



DevOps Foundation



Introduction

 Exin DevOps Foundation

À qui s'adresse cette formation ?

Introduction

DevOps Foundation

- **Consultant IT :**
 - * **Ingénieur/administrateur système ou Cloud**
 - * **Développeur, testeur ou administrateur de base de données**
- **Manager IT ou Chef de projet**
- **Débutants : étudiant ou informaticien souhaitant découvrir le monde de DevOps**

Objectifs Pédagogiques

➤ Exin DevOps Foundation

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



SOMMAIRE

- **DevOps : Présentation Générale**
- **Les principes et les valeurs DevOps**
- **Les méthodes traditionnelles vs DevOps**
- **Les pratiques DevOps**
- **Application de DevOps**
- **DevOps et les outils**
- **Bonus**



Objectifs



En complétant cette section, vous serez capable de :

- 1. Décrire le développement historique du cascade à Agile et Scrum et à DevOps.**
- 2. Décrire les origines technologiques et méthodologiques de DevOps.**
- 3. Définir DevOps, ce qui n'est pas et ses bénéfices pour le business et l'IT.**



Introduction

 C'est quoi DevOps

QUESTION : WHAT IS DEVOPS ?



1

DevOps
Présentation
générale

DevOps - Présentation Générale

Problématique

Posons la problématique !

Aujourd'hui, nous connaissons une véritable fusion entre le business et l'IT.

Toute entreprise, quelque soit son secteur d'activité, s'appuie sur des services informatiques. La réponse à des besoins de plus en plus changeants et de plus en plus complexes dépend de sa capacité à produire des services IT de qualité de plus en plus rapidement

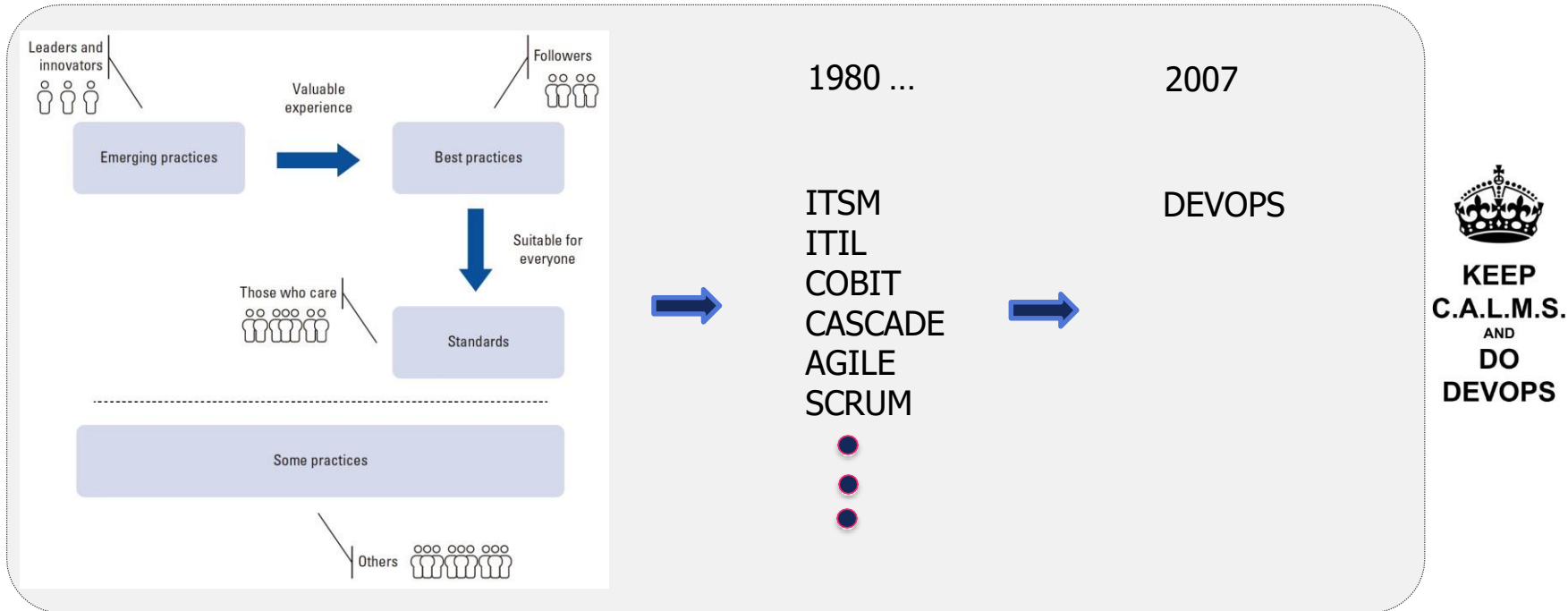
L'indicateur clé du business agile : le « Time to Market » !



Origine DevOps

DevOps - Présentation Générale

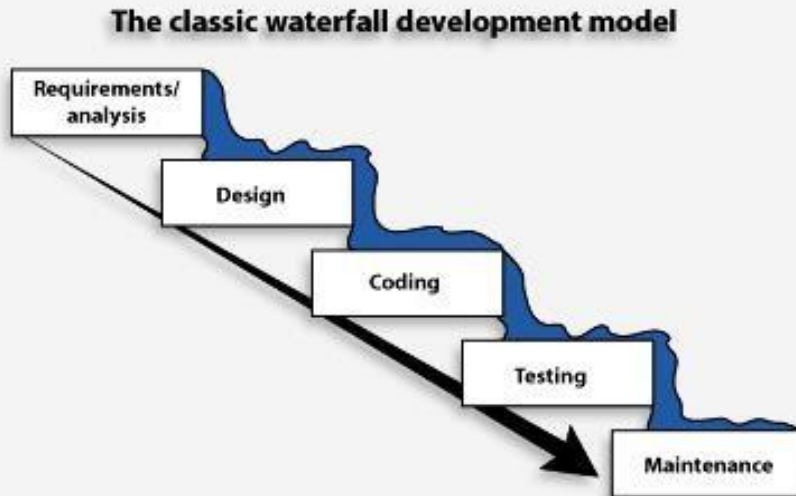
Origine DevOps



DevOps - Présentation Générale

— Du Cascade vers Agile vers Cloud Computing

➤ A - CASCADE



- Au 20ème siècle, Waterfall Model était la principale méthodologie de développement de logiciels.
- Dans les années 1990, Internet a accéléré le besoin de lancements rapides d'année en mois.

DevOps - Présentation Générale

Du Cascade vers Agile vers Cloud Computing

➤ B - AGILE

En 2001 '17' experts estimaient que le traditionnel (Cascade) ne correspondait plus aux contraintes et aux exigences des organisations en évolution rapide.

Les valeurs, principes Agile :

- Un logiciel fonctionnel
- Livraison en continue
- Collaboration entre le Business et IT
- Motivation des collaborateurs
- Agilité
- Simplicité
- Auto-organisation
- Excellence technique



Création du manifeste Agile

DevOps - Présentation Générale

— Du Cascade vers Agile vers Cloud Computing

➤ AGILE MANIFESTO



Les individus et leurs interaction

Plutôt que



Les processus et les outils



Un produit fonctionnel

Plutôt que



Une documentation exhaustive



La collaboration avec le client

Plutôt que



La négociation contractuelle



L'adaptation aux changement

Plutôt que



Le suivi d'un plan

DevOps - Présentation Générale

Origines de DevOps

➤ **LEAN SOFTWARE DEVELOPMENT :**

Les 7 types de déchets (Waste):

- 1) Defects or bugs (Defects in Lean Manufacturing)
- 2) Task switching (Transportation in Lean Manufacturing)
- 3) Extra features (Over-production in Lean Manufacturing)
- 4) Handoffs (Motion in Lean Manufacturing)
- 5) Delays (Waiting in Lean Manufacturing)
- 6) Partially done work (inventory in Lean Manufacturing)
- 7) Relearning (extra processing in Lean Manufacturing)

DevOps - Présentation Générale

Origines de DevOps

➤ **LEAN SOFTWARE DEVELOPMENT :**

Définitions :

- Le **Jidoka (autonomation)** est le premier pilier du Système de Production Toyota. C'est l'automatisation avec une touche humaine, ou comment automatiser les opérations simples et répétitives, tout en conservant un contrôle humain pour orchestrer tout cet outillage, et prendre en main les situations complexes.
- Le **J-I-T(Just in Time)** (J-àT juste à temps) est le second pilier du Système de Production Toyota. C'est la mise en flux pour livrer juste à temps (ni avant ni après) le produit demandé, dans la quantité demandée.
- Le **Kaizen (continuous improvment)**, consiste à tirer des leçons des problèmes, enrichir notre « système » et ceci à travers une démarche structurée et des outils d'analyse éprouvés (PDCA (Plan, Do, Check, Act), 5 pourquoi, diagramme de Pareto etc). On a tous envie de s'améliorer, le Kaizen nous aide à structurer et rendre efficace ces efforts d'amélioration.

DevOps - Présentation Générale

Du Cascade vers Agile vers Cloud Computing

➤ C - INFRASTRUCTURE AS A CODE

Virtualisation

- 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 - Présentation Générale

— Du Cascade vers Agile vers 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 - Présentation Générale

Du Cascade vers Agile vers Cloud Computing

➤ C - INFRASTRUCTURE AS A CODE

Cloud Computing

- Selon NIST (The US National Institute of Standards and Technology) Les caractéristiques essentiels du Cloud Computing sont :
 - On demand self-service
 - Broad network access
 - Resource pooling
 - Rapid elasticity or expansion
 - Measured service

DevOps - Présentation Générale

Du Cascade vers Agile vers Cloud Computing

Premièrement, Apparition de nouvelles méthodes d'interaction avec les clients finaux et de l'application de techniques de développement agiles.

+

Deuxièmement, Nouvelles technologies de gestion d'infrastructure:

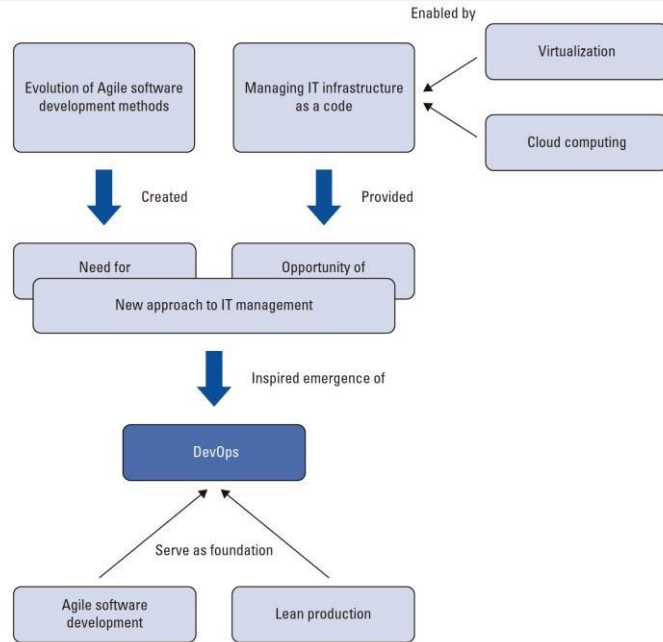
- Virtualisation
- Cloud computing

=

Il est devenu possible d'**organiser** le travail informatique différemment.

DevOps - Présentation Générale

Définition de DevOps



➤ DevOps :

- **Development** (les équipes en charges des projets / évolutions applicatives) + **Operations** (les équipes en charge du run et de l'exploitation)
- L'essence même de DevOps est de penser non seulement au développement logiciel, mais également à **l'ensemble de la chaîne de valeur.**



DevOps - Présentation Générale

Définition de DevOps

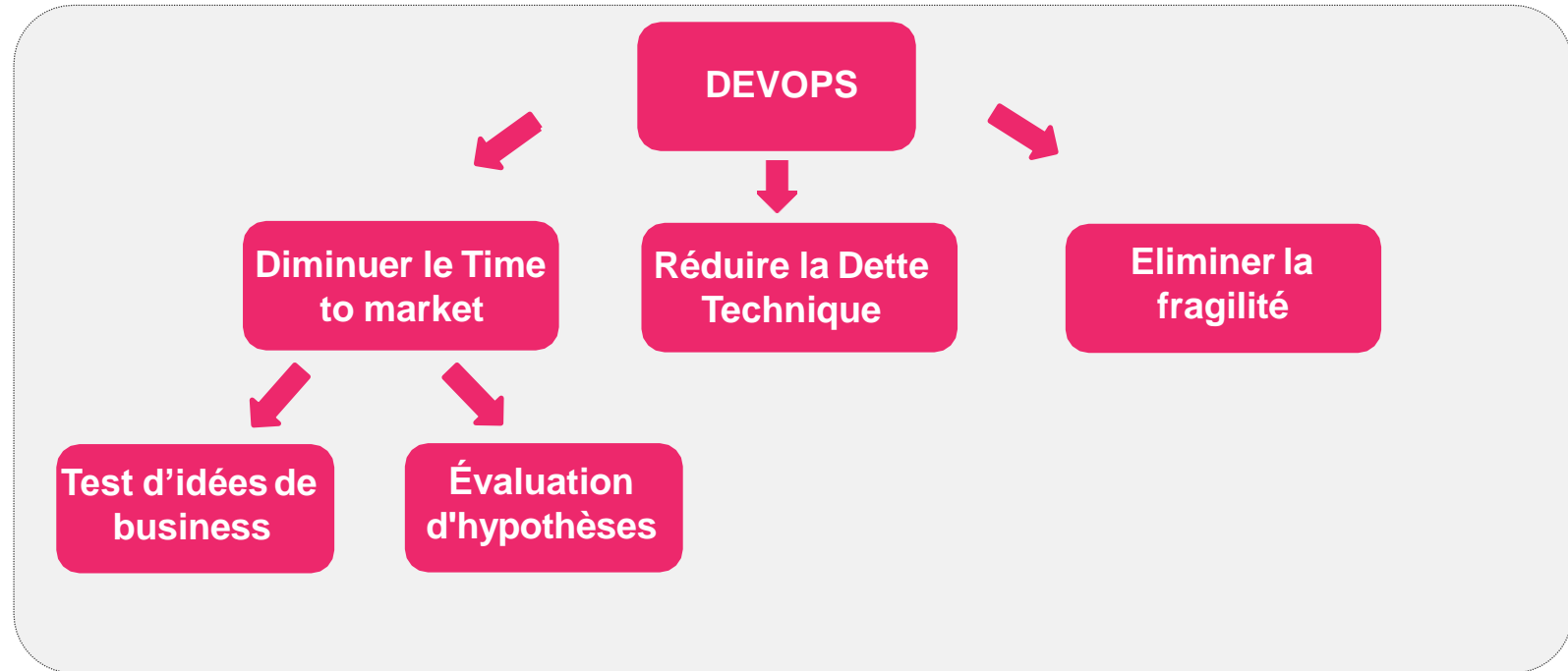
- En regroupant les différents termes de plusieurs définitions données sur le web, DevOps est :
 - Un mouvement de professionnels hérité de 2 philosophies
LEAN SOFTWARE DEVELOPPEMENT
AGILE
 - Une culture ;
 - Un processus agile sur l'ensemble de la chaîne (du développement à la mise en production)
 - Une nouvelle approche technique
 - Une autre approche humaine
 - Une réponse à des nouvelles problématiques comme le déploiement massif et le déploiement régulier.



Bénéfices DevOps

DevOps - Présentation Générale

Bénéfices



DevOps - Présentation Générale

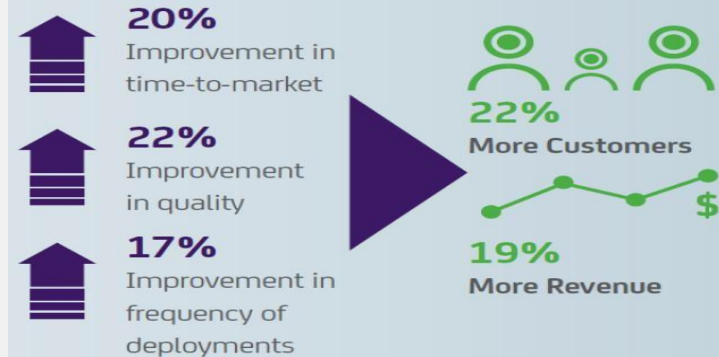
Bénéfices : Time To Market

Diminuer le Time To Market :

- Réduction de la taille du produit.
- Réduction des pertes dans la chaîne de la mise en production d'un changement.
- Amélioration en continue.
- Elimination des pertes :
 - Vérification des builds
 - Vérification tôt de la qualité
- ...
- Equipes 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 : Etude CA "What smart businesses know about devops". Panel : 1300 décideurs IT répartis dans 21 pays
Disponible sur <http://aka.ms/devopscs>

DevOps - Présentation Générale

— Bénéfices : ~~Det~~te Technique

Réduire la dette technique

*La dette survient lorsqu'un membre de **l'équipe choisit un moyen non optimal de résoudre un problème** afin de réduire le temps de développement. les solutions non optimales accumulées entraînent une détérioration progressive du produit, en conséquence, une dégradation de la qualité du produit..*

Un non-respect de la conception, intentionnel ou non, induit des coûts supplémentaires dans le futur. Ce sont les **intérêts**. C'est pourquoi l'on parle de dette technique, pour montrer l'analogie avec la dette dans les finances des entreprises. Cela sous entend qu'il vaut mieux rembourser la dette un jour plutôt que de continuer à payer sans cesse des intérêts.

Quand on code au plus vite et de manière non optimale, on contracte une dette technique que l'on rembourse tout au long de la vie du produit sous forme de temps de développement de plus en plus long et de bugs de plus en plus fréquents.

DevOps - Présentation Générale

Bénéfices : ~~Dette Technique~~

Réduire la dette technique

L'élimination de la dettes techniques créés précédemment (consciemment ou accidentellement) est planifiée au même titre que la création de nouvelles fonctionnalités.

DevOps recommande vivement d'utiliser la pratique «faire face aux problèmes aussi souvent que possible» afin de prévenir une "stagnation" de problèmes que tout le monde connaît, mais que personne ne peut mettre la main là-dessus.

DevOps - Présentation Générale

— Bénéfices :Fragilité

Éliminer la fragilité

*Ironiquement, les systèmes les plus importants et les plus rentables pour les entreprises sont les plus fragiles. Réduire la fragilité de ces systèmes est extrêmement difficile en raison des risques élevés de **perturbation de l'activité**, de la **tolérance zéro aux temps d'immobilisation** 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 rétabli et le système continue de fonctionner correctement.

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

DevOps - Présentation Générale

— Bénéfices : ~~Fragilité~~

Éliminer la fragilité

« L'anti-fragilité est une propriété des systèmes qui se renforcent lorsqu'ils sont exposés à des facteurs de stress, des chocs, de la volatilité, du bruit, des erreurs, des fautes, des attaques ou des échecs. »

Netflix Chaos Monkey





Mythe DevOps

DevOps - Présentation Générale

Idées fausses sur DevOps

Mythe ou Réalité: "DevOps n'est rien de plus que la continuation des idées Agiles"

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.

Obtenir le bénéfices de DevOps **nécessite des changements culturels** plus importants dans l'entreprise que l'adoption d'Agile.

Les **objectifs** fixés pour DevOps ne se limitent pas à accélérer la livraison: il est également nécessaire de réduire la dette technique et d'éliminer la fragilité.

DevOps - Présentation Générale

Idées fausses sur DevOps

Mythe ou Réalité: "DevOps est un superman capable de coder, de créer des tests, de déployer des environnements et de gérer l'infrastructure."

DevOps est un changement profond dans les services informatique, 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 logiciel ne garantit pas le succès. DevOps ne se limite pas à l'utilisation des outils. Il s'agit des changements organisationnels profonds qui peuvent être appuyés par ces outils.

DevOps - Présentation Générale

Idées fausses sur DevOps

➤ DEVOPS n'est

Ni un produit

Ni un référentiel

Ni une méthode

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

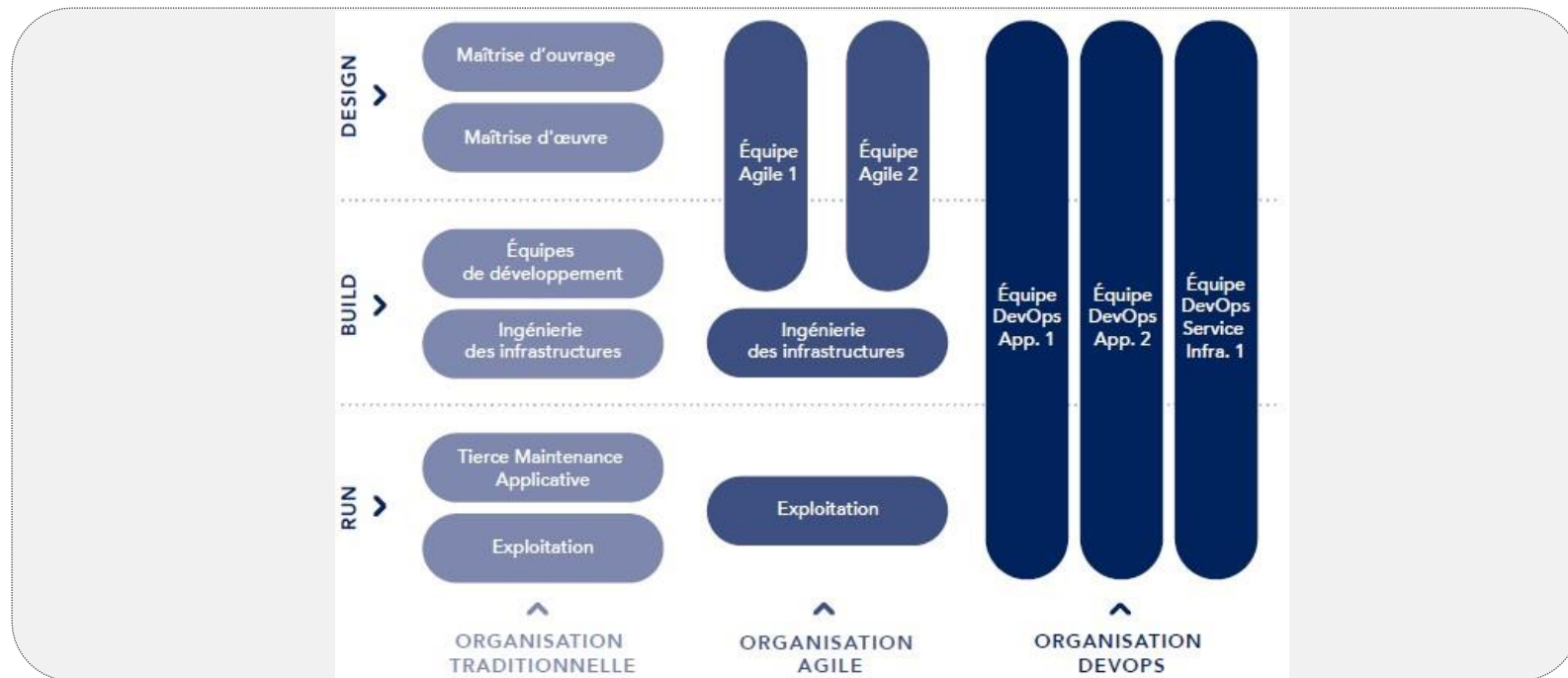
- Pas de consensus clair : le fondateur n'a pas défini DEVOPS (Patrick DEBOIS)



Organisation
DevOps

DevOps - Présentation Générale

Evolution de l'organisation





Définition de DevOps

DevOps - Présentation Générale

— Evolution de l'organisation

DevOps

est une évolution des idées de **la méthodologie de développement logiciel Agile** et de **Lean Manufacturing** appliquée à **la chaîne de valeur IT de bout en bout**,

qui permet aux entreprises de **réaliser des bénéfices** avec **les technologies de l'information modernes**

en s'appuyant sur des changements **culturels, organisationnels et techniques.**



2

Les principes
et les valeurs
DevOps

Les principes et les valeurs DevOps

— Les principes DevOps

➤ Les principes clés.

- Cartographie de la chaine de valeur
- Pipeline de déploiement
- Contrôle de versions.
- Gestion de la configuration
- Définition de terminé

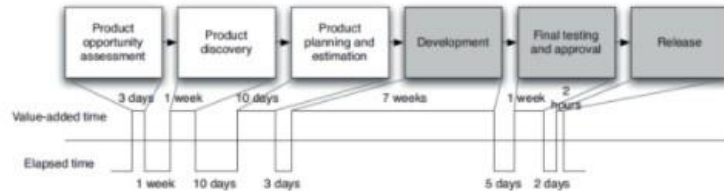


Les principes et les valeurs DevOps

Les principes DevOps

DevOps value to ALM

- Aims to deliver usable value to the business faster and often (JKK – Ji-Koutei-Kanketsu)
- Designs a value stream map of the whole process from business needs to end of life

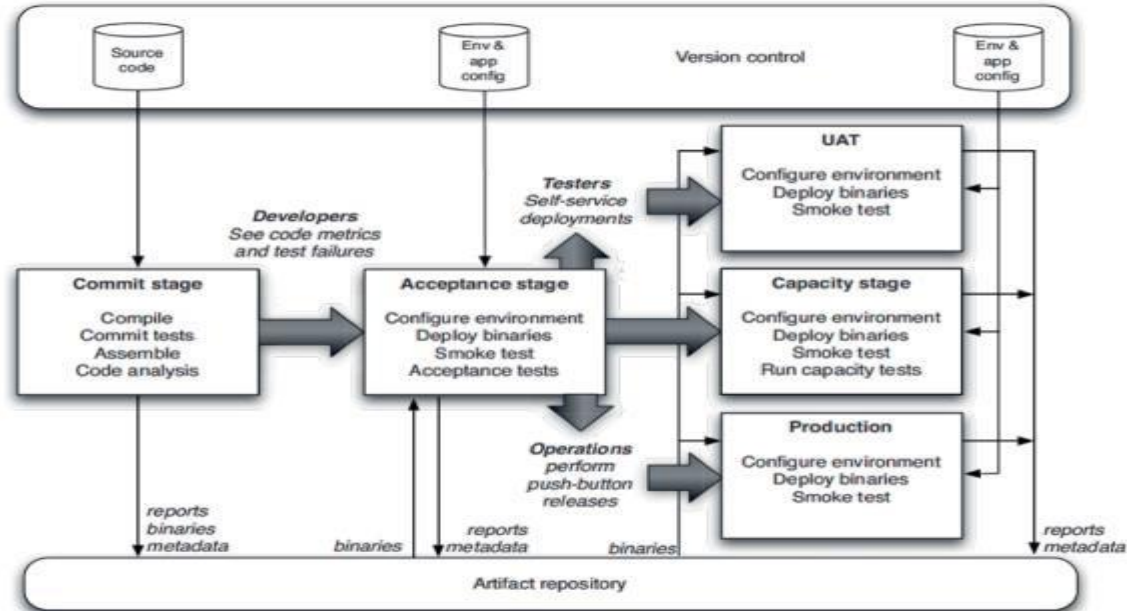


- Creates a continuous automated process for the build, deploy, test and release production-ready code (Continuous Delivery)
- Fosters Development and Operations teams collaboration and accountability thru all stages of the value stream (Definition of Done)

Les principes et les valeurs DevOps

Les principes DevOps

The Deployment Pipeline



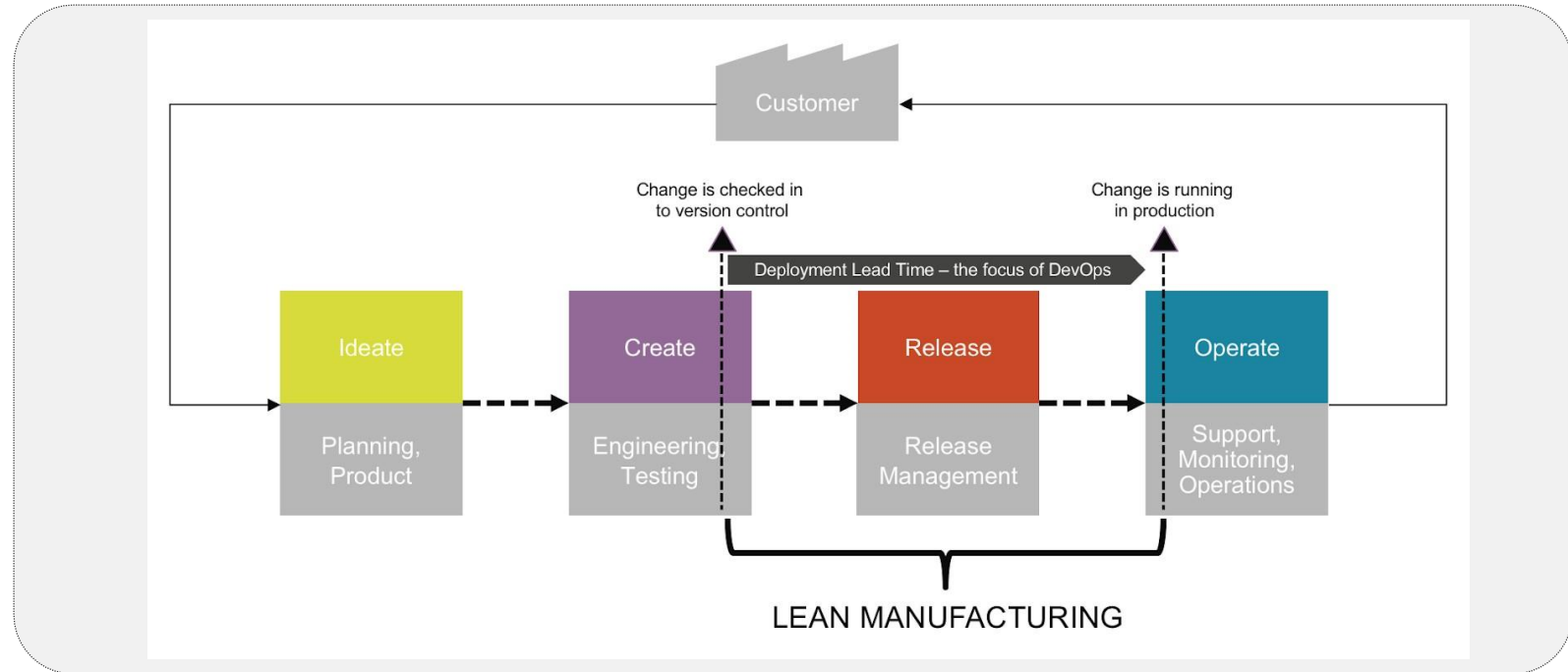


Chaine de
valeur



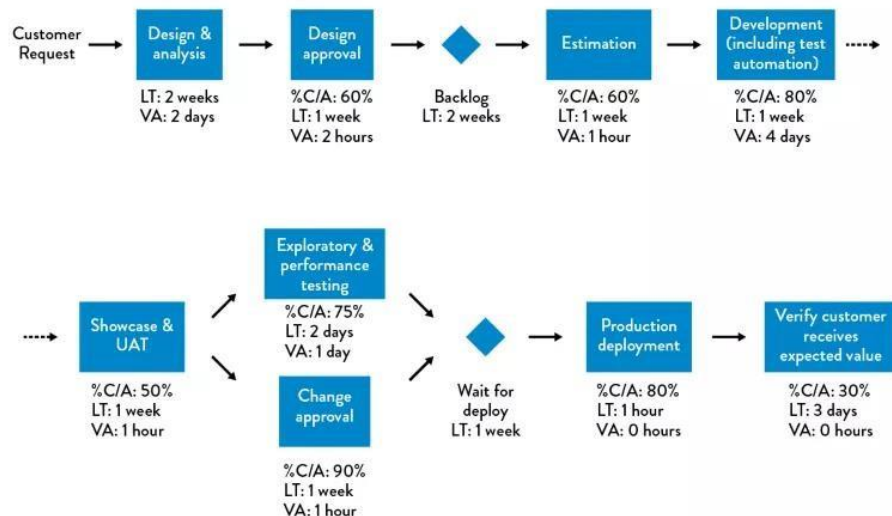
Les principes et les valeurs DevOps

Cartographie des chaines de valeur



Les principes et les valeurs DevOps

Cartographie des chaines de valeur



Aggregate values:
Total lead time: 10 weeks
Value added time: 7.5 days
Percent complete and accurate: 8.6%

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

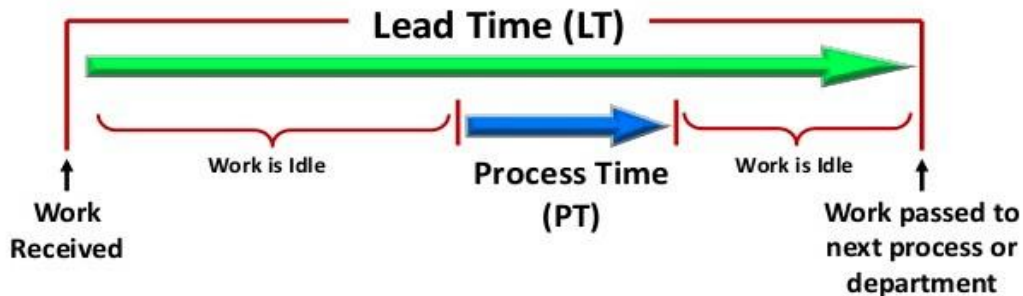
Ce pourcentage est obtenu en demandant aux étapes en aval le pourcentage du temps où ils reçoivent un travail «utilisable tel quel est», sans avoir besoin de clarifier, d'ajouter, ou corriger les informations reçues à l'étape précédente. Seule l'étape en aval peut évaluer le %C/A de l'étape précédente.

Les principes et les valeurs DevOps

Cartographie des chaines de valeur

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

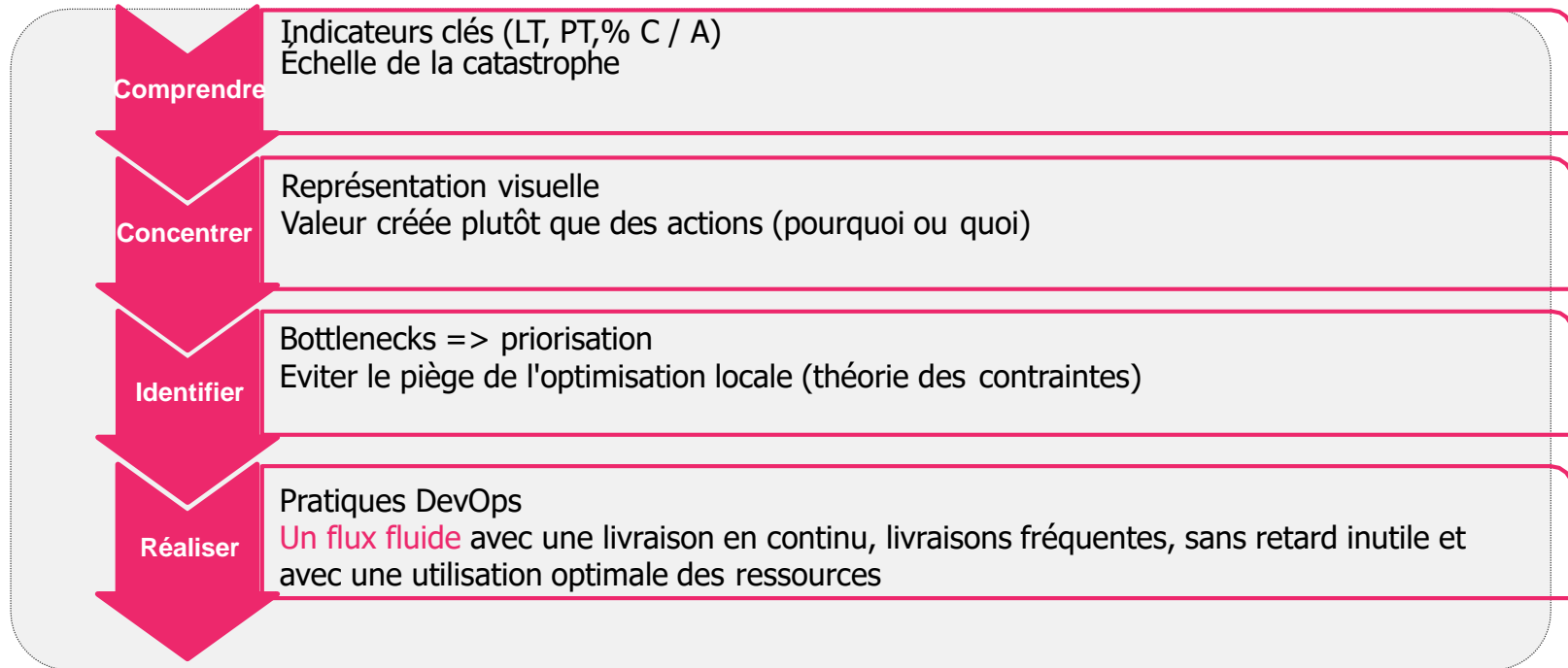


Lead Time = Elapsed time; Throughput time; turnaround time

Process Time = Touch time; work time; cycle time

Les principes et les valeurs DevOps

— Cartographie des chaines de valeur

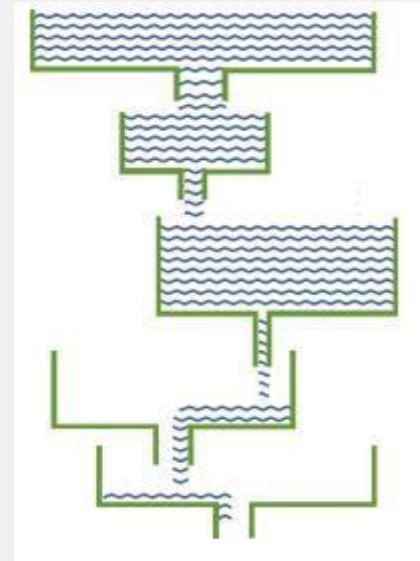


Les principes et les valeurs DevOps

Cartographie des chaines de valeur

- Théorie des contraintes. (ToC de l'anglais Theory Of Constraints)

« Puisque la force d'une chaîne est déterminée par le maillon le plus faible, la première étape pour améliorer une organisation doit être d'identifier le maillon le plus faible. » Eliyahu Goldratt



Les principes et les valeurs DevOps

Cartographie des chaines de valeur

➤ Théorie des contraintes.

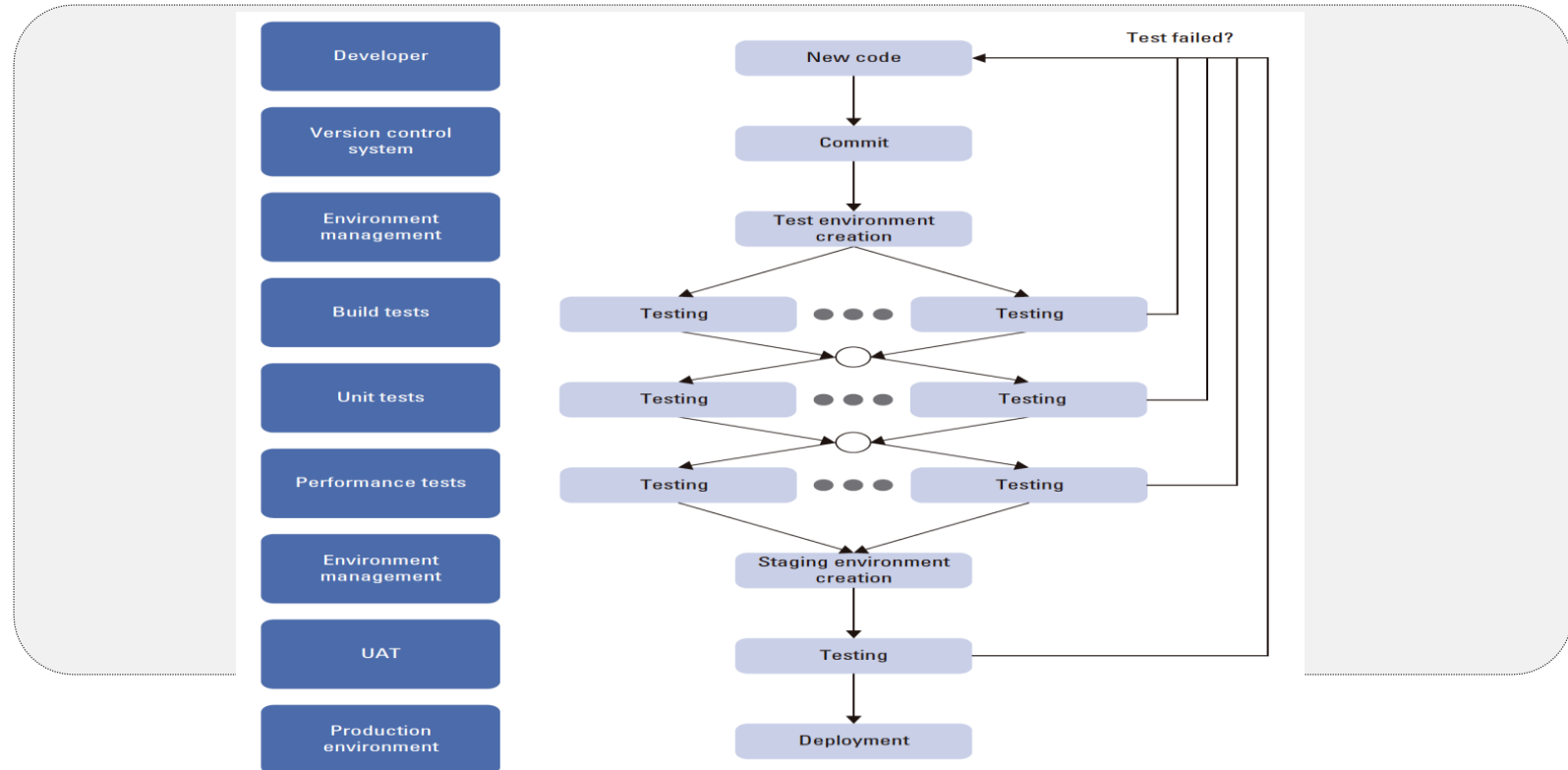
- La Théorie des Contraintes propose un processus itératif d'amélioration en cinq étapes, qui vise à focaliser les efforts sur la seule contrainte.
- Les cinq étapes sont les suivantes :
 - Identifier la contrainte (le goulot).
 - Exploiter la contrainte ; améliorer l'utilisation de sa capacité.
 - Subordonner tous les processus à la contrainte.
 - Augmenter la capacité de la contrainte si cela est pertinent.
 - Recommencer à l'étape 1 si la contrainte a changé.



Pipeline de Déploiement

Les principes et les valeurs DevOps

Pipeline de déploiement



Les principes et les valeurs DevOps

Pipeline de déploiement

➤ Les trois processus DevOps

L'intégration continue : (ou Continuous Integration) est un processus orienté études consistant à compiler, tester et déployer sur un environnement d'intégration. Le but est de tester aussi souvent et autant que possible les non-régressions du livrable pour détecter les bugs le plus tôt possible. La plupart du travail est réalisé par des outils de test. Le déploiement sur la plateforme d'intégration devient simple et peut être réalisé par les études sans faire intervenir l'exploitation.

Les principes et les valeurs DevOps

Pipeline de déploiement

➤ Les trois processus DevOps

La livraison continue : (ou Continuous Delivery) est un processus d'intégration et de production. Le but est tester et livrer une application à chaque étape de son cycle de vie (recette, pré-production, répétition, production). Cette étape est réalisée après validation des tests effectués en intégration. La phase de test correspond aux tests fonctionnels du livrable. Le passage d'un état à l'autre est entièrement automatisé, c'est pourquoi le livrable doit être constitué de tel sorte qu'il soit déployable en production dès la mise en recette.

Les principes et les valeurs DevOps

Pipeline de déploiement

➤ Les trois processus DevOps

Le déploiement continu : (ou Continuous Deployment) est un processus de production. Le but est de compiler, tester et déployer une application en production. Le « Continuous Deployment » nécessite que les processus de Continuous Integration et de Continuous Delivery aient été réalisés avec succès. Le déploiement est réalisé par un simple « press button ». Après le déploiement, il doit être possible de mesurer les éventuels impacts à l'aide d'outils de mesure de la performance et d'outils de supervision. En cas de problème, un processus automatisé de retour arrière peut être exécuté.

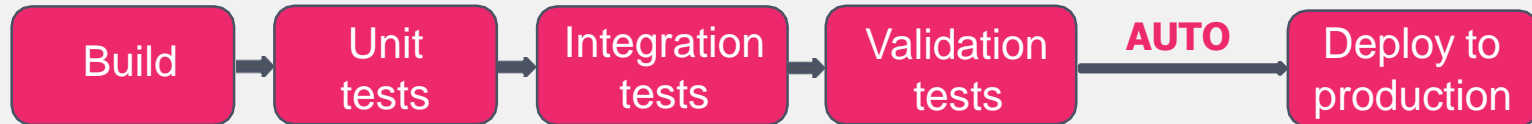
Les principes et les valeurs DevOps

Pipeline de déploiement

Continuous Delivery



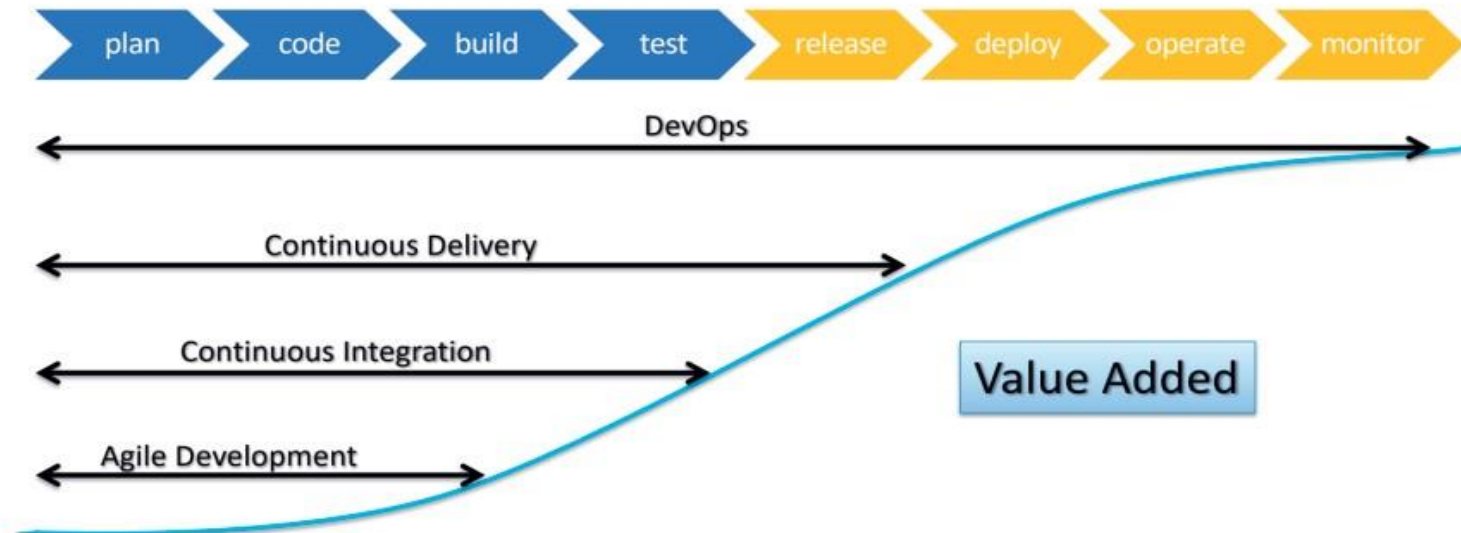
Continuous Deployment



Les principes et les valeurs DevOps

Pipeline de déploiement

Continuous Delivery vs Continuous Integration



Les principes et les valeurs DevOps

Pipeline de déploiement

- **Un processus moderne utilisant un pipeline parallèle**
 - Permet de produire des résultats beaucoup plus rapides.
- **Économise des ressources**
 - En ne commençant pas les prochaines étapes avant la fin des précédentes.
- **Assure la qualité du produit**
 - Les modifications qui ne fonctionnent pas bien n'atteignent pas la production et le système est toujours en état de fonctionnement.
- **Accélère la livraison des modifications dans l'environnement de production**
 - En maximisant l'automatisation de chaque étape.
- **Laisse constamment des enregistrements dans les journaux d'audit**

Les principes et les valeurs DevOps

Pipeline de déploiement

- Les challenges face à l'implémentation d'un pipeline de déploiement
 1. Pas suffisamment de tests prédéveloppés pour assurer le bon fonctionnement du pipeline.
 2. Tout automatisé.
 3. Une grande quantité de travail pour des petits changements



Contrôle de versions

Les principes et les valeurs DevOps

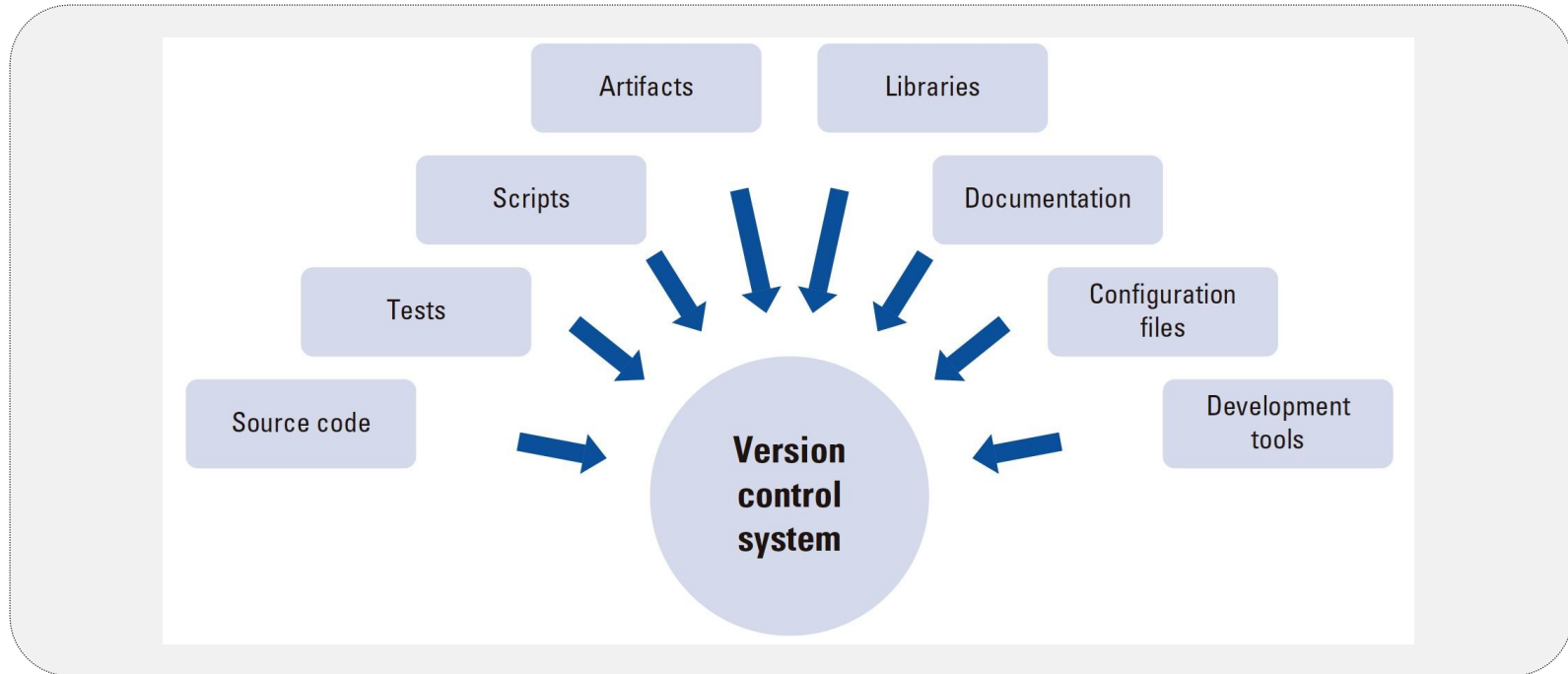
Contrôle de versions

Il s'agit de stocker non seulement le code source, mais également tout ce qui concerne le système informatique:

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

Les principes et les valeurs DevOps

Contrôle de versions



Les principes et les valeurs DevOps

Contrôle de versions

➤ **Les bénéfices**

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



Gestion de Configuration

Les principes et les valeurs DevOps

Gestion de la configuration

➤ Les bénéfices

DevOps **restructure** complètement la gestion de l'environnement de production (ainsi que de 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 d'environnements est effectuée automatiquement lorsque le pipeline de déploiement est en cours d'exécution.

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

Les principes et les valeurs DevOps

Gestion de la configuration

➤ Les bénéfices

DevOps **restructure** complètement la gestion de l'environnement de production (ainsi que de 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 d'environnements est effectuée automatiquement lorsque le pipeline de déploiement est en cours d'exécution.

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

Les principes et les valeurs DevOps

Gestion de la configuration

➤ Avantages

La gestion de la configuration de DevOps offre les mêmes avantages que ceux obtenus avec le contrôle total de version, mais les **principaux bénéficiaires sont désormais les personnes travaillant dans Operations.**

Maintenant que toutes les modifications sont contrôlées, le système peut être rapidement restauré à l'état stable. Si les membres clés quittent les lieux, les connaissances ne sont pas perdues, etc.



Définition de Terminé

Les principes et les valeurs DevOps

— Définition de terminé


➤ Définition de terminé

Ce n'est pas lorsque quelqu'un a fait sa part du travail, mais lorsque **(QUAND)** le client a reçu ou a commencé à recevoir la valeur qu'il attendait.

Cela signifie que l'intégralité du flux de valeur a été entièrement suivie jusqu'à **l'environnement de production**; alors seulement, le travail sera considéré comme terminé.

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





3 Traditionnelles vs DevOps

Traditionnelles vs DevOps

Amélioration continu

➤ Des livraisons fréquentes

Pratiques Traditionnelles	Pratiques DevOps
Livraisons de grande taille	Livraisons de petite taille
Planification des livraisons	Livraisons continues
Plusieurs jours / semaines	Communiqués hebdomadaires / quotidiens
Beaucoup de ressources	Utilisation efficace des ressources
Effort élevé	Effort régulier
Des sauvegardes	Automatisation
Documentation	Continu
Manuel	

Traditionnelles vs DevOps

Amélioration continu

➤ Les fonctionnalité les plus importantes

Pratiques Traditionnelles	Pratiques DevOps
<ul style="list-style-type: none">- Livraison-Une version est un groupe de nouvelles fonctionnalités déployées conjointement dans l'environnement de production.- Calendrier de livraison- Décision IT	<ul style="list-style-type: none">- Déploiement-Mise à disposition partielle ou complète d'une nouvelle fonctionnalité pour les utilisateurs-Déployé dès que les tests sont réussis- Décision business

Traditionnelles vs DevOps



Amélioration continu

➤ Automatisation de la pipeline

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

Ces environnements sont automatiquement libérée après utilisation.

Le fonctionnement rapide du pipeline nécessite une automatisation maximale des tests

Le déploiement et la distribution, dernières étapes du pipeline, sont également effectués automatiquement.

Traditionnelles vs DevOps



Amélioration continu

➤ Amélioration continu

Tous les défauts identifiés doivent être éliminés immédiatement.

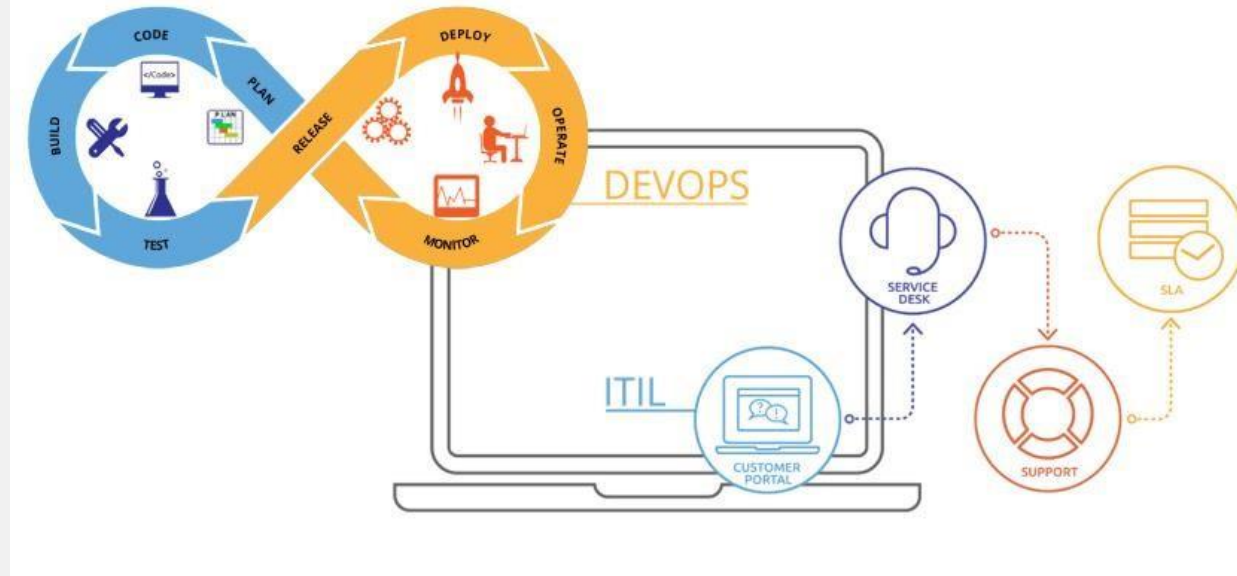
- Par exemple, si un script qui exécute le pipeline de déploiement ne fonctionne pas correctement, il doit être corrigé immédiatement.

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 ils devraient être améliorés et d'ajuster la méthode travail en conséquence.

Traditionnelles vs DevOps

DevOps et l'ITSM





4

Les pratiques DevOps

Les pratiques DevOps



➤ Les pratiques:

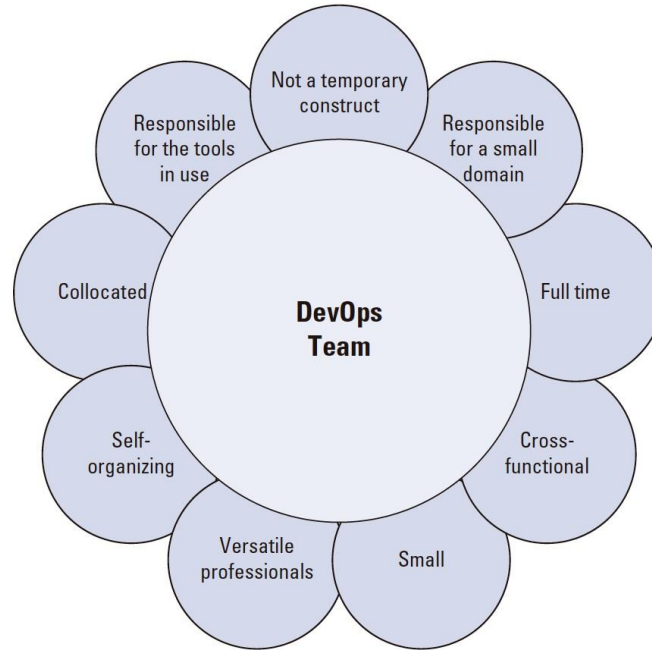
- Equipe autonome et diversifié
- Visualisation de l'avancement
- Petites livraison
- Soutenir l'innovation
- Trouver les bons moyens pour gérer les goulots d'étranglement



Equipe
Autonome

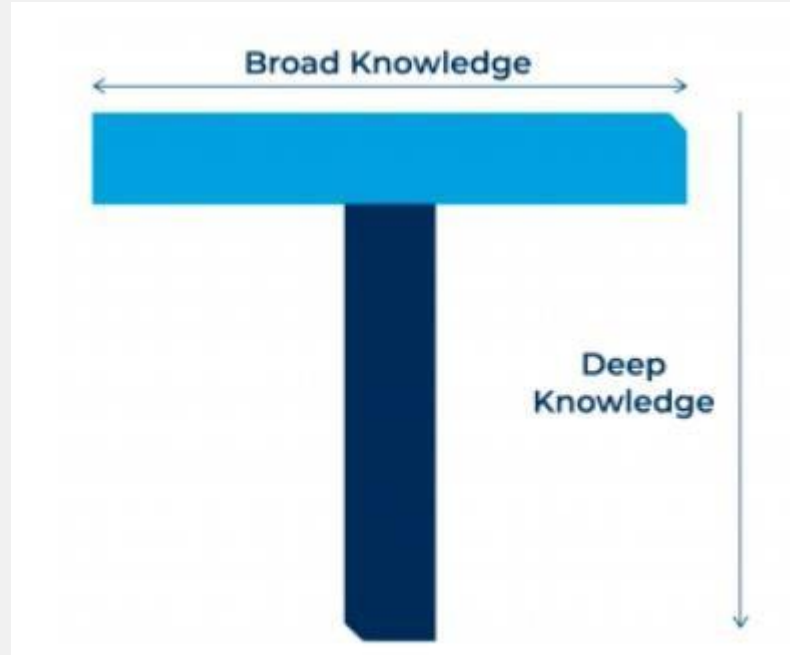
Les pratiques DevOps

Equipe autonome, diversifié



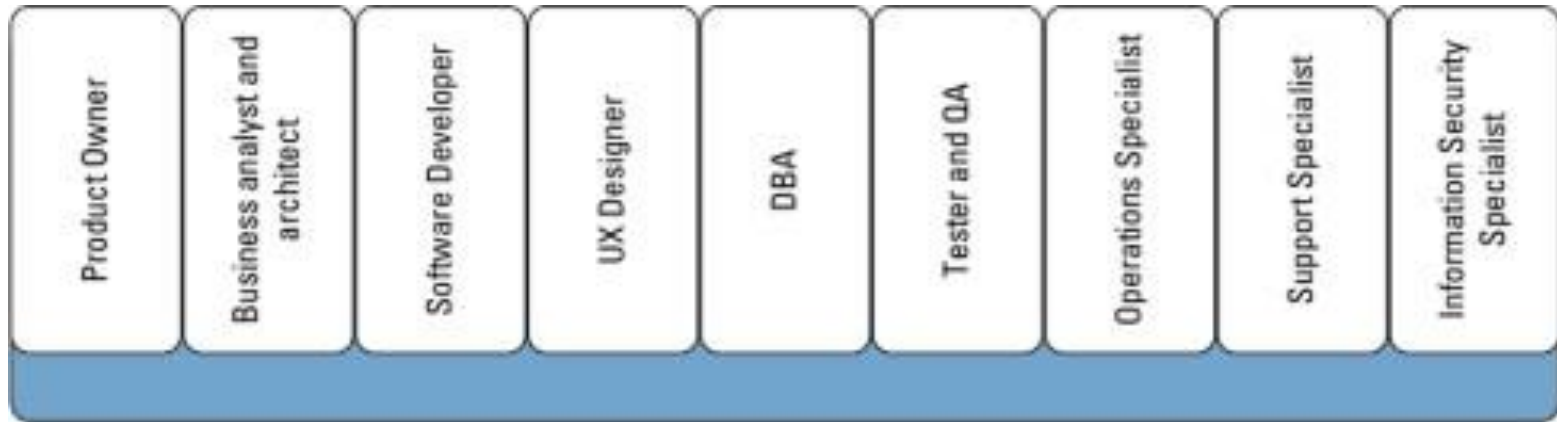
Les pratiques DevOps

T-Shaped profile/skills



Les pratiques DevOps

Exemple d'équipe DevOps



Les pratiques DevOps

Exemple de nouveaux profils DevOps

Architecte cloud - C'est une personne possédant une vaste expérience pratique de 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) - C'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.

Administrateur outillage DevOps - l'un des principaux rôles DevOps car il est le responsable de tous les outils DevOps et de la plateforme cloud utilisé pour les déploiements.



Visualisation

Les pratiques DevOps

Visualisation de l'avancement

Un outil de visualisation peut supporter le pipeline de déploiement et aider à trouver des réponses aux questions suivantes :

- Quelle étape qui ne permet pas au reste de la chaîne de fonctionner efficacement?
- Où les ressources sont déjà ou presque épuisées?
- Quelles tâches sont bloquées, afin qu'elles n'aient aucune chance d'être complétées dans cette itération?
- Que reste-t-il à faire pour la tâche qui n'a pas été complétée?
- Si nous n'avons pas le temps de faire tout le travail accepté dans cette itération, quelle partie de la tâche vaut-il la peine d'essayer de terminer pour obtenir le maximum de résultats utiles possible?

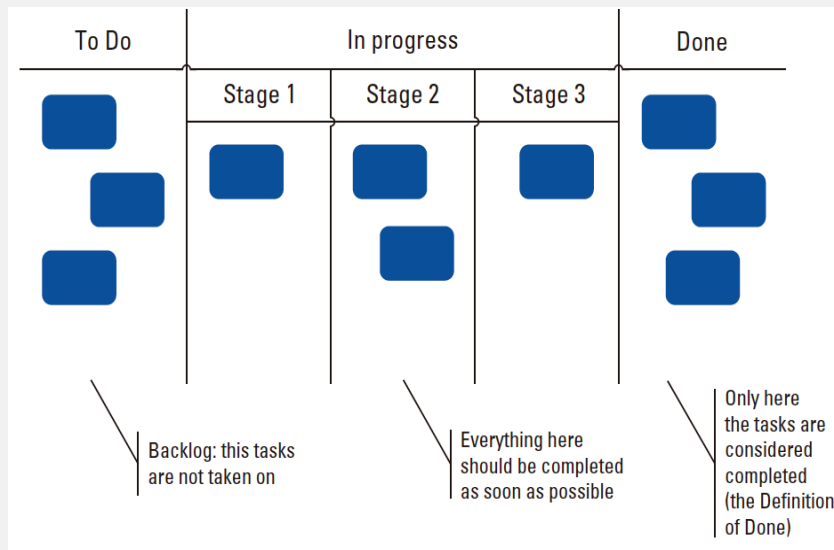
Les pratiques DevOps

Visualisation de l'avancement

➤ KANBAN

Un tableau Kanban permet de créer
Un "pull system" :

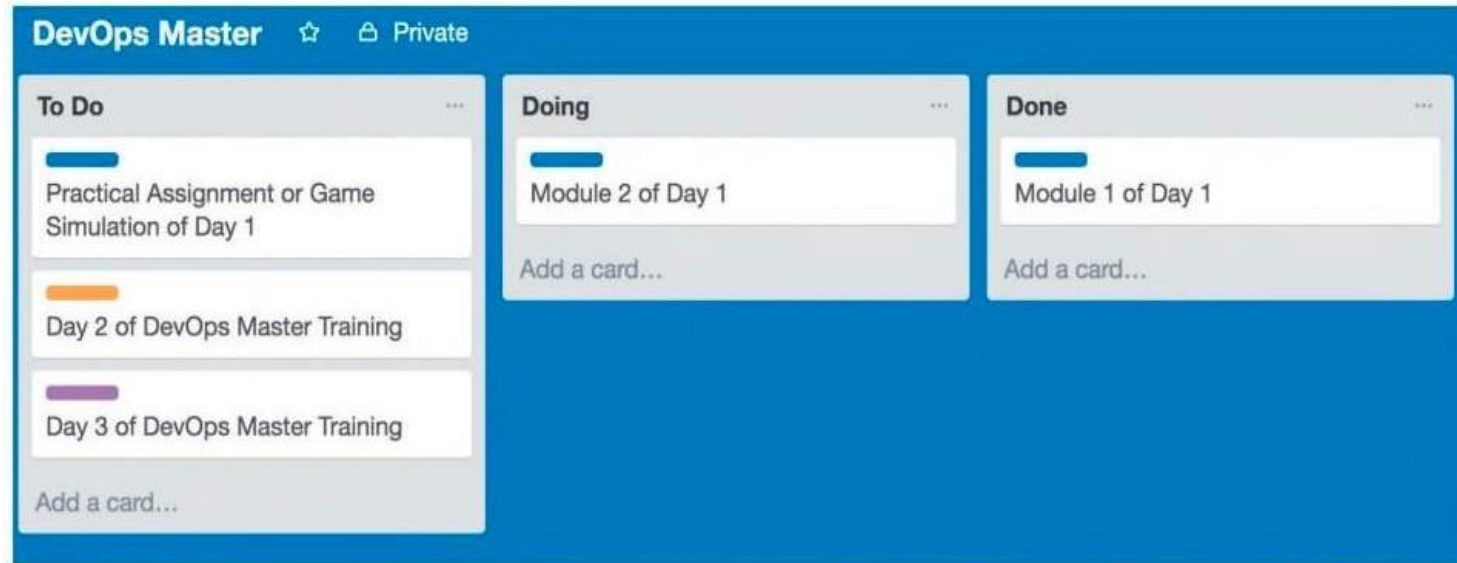
- Visualiser le flux de travail
- Mesurer et optimiser la durée du flux
- Réduit les temps d'arrêt
- Réduit le besoin de coordination
- Limiter le travail à finir (WIP)



Les pratiques DevOps

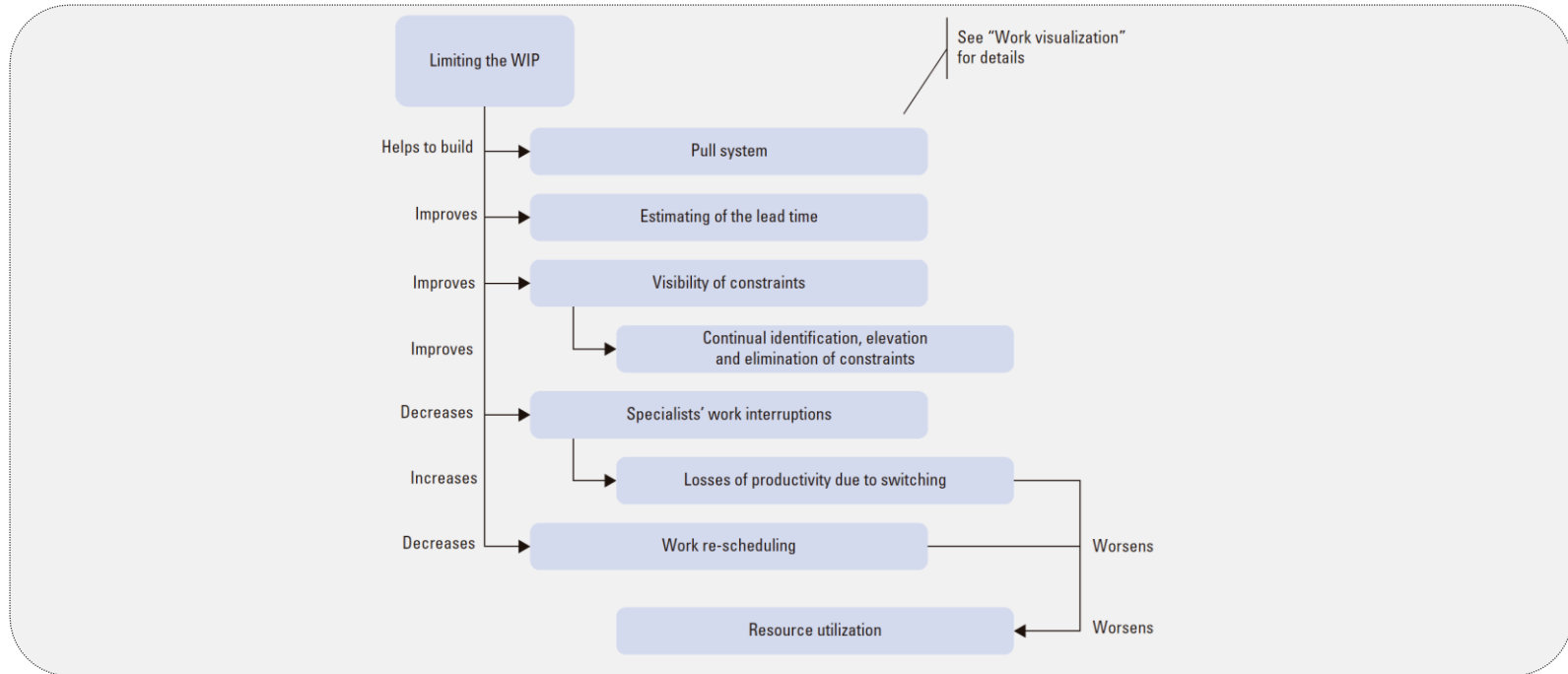
Visualisation de l'avancement

The Kanban board



Les pratiques DevOps

Visualisation de l'avancement





Petites livraison

Les pratiques DevOps

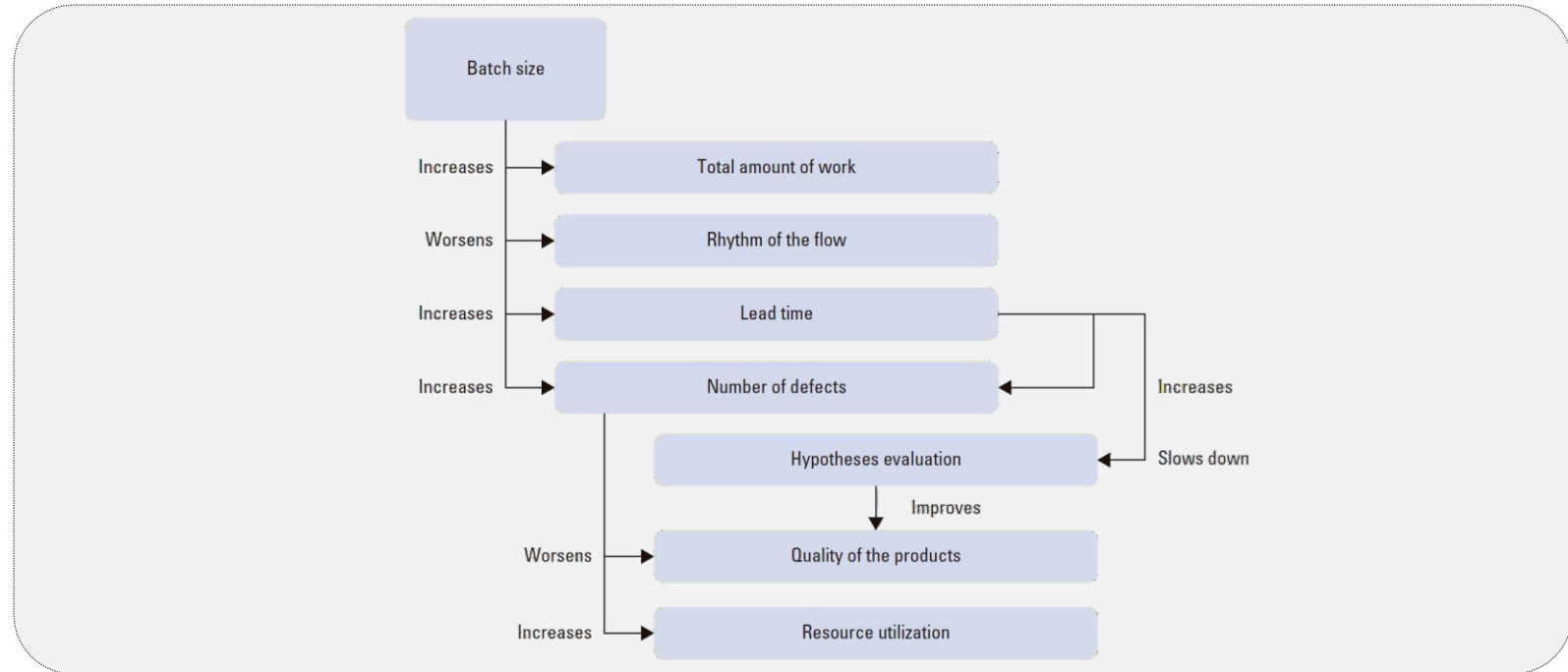
Petites livraison

Réduire la taille des lots de livraison est la deuxième façon d'améliorer le flux (WIP).

Les petits lots traversent le système plus rapidement et avec moins de variabilité, ce qui favorise un apprentissage et un déploiement plus rapides. Cela implique généralement de concentrer plus d'attention et d'augmenter les investissements dans les infrastructures et l'automatisation. Cela réduit également le coût de transaction de chaque lot.

Les pratiques DevOps

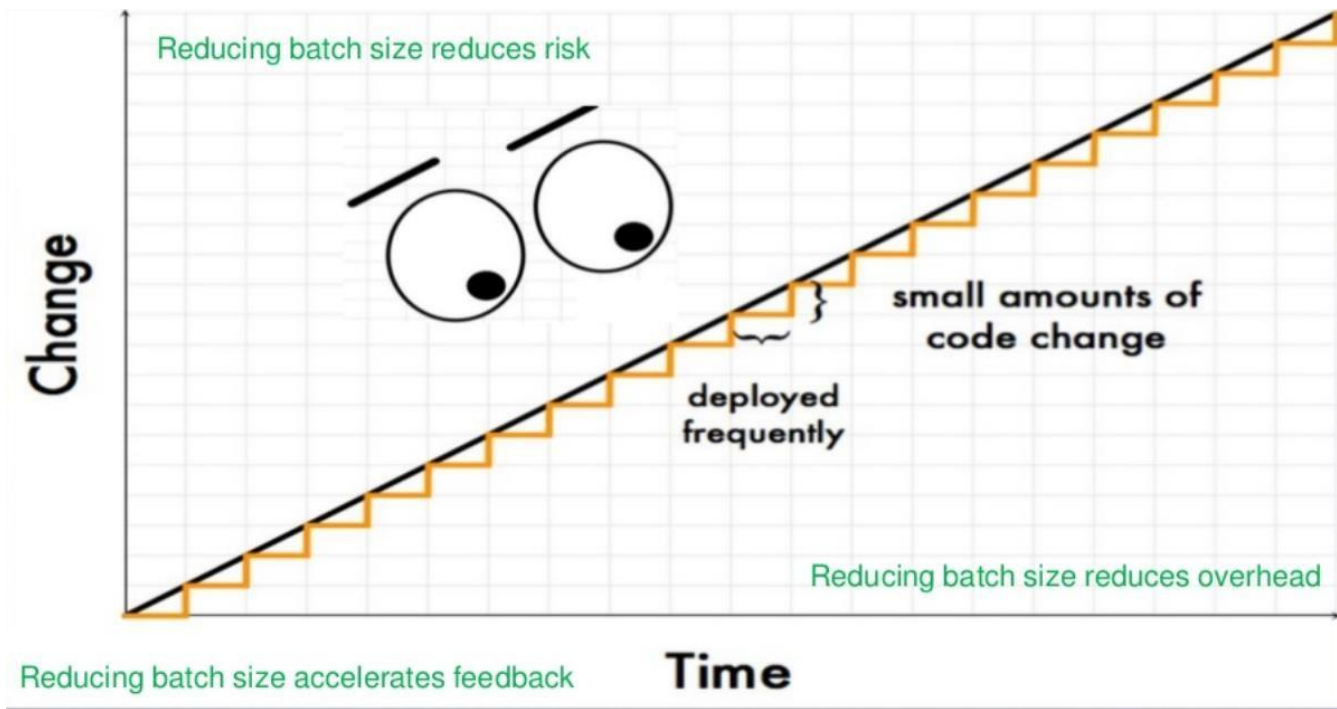
Petites livraison



Les pratiques DevOps

Petites livraison

Anti-Fragilité :





Soutenir
l'innovation

Les pratiques DevOps

Soutenir l'innovation

➤ Encourager l'innovation

Certaines entreprises commencent par attribuer une certaine proportion du temps de travail à l'amélioration et l'innovation (20%).

Un créneau spécialement alloué à l'exploration de nouvelles technologies et à la création de nouveaux produits et outils.

Sélectionnez des idées intéressantes, **investir** des ressources limitées dans certains eux.



Gérer les Bottlenecks

Les pratiques DevOps

Trouver les bons moyens pour gérer les goulots d'étranglement

➤ Bien gérer les goulots d'étranglement

Un bon pipeline de déploiement régulier sans retard n'est pas atteint du jour au lendemain et nécessite un effort considérable.

À l'aide d'outils de visualisation et de limites WIP, il est possible d'identifier les bottleneck de la chaîne.

Parmi tous bottleneck connus, il y en a un qui cause le plus grand retard - c'est celui-là sur lequel se concentrer.

Changer la méthode du travail afin de capitaliser le bottleneck l'exploiter au maximum.

Trouver un moyen d'éliminer le bottleneck.



5 Implémentation de DevOps



Application

Application DevOps

Applicabilité



➤ Le DevOps n'est pas ...

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

Ne commencez pas les pratiques de DevOps si...

- L'entreprise ne participe qu'à une partie limitée de la chaîne de valeurs.
- L'entreprise ne considère pas la technologie comme un catalyseur essentiel de son activité.

Application DevOps

Applicabilité



- DevOps devrait intéresser les entreprises lorsque les conditions suivantes sont remplies:
 - Toutes les **autres méthodes** adoptées pour l'augmentation de 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 a un **taux de changement élevé**.
 - L'activité principale (business) nécessite des **changements rapides** pour tester de nouvelles idées ou hypothèses : pour rester compétitive

Application DevOps

Applicabilité



➤ Un investissement

Les entreprises qui souhaitent réduire considérablement la dette technique ou éliminer la fragilité de l'infrastructure informatique :

- Le temps et les efforts nécessaires pour obtenir des bénéfices significatifs et continus du DevOps est très important et doit être considéré comme un investissement



Limitations

Application DevOps



Limitations

- DevOps n'est pas très approprié pour:
 - Les entreprises qui n'ont pas de développement logiciel propre.
 - Les organisations qui utilisent leurs propres logiciels, où les développeurs ne sont pas des membres du personnel.
 - Les entreprises qui travaillent depuis longtemps et ne sont pas ouvert à une restructuration majeure.

Application DevOps



Limitations

- Organisations dotées d'une architecture informatique monolithique.
 - L'introduction de petites équipes nécessite la capacité d'attribuer une zone de responsabilité distincte à chacune d'entre elles.



Application DevOps

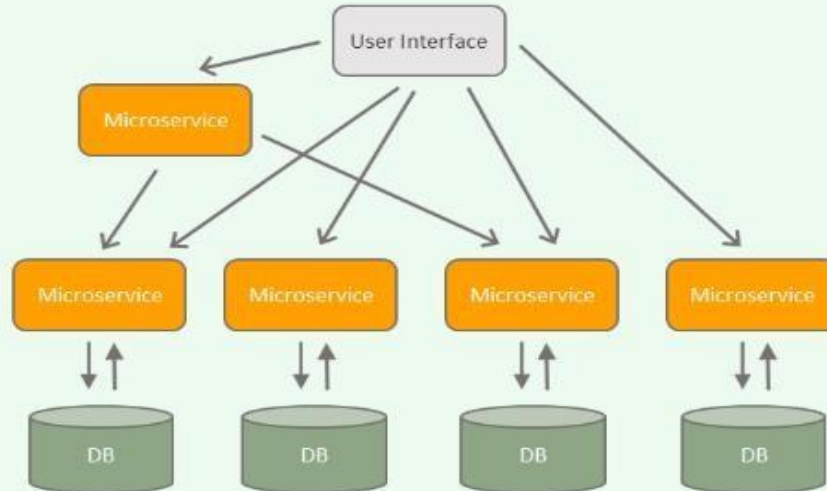
Limitations



MONOLITHIC ARCHITECTURE



MICROSERVICES ARCHITECTURE





COTS



Application DevOps

COTS

- Les risques d'un COTS
 - Evolution de la concurrence :
 - Il est nécessaire de disposer d'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.
- => Passez au **développement logiciel interne.**



Application DevOps

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.
 - Ensuite, créez votre propre script, en répliquant le travail du programme d'installation original.





Modèles d'architecture et d'organisation en évolution

Application DevOps

— Evolution d'organisation

- les difficultés d'un service IT classique :
 - Le développement (Dev) et les opérations (Ops) sont séparés de manière naturelle:

L'OPS et le support ne rentrent pas dans les détails complexes de l'architecture du système et doivent donc poser des questions, même les plus simples, aux développeurs.



Application DevOps

— Evolution d'organisation

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



Application DevOps

— Evolution d'organisation

➤ Besoin d'un changement de mindset

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

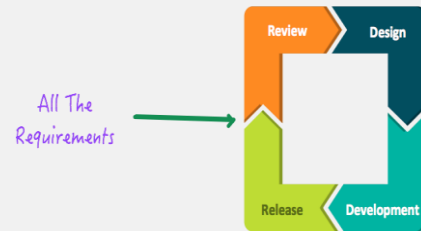


Progression Itérative

Application DevOps

Progression itérative

- Identifiez les systèmes qui sont vaguement connectés aux autres.
- Utilisez ces systèmes en tant que pilote pour appliquer les éléments de base de DevOps:
 - Chaine de valeur
 - Pipeline de déploiement
 - Système de contrôle de version
 - Gestion de configuration automatisée
- Appliquer cette expérience à d'autres systèmes.
- En commençant par des cas plus simples, vous pouvez passer à autre chose avec plus d'importance.



Application DevOps

Innovation

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

Sheryl Sandberg,
COO of Facebook

Application DevOps

Accueillir l'échec

➤ Se préparer à l'échec

- Dans le contexte de DevOps, l'échec n'est pas une infraction. Les équipes partent du principe que quelque chose va finir par mal tourner, c'est pourquoi elles mettent tout en œuvre pour détecter rapidement les bugs et les résoudre.
- Pourquoi ?

Parce que l'amélioration continue va de pair avec les défaillances.

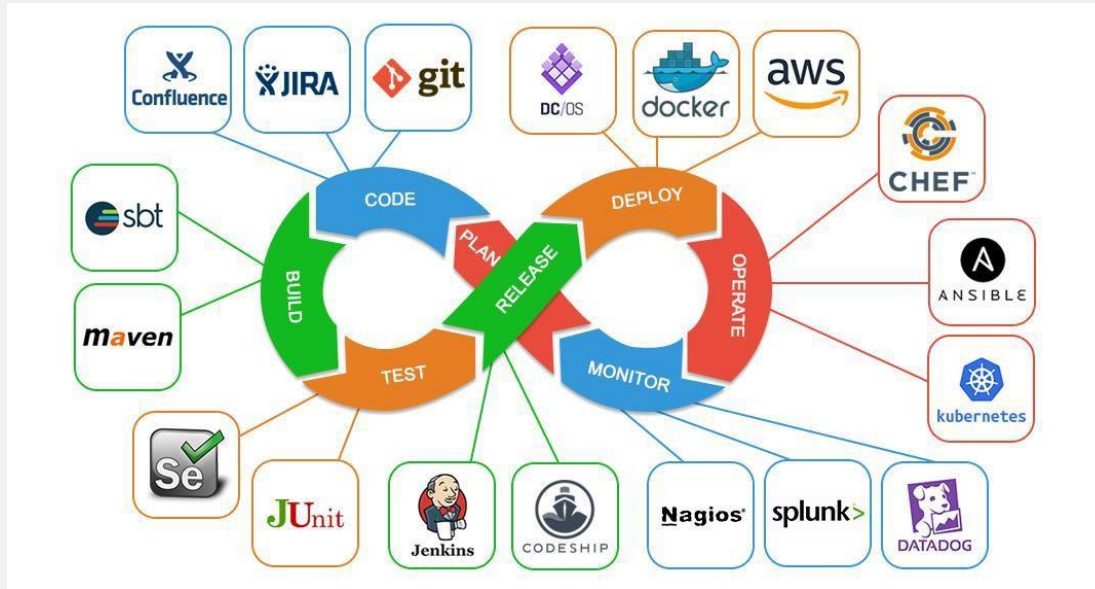


6

DevOps et
les outils

DevOps et les outils

- Les pratiques d'automatisation DevOps.



DevOps et les outils



- **Capistrano** : outil open source pour exécuter des scripts sur plusieurs serveurs, son utilisation principale est le déploiement d'applications web. Il automatise le processus de déploiement d'une nouvelle version d'une application sur un ou plusieurs serveurs web.
- **Liquibase** est une librairie open source (Apache 2.0 Licensed) pour versionner et gérer les changements de schéma de base de données.
- **Jenkins** est une application d'intégration continue qui monitore l'exécution de jobs répétés, comme les builds d'un logiciel ou des tâches automatiques.
- **VirtualBox** est une solution permettant de monter rapidement des 'Virtuals Machine'.
- **Vagrant** est un outil pour construire des VM opérables en VirtualBox via scripting.
- **Puppet/ chef** permet la gestion de configurations systèmes dynamiques.



7

Par où ...
commencer

La Culture DevOps



➤ Les Chantiers Ops.

- Automatiser les environnements
- Ouvrir le monitoring
- Autoriser & encadrer les accès en prod
- Partager la responsabilité
- Assister aux stand ups des Devs
- Développer des tests d'infrastructure
- Fournir une API de provisionning
- Versionner les configurations systèmes

La Culture DevOps



➤ Les Chantiers Devs.

- Prévoir le monitoring
- Accompagner / réaliser les installations
- Partager la responsabilité
- Fournir un packaging natif
- Développer un dashboard des évolutions
- Porter sa voix au CAB (Change Advisory Board)
- Mettre en place l'agrégation des logs
- Automatiser, Automatiser, Automatiser



8

Examen

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

Détails de l' Exam

**Durée de l'examen :**

60 minutes (15 minutes supplémentaires pour les non natifs anglophones).

Format de l'examen :

Support non autorisé

Nombre de questions :

40

Pourcentage de réussite :

65% (26 bonnes réponses ou plus)