ETML Informatique

Module: 324

DevOps Introduction

ETML Informatique

Module 324

Prendre en charge des processus DevOps avec des outils logiciels

DevOps
Objectif du module

1. Consigner et administrer de manière compréhensible et transparente les exigences et les étapes de mise en œuvre pour le développement en équipe.

DevOps
Objectif du cours

- Comprendre les principes du DevOps
- Intégration dans votre processus de travail
- Maitriser un des outils nécessaires: **Git**

« Comment parvenir à développer, puis livrer / mettre en production une application en garantissant un haut niveau de qualité et de disponibilité ? »

Lean Software Development
Lean à quoi cela ressemble-t-il ?

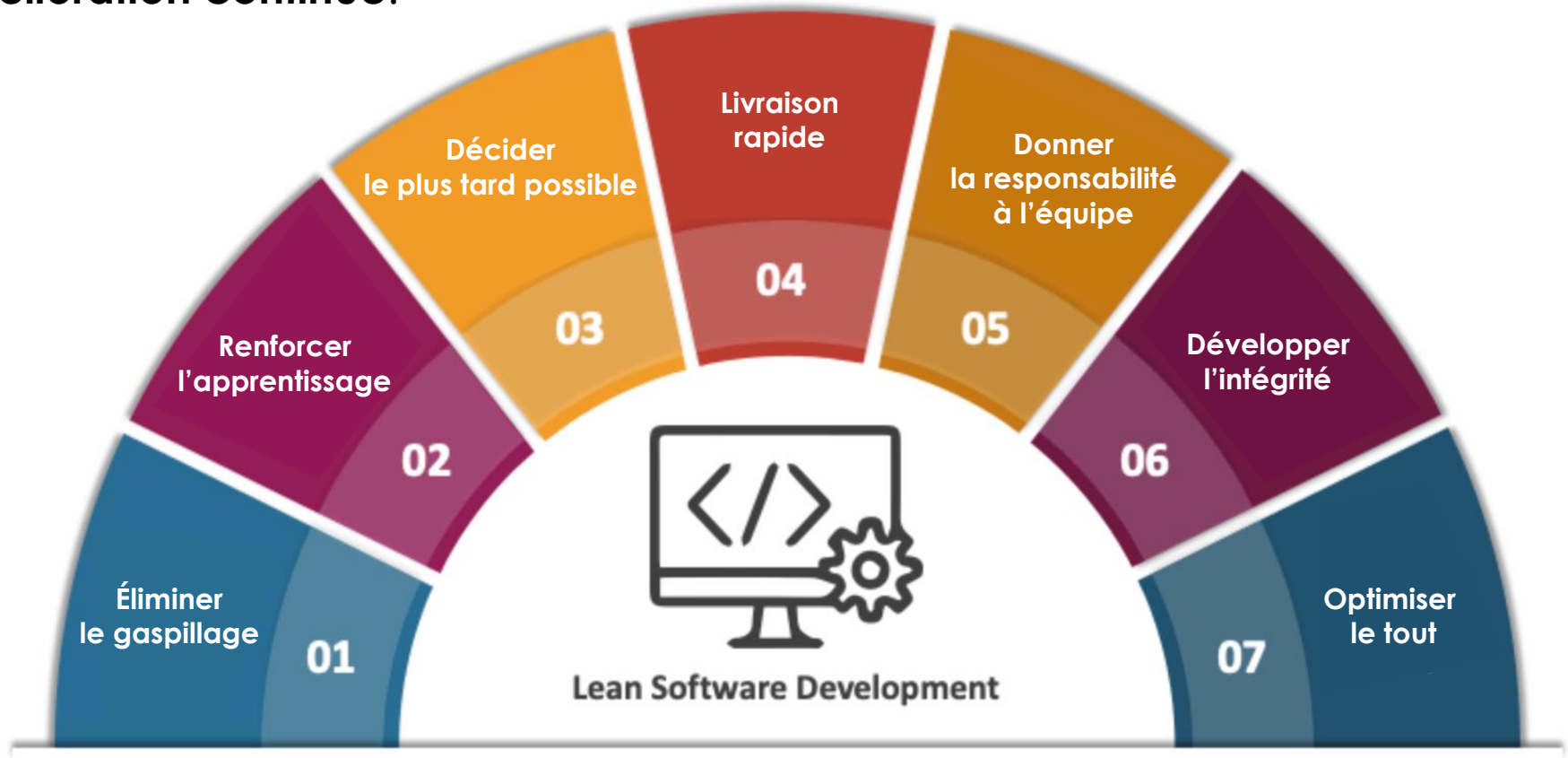
Le « **Lean Software Development** » est un cadre méthodologique dérivé des principes du Lean Manufacturing de Toyota, adapté pour le développement de logiciels.

Son objectif est de maximiser la valeur client tout en minimisant le gaspillage.

Il est basé sur **7 principes**, que l'on va découvrir.

Lean Software Development
7 principes

Ces 7 principes encouragent une approche centrée sur l'**efficacité**, la **flexibilité** et **l'amélioration continue**.



ETML Informatique

Lean Software Development
Principaux avantages

Parmi les principaux avantages du développement logiciel Lean, on peut citer :

- capacité à développer davantage de fonctionnalités en moins de temps grâce à des processus et des flux de travail optimisés ;
- éliminer les activités inutiles et, par conséquent, réduire le coût du projet ; et
- permettre à l'équipe de projet de prendre des décisions qui les motivent à atteindre leurs objectifs.

Lean et Agile
Des différences à prendre en compte

En effet, les méthodologies **Lean** et **Agile** sont très similaires.

Toutes deux visent l'optimisation des processus. En Lean, cela se traduit par le principe d'« **optimiser l'ensemble** », tandis qu'en Agile, ce principe s'apparente davantage à « **s'adapter au changement plutôt que de suivre un plan** ».

Quelle que soit la terminologie utilisée, les deux méthodologies reconnaissent l'importance de l'amélioration des processus au fil du temps.

Des bonnes pratiques
#1. Cycles d'itérations

Les itérations sont de petites parties gérables du projet qui impliquent un cycle de développement complet et aboutissent à la livraison d'une partie de la solution fonctionnelle.

Le développement logiciel Lean/Agile utilise des itérations considérablement plus courtes.

Grâce à une approche itérative, nous pouvons vérifier le résultat à chaque étape du développement, fournir des solutions opérationnelles au client et recueillir des retours.

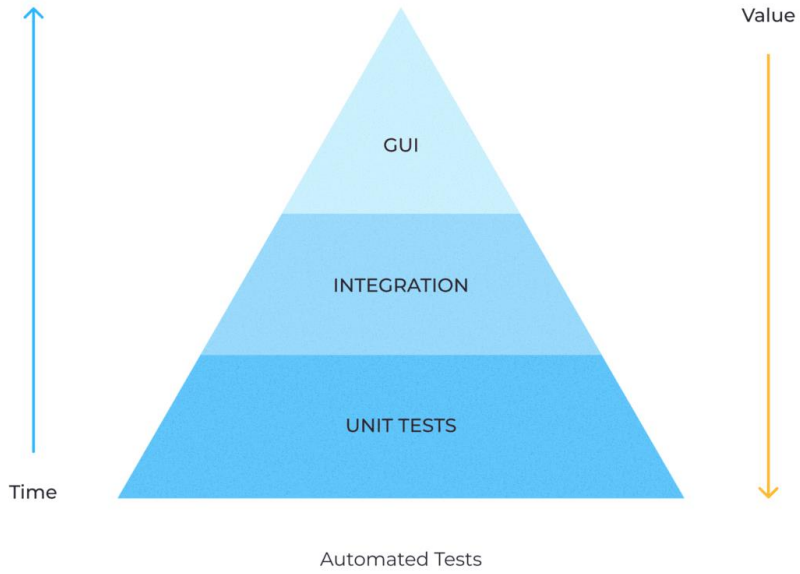
Des bonnes pratiques
#2. Réunions quotidiennes

Les réunions quotidiennes sont conçues pour coordonner le travail et synchroniser les efforts quotidiens.

Elles ont lieu une fois par jour pour recueillir les points de chaque membre de l'équipe, évaluer les progrès, identifier les points faibles et décider des actions à entreprendre aujourd'hui.

Des bonnes pratiques
#3. Automatisation des tests

L'automatisation des tests est l'un des concepts clés du développement logiciel Lean. De nombreux tests peuvent être automatisés, comme les tests unitaires, les tests d'intégration, etc.



Des bonnes pratiques
Intégration continue (CI)

L'intégration continue consiste à intégrer de petites modifications de code, aussi souvent que nécessaire, dans un référentiel partagé. Le système compile ensuite automatiquement le code et exécute des tests automatisés.

C'est pourquoi l'automatisation des tests est considérée comme un élément fondamental du pipeline CI/CD, et la CI permet de détecter davantage de bugs et de les corriger à moindre coût.

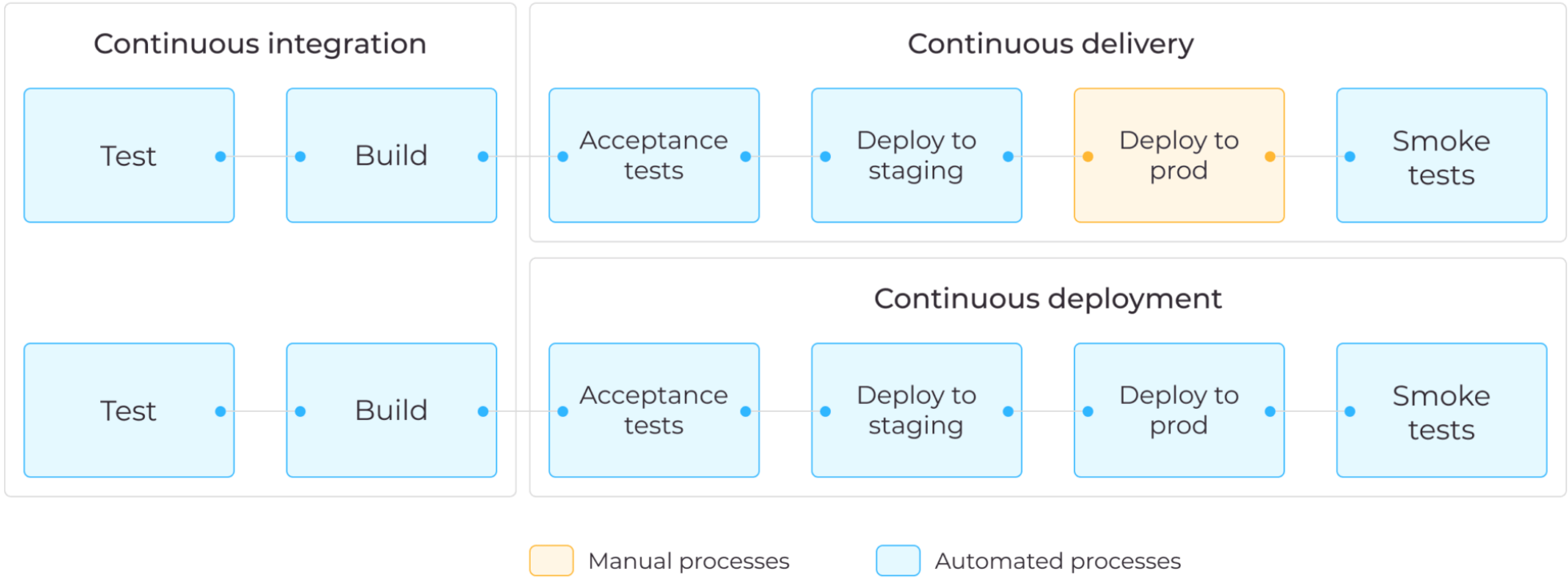
Des bonnes pratiques
Livraison continue (CD)

La livraison continue (CD) constitue la prochaine étape de l'évolution du concept d'intégration continue.

La CD compile automatiquement le code, exécute des tests automatisés et prépare une version manuelle. Si elle est correctement réalisée, les développeurs reçoivent toujours un artefact de build prêt à être déployé et ayant passé avec succès les tests standard. Cela permet des versions plus fréquentes.

Des bonnes pratiques

#4. Intégration et livraison continues (CI/CD)



ETML Informatique

Des bonnes pratiques

#5. Engagement continu des clients dans le processus de développement

L'une des principales raisons de l'échec des projets logiciels est que le produit final ne répond pas aux attentes et aux besoins des clients.

Autrement dit, les parties prenantes n'ont pas été suffisamment impliquées dans le processus de développement, pensant que chacun saurait comment s'y prendre. C'est là que le Lean fait la différence.

Lean Software Development
Conclusion

Le Lean aborde le développement logiciel et l'ingénierie en général avec une approche minimaliste et met l'accent sur les personnes, l'apprentissage et l'amélioration continue.

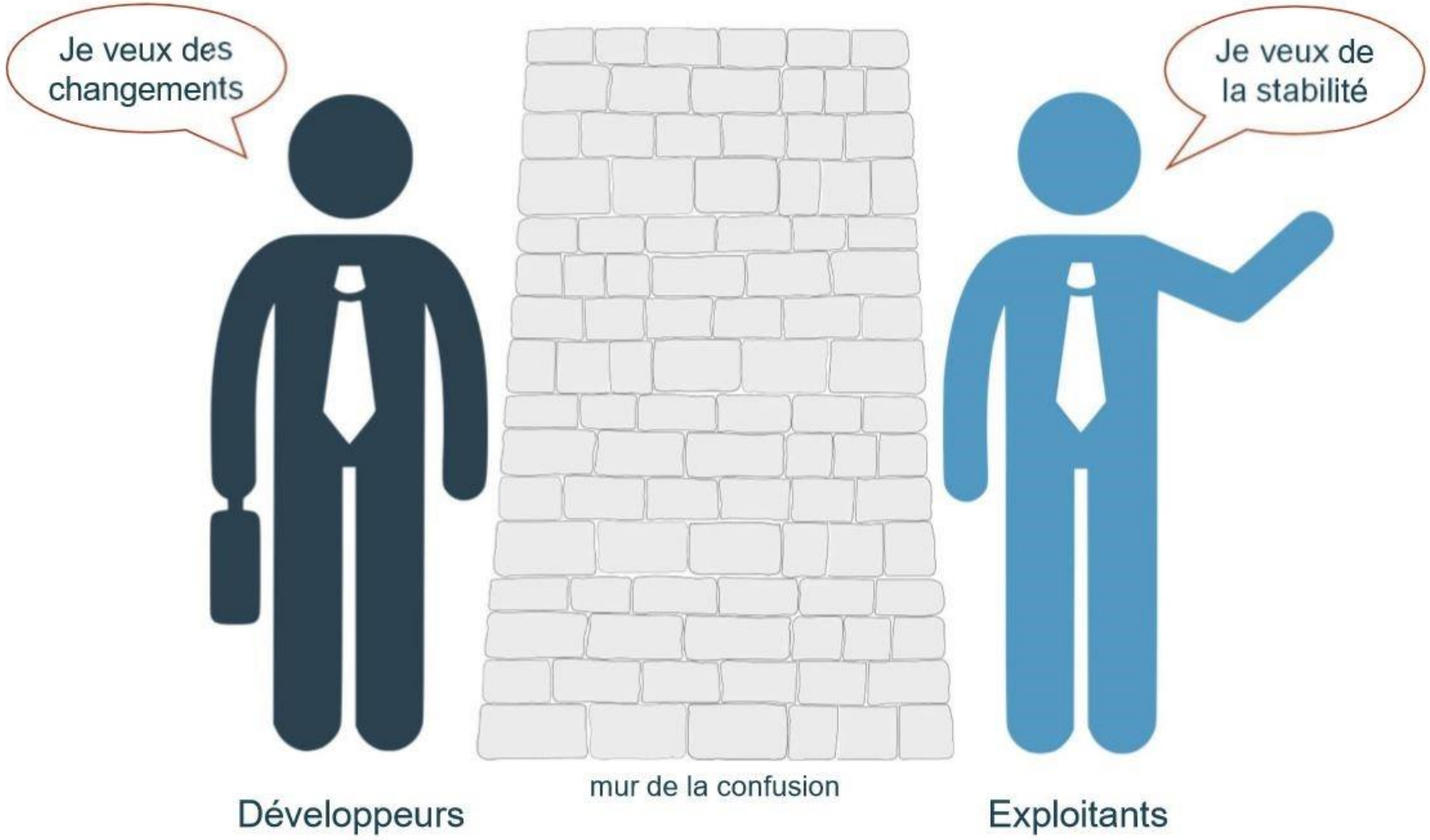
Bien que Lean ne s'appuie pas sur des règles de processus strictes, l'adaptation de ses principes au développement logiciel peut apporter de nombreux avantages, notamment l'efficacité globale du processus de développement, la livraison anticipée du produit et une meilleure prise de décision.

DevOps
Qu'est-ce que le DevOps ?

Le DevOps est une **culture + technique** qui vise à favoriser une collaboration entre deux métiers complémentaires mais qui travaillaient séparément :

- le développeur logiciel « **Dev** »
- et l'administrateur en charge des infrastructures informatiques « **Ops** ».

DevOps
Wall of confusion



DevOps
Causes du mur de la confusion

- Objectifs et cultures différents
- Transfert du travail
- Manque d'outils et de processus intégrés
- Défaut de communication

DevOps

Comment DevOps aide à démolir ce mur ?

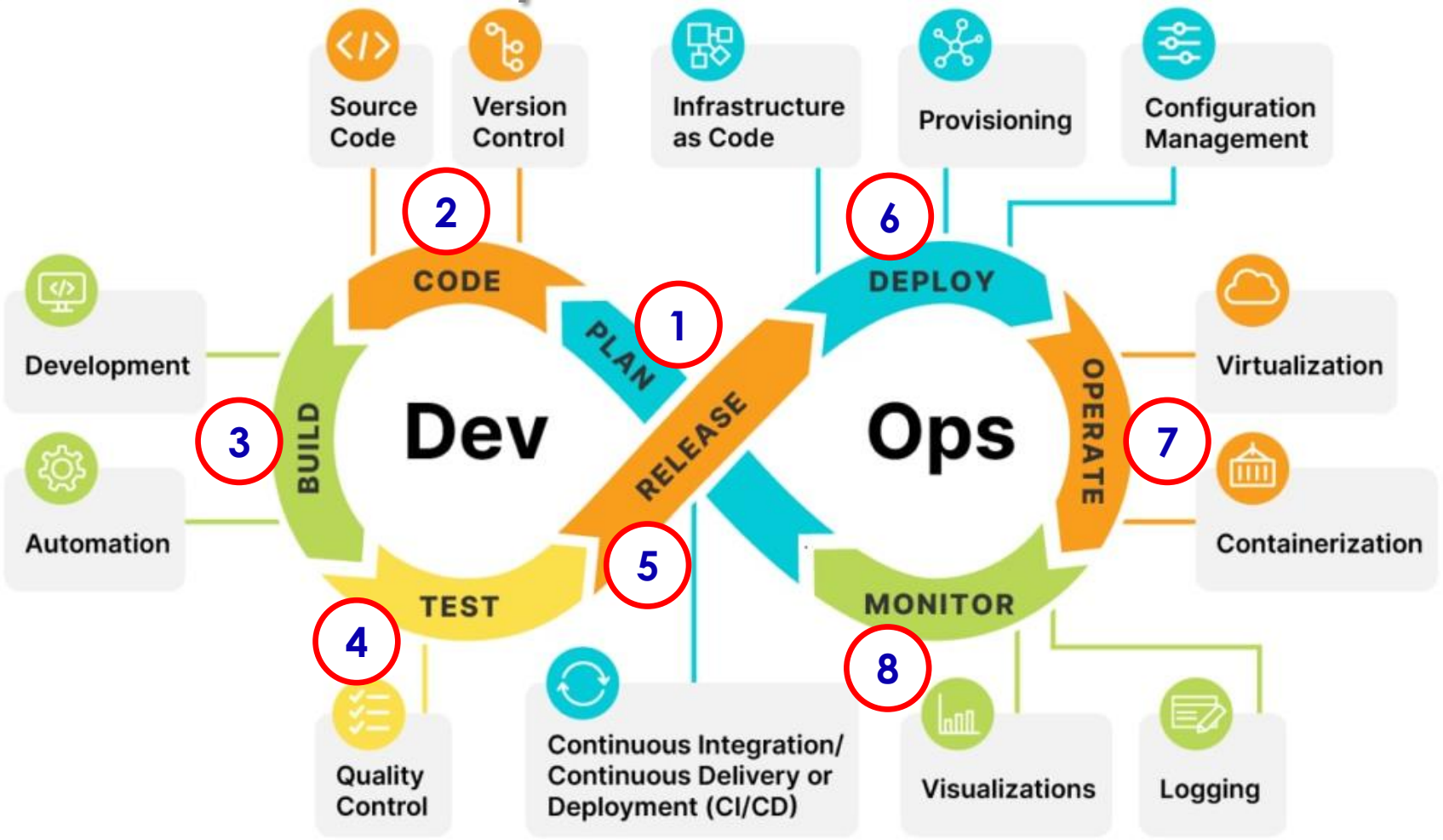
- Culture collaborative
- Intégration et automatisation
- Implication précoce des opérations
- Boucles de rétroaction continues
- Vision partagée

DevOps
Résumé: 4 fondements

- Culture et Collaboration
- Automatisation
- Intégration continue (CI)
- Livraison continue (CD)

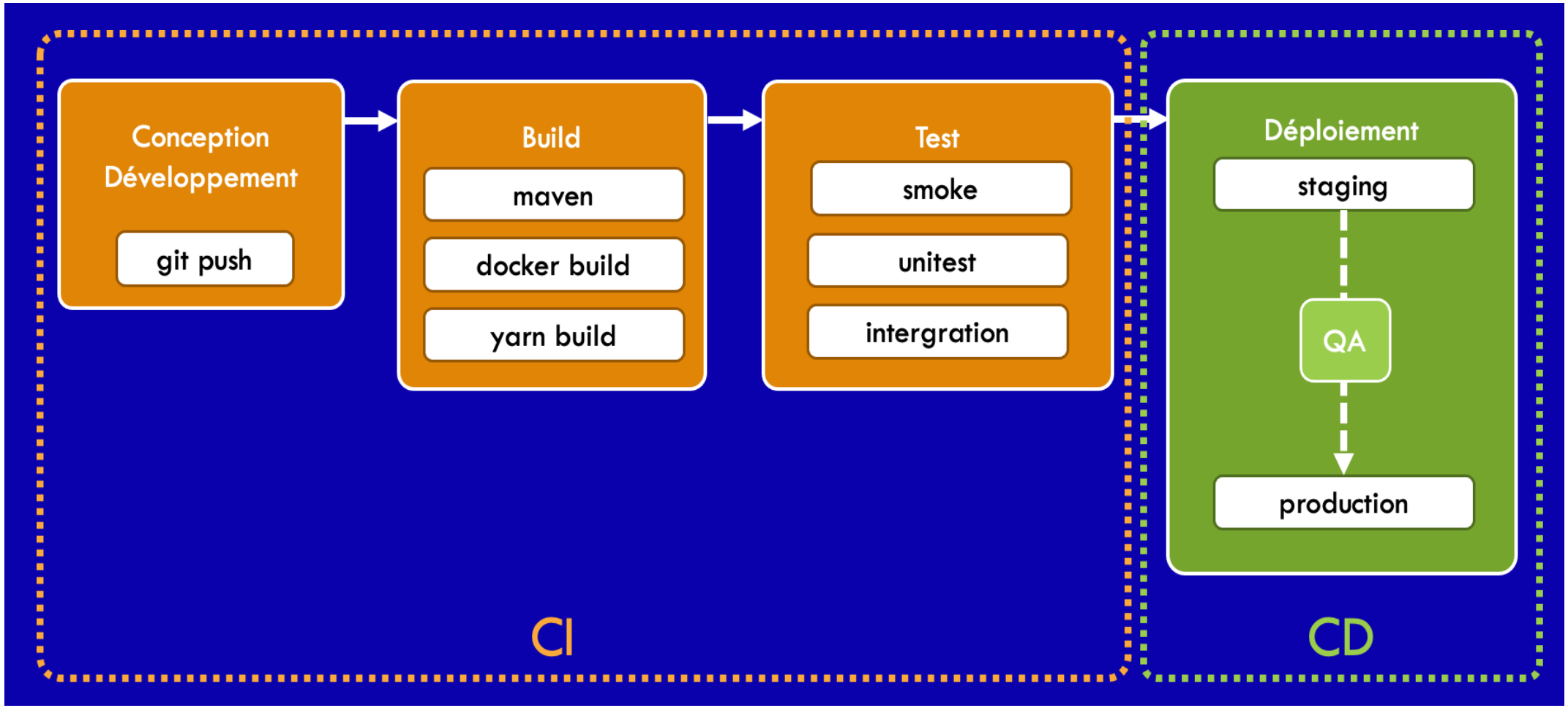
DevOps

CI/CD : 8 étapes



ETML Informatique

DevOps
CI/CD : workflow / tâches



ETML Informatique

DevOps

Outil logiciel: Git

DevOps
Outil logiciel: Git

Un des outils logiciels utilisé et plus que nécessaire dans un projet CI/CD est : **Git**

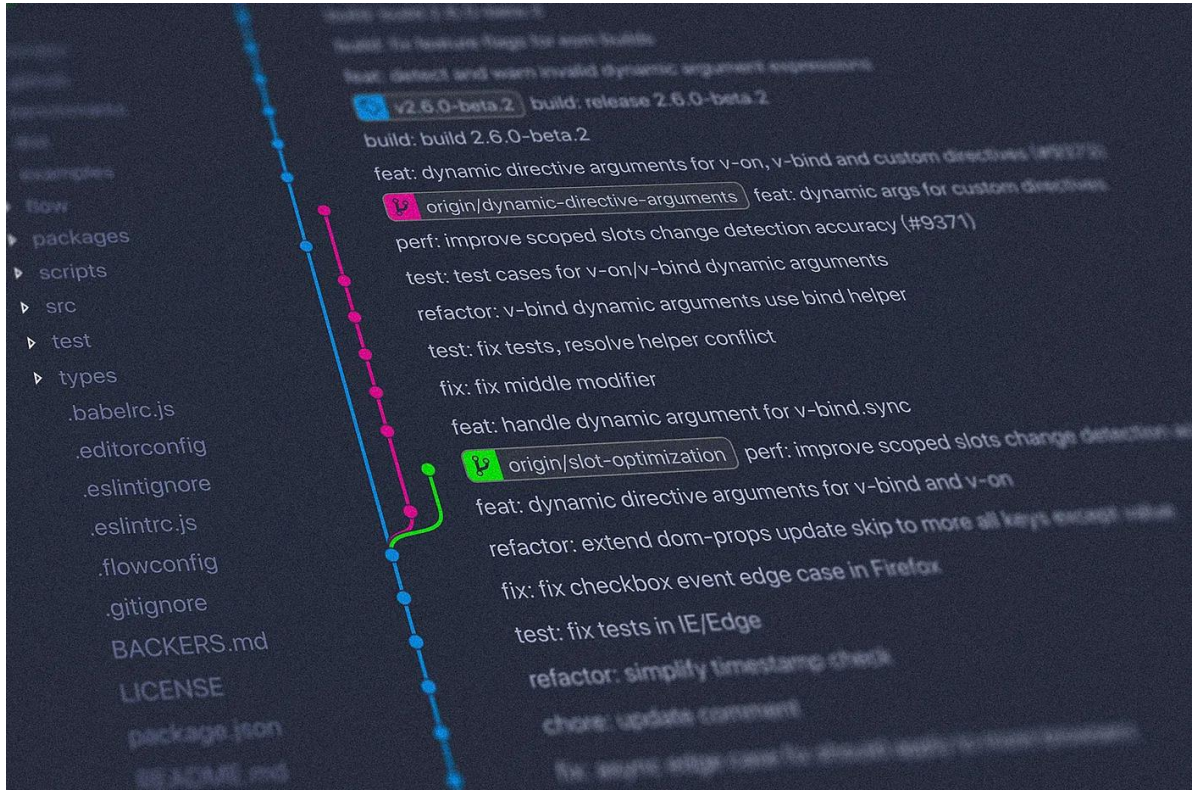
Git est un système de contrôle de version distribué utilisé pour suivre les modifications dans le code source et faciliter la collaboration entre développeurs.

Il permet aux équipes de travailler simultanément sur des projets, de gérer les versions du code, de revenir à des états antérieurs, et de fusionner des contributions de manière efficace.

C'est un outil essentiel pour le développement de logiciels modernes grâce à sa robustesse, sa flexibilité et sa rapidité.

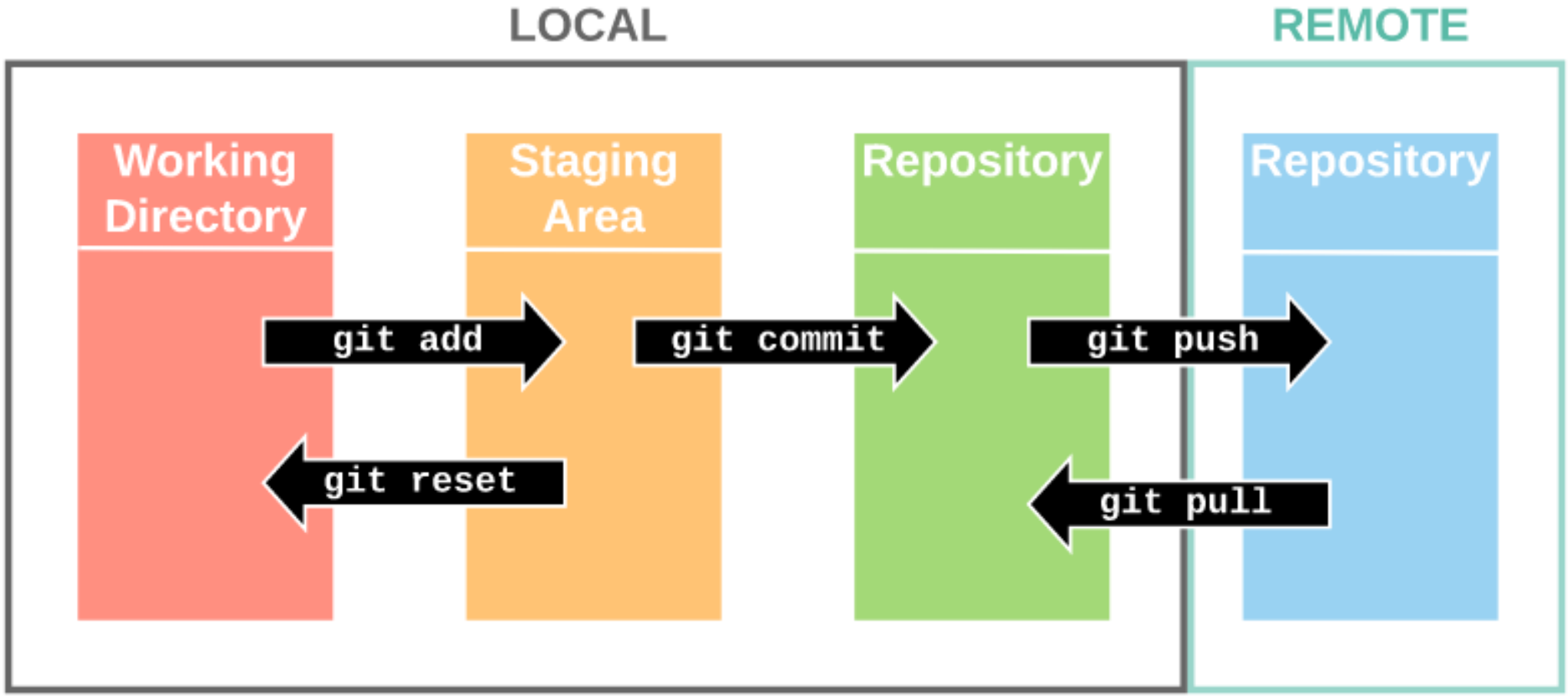
DevOps
Outils logiciels

Mes connaissances de Git



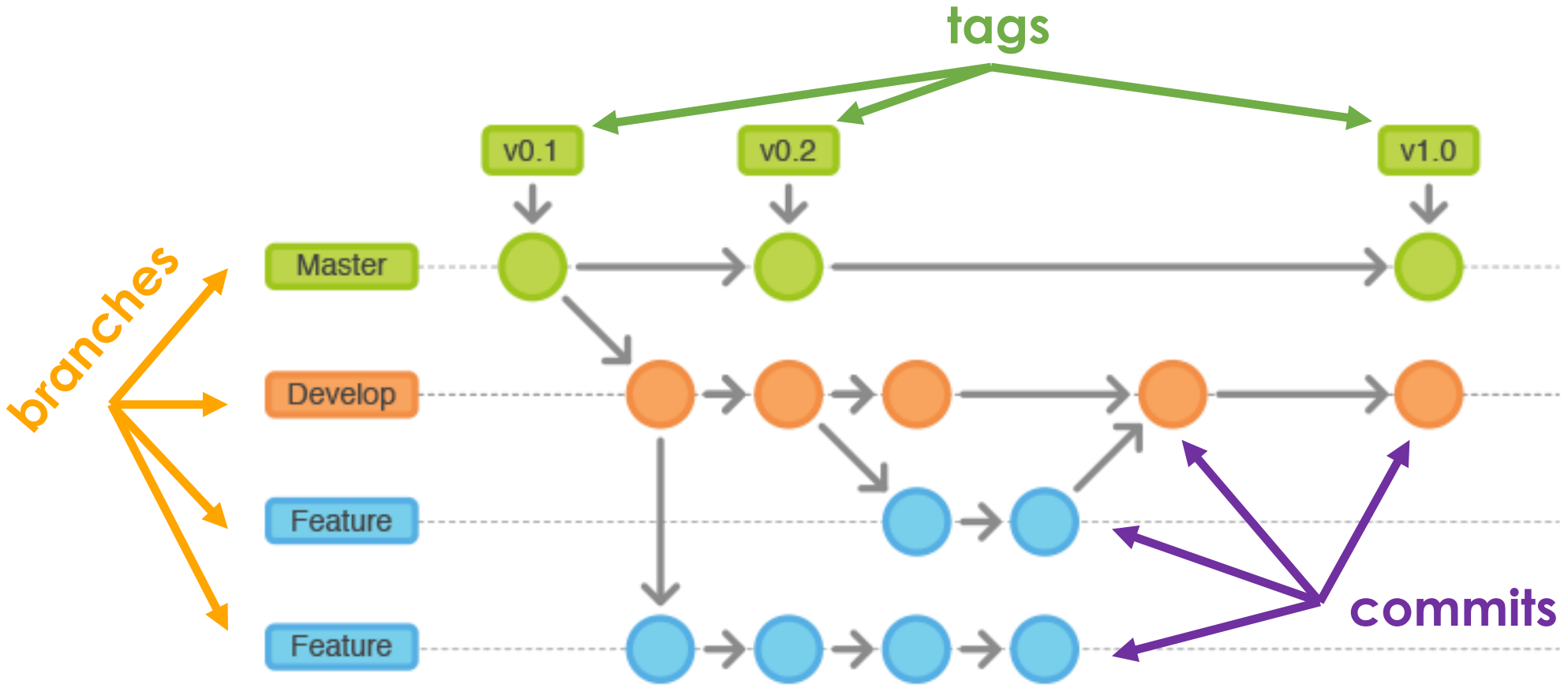
ETML Informatique

DevOps
git: principe de base



DevOps

git: commits, branches et tags



DevOps
git: exercice

1. **e-exercice-git-1.md** : commandes de base

2. **e-exercice-git-2.md** : gestion des merge issues local

DevOps
Qu'est-ce qui va dans git et ce qui n'y va pas ?

★ Les fichiers de code source : **oui**

★ Les fichiers de configuration : **oui**, mais seulement le modèle (.env.example)

★ Les librairies dépendances : **non**

- mais le fichier maître du gestionnaire de dépendance : **oui**
- *Par exemple package.json, package-lock.json*

★ Les images et autres fichiers « externes » à l'application : **oui**

- Pour aller plus loin utilisation de l'extension Git LFS pour gérer les fichiers volumineux

★ Les artéfacts (binaires générés par exemple) : **non**

Quels sont les risques de mettre dans git les fichiers de configuration ?