

# DevOps Foundation



# Introduction

## DevOps Foundation

- **La demande de DevOps a augmenté sur le marché**
- **L'ingénieur DevOps se classe parmi les cinq meilleurs salaires IT avec un salaire moyen de \$111,683.** Dice's "2019 Tech Salary Report,"
- **DevOps offre un cheminement de carrière défini qui promet une croissance constante**

### CEPENDANT

**Les facteurs qui ont tendance à constituer les plus grands défis, sont liés aux personnes et non à la technologie.** George

Spafford, Senior Director Analyst at Gartner



# Introduction

Exin DevOps Foundation

**Qui devrait suivre cette formation ?**

# Introduction

## DevOps Foundation

- **Consultants IT:**
- ✓ **Ingénieur / Administrateur Système ou Cloud**
- ✓ **Développeur, testeur ou administrateur de base de données**
- **Responsable informatique ou chef de projet**
- **Débutants : étudiant ou informaticien souhaitant découvrir le monde du DevOps**

# Objectifs

- Comprendre ce qu'est le mouvement DevOps
- Connaître les enjeux du métier DevOps
- Maîtriser les valeurs, principes et pratiques du DevOps
- Obtenez un aperçu des outils d'automatisation
- Préparer et réussir la certification





# Details de l'examen

**Durée :**

60 minutes

**Documents autorisés :**

Support not allowed

**Nombre de questions:**

40

**Pourcentage de succès:**

65% (26 réponses correctes ou plus)

# SOMMAIRE

- **DevOps : aperçu général**
- **Principes et valeurs DevOps**
- **Méthodes traditionnelles vs DevOps**
- **Pratiques DevOps**
- **Application DevOps**
- **DevOps et outils**
- **Bonus**



# Objectives

En complétant cette section, vous pourrez :

1. Décrire l'évolution historique de la méthodologie cascade à Agile et de Scrum à DevOps.
2. Décrire les origines technologiques et méthodologiques du DevOps.
3. Définissez DevOps, ses avantages pour l'entreprise et l'informatique.



# Introduction

Qu'est-ce que DevOps ?

**QUESTION : QU'EST-CE QUE DEVOPS ?**



1

# DevOps Aperçu



# DevOps – Aperçu général

## Problématique

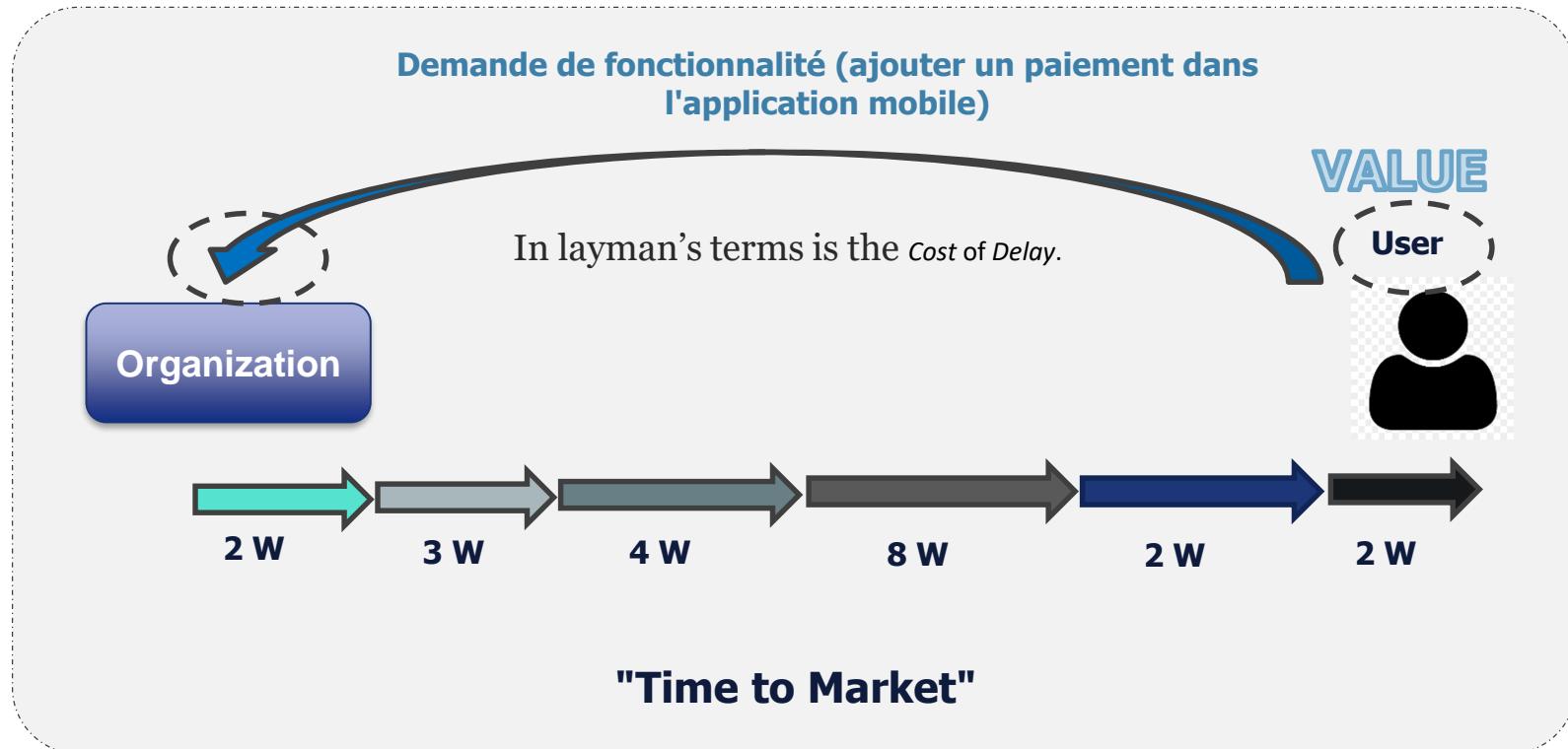
### **Posons le problème !**

Nous vivons aujourd’hui une véritable fusion entre le business et l’informatique. Toute entreprise, quel que soit son secteur d’activité, s’appuie sur les services informatiques. Répondre à des besoins de plus en plus changeants et de plus en plus complexes dépend de sa capacité à fournir des services informatiques de qualité à un rythme toujours plus rapide.

**L'indicateur clé du business agile : le « Time to Market » !**

# DevOps – Aperçu général

## Time to Market





# DevOps – Aperçu général

## Cost of Delay

In layman's terms is the *Cost of Delay*.

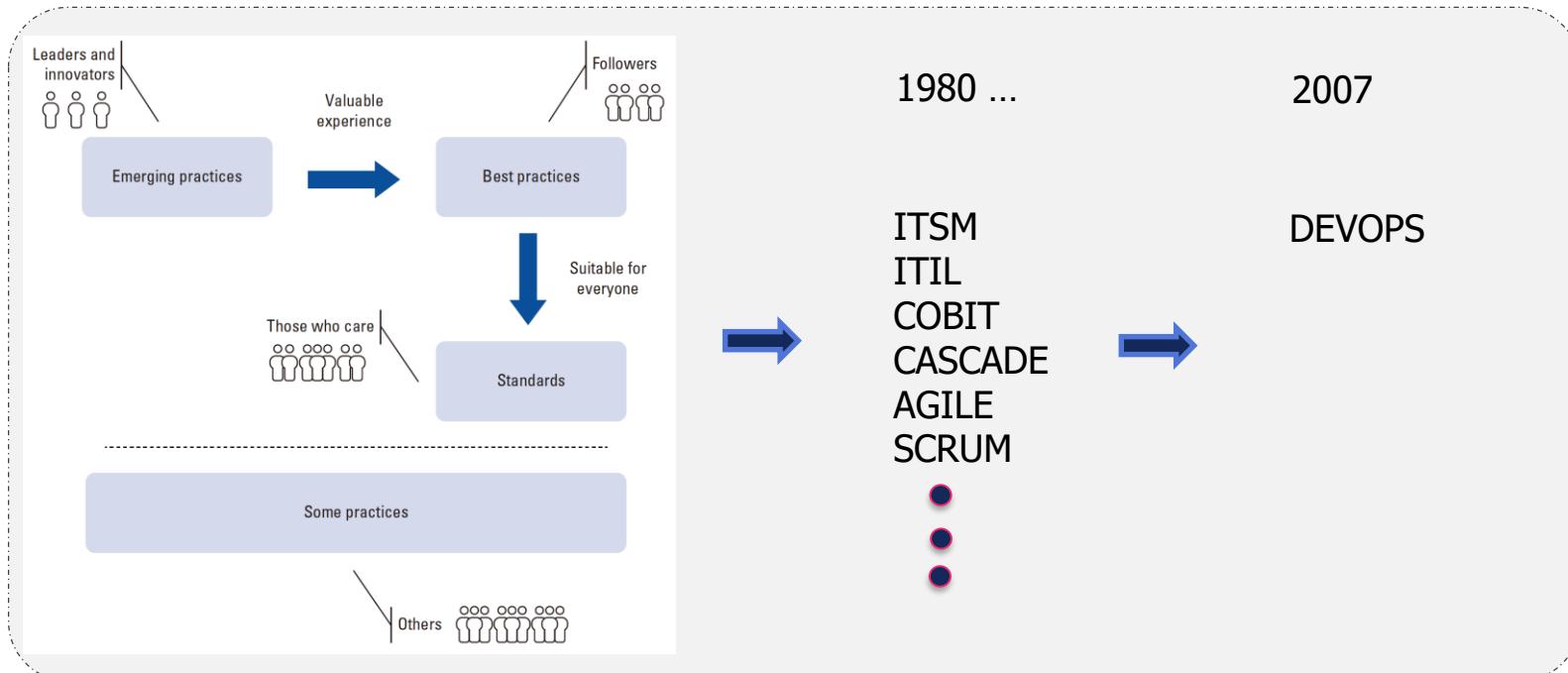
**Plus simplement, c'est la réponse à la question : « Que cela nous coûterait-il si cela était retardé d'un mois ? » Ou bien, "Qu'est-ce que cela nous vaudrait si nous pouvions l'obtenir 1 mois plus tôt ?"**



# Origines DevOps

# DevOps – Aperçu général

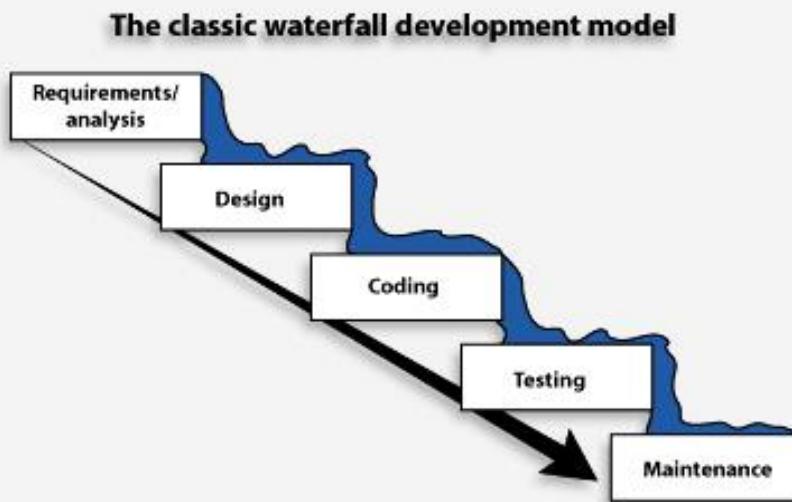
## Origines du DevOps



# DevOps – Aperçu général

## De la cascade (Waterfall) à l'agilité en passant par le cloud computing

### ➤ A - Waterfall



- Au 20ème siècle, le modèle Waterfall était la principale méthodologie de développement de logiciels.
- Dans les années 1990, Internet a accéléré le besoin de lancements rapides de quelques années à quelques mois.

# DevOps – Aperçu général

## De la cascade à l'agilité au cloud computing

### ➤ B - AGILE

En 2001, les experts du « 17 » estimaient que le modèle traditionnel (Waterfall Model) ne correspondait plus aux contraintes et aux exigences d'organisations en évolution rapide. Valeurs et principes agiles :

- Un logiciel fonctionnel
- Livraison continue
- Collaboration entre métier et informatique
- Motivation des employés
- Agilité
- Simplicité
- Auto-organisation
- Excellence technique



Création du manifeste Agile

# DevOps – Aperçu général

## De la cascade à l'agilité au cloud computing

### ➤ AGILE MANIFESTO

Individuals and interactions	over	Processes and tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

*That is, while there is value in the items on the right, we value the items on the left more*

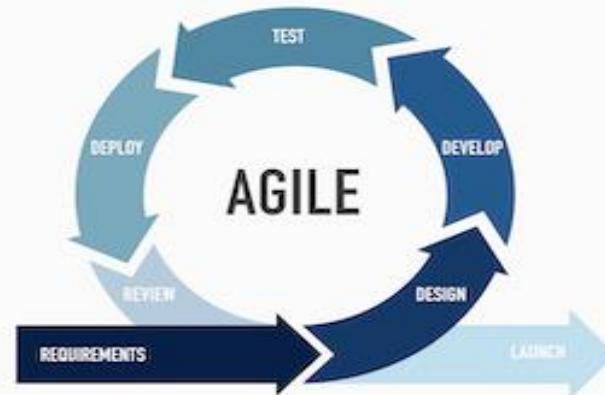
# DevOps – Aperçu général

De la cascade à l'agilité au cloud computing

## AGILE vs WATERFALL



VS



# DevOps – Aperçu général

## Les origines du DevOps

### ➤ **LEAN SOFTWARE DEVELOPMENT :**

Les 7 types de déchets :

- 1) Defects ou bugs (Defects du Lean Manufacturing)
- 2) Changement de tâche (Transport dans le Lean Manufacturing)
- 3) Fonctionnalités supplémentaires (surproduction dans le Lean Manufacturing)
- 4) Transferts (Mouvement dans le Lean Manufacturing)
- 5) Retards (attente dans le Lean Manufacturing)
- 6) Travail partiellement réalisé (inventaire en Lean Manufacturing)
- 7) Réapprentissage (traitement supplémentaire dans le Lean Manufacturing)

# DevOps – Aperçu général

## Les origines du DevOps

### ➤ LEAN SOFTWARE DEVELOPMENT :

Definitions:

- **Jidoka (autonomation)** est le premier pilier du système de production Toyota. Il s'agit d'une automatisation avec une touche humaine, ou de la façon d'automatiser des opérations simples et répétitives, tout en conservant le contrôle humain pour orchestrer tous ces outils et gérer des situations complexes.
- **J-I-T (Just in Time)** est le deuxième pilier du système de production Toyota. C'est le flux permettant de livrer juste à temps (ni avant ni après) le produit demandé, dans la quantité demandée.
- **Kaizen (continuous improvement)** consiste à tirer les leçons des problèmes, à enrichir notre « système » et cela à travers une approche structurée et des outils d'analyse éprouvés (PDCA (Plan, Do, Check, Act), 5 pourquoi, diagramme de Pareto etc). Nous voulons tous nous améliorer, Kaizen nous aide à structurer et à rendre efficaces ces efforts d'amélioration.

# DevOps – Aperçu général

## Les origines du DevOps

### ➤ LEAN SOFTWARE DEVELOPMENT :

- Code terminé non achevé dans un référentiel, code non documenté et non testé ➔ Retarde la livraison et mène à des problèmes de qualité
  - = Partially done work
- L'administrateur de la base de données doit attendre deux jours pour que le serveur et le logiciel soient installés ➔ Retard de la livraison
  - = Delay
- Revoir la documentation à chaque fois que nous avons besoin d'écrire un script de déploiement ➔ à chaque fois il faut réapprendre le comment ?
  - = Relearning

# DevOps – Aperçu général

## De la cascade à l'agilité au cloud computing

### ➤ C - INFRASTRUCTURE AS A CODE

```
{  
    "Description": "Create an EC2 instance running the Amazon Linux 32 bit AMI.",  
    "Parameters": {  
        "KeyPair": {  
            "Description": "The EC2 Key Pair to allow SSH access to the instance",  
            "Type": "String"  
        }  
    },  
    "Resources": {  
        "Ec2Instance": {  
            "Type": "AWS::EC2::Instance",  
            "Properties": {  
                "KeyName": { "Ref": "KeyPair" },  
                "ImageId": "ami-3b355a52"  
            }  
        }  
    },  
    "Outputs": {  
        "InstanceId": {  
            "Description": "The InstanceId of the newly created EC2 instance",  
            "Value": { "Ref": "Ec2Instance" }  
        }  
    },  
    "AWSTemplateFormatVersion": "2010-09-09"  
}
```

# DevOps – General overview

De la cascade à l'agilité au cloud computing

## ➤ C - INFRASTRUCTURE AS A CODE

### Virtualization

- Utilisation efficace des ressources matérielles

### Cloud Computing

- VPN pour envoyer des paquets de données privés sur des canaux partagés avec sécurité, confidentialité et qualité de service
- Les grands fournisseurs ont rendu les ressources virtuelles accessibles et fiables

# DevOps – Aperçu général

## De la cascade à l'agilité au cloud computing

### Cloud computing and virtualization

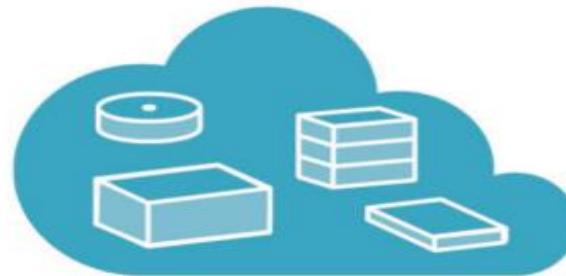
- Virtualization
  - fast provisioning of environments
  - reduced time to deploy software
  - easier to offer 'as a service' infrastructure
  - standardizing hardware
- Cloud computing
  - fast access, easily scalable
  - deploy to a completely standardized stack
  - no need to worry about configuring or maintaining testing, staging, or production environments, or virtual machine images



# DevOps – Aperçu général

## De la cascade à l'agilité au cloud computing

Resources offered by commercial providers have become affordable and reliable; they also assured the necessary level of security.



The customers' attitude to the clouds and their use has changed from “*someone else is controlling my hardware somewhere*” to “*I have an infrastructure that I manage remotely*”.

# DevOps – Aperçu général

## De la cascade à l'agilité au cloud computing

### ➤ C - INFRASTRUCTURE AS A CODE

#### Cloud Computing

Selon le NIST (The US National Institute of Standards and Technology) les caractéristiques essentielles du Cloud Computing sont :

- ✓ Libre-service à la demande
- ✓ Large accès au réseau
- ✓ Mutualisation des ressources
- ✓ Élasticité rapide
- ✓ Service mesuré



Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

This cloud model is composed of five essential characteristics:

- ✓ **On-demand self-service:** A consumer can unilaterally provision computing capabilities, such as server time and network storage, automatically as needed without requiring human interaction with each service provider.
- ✓ **Broad network access:** Capabilities are available over the network and accessed via standard mechanisms that promote use by heterogeneous thin or thick client platforms (such as mobile phones, tablets, laptops, and workstations).
- ✓ **Resource pooling:** The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There's a sense of so-called *location independence*, in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (by country, state, or data center, for example). Examples of resources are storage, processing, memory, and network bandwidth.
- ✓ **Rapid elasticity:** Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

- 
- ✓ **Measured service:** Cloud systems automatically control and optimize resource use by leveraging a metering capability at a level of abstraction that's appropriate to the type of service (storage, processing, bandwidth, or active user accounts, for example). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

# DevOps – Aperçu général

## De la cascade à l'agilité au cloud computing

### ➤ C - INFRASTRUCTURE AS A CODE

Resources offered by commercial providers have become affordable and reliable; they also assured the necessary level of security.



The customers' attitude to the clouds and their use has changed from "*someone else is controlling my hardware somewhere*" to "*I have an infrastructure that I manage remotely*".

# DevOps – Aperçu général

## De la cascade à l'agilité au cloud computing

**First**, l'émergence de nouvelles méthodes d'interaction avec les clients finaux et l'application de techniques de développement agiles.



**Second**, Nouvelles technologies pour la gestion des infrastructures :

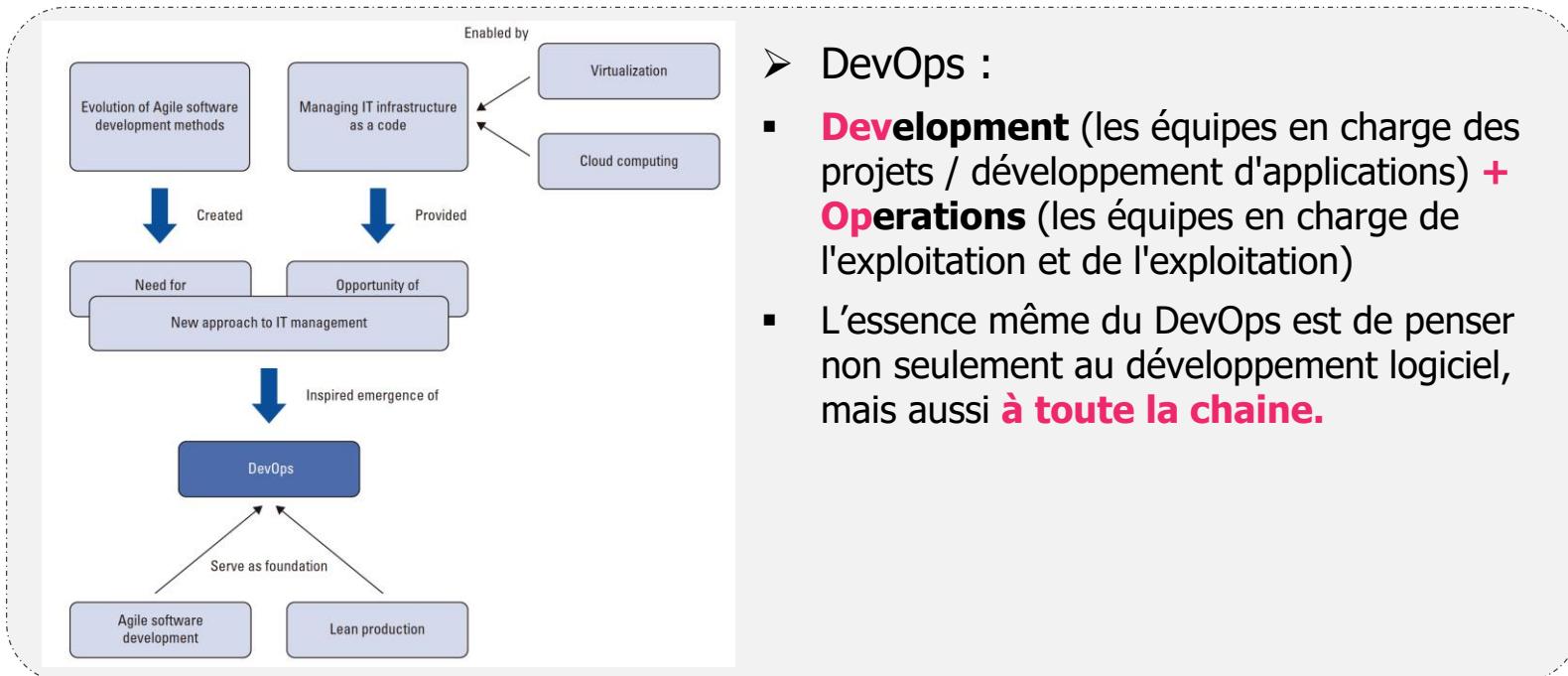
- Virtualization
- Cloud computing



Il est devenu possible de **organize** L'informatique fonctionne différemment.

# DevOps – Aperçu général

## Definition du DevOps



# DevOps – Aperçu général

## Definition du DevOps

- En regroupant les différents termes de plusieurs définitions données sur le web, DevOps c'est :
  - Un mouvement de professionnels hérité de 2 philosophies

LEAN SOFTWARE DEVELOPMENT

AGILE

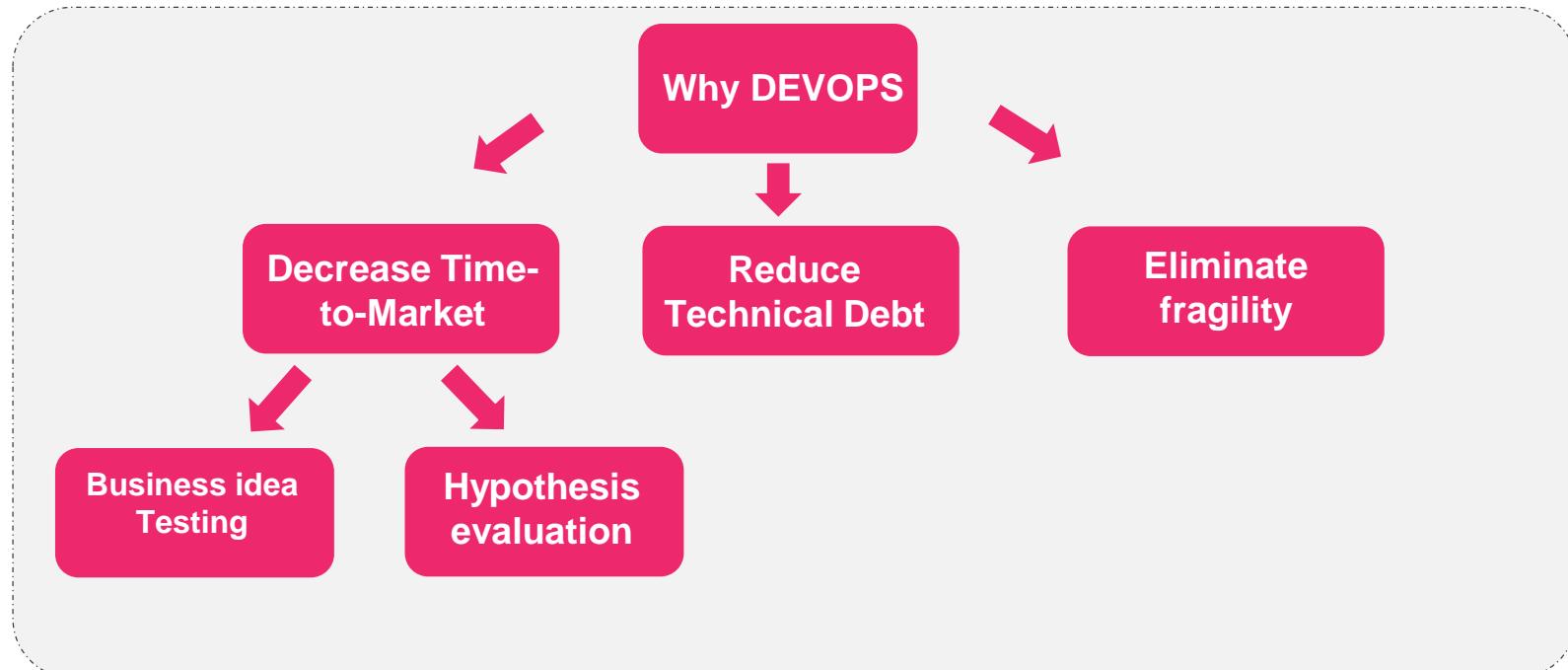
- Une culture
- Un processus agile sur toute la chaîne (du développement à la production)
- Une nouvelle approche technique
- Une autre approche humaine
- Une réponse aux problématiques nouvelles telles que le déploiement massif et le déploiement régulier.



# DevOps Benefits

# DevOps – Aperçu général

## DevOps Benefits



# DevOps – Aperçu général

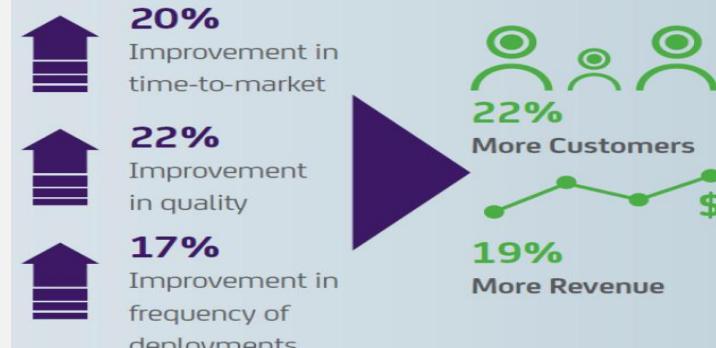
## Benefits : Time To Market

### Decrease Time-to-Market

- Réduction de la taille des produits.
- Réduction des pertes dans le déploiement d'un changement en production.
- Amélioration continue.
- Élimination des déchets :
- Vérification des builds
- Contrôle qualité précoce
- ...
- Équipes autonomes
- Automatisation

### DevOps Benefits Are Real and Quantifiable!

What Has Been the Percentage Improvement, Increase or Reduction for Each of the Quantifiable Benefits You Have Seen?



Source: CA study "What smart businesses know about devops". Panel: 1,300 IT decision-makers in 21 countries Available at <http://aka.ms/devopsca>

# DevOps – Aperçu général

## Benefits : Technical Debt



The debt occurs when a programmer chooses a non-optimal way to solve a problem in order to shorten the development time.



# DevOps – Aperçu général

## Benefits : Technical Debt

### Reduce Technical Debt

*DL'ebt survient lorsqu'un membre de l'équipe choisit une manière sous-optimale de résoudre un problème afin de réduire le temps de développement. les solutions non optimales accumulées conduisent à une détérioration progressive du produit, par conséquent, une dégradation de la qualité du produit.*

*Le non-respect de la conception, intentionnel ou non, entraîne des coûts supplémentaires à l'avenir. **Ce sont là les intérêts.** C'est pourquoi on parle de dette technique, pour montrer l'analogie avec la dette en finance d'entreprise. Cela implique qu'il est préférable de rembourser la dette un jour plutôt que de continuer à payer des intérêts encore et encore. Lorsque nous codons le plus rapidement possible et de manière non optimale, nous faisons appel à un technicien dette que nous remboursions tout au long de la vie du produit sous forme de temps de développement de plus en plus longs et de bugs de plus en plus fréquents.*



# DevOps – Aperçu général

## Avantages : Dette technique

### Réduire la dette technique

L'élimination des dettes techniques créées antérieurement (sciemment ou accidentellement) est planifiée au même titre que la création de nouvelles fonctionnalités.

DevOps recommande fortement d'utiliser la pratique « faire face aux problèmes aussi souvent que possible » pour éviter une « stagnation » de problèmes que tout le monde connaît, mais que personne ne peut mettre la main.

# DevOps – Aperçu général

## Benefits : Fragility

### Éliminer la fragilité

*Ironiquement, les systèmes les plus importants et les plus rentables pour les entreprises sont les plus*

*fragile. Réduire la fragilité de ces systèmes est extrêmement difficile en raison des risques élevés de perturbation des activités, de la tolérance zéro aux temps d'arrêt et du flux constant de nouveaux changements et améliorations liés spécifiquement à ces systèmes.*

Dans DevOps, le code et le système sont toujours fonctionnels à tout moment et si le prochain changement perturbe leurs performances, il est immédiatement annulé et le système continue de fonctionner correctement.

Les systèmes informatiques doivent réagir de manière indépendante et rapide pour détecter une panne et rétablir un fonctionnement normal, idéalement pour que les utilisateurs finaux ne le remarquent pas et que les données, bien sûr, ne soient pas perdues.

# DevOps – Aperçu général

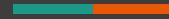
## Benefits : Fragility

### Eliminate fragility

“Anti-fragility is a property of systems that get stronger when exposed to stressors, shocks, volatility, noise, errors, faults, attacks or failures. “

Netflix Chaos Monkey





# DevOps Myths

# DevOps – Aperçu général

## Idées fausses fréquemment exprimées

### **Mythe ou réalité : « DevOps n'est rien d'autre que la continuation des idées Agile »**

Basé en grande partie sur Agile, DevOps étend les idées de développement Agile à la production informatique Agile, à l'ensemble de l'organisation, à l'ensemble du processus, à l'ensemble de la chaîne de valeur.

Tirer parti des avantages du DevOps nécessite **des changements culturels plus importants dans l'entreprise** que l'adoption de l'Agile.

Les **objectifs** fixés pour DevOps ne visent pas seulement à accélérer la livraison, mais également à réduire la dette technique et à éliminer la fragilité.



# DevOps – Aperçu général

## Idées fausses fréquemment exprimées

**Mythe ou réalité : « DevOps est un surhomme capable de coder, de créer des tests, de déployer des environnements et de gérer des infrastructures. »**

DevOps est un changement profond dans les services informatiques, qui ne peut être réalisé en embauchant plusieurs ingénieurs DevOps ou en invitant des experts DevOps.

La capacité à mettre en œuvre un pipeline de livraison de logiciels ne garantit pas le succès. DevOps ne consiste pas seulement à utiliser des outils. Ce sont là les changements organisationnels profonds qui peuvent être accompagnés par ces outils.

# DevOps – Aperçu général

## Idées fausses fréquemment exprimées

- DEVOPS est
- Ni un produit
- Ni un cadre
- Ni une méthode

N'a pas fait l'objet d'un manifeste

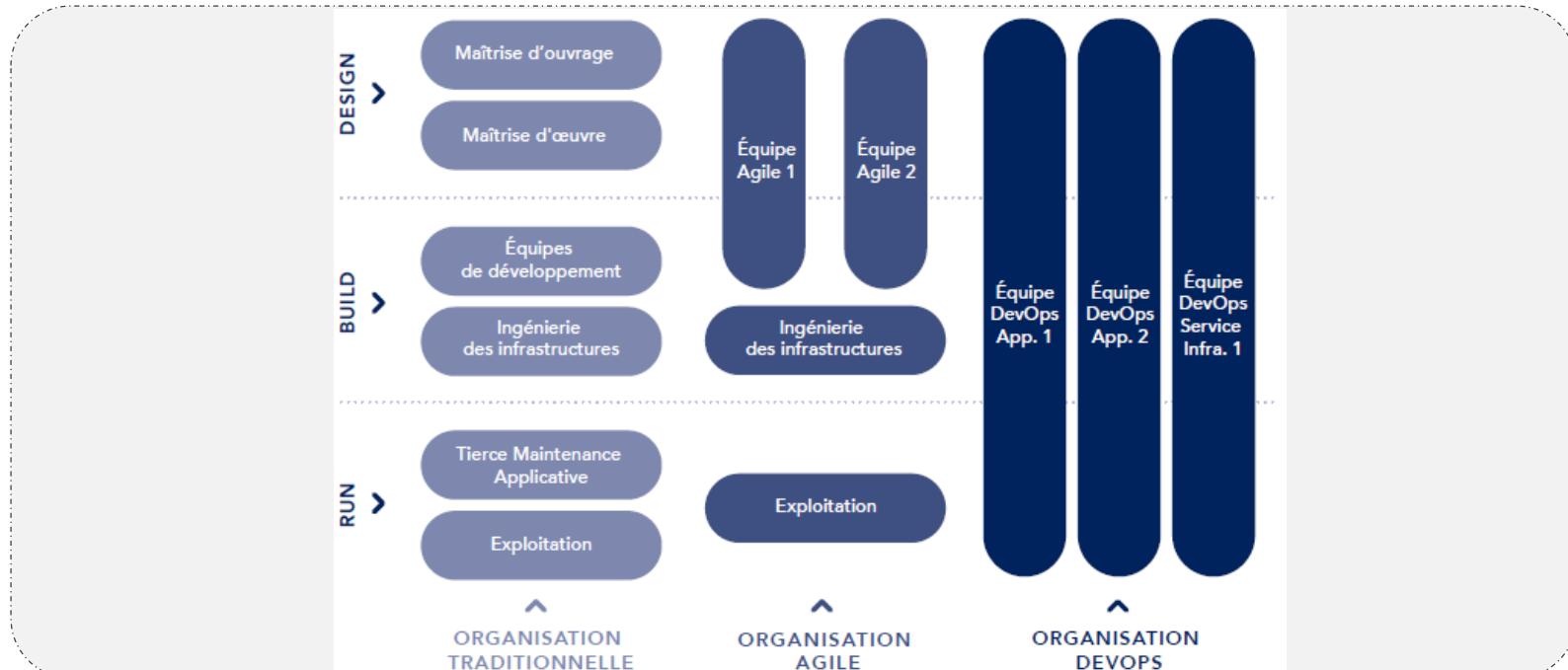
Pas de consensus clair : le fondateur n'a pas défini DEVOPS (Patrick DEBOIS)



# DevOps Organisation

# DevOps – Aperçu général

## Changement organisationnel

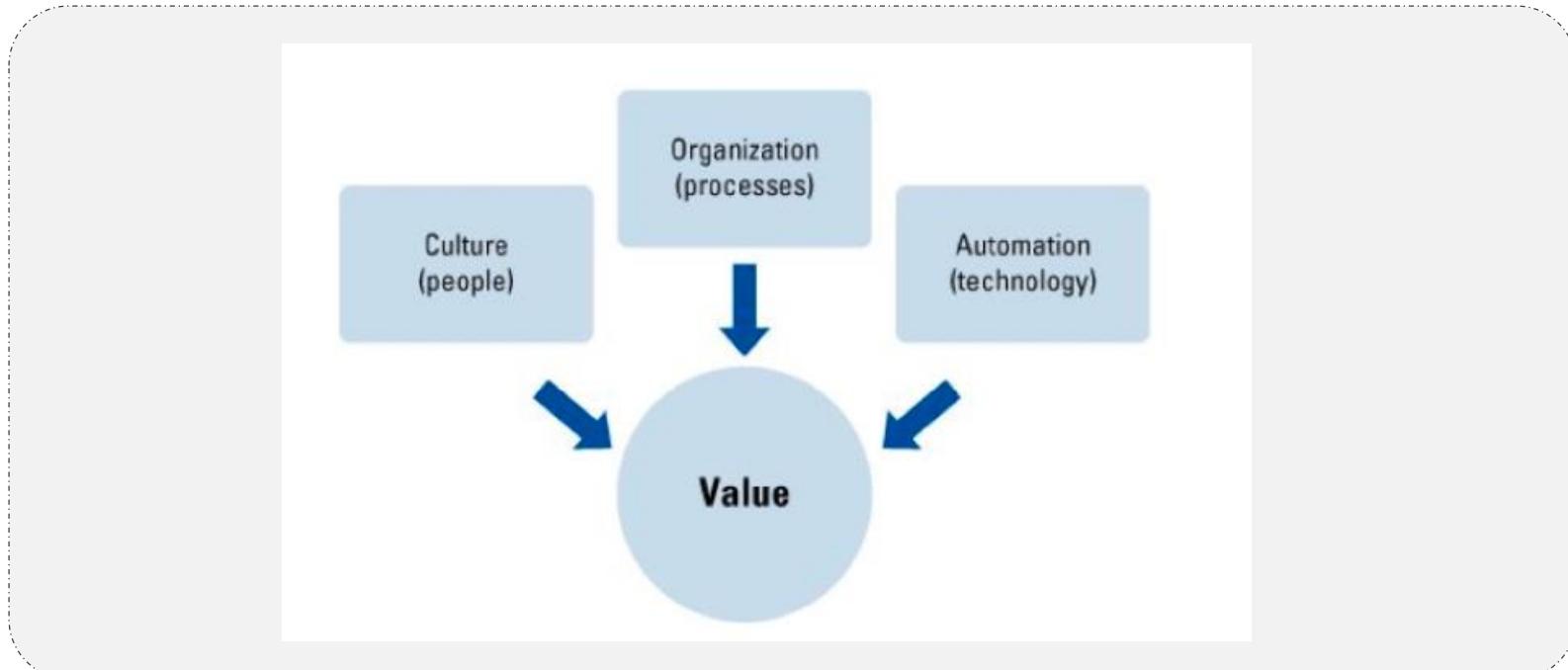




# DevOps Definition

# DevOps – Aperçu général

## La définition



# DevOps – Aperçu général

## La définition

### DevOps

est une évolution des idées de la méthodologie de développement logiciel Agile et du Lean Manufacturing appliquées à la chaîne de valeur informatique de bout en bout,

qui permet aux entreprises de réaliser des bénéfices grâce aux technologies de l'information modernes

en raison de changements culturels, organisationnels et techniques.

2

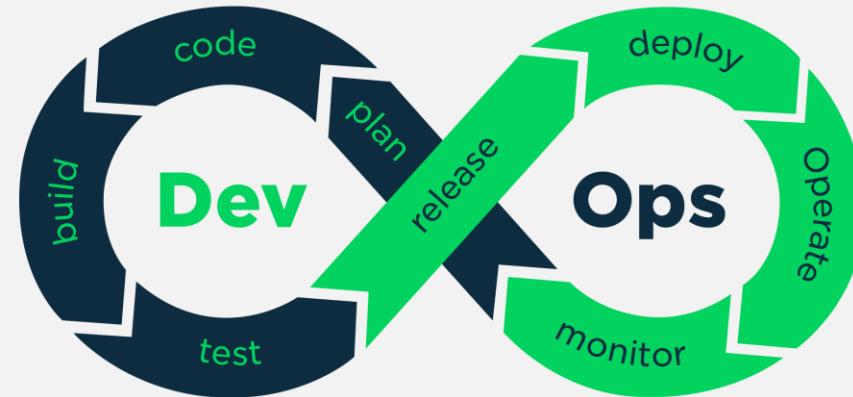
## Les principes du DevOps

# Les principes et valeurs du DevOps

## The principles of DevOps

### ➤ Les principes clés

- Cartographie des flux
- Pipeline de déploiement
- Contrôle de version.
- Gestion de la configuration
- Définition de Terminé





# Cartographie des flux

# Les principes et valeurs du DevOps

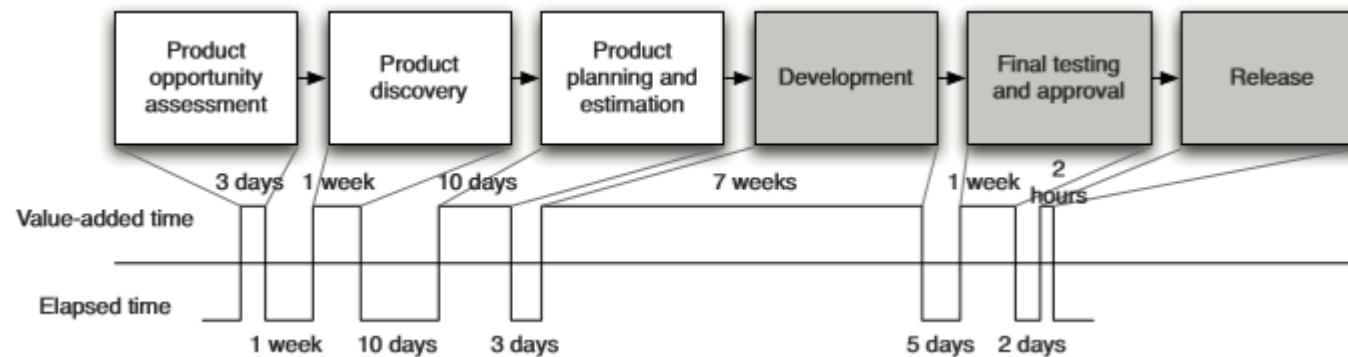
## Les principes du DevOps

**Avec un crayon et un bloc-notes en main, rendez-vous à l'endroit où une demande client arrive dans votre organisation. Votre objectif est de dresser un graphique de la demande moyenne du client, de son arrivée à son achèvement. En travaillant avec les personnes impliquées dans chaque activité, vous esquissez toutes les étapes du processus nécessaires pour répondre à la demande, ainsi que le temps moyen qu'une demande passe à chaque étape. Au bas de la carte, dessinez une chronologie qui montre combien de temps la demande passe dans des activités à valeur ajoutée et combien de temps dans des états d'attente et des activités sans valeur ajoutée.**

**Lean Software Development: An Agile Toolkit “Mary and Tom Poppendieck’s”**

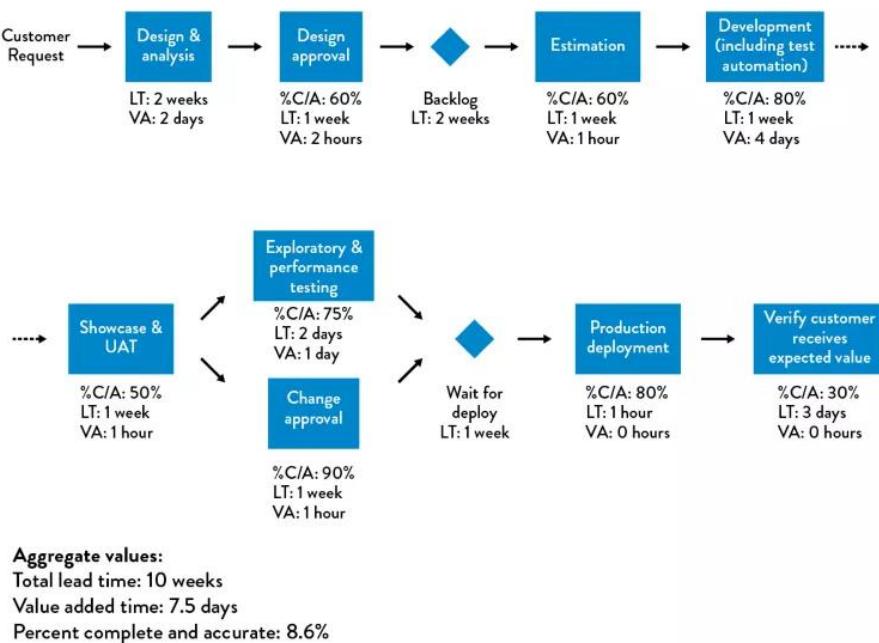
# Les principes et valeurs du DevOps

## Les principes du DevOps



# Les principes et valeurs du DevOps

# Cartographie des flux



- Lead Time (LT)
  - Process Time (PT)
  - Percent Complete and Accurate (%C/A )

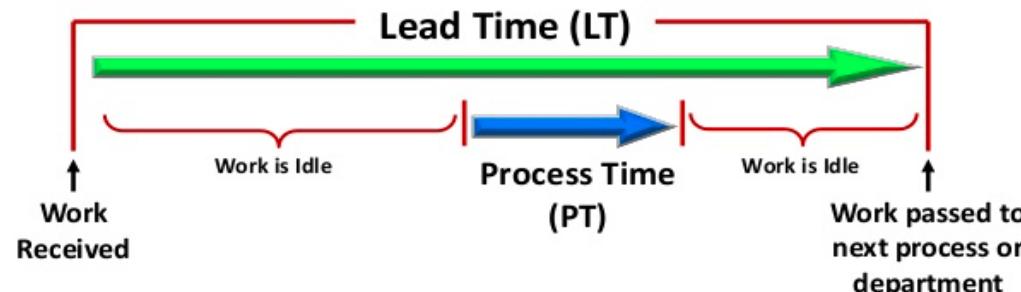
This percentage is obtained by asking downstream stages the percentage of time they receive “usable as is” work, without needing to clarify, add, or correct the information received in the previous step. Only the downstream step can evaluate the % C/A from the previous step.

# Les principes et valeurs du DevOps

## Cartographie des flux

In the Lean [Value Stream Map](#), Process Time (PT) is the time it takes to do the work. Lead Time (LT) includes the Process Time and everything else outside of that for that step. In the case of software development this can include time for ideation, QA testing, deployment time, etc.

## Lead Time vs. Process Time



**Lead Time** = Elapsed time; Throughput time; turnaround time

**Process Time** = Touch time; work time; cycle time

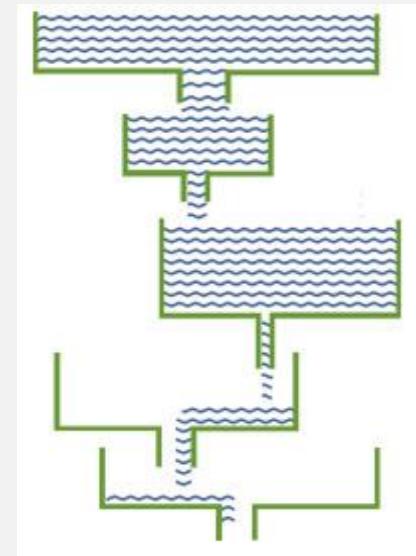
# Les principes et valeurs du DevOps

## Cartographie des flux

### ➤ Theory Of Constraints. (ToC)

“Since the strength of the chain is determined by the weakest link, then the first step to improve an organization must be to identify the weakest link”

— Eliyahu M. Goldratt, The Goal: A Process of Ongoing Improvement

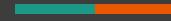


# Les principes et valeurs du DevOps

## Cartographie des flux

### ➤ Theory Of Constraints. (ToC)

- The Theory of Constraints provides an iterative improvement process in five steps, which aims to focus efforts on the constraint alone.
- The five steps are as follows:
  - Identify the constraint (the bottleneck).
  - Exploit the constraint; improve the use of its capacity.
  - Subordinate all processes to constraint.
  - Increase the capacity of the constraint if it is relevant.
  - Start over at step 1 if the constraint has changed.

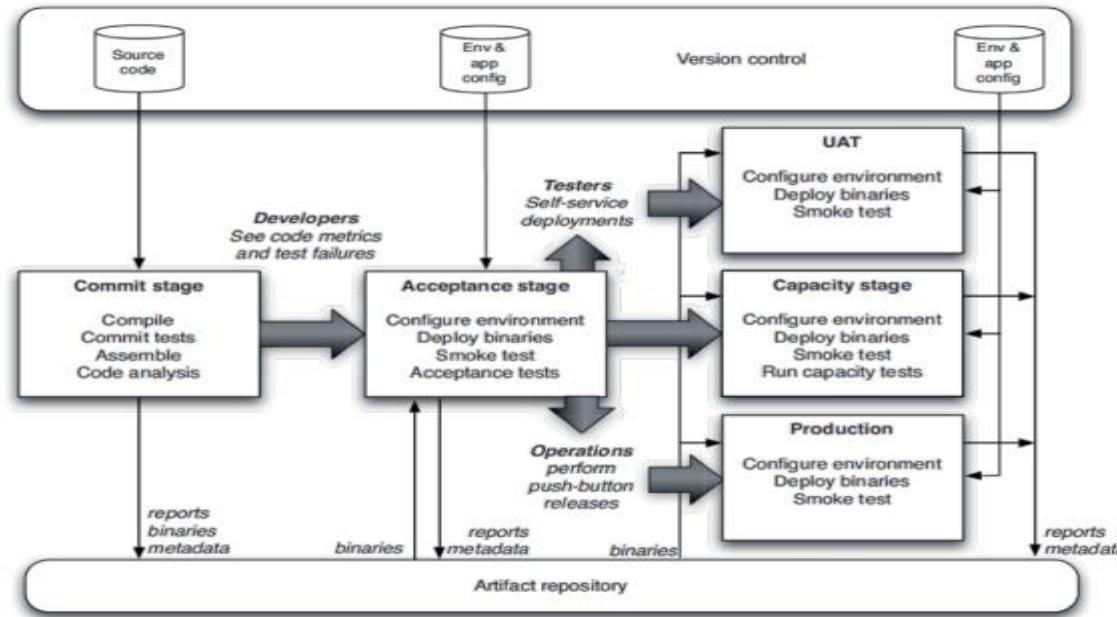


# Deployment pipeline

# Les principes et valeurs du DevOps

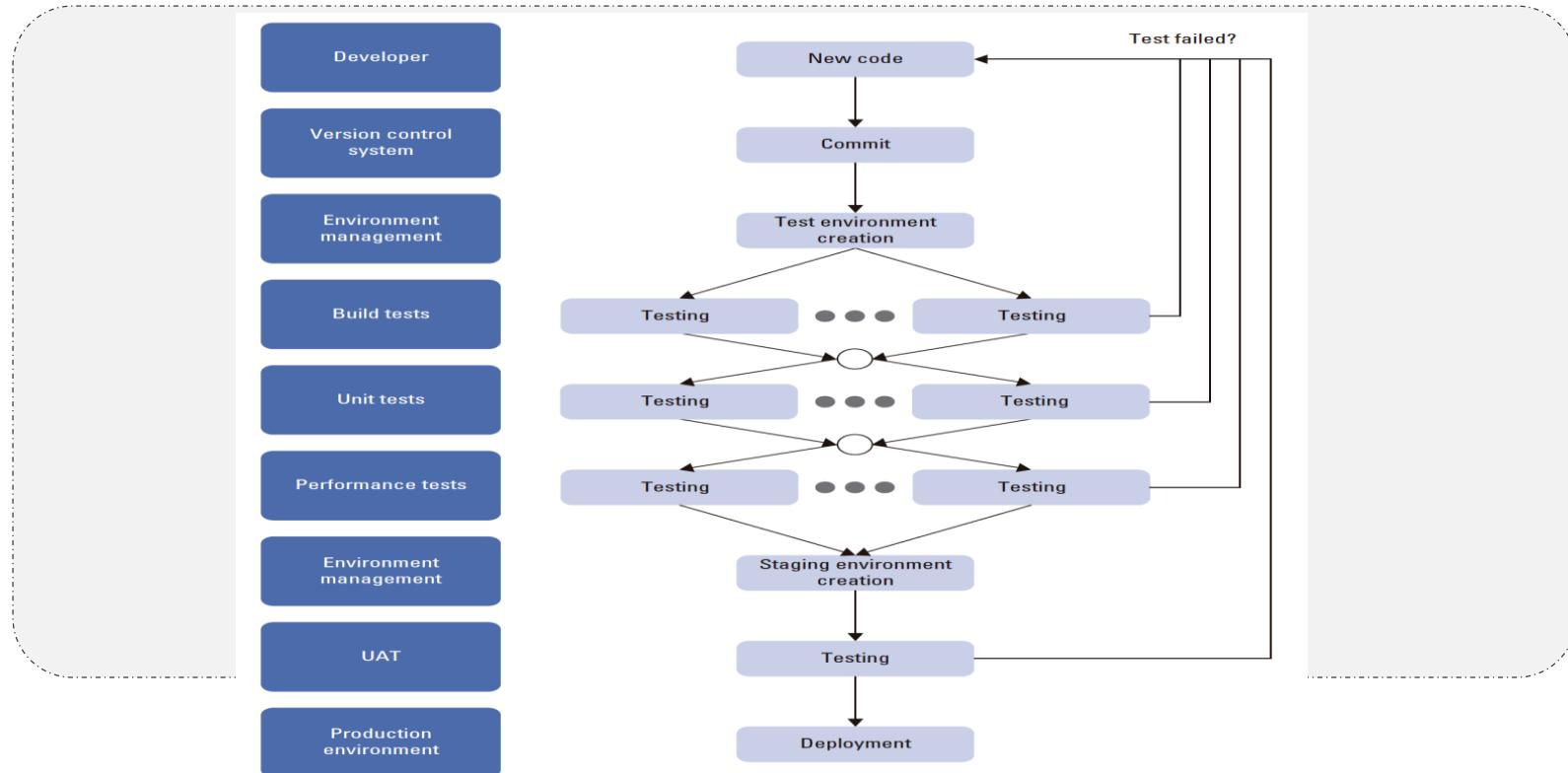
## Les principes du DevOps

The Deployment Pipeline



# Les principes et valeurs du DevOps

## Deployment pipeline



# Les principes et valeurs du DevOps

## Deployment pipeline

- **Un procédé moderne utilisant une pipeline parallèle**
  - Vous permet de produire des résultats beaucoup plus rapidement.
  - Économise des ressources en ne commençant pas les étapes suivantes avant la fin des précédentes.
- **Garantit la qualité du produit**
  - Les modifications qui ne fonctionnent pas bien n'atteignent pas la production et le système est toujours en état de marche.
- **Accélère la livraison des modifications à l'environnement de production**
  - En maximisant l'automatisation de chaque étape.
- **Laisse constamment des enregistrements dans les journaux d'audit**



# Les principes et valeurs du DevOps

## Deployment pipeline

La mise en œuvre du pipeline de déploiement introduit les défis suivants :

1. Un enthousiasme excessif pour l'automatisation au détriment de l'idéologie (processus, personnes et culture) conduit à la création d'un pipeline remarquablement automatisé que personne n'utilise.
2. Au départ, il n'y a pas suffisamment de tests pré-développés pour garantir un fonctionnement stable du pipeline.
3. Dans l'état cible, il y a tellement de tests que le passage d'une modification à travers le pipeline prend trop de temps et nécessite des ressources de calcul importantes, surtout dans le cas d'un grand nombre de petites modifications.



# Les principes et valeurs du DevOps

Many believe that the name ‘pipeline’ is taken by analogy with an assembly line, for example, in a car manufacturing plant. Others believe that the word ‘pipeline’ refers to a liquid or other substance flowing through the pipes, and the deployment pipeline should follow this analogy. Both above opinions are inaccurate.

As the authors of the term, Jez Humble and David Farley<sup>32</sup>, explain, the idea originated from the pipelining used in modern processors, where performance improvements cannot be achieved solely by increasing the clock frequency. The architectural solution used is the parallel execution of instructions, which initially come in sequence. To do this, the processor must ‘guess’ the results of processing in a parallel flow, ‘assuming’ that they will be as required to perform calculations in the current flow. If not, the results of the calculations will be discarded. The time lost due to this ‘unlucky guess’ is more than compensated by the acceleration for those cases when the ‘guess’ was correct.

So, a properly implemented deployment pipeline allows to make development and testing independent of each other in time: it is assumed that testing will be successful, so you can proceed to the next batch of work. The same logic is applied to the parallel testing.

# Les principes et valeurs du DevOps

## Deployment pipeline

- Les trois processus DevOps

**Intégration continue** : est un processus orienté construction et test consistant à compiler, tester et déployer sur un environnement d'intégration. Le but est de tester le plus souvent et le plus possible les non-régressions du livrable pour détecter les bugs le plus en amont possible. La plupart du travail est effectué par des outils de test. Le déploiement sur la plateforme d'intégration devient simple et peut être réalisé par les développeurs sans impliquer d'opérations.

# Les principes et valeurs du DevOps

## Deployment pipeline

### ➤ Les trois processus DevOps

**Livraison continue** : est un processus d'intégration et de production. L'objectif est de tester et de livrer une application à chaque étape de son cycle de vie (réception, pré-production, production). Cette étape est réalisée après validation des tests réalisés en intégration. La phase de test correspond aux tests fonctionnels du livrable. Le passage d'un état à un autre est entièrement automatisé, c'est pourquoi le livrable doit être constitué de manière à pouvoir être déployé en production dès sa mise en réception.

# Les principes et valeurs du DevOps

## Deployment pipeline

- Les trois processus DevOps

**Déploiement continu** : est un processus de production. Le but est de compiler, tester et déployer une application en production. Le déploiement continu nécessite que les processus d'intégration continue et de livraison continue aient été menés à bien. Le déploiement se fait d'un simple « clic sur le bouton ». Après le déploiement, il devrait être possible de mesurer tout impact à l'aide d'outils de mesure et de surveillance des performances. En cas de problème, un processus de restauration automatisé peut être effectué.

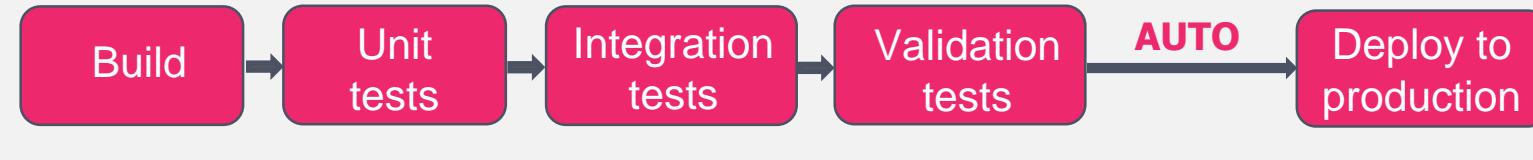
# Les principes et valeurs du DevOps

## Deployment pipeline

### Continuous Delivery



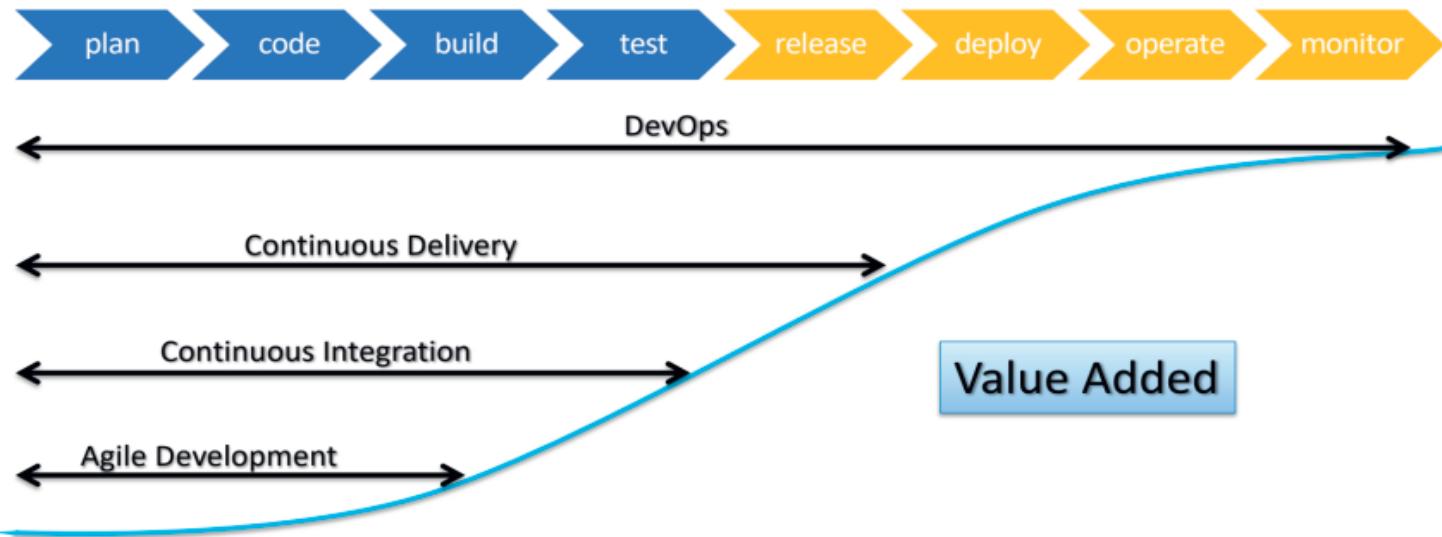
### Continuous Deployment



# Les principes et valeurs du DevOps

## Deployment pipeline

Continuous Delivery vs Continuous Integration



# Version Control

# Les principes et valeurs du DevOps

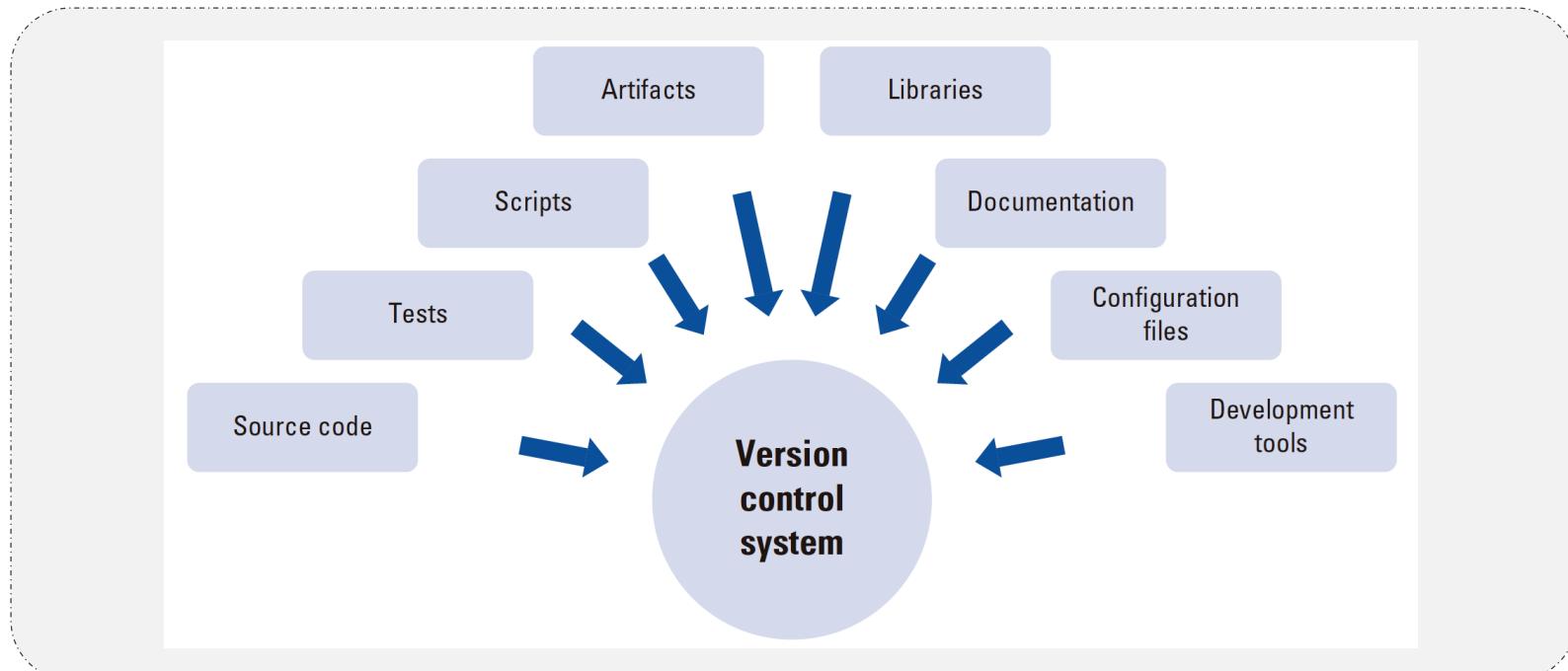
## Version Control

Il s'agit de stocker non seulement le code source, mais également tout ce qui concerne l'ensemble du pipeline :

- des essais,
- des scripts pour créer et modifier des bases de données,
- construire des scripts,
- les scripts de création d'environnement (y compris l'environnement de développement),
- les scripts de déploiement,
- des artefacts,
- bibliothèques, documentation, fichiers de configuration,
- même des outils de développement, tels que des compilateurs, des IDE, etc.

# Les principes et valeurs du DevOps

## Version Control



# Les principes et valeurs du DevOps

## Version Control

### ➤ Profits

- ✓ La capacité de déterminer ce qui a été modifié, quand et par qui.
- ✓ Possibilité de restaurer le système à tout moment, y compris de remettre le système défaillant à un état de fonctionnement garanti avec un minimum d'effort.
- ✓ Permettre à un membre de l'équipe de supprimer librement les fichiers et documents inutiles, sans risque de perte accidentelle d'informations ou de produits importants



# Configuration Management

# Les principes et valeurs du DevOps

## Configuration Management

Dans DevOps, toute modification apportée à l'environnement (infrastructure qui héberge l'application) doit être effectuée par un script stocké dans l'outil de contrôle de version.

La création des environnements se fait automatiquement lors de l'exécution du pipeline de déploiement.

Ce principe nécessite une réorganisation complète du travail et des opérations du support informatique. En effet, les administrateurs n'ont plus le droit de modifier quoi que ce soit dans l'environnement de production de la manière habituelle.

# Les principes et valeurs du DevOps

## Configuration Management

### ➤ Benefits

- DevOps restructure complètement la gestion de l'environnement de production (ainsi que tout autre environnement).
- Toute modification d'un environnement ne peut être effectuée que par des scripts stockés dans le système de contrôle de version.
- La création des environnements se fait automatiquement lorsque le pipeline de déploiement est en cours d'exécution.
- Ce principe nécessite une réorganisation complète du travail et des opérations du support informatique. En effet, les administrateurs n'ont plus le droit de modifier quoi que ce soit dans l'environnement de production de la manière habituelle.

# Les principes et valeurs du DevOps

## Configuration Management

### ➤ Advantages

- ✓ La gestion de la configuration DevOps offre les mêmes avantages que ceux dont vous bénéficiez avec le contrôle complet des versions, mais les **principaux bénéficiaires sont désormais les personnes travaillant dans les opérations.**
- ✓ Maintenant que toutes les modifications sont vérifiées, le système peut être rapidement restauré à un état stable. Si des membres clés quittent l'organisation, les connaissances ne sont pas perdues, etc.



# Definition of Done

# Les principes et valeurs du DevOps

## Definition of Done

### ➤ Definition of Done

Il ne s'agit pas du moment où quelqu'un a fait sa part du travail, mais du moment (QUAND) le client a reçu ou a commencé à recevoir la valeur qu'il attendait.

Cela signifie que l'ensemble de la chaîne de valeur a été entièrement suivi jusqu'à l'environnement de production ; ce n'est qu'à ce moment-là que les travaux seront considérés comme terminés.

En agile, une demande finie : développée, testée et livrée.





# Sommaire



## Les principes du DevOps



3

## Traditionnel vs DevOps

# Traditionnel vs DevOps

## Continual improvement

- Frequent release

Traditionnal	DevOps
Lots de grande taille Livraison programmée Plusieurs jours/semaines Ressources étendues Effort élevé sauvegardes Manuel de documentation	Petits lots Livraison continue Communiquer hebdomadairement / quotidienement Utilisation efficace des ressources Effort régulier Automatisation Continu

# Traditionnel vs DevOps

## Continual improvement

- Les différences les plus importantes

Traditionnal	DevOps
<ul style="list-style-type: none"><li>- Libérer</li><li>- Une release est un groupe de nouvelles fonctionnalités déployées ensemble dans l'environnement de production.</li><li>- Calendrier de livraison</li></ul>	<ul style="list-style-type: none"><li>- Déploiement</li><li>- Livraison partielle ou complète d'une nouvelle fonctionnalité pour les utilisateurs</li><li>- Déployé dès que les tests sont réussis</li></ul>

# Traditionnel vs DevOps

## Automation

### ➤ Pipeline Automation

Les environnements requis pour le pipeline de déploiement sont créés automatiquement par des scripts.

Ces environnements sont automatiquement libérés après utilisation.

Le fonctionnement rapide du pipeline nécessite une automatisation maximale des tests

Le déploiement et la distribution, étapes finales du pipeline, se font également automatiquement.

# Traditionnel vs DevOps

## Amélioration continue

### ➤ Amélioration continue

Tous les défauts identifiés doivent être éliminés immédiatement.

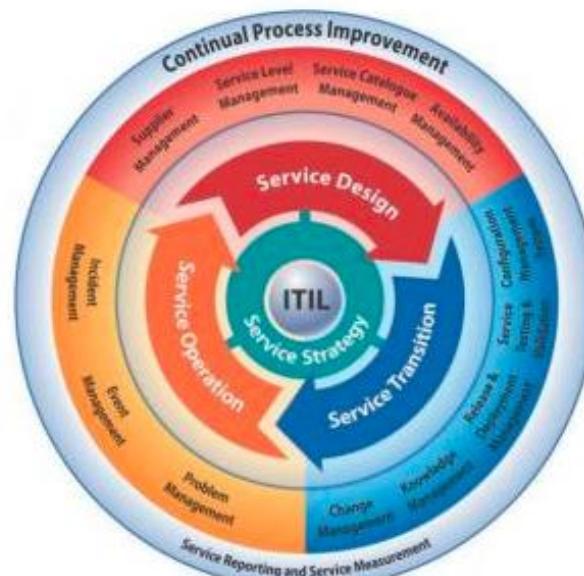
Par exemple, si un script qui exécute le pipeline de déploiement ne fonctionne pas correctement, il doit être corrigé immédiatement.

DevOps recommande de répéter les étapes problématiques aussi souvent que possible, contrairement à la pratique traditionnelle selon laquelle les problèmes peuvent être différés.

Cela permettra de mieux comprendre comment les améliorer et d'ajuster la méthode de travail en conséquence.

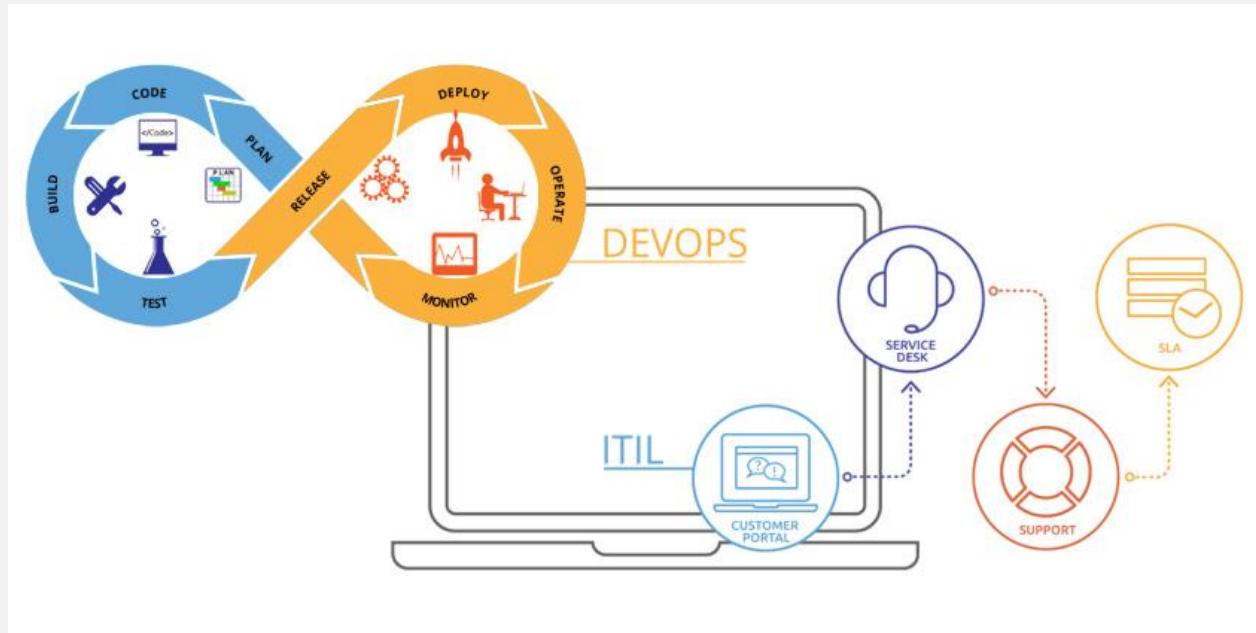
# Operations methodologies (IT Service Management)

- ITSM supports ongoing availability and maintenance of services to ensure the delivery of value and desired service outcomes
- The Service Lifecycle:
  - Service Strategy
  - Service Design
  - Service Transition
  - Service Operation
  - Continual Service Improvement
- Most widely used methodology is ITIL®



# Traditionnel vs DevOps

## DevOps and ITSM



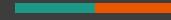


4

## Pratiques DevOps

# Pratiques DevOps

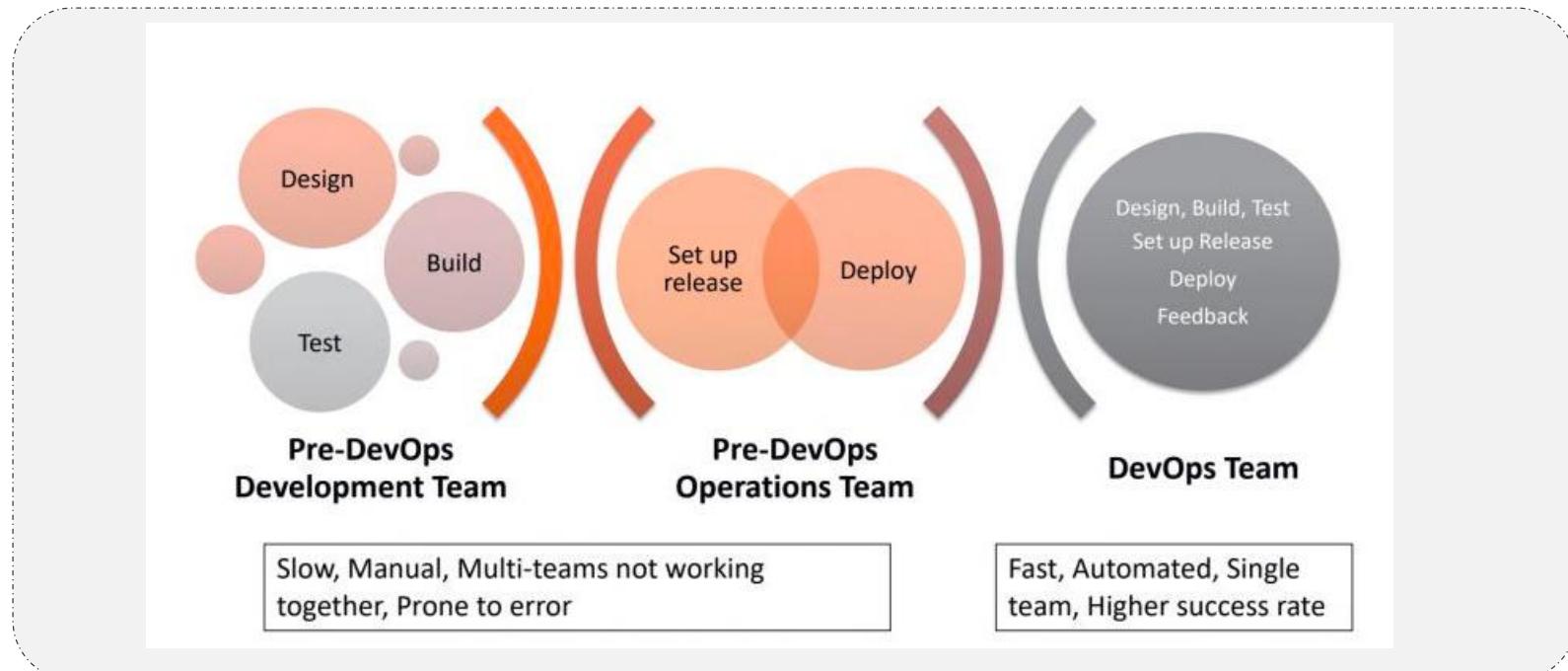
- 
- Les pratiques DevOps sont:
    - Équipe diversifiée
    - Visualiser le travail
    - Petites tailles de lots
    - Soutenir l'innovation
    - Identifier les moyens de gérer les goulets d'étranglement



Diverse team

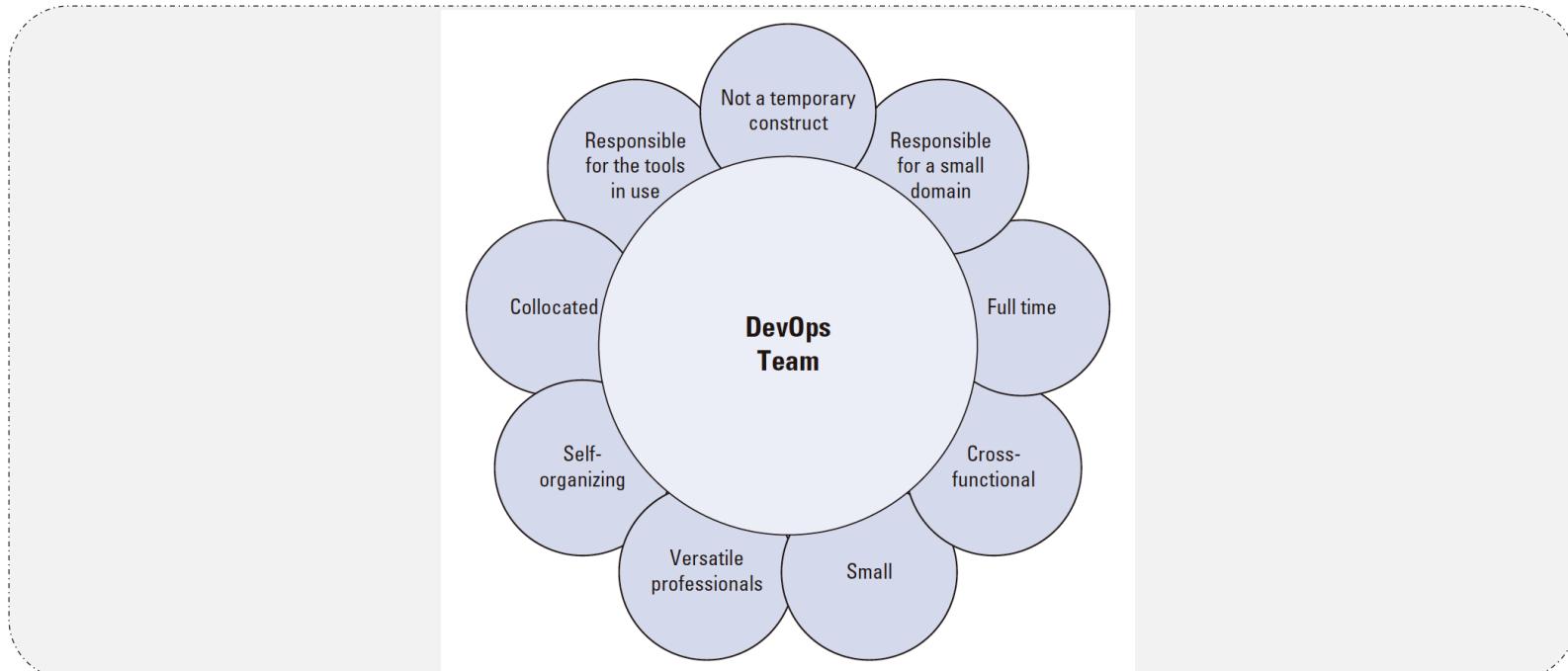
# Pratiques DevOps

## DevOps Team



# Pratiques DevOps

## Équipe autonome et diversifiée



# Pratiques DevOps

## Équipe diversifiée

Recruit a mix of individuals  
Gender, sexuality, religion,  
race, colour,  
age, ability,  
education level,  
experience

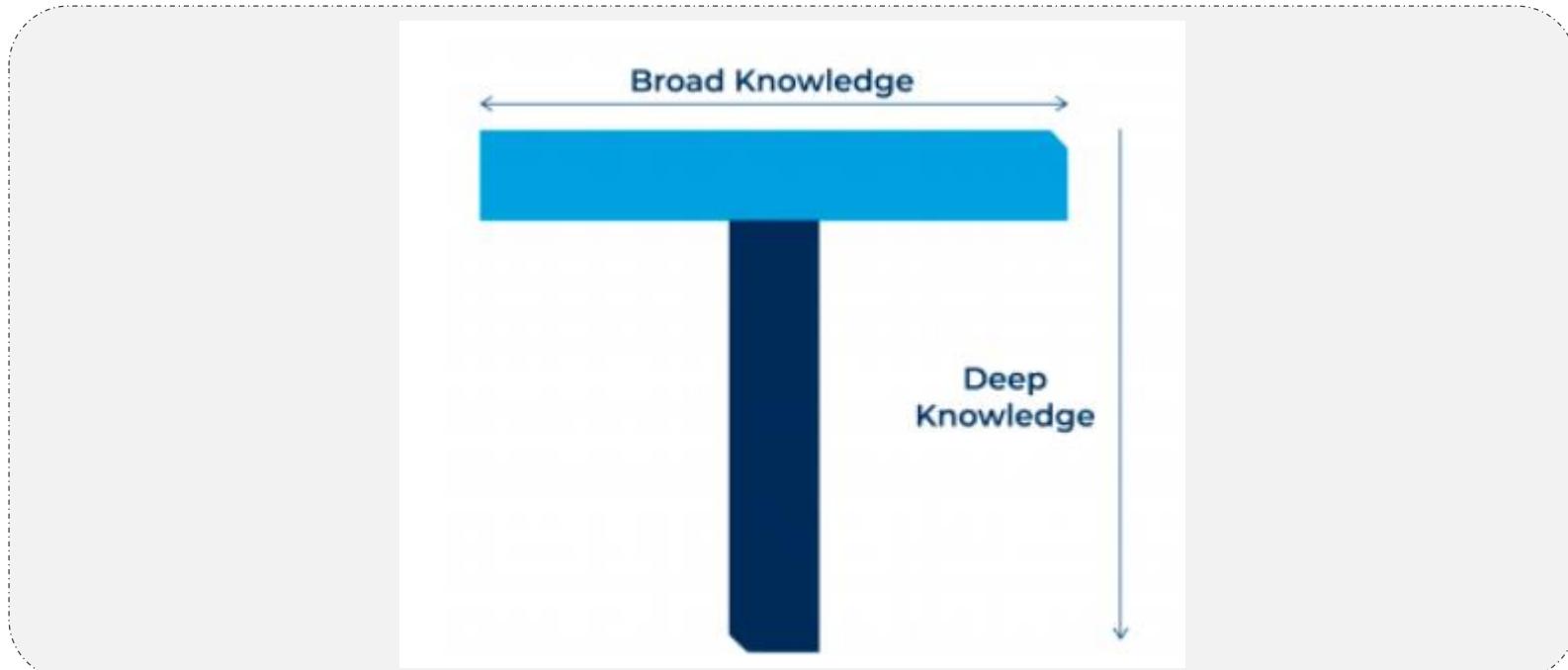


Ensure working conditions  
and methods adapt to  
the diversity

Benefits: Increases different view points and ability to challenge,  
question, innovate and feedback that generates **Learnings**

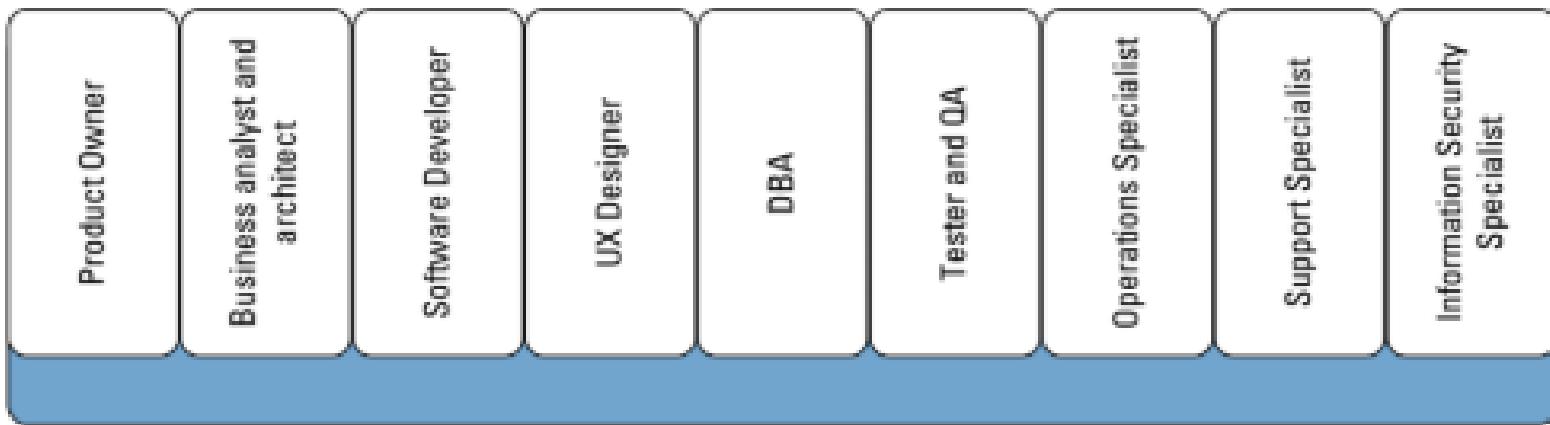
# Pratiques DevOps

## Profil/compétences en forme de T



# Pratiques DevOps

## Exemple d'équipe DevOps



# Pratiques DevOps

## Un nouvel exemple de profils DevOps

**Cloud Architect** - Une personne possédant une vaste expérience pratique dans la création d'infrastructures cloud et comprenant ce qu'elles doivent inclure pour prendre en charge divers types d'applications et de services en production.

**Site Reliability Engineer (SRE)** - Il est le spécialiste DevOps qui se concentre sur la garantie de performances stables et d'une disponibilité ininterrompue des applications à forte charge sur des systèmes à grande échelle.

**DevOps tools administrator** - l'un des principaux rôles DevOps car il est responsable de tous les outils DevOps et de la plateforme cloud utilisée pour les déploiements.



Visualiser  
le travail

# Pratiques DevOps

## Visualiser le travail

The work visualization can support the deployment pipeline and help find answers to the following questions:

- What step that does not allow the rest of the chain to function effectively?
- Where are the resources already or almost exhausted?
- What tasks are blocked, so that they have no chance to be completed in this iteration?
- What remains to be done for the task that has not been completed?
- If we don't have time to do all of the work accepted in this iteration, what part of the task is worth trying to finish to get the maximum outcome possible?

# Pratiques DevOps

## Visualiser le travail



- Managers, sales staff, designers, developers, operators, and customer support staff are one team and share all the business information on visual boards
- Visual control means “Does everybody easily understand the situation by just looking at the boards without explanations?”
- Enables fast view of progress and identification of issues

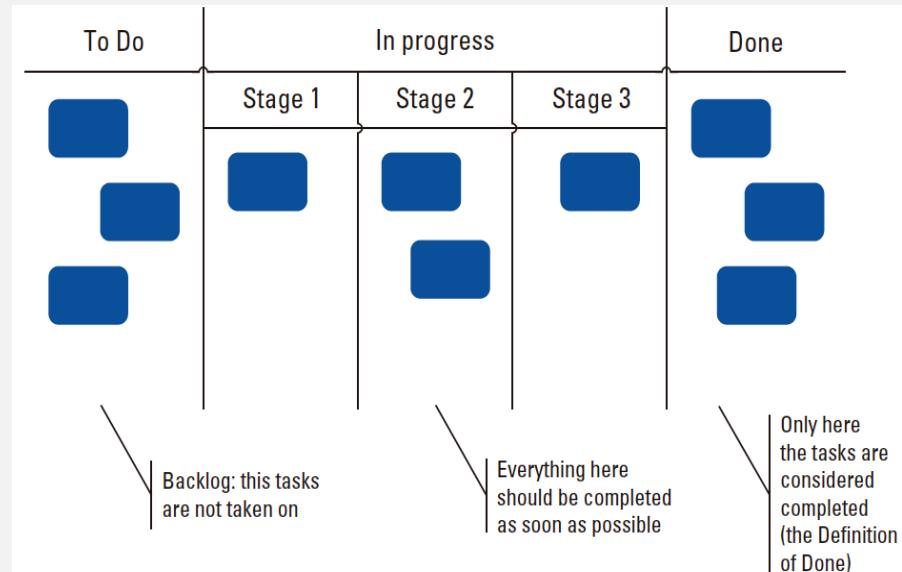
# Pratiques DevOps

## Visualiser le travail

### ➤ KANBAN

Un tableau Kanban permet de créer un « système pull » :

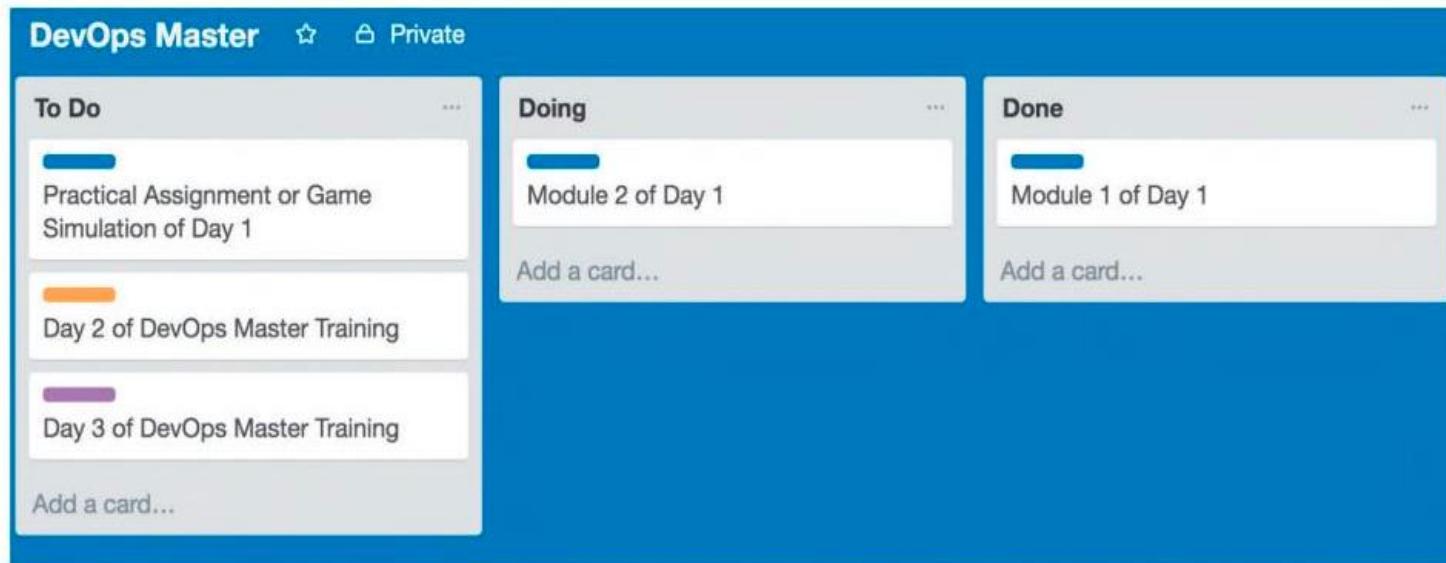
- Visualisez le flux de travail
- Mesurer et optimiser la durée du flux
- Réduit les temps d'arrêt
- Réduit le besoin de coordination
- Limiter les travaux en cours (WIP)



# Pratiques DevOps

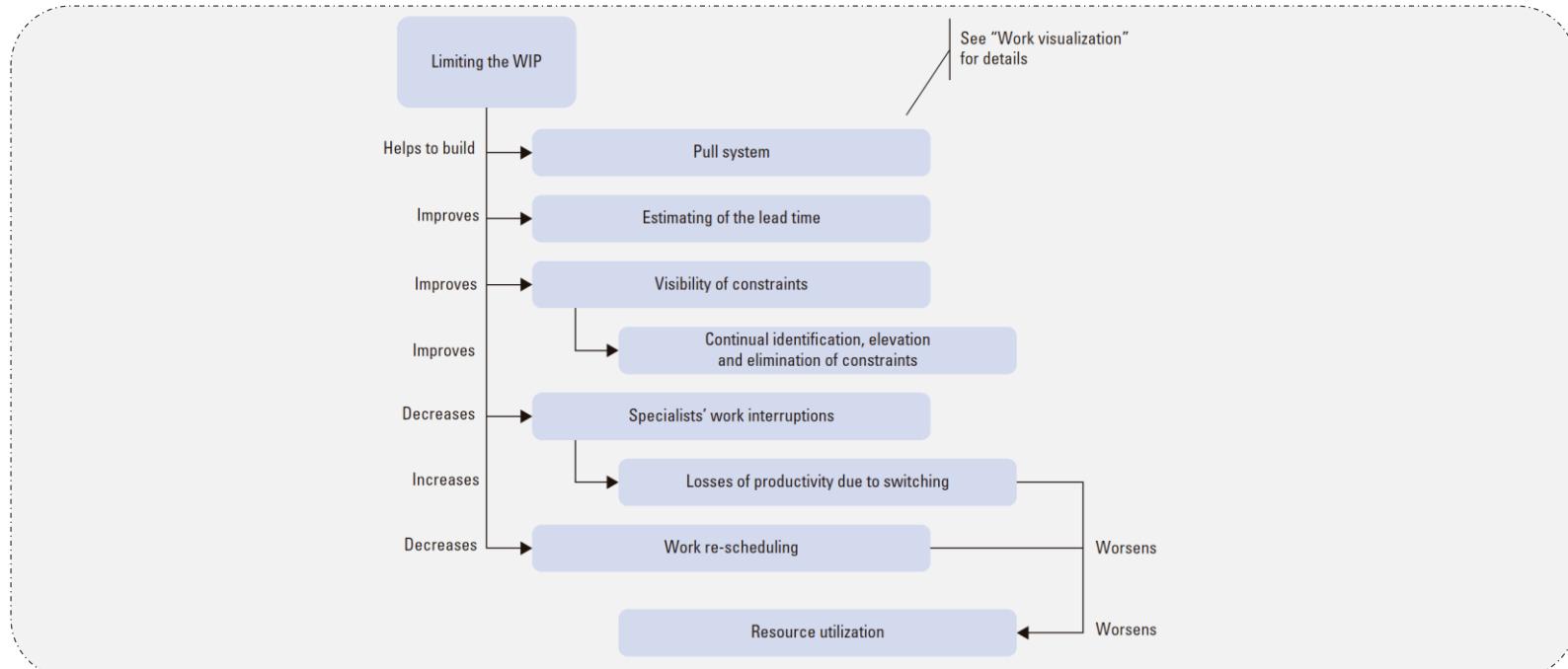
## Visualiser le travail

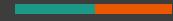
The Kanban board



# Pratiques DevOps

## Visualiser le travail





Small batch  
sizes



# Pratiques DevOps

## Small batch sizes

Réduire la taille des lots de versions est la deuxième façon d'améliorer le flux de travail way to improve the workflow (WIP).

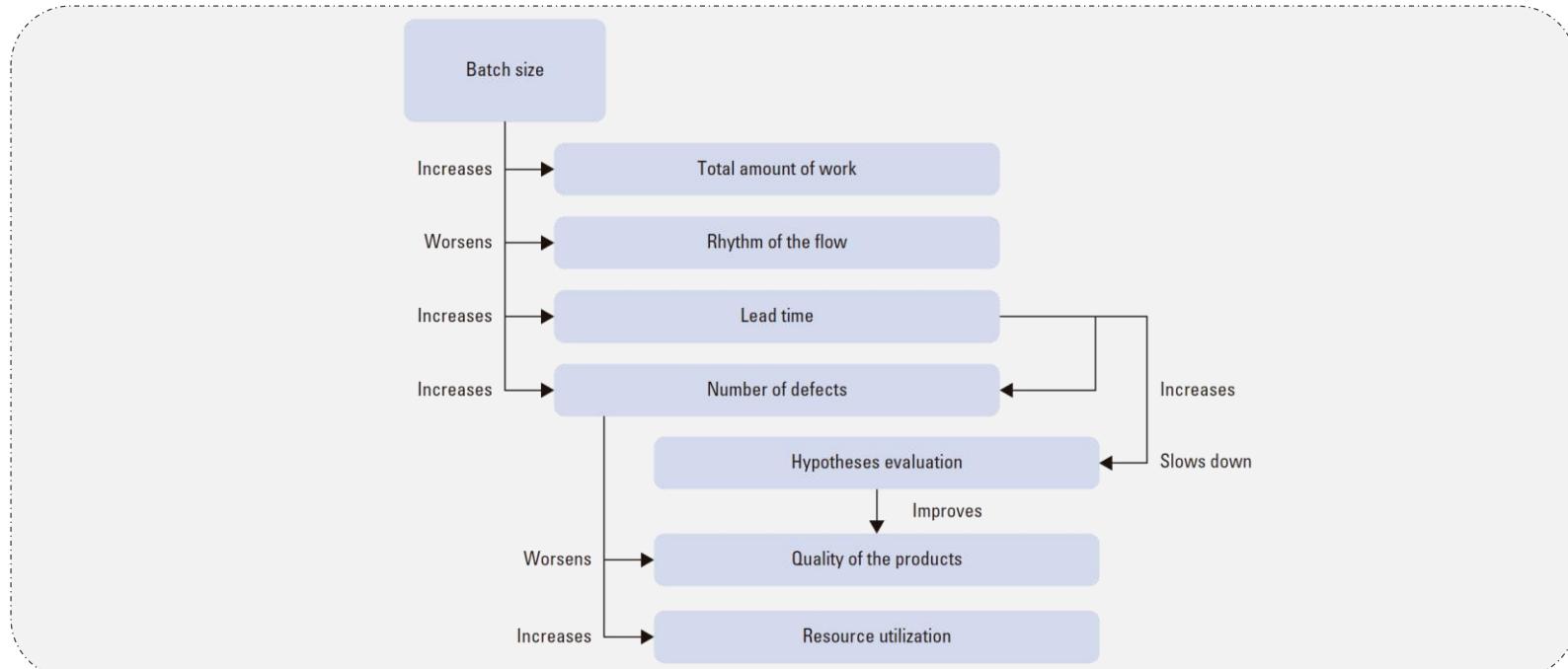
Les petits lots transitent dans le système plus rapidement et avec moins de variabilité, ce qui favorise un apprentissage et un déploiement plus rapides.

Cela signifie généralement accorder davantage d'attention et augmenter les investissements dans les infrastructures et l'automatisation.

Cela réduit également le coût de transaction de chaque lot.

# Pratiques DevOps

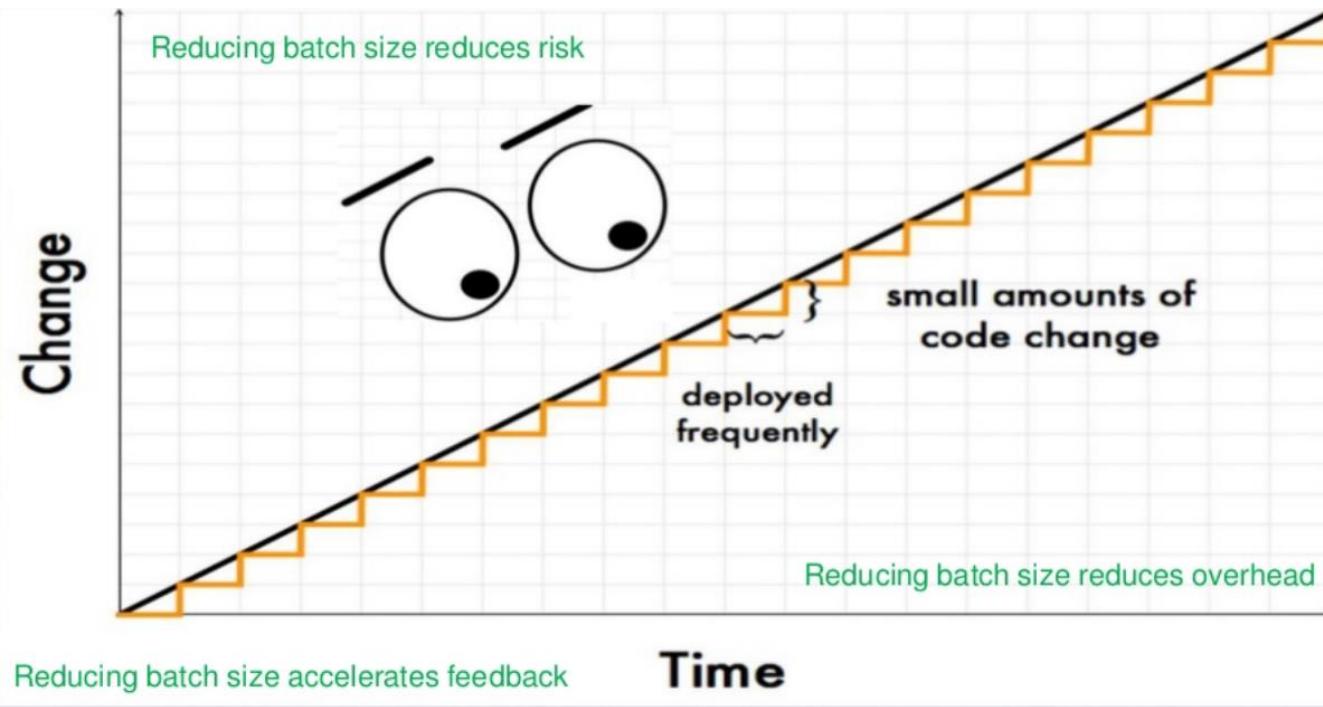
## Small batch sizes



# Pratiques DevOps

## Small batch sizes

Anti-Fragility :





Soutenir  
l'innovation

# Pratiques DevOps

## Soutenir l'innovation

- Encourage innovation

Some companies start by allocating a certain proportion of working time to improvement and innovation (20%).

A special budget allocated to the exploration of new technologies and the creation of new products and tools.

Select interesting ideas, invest limited resources in some of them.



# Dealing with Bottlenecks

# Pratiques DevOps

## Identify ways to deal with bottlenecks

### ➤ Faire face aux goulots d'étranglement

Un pipeline de déploiement efficace et fluide ne se réalise pas du jour au lendemain et nécessite des efforts considérables.

Grâce aux outils de visualisation et aux limites WIP, il est possible d'identifier les goulots d'étranglement dans la chaîne. De tous les goulots d'étranglement connus, il y en a un qui provoque le plus de décalage : c'est celui sur lequel il faut se concentrer.

Changer la méthode de travail afin de capitaliser sur le goulot d'étranglement et de l'exploiter au maximum.

Trouvez un moyen d'éliminer le goulot d'étranglement.



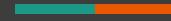
5

## Applicabilité et difficultés

# Applicabilité et difficultés

## Objectifs

- Caractériser la situation dans laquelle DevOps est réalisable
- Identifier les conditions qui rendent le DevOps intéressant pour l'entreprise
- Identifier le manque de préparation à adopter DevOps
- Caractériser l'infrastructure et l'architecture informatique monolithiques comme une limitation à l'adoption de DevOps
- Clarifier les risques et les solutions à l'utilisation des COTS dans les métiers stratégiques
- Caractériser la nécessité d'un état d'esprit flexible face au changement et à l'innovation
- Rappelez-vous que DevOps peut démarrer petit et peut être développé à partir de là



# DevOps Application

# Application

## Applicabilité



- DevOps n'est pas...

DevOps n'est pas une solution magique à tous les « problèmes ».

Ne démarrez pas les pratiques DevOps si...

- L'entreprise ne participe qu'à une partie limitée du flux de valeur.
- L'entreprise ne considère pas la technologie comme un élément clé de son activité.

# Application

## Applicabilité



- Le DevOps devrait intéresser les entreprises lorsque les conditions suivantes sont remplies :
  - **Toutes les autres méthodes adoptées** pour accroître l'efficacité **ne donnent plus de résultats significatifs**.
  - Son activité principale est fortement **dépendante** des technologies de l'information.
  - La technologie de l'information utilisée par cette organisation connaît un **taux de changement élevé**.
  - L'activité principale (business) nécessite **des changements rapides** pour tester de nouvelles idées ou hypothèses : pour rester compétitif

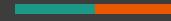
# Application

## Applicabilité



### ➤ Un investissement

- Entreprises qui souhaitent réduire considérablement leur dette technique ou éliminer la fragilité de leur infrastructure informatique :
- Le temps et les efforts nécessaires pour obtenir des avantages significatifs et continus du DevOps sont très importants et doivent être considérés comme un investissement.



# Limitations

# Application

## Limitations



- DevOps n'est pas très adapté pour :
  - Entreprises qui ne disposent pas de leur **propre développement de logiciels**.
  - Organisations qui utilisent leur propre logiciel, **dont les développeurs ne sont pas des membres du personnel**.
  - Entreprises qui fonctionnent depuis longtemps et **ne sont pas ouvertes à des restructurations majeures**.

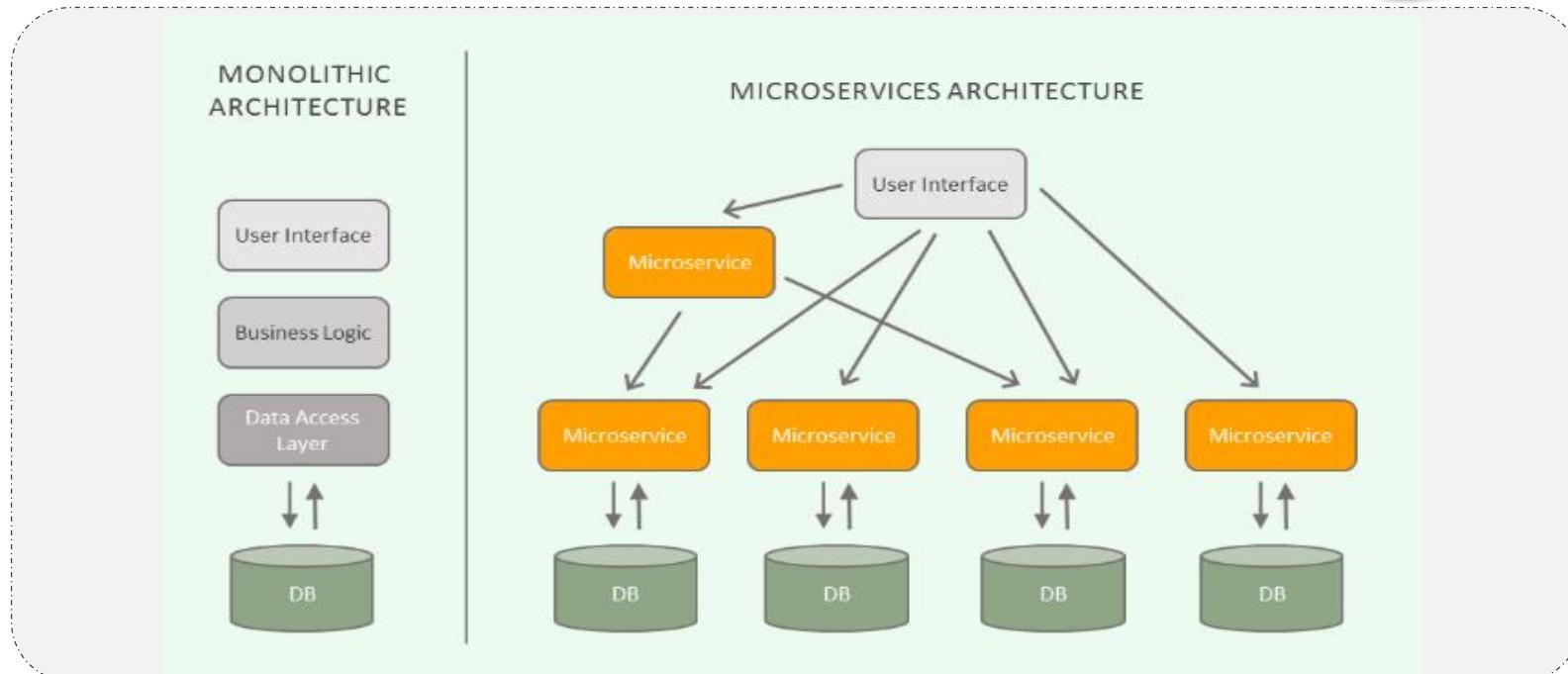
# Application

## Limitations

- Limites de l'infrastructure informatique et de l'architecture
- Organisations avec une architecture informatique monolithique.
- La mise en place de petites équipes nécessite la capacité d'attribuer un domaine de responsabilité distinct à chacune.



# Application Limitations



COTS



# Application

## COTS

- The risks of a COTS
  - Evolution de la compétition :
    - Il faut avoir un maximum de flexibilité et de contrôle, généralement inaccessible avec les logiciels commerciaux.
    - Le premier conseil que tout expert sérieux vous donnera est de vous débarrasser des COTS actifs dans les domaines les plus importants de votre entreprise.
- => Passer au développement de logiciels en interne.

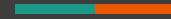


# Application

## COTS

- Quand il n'y a pas d'autre option
  - Étudiez le processus d'installation en détail, comprenez ce que fait le programme d'installation.
  - Créez ensuite votre propre script, en reproduisant le travail du programme d'installation d'origine.





# Architecture et modèles organisationnels en évolution

# Architecture et modèles organisationnels en évolution

## Organisation en évolution

### ➤ les difficultés d'un service informatique traditionnel :

- Développement (Dev) et opérations (Ops) sont séparés de manière naturelle :
- OPS et le support n'entrent pas dans les détails complexes de l'architecture du système et doivent donc poser les questions les plus simples aux développeurs.



# Architecture et modèles organisationnels en évolution

## Organisation en évolution

### ➤ Les difficultés d'un service informatique :

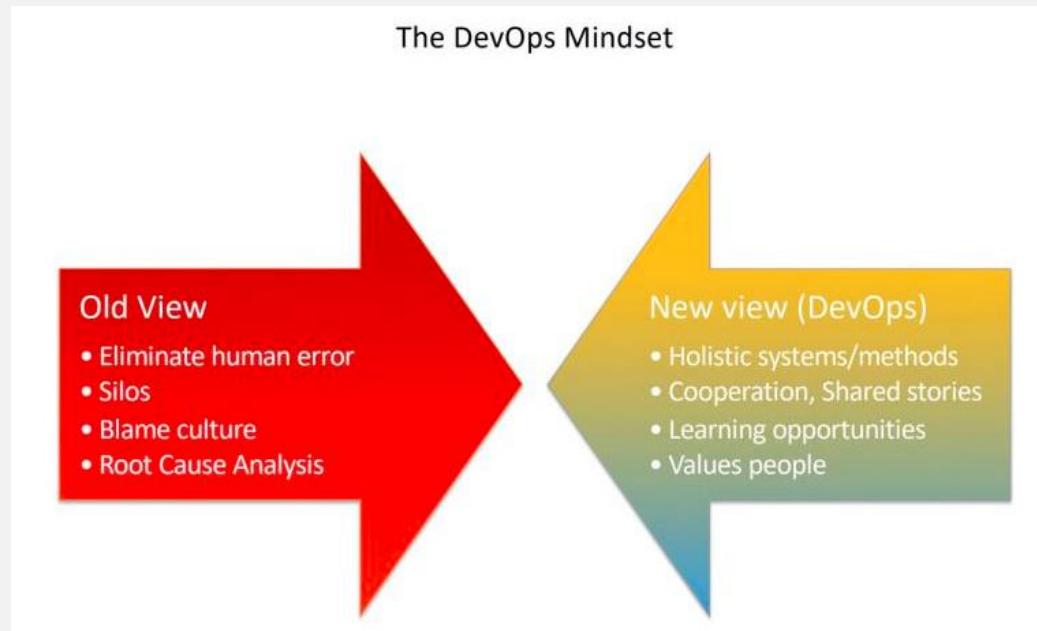
- Toute documentation sur le système informatique devient vite obsolète.
- Très peu d'employés comprennent le système informatique de manière globale, avec toutes ses dépendances et contraintes ; ces employés deviennent rapidement très précieux, irremplaçables et surchargés.
- De nombreux développeurs travaillent simultanément sur les mêmes fonctionnalités, chacun de leur côté, ce qui nécessite des ressources pour la coordination.



# Architecture et modèles organisationnels en évolution

## Organisation en évolution

### ➤ La nécessité d'un changement de mentalité



# Architecture et modèles organisationnels en évolution

## Organisation en évolution

### ➤ La nécessité d'un changement de mentalité

#### Fixed mindsets

- Talents and abilities are innate
- Smart or not smart
- Avoid failures
- Very competitive
- Comfort zone



#### Growth mindsets

- Talents and abilities are learned and improved
- No fear to fail
- Learn from mistakes
- Like challenges
- Skills can be improved

A growth mindset, allow people and organizations to learn and adapt more quickly.

# Architecture et modèles organisationnels en évolution

## Organisation en évolution

- La nécessité d'un changement de mentalité

Feature	Traditional corporate culture	Startup culture
Management style	Command, authoritarian	Autonomous
Attitude to change	Conservatism	Experiments
Organizational structure	Functional hierarchy	Network
Focus on outputs	Project-oriented	Product-oriented
Change model	Waterfall	Agile, iterative
System architecture	Monolithic, carefully designed	Loosely coupled, microservice
Technology preferences	Patented, proprietary	Open source

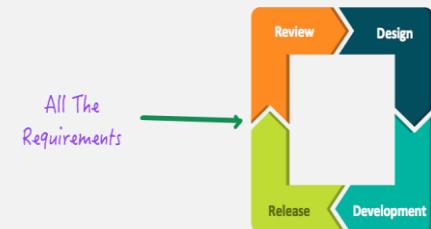


# Progression itérative

# Application

## Progression itérative

- Identifiez les systèmes qui sont faiblement connectés aux autres.
- Utilisez ces systèmes comme pilote pour appliquer les bases du DevOps :
  - Flux de valeur
  - Pipeline de déploiement
  - Système de contrôle de versions
  - Gestion automatisée des configurations
- Appliquer cette expérience à d'autres systèmes.
- En commençant par des cas plus simples, vous pouvez passer à des cas plus complexes





# Application

## Experimenting

“Always take on new challenges—even if you are not sure you are completely ready.”

Sheryl Sandberg,  
COO of Facebook

# Application

## Accepter l'échec

- Prepare for failure

Dans le contexte du DevOps, l'échec n'est pas une infraction. Les équipes supposent que quelque chose finira par mal tourner, elles font donc tout leur possible pour trouver les bugs le plus tôt possible et les corriger.

- Why ?

**Because continual improvement goes hand in hand with failures.**

# Culture DevOps

## Où commencer

### ➤ OPS culture

- Automatiser les environnements
- Surveillance ouverte
- Autoriser et superviser les accès en production
- Partager la responsabilité
- Assister aux réunions de développement
- Développer des tests d'infrastructure
- Fournir une API de provisionnement
- Configurations du système de gestion des versions



# DevOps Culture

## Où commencer

### ➤ DEV culture.

- Assurer la surveillance
- Accompagner/réaliser les installations
- Partager la responsabilité
- Fournir un packaging natif
- Développer un tableau de bord des évolutions
- Apportez votre voix au CAB (Change Advisory Board)
- Configurer l'agrégation des journaux
- Automatiser, Automatiser, Automatiser



8

Exam

# Exam details

## ABOUT EXIN

Published and designed by EXIN. EXIN is the global independent certification institute for professionals in the IT domain. With more than 30 years of experience in certifying the competences of over 2 million IT professionals, EXIN is the leading and trusted authority in the IT market. With over 1000 accredited partners EXIN facilitates exams and e-competence assessments in more than 165 countries and 20 languages. EXIN is co-initiator of the e-Competence Framework, which was set up to provide unambiguous ICT certification measurement principles within Europe and beyond.

## EXIN. THE MOST RENOWNED EXAMINATION INSTITUTE FOR IT PROFESSIONALS



- Complete portfolio of Information Management
- Exams in 165 countries
- Exams in 20 languages
- 2 million EXIN-certified professionals
- International network of accredited partners