

Rétrospective JDEV2015@Bordeaux

rdatadev 13 oct 2015



Guillaume
MELLA

Participant/contributeur



0346-CNRS-0062

Rappel de vos inscriptions:

T6.A01a | T0.P01 | T2.GT02 | T2.GT07 | T6.P | T0.E01 | T0.E02 |
T6.GT02 | T6.GT03 | T2.GT06 | T6.GT08 | T0.P02

- * T1 Systèmes embarqués, réseaux de capteurs et objets communicants
- * T2 Modélisation et ingénierie
- * T3 Données massives scientifiques (Big data), recherche par les données
- * T4 Transfert marchand et non marchand
- * T5 Infrastructures et interopérabilité: Le cloud et les architectures orientées service (SOA)
- * T6 **Les usines logicielles, le DevOps** et la virtualisation
- * T7 Javascript (paradigmes et éco-systèmes)
- * T8 Logiciels scientifiques et simulations: nouveaux modèles et enjeux

DevOps kesako ?

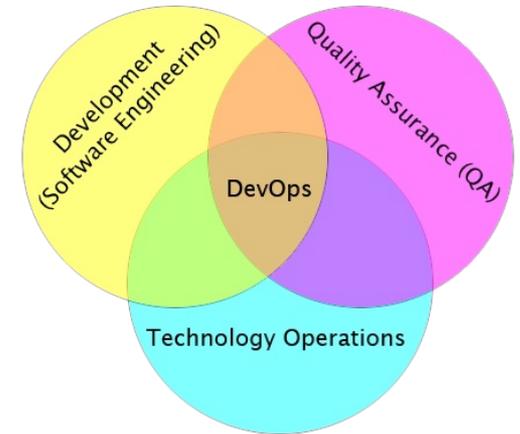
DevelopmentOperations

Ma perception : au centre de tout ce qui est nécessaire pour mettre un service ou un outil à disposition.

Wikipedia :

« Devops est un **mouvement visant à l'alignement de l'ensemble des équipes du système d'information** à commencer par les "devs" (chargés de faire évoluer le système d'information) et les "ops" (chargés d'exploiter les applications existantes).

Ce que l'on pourrait résumer en **travailler ensemble** pour produire de la valeur pour l'entreprise. Dans la majorité des entreprises, la valeur sera économique mais pour d'autres elle sera sociale ou morale. »



Mon souvenir des jdevs:

Un moyen de convergence entre les ASR/Dev :
- les outils de gestion de source
(utiles pour stocker les configuration)

(... et peut-être des outils d'automatisation ?)

T6.GT03

Usine logicielle – kesako ?

Chaîne de production où les tâches répétitives seront automatisées comme le lancement régulier de la compilation, des tests unitaires, du déploiement.

Une usine logicielle a pour objectif de fournir un support de travail collectif et aider à rendre la solution plus robuste (qualité).

→ Pour les fainéants ou ceux qui ont plusieurs milliers / millions de Lignes de code (LOC)

(« Nécessité » pour maintenir et faire face à la survie du prototype.)

Usine logicielle -

Gestion de version

- VCS (CVS, Subversion, ...) / DVCS (Mercurial, Git, Bazaar, ...) / frontal pour les accès-Forge (GitHub, Bitbucket, GitLab, Kallithea, Redmine, Trac, Bugzilla, ...)

Construction du job

- Build (CMake, Ant, Maven, NAnt, SCons, Gradle, Gulp, Grunt ...) /
- Doc (DOxygen, Javadoc, Sphinx, ...)
- Test (XUnit, Checkstyle, Sonar, FindBugs, CppCheck, JDepend, Selenium,...)

Intégration continue

- Cruisecontrol, Jenkins, Buildbot, GitLab CI, ...
- Travis-ci, Heroku, Codeship, CircleCI, Drone.io, ...

+ infra

... la suite...

Virtualisation

avec

Eric