

Introduction au DevOps

Design

Petite histoire du Devops



2008
Andrew Shafer

DEVOPSDAYS



2009
DevOpsDay

2007
Patrick Debois



2009
John Allspaw
Paul Hammond



Petite histoire du Devops

Patrick Debois



Résumé

L'histoire commence en Belgique en 2007 avec notre premier protagoniste : **Patrick Debois** qui souhaitait apprendre tout ce qu'il y avait à apprendre de l'informatique , et ce depuis tous les angles possibles.

Consultant pour le gouvernement

Mission

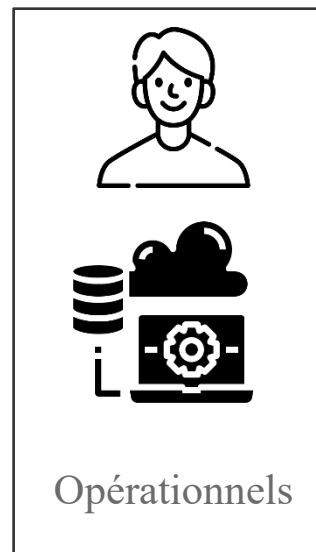
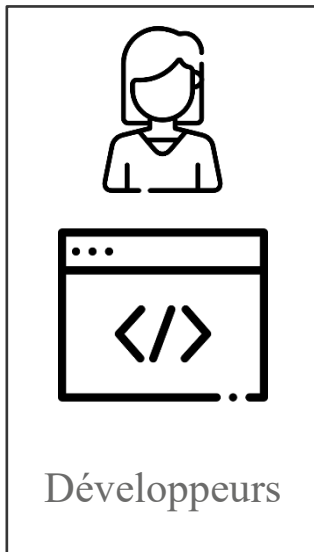


- Projet de migration de Datacenter
 - Chargé de la partie consistant à réaliser les tests sur ce projet
- => Enormément de temps à discuter avec les groupes de **développeurs** et les groupes d'administrateurs réseaux (aussi appelé **Ops**) pour réussir au mieux sa mission.

Petite histoire du Devops

Développeurs

- Méthodologie Agile (Scrum)
- Délivrer un maximum de code
- Beaucoup de temps à attendre
- Difficultés dû aux Ops



Opérationnels

- Impératifs de Production
- Imprédictibilité des infrastructures
- Fonctionnement Cycle en V
- Difficultés dû aux Devs

*« Comment est-il possible de concilier ces différents
cœurs de métiers pour les amener à retrouver
confiance et travailler ensemble de manière
beaucoup plus efficace? »*

Petite histoire du Devops



2008
Andrew Shafer

DEVOPSDAYS



2009
DevOpsDay

2007
Patrick Debois



2009
John Allspaw
Paul Hammond



Petite histoire du Devops

Conférence d'Andrew Shafer



Il souhaite présenter une idée un peu particulière, qu'il appelle l' **Agile Infrastructure**. Une idée selon laquelle les Ops pourrait travailler de manière agile.

Une seule personne s'est présentée pour discuter de ce sujet avec Andrew...Et c'est Patrick Debois.



Patrick était venu à cette conférence pour effectuer une présentation sur l'utilisation de la méthode Scrum et des autres pratiques agiles au sein d'un contexte Ops

Petite histoire du Devops



2008
Andrew Shafer

DEVOPSDAYS



2009
DevOpsDay

2007
Patrick Debois



2009
John Allspaw
Paul Hammond



Petite histoire du Devops

Velocity O'Reilly

Juin 2009, lors du Velocity O'Reilly, **John Allspaw** et **Paul Hammond** donnent une conférence appelée:
« **10+ Deploys per day: Dev and Ops Cooperation at Flickr** »



Patrick Debois regarde avec intérêt la retransmission de cette conférence par streaming vidéo depuis la Belgique mais Twitte qu'il est extrêmement déçu de ne pas avoir pu assister à la conférence.

Paul Nasser, lui rétorque alors qu'il n'a qu'à organiser son propre événement.

Petite histoire du Devops

DevOps Days

DEVOPSDAYS



Cet événement connaît un énorme succès, aussi bien au niveau des développeurs, des administrateurs systèmes et réseau, des managers, des responsables et autres

Une fois la conférence Dev Ops Day terminée, les participants continuent d'échanger sur Twitter via le hashtag #DevOps

C'est alors que de nombreuses personnes cherchent à partager leur propre expérience et à débattre d'idées sous cette nouvelle bannière DevOps.

Petite histoire du Devops

« Il est difficile de savoir où l'on va si on ne sait pas d'où l'on vient. »



Le DevOps:

- A été créé par des acteurs de l'IT pour des acteurs de l'IT (pas par un éditeur de solutions, ou par des analystes)
- N'est ni un produit, ni une spécification, ni un standard et encore moins un titre.
- N'appartient à aucune organisation qui centralise l'ensemble des pratiques ou méthodes DevOps
- Est décentralisé et ouvert à tous!
- Est un mouvement qui se base sur l'empirisme (l'acquisition de connaissances par l'expérience).

Définition du DevOps



Prenons l'exemple d'une entreprise nommée
« **BackBone Blue** »

La viabilité et le succès dépend largement de sa
capacité à proposer de **nouveaux** produits et
services **innovants** à ses clients, et ce plus **vite**
que ses concurrents.

Définition du DevOps

Bob – le développeur



Missions



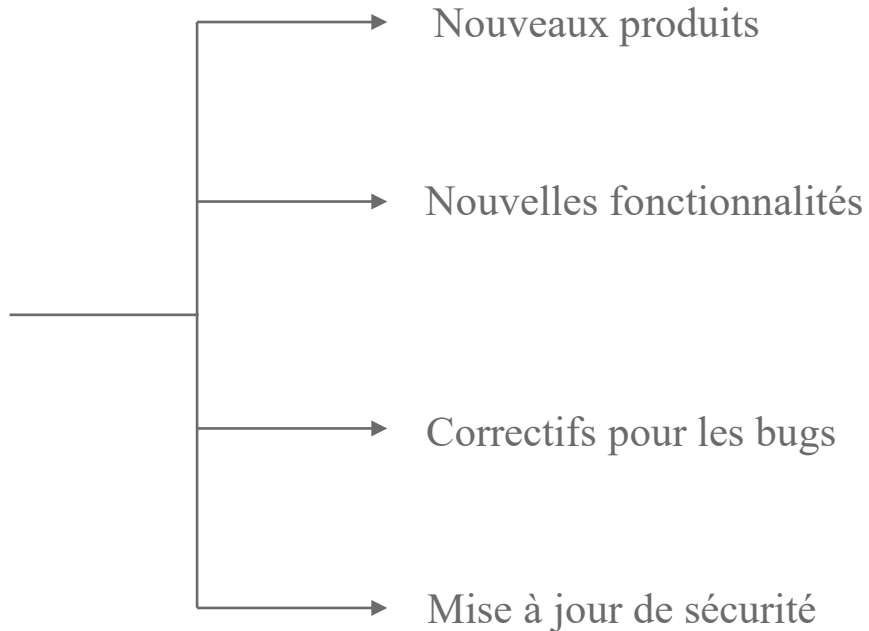
Problématiques

Définition du DevOps

Missions du développeur

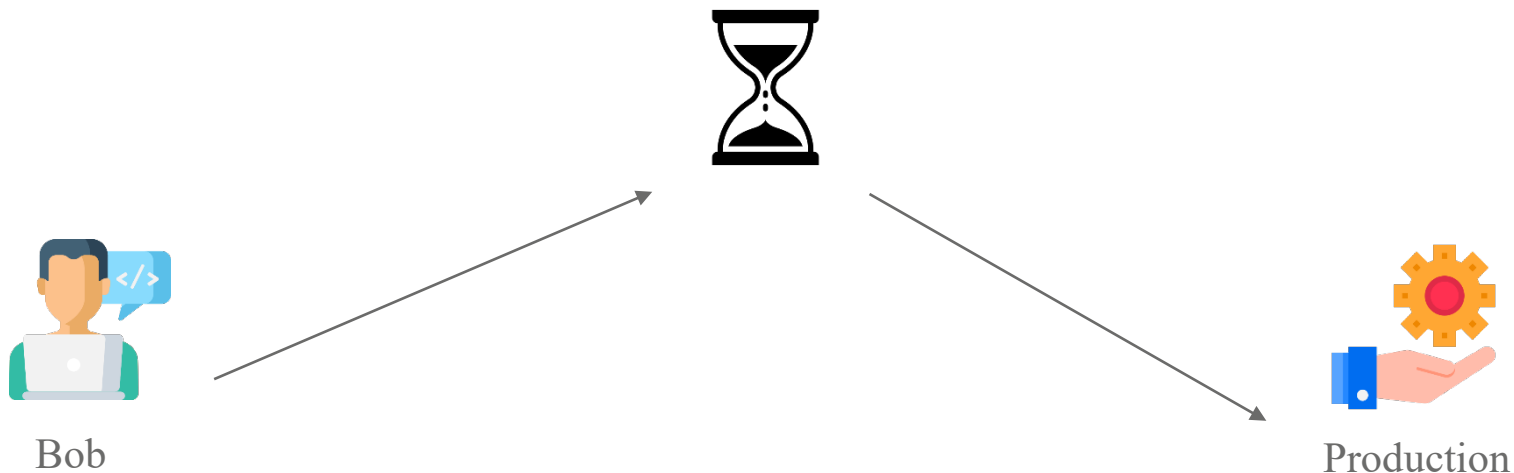


Bob



Définition du DevOps

Problématiques rencontrées par Bob



Définition du DevOps

Alice – l'opérationnelle



Missions



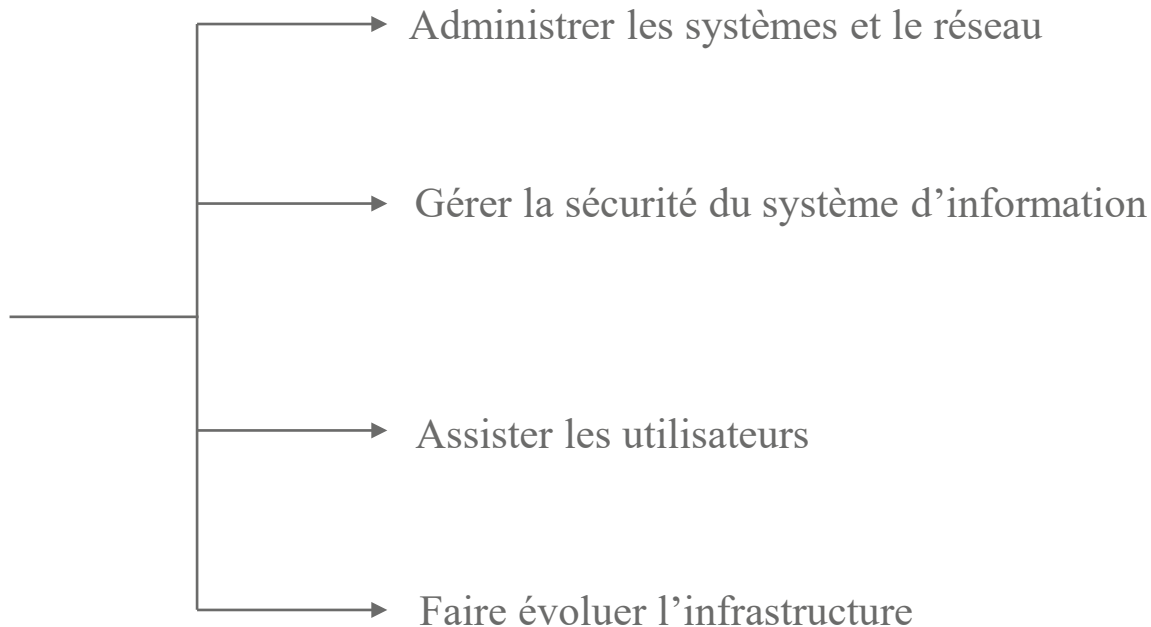
Problématiques

Définition du DevOps

Missions du développeur

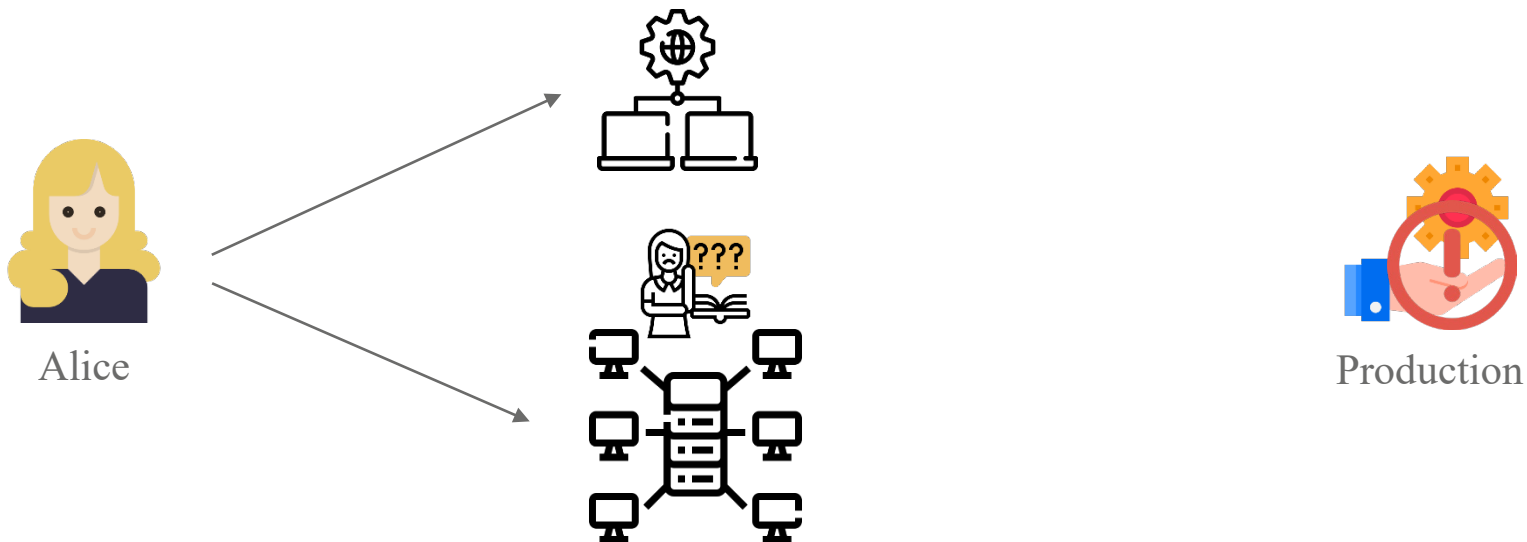


Alice



Définition du DevOps

Problématiques rencontrées par Alice



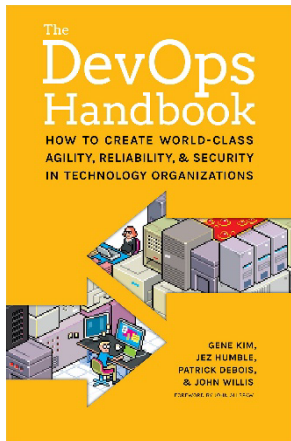
Définition du DevOps

Citations

« Imaginez un monde où les propriétaires du produit, [du service], les équipes de développement, de qualité, les opérations, et la sécurité travaillent **ensemble**. »

« Ils ne sont pas juste là pour **s'entraider**, mais aussi pour garantir la **réussite** de l'organisation, à laquelle ils appartiennent, dans son ensemble. »

« En travaillant vers **un objectif** commun, ils permettent un **flux** rapide des travaux planifiés vers la production, tout en assurant **stabilité, fiabilité, disponibilité**, et une sécurité à toutes épreuves»



Définition du DevOps

Définition

« C'est un groupe d'individus qui travaillent de manière transversale peu importe le métier auquel ils appartiennent pour s'entraider et garantir la réussite de la société avec un objectif commun. »

Définition du DevOps

Ce que le DevOps n'est pas

Titre

Uniquement
de
l'automatisation

Uniquement
une culture

Stratégie
identique
pour tous

Equipe
séparée

Outil

Anarchie

Définition du DevOps

Le DevOps n'est pas un titre

Un ingénieur DevOps ne veut pas dire grand-chose.

Certes il y a des coachs et des évangélistes qui peuvent accompagner les équipes et les entreprises dans leur organisation et leur transformation vers le DevOps, mais cela ne s'appelle en aucun cas un ingénieur DevOps.



Définition du DevOps

Le DevOps n'est pas une équipe séparée



A l'origine de tout: la volonté de réunir les Devs et Ops au sein d'une même équipe, avec un objectif commun, pour les faire avancer tous dans la même direction correspond aux fondements même du DevOps.

Définition du DevOps

Le DevOps n'est pas un outil

Les outils peuvent aider à s'organiser et à mieux gérer nos tâches, nos déploiements et notre travail d'équipe.

Cependant, n'oubliez pas que ce ne sont pas les outils qui ont amené le DevOps, mais bien la méthodologie et la culture DevOps qui a provoqué l'émergence de nouveaux besoins et donc de nouveaux outils.



Définition du DevOps



Le DevOps n'est pas UNIQUEMENT une culture

C'est en connaissant son histoire que vous pourrez répondre que le DevOps, c'est aussi une philosophie, des retours d'expériences, des idées, de l'automatisation, ou encore de l'organisation et de la méthodologie.

Définition du DevOps

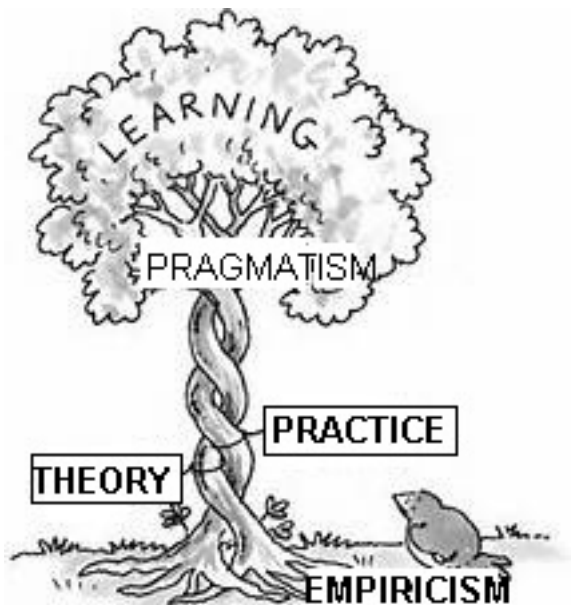
Le DevOps n'est pas UNIQUEMENT de l'automatisation

L'automatisation ne correspond qu'à une infime part de ce qu'est réellement le DevOps.

En effet, le DevOps c'est aussi une culture, une philosophie, des retours d'expériences, des idées, etc..



Définition du DevOps



**Le DevOps n'est pas une MÊME
STRATEGIE pour tous**

Empirisme et adaptation sont à la base du
DevOps

Souvenez-vous que le DevOps a été conçu
par des acteurs de l'IT, pour des acteurs de
l'IT en se basant sur les expériences de tout
le monde.

La valeur du DevOps

La valeur apportée par le DevOps?

Objectifs

Changement de
paradigme

Améliorations

Apportées par la
pratique du DevOps

CALMS

5 piliers du
DevOps

Automatisation

Augmenter l'agilité, la
robustesse, la rapidité
et l'évolutivité de
l'infrastructure.

La valeur du DevOps

Les objectifs du DevOps

- Déploiements plus **petits** et plus **fréquents**
- Réduction des **risques** et de **l'effort** nécessaire
- Réduction des **coûts**, des **itérations** et des **délais**
- Culture de **communication** et de **collaboration**
- Constance et rapidité grâce à **l'automatisation**

La valeur du DevOps

Les améliorations apportées par le DevOps

1. Amélioration du Time to Value
2. Intégration des pratiques avec le Business
3. Réactivité accrue des équipes
4. Code et déploiement de meilleure qualité
5. Productivité plus importante
6. Visibilité des actions menées

La valeur du DevOps

Les 5 piliers du DevOps (CALMS)

CULTURE

Focus sur les gens, les changements et l'expérimentation

AUTOMATISATION

Déploiement Continu et Infrastructure as Code (IaC)

LEAN

Focus sur la production de valeur et la faible taille des release

MESURES

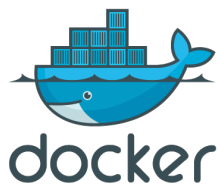
Tout mesurer ! Montrer l'amélioration et la progression

SHARING

Partage ouvert de toutes les informations => Collaboration et Communication

La valeur du DevOps

L'automatisation au cœur du DevOps



L'importance du DevOps

Pourquoi le DevOps prend de l'importance?

Cadence de
production

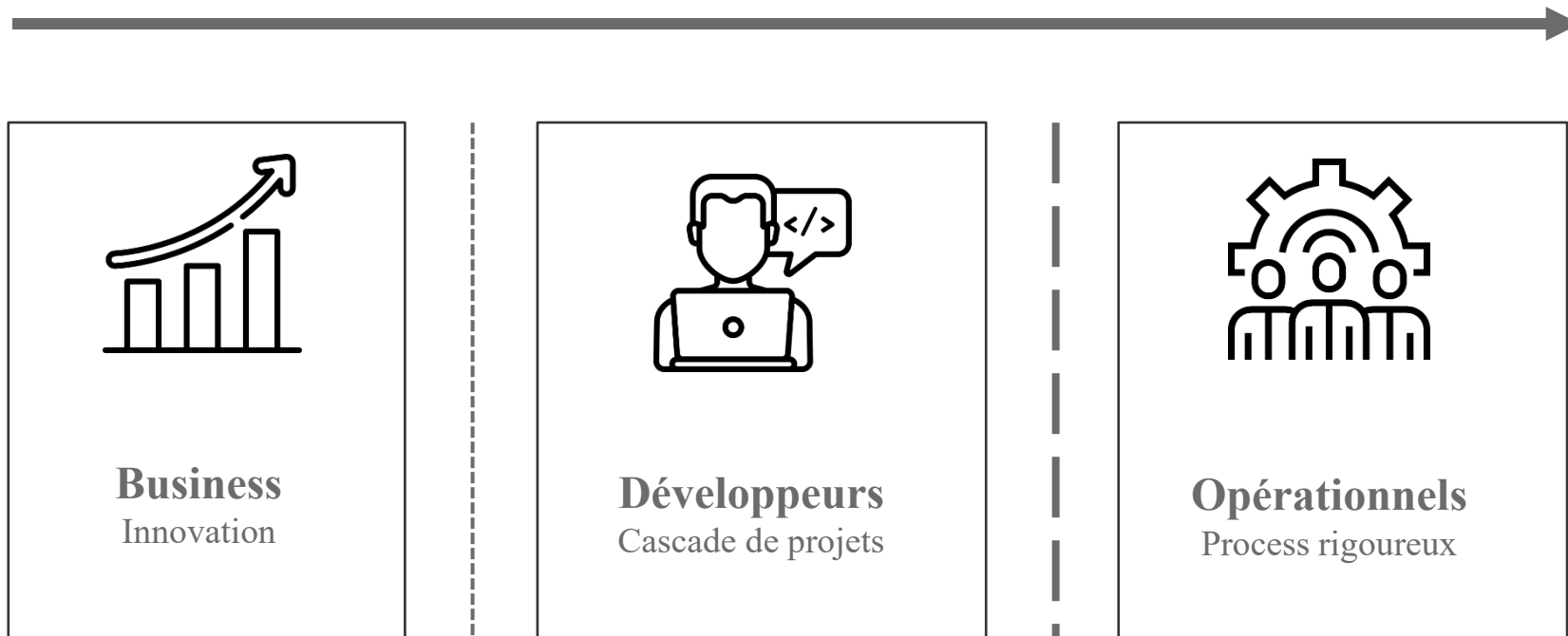
Quelques
statistiques

Retours d'expériences !



L'importance du DevOps

Cadence de production



L'importance du DevOps

Quelques statistiques

D'après « **Accelerate State of DevOps de 2019** », les entreprises qui ont une organisation DevOps avancée:

208 fois plus de déploiements de code

106 fois plus rapide du commit jusqu'au déploiement

2604 fois plus rapide pour un retour après incident

7 fois moins d'incidents en cas de changement

L'importance du DevOps

Pourquoi le DevOps prend de l'importance?

Cadence de
production


Quelques
statistiques

Retours d'expériences !



L'importance du DevOps

Compuware



Réduction du
temps de
déploiement

66%

Délivré **2**
fois plus de
code par
développeur

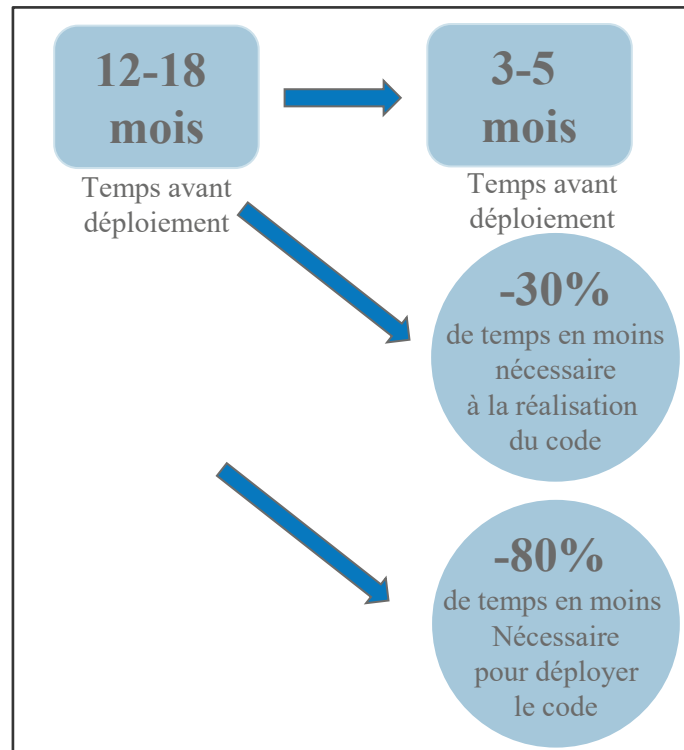
Réduction de
7% des
bugs

L'importance du DevOps

IBM

Il a été décidé d'arrêter tous les développements de l'équipe concernée pendant **4 mois**.

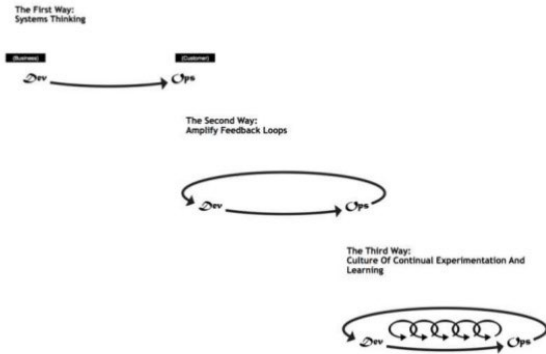
=> Temps de mettre en place les outils et l'organisation nécessaires.



Le « Three Ways »

Three Ways de Gene Kim

“3 Ways” of DevOps



Nous allons aborder le « Three Ways » de Gene Kim correspondant à un certain nombre de principes à partir desquels, tous les modèles DevOps sont et pourront être dérivés.

Ce sont ces principes qui sont abordés dans le livre de référence « DevOps Handbook »

Le « Three Ways »

Three Ways de Gene Kim

- ✓ Définir le « First Way » et la notion de flux
- ✓ Comprendre l'importance du feedback dans le « Second Way »
- ✓ Nous attarder sur l'expérimentation et l'apprentissage continu
introduit dans le « Third Way »
- ✓ Aborder les concepts d'Ingenierie du Chaos avec Netflix

Le « First Way »

Le « First Way » et la notion de flux

Le « First Way »

Qu'est ce que c'est ?

Théorie des contraintes

M. Eliyahu Goldratt

Le « First Way »

Qu'est ce que le « First Way »?

Le « **First Way** » souligne l'importance de la performance de **tout le système** et non pas d'un département ou d'un service spécifique.

(Business)

DEV

(Clients)

OPS



Consiste à **comprendre et améliorer le flux de travail** entre les développeurs et les opérationnels.

Le « First Way »

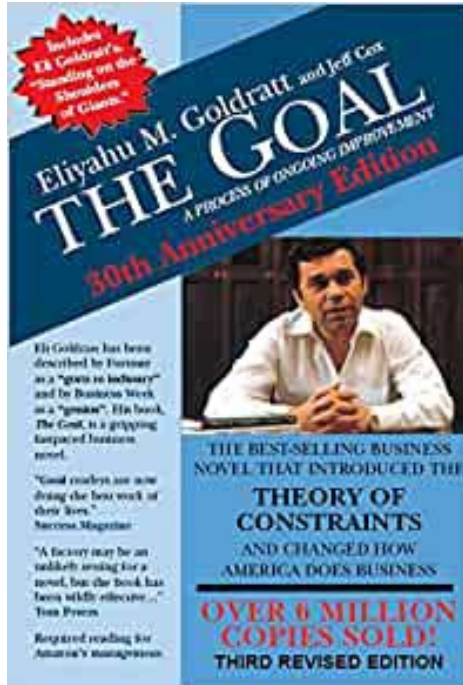
Implications du « First Way »

- ❖ Améliorer et augmenter ce flux grâce à la compréhension de l'ensemble des contraintes qui existent (de la gauche vers la droite)
- ❖ Un défaut connu ne sera jamais ignoré, et ne pourra pas passer en aval sans être traité.
- ❖ Interdiction des optimisations locales qui pourraient causer une dégradation globale.

Nous utilisons CentOS pour faire nos développements, même si la production est sur Debian, on se sent plus à l'aise sur ce système

On ne peut pas valider ce processus, il faut demander à Jacques du service Testing d'effectuer une revue, même s'il ne sait pas en quoi il consiste »

La théorie des contraintes



M. Eliyahu M. Goldratt a introduit cette théorie dans son ouvrage « The Goal » en 1994.

La TOC (la théorie des contraintes) est une méthodologie permettant d'identifier le facteur limitant le plus important (la contrainte).

=> Améliorer systématiquement la contrainte jusqu'à ce qu'elle ne soit plus le facteur limitant.

La théorie des contraintes

Les étapes du TOC

- 1 Identifier la contrainte
- 2 Réfléchir à comment maximiser l'utilisation de la contrainte
- 3 Subordonner l'organisation et toute l'action à l'étape 2
- 4 Résoudre ou réduire la contrainte
- 5 Retour à la première étape

Exemple de contraintes

- Délais de développement
- Création des environnements
- Installation et exécution des tests
- Déploiement du code
- Architecture obsolète
- Gestion trop complexe des processus administratifs
- ...

Le « Second Way »

Le « Second Way »

L'importance du feedback

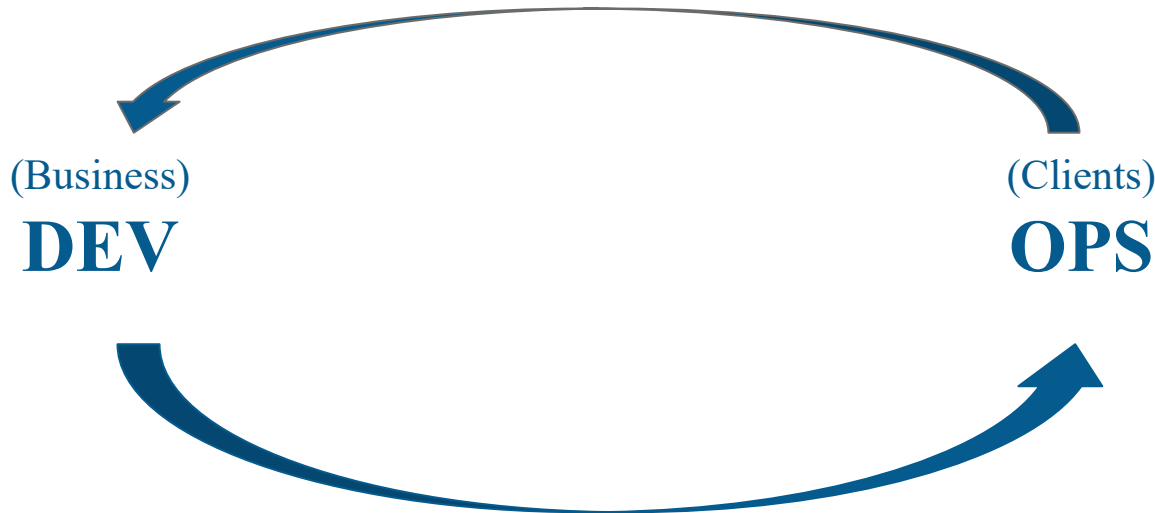
Exemples

Mieux comprendre le
« Second Way »

Le « Second Way »

Qu'est ce que le « Second Way »?

Le « Second Way » souligne l'importance du Feedback. On peut ainsi réduire le délai en effectuant des boucles de feedback plus courtes.

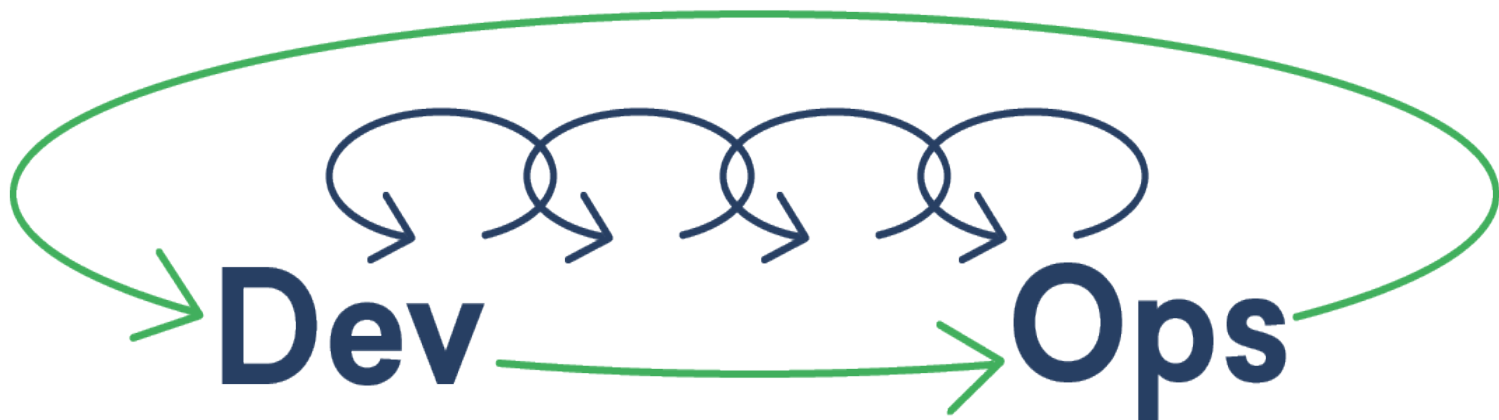


Cette boucle incite à comprendre et répondre aux besoins de tous les clients internes comme externes.

Le « Third Way »

Qu'est ce que le « Third Way »?

Le « Third Way » souligne l'idée de créer une culture qui favorise l'expérimentation continue incluant de prendre des risques et d'apprendre de ses échecs.



Le « Third Way »

Qu'est ce que le « Third Way »?

La **répétition** et la **pratique**, et donc les échecs potentiels qui en découlent, sont des prérequis indispensables à la **maîtrise**.

Allouer du temps spécifiquement dédié pour améliorer le travail quotidien.

Créer des **rituels** permettant de récompenser l'équipe lorsqu'elle prend des risques

Planifier du temps pour effectuer des expérimentations en toute sécurité et participer à l'innovation (hackathons)



L'ingenierie du Chaos

Qu'est ce que l'ingenierie du Chaos?



Consiste à mettre en place un outil en production qui va VOLONTAIREMENT causer des pannes.

La résilience des logiciels conçus devient une **obligation** et non plus une option.

Chaos Monkey et Netflix

En 2011, Netflix choisit au hasard des instances AWS en production et les met volontairement **hors service**.

Ces robots, codes et concepts ont été rendus OpenSource en 2012 et disponible à l'adresse:
<https://netflix.github.io/chaosmonkey/>



L'ingenierie du Chaos

La Simian Army



Chaos Gorilla Rend hors service une zone complète de disponibilité AWS

Chaos Kong Rend hors service une région complète de disponibilité AWS

Janitor Monkey Rend hors service toutes instances non utilisées afin d'éviter la surconsommation de ressources

Latency Monkey Introduit des délais dans les échanges permettant de tester la tolérance en cas de diminution de performances d'un composant externe

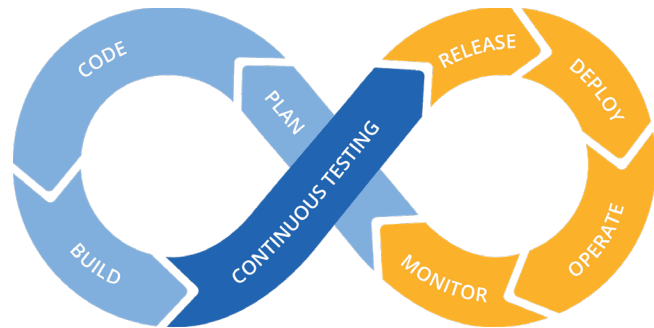
L'ingenierie du Chaos

Le DevOps

Le **Chaos Monkey** contribuent directement à la boîte à outils DevOps en répondant au besoin de **Continuous Testing**.

Cela s'inscrit dans le modèle **Design For Failure** (conçu pour supporter la défaillance).

Une application ou un service doit être capable de supporter la panne de n'importe quel composant logiciel ou matériel sous-jacent.



Le DevOps

Il est temps d'aborder les aspects techniques du DevOps, notamment en traitant les différentes couches techniques qui sont nécessaires au « Three Ways » abordées précédemment.

Le DevOps

- Comment les tests automatiques sont-ils mis en œuvre dans les équipes DevOps?
- Qu'est ce que l'intégration continue?
- Le Continuous Delivery au service d'un service toujours opérationnel et utilisable par les clients.
- Qu'est ce que le SRE (Site Reliability Engineering)
- Le DevSecOps et le ChatOps, d'autres métiers du DevOps.

Le « Continuous Testing »

Le « Shift Left »

Intégration de la qualité
dans le processus de
développement

Les Tests Fonctionnels

Qu'est ce que c'est?

Les Tests Non Fonctionnels

Qu'est ce que c'est?

Les pratiques techniques du Devops

Shift Left

Continuous Testing



Correspond au processus d'exécution de tests automatiques.

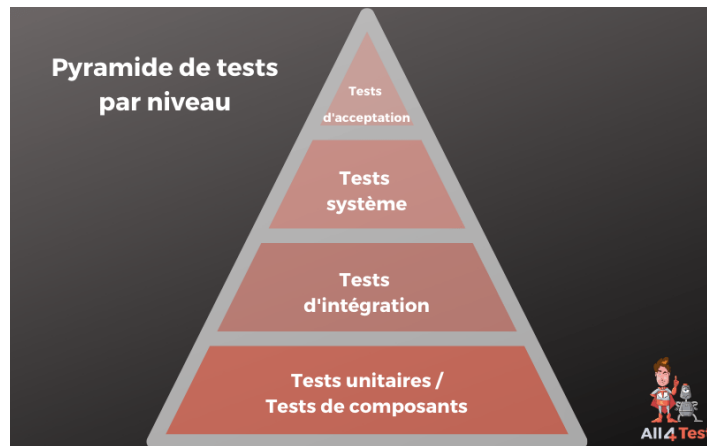
Ces tests doivent être intégrés au pipeline de déploiement du code pour obtenir un feedback le plus rapidement possible.

Vérifie l'absence de risque business associé à la nouvelle version.

Les tests fonctionnels

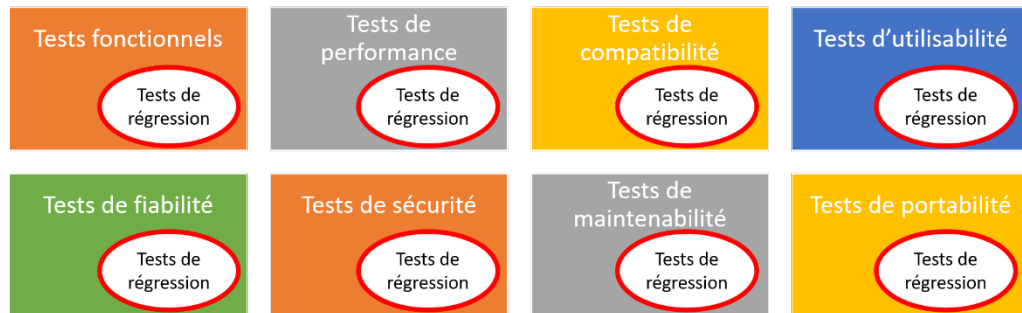
Parmi les tests fonctionnels, on retrouve les plus importants:

- Tests Unitaires
- Tests d'API
- Tests d'intégration
- Tests de système



Les pratiques techniques du Devops

Les tests non-fonctionnels



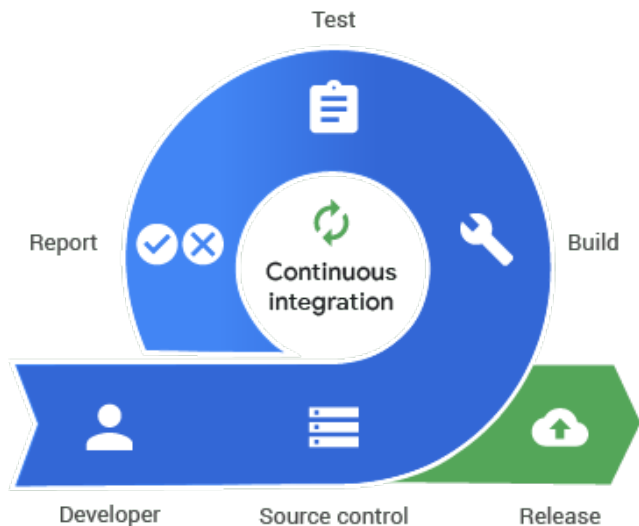
Parmi les tests non-fonctionnels, on retrouve les plus importants:

- Tests de performance
- Tests de sécurité
- Tests de compatibilité

Ce sont les tests qui permettent de tester les performances de la fonctionnalité.

Les pratiques techniques du Devops

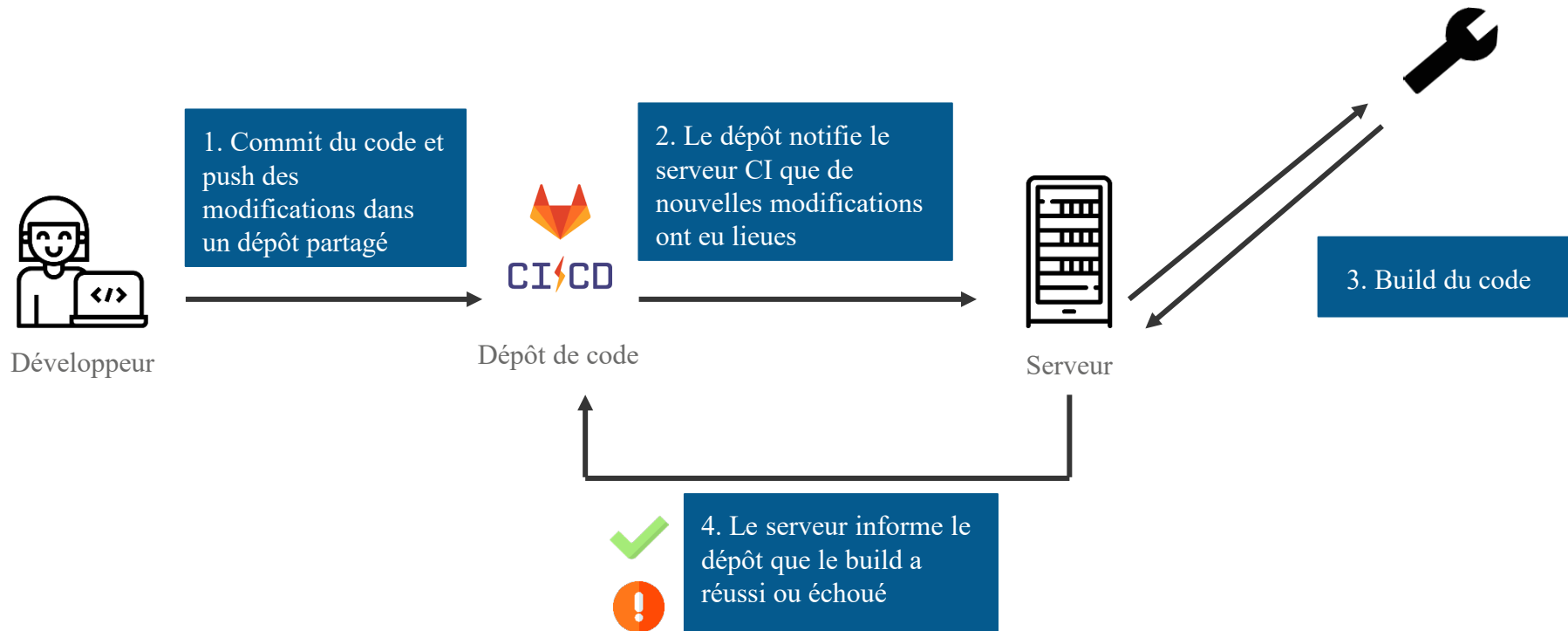
L'intégration continue (CI)



L'intégration continue correspond à des pratiques de développement, impliquant le commit du code dans un dépôt partagé par les développeur.

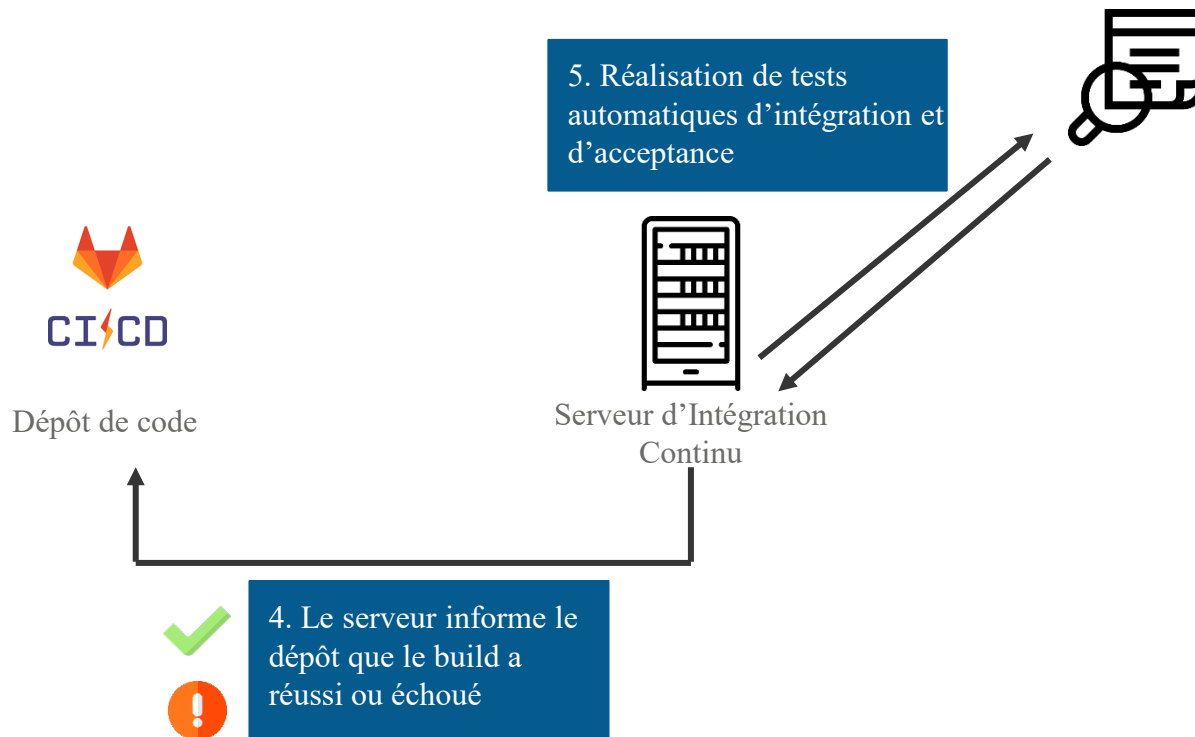
Les pratiques techniques du Devops

La phase de Build



Les pratiques techniques du Devops

La phase de Tests



Les pratiques techniques du Devops

Les pré-requis

- Définir le standard de code que l'on souhaite utiliser dans l'organisation (PEP8, FLAKE8, Pylint, etc..)
- Implémenter un dépôt de code permettant du versionning (Github, Gitlab)
- Mettre en place des serveurs d'Intégration Continue pour collecter, builder et tester les codes qui ont été comités.
- Effectuer ces tests sur des environnements proches de ceux de production.



Continuous Delivery



Capacité à fournir un software (ou du code) qui est toujours dans un **releasable state** (disponible pour l'utilisation)) travers son cycle de vie.

Amène l'**intégration continue** à un niveau supérieur.

Les pratiques techniques du Devops

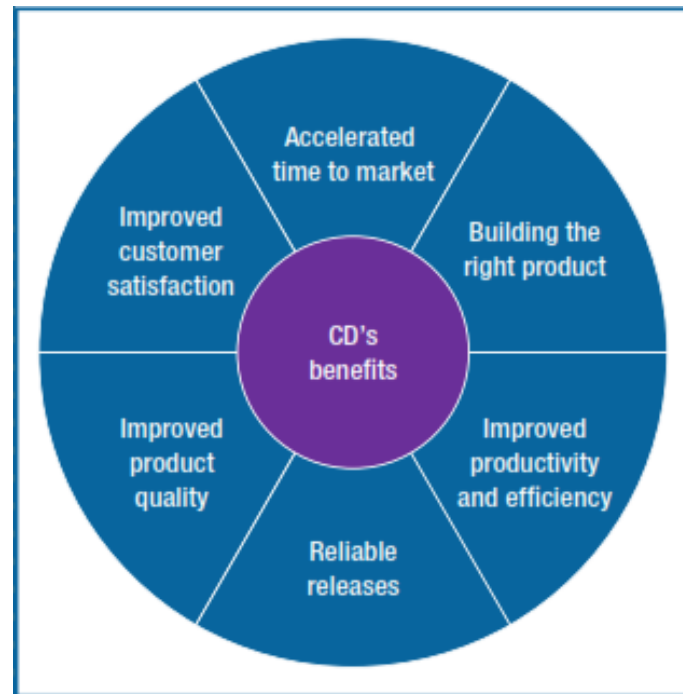
Les bénéfices du Continuous Delivery

Fournir rapidement des feedback automatiques sur le système de préproduction et de production.

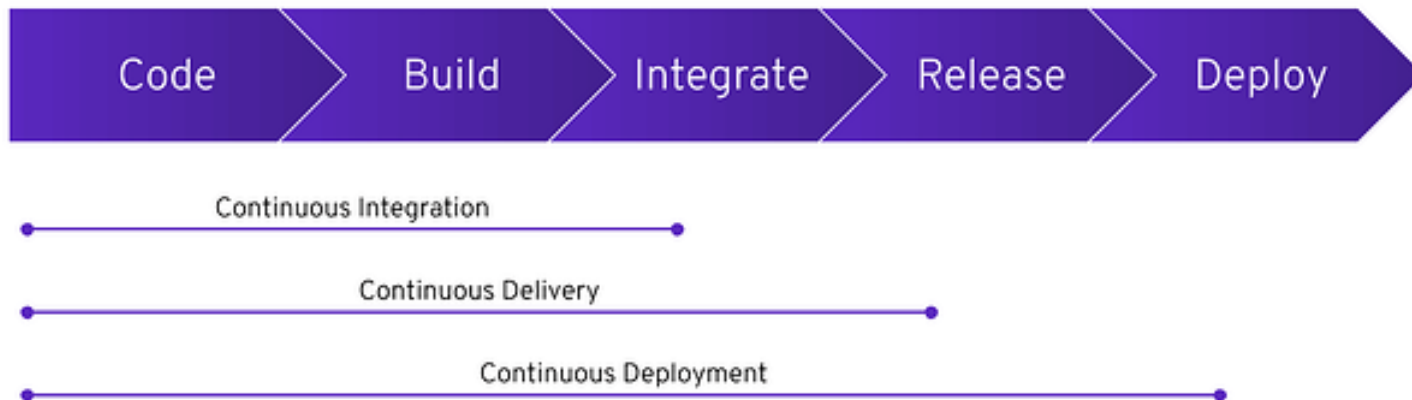
Garde une version logicielle releasable et déployable même lorsque les nouvelles fonctionnalités sont en cours de développement

Mise en place d'un pipeline qui permet de faire du déploiement à la demande

Réduit les coûts, le temps et les risques inhérents aux changement incrémentaux réalisés



Continuous Delivery vs Continuous Deployment



Les pratiques techniques du Devops

Continuous Integration vs Continuous Deployment



Les ingénieurs SRE (Site Reliability Engineering)

Le SRE

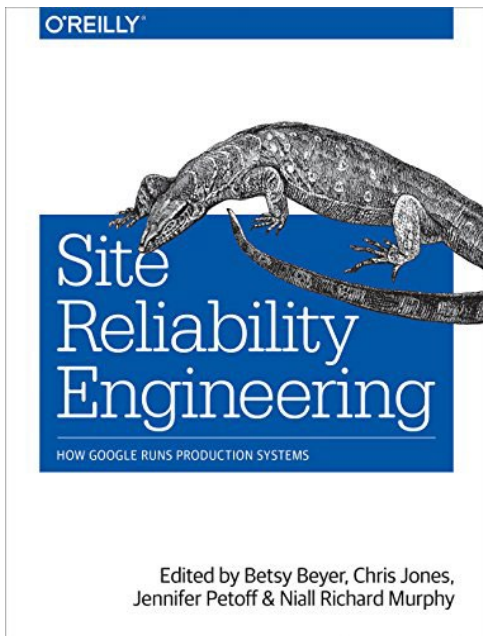
Site Reliability
Engineering

**La « Resilience
Engineering »**

Qu'est ce que c'est?

Les pratiques techniques du Devops

Les ingénieurs SRE (Site Reliability Engineering)



Apparu chez Google

Lorsqu'ils ont commencé à donner des tâches de type « opérationnelles » à des ingénieurs software (qui font du développement logiciel).

● 50% du temps

Tâches de développement comme la création de nouvelles fonctionnalités, de la montée en charge ou de l'automatisation.

Objectifs : Systèmes évolutifs et fiables

Le but du SRE est de pouvoir créer des systèmes software ultra-évolutifs et fiables.

● 50% du temps

Tâches liées aux « Opérations » comme les incidents, les interventions manuelles, les problématiques réseaux...

La « Resilience Engineering »

Un système est résilient s'il peut ajuster son fonctionnement, avant, pendant ou après que des événements (changements, perturbations, et opportunités) se sont déroulés, et ainsi maintenir les opérations fonctionnelles malgré des conditions à la fois attendues, mais aussi inattendues »



Les pratiques techniques du Devops

Les concepts associés

Attaques

La meilleure défense
correspond aux bonnes
attaques (cf. Chaos
Monkey)

Tout indissociable

Nécessite de voir les
fonctions d'une organisation
comme un tout indissociable

L'Humain

Prendre en considération à
la fois les éléments
techniques mais également
humain

Revues post-incidents

Effectuer des revues post-
incidents agressives,
irréprochables et systémiques

Weakest Link

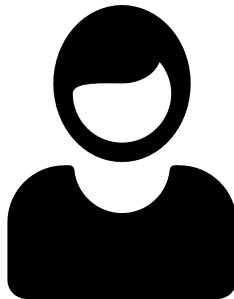
Les systèmes doivent être plus
forts que leur lien le plus faible
(weakest link)

Les autres métiers du Devops

DevSecOps et ChatOps



DevSecOps



ChatOps

Les autres métiers du Devops

DevSecOps

Qu'est ce qu'un DevSecops ?

Le but du
DevSecOps est de créer un réel état
d'esprit dans lequel « Tout le monde est
responsable de la sécurité »



Les autres métiers du Devops

ChatOps

Qu'est ce qu'un ChatOps ?

Le ChatOps est une méthode de collaboration centrée sur la communication, qui connecte les personnes, les outils, les fichiers et l'automatisation dans un flux de travail transparent.

La transparence induite permet d'avoir des boucles de feedback plus courtes, d'améliorer le partage d'informations, la collaboration et la réduction du MTTR (Mean Time to Repair).



Merci pour votre attention!