# DevOps & Qualification continue

*Outils et principes DevOps dans la mise en œuvre d'une plateforme de qualification continue*

Auteurs : Jean-François Parguet, ASIP Santé
Patrick Mortas, Henix

Journée Française des Tests Logiciels 2016
Comité Français des Tests Logiciels
ISTQB

# Sommaire

- **[Contexte]** La qualification à l'ASIP Santé
- **[Technique]** Démarche de mise en place d'une plateforme de Qualification Continue DevOps
  - Intégrer en continu
  - Provisionner les environnements
  - Livrer en continu
  - Orchestrer le pipeline
  - En perspective, automatiser en continu
- **[Retour d'expérience]** Contrôler l'externalisation des développements à l'ASIP Santé
  - Les rôles et responsabilités
  - Les gains
  - Les points de vigilance
  - Les perspectives

Contexte

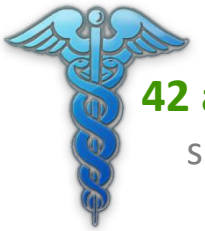
# La qualification à l'ASIP Santé

# Le contexte

- Un grand compte public déléguant dans le cadre de prestations sous-traitées :
  - → A des équipes projet dédiées au développement de nouvelles applications;
  - → La maintenance des applications en production à une TMA ;
  - → La recette et l'intégration à une équipe de TRA ;
  - → L'exploitation à un infogérant,
- Souhaite renforcer la maîtrise de la sous-traitance :
  - → En imposant l'utilisation de sa propre usine de développement dans les appels d'offres (en cours de généralisation)
  - → En fournissant une infrastructure industrialisée pour la mise en œuvre des tests.
- Pour satisfaire à :
  - → une exigence d'évolutions rapides (La réglementation évolue en fonction des décisions politiques)
  - → Une nécessaire continuité de service (au-delà des SLAs de production)

# La TRA de l'ASIP Santé (depuis 2012)

**2 millions de lignes de code**
(Java, C, PL/SQL, PHP,…)



**42 applications gérées**
sur 4 S.I. : CPS, RRPS, RASS, MSS

**Entre 5 et 17 ETP**
mobilisés en TRA

**332 Campagnes**
de qualification réalisées

**8 outils dédiés aux tests**
tests de performance, automatisation, qualité du code, sécurité, patrimoine documentaire de test

**Un patrimoine de 8 500 cas de test**
(20% techniques, 80% fonctionnels)

# Problématique et objectifs

- La problématique :

→ Un cycle de production logicielle scindé en deux phases : l'intégration (continue) d'une part et la qualification traditionnelle de l'autre.

→ Aucun test (ou presque) avant la fin des développements.

→ Des environnements (très) différents de ceux de production avec des délais conséquents de mise à disposition.

→ Dans le cadre de développements externalisés, des charges d'initialisation très lourdes à chaque changement de sous-traitant.

- Les objectifs :

→ Tester plus vite pour éviter les effets tunnel à chaque étape de la production logicielle

→ Tester plus régulièrement pour avoir plus de visibilité sur la qualité de l'application

→ Accélérer les cycles de qualification en améliorant les conditions d'exécution

- Les moyens :

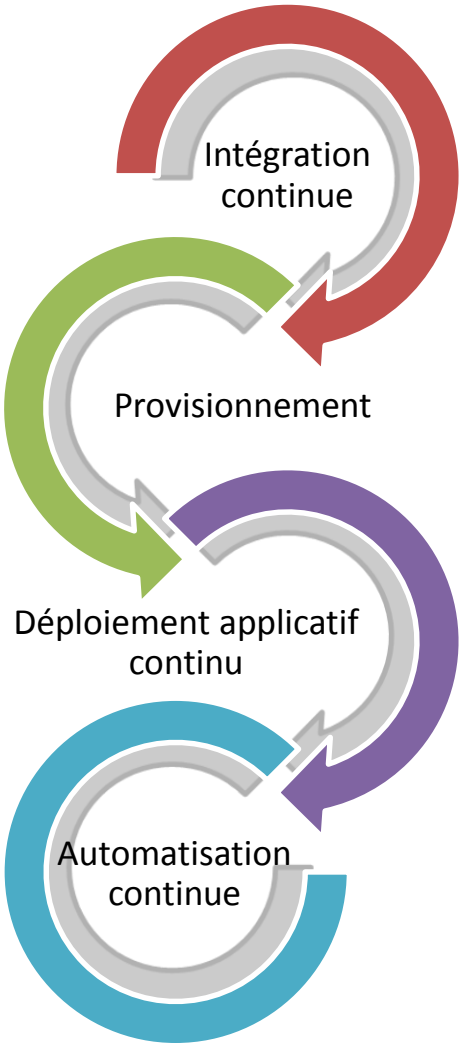
→ Mettre en place une plateforme de qualification continue avec des outils open source issus de DevOps.

→ Redéfinir les rôles et responsabilités des différents acteurs du cycle de production logicielle

Démarche technique

# METTRE EN PLACE UNE PLATEFORME DE QUALIFICATION CONTINUE

# INTRODUCTION – *LES BRIQUES DE LA PLATEFORME DE QUALIFICATION CONTINUE*




❶ Mise en place d'une usine logicielle pour intégrer en continu les sources et être en capacité de livrer vite et régulièrement.

❷ Provisionnement d'environnements standardisés en libre-service pour rapidement et à la demande mettre en œuvre les conditions nécessaires à la qualification.

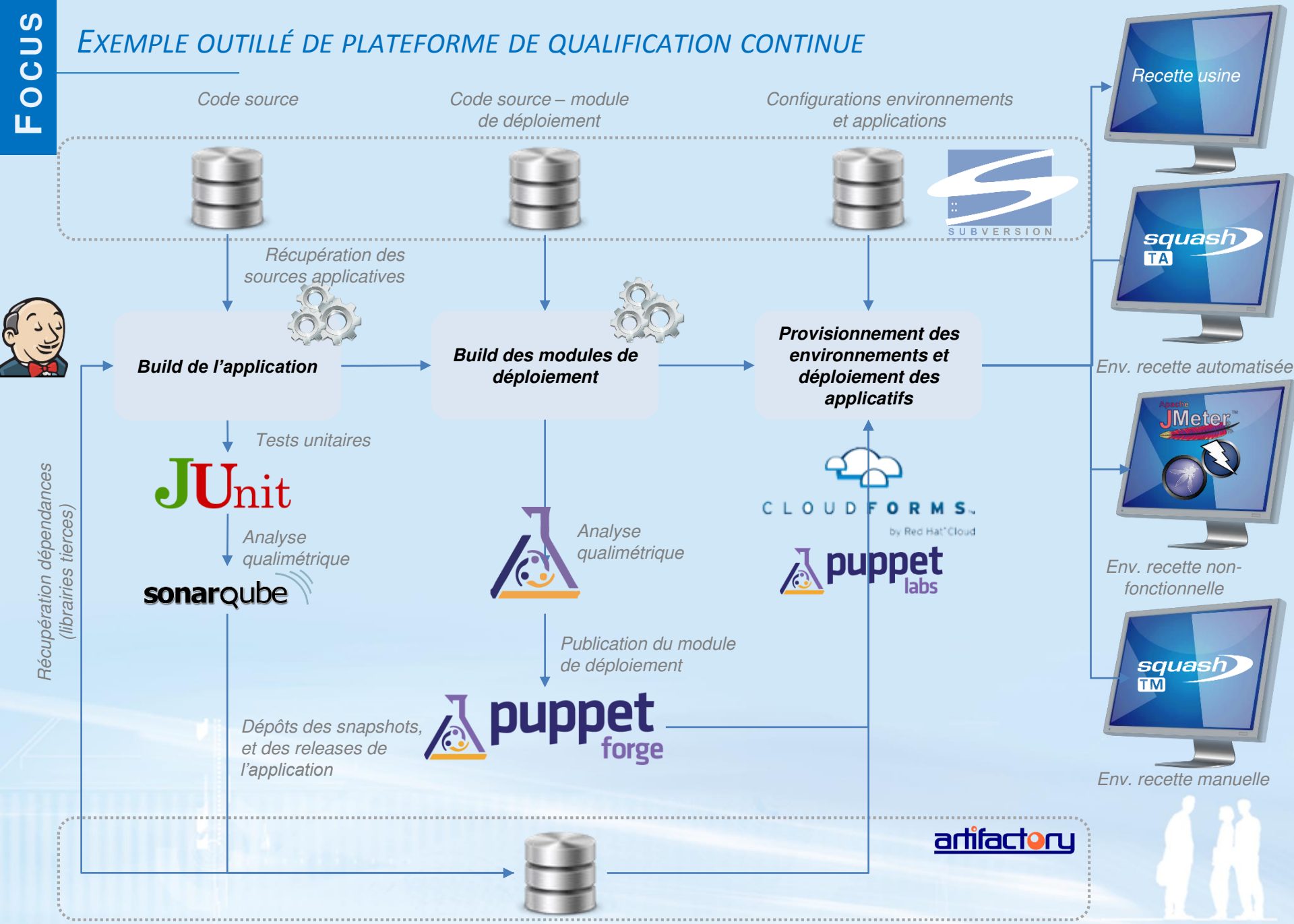
❸ Déploiement de services applicatifs en continu pour automatiser l'intégration et paralléliser les tests applicatifs (fonctionnels, sécurité, performance).

❹ Automatisation des tests fonctionnels et non-fonctionnels applicatifs

❺ Orchestration du passage entre chaque brique de la plateforme afin de mettre en œuvre une chaine complète intégralement automatisée.

FOCUS

# EXEMPLE OUTILLÉ DE PLATEFORME DE QUALIFICATION CONTINUE



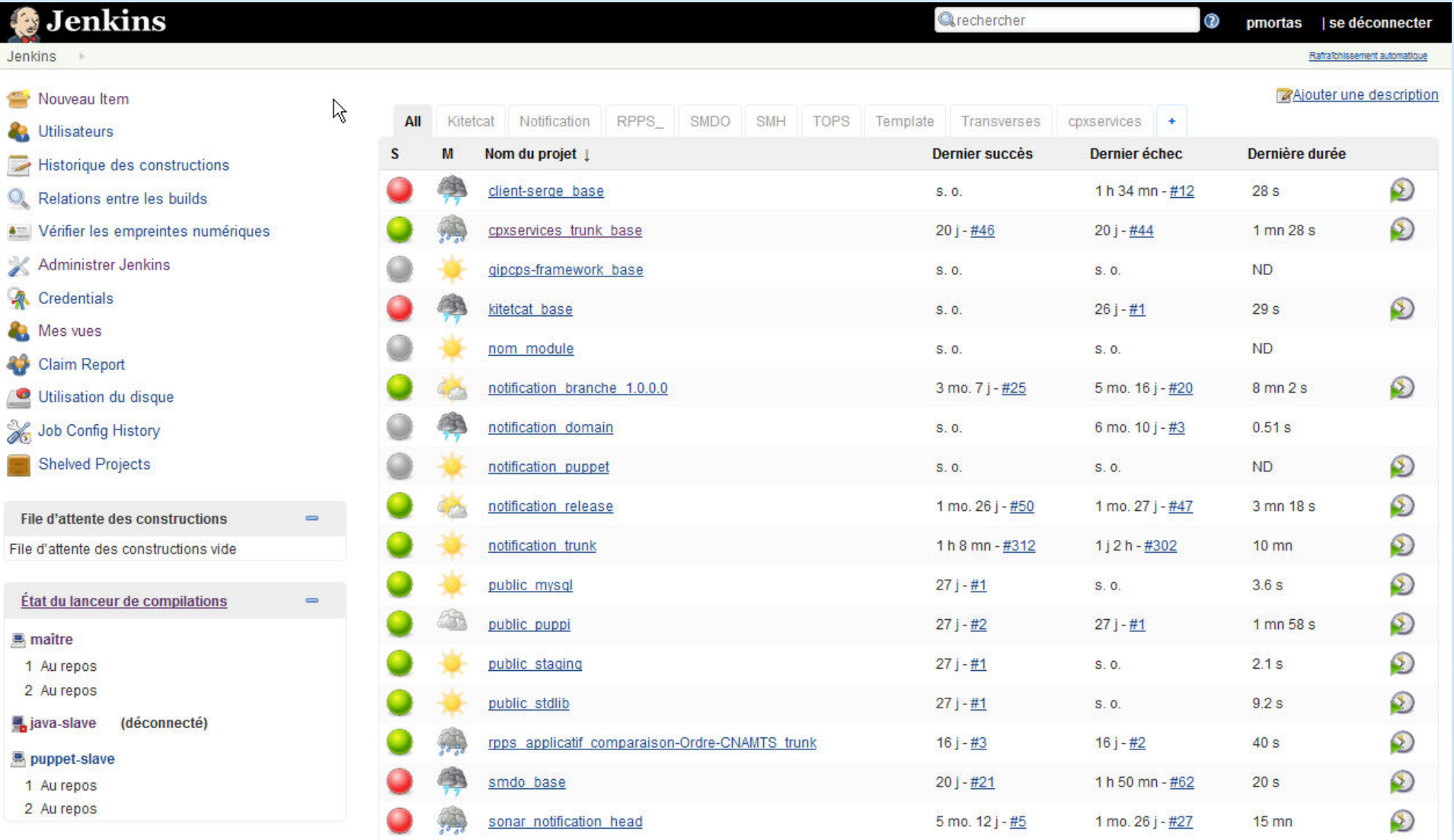
# INTÉGRER EN CONTINU

- Une garantie de :
  - → Gestion centralisée des sources
  - → Processus de build normalisé
  - → Livraisons régulières des sources
  - → mise à disposition des binaires dans le référentiel de l'organisation

- En termes de qualification, elle permet :
  - → La réalisation d'analyses qualité au fil de l'eau
  - → La mise en place de tests unitaires
  - → L'exécution de certains tests d'intégration

# EXEMPLE D'IMPLÉMENTATION

Bamboo
TC
Microsoft Visual Studio Team Foundation Server

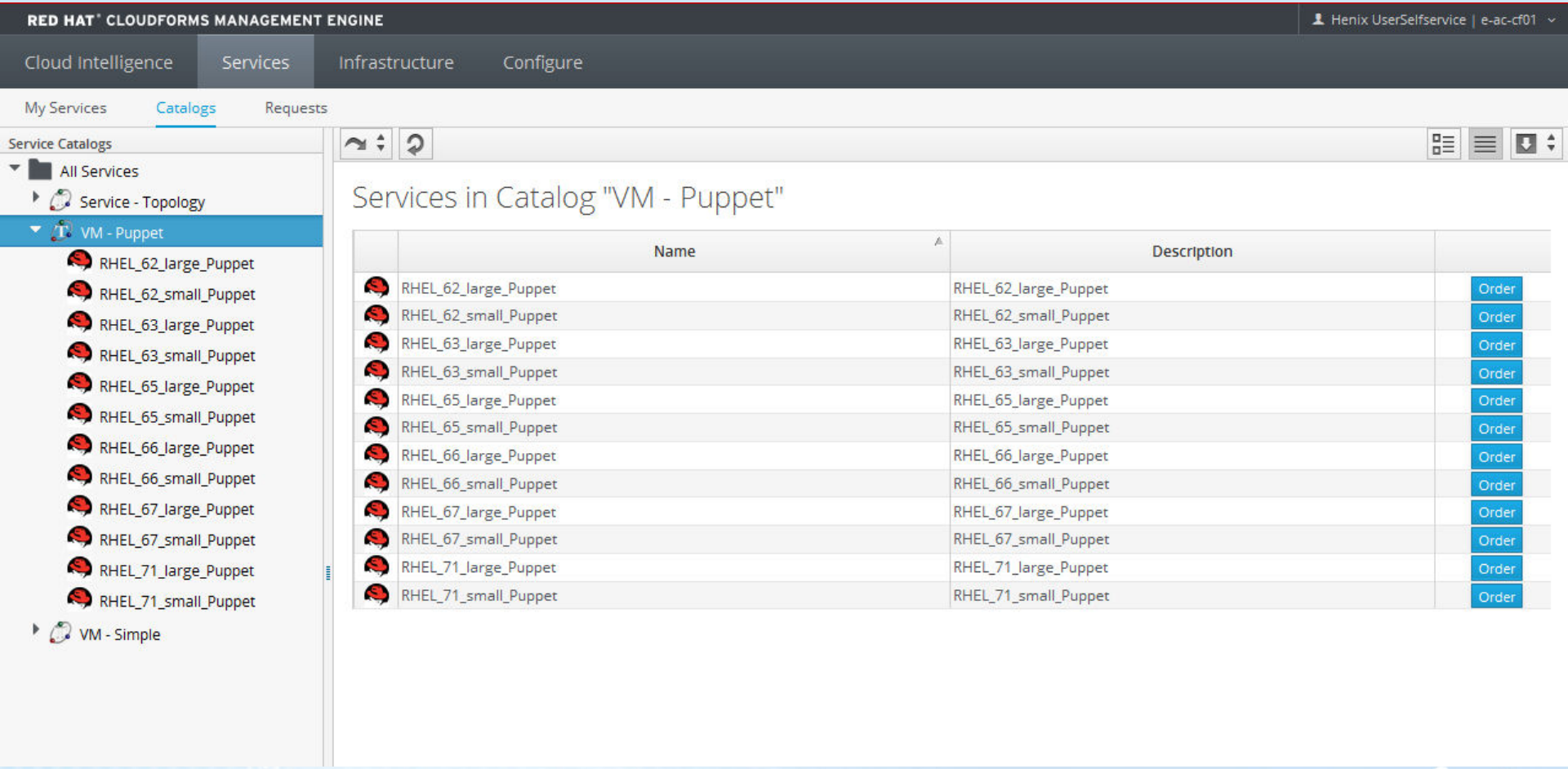


# PROVISIONNER LES ENVIRONNEMENTS

- Virtualiser les environnements :
  - → La création de différentes configurations matérielles
  - → La mise à disposition d'un catalogue de modèles
- Accélérer les cycles de qualification :
  - → La standardisation des environnements, mis à disposition en libre-service
  - → La création dynamique de nouveaux environnements selon les projets
- Rapprocher les environnements de recette et de prod :
  - → La qualification réalisée par la suite est faite en conditions réalistes
  - → Le déploiement applicatif automatisé est fiabilisé et facilement reproductible.

# EXEMPLE D'IMPLÉMENTATION

Cloudify bmcsoftware ManageIQ CLOUDFORMS vmware vRealize™




*Source : Redhat Cloudforms*

# LIVRER LES APPLICATIFS EN CONTINU

- Livrer en continu des packages et des configurations logicielles.
  - → Base de données des configurations déployées (CMDB)
  - → Formalisation de « recettes » afin de décrire l'état souhaité d'un serveur
  - → L'orchestration du déploiement des différents nœuds dans le cadre d'un service applicatif complet
- Dans le cadre spécifique de la PQC :
  - → L'industrialisation des déploiement middleware et applicatifs
  - → L'utilisation de templates OS banalisés
  - → L'automatisation des déploiement de configurations systèmes
  - → Le déploiement automatisé de patches de sécurité
  - → Le déploiement automatisé de services applicatifs complets
- Les applications existantes (quelque soit le legacy) peuvent être déployées en continu sans modification.

# Outillage de livraison continue – Focus sur Puppet

SALTSTACK ANSIBLE Chef puppet labs

puppet enterprise console

node requests (0) admin Help

Nodes Groups Classes Reports Inventory Search Live Management

**Background Tasks**

All systems go

**Nodes**

6 Unresponsive
0 Failed
0 Pending
0 Changed
9 Unchanged
1 Unreported
16 All

Add node Radiator View

**Groups**

| | |
|---|---|
| default | 16 |
| nixes | 10 |
| puppet_console | 1 |
| puppet_master | 1 |
| webservers | 2 |

Add group

**Classes**

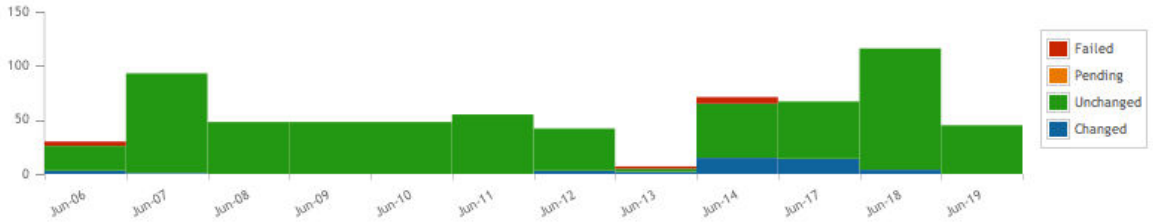
| | |
|---|---|
| motd | 1 |
| ncc::web | 0 |
| ntp | 0 |
| pe_accounts | 0 |

## Nodes

Search

### Daily run status

Number and status of runs during the **last 30 days**:




### Nodes

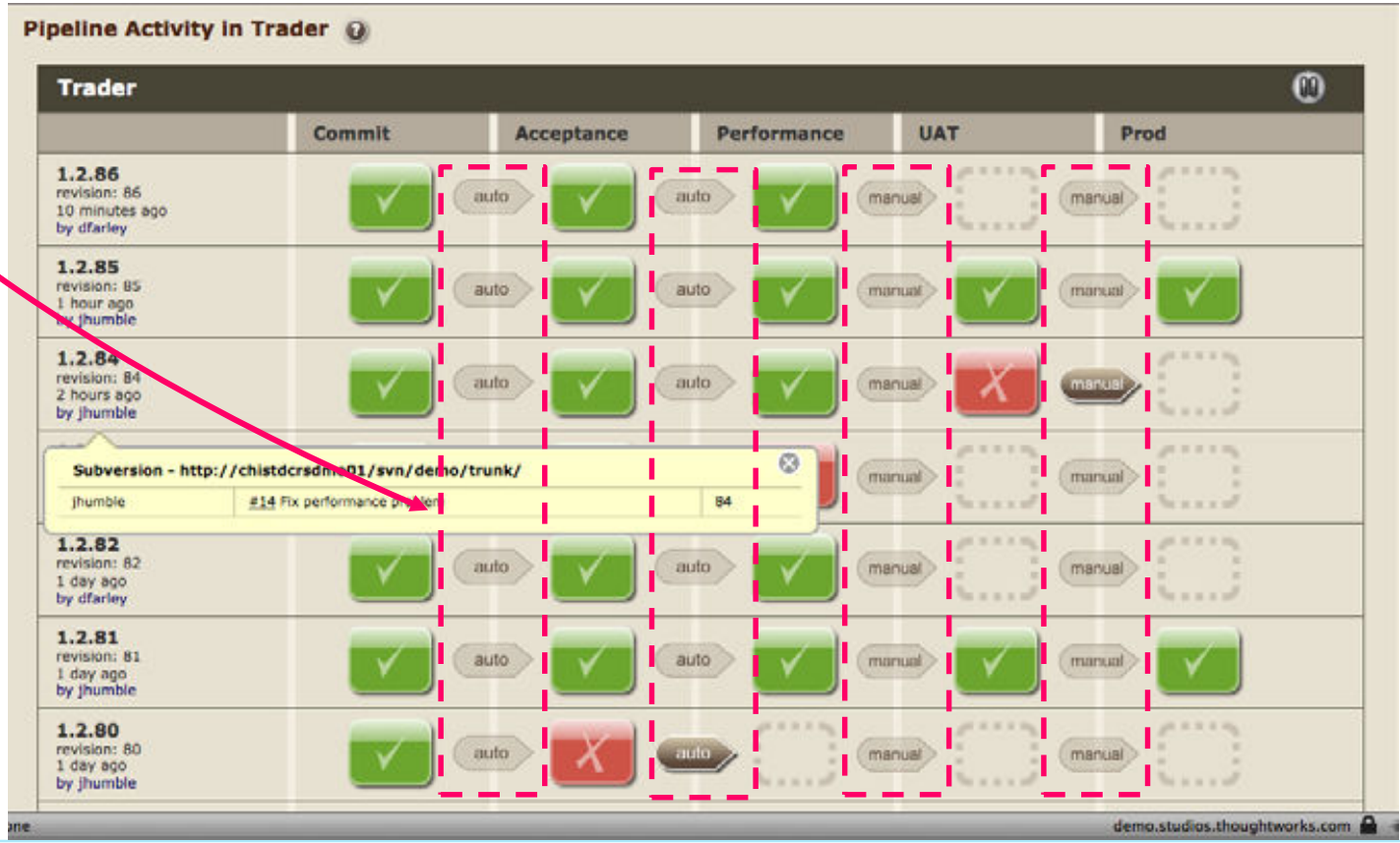
Export nodes as CSV

| Node | Latest report | Resources: Total | Failed | Pending | Changed | Unchanged |
|---|---|---|---|---|---|---|
| Total | | 1559 | 0 | 0 | 5 | 1554 |
| dhcp59.docsteam.backline.puppetlabs.net | Has not reported | | | | | |
| dhcp51 | 2013-06-19 02:24 UTC | 109 | 0 | 0 | 0 | 109 |
| dhcp53.docsteam.backline.puppetlabs.net | 2013-06-19 02:12 UTC | 93 | 0 | 0 | 0 | 93 |
| ftp01 | 2013-06-19 02:12 UTC | 93 | 0 | 0 | 0 | 93 |
| web02 | 2013-06-19 02:12 UTC | 155 | 0 | 0 | 0 | 155 |
| crm01 | 2013-06-19 02:12 UTC | 93 | 0 | 0 | 0 | 93 |
| dhcp50.docsteam.backline.puppetlabs.net | 2013-06-19 02:11 UTC | 122 | 0 | 0 | 0 | 122 |
| crm02 | 2013-06-19 02:10 UTC | 93 | 0 | 0 | 0 | 93 |
| dhcp52.docsteam.backline.puppetlabs.net | 2013-06-19 02:10 UTC | 93 | 0 | 0 | 0 | 93 |
| sql01 | 2013-06-19 02:10 UTC | 93 | 0 | 0 | 0 | 93 |
| web01 | 2013-06-18 23:17 UTC | 155 | 0 | 0 | 0 | 155 |
| dhcp58.docsteam.backline.puppetlabs.net | 2013-06-17 20:29 UTC | 92 | 0 | 0 | 1 | 91 |
| dhcp54.docsteam.backline.puppetlabs.net | 2013-06-17 20:29 UTC | 92 | 0 | 0 | 1 | 91 |
| dhcp56.docsteam.backline.puppetlabs.net | 2013-06-17 20:28 UTC | 92 | 0 | 0 | 1 | 91 |

*Source : Puppet Entreprise*

# ORCHESTRER LE PIPELINE DE PRODUCTION LOGICIELLE

- L'orchestration du pipeline permet l'automatisation du passage entre les différentes étapes de la production logicielle :
  - → Les étapes du pipeline sont modélisées
  - → Un déclenchement automatique ou à la demande est paramétré
  - → Les exécutions sont, au besoin, parallélisées




*Source : GoCD*

# EN PERSPECTIVE, AUTOMATISER LES TESTS NON/FONCTIONNELS

- Dernière brique de la plateforme, l'automatisation continue (ou extreme automation) devient possible dans le cadre d'une PQC.
- Elle permet l'automatisation des tests fonctionnels (IHM, web services, batchs) et non-fonctionnels (APIs),
- Il est donc possible :
  - → De tester plusieurs fois par jour sur des environnements (quasi) identiques à la production
  - → De qualifier l'application au fur et à mesure de son développement
  - → De détecter au plus tôt les défauts les plus critiques

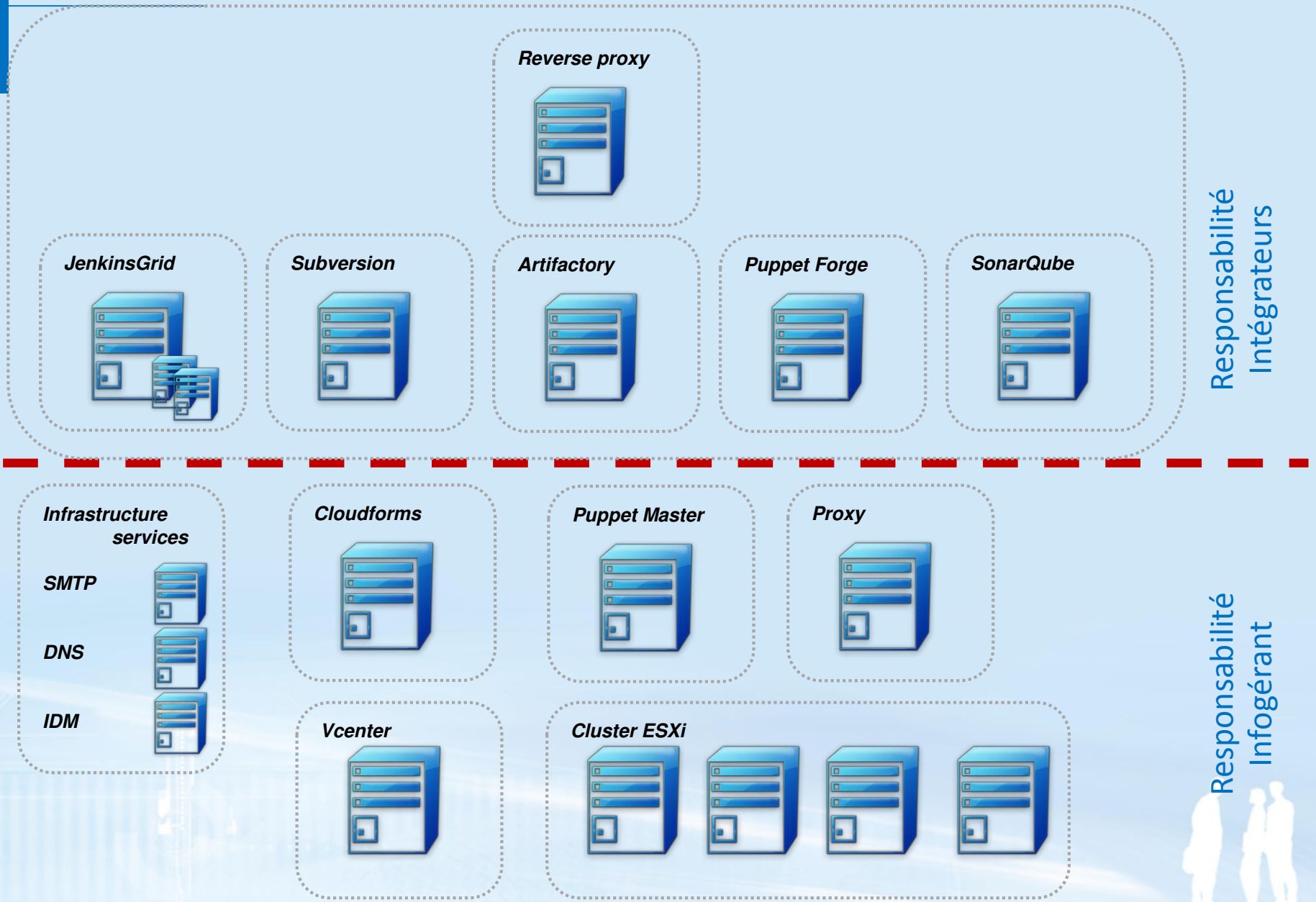
Retour d'expérience

# CONTRÔLER L'EXTERNALISATION DES DÉVELOPPEMENTS À L'ASIP SANTÉ

*2007 à 2011 : Installation de la TRA Henix*

*Depuis 2012 : Industrialisation des développements*

# LE CONTOUR DE LA PLATEFORME DE QUALIFICATION CONTINUE MISE EN PLACE À L'ASIP



# RÔLES ET RESPONSABILITÉS DES ACTEURS

***TMA :***
*Fournit du code directement dans l'usine logicielle*

***Production interne :***
*après accord MOA, déploie directement les binaires qualifiés dans la plateforme de qualification continue*

***Tierce Recette Applicative :***

- *Tests fonctionnels automatisés*
- *Tests non fonctionnels automatisés (performance et sécurité)*
- *Produit les référentiels de bonnes pratiques de développement et de packaging*
- *Contrôle de la portabilité et de la capacité de transfert du code produit à l'extérieur*

***MOE projets externes :***
*Livre son code à intervalles réguliers pour analyse*

***Production externalisé :***
*Fournit les modules de déploiement pour analyse régulière*

# LE NOUVEAU FONCTIONNEMENT



## Avant

- Démultiplication des PIC par sous-traitant
- Environnements des sous-traitants non conformes à la cible (conflits permanents entre la TMA et la production)
- Installations manuelles des applications
- Charge de transfert importante liée aux changements réguliers de prestataires de TMA ou d'infogérance dans le cadre des marchés publics.
- Exécution des tests par la recette après livraison uniquement.
- Démultiplication des environnements de test par sous-traitant

## Après

- Une unique PIC mise à disposition de l'ensemble des sous-traitants.
- Dès les premières phases de test, les environnements sont identiques à la cible.
- Installation automatique de la majorité des applications
- Plateformes de build et de test partagées par tous les prestataires.
- Exécution de tests automatisés avant la livraison
- Construction dynamique, à la demande des environnements de test.

# LES GAINS

(2/2)


## Qualité

- Reproduction des installations sur n environnement **sans erreur**
- Mise en œuvre de contrôles de **qualimétrie**
- **TNRA** joués systématiquement

## Coûts

- Réduction des **charges d'installation et d'intégration**
- Baisse conséquente des **coûts de réversibilité**

## Délais

- **Visibilité** continue de l'avancement des travaux
- Réduction des **délais de qualification**
- Impact positif sur la **qualité de service** et les SLAs

# LES POINTS DE VIGILANCE LORS DE LA MISE EN ŒUVRE DE LA PLATEFORME

## Rôles et responsabilités

- Définir les frontières de responsabilité entre tous les intervenants (difficulté de contractualisation).
- Résister à la créativité des équipes métiers (nouvelles technologies coûteuses à intégrer)

## Outillage

- Définir les bonnes pratiques Puppet : c'est une boite à outils qu'il faut maitriser afin de créer des installeurs maintenables.
- Rationaliser et normaliser les architectures de déploiement et les middlewares utilisés.
- Mettre à jour le processus de production logicielle.
- Suivre et mesurer l'usage de la plateforme par les différents intervenants (La plateforme devient un logiciel critique et sa maintenance doit être prise en compte (upgrade des différents composants...).

## Conduite du changement

- Ne pas sous estimer la charge d'accompagnement et de conduite du changement.

# LES PERSPECTIVES

**Fonctionnelles**

- Gestion des changements dans les bases de données

**Organisationnelles**

- Délimiter clairement les frontières de compétences et de responsabilités de chacun (TMA/TRA/Production) à partir du retour d'expérience
- Mise à niveau des compétences de l'infogérant pour industrialiser le déploiement des middlewares

**Techniques**

- Intégration de composants complexes à déployer automatiquement (Oracle DB, SAP BO)
- Etude de faisabilité à prévoir pour une intégration au processus standard de déploiement
- Mise en place d'un déploiement continu en production

# CONCLUSION

# CONCLUSION

- Avec des outils tels que les plateformes de qualification continue, les métiers de TRA évoluent des tests fonctionnels vers un contrôle du patrimoine applicatif avec un pilotage beaucoup plus proche du développement.
- La mise en place d'outils DevOps entraîne la prise en charge d'une partie des tests par le TMA.
- Un nouveau paradigme et une redistribution des fonctions et des responsabilités s'instaure :
  - Une partie des tests fonctionnels est intégré dans le périmètre de la TMA
  - L'exécution des tests (non) fonctionnels est réalisé avec beaucoup plus de flexibilité et de et de réactivité (réduction du « time to market »)
  - Les TRA interviennent à présent en assistance au contrôle des développements