

Comment favoriser l'approche DevOps ?

<https://etherpad.in2p3.fr/p/JDEV2015.T6.GT03>

Johan Moreau

IRCAD/IHU

2 juillet 2015

Entre 1000 et 1 million de LOC

Les petits :

- Annuaire de laboratoire en microservices,
- Conteneur de backup,
- Outil de gestion de médicaments pour animaux, ...

Les gros :

- Logiciel de traitement d'image pour la modélisation anatomique
- Logiciel de suivi intra-opératoire de réalité augmentée, ...

Problématiques différentes (performance, distribution, ...) et des choix différents

Problématique de déploiement

Clients lourds :

- mode de déploiement traditionnel (msi, dmg/pkg, deb/rpm, tar.gz, ...)
- il faut avoir toutes les cibles avec toutes les subtilités (fr/en/..., 32/64, ...)
- impose des outils de déploiement

Clients légers :

- tout ce fait sur le serveur, mais demande une bonne gestion des navigateurs
- certaines technologies ne sont pas encore bien adaptées (3d, mobile, ...)

Ils semblent difficiles de cohabiter :

- Suivant les structures les missions des ASR (DSI/IT) sont plus ou moins fixes (limitées)
- A l'inverse l'IE/IR travaille sur des missions spécifiques
- Les outils sont aussi différents (besoin général vs besoin spécifique)

Comment faire pour faire converger ces rôles afin de les faire travailler ensemble ?

Les notes de l'etherpad des participants 1/2

- Souvent les équipes de développeurs sont isolées des ASR (Administrateurs Système et Réseau)
- Périmètre fixe pour les ASR dans certains labo
- Redmine permet de tagger des tickets comme des histoires. Ce sont des tickets parlants pour l'utilisateur. Les autres tickets sont réservés aux développeurs
- Pour une release on aura entre 5 et 10 histoires et pour chaque histoire cela représente de 4-5 tickets pour le dev

Un moyen de convergence entre les 2 rôles : les outils de gestion de source (qui peuvent aussi être utile pour stocker les configuration).

Les notes de l'etherpad des participants 2/2

- différence d'organisation entre ASR et dev.
- ASR ont plutôt la culture de l'encours (tickets de support) et les dev ont une approche globale
- Future approche : on programme les machines (ex : 18000 machines / ASR chez facebook) -> convergence des deux domaines ASR et dev
- Pour les dev : les nouvelles applis sont de plus en plus sous forme client-serveur ou avec des problématiques de réseau et système
- Saltstack ou Ansible = outil qui permet d'automatiser des déploiements sur des serveurs ou même sur des postes clients - pour ASR et Dev grand intérêt de rentrer dans cet outil
- Les conteneurs sont très utiles pour cette intégration devops (car oblige aussi les recettes de déploiement)