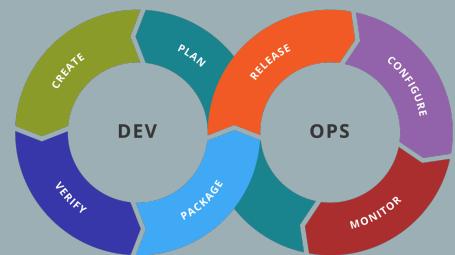


# Optez pour une transformation digitale intelligente

L'INTELLIGENCE ARTIFICIELLE AU SERVICE DE VOTRE BUSINESS.



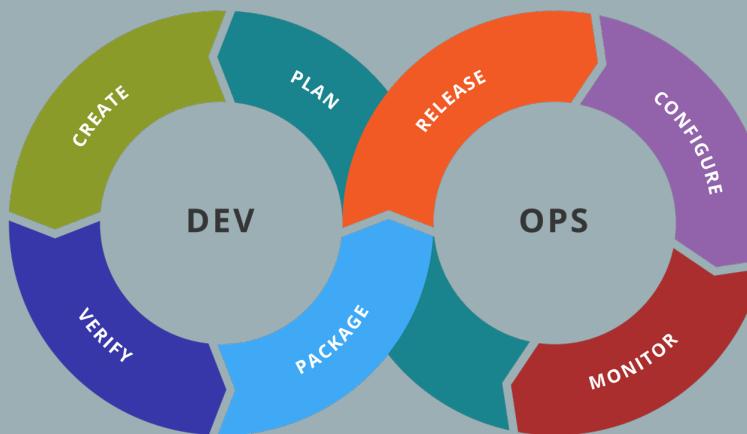
## DEVOPS



# SOMMAIRE

1. Les origines du DevOps
2. Qu'est ce que le DevOps?
3. Les avantages de la démarche DevOps.
4. Les caractéristiques du DevOps.
5. Les objectifs de la méthodologie DevOps.
6. Les principes fondamentaux du DevOps– Three Ways
7. Les pratiques techniques du DevOps.
8. Les Frameworks Business et Techniques utilisés en DevOps

# 0 – Préambule : Fondations et certifications



## 0 – Les Evènements

### Collective Body of Knowledge (CBOK)

Le DevOps émerge à travers un ensemble de  
**connaissances partagées et collectives**



Publications et Conférences



Groupes et rencontres



Vidéos et webinars



Etudes de cas

# 0 – Les Evènements DevOps

## Evénements du DevOps



## 0 – Les Evènements DevOps

### DevOps Enterprise Summit

Conférence qui existe depuis **2014** destinée aux entreprises de taille importante implémentant (ou essayant de le faire) les principes et pratiques DevOps.

Conférences plutôt orientée Business et organisationnelles.

Fondée par **Gene Kim** dont nous reparlerons plus tard.

<https://events.itrevolution.com/>



## 0 – Les Evènements DevOps

### DevOps Days

Les DevOps Days sont nés en Belgique en 2009, sous l'impulsion de **Patrick Debois**.

Séries de conférences techniques, ayant lieues plusieurs fois par mois dans le monde entier couvrant les sujets de développements logiciels, infrastructures, et interactions entre les deux.

<https://devopsdays.org/>



**DEVOPS  
DAYS**



# 0 – Fondation - certifications DevOps

## 1 er organisme

DevOps Institute



Le DevOps Institute est dédié à l'amélioration de l'aspect "humain" pour le succès du DevOps/

Il fournit un certain nombre de ressources, comme des livres blancs, des partenaires, des webinars, etc...

Il permet aussi de préparer un certain nombre de certifications



# 0 – Fondation - certifications DevOps

## Certifications



[https://devopsinstitute.com/  
certifications/devops-  
foundation/](https://devopsinstitute.com/certifications/devops-foundation/)



[https://devopsinstitute.com/  
certifications/devops-  
leader/](https://devopsinstitute.com/certifications/devops-leader/)



[https://devopsinstitute.com/  
certifications/devsecops-  
engineering/](https://devopsinstitute.com/certifications/devsecops-engineering/)

## 0 – Fondation - certifications DevOps

### 2nd organisme

DASA



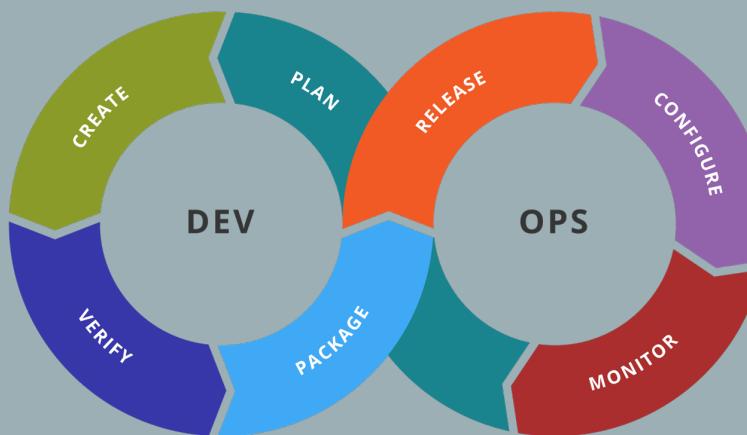
La **DevOps Agile Skills Association (DASA)** se définit comme une association indépendante et ouverte, promouvant le développement des certifications et des ressources pour les DevOps



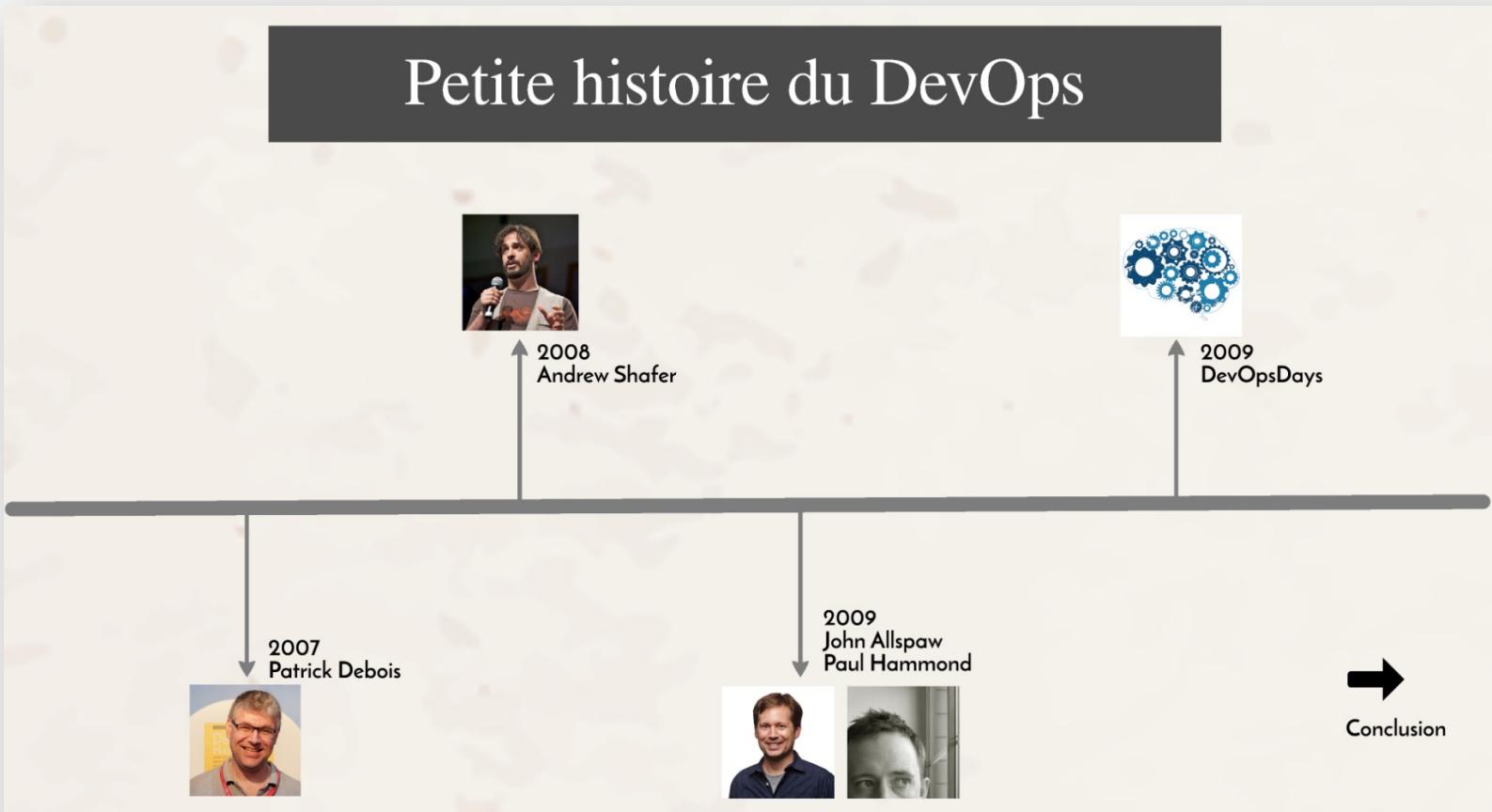
# 0 – Fondation - certifications DevOps



# I- Les origines du DevOps.



# I - Les origines du DevOps.



# I- Les origines du DevOps.

2007 - Belgique - Patrick Debois



## Résumé

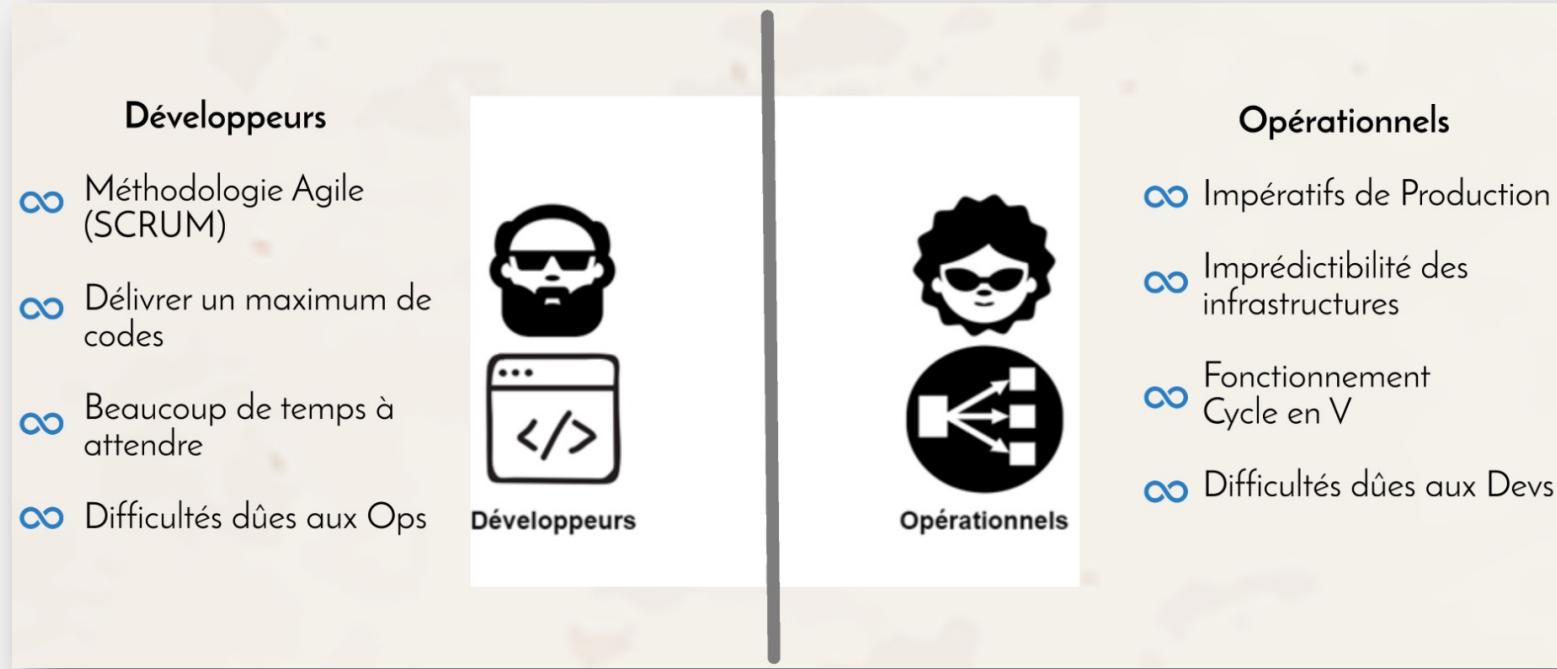
L'histoire commence en Belgique en 2007 avec notre premier protagoniste : **Patrick Debois** qui souhaitait apprendre tout ce qu'il y avait à apprendre de l'informatique, et ce depuis tous les angles possibles.

## I-Les origines du DevOps.

Le fait d'avoir une séparation très nette entre une équipe Dev (Développement) et une équipe Ops (Opérationnelle) est liée à la divergence de leurs objectifs :

- ✓ *En effet, l'objectif principal d'une **équipe ops** est de garantir la stabilité du système.*
- ✓ *De son côté, l'**équipe de développement** a pour objectif principal d'apporter les changements nécessaires au moindre coût et le plus vite possible*

# I - Les origines du DevOps.



"Comment concilier ces différents coeurs de métiers pour les amener à retrouver confiance et travailler ensemble de manière beaucoup plus efficace ?"

## I - Les origines du DevOps.

*La gestion d'une organisation avec des objectifs par équipe indépendants des uns des autres avec des indicateurs spécifiques à chaque équipe aura pour conséquence :*

- ❖ Des optimums locaux.
- ❖ Des guerres entre équipes.

## I - Les origines du DevOps.

2008 - Canada - Andrew Shafer

### Conférence d'Andrew Shafer

Il souhaite présenter une idée un peu particulière, qu'il appelle l' « **Agile Infrastructure** ».

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.

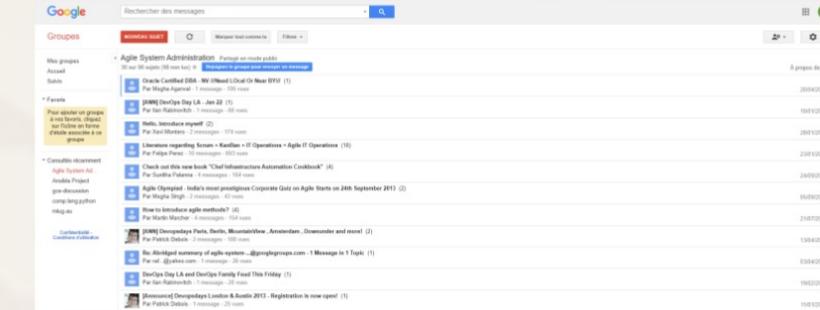


## I - Les origines du DevOps.

### Naissance du premier groupe

Patrick était tellement heureux que quelqu'un d'autre que lui partage sa **frustration**, qu'ils décident ensemble, après avoir discutés un peu, de former le groupe Google : « **Agile System Administration** », lieu de discussion de ce sujet, et qui malgré le peu d'engouement généré, existe toujours.

<https://groups.google.com/forum/#!forum/agile-system-administration>



## I - Les origines du DevOps.

### Velocity O'Reilly



#### Conférence

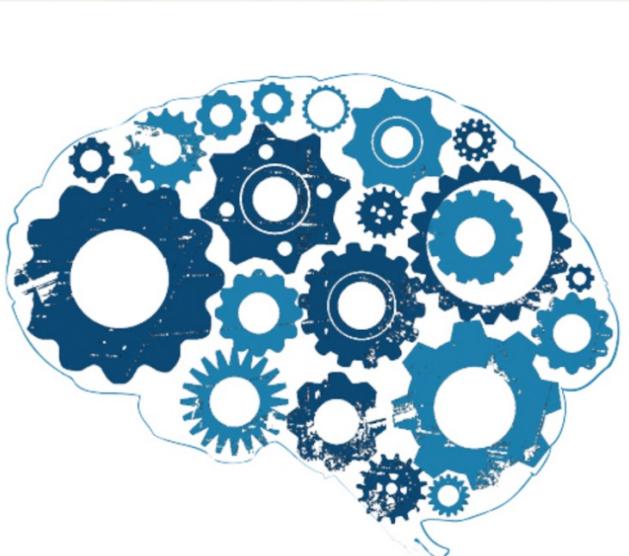
Juin 2009, lors du Velocity O'Reilly, **John Allspaw** et **Paul Hammond** donnent une conférence appelée :  
**« 10+ Deploys per day : Dev and Ops Cooperation at Flickr »**

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 cette conférence.

Paul Nasser, lui retorque alors qu'il n'a qu'à organiser son propre évènement...

## I - Les origines du DevOps.

### Les premiers DevOps Days



- ∞ Cet évènement connaît un énorme succès, aussi bien au niveau des développeurs, des administrateurs système et réseau, des managers, des responsables et autres....
- ∞ Une fois la conférence DevOps Days 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.

# I - Les origines du DevOps.

## Pourquoi cet historique est important ?

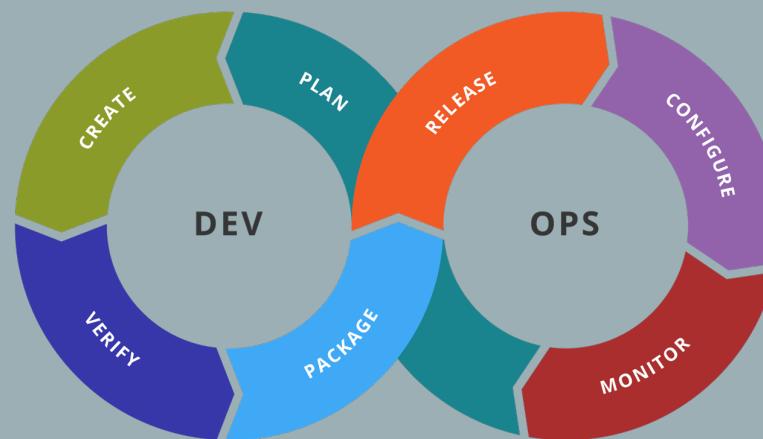
*"Il est difficile de comprendre où on va, si l'on ne sait pas d'où on vient"*



### 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
- ∞ Est donc décentralisé et ouvert à tous !
- ∞ Est un mouvement qui se base sur l'empirisme (l'acquisition de connaissances par l'expérience)

## 2- Qu'est ce que le DevOps?



## 2 - Qu'est ce que le DevOps?

Contexte avec un exemple avec une société fictive

Comment peut-on définir le 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.

## 2 - Qu'est ce que le DevOps?

Contexte avec un exemple avec une société fictive

Bob le développeur



Missions

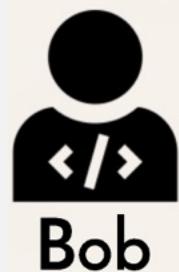


Problématiques

## 2 - Qu'est ce que le DevOps?

Contexte avec un exemple avec une société fictive

### Les missions du développeur



## 2 - Qu'est ce que le DevOps?

Contexte avec un exemple avec une société fictive

Problématiques rencontrées par Bob



Bob



Production

## 2 - Qu'est ce que le DevOps?

Contexte avec un exemple avec une société fictive

Alice l'opérationnelle



Missions



Problématiques

## 2 - Qu'est ce que le DevOps?

Contexte avec un exemple avec une société fictive

### Les missions de l'opérationnel



## 2 - Qu'est ce que le DevOps?

Contexte avec un exemple avec une société fictive

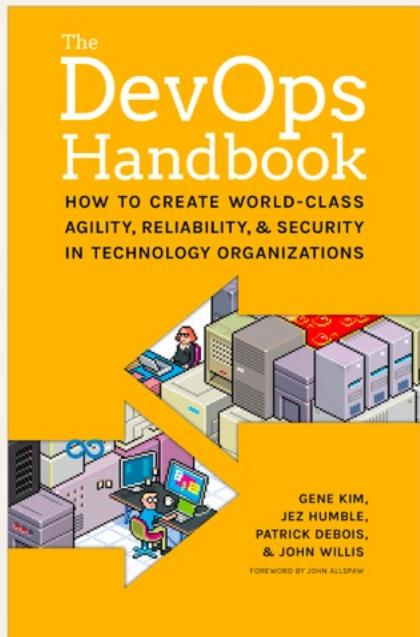
Problématiques rencontrées par Alice



## 2 - Qu'est ce que le DevOps?

Contexte avec un exemple avec une société fictive

### Le DevOps HandBook



**DevOps HandBook - How to create world-class agility, reliability & security in technology organizations**

Ouvrage de référence pour définir ce qu'est le DevOps

Je vous invite TRES fortement à le lire :

- ∞ Pas comme un roman du début à la fin
- ∞ Comprendre l'**esprit** dans lequel cet ouvrage a été réalisé
- ∞ En retirer ne serait-ce qu'un **mindset**, un état d'esprit

## 2 - Qu'est ce que le DevOps?

Contexte avec un exemple avec une société fictive

### Citations

Quelques citations du livre qui définissent le DevOps

"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"

## 2 - Qu'est ce que le DevOps?

Le **devops** — ou **DevOps** (selon la graphie habituellement utilisée en langue anglaise) — est un mouvement en ingénierie informatique et une pratique technique visant à l'unification du développement logiciel (**dev**) et de l'administration des infrastructures informatiques (**ops**), notamment l'administration système.

**DevOps** est une combinaison de concepts sociaux, de procédures et de ressources qui améliore la capacité d'une organisation à fournir des services et des applications à une vitesse maximale.

DevOps permet de franchir les obstacles érigés entre le développement et la production.

## 2 - Qu'est ce que le DevOps?

Ce que le DevOps n'est pas



## 2 - Qu'est ce que le DevOps?

**Le DevOps n'est pas un Titre**

Un ingénieur DevOps ne veut pas dire grand-chose.

Les Coachs ou les Evangélistes peuvent prétendre pouvoir accompagner les équipes et les entreprises dans leur organisation et leur transformation vers le DevOps.



## 2 - Qu'est ce que le DevOps?

### Le DevOps n'est pas une Equipe séparée

A l'origine de tout: la volonté de réunir les Devs et les 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.



## 2 - Qu'est ce que le 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.

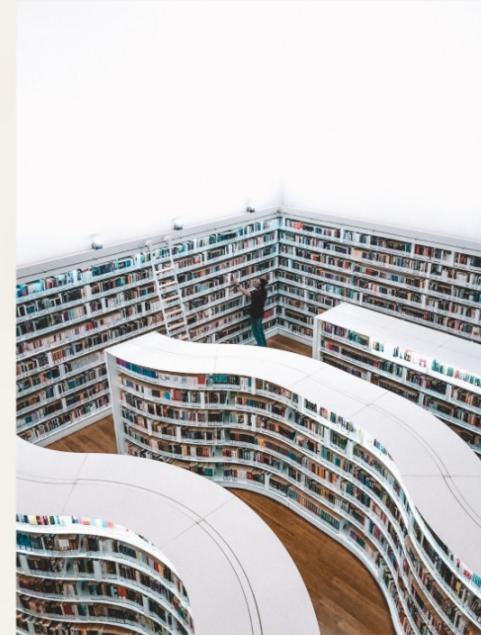
Cependant n'oubliez pas que ce ne sont pas les outils qui ont amené le DevOps, mais bien la méthodologie et la culture DevOps qui a provoqué l'émergence de nouveaux besoins et donc de nouveaux outils.



## 2 - Qu'est ce que le 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.



## 2 - Qu'est ce que le DevOps?

**Le DevOps n'est pas UNIQUEMENT  
de l'automatisation**

Ne serait-ce qu'en lisant le point précédent,  
vous aurez la capacité de saisir que  
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...



## 2 - Qu'est ce que le DevOps?

### Le DevOps n'est pas l'Anarchie

En réalité, c'est tout le contraire, et vous le verrez dans les parties suivantes, car si le DevOps était anarchique, comment pourrions-nous établir des plans prévisionnels pour définir de la valeur aux clients ? Comment des sociétés comme Netflix pourraient continuer de fonctionner ?



## 2 - Qu'est ce que le DevOps?

**Le DevOps n'est pas une Même Stratégie 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.



## 2 - Qu'est ce que le DevOps?

**Le DevOps est devenu important !**

### **Start-up**

Concurrence avec  
l'agilité et les  
déploiements rapides

### **Cloud**

Utilisation qui  
s'accélère chaque  
année d'avantage

### **Time to Value**

Accélération  
importante lors de la  
dernière décennie

*Comment peut-on continuer à fonctionner en silos, dans des structures lentes et complexes alors qu'il y a de plus en plus de données disponibles pour le Business ?*

## 2 - Qu'est ce que le DevOps?

Déetecter le besoin d'une démarche DevOps avec quelques signaux :

Vous détectez les défauts des logiciels à un stade avancé du cycle de vie ou, pire, en production.

1

Vos développeurs et testeurs attendent sans cesse de pouvoir accéder aux ressources dont ils ont besoin, ce qui entraîne des retards.

2

Vous utilisez Agile pour accélérer le développement, mais tous les bénéfices s'évaporent lorsque l'application passe en production.

3

Vous constatez que de simples erreurs humaines causent des ravages lors du développement et du déploiement.

4

Vous ne parvenez pas à identifier les problèmes au cours du développement, des tests et de la production.

5

Le développement considère que son travail est terminé lorsque l'application passe en production.

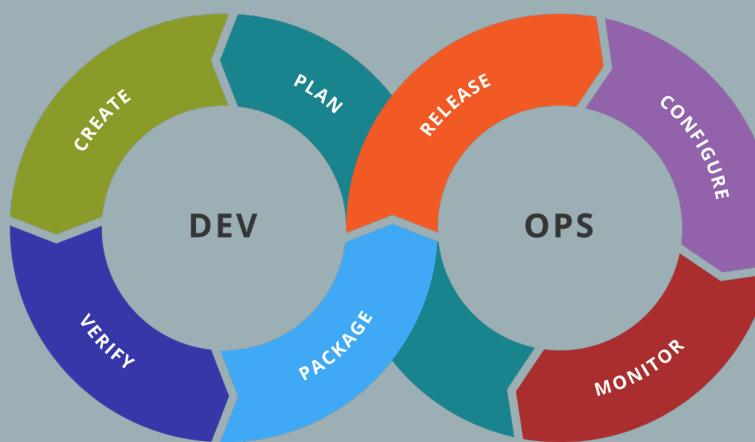
6



## UN PETIT QUIZ



### 3- Les avantages de la démarche DevOps.



# Les avantages de la démarche DevOps.

**2 avantages principaux et 6 avantages subjacents**

L'accélération des  
déploiements  
appliquatifs

La réduction du  
Time-to-Market.

# Les avantages de la démarche DevOps.

## Livraison rapide

DevOps Augmente le rythme et la fréquence des sorties afin que les entreprises puissent innover et améliorer leurs produits plus rapidement.

## Vitesse et taux

Les entreprises peuvent évoluer rapidement afin de permettre une innovation plus rapide pour les consommateurs, de s'adapter aux changements des marchés et de devenir plus efficaces pour obtenir des résultats commerciaux.

## Échelle

Gérer et exploiter les processus de développement et l'infrastructure à l'échelle. L'automatisation et la régularité permettent aux organisations de contrôler des systèmes sophistiqués ou changeants efficacement et avec moins de risques.

# Les avantages de la démarche DevOps.

## Sécurité

Les entreprises peuvent adopter un modèle DevOps sans avoir à sacrifier la sécurité en utilisant des politiques de conformité, des contrôles et des méthodes de gestion de configuration.

## Fiabilité

S'assurer de la qualité des mises à jour des applications logicielles et des ajustements de l'infrastructure afin que les entreprises puissent livrer plus rapidement et efficacement tout en conservant une expérience positive pour les utilisateurs du produit ou service.

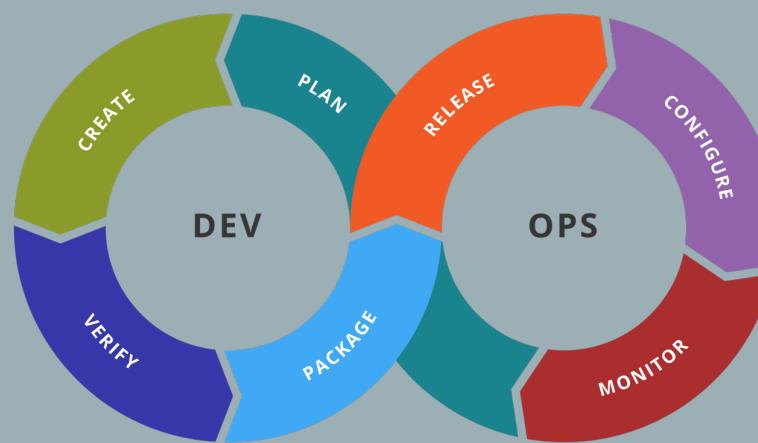
## Coopération accrue

Développer des équipes plus efficaces dans le cadre d'un modèle social DevOps qui met l'accent sur des valeurs telles que l'appropriation et la responsabilité.

## Les avantages de la démarche DevOps.

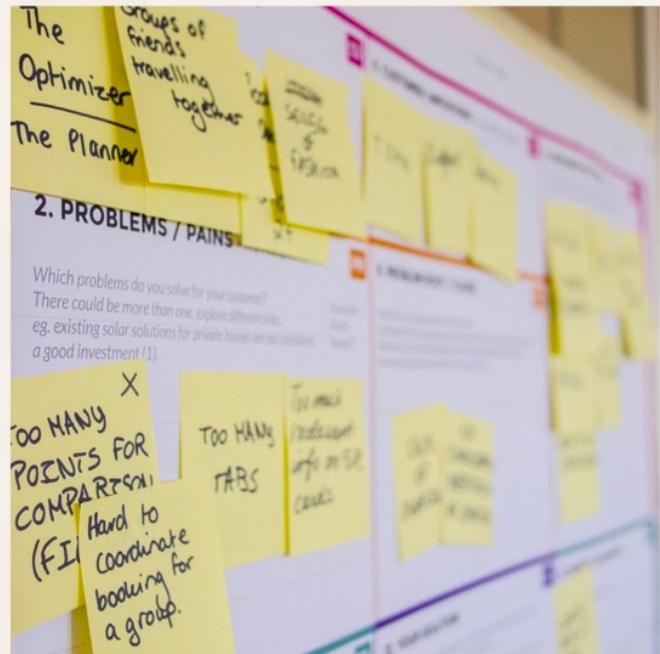
- DevOps s'est développé à partir d'une stratégie visant à amener les développeurs et les équipes d'exploitation à une méthode permettant de transformer une organisation entière en une seule entité opérationnelle.
- En facilitant :
  - La connexion.
  - Les relations.
  - L'intégration dans l'ensemble de l'organisation.

## 4- Les caractéristiques du DevOps.



## 4- Les caractéristiques du DevOPs.

Quid des autres méthodes ?



Quelles sont les autres méthodes  
qui sont capables de s'articuler  
avec le DevOps ?



Gestion de  
Projets



Application  
au spectre IT

## 4- Les caractéristiques du DevOps.



### Gestion de Projets

1  
SCRUM

2  
Lean

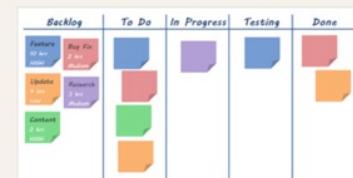
3  
Kanban



Cérémonies et post-its



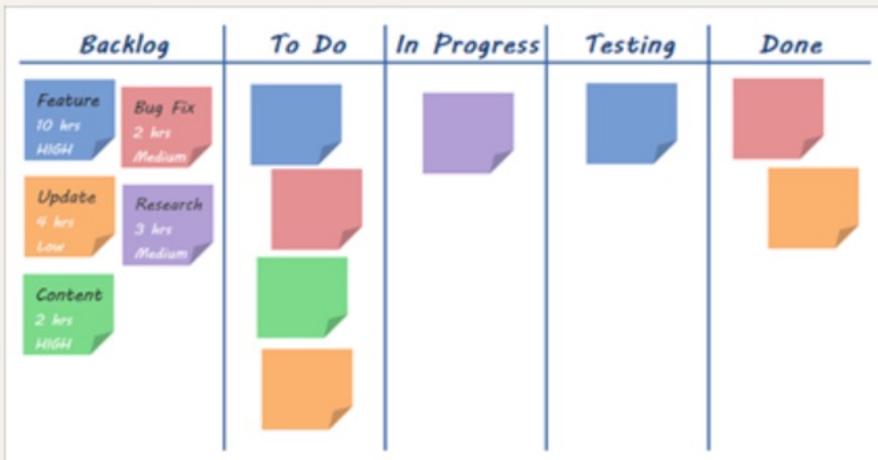
Les principes de l'industrie appliqués à l'IT



Focus sur le Kanban

## 4- Les caractéristiques du DevOps.

### Focus sur le Kanban



- ∞ Visualisation de organisation du flux de travail (**workflow**)
- ∞ Limite le **Work in Progress** (WIP) à la capacité de l'équipe
- ∞ Pousse les individus à travailler de manière **collaborative**
- ∞ Mesure de la **vélocité** de l'équipe

## 4- Les caractéristiques du DevOps.

**Le DevOps s'applique à tout le spectre IT**



Personnes



Processus



Automation

## 4- Les caractéristiques du DevOPs.

Piliers  
C.A.L.M.S

On peut identifier 5 piliers principaux :

1. La **Culture**.

Focus sur les gens, les changements et l'expérimentation

2. L'**Automatisation**.

Déploiement Continu et Infrastructure as Code (IaC)

3. **Lean**.

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

4. **Mesure**.

Tout mesurer ! Montrer l'amélioration et la progression

5. **Share**.

Partage ouvert de toutes les informations => Collaboration et Communication

## 4- Les caractéristiques du DevOps.

### La culture

Le DevOps résout, en premier lieu, des problèmes humains, des problèmes de communication et des problèmes de responsabilité entre équipes.

- *Inclusion de toutes les équipes à toutes les étapes du projet (dev, ops, chef de projet, PO, testeur, etc.).*
- *Partage des objectifs, des moyens, des problèmes et des solutions.*
- *Rituels formels (daily, réunion, etc) et informels (sorties, etc.).*
- *Pollinisation croisée : partage d'expériences, solutions, informations entre équipes.*

Equipe par compétence



Equipe orientée produit

**Objectif:** Effectuer la mission de son pôle de compétence

**Objectif:** Délivrer la meilleure application, produits dans le meilleur délai et la meilleure qualité

## 4- Les caractéristiques du DevOps.

### Automatisation

La culture DevOps est importante mais elle n'est pas suffisante.

Des outils sont nécessaire et indispensable afin de fluidifier et faciliter les relations de travail, notamment au travers des différentes phases de déploiements => Automatisation.

Déployez plus souvent.

Configurez vos déploiements

Créez des environnements à la demande

Testez plus fréquemment

## 4- Les caractéristiques du DevOps.

### Lean

Le Lean, ou **Lean Management**, vient de l'industrie, et plus particulièrement de Toyota dans **les années 1990**. Lean sert à qualifier une **gestion des ressources sans gaspillage**.

Dans le contexte du **DevOps**, Lean va alors s'intéresser à délivrer de la valeur ajoutée au client final (dans le cadre d'une application grand public, le public), tout en minimisant les processus longs, coûteux, sans valeur ajoutée.



### Les erreurs sont inévitables !

Il est alors important dans le DevOps de ne pas reprocher à quelqu'un le problème de déploiement, mais plutôt **d'encourager l'équipe à parler librement lorsqu'une erreur arrive.**

Afin d'identifier celle-ci, ses causes et mettre en place un processus garantissant la non-reproduction de cette erreur.

## 4- Les caractéristiques du DevOps.

### Mesure

Comme toute transformation d'entreprise, il est nécessaire d'avoir des **indicateurs de performance clés** (KPI ou Key Performance Indicator) afin de savoir si les efforts de transformation et d'amélioration continue transforment quelque chose.

Nombre bug récurrent

Combien de personne utilisent le produit en temps réels

Délai de la nouvelle fonctionnalité pour passer en production

Combien d'utilisateurs gagnés ou gagnés sur une semaine?

## 4- Les caractéristiques du DevOps.

### Share (Partage)

Favoriser la communication et le partage mutuelle sur les objectifs, les missions, les réussites, les échecs, les problèmes, les bonnes pratiques, le rôle de chacun, etc..

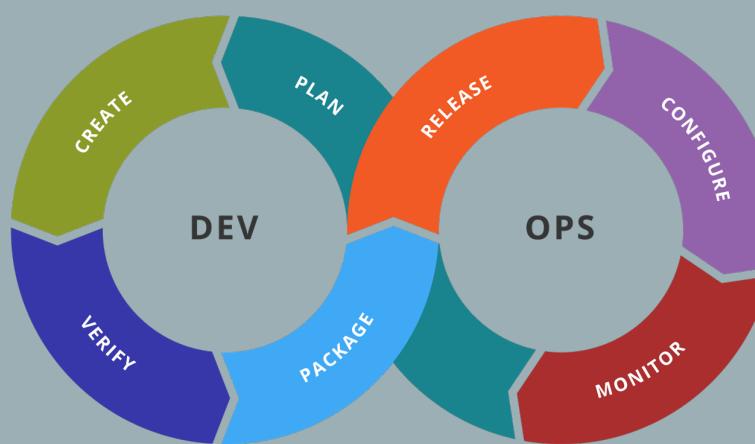
**L'objectif est** les opérations et les développeurs doivent travailler de concert sur le **cycle de vie complet** de l'application.

**Ex :**

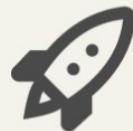
*Les équipes mettant en place le DevOps ont souvent un rôle tournant : les développeurs corrigent les problèmes des utilisateurs finaux, mais aussi participent à la **Résolution des problèmes de production.***



## 5- Les objectifs de la démarche DevOPs.



## 5- Les objectifs de la démarche 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**

## 5- Les objectifs de la démarche DevOPs.

1

Amélioration du Time To Value

2

Intégration des pratiques avec le Business

3

Réactivité accrue des équipes

4

Code et déploiement de meilleures qualités

5

Productivité plus importante

6

Visibilité des actions menées par les équipes

7

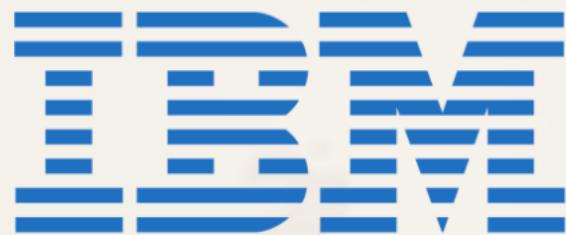
Visibilité des actions menées par les équipes

## 5- Les objectifs de la démarche DevOPs.

Quelques résultats significatifs avec :

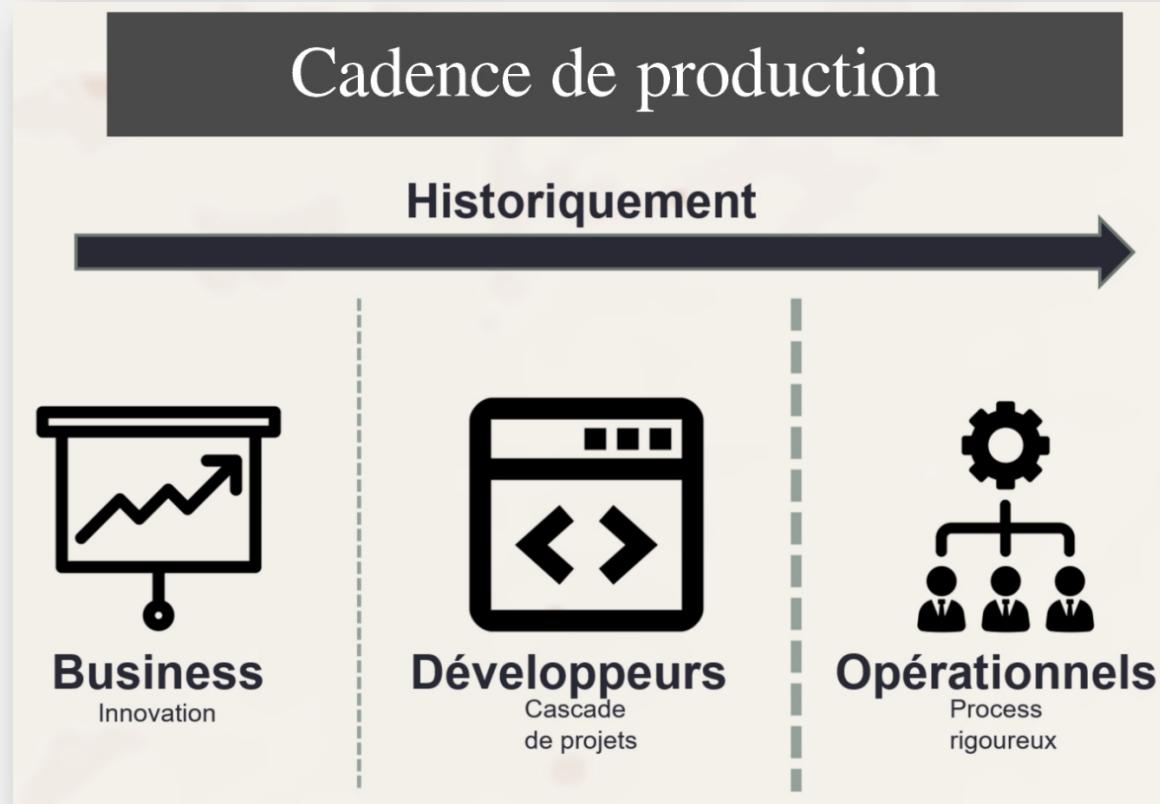
Cadence de production

Quelques statistiques



## 5- Les objectifs de la démarche DevOPs.

Quelques résultats significatifs :



## 5- Les objectifs de la démarche DevOPs.

Quelques résultats significatifs :

### 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

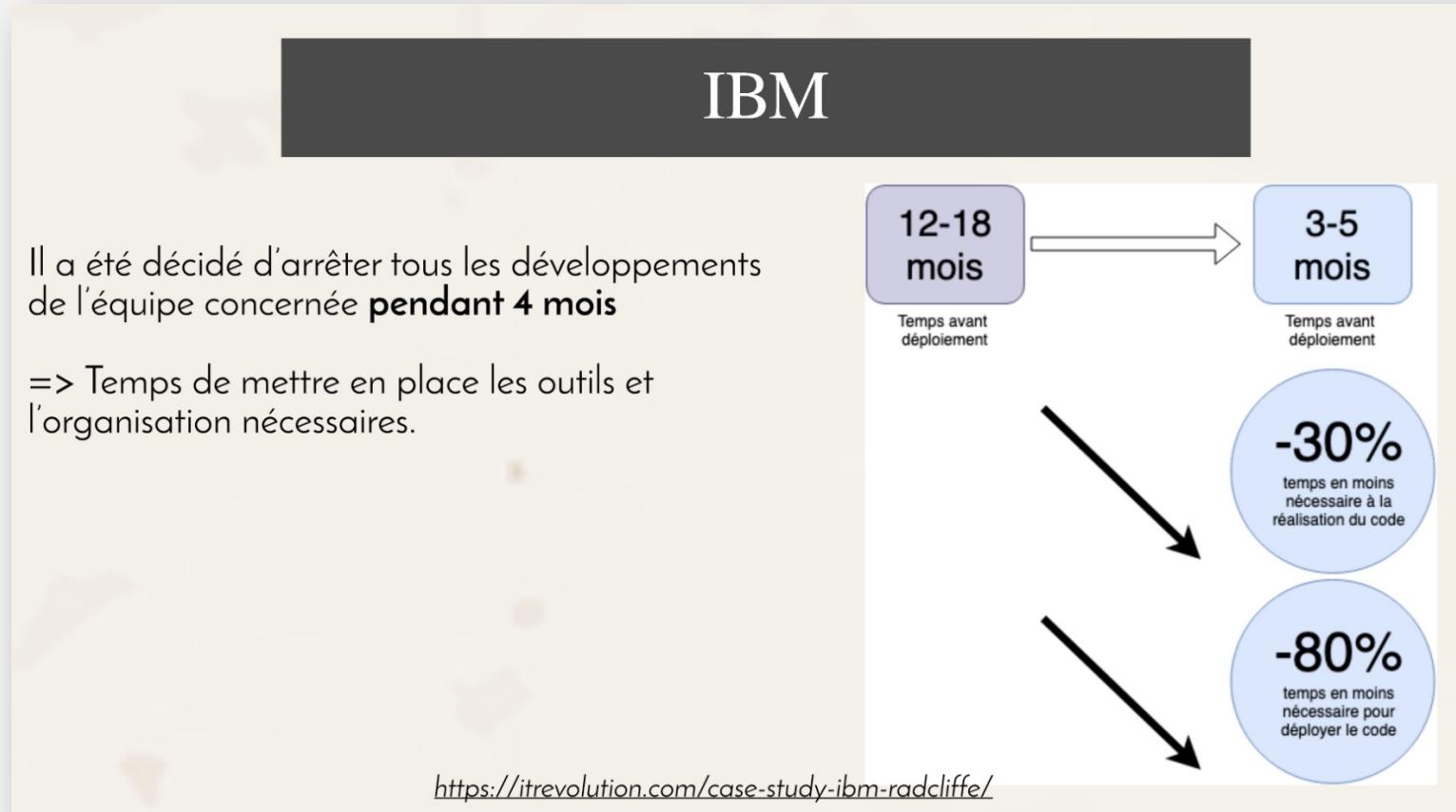
106 fois plus rapide du commit jusqu'au déploiement

2604 fois plus rapide pour un retour après incident

7 fois moins d'incidents en cas de changements

## 5- Les objectifs de la démarche DevOPs.

Quelques résultats significatifs :



## 5- Les objectifs de la démarche DevOPs.

Une double vision du DevOps :

Business



Technique



## 5- Les objectifs de la démarche DevOPs.

Une double vision du DevOps (Business – Technique ) :

The diagram consists of a large grey trapezoid representing the 'Golden Circle'. Inside the trapezoid, there is a smaller white trapezoid containing a dark grey tag-like shape with the text 'Golden Circle' on it. To the right of the trapezoid is a blue-bordered image of a person's hand holding a newspaper with the word 'Business' on the front page. Above this image, the word 'Business' is written in a bold, dark blue font. Below the main trapezoid, four statements are listed in white text:

- Tous les business tendent à devenir des business tech
- Les consommateurs ont développé des mentalités "apps"
- Les consommateurs préfèrent la valeur au produit lui-même
- Le Time to Value remplace petit à petit le Time to Market

**Business**

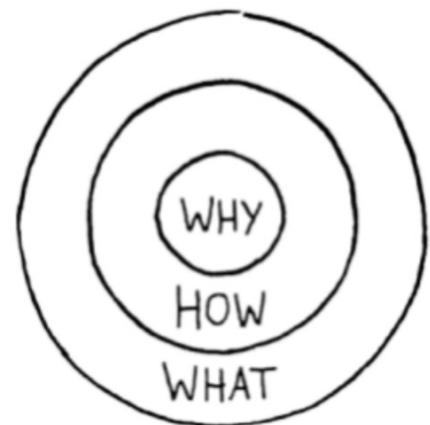
## 5- Les objectifs de la démarche DevOPs.

Une double vision du DevOps (Business – Technique ) :

Start with the "Why" - Le Golden Circle

**Business**

The Golden Circle



**WHY**

Ne fait pas référence à l'argent, il s'agit d'un résultat, un but, ou une croyance.  
La raison pour laquelle la société existe !

**HOW**

Il s'agit des choses qui rendent spéciales ou qui distinguent de la concurrence.

**WHAT**

Il s'agit des produits vendus ou des services offerts

Auteur: **Simon Sinek**



## 5- Les objectifs de la démarche DevOPs.

Une double vision du DevOps (Business – Technique ) :

**Business**

### Respect du Golden Circle

"Dans tout ce que nous faisons, nous croyons dans le fait de challenger le status quo actuel, nous croyons penser différemment. Nous challengeons le status quo en fabriquant des produits magnifiques, simples à utiliser, et très agréables. Nous arrivons donc à fabriquer de supers ordinateurs, voulez-vous en acheter un ?"

Les gens n'achètent pas ce que vous vendez, mais achètent POURQUOI vous le vendez !



**Bon message**

### Mauvais message

"Nous fabriquons de supers ordinateurs, ils sont beaux, faciles et agréables à utiliser. Est-ce que vous voulez en acheter un ?"

On commence par le WHAT, puis le How sans même parler du WHY...



**Mauvais message**

## 5- Les objectifs de la démarche DevOPs.

Une double vision du DevOps (Business – Technique ) :

Technique

Challenges  
de l'IT

Les méthodes agiles ont joué un rôle déterminant dans l'amélioration du développement et la confiance des entreprises dans les services qu'elles proposent

Elles ont laissé involontairement les Opérations à la traîne

DevOps est donc un bon moyen pour l'entreprise de regagner confiance dans TOUTE son organisation



## 5- Les objectifs de la démarche DevOPs.

Une double vision du DevOps (Business – Technique ) :

Technique

### Les challenges de l'IT

**Qualité** L'IT doit toujours aller de plus en plus vite sans augmenter les risques de diminution de la qualité.

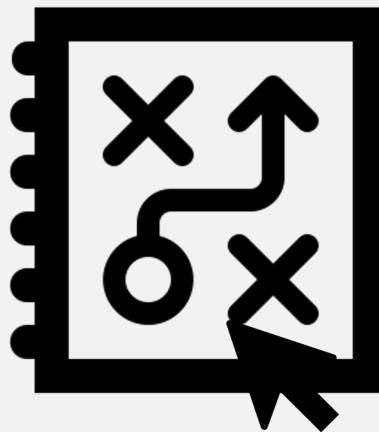
**Investissements** Les investissements précédents (Agile, ITSM, Automation) ne suffisent pas à eux seuls à apporter de la valeur sur l'ensemble du spectre de l'IT.

**Culture** La culture en **silos** limite la capacité à pouvoir fournir de la valeur.

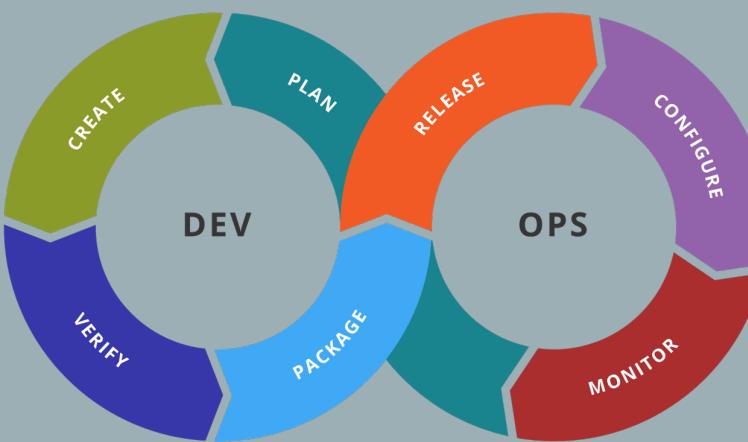
L'IT n'a plus besoin de s'aligner ou de s'intégrer avec le business, car **l'IT est devenue le business**.



## UN PETIT QUIZ



## 6- Les principes fondamentaux du DevOps— Three Ways



## 6- Les principes fondamentaux du DevOps– Three Ways

- Le **Three Ways** de Gene Kim correspond à un certain nombre de principes à partir desquels, tous les modèles DevOps pourront être dérivés.
- **La théorie des contraintes.**
- **L'ingénierie du Chaos.**

## 6- Les principes fondamentaux du DevOps– Three Ways

### **Le First Way et le notion de flux**

**Le "First Way"**

Qu'est-ce que c'est ?

**Théorie des  
contraintes**

M. Eliyahu Goldratt

## 6- Les principes fondamentaux du DevOps– Three Ways

### 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.



Consiste à **comprendre et améliorer le flux de travail**, de la gauche vers la droite sur le schéma ci-dessus.

## 6- Les principes fondamentaux du DevOps– Three Ways

### Implications du First Way



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



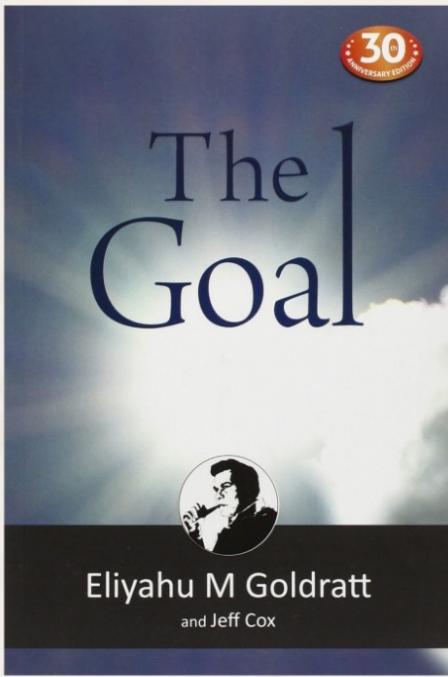
Interdiction des optimisations locales qui pourraient causer une dégradation globale

« 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 »

## 6- Les principes fondamentaux du DevOps– Three Ways

### La théorie des contraintes



M. **Eliyahu M. Goldratt** a introduit cette théorie dans le livre "**The Goal**" en 1994

La TOC (Theorie of Constraints) est une méthodologie permettant d'identifier le **facteur limitant le plus important** (la contrainte)

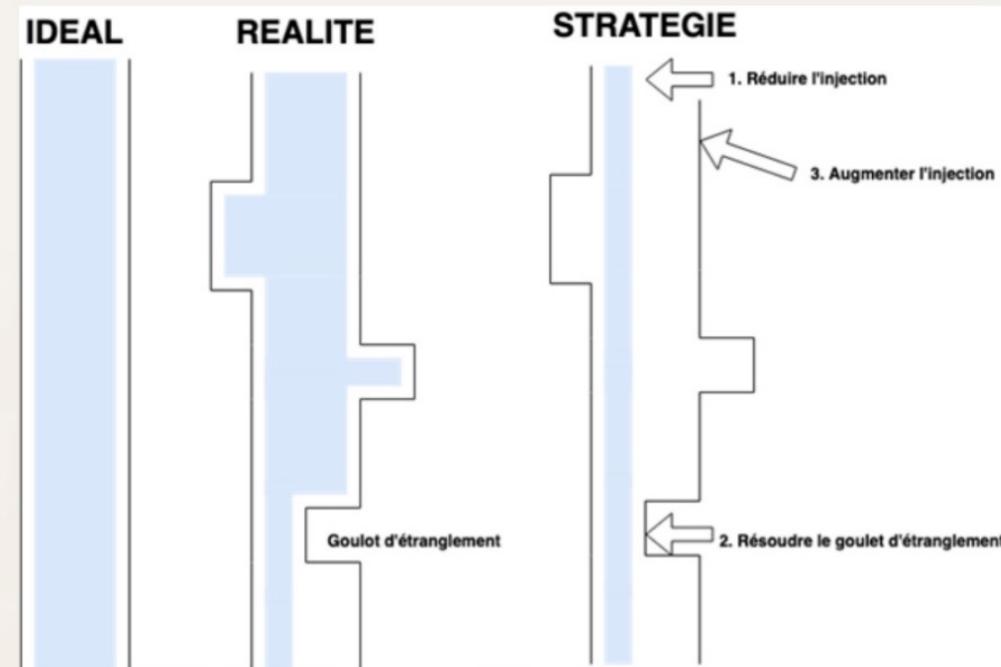
=> Améliorer systématiquement la contrainte jusqu'à ce qu'elle ne soit plus le facteur limitant

## 6- Les principes fondamentaux du DevOps– Three Ways

### Explication avec un schéma :

Tous les processus ont un "**goulot d'étranglement**" qui affecte la possibilité d'atteindre son but de manière constante

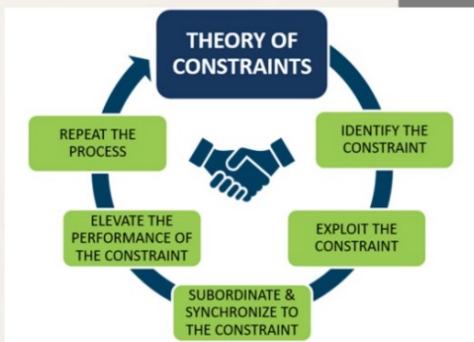
Le processus va devenir dépendant de sa contrainte et ne pourra jamais être **plus performant que son maillon le plus faible.**



## 6- Les principes fondamentaux du DevOps– Three Ways

### Les étapes de la TOC

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



#### Exemples de contraintes

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

## 6- Les principes fondamentaux du DevOps– Three Ways

### Le "Second Way"

#### Le "Second Way"

L'importance du feedback

#### Exemples

Mieux comprendre le "Second Way"

## 6- Les principes fondamentaux du DevOps– Three Ways

### 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.

## 6- Les principes fondamentaux du DevOps– Three Ways

### Exemples de boucles de feedback

#### Tests automatiques

Effectués lors du push du code vers le dépôt Gitlab ou GitHub.

Vérifie que le code fonctionne correctement, qu'il ne présente pas de failles de sécurité, qu'il est compatible avec l'environnement de production, etc...

#### Peer Review

Demander à un collègue de vérifier les changements effectués sur le code et qu'il les valide avant d'utiliser ce code en production

#### Supervision

Utiliser Centreon ou Nagios pour monitorer les éléments en production et en qualification.

Kibana permet de mettre en place de véritables tableaux de bords pour visualiser les statistiques et les logs

## 6- Les principes fondamentaux du DevOps– Three Ways

### Qu'est-ce que le Third Way ?

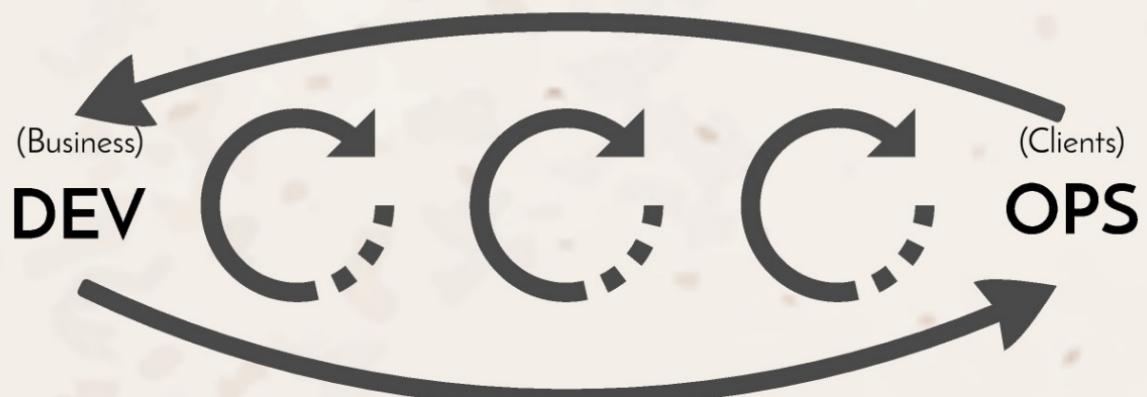
Le « Third Way » souligne l'idée de créer une culture qui favorise **l'expérimentation continue** incluant de prendre des risques et d'apprendre de ses échecs



## 6- Les principes fondamentaux du DevOps– Three Ways

### 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



## 6- Les principes fondamentaux du DevOps– Three Ways

### Les éléments clés du "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)



## 6- Les principes fondamentaux du DevOps– Three Ways

### Ingénierie 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  
de Netflix



Simian Army



Et le  
DevOps ?

## 6- Les principes fondamentaux du DevOps– Three Ways



### Chaos Monkey de Netflix

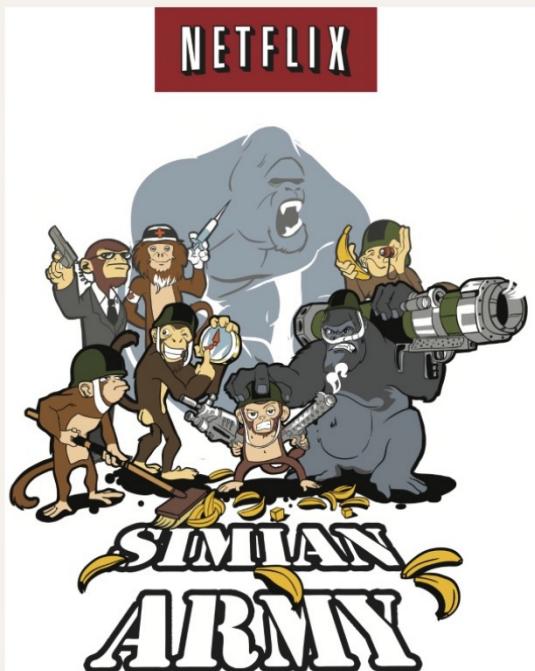
En 2011, Netflix choisit au hasard des instances AWS en production et les met volontairement **hors service**.

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



## 6- Les principes fondamentaux du DevOps– Three Ways

### 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

<https://github.com/Netflix/SimianArmy>

## 6- Les principes fondamentaux du DevOps– Three Ways

### Le DevOps

Les **Chaos Monkey** contribuent directement à la Toolchain DevOps en répondant au besoin de **Continuous Testing**



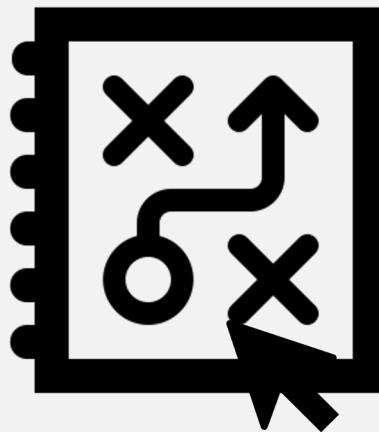
Inscription dans le modèle **Design For Failure** (conçu pour supporter la défaillance).

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

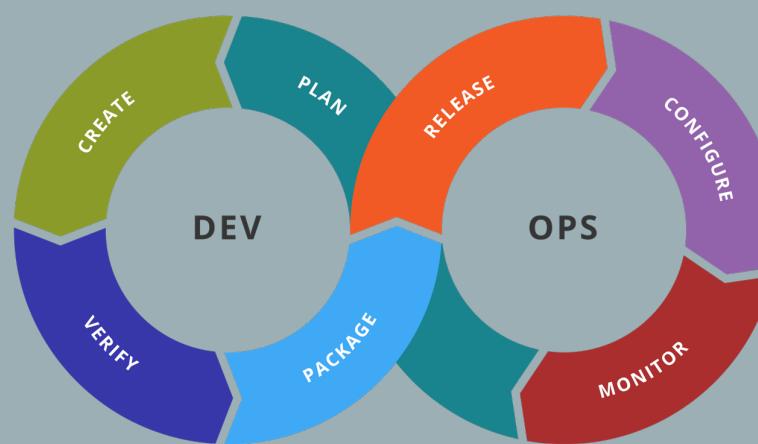




## UN PETIT QUIZ



## 7- Les pratiques techniques du DevOps



## 7 - Les pratiques techniques du DevOps

1. Comment les tests automatiques sont-ils mis en œuvre dans les équipes DevOps?
2. Qu'est ce que l'intégration continue (Continuous Intégration : CI)?
3. Qu'est ce que la livraison continue ( Continuous Delivery : CD )?
4. Qu'est ce que le SRE (Site Reliability Engineering)?
5. Le DevSecOps et le ChatOps, d'autres métiers du DevOps.

## 7 - I La mise en œuvre des tests automatiques

### Le Continuous Testing

#### Le "Shift Left"

Intégration de la qualité dans le processus de développement

#### Les Tests Fonctionnels

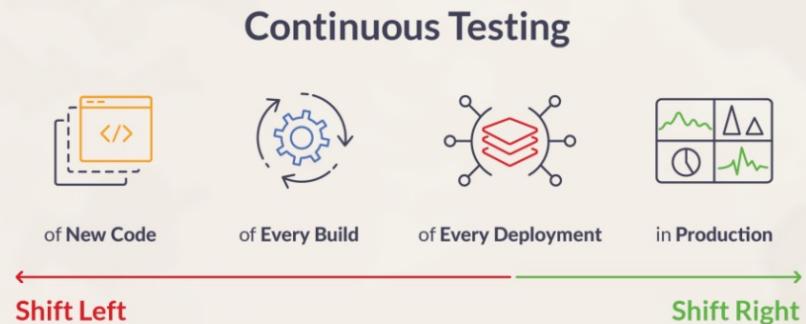
Qu'est-ce que c'est ?

#### Les Tests Non Fonctionnels

Qu'est-ce que c'est ?

## 7 - I La mise en œuvre des tests automatiques

### Qu'est-ce que le "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 **risques business** associés à la nouvelle version du code qui sera déployé en production

## 7 - I La mise en œuvre des tests automatiques

### Les Tests Fonctionnels

Parmi les tests fonctionnels on retrouve les plus importants :

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

Ce sont les tests pour lesquels jusqu'à récemment, un **action manuelle** était nécessaire.



## 7 - I La mise en œuvre des tests automatiques

### Les Tests non-fonctionnels



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

- Tests de Performance
- Tests de Sécurité
- Tests Capacitaires
- Tests de Compatibilité

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

## 7 – 2 L'intégration continue

### Intégration Continue

The diagram illustrates the Continuous Integration cycle as a circular process. It starts with a blue arrow pointing right labeled "Developer" containing a person icon. This leads to a blue circle labeled "Source control" containing a server icon. From there, it moves clockwise through three main phases: "Test" (represented by a clipboard icon), "Build" (represented by a wrench icon), and "Release" (represented by a cloud with an upward arrow icon). The cycle is labeled "Continuous integration" at its center. Around the top of the circle, the words "Test", "Build", and "Release" are placed above their respective icons. On the left side of the circle, there are two small icons: a checkmark and an X inside circles, labeled "Report".

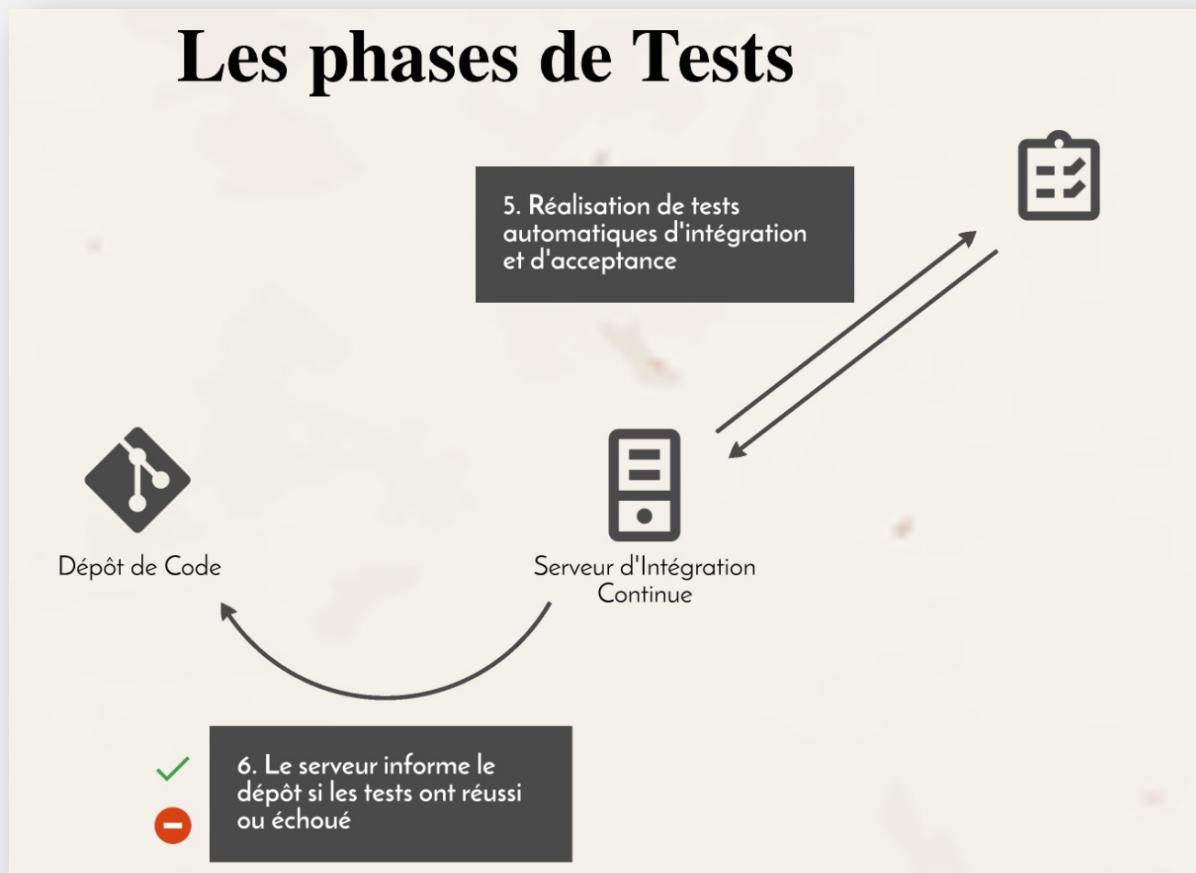
L'intégration Continue (CI) correspond à des pratiques de développement, impliquant le **commit du code** dans un **dépôt partagé** par les développeurs

La phase de Build

Les phases de Tests

Prérequis pour les tests

## 7 – 2 L'intégration continue



## 7 – 2 L'intégration continue

### Que faut-il en prérequis ?

- Définir le **standard de code** qu'on souhaite utiliser dans l'organisation (PEP8, FLAKE8, Pylint, etc...)
- Implémenter un **dépôt** de code permettant du versionning (Github, ou 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
- Permettre la **détection** et la **remédiation rapide** des erreurs avant de pousser le code en production



## 7 – 3 La Continuous Delevry

### Continuous Delivery



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

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



Bénéfices



Differences  
avec le CD



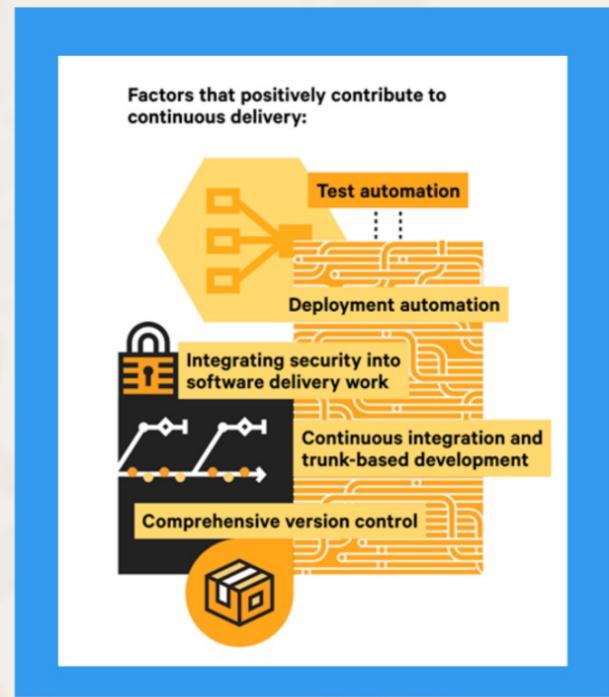
Exemple

## 7 – 3 La Continuous Delevry



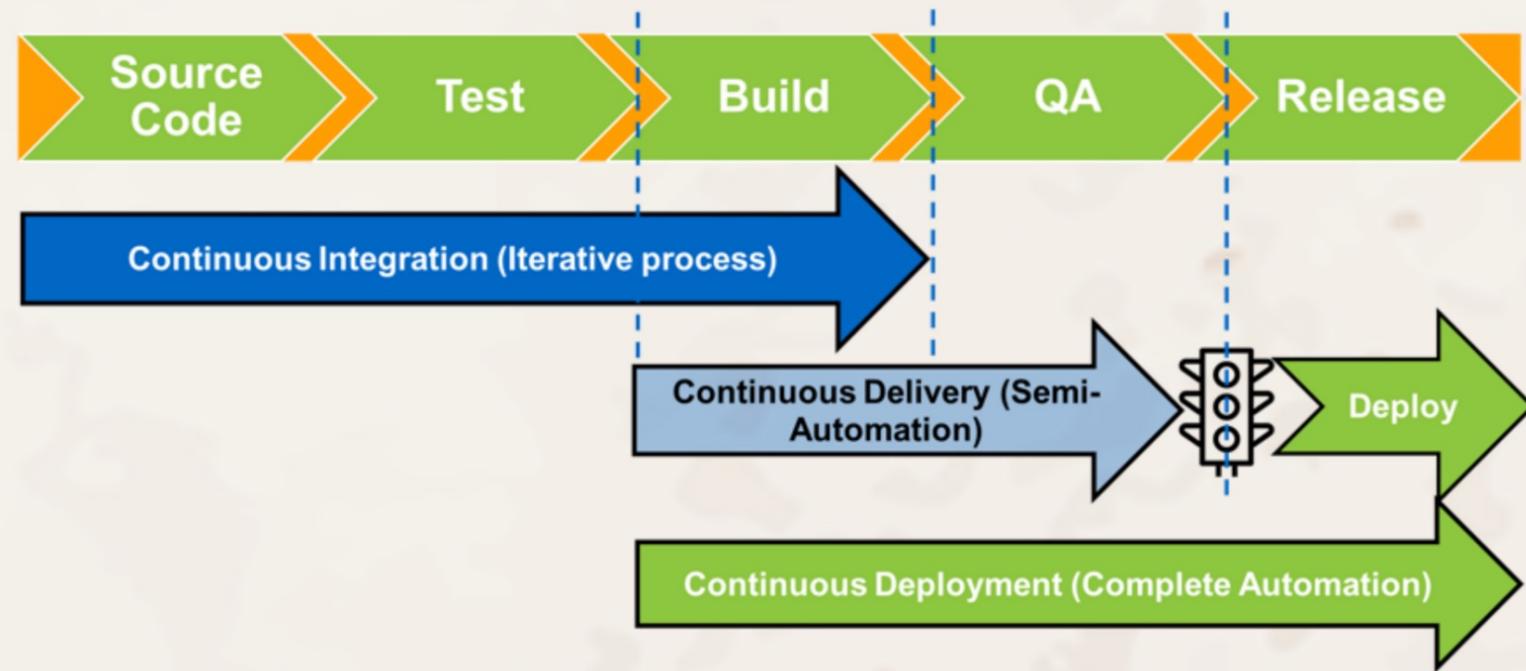
### Les bénéfices du Continuous Delivery

- Fournir rapidement des **feedback automatiques** sur le système de préproduction et de production
- Garde une version logicielle **releasable** et **déployable** même lorsque de nouvelles fonctionnalités sont en cours de développement.
- Mise en place d'une pipeline qui permet de faire du **déploiement à la demande**.
- Réduit les **coûts**, le **temps** et les **risques** inhérents aux changements incrémentaux réalisés



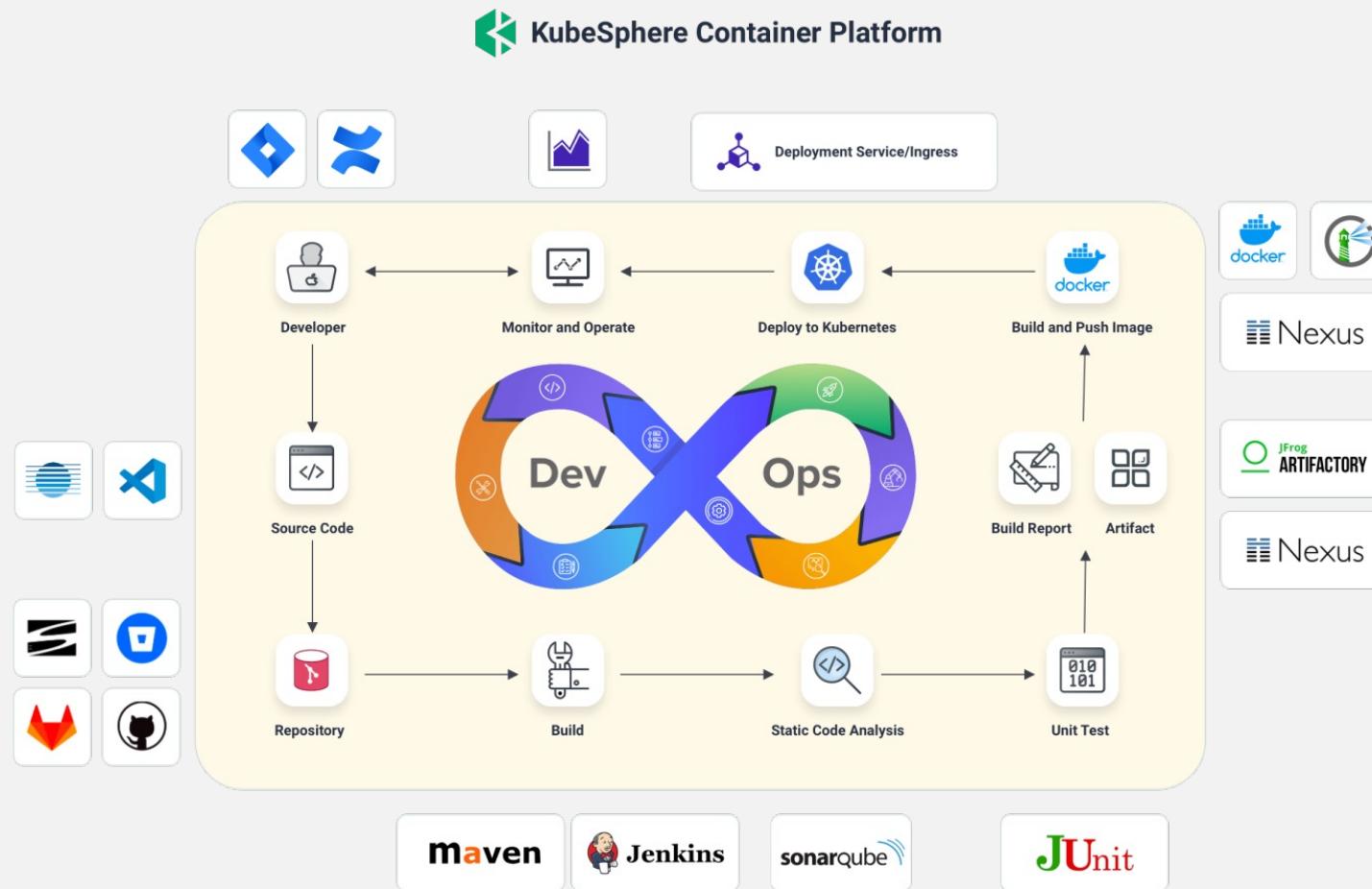
## 7 – 3 La Continuous Delevry

### Différences avec le Déploiement Continu



## 7 – 3 La Continuous Delevry

Vision  
technique et  
avec outils  
du CI/CD



## 6 – 3 La Continuous Delevry



## 7 – 4 Qu'est ce que le SRE?

Les ingénieurs SRE (Site Reliability  
Engineering)

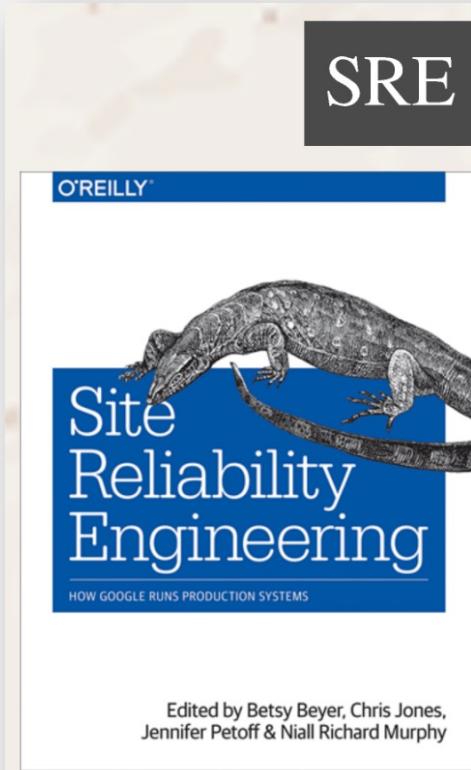
**Le SRE**

Site Reliability  
Engineering

**La "Résilience  
Engineering"**

Qu'est-ce que c'est ?

## 7 – 4 Qu'est ce que le SRE?



### SRE (Site Reliability Engineering)

**O'REILLY®**

**Apparu chez Google**

Lorsqu'on a commencé à donner des tâches de type "Opérationnelles" à des ingénieurs software

**Systèmes évolutif et fiables**

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

**1500 SRE chez Google aujourd'hui**

**50% du temps**

Tâches liées aux "Opérations" comme les incidents, sur appel ou via des interventions manuelles

**50% du temps**

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

## 7 – 4 Qu'est ce que le SRE?

### La "Resilience Engineering"

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



Erik Hollnagel

## 7 – 4 Qu'est ce que le SRE?

### 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 l'**humain**

#### Revues post-incident

Effectuer des revues post-incident **aggressives, irréprochables, et systémiques**

#### Weakest Link

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

## 7 – 5 Les autres métiers du DevOps

Les autres métiers importants



DevSecOps



ChatOps

## 7 – 5 Les autres métiers du DevOps

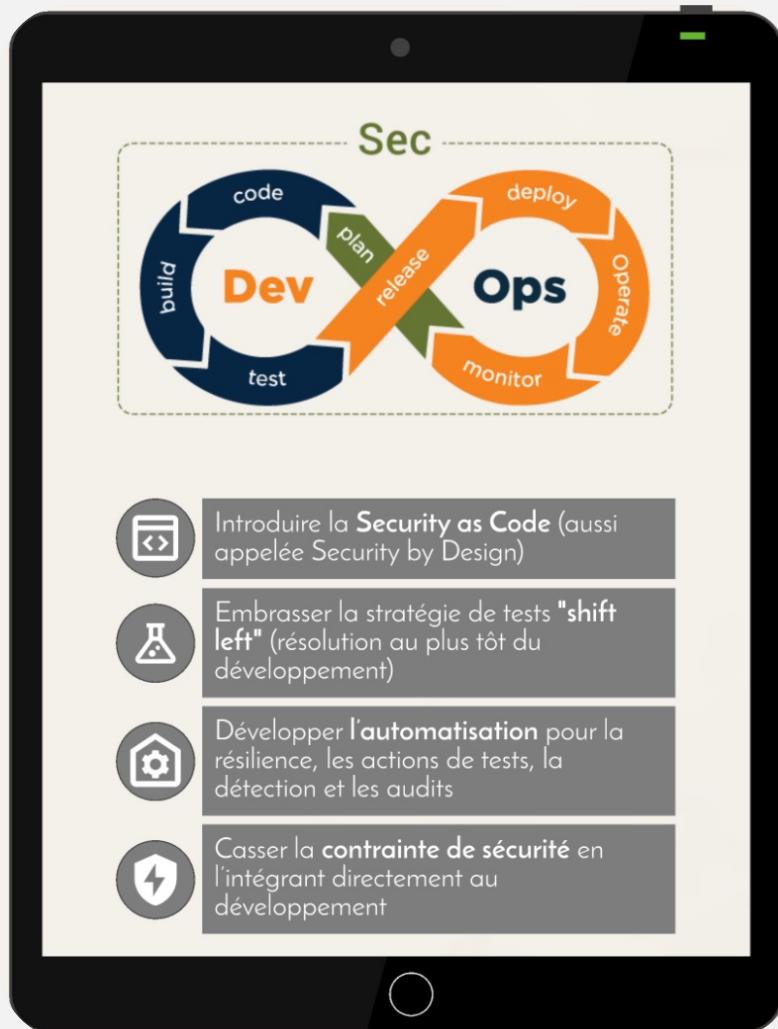
### Qu'est-ce qu'un DevSecOps ?

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

L'idée étant de prendre des décisions côté sécurité, de manière sûre, rapide et évolutive pour conserver le haut niveau de déploiement nécessaire.



## 7 – 5 Les autres métiers du DevOps



## 7 – 5 Les autres métiers du DevOps

"Permet d'afficher des messages automatiques à chaque fois qu'un développement ou qu'un déploiement est effectué ou lorsque du support est nécessaire !"

### 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)



## 7 – 5 Les autres métiers du DevOps

### Un exemple de bot



arma 8:58 PM  
!status all

stanley BOT 8:58 PM  
I'm on it! Your execution ID is 5772bad30a84b440c5e7807f (details available at <https://chatops/#/history/57>)  
@arma: Here is your status for `all` host(s):

```
localhost | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}  
db | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}  
web | UNREACHABLE! => {  
    "changed": false,  
    "msg": "Failed to connect to the host via ssh.",  
    "unreachable": true  
}
```

| Alive | Dead |
|-------|------|
| 2     | 1    |

5772bad30a84b440c5e7807f

+

## 7 – 5 Les autres métiers du DevOps

### Un autre exemple de bot



**rickmak** 11:41 AM

chima deploy trelhub/edge to pandawork



**chima** BOT 11:41 AM

rickmak: Deployment of trelhub/edge to pandawork created



**faseng** BOT 11:41 AM

#1407816 - trelhub / edge / pandawork

| #1407816 : rickmak is deploying [trelhub](#) to pandawork ([compare](#))



**rickmak** 11:41 AM

Hi faseng



**faseng** BOT 11:42 AM

| #1407816 : rickmak's pandawork deployment of [trelhub](#) is done!

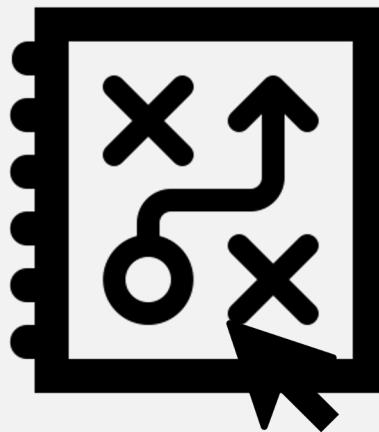
## 7 – 5 Les autres métiers du DevOps

### Les différents outils de communication utilisable

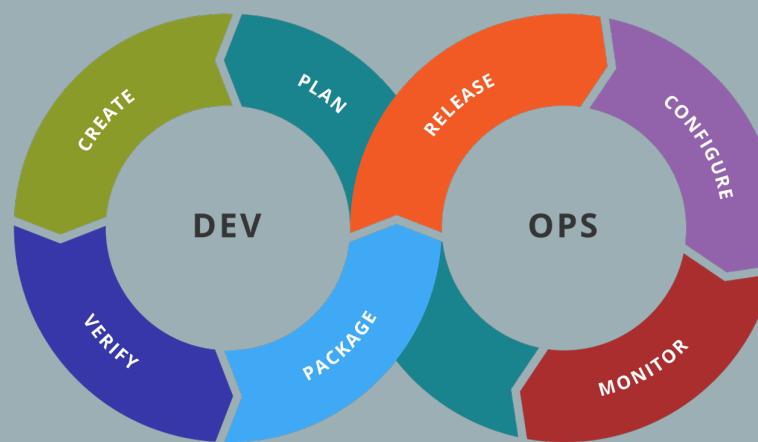




## UN PETIT QUIZ



## 8 - Les Frameworks Business et Techniques utilisés en DevOps



## 8 - Les pratiques techniques du DevOps

1. Le manifeste agile.
2. Les méthodes agiles SCRUM et SAFe.
3. L'IT Service Management.
4. La méthode ITIL (IT Infrastructure Library)
5. L'Agile SM – Un peu ITSM dans de l'Agilité.
6. La méthode Lean – Efficacité et Gaspillage.
7. La méthode Lean – Cartographie des flux de valeurs.
8. La méthode Lean – L'improvement Kata.
9. Learning Organizations – The Fifth Discipline

## 8 - Les pratiques techniques du DevOps

### Les méthodes agiles

#### Le manifeste Agile

Les règles de base de toutes les méthodes agiles

#### La méthode SCRUM

Qu'est-ce que c'est ?

#### La méthode SAFe

Qu'est-ce que c'est ?

## 8 – I Le manifeste Agile

### Le manifeste agile

Mieux développer des logiciels par la pratique et en aidant les autres à le faire !

#### Projet agile

Individus et leurs interactions



Logiciels opérationnels



Collaboration avec les clients



Adaptation au changement



#### Projet classique

Processus et outils

Documentation exhaustive

Négociation contractuelle

Suivi d'un plan



## 8 – 2 La méthode SCRUM et SAFe

### La méthode SCRUM



## 8 – 2 La méthode SCRUM et SAFe

### La méthode SAFe

SAFe (**Scaled Agile Framework**) correspond à un framework permettant d'appliquer les principes et la pratique de l'agilité et du Lean au service du développement d'une entreprise

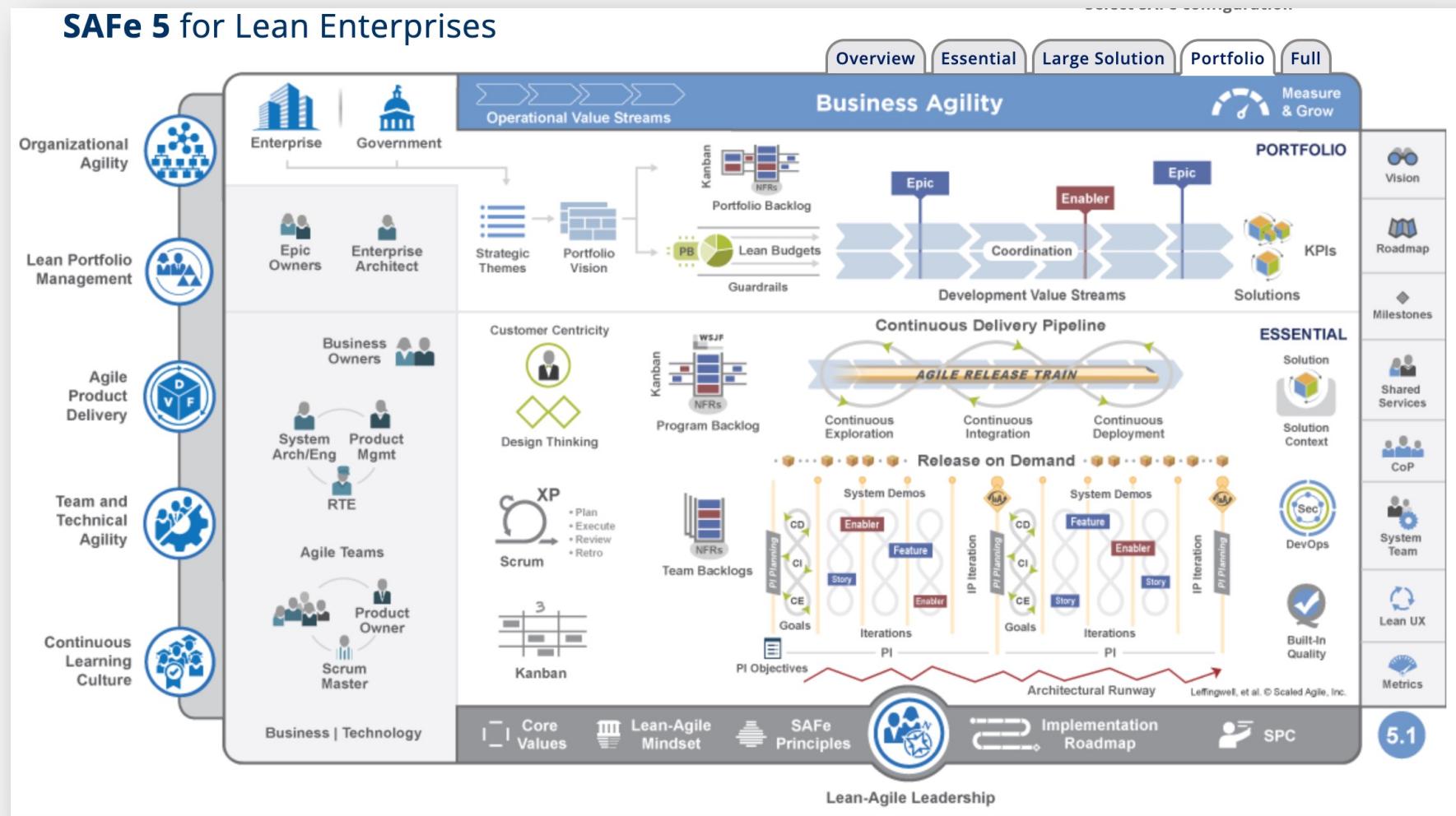


Intègre la philosophie **Lean** et **Agile** au développement logiciel

Peut être appliquée aux **grandes sociétés** avec un grand nombre d'équipes et de membres.

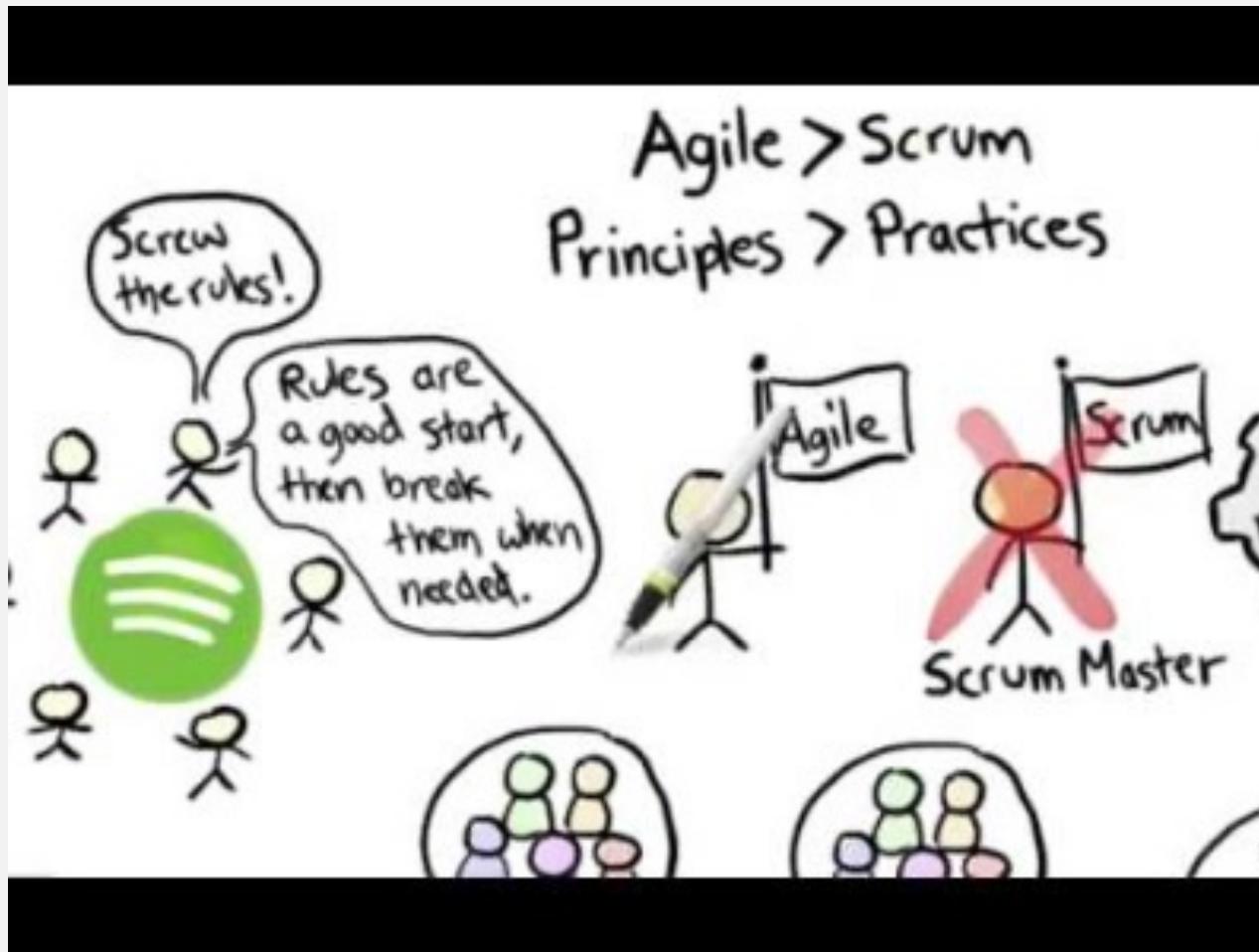
## 8 – 2 La méthode SCRUM et SAFe

### Schéma d'une organisation SAFe



## 8 – 2 La méthode SCRUM et SAFe

Exemple avec Spotify :



## 8 – 3 L'IT Service Management

### IT Service Management (ITSM)



Qu'est-ce que  
l'ITSM ?



La méthode  
ITIL

## 8 – 3 L'IT Service Management

### Qu'est-ce que l'ITSM ?

Implémentation et une gestion de la **qualité des services IT** qui correspond exactement aux besoins du **business**



Il fournit une structure des processus comme :

Change

Configuration

Release

Incident

Problem Management

Les processus de l'ITSM sous-tendent tout le cycle de vie des services, partant de la stratégie, la conception, la transition, les opérations, l'amélioration continue et la création de valeur

## 8 – 4 L'IT Service Management

### La méthode ITIL



ITIL (IT Infrastructure Library) correspond à une liste des **meilleures pratiques** pour la gestion d'un service IT

La méthode fournit 5 livres couvrant l'ensemble des bonnes pratiques avec :

- ITIL Service Strategy
- ITIL Service Design
- ITIL Service Transition
- ITIL Service Operation
- ITIL Continual Service Improvement

## 8 – 4 L'IT Service Management

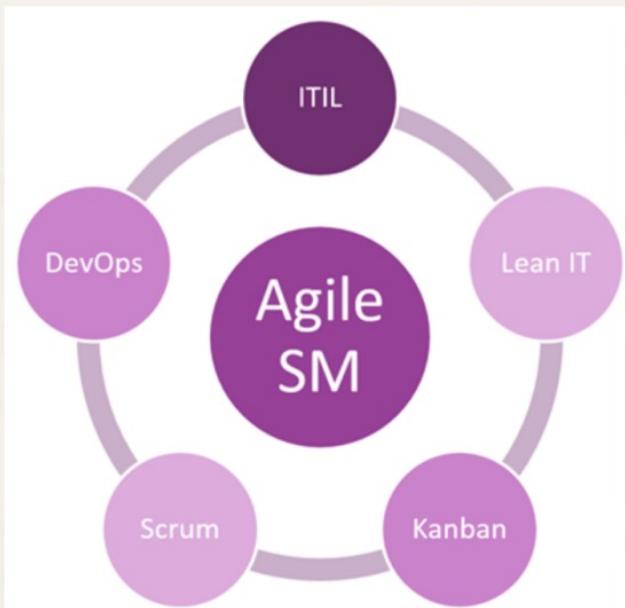
ITSM et DevOps

ITIL et procédures

- Etapes à effectuer**
- Ordre chronologique des actions**
- Responsabilités de chacun**
- Délais et seuils**
- Procédures d'escalades**

## 8 – 5 L'Agile SM – Un peu d'ITSM dans de l'Agilité

### Un peu d'ITSM dans l'agilité



Agile SM s'assure que les processus ITSM reflètent les **valeurs de l'Agilité** et sont conçus avec « juste assez » de contrôle et de structure pour fournir de manière efficace et efficiente les résultats clients.



**Notions clés**

## 8 – 5 L'Agile SM – Un peu d'ITSM dans de l'Agilité

### Les notions clés de l'Agile SM



#### Adaptation

Adapter les pratiques Agile à la conception des processus ITSM



#### Valeurs de l'agilité

S'assurer que les processus ITSM reflètent les valeurs de l'agilité depuis la conception initiale jusqu'au CSI (Continual Service Improvement)



#### Petit à petit

Implémenter la gestion des services, dans de petits incrémentés intégrés



#### MVP (Minimum Viable)

Encourager le « minimum viable » et le « juste assez » pour améliorer la vitesse et la conformité

## 8 – 6 La méthode LEAN

Efficacité et  
Gaspillage

Cartographie  
des flux de  
valeurs

Improvement  
Kata

## 8 – 6 La méthode Lean – Efficacité et Gaspillage.

### Du point de vue du Lean

#### Lean Manufacturing

Le DevOps voit ses racines dans le monde du Lean Manufacturing, dont le but est de produire de manière efficace et sans gaspillage.

#### Le Lean IT

Ainsi le Lean IT applique ces concepts du Lean au développement et à la gestion des produits et services de l'IT.

#### Les principes clés



-  **1 Eliminer le gaspillage (Mudas)**  
Il faut maximiser la valeur en éliminant le gaspillage.
-  **2 Respecter l'équipe**  
Développer les projets autour des équipes motivées.
-  **3 Livrer rapidement**  
Maximiser le ROI avec des livraisons rapides et fréquentes.
-  **4 Optimiser le système**  
Optimiser tout le système.

## 8 – 6 La méthode Lean – Efficacité et Gaspillage

### Les termes "waste" à connaître

| Terme                | Description   | Exemples  |
|----------------------|---|---|
| Defects              | Déviation des pré-requis, erreurs                                 | Dysfonctionnement, erreurs connues, manque d'informations                       |
| Overproduction       | Produire plus ou plus vite que ce qui est requis                  | Documentation et/ou code excessifs  |
| Waiting              | Attente sur une étape précédente                                  | Retards sur les décisions, les validations et les réponses                      |
| Non-use              | Non utilisation de la connaissance ou de la créativité            | Compétences non utilisées, innovation et communication inefficaces              |
| Transportation       | Déplacer des produits d'un endroit vers un autre                  | Perte de temps avec la rédaction d'e-mails ou de meetings                       |
| Inventory            | S'occuper de plus de matériaux que nécessaire                     | Logiciels ou infrastructure non utilisés, blacklogs excessifs                   |
| Motion               | Déplacer des personnes ou des éléments plus que ce qui est requis | Déplacement de code ou d'infrastructure trop important                          |
| Excessive processing | Faire plus que ce qui est requis                                  | Over-ingénierie, échec à créer des templates et d'autres éléments réutilisables |

## 8 – 7 La méthode Lean – Cartographie des flux de valeurs

### Qu'est-ce qu'un flux de valeurs ?

La cartographie des flux de valeurs est un outil Lean qui décrit les flux **d'informations**, de **matériaux**, et de **travail** à travers les silos fonctionnels.

L'accent est alors mis sur la quantification des « **déchets** » incluant le temps et la qualité.

## 8 – 7 La méthode Lean – Cartographie des flux de valeurs

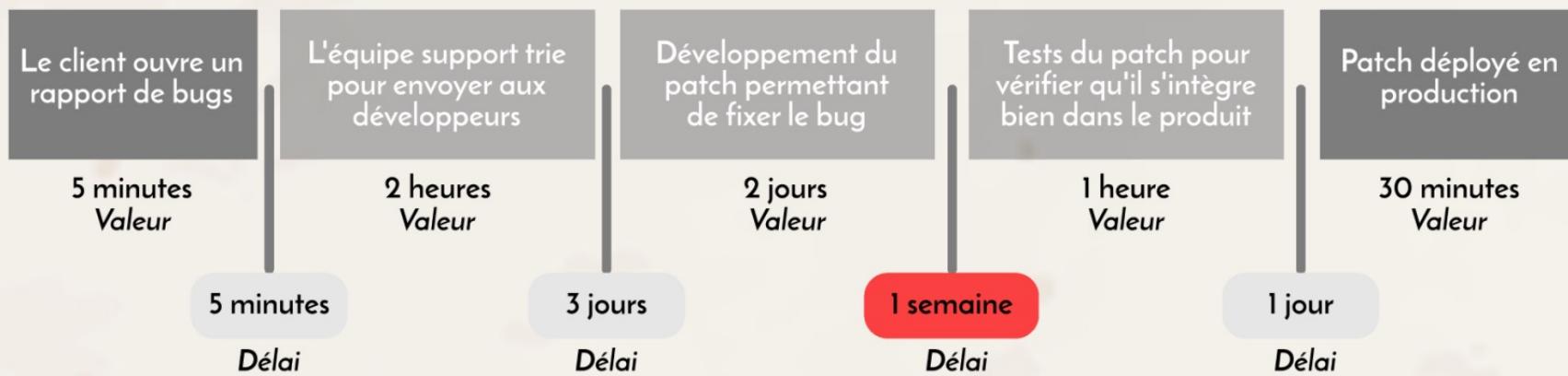
### Représentation du flux de valeur

Ces flux de valeurs correspondent aux activités nécessaires pour la **Conception**, la **Production** et la **Livraison** d'un produit ou service spécifique



## 8 – 7 La méthode Lean – Cartographie des flux de valeurs

### A quoi sert la cartographie des flux ?



→ Avoir une vision globale d'un flux de valeur

→ Identifier les zones où s'accumulent les « déchets » n'apportant aucune valeur, et qui pourraient être éliminées

## 8 – 8 La méthode Lean – L'Improvement Kata.

### L'Improvement Kata de Toyota

Il s'agit d'un moyen structuré de créer une **culture d'apprentissage et d'amélioration continue**.

On parle de Kata (comme dans les arts martiaux) pour faire référence à la mise en place de **routines quotidiennes et d'habitudes**.



## 8 – 8 La méthode Lean – L'Improvement Kata.

### 1. Comprendre la vision



Avoir une **vision commune** à moyen et long-terme de ce que l'entreprise pense être le mieux.

Par exemple Toyota a donné sa vision à long terme comme :

- 0 défauts
- 100% de valeur ajoutée
- Un flux d'une seule pièce, en sequence, à la demande
- La sécurité pour tous (pas d'accidents)

## 8 – 8 La méthode Lean – L'improvement Kata.

### 2. Saisir la situation actuelle



Une fois la vision déterminée, il devient nécessaire de **comprendre comme vous travaillez réellement**, pour mieux saisir comment améliorer ces différents processus.

On peut ainsi proposer un simple schéma des différentes étapes et des mesures (résultats, temps, etc...)

## 8 – 8 La méthode Lean – L'Improvement Kata.

### 3. Etablir le prochain état cible

L'accent doit être mis sur les processus et non sur les résultats (car il s'agit d'un état intermédiaire) :

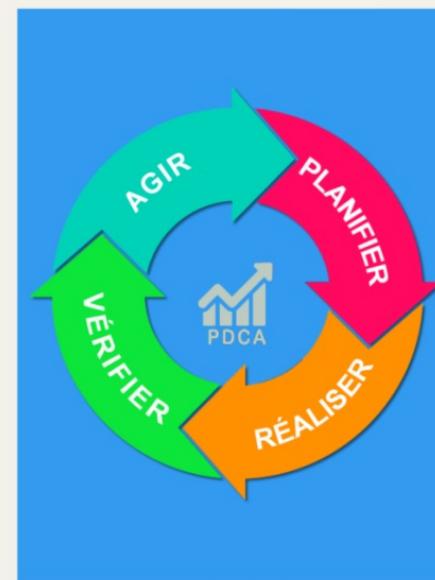
- L'état cible doit être en nombre absolu et pas relatif
- Crédit à un sentiment « d'urgence » pour motiver l'équipe
- Règle des Boucles d'or (pas trop dur, mais pas trop facile)



## 8 – 8 La méthode Lean – L'Improvement Kata.

### 4. Etablir le prochain état cible

- Formulation d'une **hypothèse**
- **Prédiction** explicite et détaillée du résultat prévu
- **Exécution** de l'expérience et collecte des mesures
- **Comparaison** de la différence entre la prédiction et le résultat réel de l'expérience
- **Ajuster** le modèle théorique grâce à cette expérience



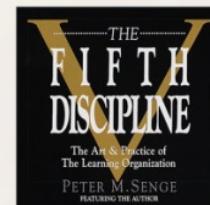
## 8 – 9 Learning Organizations – The Fifth Discipline

### Culture et philosophie de la bienveillance

Elle correspond aux attitudes, croyances, perceptions, et valeurs que les employés partagent en termes de sécurité:

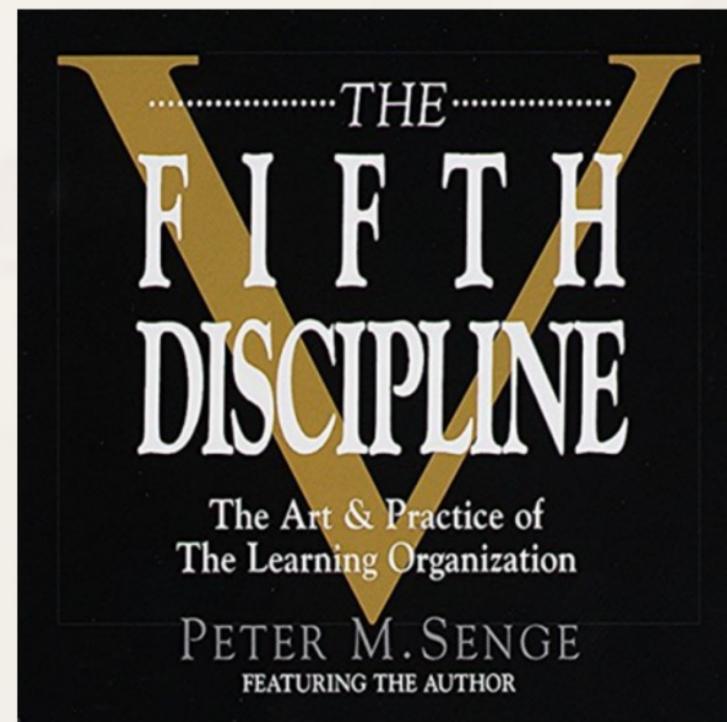
- **Postmortem** sans blâmes contre l'individu ou le groupe
- Evaluation rigoureuse des **incidents**
- Identifier les **SPOF** (Single Point of Failure)
- **Andon Cord** (Lever la main et demander de l'aide aux autres pour résoudre le problème)

Voir les incidents comme une opportunité **d'apprendre** et **d'améliorer** l'efficacité des collaborateurs.



## 8 – 9 Learning Organizations – The Fifth Discipline

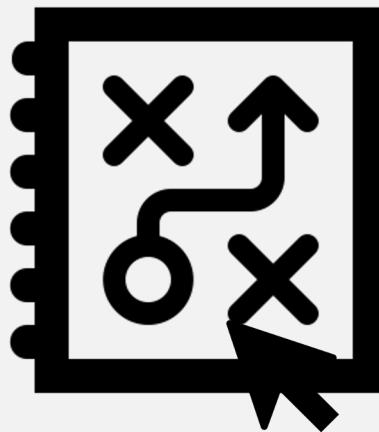
- Promouvoir l'apprentissage
- L'amélioration nécessite d'apprendre quelque chose de nouveau
- Ne pas apprendre crée une dette culturelle
- Les humains aiment maîtriser et cela les rassure
- L'engagement de la direction dans ce processus est essentiel



*Peter M. Senger - The Fifth Discipline*



## UN PETIT QUIZ





## UN PETIT QUIZ

