

Pourquoi DevOps ne tient pas ses promesses ?

Date : 21 avril 2022

Speaker : Gérôme Egron (WeScale) et Guillaume Mathieu (WeScale)

Format : conférence (45mn)

Introduction

On parlait déjà de DevOps en 2012 au premier DevOxx France.

Gérôme commence cette conférence par nous expliquer que, pendant le Covid, il avait l'impression qu'il y'avait 2 catégories de population :

1. Celle au chômage technique sans enfant qui cherchait des sources d'occupation pour ne pas s'ennuyer
2. Celle en télétravail avec enfants à la maison qui faisaient des doubles journées

Chaque catégorie avait ses propres aspirations. Cette défiance construit le mur de la confusion.

Ce souvenir permet d'introduire la métaphore tristement célèbre du mouvement DevOps : le **mur de la confusion** ; mur imaginaire séparant l'équipe de développement de celle d'exploitation. Chaque équipe a ses propres objectifs. Ce mur représente la différence d'objectifs des deux équipes.

Constats / Observations

Modèle **CALMS** appliqué au DevOps :

- Culture : responsabilités partagées, embrasser le changement
- Automation : CICD, infra as a code
- Lean
- Measurement
- Sharing

Guillaume fait remarquer que dans la mise en place du DevOps, on aborde en premier l'automatisation et on oublie bien trop souvent tous les autres aspects non techniques.

Pourquoi veut-on faire du DevOps ?

- Optimiser la valeur du produit livré
- Livrer plus vite
- Favoriser la communication et la collaboration

Des objectifs historiquement différents. 2 populations avec des objectifs différents

En tant que dév	En tant qu' Ops
Je dois faire évoluer les applications en développant de nouvelles fonctionnalités.	Je dois m'assurer que la plateforme est toujours disponible. Pour réussir, je limite tout ce qui pourrait la

Pour réussir, je veux livrer régulièrement des évolutions de code.	perturber, notamment les évolutions de code.
--	--

Nécessité d'aligner les objectifs : tooling et collaboration. La collaboration est la partie la plus difficile.

Définition du DevOps

Les speakers nous invitent à construire un lexique commun à partir d'un petit quiz en ligne nous demandant si une situation est DevOps ou pas DevOps.

En voici les résultats

1. Dire aux Ops « mon build Jenkins ne fonctionne pas » : **pas DevOps**. Arriver avec la phrase « ça marche pas » ne facilite pas la collaboration. Allez chercher de la collaboration et non du service.
2. Prévenir l'Ops le matin qu'il y'a un déploiement en prod à faire : **pas DevOps**. Ce n'est pas à l'Ops de livrer en production.
3. Demander d'ouvrir un ticket Jira pour chaque tâche : **pas Devops**. Privilégier les modes de communication synchrones et collaboratifs. Importance néanmoins de tracer ce qui a été décidé.
4. Utiliser un logiciel de contrôle de version (Git) : **DevOps**. Git fait partie de l'outillage de base. C'est le fondement des pratiques de livraison continue et de déploiement continu.
5. Incriminer l'équipe responsable du dernier bug en prod : **pas Devops**. Il y'aura toujours des erreurs. L'important est de ne pas les répéter. Permet de travailler ensemble pour éviter qu'elles se reproduisent.

État d'esprit / culture et pratiques techniques :

- Communication
- Coopération
- Automatisation

Et ceci pendant tout le workflow de réalisation. On n'est pas là pour collaborer lorsqu'on a seulement besoin de l'autre. Nécessité d'avoir des points de vue différents et de trouver des occasions.

Balade empirique / REX

L'impact de l'organisation joue un rôle sur la mise en place du DevOps.

Comment faire du DevOps dans un contexte à l'échelle ?

SAFe : framework d'agile à l'échelle le plus répandu. Des principes y sont à reprendre

SAFe préconise de créer un **system team** :

- Objectifs communs
- Partage de bonnes pratiques
- Homogénéité des outils

Selon le modèle d'organisation, on retrouve l'Ops a plusieurs endroits.

Cas 1 : Equipe Ops transverse

- l'équipe Ops a mis en place l'usine logicielle et tout se passe bien
- on peut communiquer via des channels communs

Risque : plus besoin de collaborer car l'outillage devient mature. Le dialogue peut s'arrêter.

Cas 2 : Ops intégré dans chaque équipe à mi-temps

Risques : pas évident d'être dans 2 équipes à la fois, formation de silos

Nécessité de faire du team building. Exemple : scripts de déploiement Cloud ou outillage autour de la CI/CD faits en pair-programming

Cas 3 : désigner un Monsieur DevOxx

Risque : ça reste son affaire, déresponsabilisation des équipes

Take away

- Finalement l'organisation n'est pas le critère déterminant pour mettre en place une culture DevOps
- Par contre elle peut avoir un effet de nuisance (cloisonnement des équipes)

Citation extraite du manifeste Agile et valable dans un contexte DevOps :

« *individuals and interactions over processes and tools* »

La confiance entre équipe se travaille. Nécessité de créer un cadre respectueux et bienveillant :

- Intéresser vous aux faits
- Respecter l'expertise de chacun
- N'oubliez pas que les maladresses existent
- Adoptez une posture adéquate
- Prenez en considération les frustrations des uns et des autres

Faire preuve d'exemplarité et de fiabilité :

- Adopter soit même une attitude positive
- Soyez cohérents entre vos paroles et vos actes
- Coopérez au plus tôt
- Anticipez au plus tôt les contraintes de chacun
- Ne cherchez pas à changer le comportement de l'autre

Trucs et astuces pour réussir

Profiter des opportunités au quotidien pour collaborer, par exemple lors des rituels agiles. Toujours les mêmes participants : PO, Scrum Master, Dev, QA, Ops et parties-prenantes

Commencez par les **Sprint Review** :

- Gagner en visibilité sur ce qui se passe dans les équipes de réalisation

- Voir les choses avancer avant qu'elles n'arrivent
- Poser des questions et s'intéresser

Daily Stand up Meeting

- Commencez à échanger
- Donnez de la visibilité sur les sujets qui concernent l'équipe
- Levez les obstacles connus

Commencez par inviter 1 fois, 2 fois et plus les nouveaux participants.

Venez de temps en temps en **Backlog Refinement**

- Lorsque vous avez de la valeur ajoutée
- Pour comprendre comment travaillent vos collègues

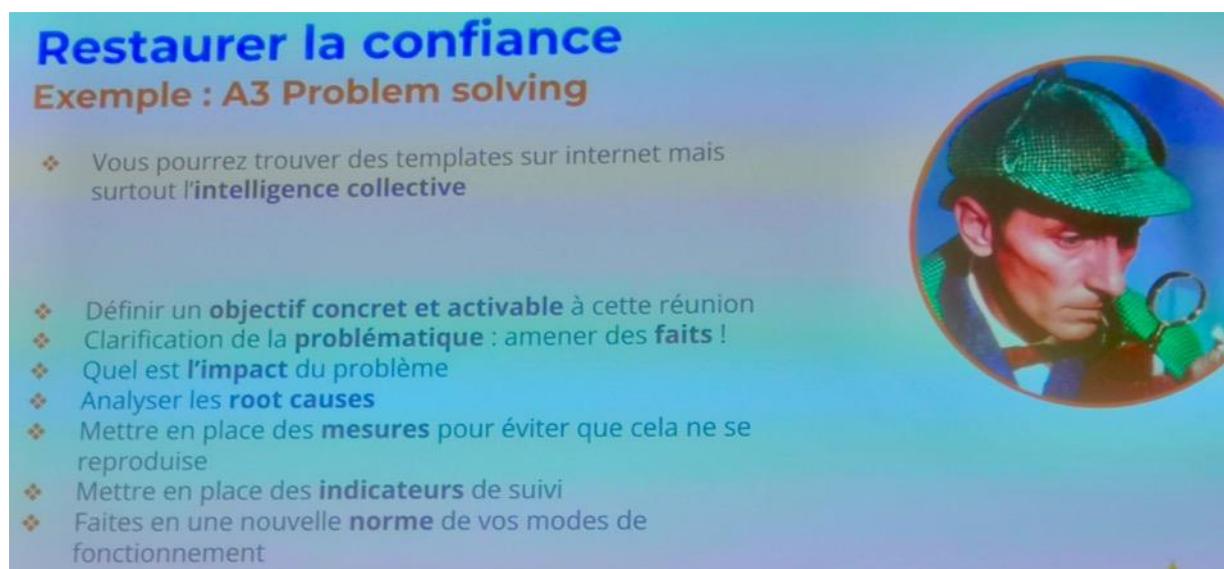
La **Rétrospective** :

- Profitez des victoires de l'équipe
- Soyez un membre actif de la démarche d'amélioration continue dans l'équipe

Lors d'un problème en production, organiser un **atelier post-mortem** pour trouver la cause racine et faire fonctionner l'amélioration continue.

Comment :

- Choisir un format adapté d'atelier commun ([A3](#) ou les [5 why](#))



Restaurer la confiance
Exemple : A3 Problem solving

- ❖ Vous pourrez trouver des templates sur internet mais surtout **l'intelligence collective**
- ❖ Définir un **objectif concret et activable** à cette réunion
- ❖ Clarification de la **problématique** : amener des **faits** !
- ❖ Quel est **l'impact** du problème
- ❖ Analyser les **root causes**
- ❖ Mettre en place des **mesures** pour éviter que cela ne se reproduise
- ❖ Mettre en place des **indicateurs** de suivi
- ❖ Faites en une nouvelle **norme** de vos modes de fonctionnement

Adapter les règles au télétravail

Ne changer rien aux bonnes pratiques adoptées en présentiel :

- points formels
- points informels (café, moments chill après les rétros, jeux le midi)
- tableaux blancs numériques,
- appliquer les bonnes pratiques de réunion à distance (ponctualité, caméra, focus)

La **règle des 2 pieds** s'applique : lorsqu'on n'a pas de valeur ajoutée, on ne vient pas et on prévient l'organisateur.

Conclusion et take away

En tant que dév	En tant qu' Ops
Suivre son code jusqu'à la prod et surveiller les impacts (mettre son nez dans le monitoring) Collaborer avec les Ops pour la mise en place de filets de sécurité pour le déploiement.	Faciliter sécuriser le changement en fournissant aux dévs des outils leur permettant d'être autonomes. N'ont plus à s'occuper des tâches de déploiement pour se concentrer sur l'outillage.

Conclusion

- ❖ Pourquoi DevOps ne tient pas ses promesses ?
 - > L'importance des relations humaines est trop souvent ignorée, du coup on ne profite pas de **l'intelligence collective**
- ❖ Tout le monde doit prendre sa part et être **exemplaire** dans l'établissement de bonnes relations humaines
- ❖ DevOps est un **changement de culture** où chacun doit être acteur et ne peut/doit pas reposer sur une personne
- ❖ N'oubliez pas de mettre en place des indicateurs pour rationaliser les décisions

Take away

- Au prochain incident, pensez à déclencher un post mortem
- Ne préjugez pas de l'incompétence de l'autre mais échangez avec lui pour voir comment l'aider
- **Appelez à l'aide après avoir creusé et creusez ensemble après**
- Permettez à vos équipes de travailler ensemble (situation bien démocratisée en entreprise aujourd'hui)