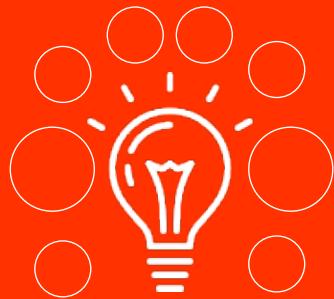


PeopleCert DevOps « Les fondamentaux »



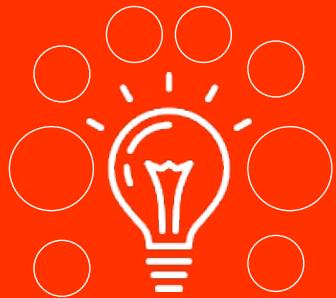
## Module 2 : **Principes et concepts clés de DevOps**

---

### **Sujets :**

---

- C.A.L.M.S. et les trois principes
- Livraison continue à travers le pipeline de déploiement
- La portée du Full Stack de DevOps



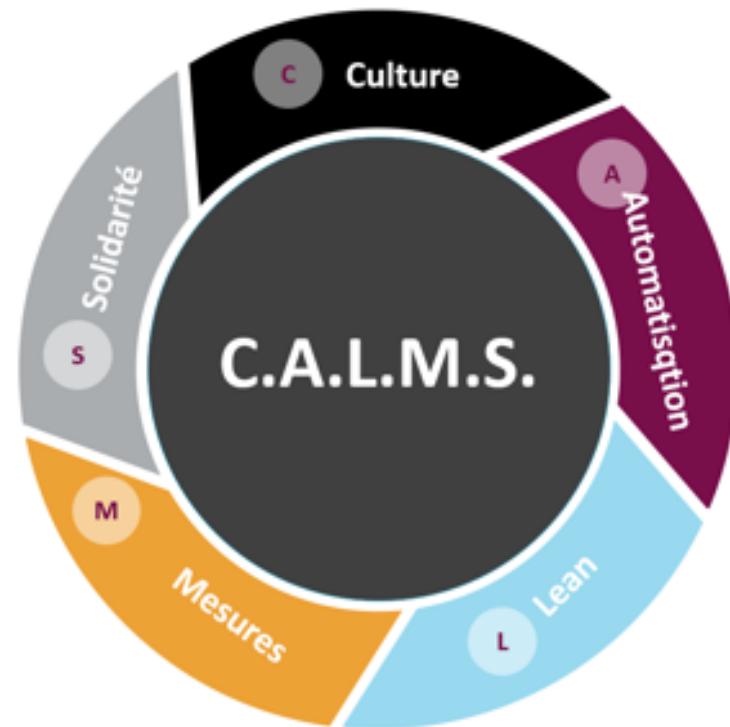
## Module 2 : Principes et concepts clés de DevOps

C.A.L.M.S et les trois principes



## C.A.L.M.S.

- CALMS est un acronyme développé à l'origine par John Willis et Damon Edwards en 2010, puis perfectionné par Jez Humble, afin de comprendre les piliers et les concepts clés qui sous-tendent DevOps.
- **Culture** : il est important de comprendre que les valeurs, les croyances et les attitudes qui imprègnent une organisation et son informatique doivent évoluer.
- **Automatisation** : la conviction que tout peut être automatisé et que les processus manuels doivent l'être dans la mesure du possible.
- **Lean** : accent mis sur les idées du système de valeurs et de la méthodologie afin de réduire la complexité des processus et rationaliser le flux de travail.
- **Mesure** : croire en l'importance de tout mesurer et de mettre en place des processus visibles et transparents.
- **Solidarité** : accent mis sur la collaboration et la communication entre le développement et les opérations, afin de leur permettre d'avoir un but commun.



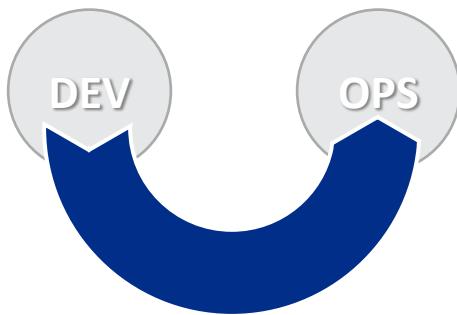


## Les trois principes clés (Flux – 1er principe)



Pour plus d'informations, vous pouvez regarder cette vidéo :

*The Three Ways Of The Phoenix Project:* <https://www.youtube.com/watch?v=nUOXDEvplRc&x=mobref>



**Le premier principe** est axé sur la compréhension du flux de travail - de sa manière la plus élémentaire - comment le travail passe-t-il de Dev à Ops, puis des zones fonctionnelles métier aux clients, de gauche à droite, d'une équipe à l'autre.

Il s'agit d'identifier comment les éléments fonctionnent les uns avec les autres (comprendre ce qui passe d'une équipe à l'autre, pourquoi cela arrive et les conséquences que cela entraîne) .

Le flux consiste à identifier les contraintes évoquées précédemment (à savoir où se trouvent les goulots d'étranglement et où la productivité est la plus lente) afin de pouvoir ensuite établir une priorisation d'actions pour obtenir l'impact le plus grand impact.

Ce n'est que s'il y a compréhension de comment le travail se déroule, qu'il y aura identification des obstacles contribuant au « mur de la confusion » et ensuite élimination.

Source : <https://itrevolution.com/the-three-ways-principles-underpinning-devops>



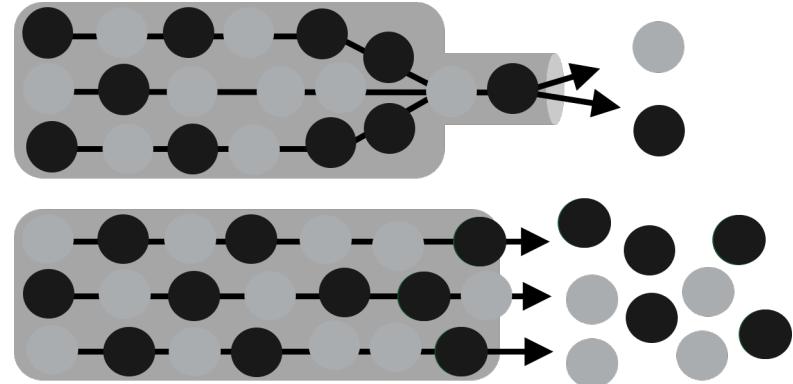
## Les trois principes clés (la théorie des contraintes)

La théorie des contraintes du Dr. Eliyahu Goldratt stipule qu'aucun système ou processus complexe ne peut être plus efficace ou plus puissant que son goulot d'étranglement ou sa contrainte la plus limitante.

Augmenter le flux signifie identifier les contraintes existantes. Cela signifie également avoir une compréhension claire de :

- ce qui se passe tout au long de la chaîne de valeur pour chaque processus,
- l'ensemble des activités qui sont mises en œuvre et des personnes impliquées dans un processus.

Pas de manière vague, mais selon les hypothèses les plus probables.



Source : *The Goal*, Dr. Eliyahu Goldratt

### Comment augmenter le flux ?

Augmenter la transparence et la visibilité du travail

Mettre des limites sur le travail en cours (Work In Progress WIP)

Travailler en petits lots

Réduire et limiter le nombre de transferts

Identifier et prioriser les contraintes et les goulets d'étranglement

Améliorer constamment pour éliminer le gaspillage là où c'est possible

Source : <https://itrevolution.com/the-three-ways-principles-underpinning-devops>

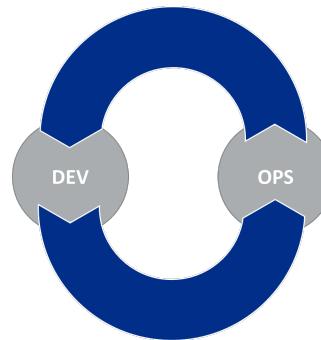
## Les trois principes clés (feedback ou retour d'informations – 2ème principe)

Le deuxième principe se concentre sur la collecte et le traitement de feedbacks tout au long du flux de travail. Pour cela, il est nécessaire de collecter activement des feedbacks à chaque étape de la chaîne de valeur.

En informatique, le travail s'effectue au sein de systèmes complexes. Le risque de défaillances ou de conséquences catastrophiques est très élevé. Il existe généralement de fortes interconnexions entre des composants étroitement couplés, et cette complexité signifient que faire la même chose deux fois ne produira pas nécessairement les mêmes résultats.

Aussi, la sécurité, la fiabilité et la stabilité deviennent très importantes. La réflexion sur les systèmes d'information devient cruciale.

Les boucles de feedback permettent de créer des systèmes plus sûrs et plus résilients, car elles offrent davantage de possibilités de détecter et de corriger les erreurs.



### Comment augmenter la qualité et la résilience ?

Déetecter les problèmes plus tôt, dès qu'ils surviennent

Se mobiliser collectivement pour résoudre les problèmes et acquérir de nouvelles connaissances

Pousser la qualité au plus près de la source

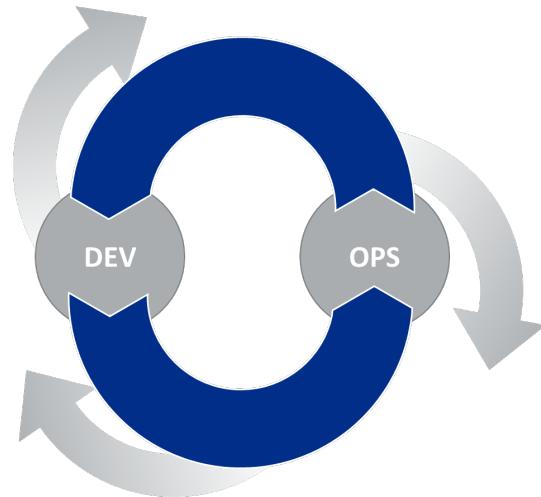
Optimiser globalement plutôt que localement

Source : <https://itrevolution.com/the-three-ways-principles-underpinning-devops>

## Les trois principes clés (expérimentation et apprentissage en continu – 3ème principe)

Le troisième principe consiste à diffuser au sein de l'organisation, une culture basée sur l'expérimentation et l'apprentissage en continu.

Cela implique de briser la culture de silo, qui est souvent une culture de peur et de faible confiance, d'aider l'informatique à considérer l'erreur comme une expérience d'apprentissage et à comprendre l'intérêt d'innover.



### Comment encourager l'expérimentation et l'apprentissage ?

Favoriser une culture dans laquelle il est possible d'apprendre en toute sécurité

Institutionnaliser l'amélioration continue au quotidien

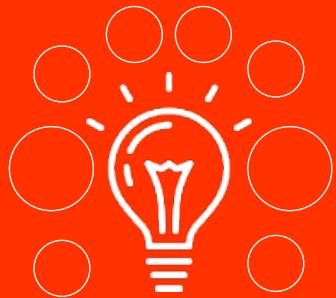
Transformer les connaissances locales en améliorations globales

Implémenter des modèles de résilience dans le travail au quotidien

Transformer le leadership pour encourager l'apprentissage

Considérer les erreurs comme une occasion d'apprentissage

Source : <https://itrevolution.com/the-three-ways-principles-underpinning-devops>



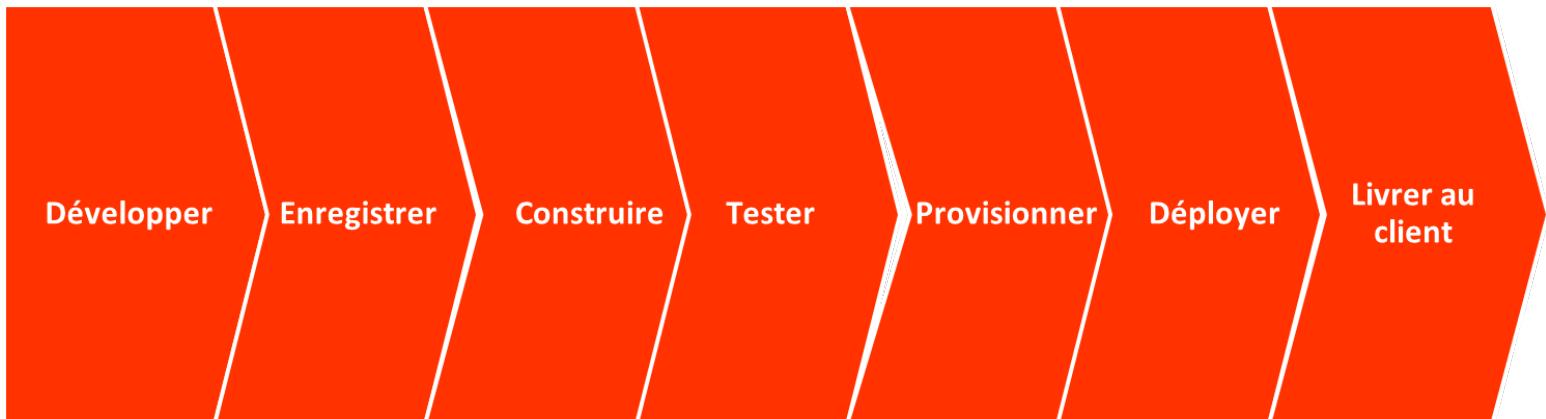
## Module 2 : Principes et concepts clés de DevOps

Livraison continue à travers le pipeline de déploiement



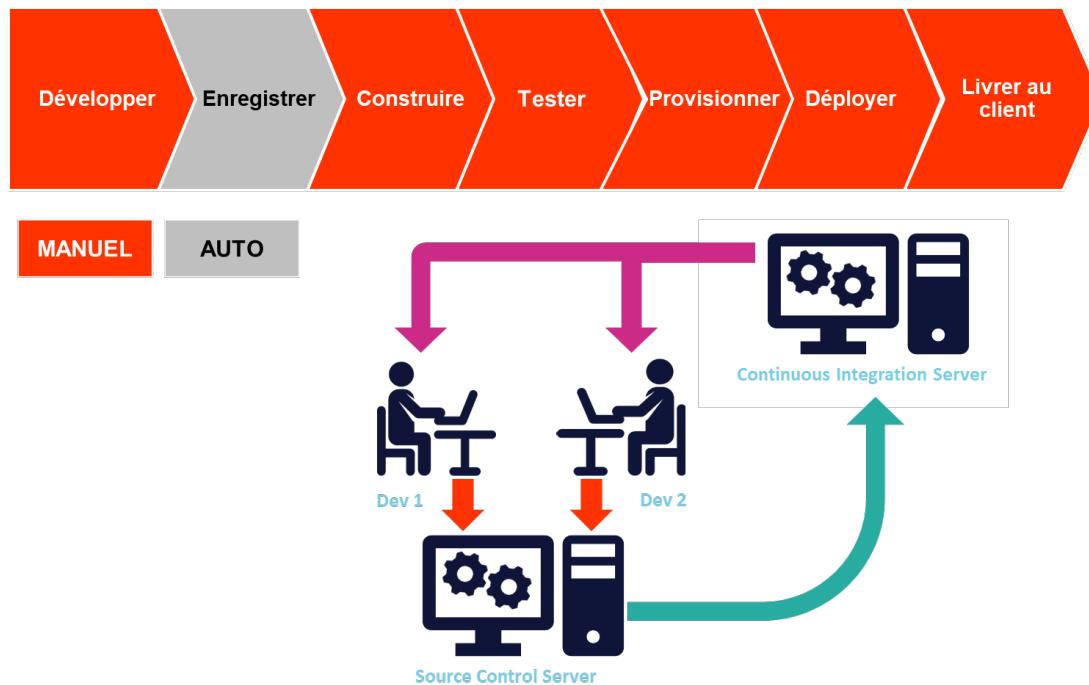
## Le pipeline de déploiement

**Le pipeline de déploiement** est un modèle de processus sous forme d'étapes qui fournit un point de départ pour assurer la transparence et les contrôles au fur et à mesure que le travail progresse dans les différents ensembles de tests et de déploiements en vue de la mise en production.



## Le pipeline de déploiement (intégration continue)

L'intégration continue consiste à fusionner tous les développements en cours dans une branche principale partagée (un référentiel de code ou un tronc commun) tout au long de la journée.

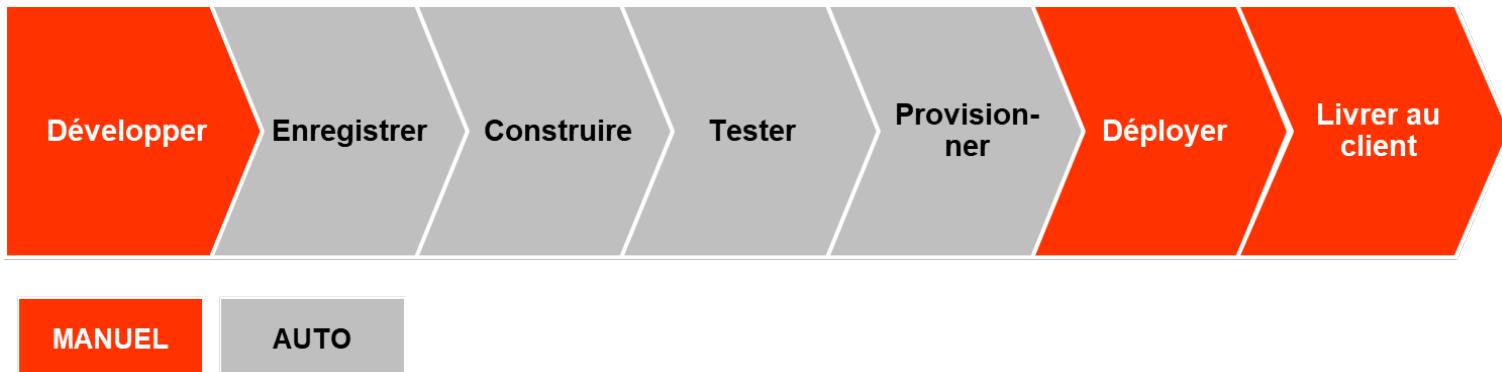


Dans un processus automatisé de livraison continue, l'intégration continue couvre principalement la phase de construction. Généralement, l'intégration continue couvre l'intégration, la construction et les tests de code dans l'environnement de développement.



## Le pipeline de déploiement (livraison continue)

La **livraison continue** est un ensemble de pratiques conçues pour garantir que le code peut toujours être déployé en production rapidement et en toute sécurité tout au long de son cycle de vie. Cela est réalisé en déployant les fichiers exécutables dans un environnement similaire à la production et en effectuant des tests automatisés pour détecter les problèmes.

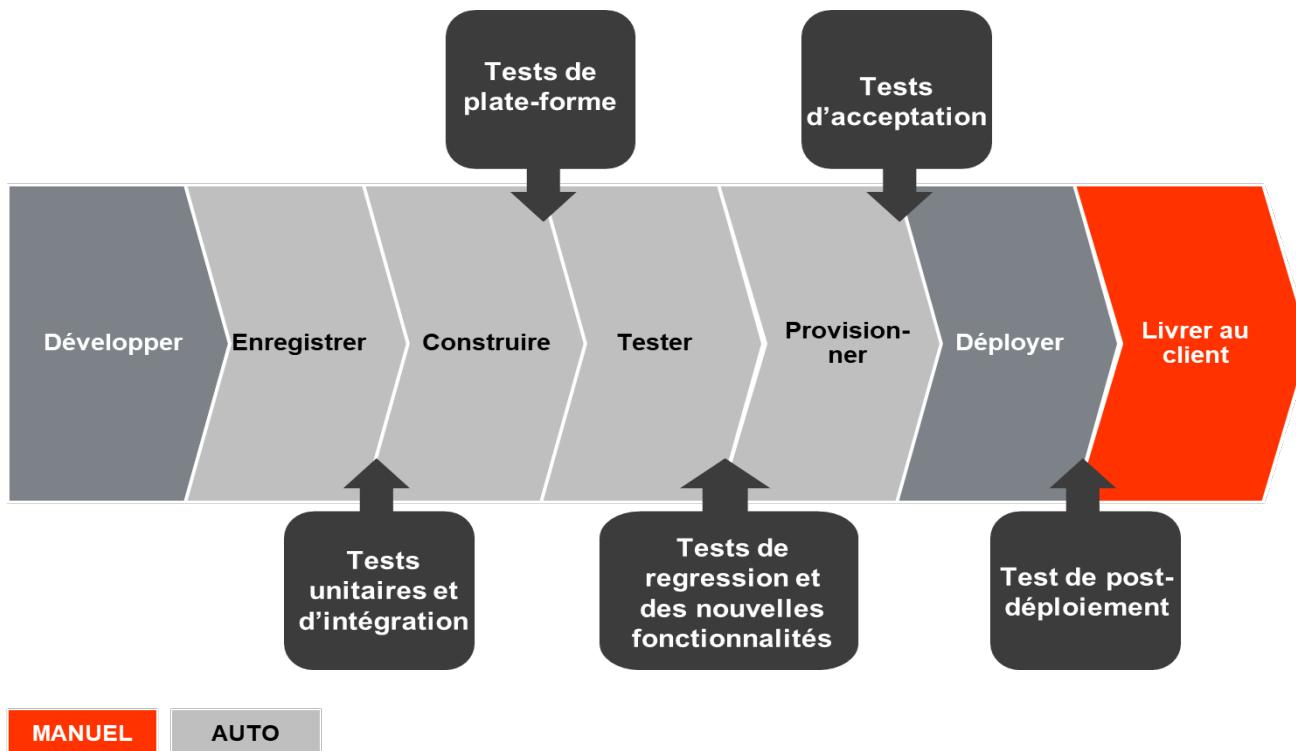


La livraison continue ne signifie pas que chaque modification est déployée en production. Cela signifie simplement que chaque changement peut être déployé en production à tout moment. **Ce n'est pas la même chose que le déploiement continu.** Dans la livraison continue, **la mise en production est une décision manuelle** et humaine. La livraison continue nécessite que, si une modification entraîne une erreur d'un test automatisé, les développeurs arrêtent le pipeline de déploiement et remettent le système dans un état déployable. La livraison continue est une pratique essentielle de DevOps. Ce n'est que par la livraison continue qu'il est possible de s'assurer que les modifications génèrent de la valeur métier.



## Le pipeline de déploiement (tests en continu)

**Les tests en continu** impliquent l'exécution de tests automatisés à chaque étape du pipeline de déploiement. Fournit un feedback immédiat à chaque étape pour limiter les risques. Les tests en continu automatisés sont un élément clé de l'intégration et de la livraison continues. Ils garantissent que le code et l'environnement fonctionnent correctement et restent dans un état déployable.

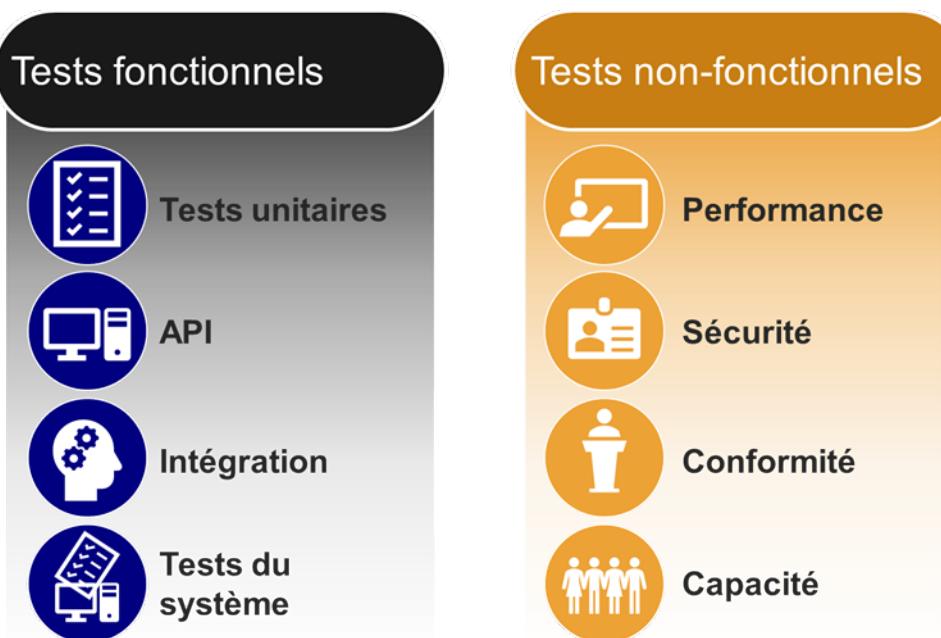


## Le pipeline de déploiement (tests fonctionnels et non-fonctionnels)

Il existe deux types de tests clés qu'il est possible de distinguer dans les tests en continu : les tests fonctionnels et les tests non fonctionnels.

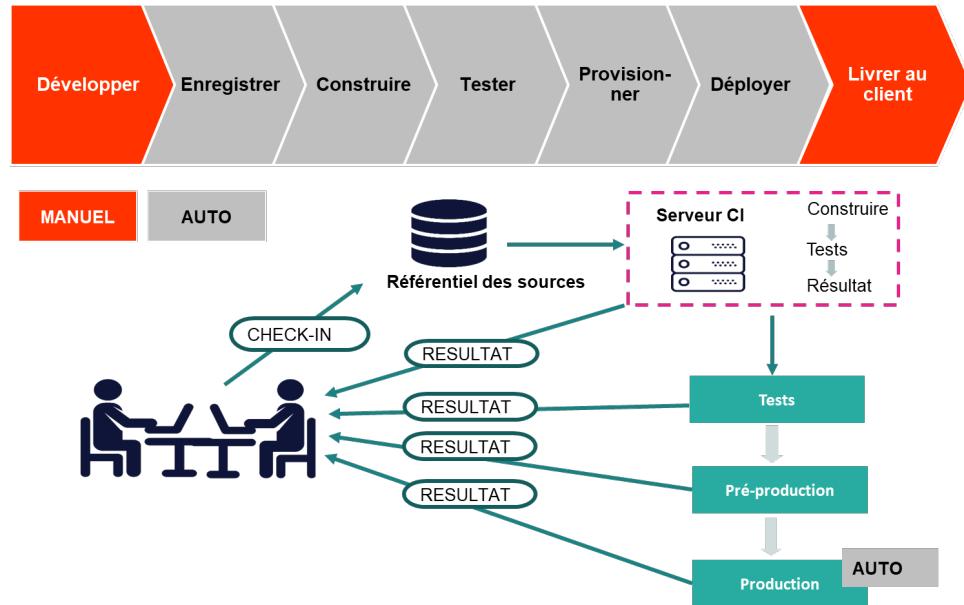
Les tests fonctionnels correspondent aux tests des fonctionnalités nécessaires au fonctionnement du produit. Ils testent les composants critiques pour le client et que ce dernier rejeterait en cas d'erreur.

Les tests non fonctionnels sont les tests du fonctionnement du système. Il s'agit d'effectuer des tests par rapport à des éléments dont l'utilisateur final (ou le client) n'est pas forcément conscient et qui sont à la base des performances fonctionnelles qu'il constate.



## Le pipeline de déploiement (déploiement continu)

Le déploiement continu est une extension du concept de livraison continue, dans lequel tous les changements qui passent avec succès les tests automatisés sont automatiquement mis en production. Cela permet d'automatiser les étapes manuelles de la livraison continue et permet plusieurs déploiements par jour.



- Le déploiement continu devrait être l'objectif de DevOps. Il faut toutefois savoir qu'il peut y avoir des cas pour lesquels il n'est pas atteignable. De nombreuses entreprises sont soumises à des exigences réglementaires ou autres rendant le déploiement continu non-viable.
- **Le déploiement n'est pas nécessairement la même chose que la mise en production. La mise en production est toujours une décision manuelle.**



# Les avantages de la livraison continue

**La livraison continue à travers le pipeline signifie...**

Gestion efficace des données de test	Automatisation complète, rapide et fiable des tests et du déploiement	Développement et intégration continue basée sur un tronc (Trunk Based Development)	Code source, application et configuration du système gérés en versioning	Intégration de la sécurité dans le processus de livraison
--------------------------------------	---	--	--	---

**Ce qui amène à...**

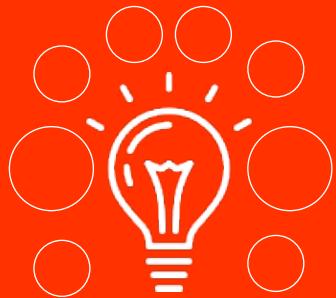
Moins de reprise	Déploiement plus facile	Des niveaux de performance informatique plus élevés (débit et stabilité augmentés)	Taux d'échec des changements plus faible	S'identifier clairement à l'organisation
------------------	-------------------------	--	--	--

**Des niveaux de performance organisationnelle plus élevés.**

**Pour plus d'informations, vous pouvez regarder cette vidéo :**

*Continuous Delivery Is A Team Sport:* <https://www.youtube.com/watch?v=9XmvFvdTObY>

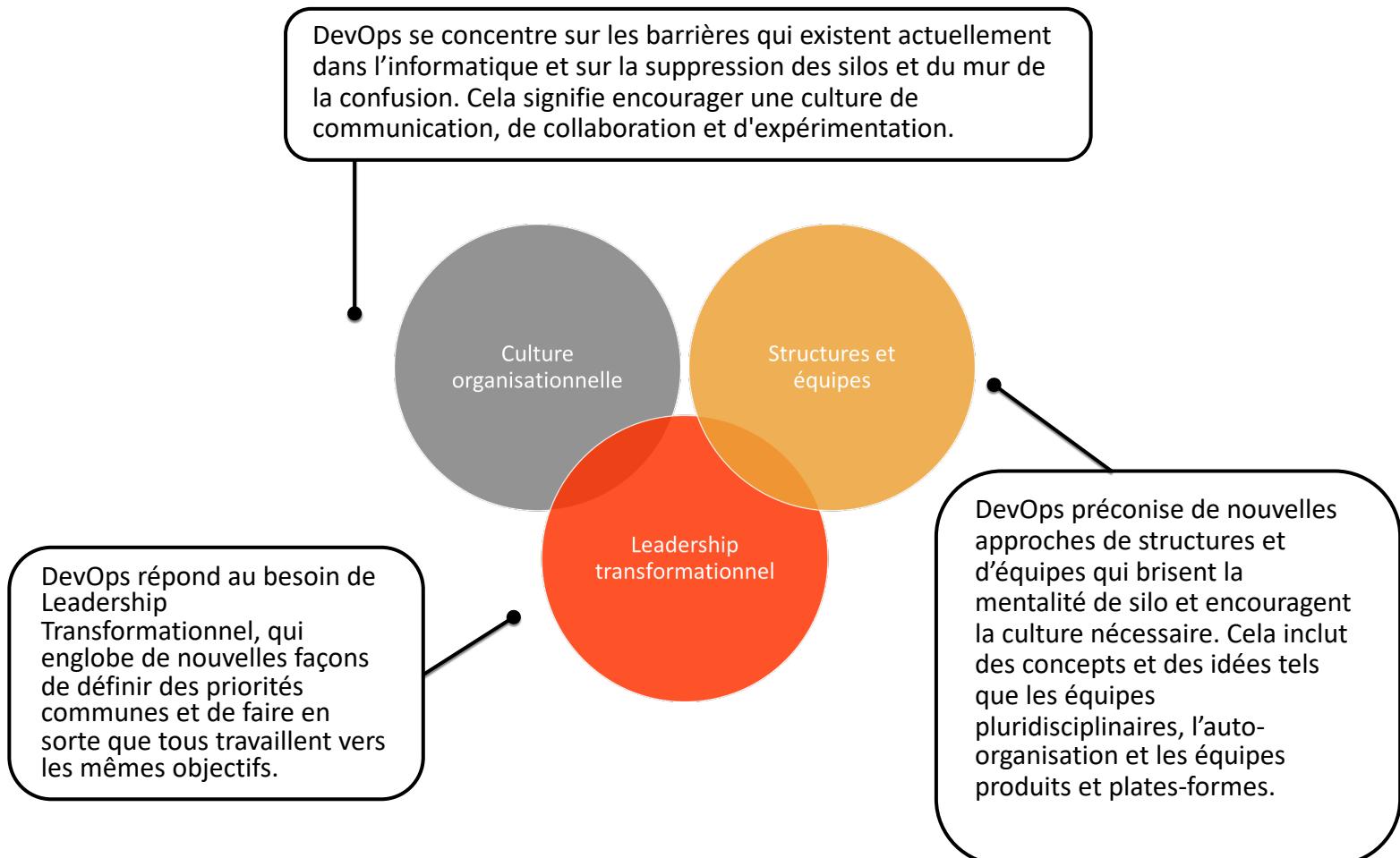
Source : 2016 State Of DevOps Report, Puppet Labs



## Module 2 : Principes et concepts clés de DevOps

La portée du Full Stack de DevOps

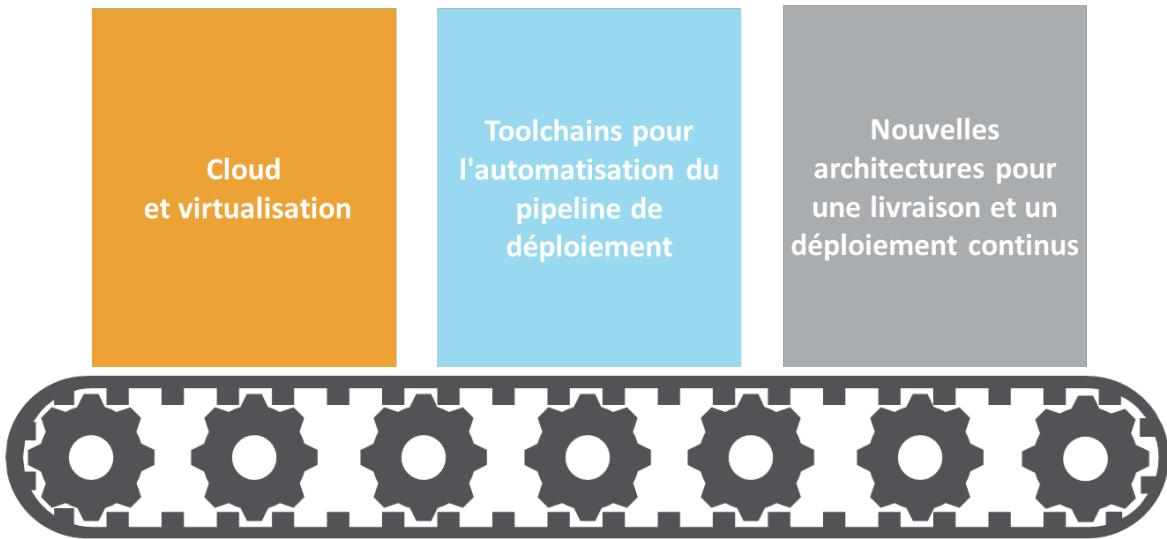
## DevOps et l'impact sur les personnes et la culture



## DevOps et l'impact sur les processus et les pratiques



## DevOps et l'impact sur la technologie et l'automatisation



DevOps comprend une couche qui aborde les défis liés à la technologie : comment utiliser les processus et les pratiques appropriés et les améliorer davantage grâce à l'intégration et à l'automatisation. Cela implique des chaînes d'outils (toolchains) intégrés (comprendre quels outils utiliser, comment les intégrer au sein d'une équipe et permettre l'automatisation d'une grande variété de processus).

Cela implique un niveau élevé de sophistication et de maturité de l'automatisation du pipeline de déploiement et de comprendre comment tirer parti du Cloud et de la virtualisation pour accélérer les choses et augmenter la collaboration et l'efficacité.