

Transformation DevOps

Plan

1 Introduction

2 La culture devops

Objectifs

"Il ne suffit pas d'avoir les mains propres,
il faut aussi avoir l'esprit pur."

Thalès; VIe s. av. J.-C.

Il s'agit de :

- concevoir, maîtriser et architecturer un logiciel,
- choisir un cycle de développement / livraison approprié,
- avoir l'esprit d'un architecte (logiciel),
- avoir les capacités de synthèse d'un (développeur) intégrateur,
- savoir décomposer en sous-problèmes et affinements successifs pour simplifier et optimiser les processus,
- savoir travailler d'abord en binôme, puis en équipe.

Contexte

- Première apparition du terme en 2008/2009 lors de la conférence de Andrew Shafer et Patrick Debois intitulée "Agile Infrastructure"
- Contraction de "Development" et "Operations"

Objectifs

- Automatiser (et donc fluidifier) les étapes du cycle de vie d'un logiciel, du développement à la production.
- Superviser l'ensemble de ces étapes afin d'obtenir des indicateurs concrets de la "santé" du projet.

Plan

1 Introduction

2 La culture devops

Un mouvement

- Intimement lié aux pratiques "Agiles" (cf [http ://agilemanifesto.org/](http://agilemanifesto.org/))
- Un socle de valeurs et des pratiques en constante évolution

Une culture

- Un rapprochement fort (voir fusion) des métiers du développement et du suivi opérationnel.
- Un échange permanent (culture du partage) afin de continuer à améliorer les pratiques de développement et de suivi de production.

La planification en cycle infini

- Privilégier les cycles courts : "release often, release early".
- S'appuyer sur les outils méthodologiques proposés par les méthodes agiles (SCRUM, Kanban, Extreme Programming...).
- Définir les objectifs à court/moyen terme en gardant à l'idée les objectifs à long terme (découpage des tâches).

Créer

- S'appuyer sur un gestionnaire de version des sources solide (Git, Mercurial...). Il est central dans le processus DevOps.
- Sélectionner une forge logicielle pour gestionnaire de version des sources (Gitlab, Gitea...).
- Choisir une méthodologie de travail commune (git flow, git ops,...) à l'ensemble de l'équipe et cultiver la rigueur quant à son usage.
- Former régulièrement les membres de l'équipe à l'usage de l'outil de gestion de version des sources.
- Définir clairement les objectifs de qualité attendus par l'équipe pour sa base de code source.

Vérifier

- Maintenir un haut niveau de qualité du code en intégrant des mécanismes comme la **revue de code** et le **travail en binôme**.
- Cultiver une approche du **développement piloté par les tests**.
- Intégrer dans le processus de développement des outils permettant de vérifier l'état du projet de manière automatisée (qualité du code, tests de non régression...). Ces outils prennent aujourd'hui principalement la forme de **serveurs d'intégration continue** (Jenkins, Gitlab CI).

Le packaging

- Choisir un système d'empaquetage pour la diffusion du logiciel (Archive compressée ? Paquet GNU/Linux ? Images de conteneur, de machine virtuelle ?).
- Implémenter l'empaquetage du logiciel à partir des sources.
- Définir les évènements qui déclencheront un empaquetage du logiciel.

Livraison (release)

- Définir la méthodologie de diffusion des "artefacts" générés après la phase d'emballage.
- Définir la méthodologie de versionnement de ces artefacts.
- Mettre en place des mécanismes de traçabilité des artefacts (signature cryptographique, somme de contrôle).

Configurer

- Définir la méthodologie de déploiement du logiciel dans ses différentes cibles de déploiement (développement -> qualification -> production).
- Automatiser la mise à jour de ces différentes cibles lorsqu'une nouvelle version du logiciel est disponible.

Superviser (le monitoring)

- Vérifier que toutes les phases précédentes se déroulent comme prévues.
- Vérifier que les différentes cibles de déploiement sont opérationnelles avant et après mise à jour.
- Générer des indicateurs concrets sur le comportement du logiciel afin d'améliorer ses fonctionnalités/performances dans la prochaine itération.

Remerciements

Je remercie la communauté du logiciel libre pour tous ces merveilleux logiciels sans lesquels si peu de choses existeraient.

Cette présentation a été réalisée grâce aux logiciels libres suivants :

- Le système de composition \LaTeX [TeX Live](#)
- L'éditeur de texte [Vim](#)
- L'environnement graphique [Gnome](#)
- Le système d'exploitation [Debian](#)



Licence

Cette présentation est mise à disposition sous licence
Creative Commons BY-SA 4.0

- Attribution
- Partage dans les mêmes conditions

Vous pouvez obtenir une copie de la licence

par Internet

<http://creativecommons.org/licenses/by-nc-sa/4.0>

par courrier postal

*Creative Commons
444 Castro Street, Suite 900 Mountain View,
California, 94041, USA.*