3 - Keep C.A.L.M.S

3. Keep C.A.L.M.S

Le DevOps peut être résumé en cinq lettres avec l'acronyme **C.A.L.M.S.** (Culture, Automatisation, Lean, Mesure et Solidarité).
Je considère personnellement la phrase ci-dessous comme le **mantra du DevOps** à **retenir** et **appliquer en toute circonstance.**

3.1 Culture

Contrairement à ce que certains laissent penser, le DevOps ne se limite pas à l'utilisation d'un outil, ou à la mise en place de nouveau processus de déploiement. Il doit **modifier en profondeur la culture de l'entreprise**, aussi bien sur le plan **organisationnel**, que sur le plan **managérial**.

Ces modifications sont d'autant plus compliquées, car elles nécessitent un changement fondamental sur la collaboration des différentes équipes :

- **Développeurs** communément appelés "Dev", ils cherchent à **apporter des changements** avec de nouvelles fonctionnalités.
- **Opérations** communément appelées "Ops", ils cherchent à **garder une stabilité** d'infrastructure.
- **Assurance Qualité** communément appelée "QA" pour **quality assurance**, ils cherchent à **réduire les risques** apportés par les changements.
- **Responsables produit**s communément appelés "PO" pour **product owner**, ils cherchent à faire **correspondre le produit avec les besoins** des clients.
- **Sécurité** communément appelée "SSI" pour **sécurité des systèmes d'information**, ils cherchent à **sécuriser l'ensemble des solutions** de l'entreprise.
- **Ressources Humaines** communément appelées "RH" , ils cherchent à **harmoniser le travail** des collaborateurs.

Chaque groupe ayant des **objectifs différents**, voire opposés, il va falloir relever un défi de taille pour leur trouver un **objectif commun** : l'adoption de la **culture DevOps à tous les niveaux**.

Le manifeste Agile reste une grande source d'inspiration pour la culture DevOps.
Le premier point du manifeste **« Les individus et les interactions plutôt que les processus et les outils »** va clairement dans ce sens.

3.2 Automatisation

Les machines informatiques sont excellentes pour effectuer des tâches répétitives. Il serait dommage d'effectuer une **transformation digitale** sans en profiter.

Il y a donc un mot d'ordre : **il faut automatiser autant que possible !**

C'est une habitude qu'on prit les équipes Ops afin de **rationaliser** et de **faciliter** leur travail. Malheureusement il n'est pas toujours reconnu comme un processus nécessaire au bon fonctionnement d'une entreprise.

Cependant, c'est grâce à ce type d'automatisation qu'une organisation arrive à gérer des milliers de serveurs sans avoir d'équipe conséquente. Il parait évident qu'avec le nombre croissant de serveurs que gère une entreprise, une gestion manuelle de ces derniers n'est plus envisageable. C'est pourquoi le mot d'ordre est devenu : **"Automatiser"**, heureusement pour y arriver, vous pourrez vous appuyer sur des outils tels qu'Ansible Puppet, Chef, Terraform et autres.

3.3 Lean

Le **« Lean »** est également un des fondements du DevOps.

Si vous voulez respecter les **principes du Lean**, vous devrez traiter **tous les aspects** mentionnés ci-dessous et non seulement certains points : vous pourrez ainsi **optimiser le pipeline de production**.

Selon Taiichi Ohno qui est à l'origine du système Lean de « Toyota », il existe 7 sources de gaspillage dont 5 peuvent aisément être transposer à une approche DevOps:

- Surproduction et stock excessifs : on va parler de fonctionnalités inutiles, qui sont développées pour rien. Il faut bien définir une fonctionnalité pour qu'elle réponde aux besoins du client, ni

plus, ni moins.
- Les défauts ou retouches : on va parler du temps perdu en correction de bug, il est plus rentable de déployer une fonctionnalité qui a été suffisamment testée.
- Le temps d'attente : on va identifier et supprimer les goulets d’étranglement du traitement de l'information de façon à fluidifier les traitements.
- Les transports et déplacements inutiles : on va parler du transport de l'information afin de la traiter et de la stocker au plus proche de son lieu d'utilisation.
- Les traitements inutiles : on va uniquement récolter et traiter les informations utiles (cette approche est un peu réfutée par le bigdata qui a tendance à tout stocker pour laisser la possibilité de traiter les informations plus tard).

Les équipes Devs, Ops, QA, PO et autres vont donc devoir travailler main dans la main afin d'identifier les taches inutiles et les supprimer en suivant la logique PDCA (Plan, Do, Check, Act) faisant référence à la roue de Demings .

3.4 Mesure

Il apparaît impossible d’**améliorer les processus et d’identifier les taches inutiles**, ou celles qui peuvent être améliorées, sans un système de collecte de **métriques** fiable et pertinent. Supposons qu’une nouvelle version du logiciel devrait rendre l’application plus rapide. Comment savez-vous qu’elle est plus rapide ? Le seul moyen est de le mesurer.

Dans les organisations informatiques traditionnelles, la surveillance des infrastructures était la responsabilité des Ops. Cette approche est quand-même réductrice, car le service ne se limite pas à un ensemble de serveurs. Avec la participation de l’équipe de développement nous pouvons fortement améliorer le monitoring. Les développeurs sont souvent les mieux placés pour définir les métriques pertinentes à mesurer.

3.5 Solidarité

Il est très important d’aligner **toutes les parties prenantes sur les mêmes objectifs**. Tous les intervenants devraient participer et travailler **ensemble** pour formuler les objectifs finaux, car cela crée le sens de la propriété.

Les gens sont prêts à travailler ensemble quand leurs **pensées** et leurs **opinions** sont **entendues**.

- Il est important d’**éliminer les goulets d’étranglement** à l’intérieur des départements ou entre eux ;
- Il est également important de **partager les avantages** entre les équipes. Il ne doit pas y avoir de favoritisme.

Pour voir la fiche complète et les documents attachés, rendez-vous sur
https://elearning.26academy.com/course/play/5aa2652f53e5ad694aa0e609