

Ch2- Le courant DEVOPS

Licence pro DEVOPS

IUT Lyon1

V. Deslandres

Références

- <http://devops.fr/>
- <http://www.timspirit.com/publications/livres/la-revolution-devops/>



*Mars 2016, pour
débutants*

Enjeux de Devops

- Témoignages fréquents... *(à votre avis, de qui ?)*
 - « Entre le moment où *on veut livrer* et la *livraison*, il s'écoule un mois ! »
 - « Mes Dév font la livraison en Prod, car la Prod ne sait pas le faire... »
 - « Je trouve que les Dév font n'importe quoi en Prod »

Terme créé par Patrick Debois en 2009 belge

Devops, plus court que 'Agile System Administration'

DEF « DevOps est la livraison agile de systèmes logiciels »
avec une déclinaison culturelle et technique (Rob England)

Présentation



- **DevOps** vise à raccourcir les **délais de livraison**
- **Comment ?** En rapprochant les équipes de développement, de recette et de production
 - Obj.: livrer plus vite les nouvelles applications tout en améliorant leur qualité
- **Devops** permet de **répliquer** l'infrastructure de production
 - disposer d'une configuration de test efficace, proche des conditions réelles d'exploitation
 - Automatiser le déploiement

DEV & OPS : des besoins contradictoires

Nouvelles fonctionnalités, amélioration innovation

Culture du produit



→ **Augmenter** le nb des mises en production et les livrer rapidement

Stabilité, Rationalisation

Culture du service



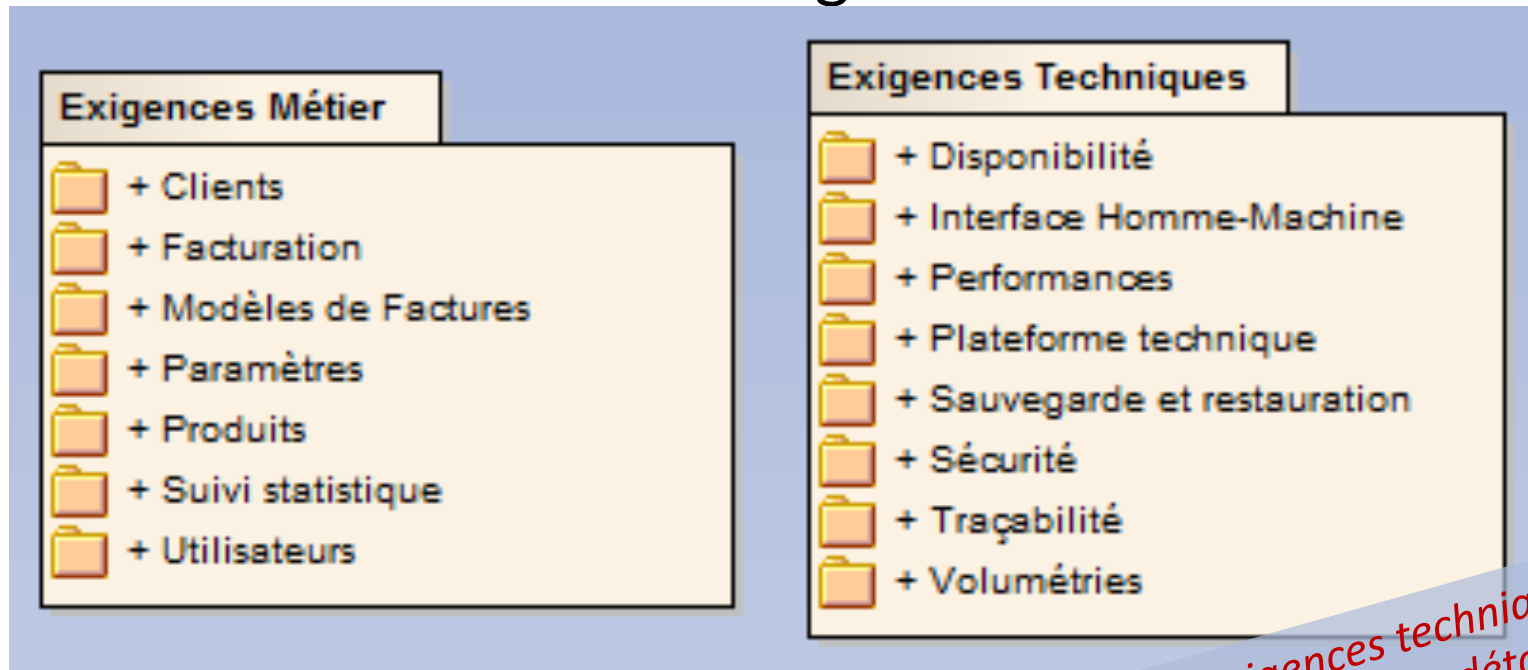
→ **Limiter** le nb de mise en production

Côté Dév

Besoins
fonctionnels

Besoins
techniques

- Considérer 2 besoins
 - Un ex. de cahier des charges :



*Exigences techniques :
souvent peu détaillées*

Côté Dév



- Les besoins utilisateurs sont le fruit de la collaboration de 3 types d'acteurs:



LES 3 AMIGOS

Côté Dév

Besoins techniques

- Alors que les fonctionnalités **techniques** sont écrites, priorisées, développées, testées, validées... uniquement par les DEV !



Ex. fonctionnalités techniques

- Mise à jour automatique du DNS
- Empêcher la duplication des noms de déploiement
- Échec rapide si opération non autorisée
- Mise à jour automatique des variables d'environnement
- Retour à la version précédente (rollback) manuelle
- Auto-rollback en cas d'échec d'une nouvelle livraison
- Améliorer les performances de copie de BD
- Etc.

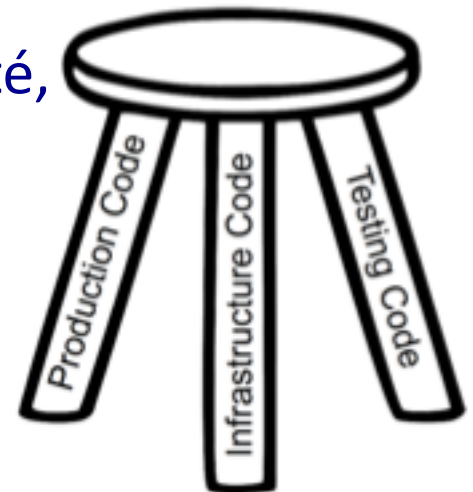
Coté Production

Autres exigences : par ex. passage à l'échelle, faible couplage entre composants, etc.

D'autre part, on code... « à l'ancienne »

Le code nécessaire à **l'exploitation** devrait être traité avec autant de soin que le code **fonctionnel** : considérer des flux, des plus-values, des priorités... et bien sûr du code testé, optimisé, etc.

→ Utiliser les méthodes et outils du DEV en PROD



Les avantages DEVOPS

Coté Production

- Simplifier le déploiement d'application dans leurs **diverses versions** et sur **différents** environnements
- **Automatiser** un maximum de tâches :
 - déploiements d'infrastructure
 - optimisation des environnements
- Voir **ce qui est déployé** et **où**, pouvoir facilement répéter les opérations



Illustration avec *la Méthode du blue / green* :

- Tout tourne avec une version n ;
- On teste la version n+1 sur une plateforme dupliquée : si ça marche, on bascule les users sur cette machine ;
- Si jamais il y a un pb quelques mois après, on peut revenir à la plateforme antérieure, c'est immédiat.

Arch. micro-services vs. n-tiers

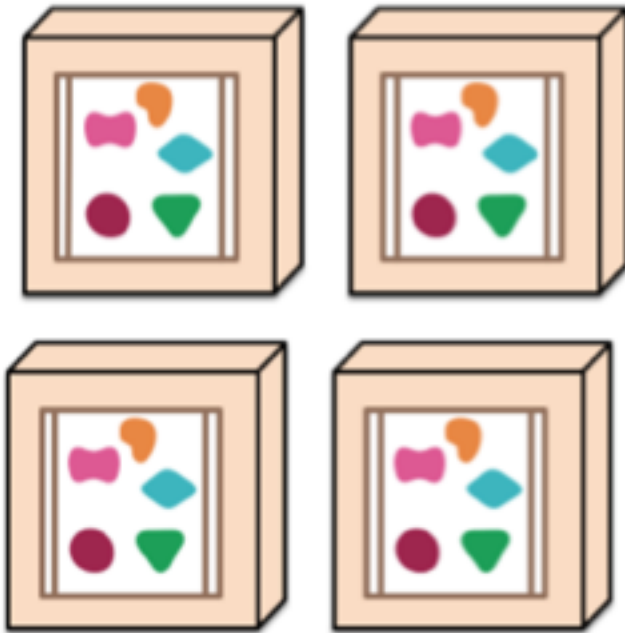
A monolithic application puts all its functionality into a single process...



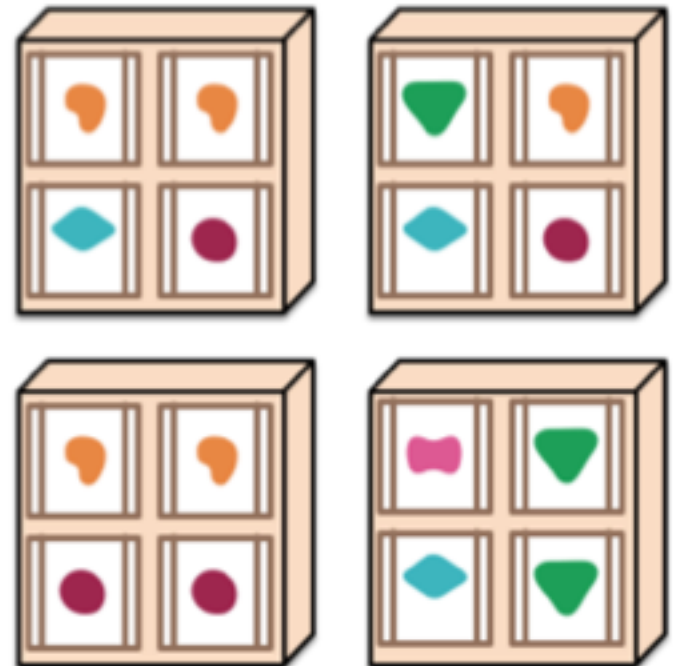
A microservices architecture puts each element of functionality into a separate service...



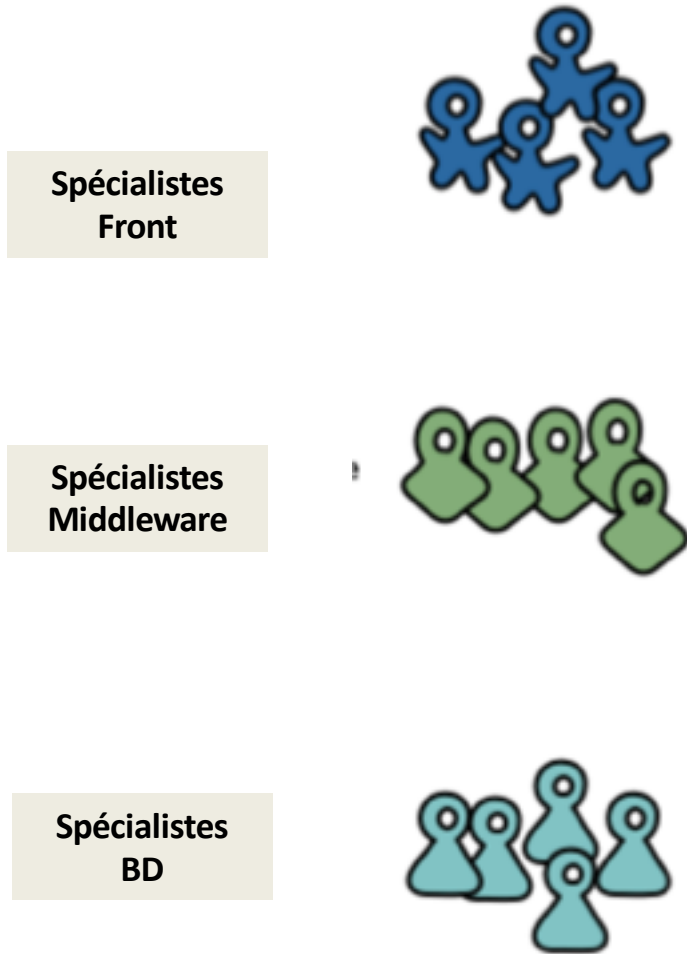
... and scales by replicating the monolith on multiple servers



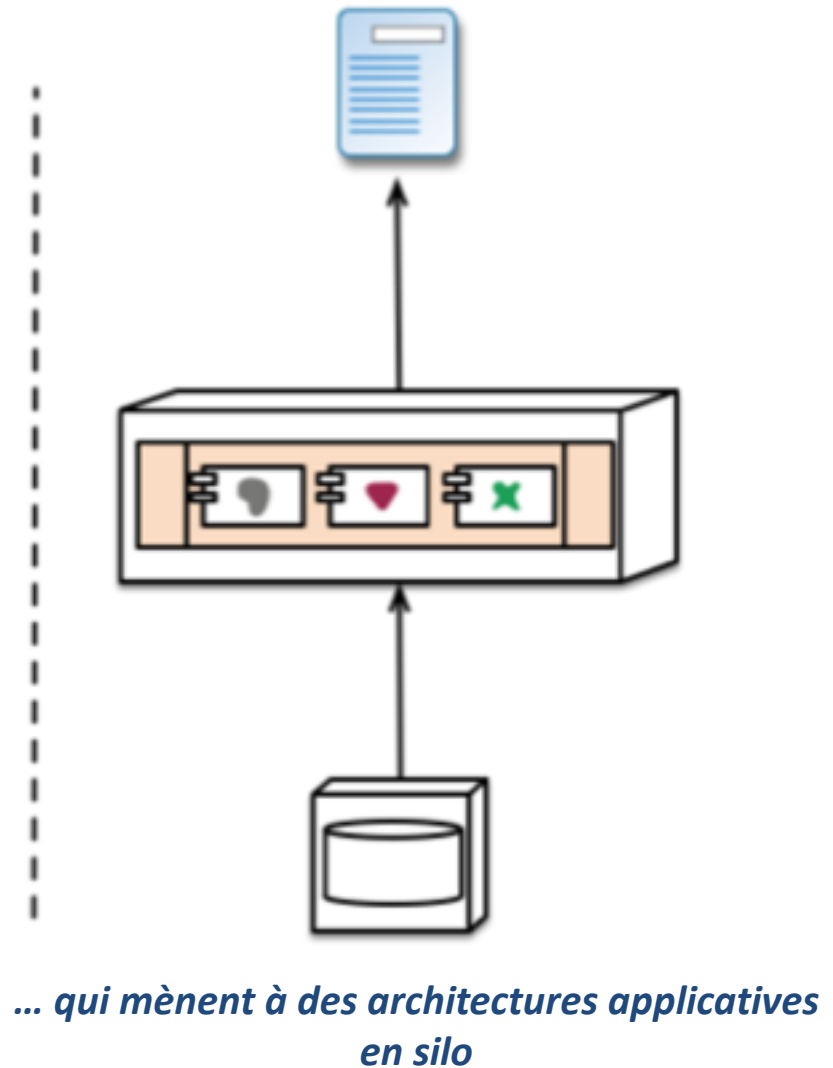
... and scales by distributing these services across servers, replicating as needed.



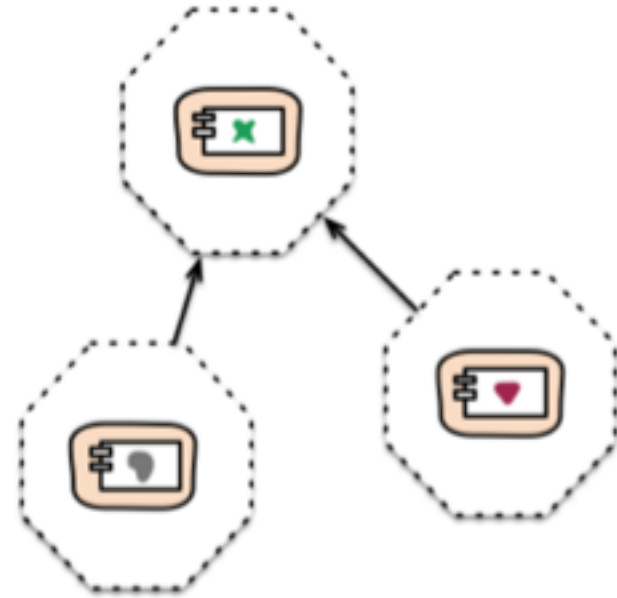
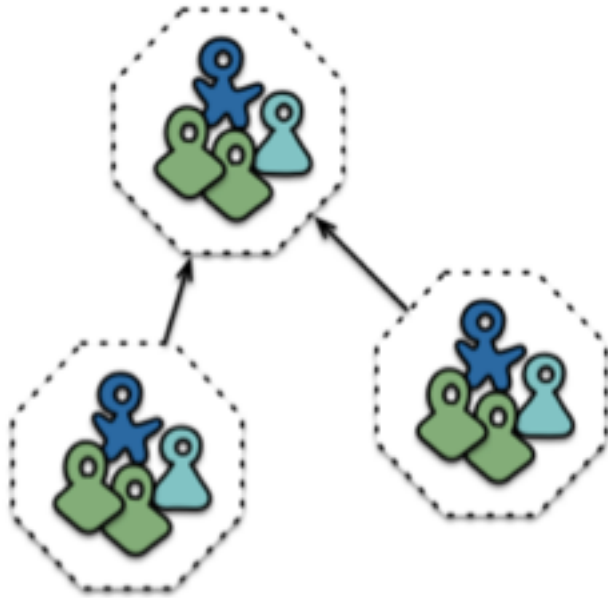
D'équipes monofonctionnelles...



Des équipes fonctionnelles en silo...



... à des équipes plurifonctionnelles



- Equipes de moins de 12 personnes
- Cross-fonctionnelles

- Services individuels opérationnels réalisant une tâche donnée
- Définis pour un contexte donné

Facilite les **très fortes montées** en charge

Car les containers sont *bootables* en quelques secondes

Technologies DEVOPS

- **Application Performance Management**
 - **Stackify Retrace** : la meilleure APM
 - permet de voir les pbs
 - à partir de 10€/mois
 - Et aussi : **NewRelic**, **AppDynamics** (free)
- **Gestion de configuration**
 - Les outils open source de gestion de configuration **Puppet** et **Chef** sont les technologies DevOps les plus souvent utilisées (avec tous deux un taux d'adoption de 28%)
 - **Ansible** (21%) : technologie simple à mettre en place, préférée des développeurs
- **Plateformes d'entreprise**
 - Microsoft **Azure DevOps** – partenariat avec Zend
 - **Amazon s3**
 - **OpenStack**
- **DOCKER** est une techno très utilisée en DEVOPS, présente dans 37% des entreprises (contre 13% en 2015)
 - Docker déplace le pouvoir vers le développeur plutôt que vers les grands fournisseurs de plateforme du *cloud*
 - Utilisée par de nombreuses plateformes



Enquête PUPPET sur DEVOPS

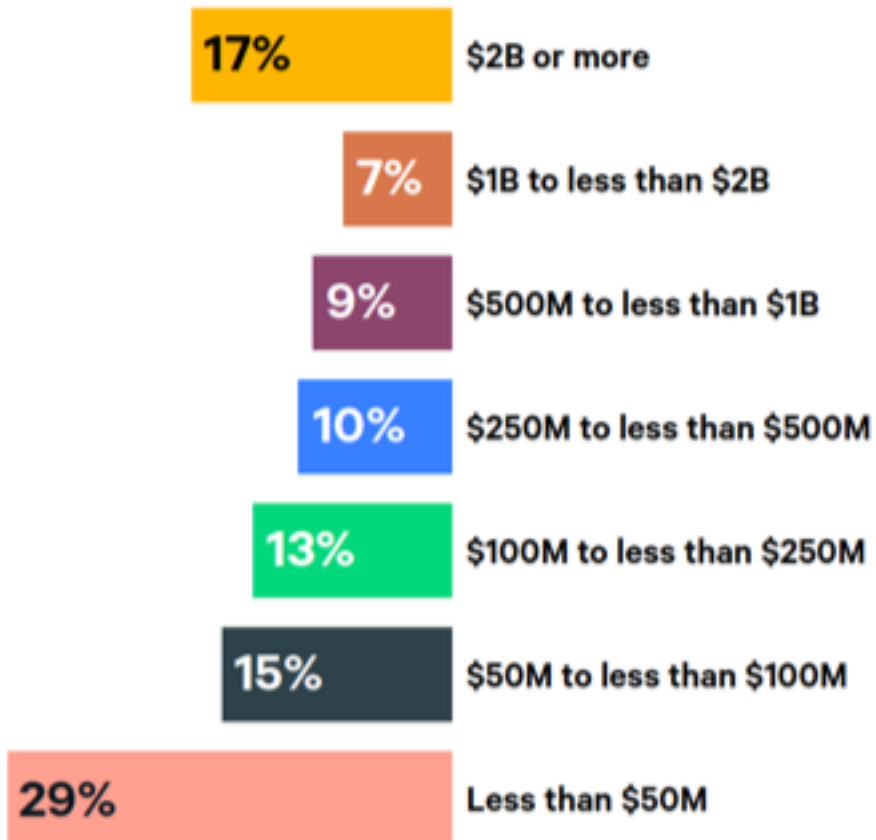
www.puppet.com

- Puppet Enterprise est une plateforme DEVOPS qui permet de distribuer, d'exécuter et de sécuriser automatiquement les infrastructures

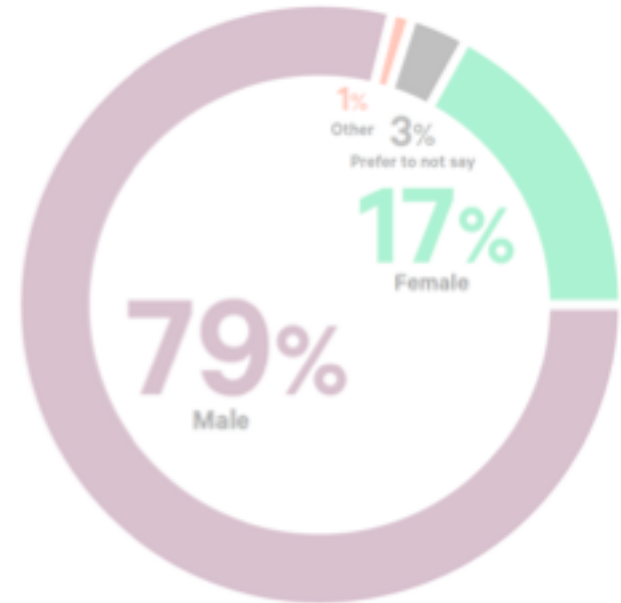


- Juin 2018, 3000 professionnels ont répondu
 - Enquête annuelle, depuis 5 ans, internationale
 - Résultats = des **chiffres + infos qualitatives** sur :
 - how to integrate security into the software development cycle;
 - the value of experimentation;
 - evaluating the ROI of DevOps;
 - how employee engagement affects organizational success
- <https://puppet.com/resources/whitepaper/state-of-devops-report>

CA des entreprises répondantes :
toutes les E. se sentent
concernées par DEVOPS

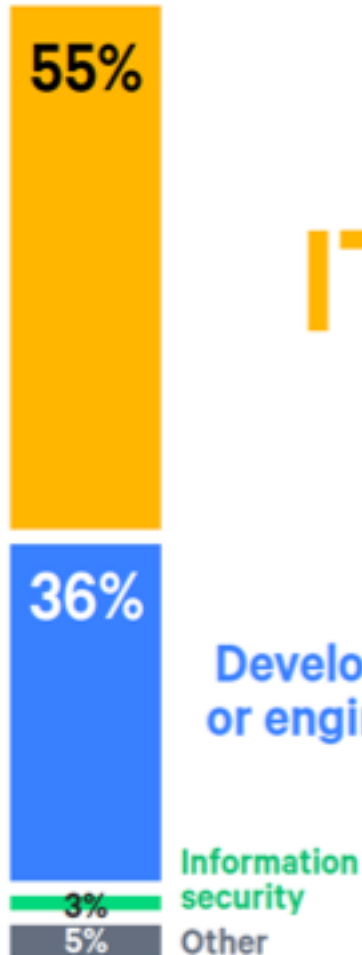


*Forte augmentation du
taux de répondants
Féminin (6% en 2017)*

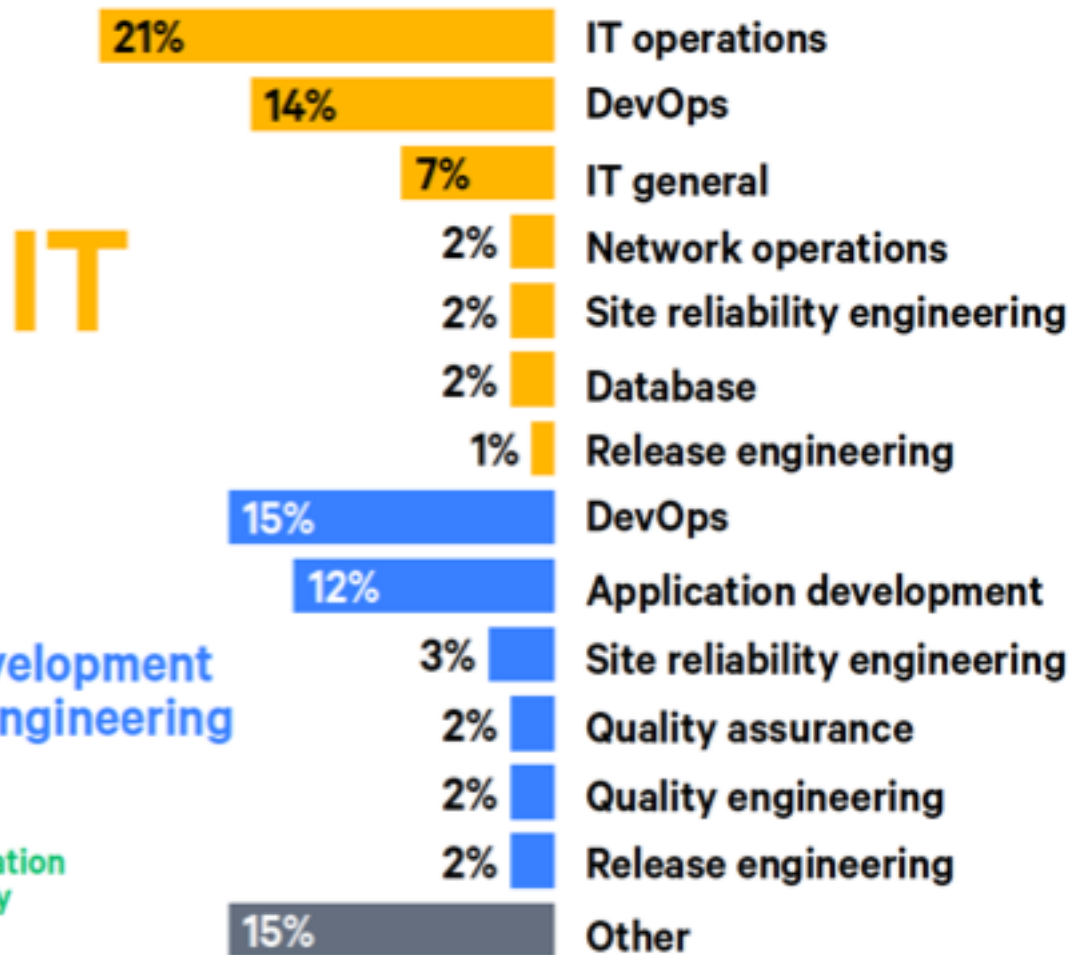


Les services
des
répondants

Department



Team



**What server OSes have you widely deployed?
Select all that apply.**

**# of
Responses**

Linux - Enterprise Linux variants (RHEL, Oracle, CentOS)	3,159
Windows 2012/2012R2	2,446
Windows 2008/2008R2	2,017
Linux - Debian/Ubuntu variants	2,016

# of OSes Selected	# of Responses	% of Total
0	25	1%
1	1,035	22%
2	1,038	23%
3	883	19%
4	636	14%
5	420	9%
6	234	5%
7	139	3%
8	86	2%

78%

78% of respondents are widely deployed on 1-4 different operating systems.

Enquête PUPPET (2016) : quels OS ?

PUPPET : les équipes performantes

	Équipe performante	Equipe medium	Equipe moins performante
Fréquence des déploiements	À la demande	Entre 1 par sem. et 1 par mois	Entre 1 par mois et 1 tous les 6 mois

PUPPET : les équipes performantes

	Équipe performante	Equipe medium	Equipe moins performante
Fréquence des déploiements	À la demande	Entre 1 par sem. et 1 par mois	Entre 1 par mois et 1 tous les 6 mois
Temps nécessaire avant mise en production opérationnelle après des modifications (<i>lead time</i>)	Moins d'une heure	Entre une sem. et un mois	Entre 1 et 6 mois

PUPPET : les équipes performantes

	Équipe performante	Equipe medium	Equipe moins performante
Fréquence des déploiements	À la demande	Entre 1 par sem. et 1 par mois	Entre 1 par mois et 1 tous les 6 mois
Temps nécessaire avant mise en production opérationnelle après des modifications (<i>lead time</i>)	Moins d'une heure	Entre une sem. et un mois	Entre 1 et 6 mois
Temps nécessaire à la restauration d'une version propre après un incident (déni de service, coupure imprévue : <i>recovery time</i>)	Moins d'une heure	Moins d'un jour	Moins d'un jour

TABILITE

PUPPET : performances

	Équipe performante	Equipe medium	Equipe moins performante
Fréquence des déploiements	À la demande	Entre 1 par sem. et 1 par mois	Entre 1 par mois et 1 tous les 6 mois
Temps nécessaire avant mise en production opérationnelle après des modifications (<i>lead time</i>)	Moins d'une heure	Entre une sem. et un mois	Entre 1 et 6 mois
Temps nécessaire à la restauration d'une version propre après un incident (déni de service, coupure imprévue : <i>recovery time</i>)	Moins d'une heure	Moins d'un jour	Moins d'un jour
Part des modifications dues à un service dégradé (<i>change failure rate</i>)	0-15%	16-30%	31-45%

STABILITE

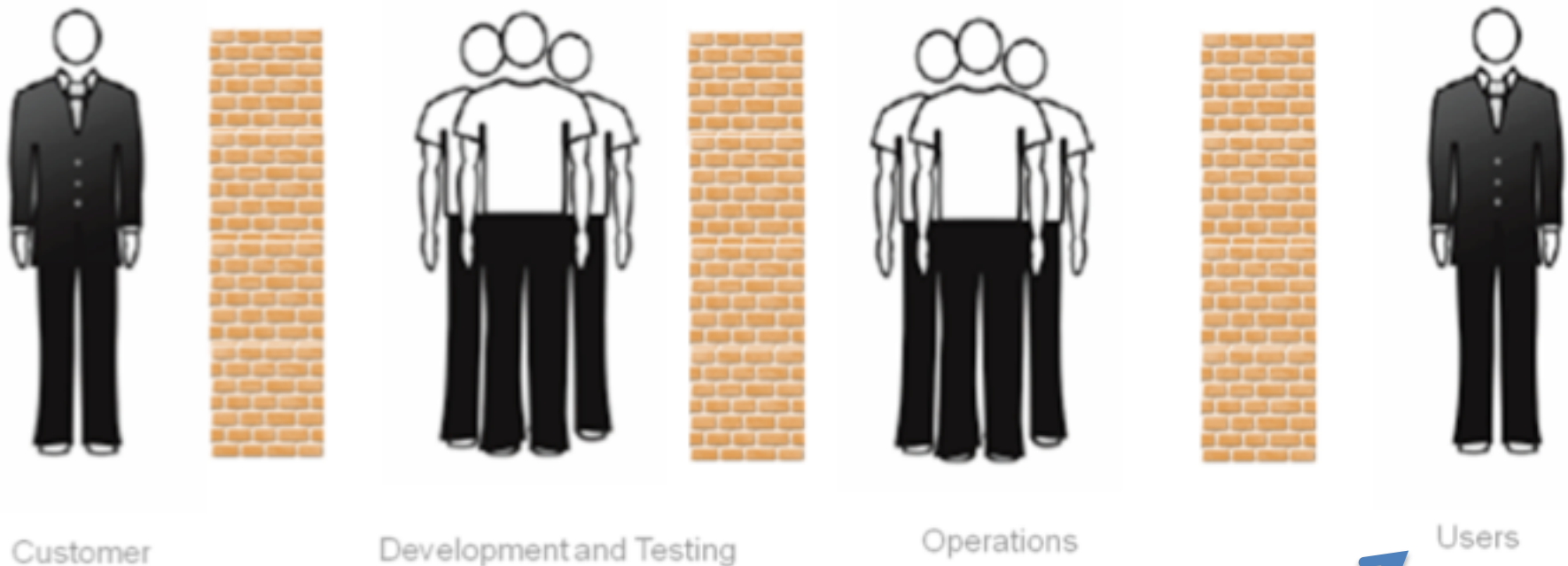
Enquête PUPPET : informations qualitatives

- Les **équipes performantes** dépensent seulement **21%** de leur temps sur du **travail non planifié** ou des **corrections de bug**
 - Et 49% de leur temps à de *nouveaux jobs* (créer des fonctionnalités etc.)
- Les employés des **équipes performantes** recommandent **2.2 fois plus leur organisation**
 - comme un bel endroit pour travailler
 - par rapport aux équipes moins performantes



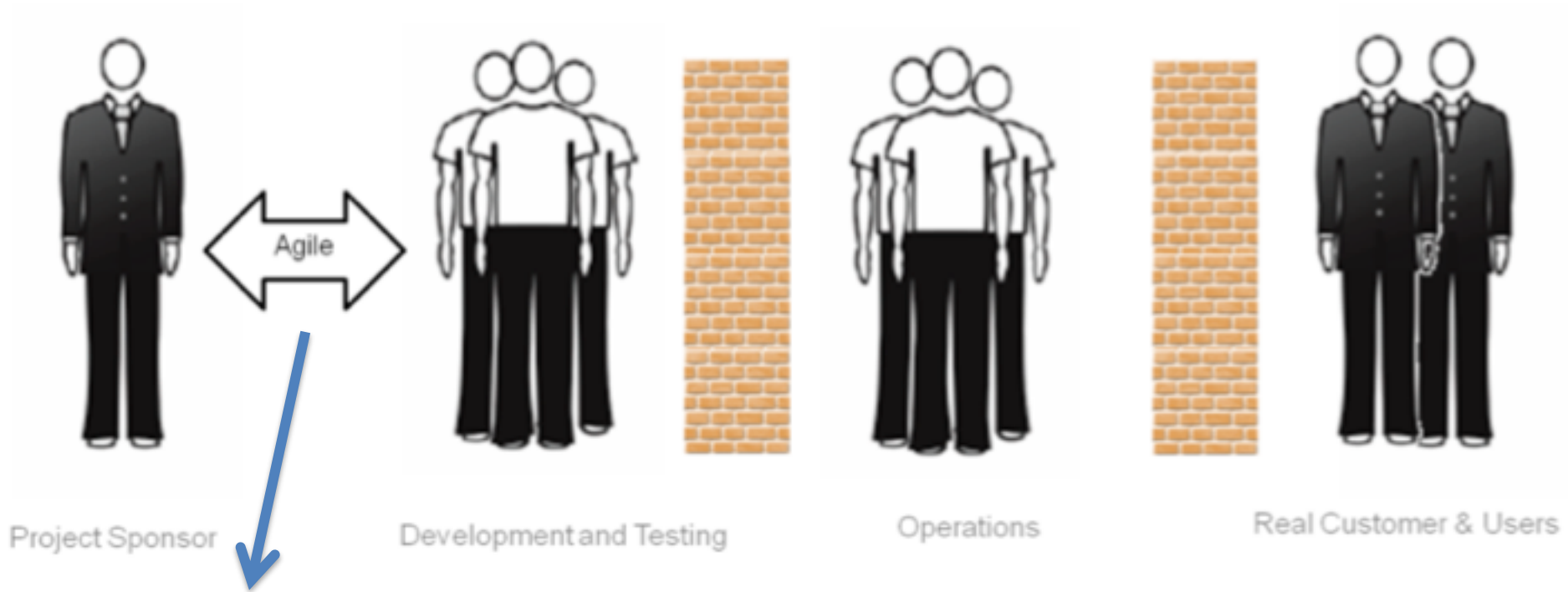
« Les murs de la confusion » du développement logiciel

Source : Henry Jacob, 2015



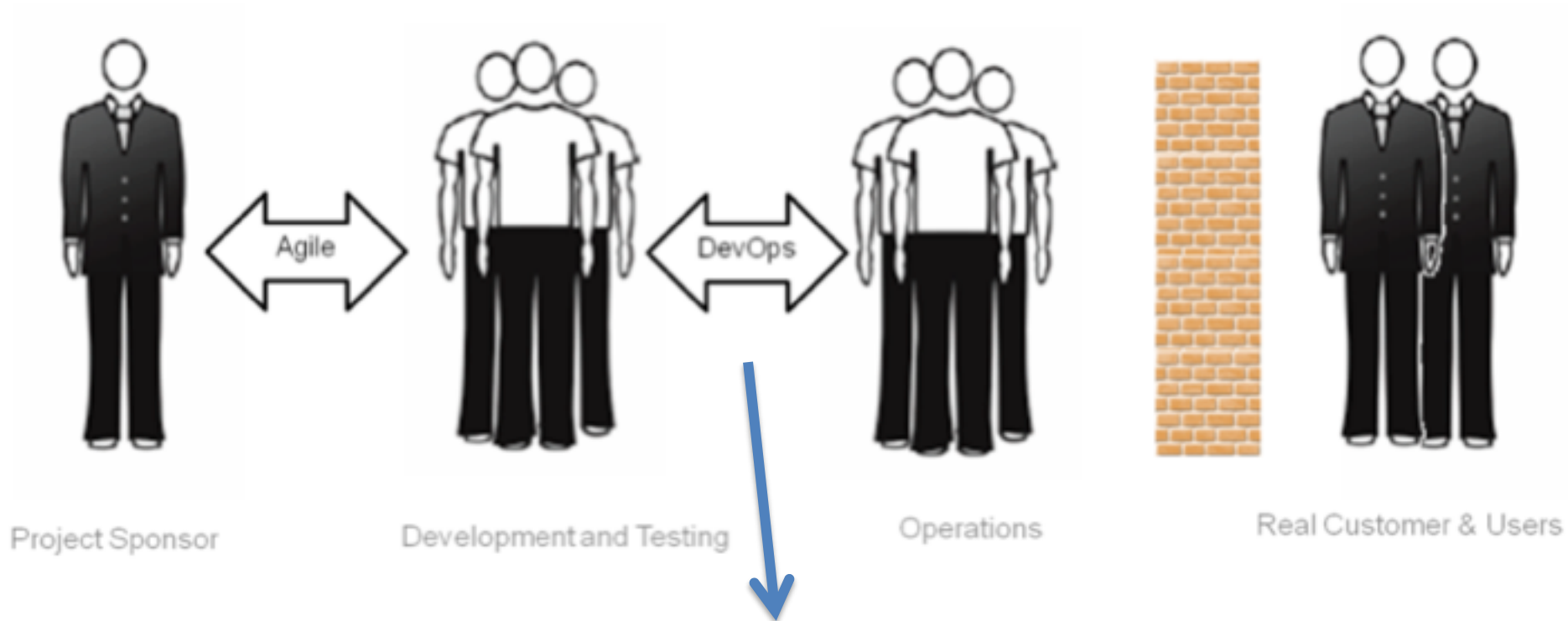
Souvent, le 'Client' N'EST PAS le bon représentant des besoins. C'est un promoteur du produit / projet, qui croit savoir ce que VEUT le client et comment les utilisateurs feront USAGE du logiciel.

« Les murs de la confusion » (2)



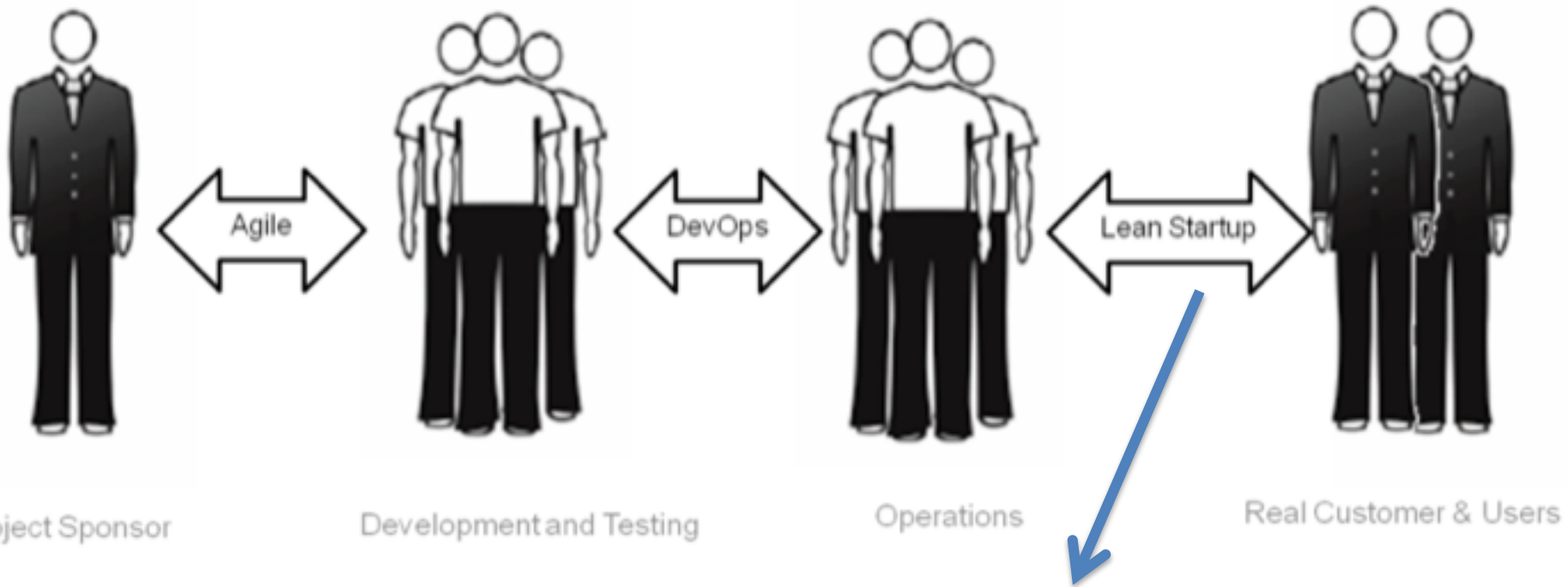
Méth AGILES : réduisent ce blocage avec des démonstrations et des cycles courts permettant d'avoir le feedback du « représentant Client »

« Les murs de la confusion » (3)



DevOps : permet aux *dév* et à la *prod* de mieux travailler ensemble, en changeant les **mentalités**, en apportant des **techniques** et des **méthodes**.

« Les murs de la confusion » (4)



Le LeanStartUp et la Livraison Continue : permet de mieux répondre aux besoins des vrais utilisateurs, en les impliquant très tôt dans la boucle.

DEVOPS : JE RETIENS...

- La **philosophie**
- Les raisons des incompréhensions entre DEV et OPS
- Les attentes de la *prod*, les besoins techniques
- Le besoin **d'automatisation** pour le déploiement
- Le lien avec **l'agilité**, le *lean*
 - L'existence des 'murs de confusion'
- L'intérêt croissant des entreprises pour DEVOPS
- Ses **avantages**