

Engie Digital - Retour d'Expérience sur la mise en place d'une plateforme d'intégration, de qualification et de déploiement continu autour d'outils et principes Agile et DevOps.



Auteurs : Maxime Herbelot, Henix Jean-Hervé Laveau, Engie Digital





Sommaire

- [Chapitre 1] Contexte Engie vers la transformation « digitale »
 - I. Contexte général du groupe Engie
 - II. Engie Digital et la transformation numérique
- [Chapitre 2] Mise en œuvre de la plateforme de qualification continue Engie Digital
 - I. Enjeux, Prérequis méthodologiques et techniques
 - II. Méthodologie Engie Digital pour le suivi des projets applicatifs
 - III. Les briques techniques de la plateforme
- [Chapitre 3] Retour d'expérience Engie Digital
 - I. Bilan d'étape
 - II. Avantages et Limites
 - III. Perspectives



CONTEXTE ENGIE VERS LA TRANSFORMATION « DIGITALE »

Chapitre 1



Le contexte général du groupe ENGIE

Un grand groupe international :

- → 153090 employés
- → Présent dans 70 pays
- → CA d'environ 66,6 M €

Présent dans de nombreux domaines :

- → Production d'énergie (électricité, gaz naturel, énergies renouvelables ...)
- → Services énergétiques (installation, maintenance, smart city, mobilité verte ...)

Organisé en :

- → 24 Business Units: 11 géographiques, 8 en France, 5 globales
- → Nombreuses Business Entities / Filiales: Altiservice Engie, Engie Axima, Engie Cofely, Engie Ecova, Engie Electrabel, Engie Endel, Engie Fabricom, Engie Global Markets, Engie Home Services, GBS/Engie IT (Information & Technologies), Engie Lab, Engie MtoM, Engie Réseaux, Solairedirect, Tractebel Engie...



Le contexte ENGIE DIGITAL

Quelques dates :

- → Avril 2015 : GDF Suez devient Engie
- → Mai 2016 : Isabelle Kocher est nommée PDG d'Engie
- → Été 2016 : Vision du futur d'Engie Dé carboné, décentralisé et numérique
- → Octobre 2016 : Création d'Engie Digital

Engie Digital en quelques mots :

- → Est un assistant pour la transformation numérique d'Engie
- → Aide les BU à imaginer, définir, créer et produire des applications numériques dont elles sont propriétaires
- → Se base sur un socle technique commun faisant appel à des partenaires privilégiés : AWS, C3, Kony & Thales
- → L'objectif est qu'à terme l'ensemble des développements Engie soient agiles et déployés dans le cloud
- → Engie Digital construit un backbone AWS pour héberger les développements numériques de BU



Chapitre 2

MISE EN ŒUVRE DE LA PLATEFORME DE QUALIFICATION CONTINUE ENGIE DIGITAL

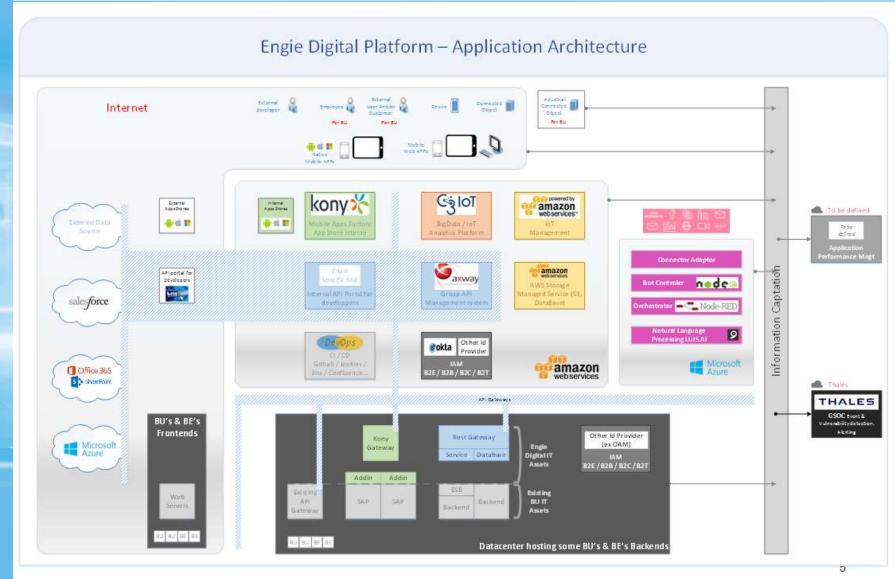


Chapitre 2.1

ENJEUX, PRÉREQUIS MÉTHODOLOGIQUES ET TECHNIQUES



INTÉGRATION DANS L'ENVIRONNEMENT ENGIE DIGITAL





PARADIGMES

Tous les composants d'une application doivent être versionnés :

- Versionnement du code & des binaires applicatifs
- Versionnement du code de déploiement
- Versionnement des scripts de base de données
- Versionnement des images des conteneurs de build et de déploiement
- Versionnement des environnements de déploiement

Convention over configuration :

Objectifs:

- Réduire au minimum les configurations à réaliser pour les développeurs au niveau des outils
- Gérer la complexité organisationnelle d'Engie
- Pouvoir réutiliser les artifacts générés par d'autres projets

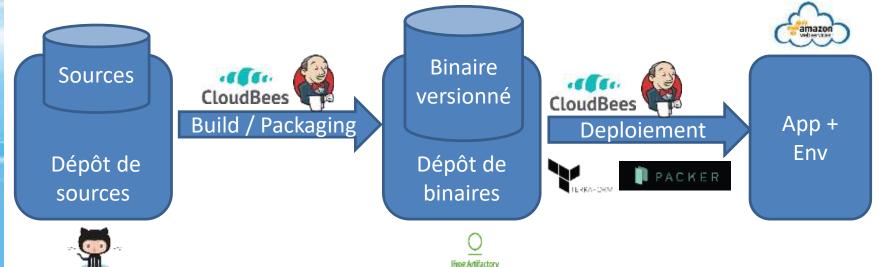
Moyens:

- Conventions de nommage des artifacts des projets
- Mise à disposition de templates de création projet
- Mise à disposition de templates de pipelines et d'une librairie partagée de code



PARADIGMES

- Immutabilité des environnements de build & de déploiement
 - Immutabilité des instances de build & de déploiement des applications
 - Immutabilité des instances déployées sur les environnements
- Isomorphisme des environnements
 - Isomorphisme des environnements de DEV, UAT et PROD sur lesquels sont déployés les applications
- Pattern de déploiement générique





PARADIGMES

- Infrastructure As Code
 - Les déploiements sur les différents environnements doivent être complètement automatisés et ne doivent différer que sur les valeurs de leurs paramètres
 - L'infrastructure doit être entièrement codée. Pas d'action via la console
 - L'infrastructure As Code doit permettre de gérer :
 - L'initialisation
 - La reprise sur incident
 - La mise à jour de la plateforme & des applications du tooling



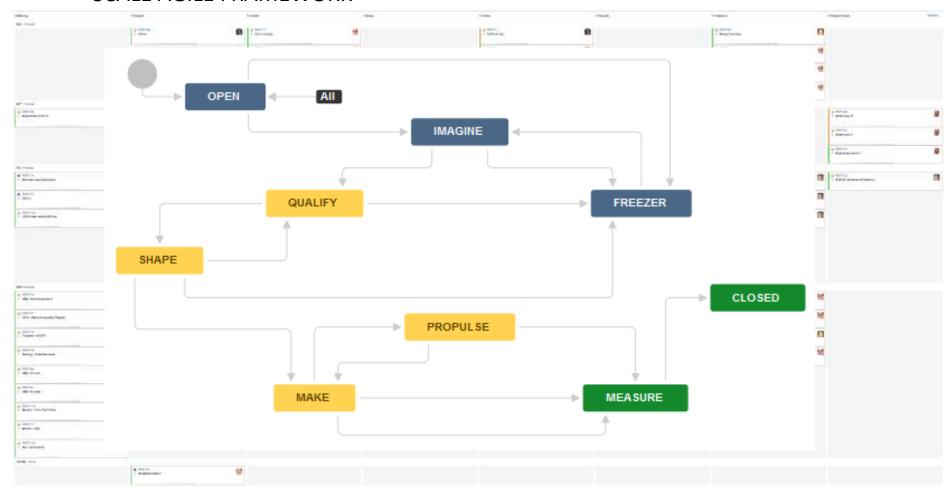
Chapitre 2.2

MÉTHODOLOGIE ENGIE DIGITAL POUR LE SUIVI DES PROJETS APPLICATIFS



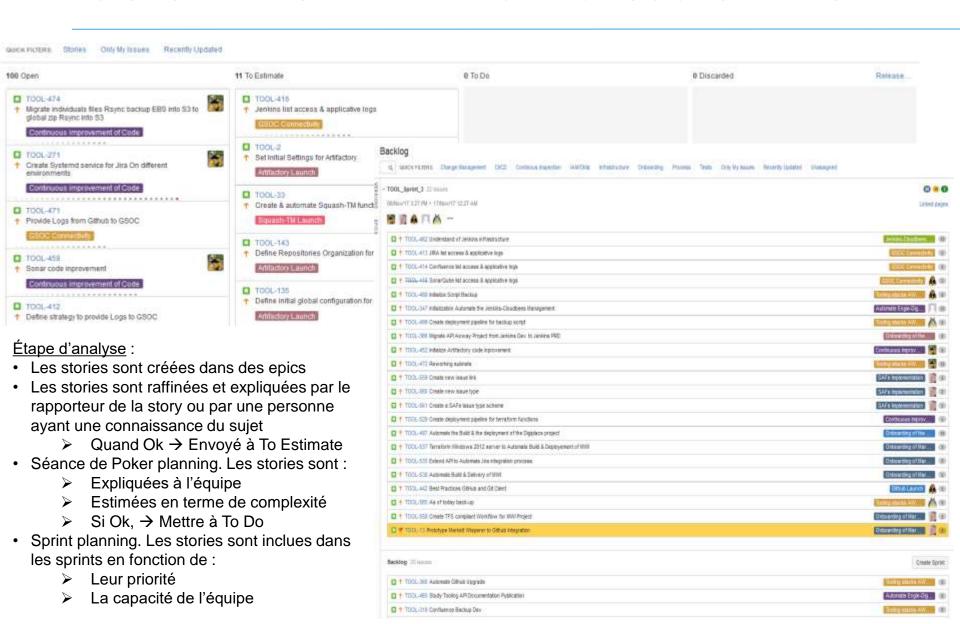
GESTION DU PORTFOLIO DE PROJETS: METHODE SAFE

SCALE AGILE FRAMEWORK

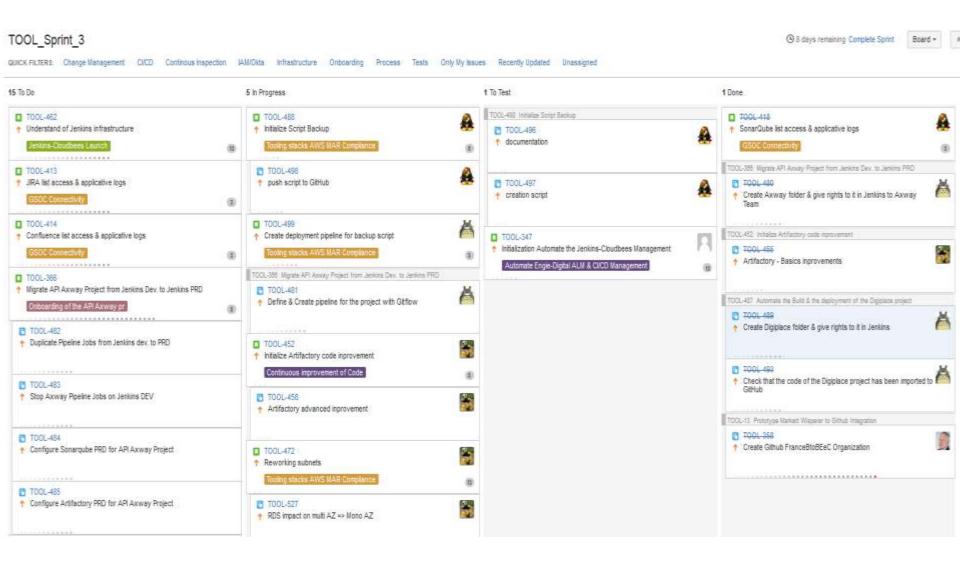


BACKLOG, IMAGINE, QUALIFY, SHAPE, MAKE, PROPULSE, MEASURE, FREEZE/CLOSE

GESTION DE PROJET : RAFFINAGE BACKLOG & ESTIMATION



GESTION DE PROJET : SPRINTS



LES BRIQUES TECHNIQUES DE LA PLATEFORME

Chapitre 3



BESOINS IDENTIFIÉS

Code:

- Infrastructure:
 - Network : VPC, Subnets
 - IAM
 - Security : Security groups
- Application platforms
- IoT
- API management
- Data Analytics Functions
- UX

Continuous Build:

- Build Apps
- Package modules

Continuous Configure:

- Infrastructure
- Apps

Plan:

- Features
- Sprints
- Modules Releases
- Apps Releases
- ArchitectureSecurity

Define:

Infrastructure

UX Design

- Deployment
- Features : Epic, Stories, Tasks

Communicate:

- Into Team
- Into Project
- Share with
 - Company
 - **-** .

Plan release configure monitor

Release, Continuous Deploy, Automate Continuous Configuration:

- Infrastructure
- Modules
- Releases
- Applications
- Test, UAT, Prod Deployments

Coordinate:

- Release & Deployments
- Incidents
- Customers needs

Structure & Describe Tests:

- Scenarios / Plans
- Campaigns

Continuous Test:

- Unit Tests
- Integration / Regression tests

Continuous Assessment for Code Quality:

- Complexity & Technical debt
- Best coding practices

Code Security

Code Performance

Versioned Binaries Centralization:

- Produced Apps & Modules packages
- Input OSS & modules

Measure & Monitor:

- Infrastructure performance
- Application performance

Alert on:

- Security issues
- Performance issues

OUTILS SÉLECTIONNÉS POUR LE MONTAGE DE LA PLATEFORME

plan

Code:

- Infrastructure:
 - Network: VPC, Subnets
 - IAM
 - Security: Security groups
- Application platforms
- IoT
- API management
- **Data Analytics Functions**
- UX





axway

Continuous Build:

- **Build Apps**
- Package modules



- Infrastructure
- Apps



ell ffe.

CloudBees

Plan:

- **Features**
- Sprints
- Modules Releases

AGILE

verify

Apps Releases



Define:

- **UX** Design
 - Architecture
 - Security
 - Infrastructure Deployment

Features: Epic, Stories, Tasks

release

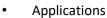
Communicate:

- Into Team
- Into Project
- Share with





Release,



Modules

Configuration:

Test, UAT, Prod **Deployments**

Continuous Deploy.

Automate Continuous

Infrastructure



el Tar

CloudBees







Coordinate:

- Release & Deployments
- Incidents
- Customers needs







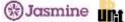
Structure & Describe Tests:

- Scenarios / Plans
- Campaigns



Continuous Test:

- **Unit Tests**
- Integration / Regression tests











Continuous Assessment for Code Quality:

- Complexity & Technical debt
- Best coding practices

Code Security



Code Performance

Versioned Binaries Centralization:

Produced Apps & Modules packages

monitor

Input OSS & modules



Measure & Monitor:

- Infrastructure performance
- Application performance

Alert on:



Performance issues



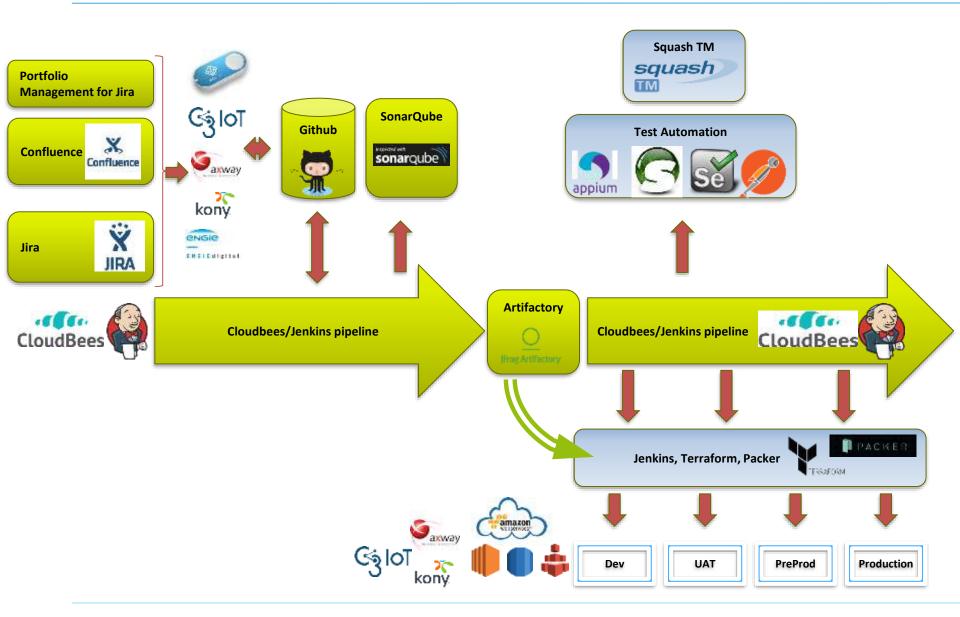


Grafana





PIPELINE D'INTÉGRATION ET DE LIVRAISON/DÉPLOIEMENT CONTINU



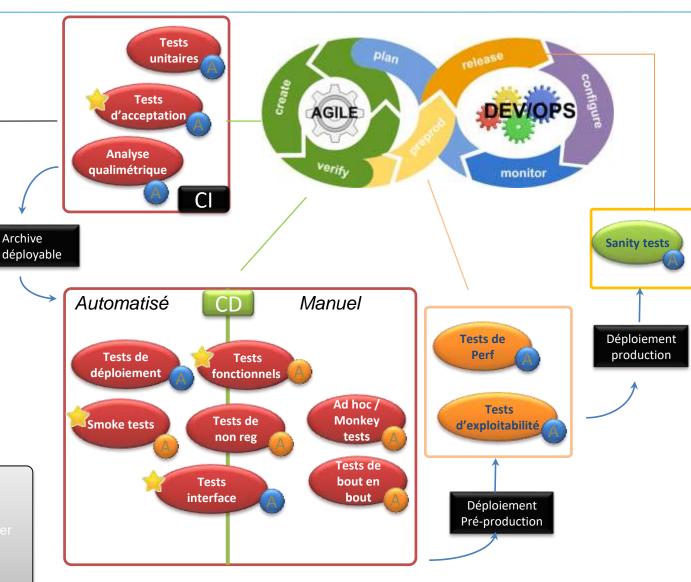
TEST EN CONTINU



d'acceptation

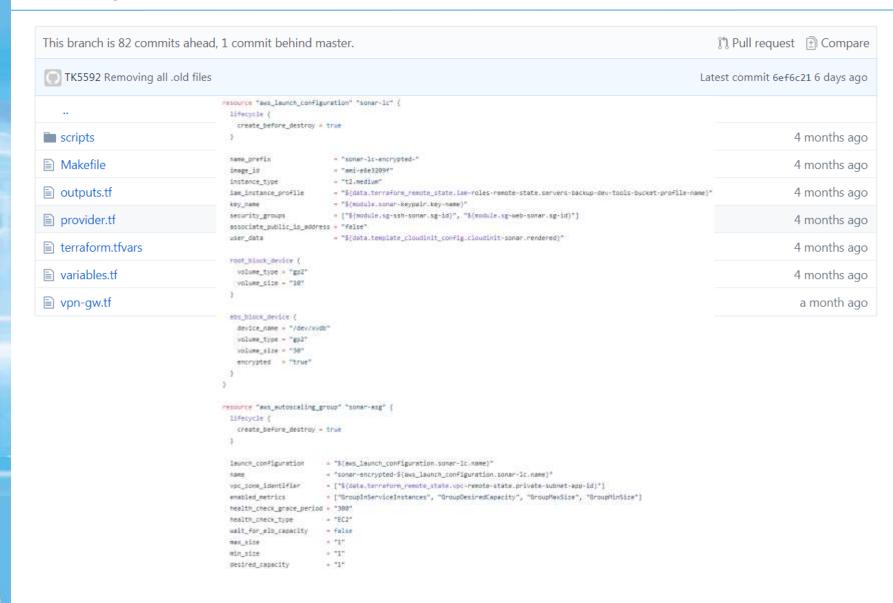
 Automatisation complète des smoke tests et des tests de non régression







LES TECHNOS SOUS-JACENTES : GESTION DE L'INFRASTRUCTURE AVEC TERRAFORM





LES TECHNOS SOUS-JACENTES : GESTION DES BUILDS AVEC DOCKER & MESOS

```
FROM ubuntu:16.04
‡ File Name
                    : Jean Thierry(jean.thierry@external.engie.com)
# Creation Date
                 : 2017/10/24
                    : 0.2
# Top
# Company
                    : engie
                    : gbs
‡ Be
                    : engiedigital
# Project
                    : mastermind
# Service Provider : mmd
# Description
                    : Dockerfile for the project mastermind
# Updates history :
# Version | Date | Author
                                    | Description
# 0.2 | 2017/10/26 | Jean Thierry | Adding Python3
# Labels #
!----
# Name artifact
LABEL top="com" \
     company="engie" \
      bu="gbs" \
     be="engiedigital" \
     project="mastermind" \
     service_provider="mmd"
# Components
LABEL openjdk.version="8" \
     terraform.version="0.10.7"
# Install OpenJDK #
‡ -----:
RUN apt-get update
RUN echo "===> Installing openjdk-8 for Jenkins connection" && \
   apt-get install -y openjdk-8-jdk
# Install Gradle #
± -------
RUN echo "===> Installing Gradle" && \
 apt-get update && \
  apt-get -y upgrade && \
  apt-get install -y gradle
# Install Terraform #
RUN echo "===> Installing Terraform" && \
 apt-get update 66 \
  apt-get -y upgrade && \
  apt-get install -y wget 66 \
  wget https://releases.hashicorp.com/terraform/0.10.7/terraform_0.10.7_linux_amd64.zip && \
  unsip terraform 0.10.7 linux amd64.sip -d /opt/
```



LES TECHNOS SOUS-JACENTES : CONSTRUCTION DES INSTANCES AVEC PACKER

```
"variables": {
  "aws_access_key": "",
  "aws secret key": ""
"builders": [{
  "type": "amazon-ebs",
  "access_key": "{{user `aws_access_key`}}",
  "secret_key": "{{user `aws_secret_key`}}",
  "region": "us-east-1",
  "source ami_filter": {
    "filters": {
    "virtualization-type": "hvm",
    "name": "ubuntu/images/*ubuntu-xenial-16.04-amd64-server-*",
    "root-device-type": "ebs"
    "owners": ["099720109477"],
    "most_recent": true
  },
  "instance_type": "t2.micro",
  "ssh_username": "ubuntu",
  "ami name": "packer-example {{timestamp}}"
}]
```



LES TECHNOS SOUS-JACENTES: DESCRIPTION DU WORKFLOW - JENKINSFILE

```
node('jdk-8-mvn-3.5-sonar-3') {
     .....
   def projectUrl = 'https://github.dev.tools.digital.engie.com/Engie-Digital/git workflow test.git'
   def projectBranch
                  = env.BRANCH NAME
                   = env.CHANGE TARGET
   def projectTarget
   echo "The branch is the following : ${projectBranch}"
   .....
                      Branches
       */
   // Types of branches :
   // - master
   // - Pull Request (Between release & master or hotfix & master)
   // - hotfix
   // - release
  // - develop GITFLOW // - Pull Request (Between feature & develop)
   // - feature
   // Master Branch (CI + Deploy ROO) HUB HOOK if (projectBranch == 'master HUB HOOK
     prepareDirectory()
     checkout()
     unitTests()
     GITHUB ORGANIZATION FOLDER PLUGIN
     publishArtifacts()
     deployOnProd()
     sendMail()
   // Pull Request Branch to Master Branch (CI + Deploy UAT)
   else if ((projectBranch.startsWith('PR-')) && (projectTarget == 'master')) {
     prepareDirectory()
     checkout()
     unitTests()
     staticAnalysis()
     generateArtifacts()
     publishArtifacts()
     deployOnUat()
     sendMail()
```



RETOUR D'EXPÉRIENCE ENGIE DIGITAL

Chapitre 3



BILAN D'ÉTAPE

Chapitre 3.1



Plateforme TOOLING Engie Digital (Décembre 2017)

268 repos de code regroupés dans 35 organisations GitHub

(Java, C++, PHP, JS, Python, HCL, Shell, TypeScript ...)











Jira & Github Licences & Projects





7 outils dédiés au développement Logiciel

qui tournent 7/7 – 24/24

555 utilisateurs

intégrés à la plateforme



AVANTAGES ET LIMITES

Chapitre 3.2





Points Négatifs

- → Plateforme et process non encore complètement scalables
- → Diversité des technologies projet
- → Évangélisation complexe à mettre en œuvre
- → Agilité & DevOps vs sécurité

→ Point sécurité sur le déploiement continu



Points positifs

- → Une unique PQC mise à disposition de l'ensemble des BU/BE d'ENGIE
- → 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



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).

Outillage

- Définir les bonnes pratiques de développement et de déploiement : c'est une boite à outils qu'il faut maitriser afin de créer des applications maintenables.
- Rationaliser et normaliser les architectures de déploiement et les middlewares utilisés.
- 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.



PERSPECTIVES

Chapitre 3.3



ROADMAP 2018

