Comment la virtualisation peut-elle améliorer le déploiement d'une application?

Mise en place du problème

MÉTHODES AGILES -> LES LIMITES

Méthodes Agiles
Equipe Scrum
Feature Team

Développement / Exploitation
Un but commun -> délivrer le meilleur logiciel au client de l'entreprise
Des objectifs différents ->

- Développement: évolution du système
- Exploitation: stabilité du système

Constat

- Conflit entre les équipes de développement et d'exploitation. Une équipe d'exploitation est primée sur la stabilité du système alors que l'équipe de développement est récompensée à chaque nouvelle fonctionnalité livrée, il est évident que ces deux équipes vont se retrouver en conflit perpétuel.
- L'agilité, quand elle n'est appliquée qu'au développement, se trouve néanmoins freinée par les tâches d'exploitation qui surviennent après chaque livraison.

DEVOPS -> NOUVELLE SOLUTION

Créer une synergie entre les équipes de développement et d'exploitation.

Coopérer

Fluidifier
Usine logicielle
Infrastructure
Gestion de configuration
Monitoring et Alertes
Analyse d'incident
Mesure de la performance
Value Stream Map

Livrer

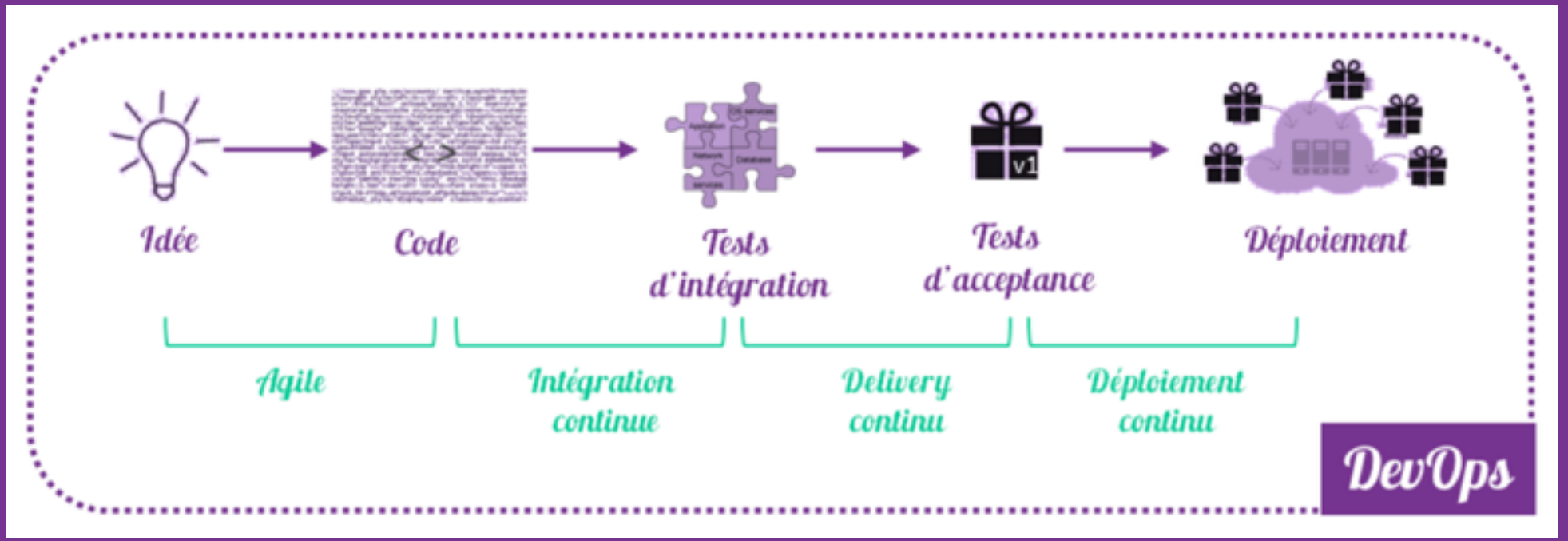
Etude de l’Art

AXA TECH

BLUEMIX (IBM)

La nouvelle génération de solution

DOCKER + ANSIBLE + OPENSHIFT + SERVEURS VIRTUELS DANS LE CLOUD


Idée
Code
Tests d'intégration
Tests d'acceptance
Déploiement
Agile
Intégration continue
Delivery continu
Déploiement continu
DevOps

BÉNÉFICES ATTENDUS

Des cycles de déploiement plus courts

Les DevOps jouent un rôle clé dans la réduction du temps du cycle de déploiement des logiciels, passant de quelques semaines à seulement quelques heures, permettant une plus grande flexibilité quant aux nouvelles fonctionnalités et changements à apporter au produit initial.

Mise à disposition de nouveaux services plus rapidement

Des déploiements fréquents associés à des délais de livraison plus rapides permettent une agilité opérationnelle.

Une satisfaction client améliorée

Grâce à des applications ciblées et de qualité, conformes aux retours clients end to end.

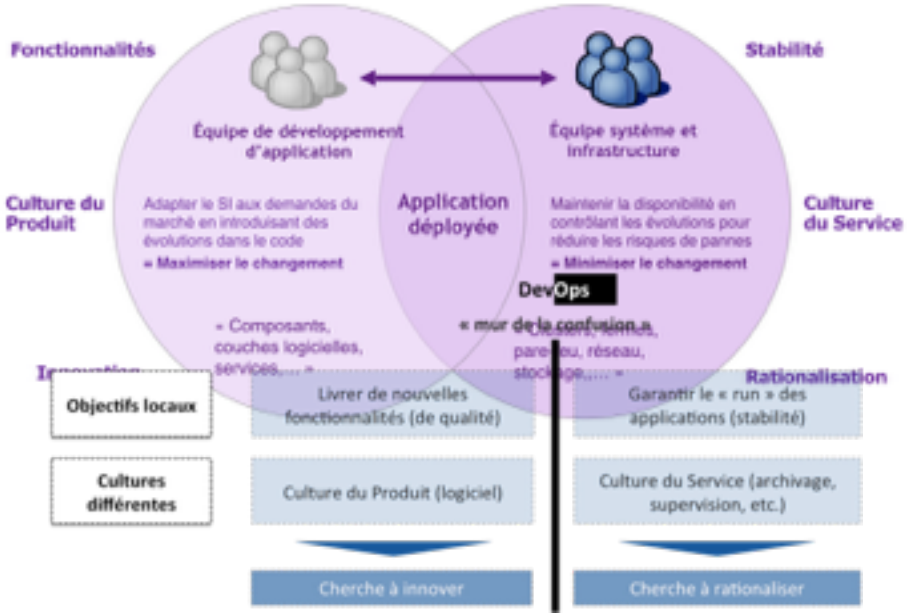
Des coûts réduits

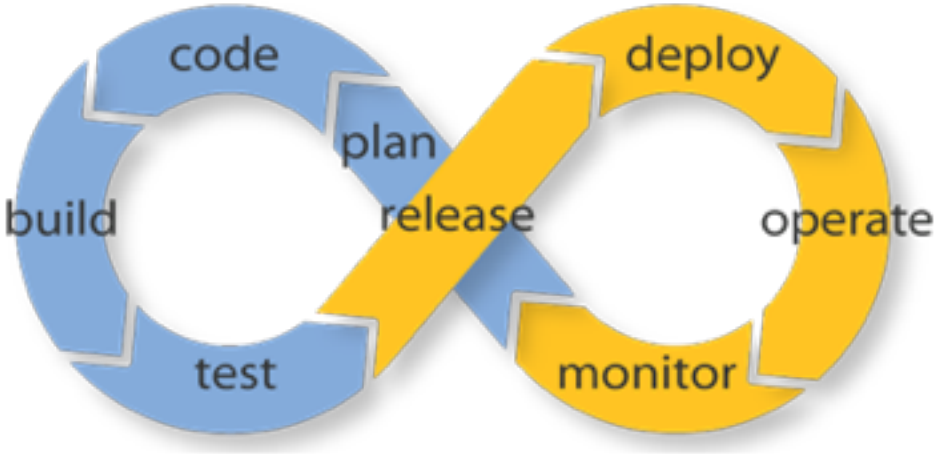
L'automatisation permet aux équipes de réaffecter des ressources précieuses à des tâches à plus haute valeur.

Conformité et Gouvernance

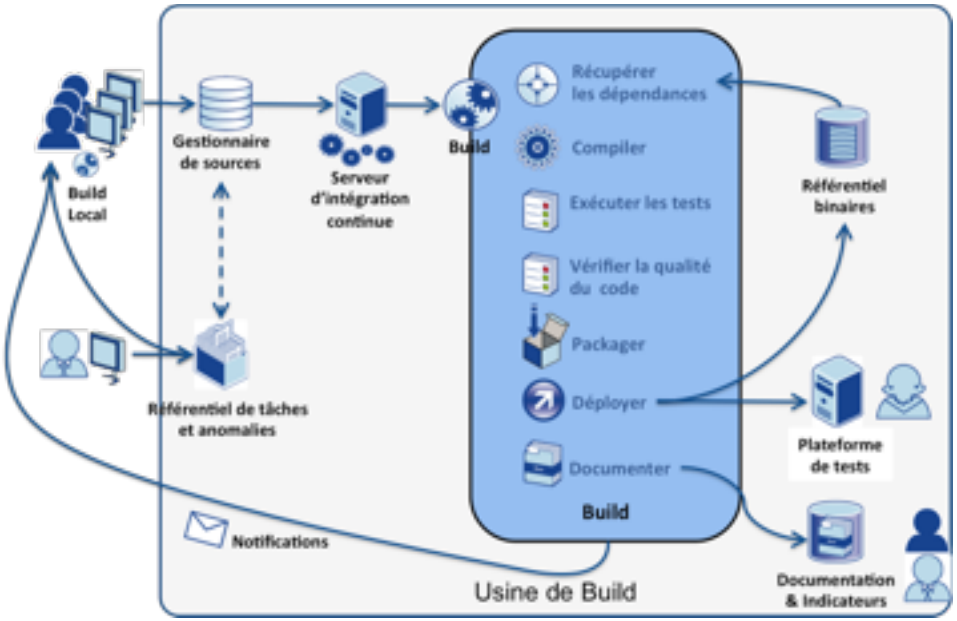
Automatisation du tracking et reporting end-to-end sur les phases de livraison/déploiement contenu.

« THE WALL OF CONFUSION »





Continuous Integration



« YOU BUILD IT, YOU RUN IT »

LES CONCEPTS CLÉS DE LA MÉTHODE

Le versionning

La gestion des dépendances

La gestion de l’environnement

Continuous Deployment

LA « BÊTA PERPÉTUELLE »

Le principe de la « bêta perpétuelle » a été introduit pour la première fois par Tim O'Reilly dans le manifeste du Web 2.0, qui expose le principe selon lequel:

> *« Les utilisateurs doivent être considérés comme des co-développeurs, en suivant les principes Open Source (...). Avec le Web 2.0, ce principe évolue vers une position plus radicale, la bêta perpétuelle, dans laquelle le produit est développé de manière ouverte, avec de nouvelles fonctionnalités offertes sur une base mensuelle, hebdomadaire, ou même quotidienne. »*

Le terme « bêta perpétuelle » désigne le fait que notre application n'est jamais finalisée. Elle s'absout des contraintes habituelles liées au cycle de développement en « release » au profit d'une livraison en continu des nouvelles fonctionnalités.

Release early, release often

Derrière ce nouveau concept se cache un concept déjà bien en place chez les agilistes (du point de vue de l'itération courte) et dans le monde de l'Open Source (du point de vue de la récolte continue du feedback), le « Release early, release often », traduit en français par « Publiez tôt, publiez souvent ».

Cette pratique a été décrite par Eric Steven Raymond dans "La cathédrale et le bazar" où il formulait explicitement:

> *« Publiez tôt. Publiez souvent. Et écoutez vos clients »*

Cette méthodologie vise à réduire les temps de développement et améliorer l'implication de l'utilisateur dans la conception du logiciel afin de créer un produit correspondant à ces attentes et ainsi éviter la création d'un logiciel que personne n'utilisera.

Les services en ligne (Software As A Service)

Le concept de « bêta perpétuelle » a été rendu possible grâce au Cloud Computing et plus particulièrement à la généralisation du service en ligne ou « Software As A Service ». L'hébergement de l'application par l'éditeur permet d'absoudre ce dernier au traditionnel cycle de déploiement d'un logiciel et de ne gérer qu'une seule version de son application.

Les services en ligne sont continuellement mise à jour sans pour autant en informer l'utilisateur. Les nouvelles fonctionnalités, découverte au fur et à mesure par l'utilisateur, permettent un apprentissage progressif des nouveautés applicatives.

La « Customer driven roadmap »

L'hébergement de l'application sur serveur offre à l'éditeur une maîtrise total de sa plateforme de production. Il peut ainsi mettre en place des sondes analytiques afin de récolter des informations sur l'usage de notre application et l'accueil réservé à nos nouvelles fonctionnalités par l'utilisateur.

Les pré-requis

La mise en place d'une stratégie de « bêta perpétuelle » requiert certains pré-requis pour en garantir le succès:

- une intégration continu,
- une livraison continue,
- un déploiement continu,
- une stratégie de type « One click deployment / Rollback » pour une restauration rapide de l'application au dernier état stable.

Conclusion

Le concept de la « bêta perpétuelle » est présent chez de nombreux géants du Web tel que Google, Facebook,

Amazon, Twitter, Flickr… Peu en font mention dû à la mauvaise image du terme « bêta », qui pour la conscience collective, se réfère à un produit non fini et peu fiable.

Prenons en exemple Gmail, la boite aux lettres mails développé par Google, qui jusqu'en 2009 intégrait la mention « bêta » dans son logo. De petites fonctionnalités unitaires sont fréquemment proposés aux utilisateurs. En fonction de leur niveau d'adoption Google les intègrent ou non à la version standard de son service.

ZERO DOWNTIME DEPLOYMENT

Les patterns

Blue/Green Deployment

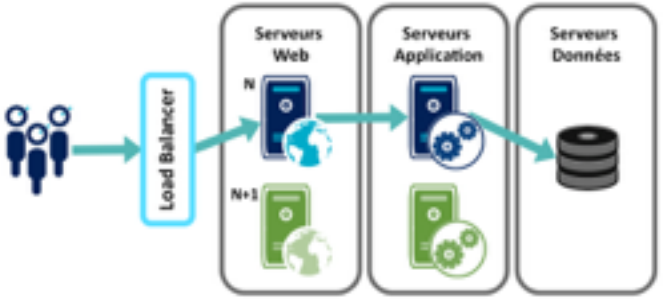
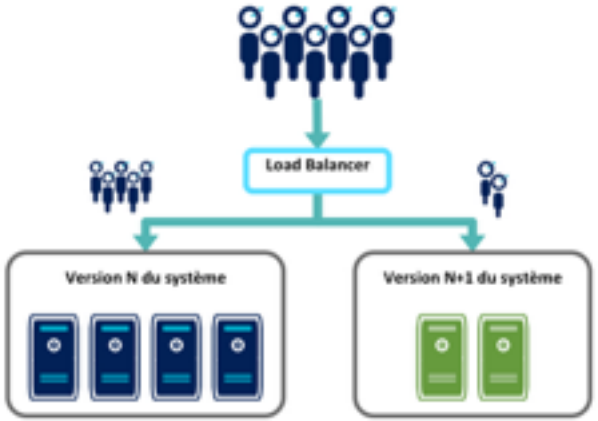


Schéma de pattern Blue/Green Deployment

Canary Release



Dark Launch

FLIPPING FEATURE