion simple du démarrage des services et des contenaires immuables.

**Kubernetes** : cet outil se présente comme une plateforme d'orchestration des containers, elle abrite une quantité importante de paquets. Au-delà de son service de plateforme, Kubernetes automatise le processus de gestion d'une large palette de ces containers.



## PODCAST

WEE TALK TECH



Avec Didier et Sebastien, Consultants chez WEENEO

### Sources :

Atlassian.com, Bluesoft-group.com, Cookie connecté, Freepik, Netapp.com, Oracle.com, Syloe.com, Qrpinternational.fr, Wikipedia