



DEVOPS

L'Introduction

Culture, Collaboration et Performance dans l'IT moderne



Formateur : Ovyn Flavian / BStorm

Objectif de la Journée : Découvrir pourquoi le DevOps est indispensable à la transformation digitale et identifier les piliers de cette approche

Pourquoi Sommes-nous Là ?

Le Contexte

Le marché digital exige plus de vitesse et de qualité que jamais.



L'accélération numérique

Objectifs de la Formation



Comprendre le DevOps

Concept et histoire



Métriques DORA

Mesurer la performance



Maîtriser CALMS

Les 5 principes culturels



Anti-patterns

Identifier les pièges à éviter



Prérequis : Aucun – Seule votre curiosité compte !

La Définition Simple du DevOps



DevOps = Development + Operations

Une méthodologie visant à unifier et automatiser les processus entre les équipes de développement logiciel (Dev) et les équipes d'exploitation informatique (Ops).



Le But

Raccourcir le cycle de vie du développement, améliorer la fréquence des livraisons et garantir la stabilité des systèmes.

Contexte Historique : Du Chaos à la Vitesse



“

Évolution vers une approche qui unit les équipes de développement et d'exploitation

”

Approches Traditionnelles vs. DevOps

Le mur entre les équipes de développement et d'exploitation : une confrontation de philosophies

Caractéristique	Traditionnel (Silos)	Approche DevOps
 Relation Dev/Ops	 "Lancer le code par-dessus le mur"	 Collaboration et Confiance totale
 Fréquence de Déploiement	 Rare (Mois/Trimestre)	 Fréquente (Jours/Heures)
 Gestion des Erreurs	 Recherche de faute (Blame)	 Apprentissage (Blameless)
 Infrastructure	 Manuelle, unique (Mise à jour d'un serveur)	 Automatisée, réutilisable (IaC)

Le passage d'une approche traditionnelle à DevOps représente une transformation culturelle et technique

CALMS – Les Cinq Piliers Culturels



Le DevOps est **80% Culture et 20% Outils**



Culture

Casser les silos et favoriser l'empathie entre Dev et Ops



Automation

Automatiser les tâches répétitives pour améliorer la productivité



Lean

Optimiser le flux de valeur et éliminer le gaspillage



Measurement

Mesurer la performance technique et l'impact business



Sharing

Partager les connaissances, les outils, les succès et les échecs



Les cinq piliers forment une approche holistique du DevOps

C – Culture (Le plus important)



Définition

Casser les silos. Favoriser l'empathie et les objectifs communs entre Dev et Ops.



Le DevOps est 80% Culture et 20% Outils

En Pratique



Équipes Multifonctionnelles

Combinaison d'experts techniques et métiers pour livrer des solutions complètes



Blameless Post-Mortem

Analyser les échecs sans chercher de coupable, pour apprendre collectivement



Production est la Responsabilité de Tous

Chacun est responsable de la qualité du service livré en production

“ La culture est le fondement de toute bonne pratique DevOps

A – Automation (La Productivité)



Définition

Automatiser les tâches répétitives, sources d'erreurs et de lenteur.



CI/CD

Intégration et Déploiement Continus

- ✓ Intégration automatique du code
- ✓ Tests automatiques
- ✓ Déploiement automatique



Tests Automatisés

Qualité du code et couverture des tests

- ✓ Tests unitaires
- ✓ Tests d'intégration
- ✓ Tests d'acceptation



Infrastructure as Code

Gérer l'infrastructure avec du code

- ✓ Terraform
- ✓ Ansible
- ✓ Docker



Automatiser = Productivité + Qualité + Vitesse

L – Lean (L'Efficacité)

“ Définition

Optimiser le flux de valeur et éliminer le gaspillage :

- ⌚ Temps d'attente inutile
- ☰ Tâches inutiles
- 🚧 Travail en cours excessif

⚙️ En Pratique



Limiter le WIP

Restreindre le nombre de tâches en cours pour améliorer la concentration et réduire les transitions



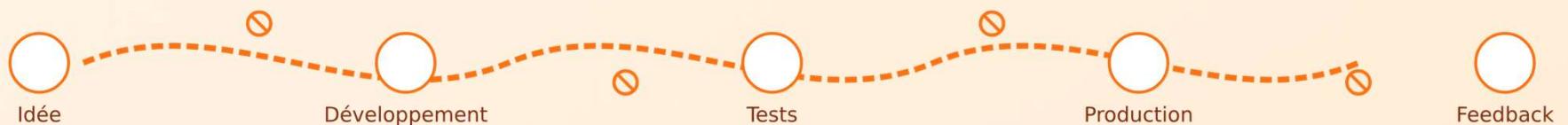
Améliorer le Flow

Optimiser le flux de travail pour réduire les goulets d'étranglement et les interruptions



Livrer par petits lots

Moins risqué, feedback plus rapide, et plus facile à adapter aux changements



💡 Lean = Éliminer le gaspillage, optimiser le flux

M

Measurement

L'Amélioration

E Définition

Mesurer la performance technique et l'impact business pour prendre des décisions basées sur des faits.



La mesure est le fondement de l'amélioration continue

En Pratique



Monitoring des Systèmes

Suivi de la disponibilité, de la latence et de la performance des systèmes en temps réel.



Métriques DORA

Les métriques standard de l'industrie pour mesurer la performance de livraison logicielle.



Lead Time



Déploiement



MTTR



Taux d'échec



La mesure permet d'optimiser les processus DevOps

S – Sharing (L'Apprentissage)

La Définition

Partager les connaissances, les outils, les succès et les échecs.



"La connaissance collective est plus puissante que l'individuelle"

En Pratique



Documentation

Wikis centralisés pour les connaissances



Standards

Outils et standards communs



Code Review

Revue de code et partage d'expertise



Retours d'Expérience

Partager succès et échecs



Partagez pour apprendre, apprenez pour partager

Les Bénéfices Business du DevOps



ROI Accéléré

Mise sur le marché plus rapide

- ⌚ Temps réduit pour livrer les fonctionnalités

 Retour sur investissement accéléré



Qualité

Moins de bugs en production

-  Tests automatisés et déploiements fréquents

 Amélioration de la stabilité des systèmes



Compétitivité

Réactivité face aux besoins

-  Adaptation rapide aux exigences clients

 Avantage concurrentiel sur le marché



Le DevOps transforme les équipes IT en moteurs de valeur pour l'entreprise

Les Métriques DORA

Les métriques DORA mesurent la performance de la livraison de logiciels. Elles sont le **standard de l'industrie**.

Elles mesurent deux aspects : **La Vitesse** et **La Stabilité**

La Vitesse

Lead Time for Changes

Temps écoulé entre le début du travail et sa livraison en production

Deployment Frequency

Nombre de déploiements réussis en production sur une période donnée

La Stabilité

Mean Time to Recovery (MTTR)

Temps nécessaire pour restaurer le service après un incident

Change Failure Rate

Pourcentage de déploiements qui entraînent un échec, nécessitant un correctif

Les métriques DORA sont le standard de l'industrie pour évaluer la performance DevOps

Vitesse : Livrer Vite de la Valeur



Lead Time for Changes

Qu'est-ce que c'est ?

Temps écoulé entre le début du travail (idée ou code écrit) et sa livraison en production.

Objectif

Être le plus court possible (Idéalement quelques heures).



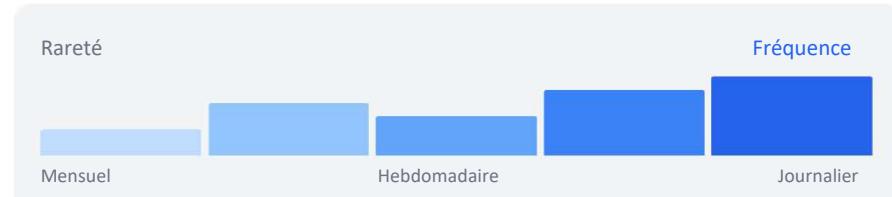
Deployment Frequency

Qu'est-ce que c'est ?

Nombre de déploiements réussis en production sur une période donnée.

Objectif

Déployer souvent (Plus c'est souvent, moins chaque déploiement est risqué).



Stabilité : Maintenir la Qualité

Deux métriques clés pour mesurer la **résilience** et la **qualité** de vos déploiements



Mean Time to Recovery (MTTR)

Temps nécessaire pour restaurer le service après un incident

❶ Définition

Temps écoulé entre l'incident et la restauration complète du service

❷ Objectif

Être très court → Mesure la résilience de l'Ops



Change Failure Rate

Pourcentage de déploiements qui entraînent un échec

❶ Définition

Pourcentage de déploiements nécessitant un correctif ou un rollback

❷ Objectif

Être faible → Mesure la qualité du processus et des tests



Ces métriques sont le reflet de la qualité technique et de la maturité culturelle de votre équipe

Les Anti-Patterns à Éviter



Voici quatre pièges majeurs à éviter lors de votre transformation DevOps



L'Équipe DevOps Isolée

Créer un nouveau silo appelé "DevOps Team" qui fait le travail des autres.

DevOps est la responsabilité de tous !



L'Outil-Centrisme

Acheter des outils sans changer la culture de travail.

L'outil est un moyen, pas un but.



Blâmer l'Humain

Sanctionner après un incident.

La peur tue l'apprentissage (Culture Blameless).



Automatiser le Chaos

Automatiser des processus qui sont déjà dysfonctionnels.

Penser Lean d'abord, automatiser ensuite.



Évitez ces anti-patterns pour une transformation DevOps réussie

Synthèse et Mots-Clés

Le DevOps n'est pas un outil, mais une philosophie de travail pour l'ère numérique.



💡 DevOps = Culture + Flux + Mesure

Prochaines Étapes et Ressources

Lecture Recommandée

The Phoenix Project

Gene Kim

Roman de référence sur le DevOps qui explique les principes through the story of a IT-driven business transformation.



D'autres livres seront mentionnés pendant la formation

Outils Techniques à Explorer



Git

Gestion de versions



Docker & Kubernetes

Conteneurisation



Jenkins / GitLab CI

CI/CD



Terraform

Infrastructure as Code

Ces outils sont essentiels pour l'employabilité dans le domaine DevOps

Pour continuer votre journey DevOps après cette formation

Cas Pratique : Transformation DevOps



Avant la Transformation

- Modèle en cascade avec phases séquentielles
- Déploiements rares (mois/trimestre)
- Gestion des erreurs par recherche de coupable
- Infrastructure manuelle et unique



Après la Transformation

- + Approche DevOps avec collaboration continue
- + Déploiements fréquents (jours/heures)
- + Culture d'apprentissage blameless
- + Infrastructure automatisée et réutilisable



Facteurs Clés du Succès



Culture

Cassage des silos



Automatisation

CI/CD et tests



Mesure

Métriques DORA



Partage

Connaissances



Roadmap de Mise en Œuvre

Étapes recommandées pour débuter une transformation DevOps dans votre organisation

1. Évaluation

- Diagnostic de l'état actuel
- Identification des défis

2. Business Case

- Démontrer la valeur business
- Mettre en avant les bénéfices

3. Culture

- Casser les silos
- Blameless post-mortems

4. Automation

- CI/CD
- Tests automatisés

5. Lean

- Limiter le WIP
- Améliorer le Flow

7. Measurement

- Métriques DORA
- Monitoring des systèmes

8. Sharing

- Documentation centralisée
- Code Review

6. Amélioration

- Itérations continues
- Feedback loops



Conseil : Commencez par les aspects culturels avant les outils ! Le DevOps est 80% Culture et 20% Outils.

Outils DevOps Essentiels

CI/CD



Jenkins

Solution open-source de CI/CD et d'intégration continue. Automatise les étapes de build, test et déploiement.



GitLab CI

CI/CD intégré à GitLab. Permet de définir des pipelines dans un fichier `.gitlab-ci.yml`.



GitHub Actions

Workflows d'CI/CD directement dans GitHub. Très intégré avec les dépôts et les issues.

Conteneurisation



Docker

Plateforme de conteneurisation. Crée, déploie et exécute des applications dans des conteneurs isolés.



Kubernetes

Orcheteur de conteneurs. Gère le déploiement, la mise à l'échelle et la gestion des applications conteneurisées.



Docker Compose

Outil pour définir et exécuter des applications multi-conteneurs à l'aide d'un fichier YAML.

Infrastructure



Terraform

Outil d'Infrastructure as Code. Permet de définir et de provisionner l'infrastructure de manière déclarative.



Ansible

Outil d'automatisation et de configuration. Simplifie la gestion de la configuration et le déploiement d'applications.



Packer

Outil pour créer des images d'infrastructure uniformes pour plusieurs plates-formes. Idéal pour les conteneurs.

Ces outils sont essentiels pour une implementation DevOps réussie