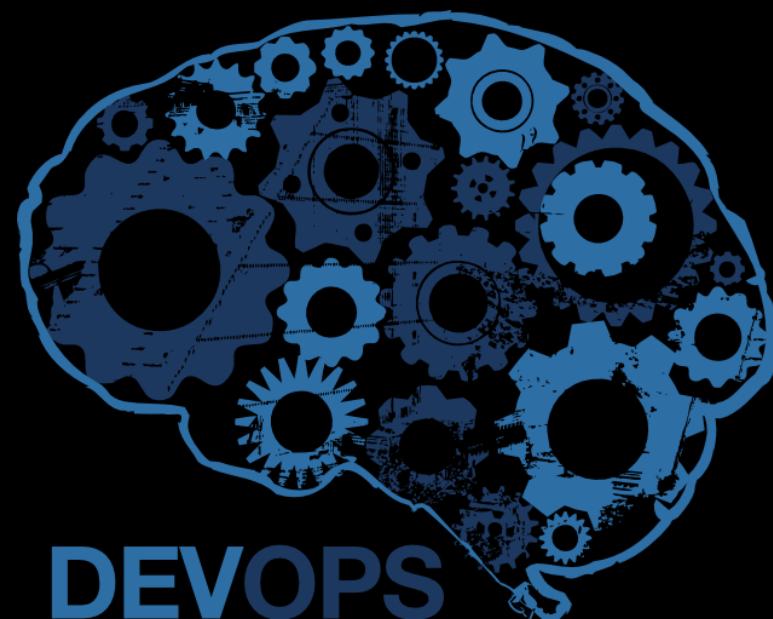


Le mouvement devops



Gildas Le Nadan

10 ans + production
informatique

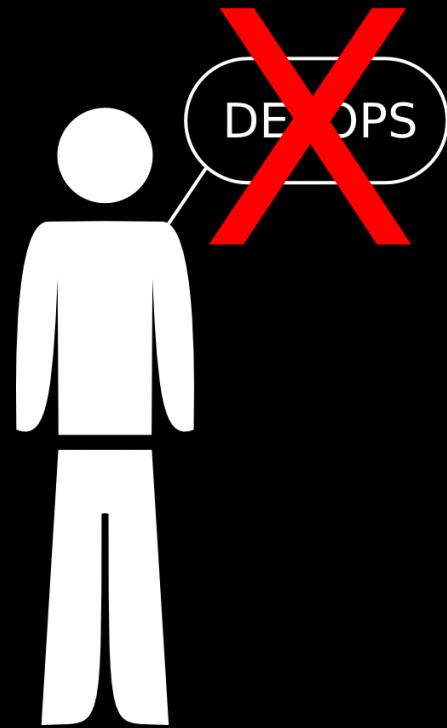
Freelance

Du
jeu de lego®

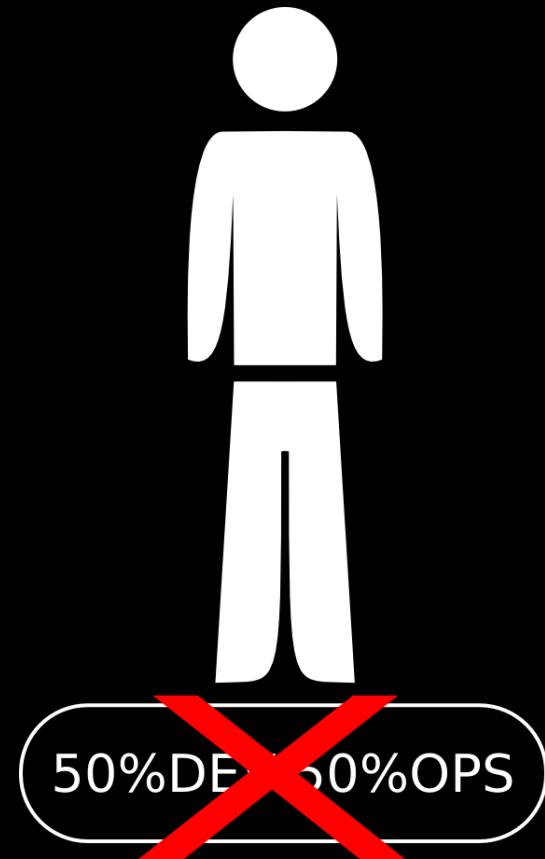
aux
discussions à la machine à
café

Devops

**méprises, rumeurs et autres
croyances erronées**



Devops n'est
ni un rôle
ni une étiquette
ni un label



**Pas besoin de savoir faire les
deux métiers**

Devops

Un mot-valise

**Pas uniquement un problème
de techniciens (devs, ops)**

Quelle
définition
de
Devops?

Devops
est un mouvement
visant à
**l'alignement du SI sur les
besoins de l'entreprise**

(à commencer par l'alignement
développement/exploitation)

Pourquoi
devops ?

A ses débuts, l'informatique
d'entreprise était

**un incontournable
différentiateur!**

➔ mise en place coûte que
coûte

De nos jours, l'informatique
d'entreprise est perçue
comme

un centre de coûts!

➔ mitigation par ITIL, etc

**Le succès des grands du web
est la preuve que**

**l'informatique peut toujours
être un différentiateur!**

➔ apparition de devops!

Aux sources de devops

Web

OpenSource

Agile

!

Web

**J'ai eu une révélation. La raison
pour laquelle #devops a émergé :
ceux qui ont fait le web il y a 10 ans
et + sont désormais aux
commandes.**

@brianlmoon

~~Serveurs~~

Services

MTBF

Mean Time Between Failure

MTTD

Mean Time To Diagnose

MTTR

Mean Time To Repair

**Le swap : pour tous les moments où
vous désirez une baisse massive
des performances de manière plus
ou moins silencieuse au lieu d'un
plantage et d'un retour de service
rapides.**

@jordansissel

MTTC

Mean Time To Communicate

webmetrics

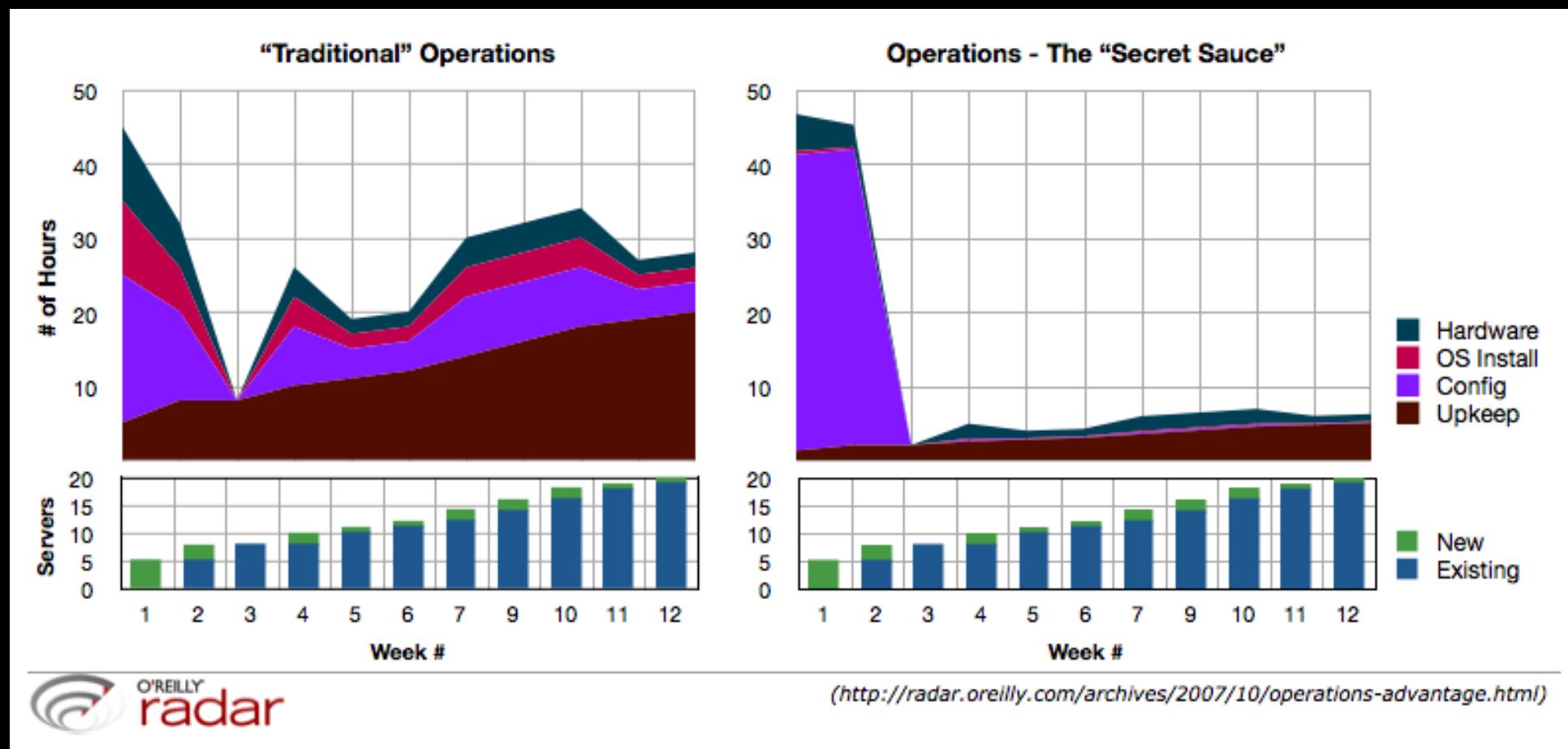
Les leçons du web

- Serveurs = **silos**
- Orienté Service
- **Keep It Simple Stupid**
- Couplage faible
- Self Service

OpenSource

Dette d'infrastructure

Opscode, Andrew Shafer, Israel Gat



contournement_en_prod

=

dette_infrastructure++

Stack

Applicatif

Framework

Système d'Exploitation

Hardware

Pousser la complexité vers le
bas du stack

=

dette_infrastructure++

Opensource?

Nécessaire
diagnostic,
adaptabilité,
pérennité,
diminution dette infrastructure

Pas suffisant
en soi pas un critère de qualité

Agile

[Agile manifesto]

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

Agile System Administration ou les limites de l'Agile Manifesto

Dev. **vs** Prod.

Le problème de l'administrabilité et des déploiements

**Je viens tout juste d'entendre parler
d'un endroit où les bonus de la
prod sont liés à la dispo et ceux
des devs aux milestones
**#payerlesgenspourquilssebattent
#pasdevops :)****

@allspaw

Fonctionnalités produites
vs.
Disponibilité d'une application

**Créer source de valeur
(potentiel)**

vs.

**Créer de la valeur via la
fourniture d'un service
(réalisé)**

**La valeur n'existe que si le
service est rendu!**

➔ **Importance des besoins
non fonctionnels (NFR)**

Besoins non fonctionnels

Diagnostic

Packaging

Montée en charge

Stabilité

Gestion des versions

Gestions des mises à jour

Sécurité/traçabilité/auditabilité

Contrôle d'accès

Gestion de ressources partagées

...

Une application

=

2 classes d'utilisateurs

- ➡ Administrateurs
- ➡ Utilisateurs finaux

**Vos fichiers de config et vos logs
sont souvent les seules interfaces
utilisateurs des admins. Pensez à
l'expérience utilisateur associée
aux fichiers XML et aux exceptions.**

#devop

@ripienaar

Solutions aux problèmes de déploiements

- ➔ communication
- ➔ packaging
- ➔ automatisation / CI
- ➔ information radiators
- ➔ etc...

Devops

Pas de dogme
Pas de manifeste

Des valeurs

4 axes

Culture

Automatisation

Mesure

Partage

(adapté du “CAMS” de John Willis & Damon Edwards)

Partage

Vision commune
Problèmes communs
Glossaire commun
Connaissance commune
Outils communs

Importance du feedback

(Eat your own dogfood!)

Measure

[ici des graphes]

Alignement au business

➔ KPI, mesures de risque,
aide à la décision

Automatisation

Gérer 1000+ serveurs dans une salle
machine devrait être aussi facile
que de monter un placard Ikea
#devops

@geogdog

Infrastructure as Code

Andrew Shafer

Il nous faut des APIs!

Adaptation des outils utilisés en développement

**Gestion des sources, tests
unitaires, intégration
continue, ...**

Adaptation des techniques utilisées en développement

**scrum, scrumban, kanban,
lean, standup meeting,
retrospectives, pairing, ...**

Culture

Coopération
vs.
Silos

Automatisation vs. Procédures

Création d'équipe pluri-disciplinaires dédiées à des projets

“*Empowerment*”

**(responsabilisation, délégation,
autonomie, confiance, ...)**

Innovation
Créativité
Fierté
Plaisir
Attribution
Reconnaissance

Devops

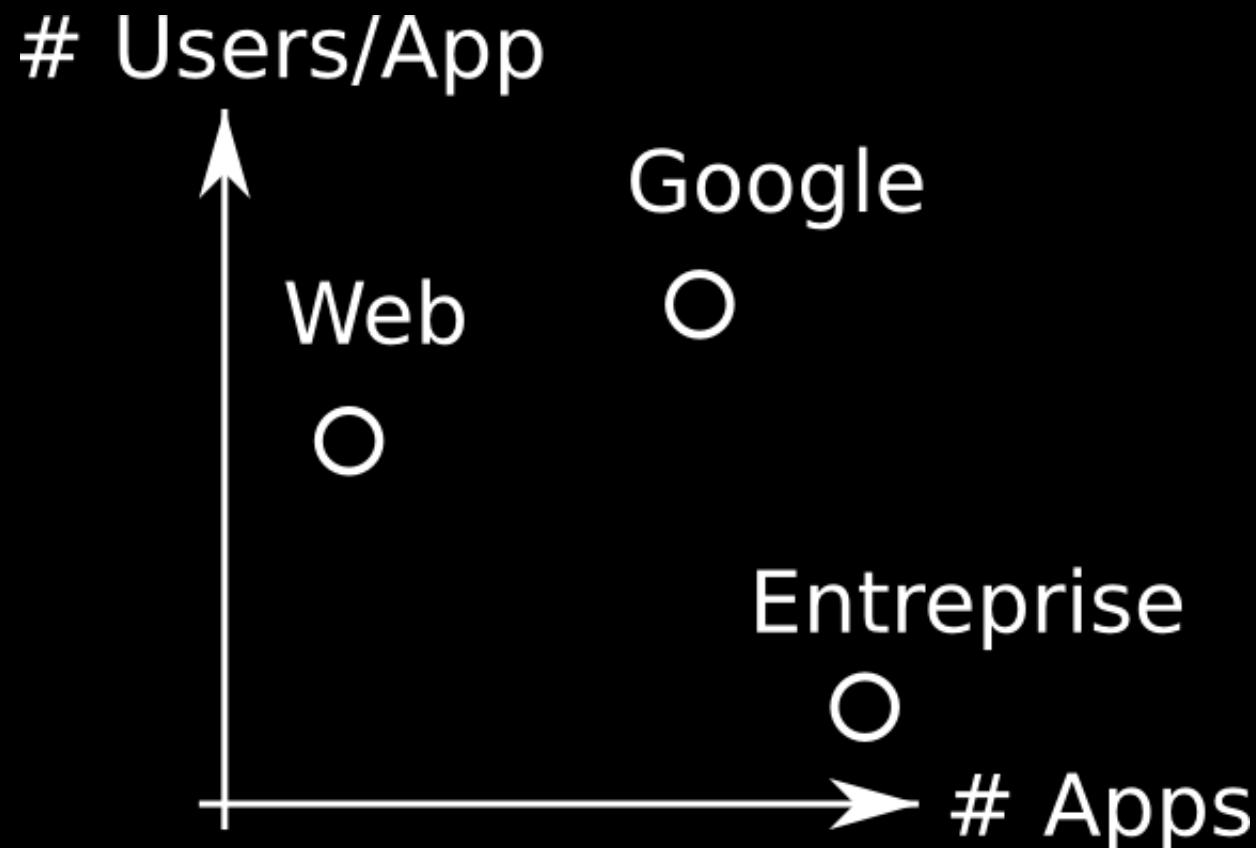
les chantiers
actuels

Mettre en place une approche
devops dans une entreprise
“traditionnelle”?

“*Web natives*”

vs.

entreprises “classiques”



Monitoring!

Orienté service
Adaptable
Qui monte en charge

Packaging

(distributions et applications externes)

Systématisation des tests de charges dans l'intégration continue

**“OS Boursouflés” et
frameworks qui n'assurent
pas!**

Questions?

En savoir plus

<http://www.devopsdays.org/>

<http://www.devopscafe.org/>

<http://www.planetdevops.net/>

<http://devops.fr/mixit/slides.pdf>