

DevOps - Rappels



Objectifs

- Rappels sur le DevOps
- Intégration continue
- Déploiement continue

Historique

- Premières applications :
 - Taille limitée
 - Autonomes : n'avaient que peu de besoins d'interagir avec d'autres applications
 - L'équipe qui développait l'application devait aussi la maintenir

Historique

- Des systèmes de plus en plus gros :
 - Apparition des progiciels de gestion intégré (PGI) ou Enterprise Resources Planning (ERP)
 - Applications ont besoins de communiquer avec d'autres applications / systèmes
- ⇒ Besoin d'être efficace et séparation des tâches plus développement et plus infrastructure

Historique

- Trois enjeux en gestion de projets :
 - Coût
 - Objectifs
 - Délais
- Opérateurs : Stabilité
 - Augmenter la qualité
 - Au détriment du coût et du temps
- Développeurs : Livrer
 - Ajouter / modifier des fonctionnalités vite et à faible coût
 - Au détriment de la qualité



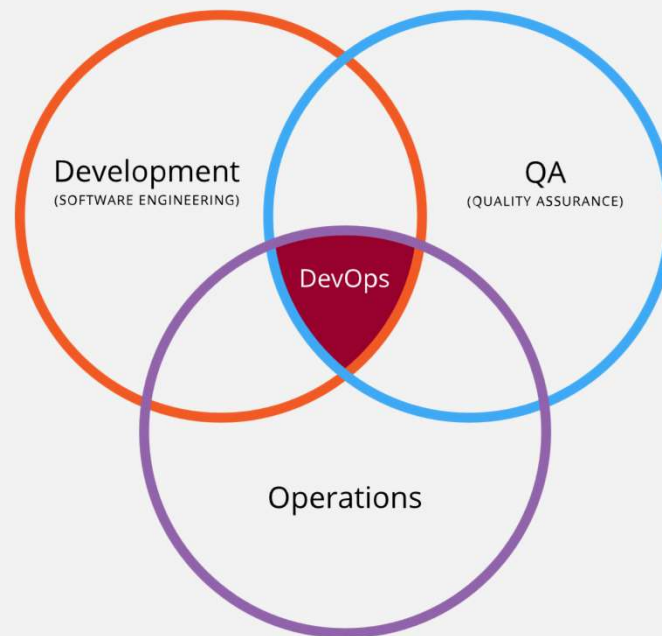
<https://fr.slideshare.net/rtang03/dev-ops-storyboardv1>

<https://fr.wikipedia.org/wiki/Devops>

DevOps

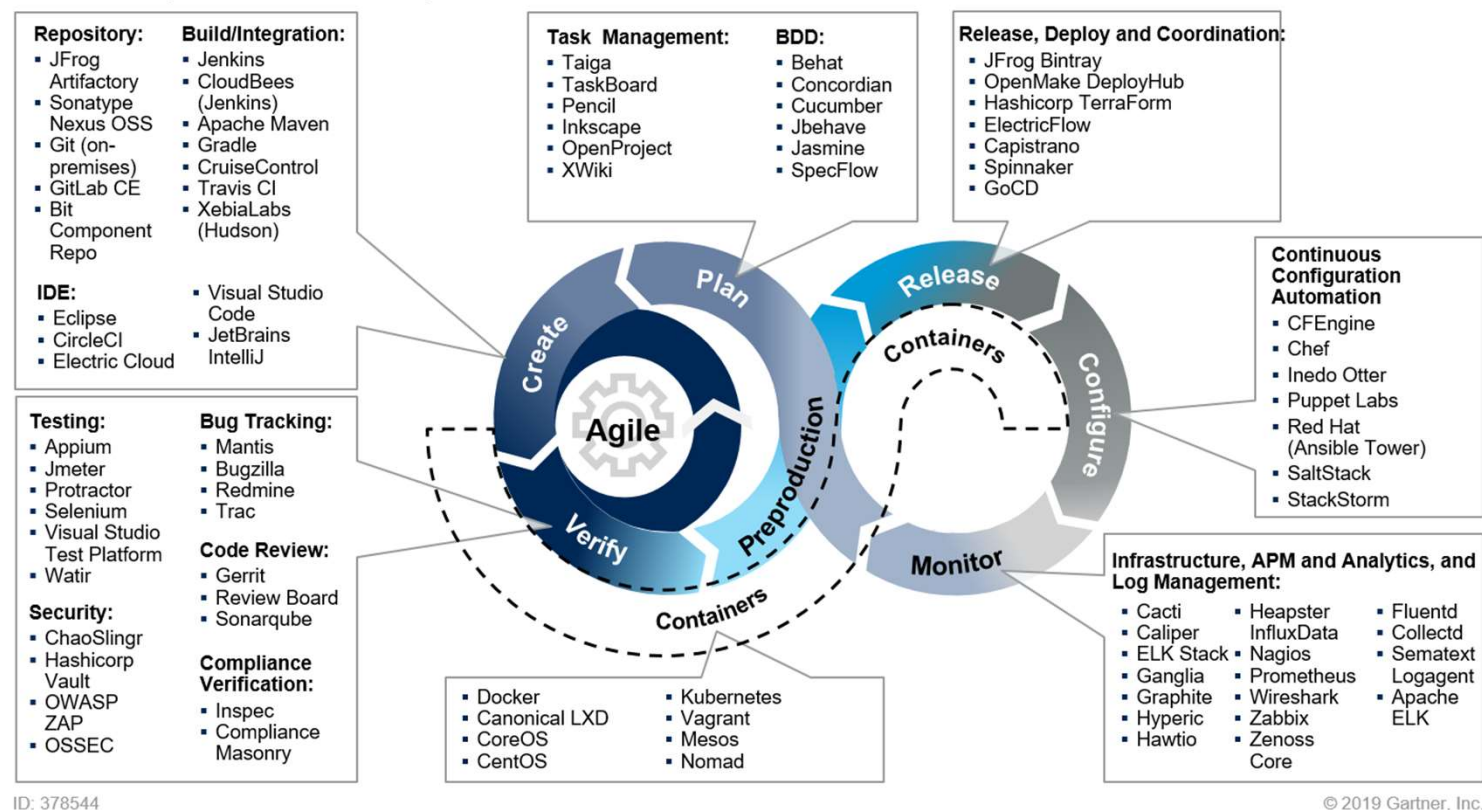
- Contraction de « développement » et d'« opération »
- Idée, c'est de faire collaborer les deux mondes, de rapprocher les équipes
⇒ ce n'est donc pas un outil !
- Pratiques DevOps
 - Utilisation des méthodes agiles pour le développement mais aussi pour l'exploitation
 - Automatiser le plus possible :
 - Création des artefacts
 - Tests (qualité ↗, coût des erreurs ↘)
 - Infrastructure
 - Installation
 - Boucle de rétroaction :
 - Livraison en continue : fréquence ↗, maîtrise du processus de déploiement ↗
 - Livrer de petits incréments : difficulté ↘
 - Livrer souvent : pratique ↗
 - Avoir une rétroaction rapide : ajustement ↗, qualité ↗

DevOps



DevOps

Build an Open-Source DevOps Toolchain

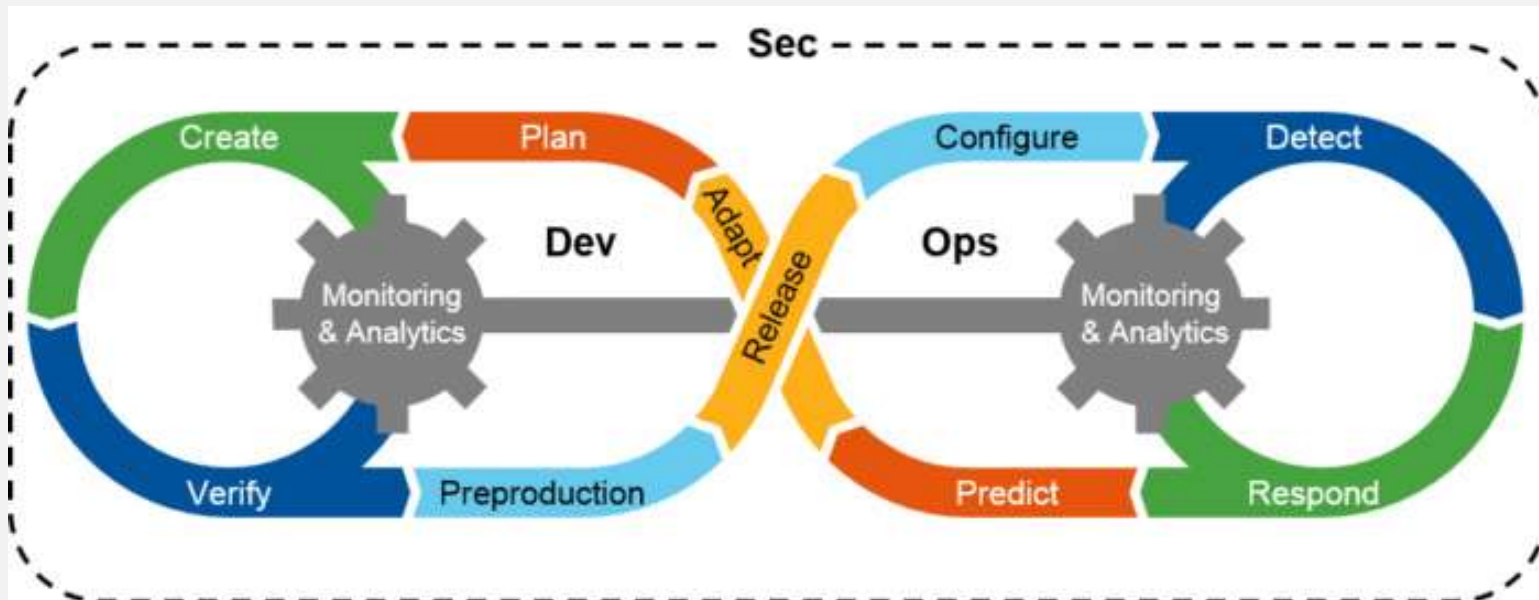


ID: 378544

© 2019 Gartner, Inc.

<https://blogs.gartner.com/manjunath-bhat/2019/02/08/four-steps-to-adopt-open-source-software-as-part-of-the-devops-toolchain/>

DevSecOps



Intégration continue / déploiement continu

- L'intégration continue / Continuous integration (CI) :
 - Générer des artefacts à partir d'un gestionnaire de source / source control management (SCM)
 - Exécuter les tests
- Le déploiement continu / continuous delivery (CD) :
 - Prendre les artefacts
 - Les déploiements dans les environnements ciblés
 - Souvent, il y a un flux de travail pour migrer sur plusieurs environnements avec des validations à chaque niveau

On parle souvent de CI/CD

Intégration continue - principes

L'intégration continue consiste à avoir un ensemble de processus automatisés qui s'enclenchent lorsque la production pousse de nouvelles choses.



Automatisme ne veut pas dire Perte de contrôle, il possible de débrayer certain comportements ou d'avoir une ou plusieurs validations manuelles.

Dans le jargon du développement, on dira qu'on « **build et est test lors des commits** »



Parmi ces processus, on vas également automatiquement mettre à jour l'avancée dans le sprint courant.

Les process d'intégration continue doivent être effectuées sur des environnements neutres

Déf . : L'intégration continue ou *Continuous Integration (CI)* en anglais, est un ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développé.

Intégration continue - objectifs

S'assurer que la production actuelle ne comporte pas de régression.

S'assurer que la version actuelle est dans un état livrable et compatible

Maintenir à jour l'avancée du sprint et vérifier si les délais et objectifs sont tenus.

Avoir toujours à disposition une version pour tester, faire une démo ou livrer.

Intégration continue - avantages

D'être réactif vis-à-vis du client (démonstration, livraison, test)

D'avoir un feedback immédiat sur la production effectuée, grâce aux tests automatisés et à la mise à jour automatique du travail.

Éviter les problèmes de déploiement ou l'intégration de nouvelles ressources au projet.

Être en capacité de livrer une correction ou de relivrer n'importe quelle version de l'application car tout est historisé.

Références

- <https://fr.wikipedia.org/wiki/Devops>
- <https://docs.microsoft.com/en-us/dotnet/architecture/containerized-lifecycle/>
- <https://channel9.msdn.com/Events/Microsoft-Azure/Azure-DevOps-Launch-2018/A101/player>
- <https://channel9.msdn.com/Events/Microsoft-Azure/Azure-DevOps-Launch-2018/A102/player>