



12 ans d'usine logicielle



Du petit garage à l'industrialisation



Qui suis-je ?

Architecte logiciel / DevOps

Développeur Java

Geek

Co-fondateur de [TADx](#)

Agilité & Scrum

@HM depuis 12 ans



Stéphane Philippart





Disclaimer

Du vrai “c’était mieux avant” (ou pas)

Du vrai troll

Du vrai “ah ouais on faisait ça aussi avant”

HARMONIE MUTUELLE EN BREF



4,8 M

De personnes protégées

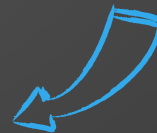
5000

collaborateurs



Dont 500

Dans l'IT



PHASE 1 : 2008 - 2010

LA GÉNÈSE

Vous avez dit legacy ?

LE 1er ESSAI

L'établi



LA GÉNÈSE

Tout a un début !

Il n'y avait vraiment rien ?

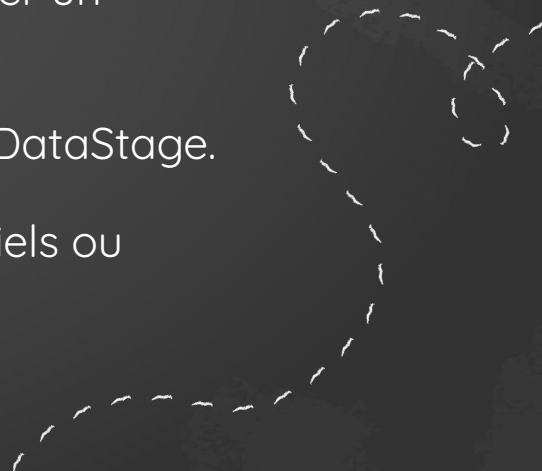
Avant 2008 : je ne sais pas vraiment je n'étais pas là :)

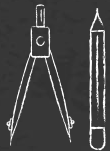
2006 : naissance du GIE SIHM (principalement rapprochement d'Harmonie Mutualité et de Prévadiès).

Le principal enjeu : faire cohabiter deux SI ensembles et créer un nouvel SI.

Donc beaucoup (trop ?) de batchs de nuit avec notre ETL : DataStage.

Très peu de développements temps réel, hormis des progiciels ou quelques applications intranet.





2008

LES TECHNOS ET OUTILS UTILISÉS

C'était hype à l'époque !
(enfin presque :))

Maven

JEE 5

JSF (Tomahawk)

EJB 3.0

Eclipse / BEA Workshop

Weblogic

DTS

BO



JEEs, JSF, EJB ... dans Weblogic

- EAR ... ouch
- La promesse (ou les mensonges) de JSF
- Tomahawk
- Stuck Threads

The screenshot shows the Oracle WebLogic Administration Console interface. The main content area is titled "Récapitulatif des déploiements" (Deployment Summary). It includes a "Contrôle" (Control) tab and a "Surveillance" (Monitoring) tab. Below the tabs, there is a paragraph explaining that the page displays a list of Java EE applications and modules installed in the domain, and provides instructions on how to start, stop, or update them. A "Personnaliser cette table" (Customize this table) link is also present. The table below lists several deployed applications, all with an "Actif" (Active) status and a "OK" health indicator.

<input type="checkbox"/>	Nom	Etat	Etat
<input type="checkbox"/>	api-ref-ressources-objets-bouchons	Actif	OK
<input type="checkbox"/>	api-ref-ressources-objets-bouchons	Actif	OK
<input type="checkbox"/>	api-ref-ressources-objets-bouchons	Actif	OK
<input type="checkbox"/>	api-ref-ressources-objets-bouchons	Actif	OK
<input type="checkbox"/>	attitudes	Actif	OK

BO & DATASTAGE

- Legacy, vous avez dits legacy ?
- « Administration »



NORMES & BONNES PRATIQUES

- Principalement sur le développement (très peu sur la conception & l'architecture)
- Articles dans un Wiki (mediawiki)



LIBRAIRIES

- Faciliter l'application des normes
- Début de factorisation de code dans des librairies maison
- Maintenance des ~~shared lib~~ Weblogic



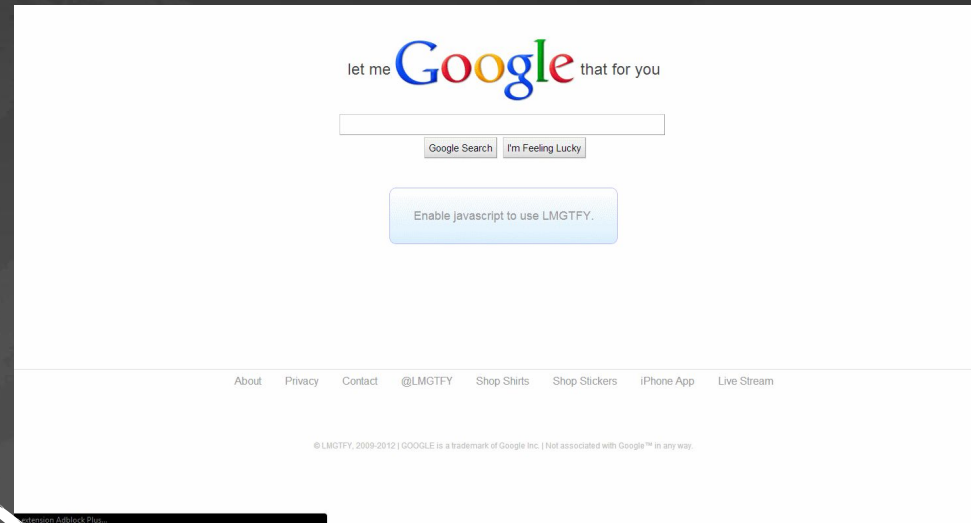
ELIPSE vs INTELLIJ

- Bataille de clochers
- Maven au secours !!
- Ne plus dépendre de l'IDE



SUPPORT

- Problèmes de configuration de postes : maven, eclipse (BEA Workshop), Weblogic, ...
- Problèmes de développements : bugs avec les frameworks utilisés, voire le code produit par les développeurs
- Conclusion logique de vouloir tout normer au niveau de développement





ADMINISTRATION

- Dev to prod : la production est un vrai métier !
- Dans le doute reboot : on ne s'invente pas administrateur de plateforme lorsque l'on est développeur

L'ancêtre du DevOps :p





➔ L'ÉTABLI

1er essai

MAVEN 2

- Gestionnaire de dépendance et de packaging
- Éviter de devoir gérer le classpath de compilation dans chaque IDE
- Intégration précaire dans Eclipse
- Première configuration centralisée (profils, proxy d'entreprise, ...)



GESTION DES SOURCES

- ~~Gestion des droits~~
- Tortoise SVN
- Sauvegarde avant tout
- Début de la culture du commit
- Centralisation et distribution



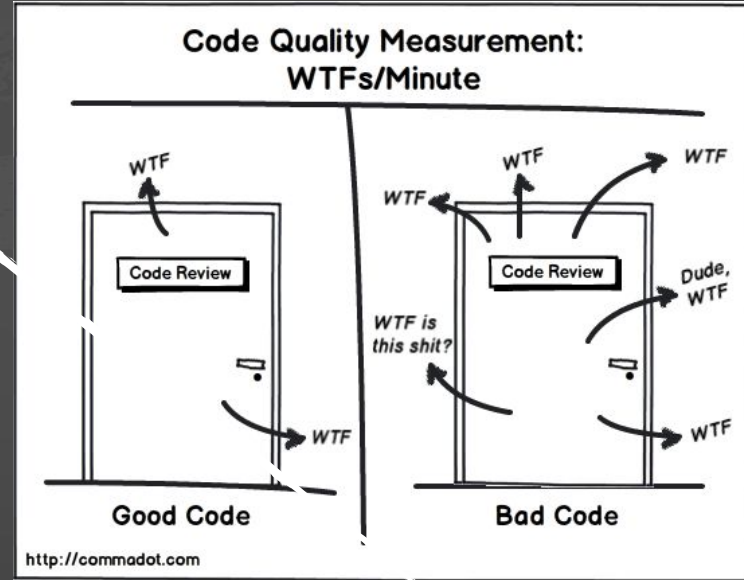
SHARED LIB & WLST

- N'avoir qu'un jeu de lib externes
- "Maîtriser" les libs externes
- Déploiement et administration "automatiques"



REVUE DE CODE (et procédures)

- Pas d'outils
- Beaucoup de doc
- Beaucoup d'énergie
- Beaucoup de frustration



Bilan de la phase 1 (2008 - 2010)



- FACTORISER LE CODE (LIB)
- GESTIONNAIRE DE SOURLES
- MAVEN



- NORMER À OUTRANCE
- IMPOSER UN IDE
- SUPPORT AUX DEVS
- ADMINISTRER LA PROD
- REVUE DE CODE MANUELLE
- CONTRÔLE DES LIBS EXTERNES

PHASE 2 : 2010 - 2014



UPGRADE

Atelier de quartier

JEE forever ...

- JEE6
- PrimeFaces & jQuery
- Essayer de suivre plus rapidement les évolutions
- Rendre les devs plus “heureux”



LA PROD C'EST UN VRAI MÉTIER

- Enfin une vraie exploitation
- On se recentre sur tout ce qui touche au développement



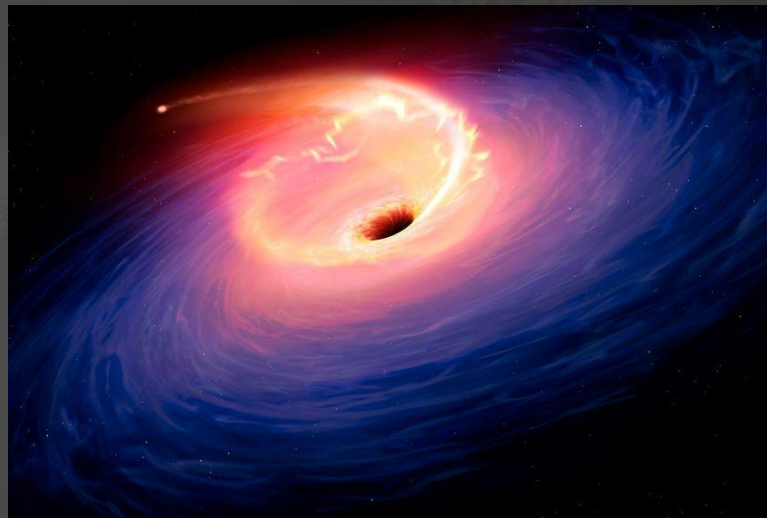
JENKINS ... LI TU SERAS MIENNE

- Ah enfin un peu d'automatisation
- Rendre de l'autonomie aux devs
- Outil central
- Le poste de développement pour ... développer
- ~~Beaucoup de jobs, beaucoup de maintenance~~
- Administration



NEXUS

- Repository Maven
- Ne pas tout exposer
- Simplifier la configuration
- Une mémoire de nos applications
- Proxy vers l'extérieur



SONAR ... JUSTE MERLI



- Finies les revues de code (enfin presque)
- ~~Profil maison~~
- ~~Toi aussi joue à la qualité~~
- Qualité avant le déploiement staging
- Principe de la fuite d'eau
- Rassure la production



MAVEN AVANCÉ ... FBI ?

- Archetypes
- Pom entreprise
- ~~Gestion avancée des dépendances (base line)~~
- ~~Centralisation de la configuration des pom~~
- Simplification des pom projets
- ~~Très (trop) complexe à maintenir~~



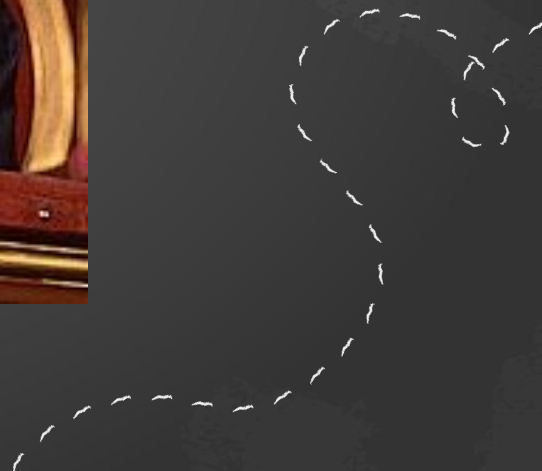
Bilan de la phase 2 (2010 - 2014)



- PROD
- JENKINS
- NEXUS
- SONAR



- JEE
- JENKINS
- SONAR
- POM ENTREPRISE



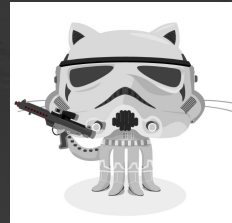
PHASE 3 : 2014 - 2019



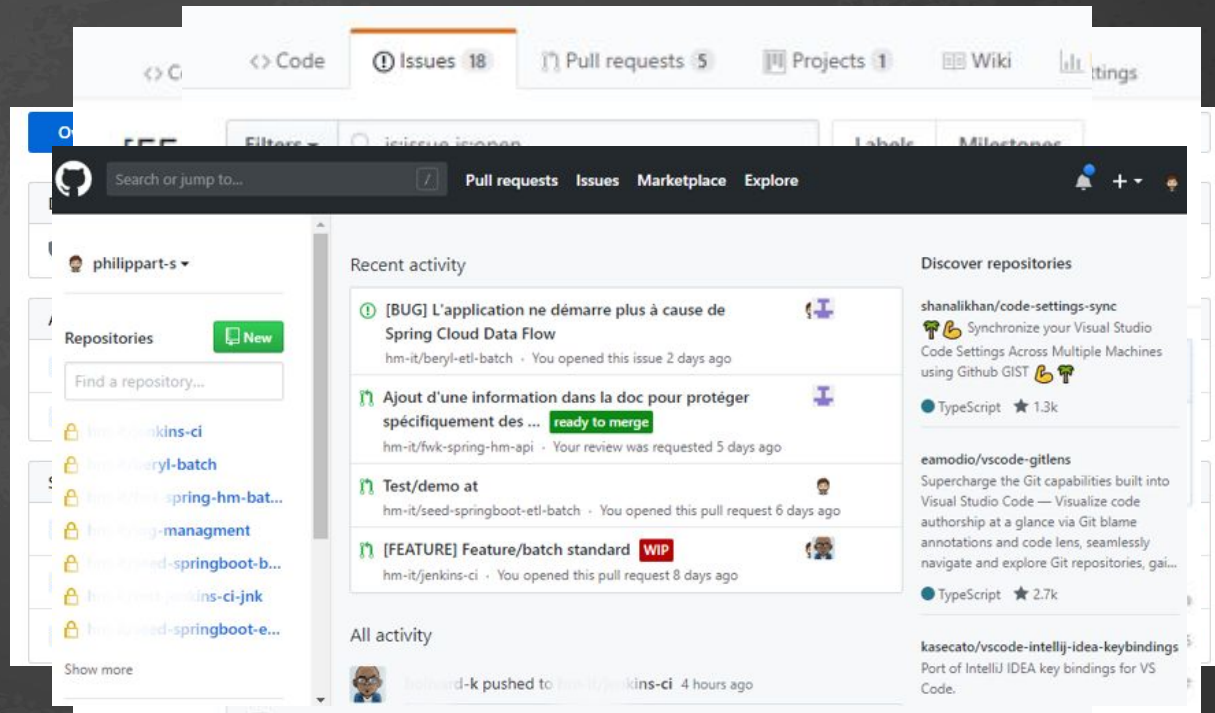
SCALE

usine Logicielle

GIT & GITHUB

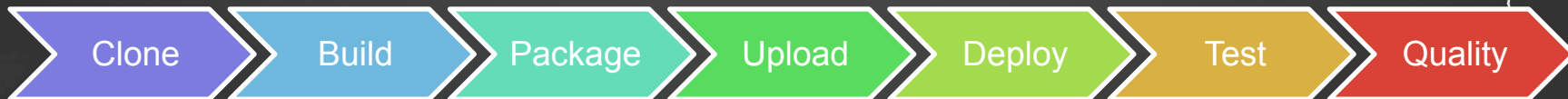
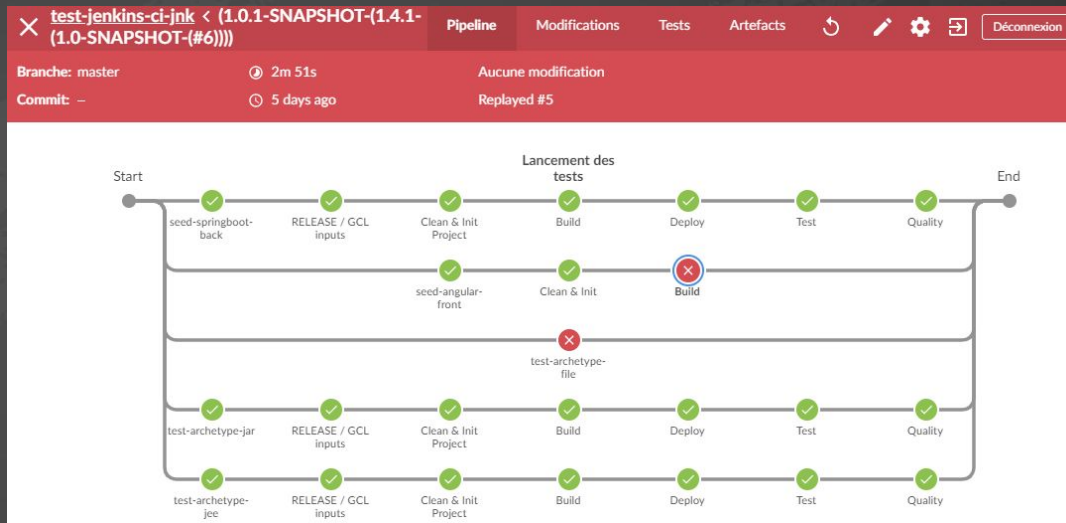


- Remplacer SVN
- Équipes en France
- ADFS / SAML
- SaaS
- Pas de migration
- Administration light
- Feature branches
- Collaboratif
 - PullRequest
 - Issues
 - Portail (MD)



JENKINS 2 - PIPELINES

- Ne plus utiliser l'interface pour paramétrer les jobs
- Factoriser / réutiliser les étapes de build
- Avoir une finesse des étapes de jobs
- Parallélisation



JENKINS 2 - SHARED LIBS

- Projet Groovy
- Pipelines as code
- Factorisation de code
- Hébergée sous GitHub
- Compatible feature branches
- Définition d'un DSL pour les Jenkinsfile
- Monolithe
- Un peu figé
- Mode de développement un peu lourd
- Tests unitaires complexes

Global Pipeline Libraries

Sharable libraries available to any Pipeline jobs running on this system. These libraries will be trusted, meaning they run without "sandbox" restrictions and may use @Grab.

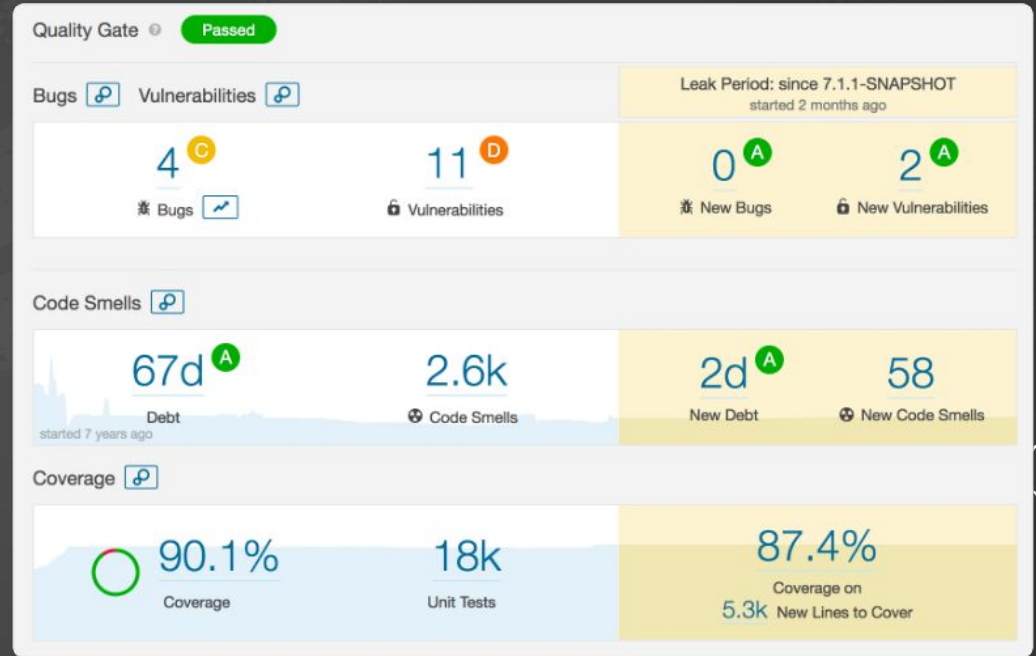
Library	
Name	<input type="text" value="my-shared-library"/>
Default version	<input type="text" value="master"/>
Load implicitly	<input type="checkbox"/>
Allow default version to be overridden	<input checked="" type="checkbox"/>
Retrieval method	
<input checked="" type="radio"/>	Modern SCM

JENKINSFILE

```
Jenkinsfile x
1  #!/usr/bin/groovy
2
3  //@Library('jenkins-hm@bug/credentials-sonar') _
4
5
6  mavenNode(['node_qualifier':'SpringBoot']) {
7      email='ab0f81e6.harmoniesuuelle.onmicrosoft.com@amazteams.ms'
8      mavenBoot{
9          gcl_folder="/DEV_EAND/cycledev/tags"
10         deployment=false
11         deployment_target='xxxxx'
12         service_name='intl_beryl-batch_21001.service'
13     }
14 }
```

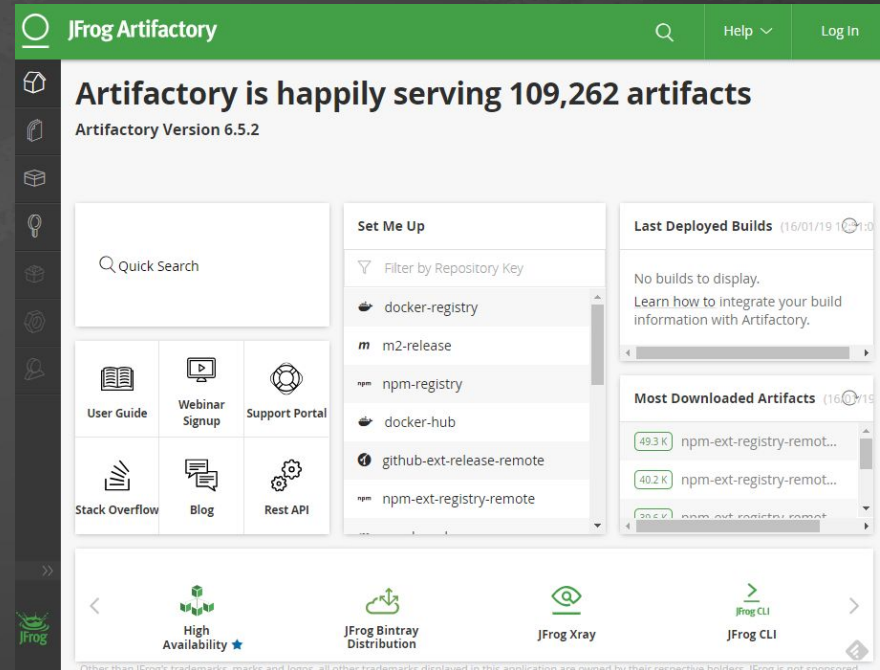
SONARQUBE

- Mise à jour (LTS)
- Utilisation des profils et quality gate Sonar
- Pas d'analyse de branches (payant)



AU REVOIR NEXUS ... ET MERCI

- Nexus 2 vieillissant
- Version Pro X (24/7 + XRay)
- Support efficace
- Ux



Jfrog Artifactory Help Log In

Artifactory is happily serving 109,262 artifacts

Artifactory Version 6.5.2

Quick Search

Set Me Up

Filter by Repository Key

- docker-registry
- m2-release
- npm-registry
- docker-hub
- github-ext-release-remote
- npm-ext-registry-remote

Last Deployed Builds (16/01/19 10:00)

No builds to display.
Learn how to integrate your build information with Artifactory.

Most Downloaded Artifacts (16/01/19 10:00)

- 49.3 K npm-ext-registry-remot...
- 40.2 K npm-ext-registry-remot...
- 38.2 K npm-ext-registry-remot...

High Availability ★

JFrog Bintray Distribution

JFrog XRay

JFrog CLI

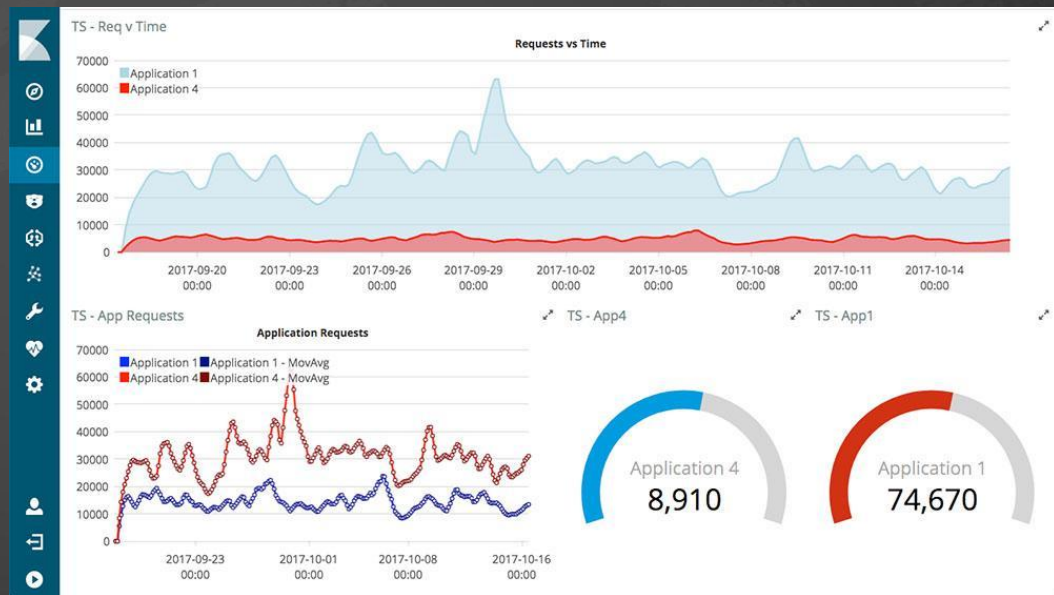
DEPLOY

- Ne plus faire les déploiements avec Jenkins
- Dév maison (PHP + Perl)
- Adossé à Rundeck
- Couplé à notre GCL : livraison & déploiement automatique
- Chaque typologie de déploiement est normalisée
- SQL, DTS, Java, Angular
- Vision de l'état du déploiement de chaque composant



LOG ME TENDER

- Stack ELK
- Un seul endroit pour tous les logs (JEE, ESB, Boot, ...)
- Kibana



Bilan de la phase 3 (2014 - 2019)



- GIT / GITHUB
- JENKINS 2
- PIPELINES
- SONARQUBE
- ARTIFACTORY
- DEPLOY
- LOG MANAGEMENT



- 1 SEULE SHARED LIB

ET LE CODE ?



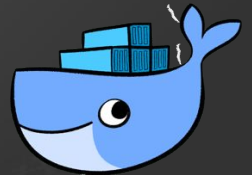
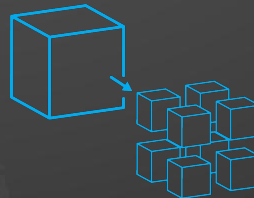
SOA

- J+1 vers le temps réel
- SOAP vers REST
- ESB / API Gateway
- Nécessité d'avoir quelque chose de plus léger que JEE
- Vers les micro services ?



SPRING ... ON RESPIRE

- JEE trop rigide
- Emancipation des développeurs
- Spring*
- Fat jar
- Ouverture vers les micros services et Docker



SPRING*

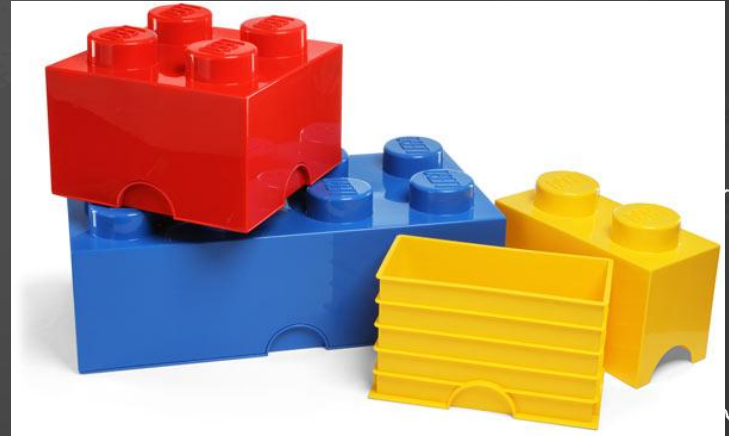


1.5 puis 2.X



4.X

SEEDS & FRAMEWORKS



ANGULAR C'EST COOL ... VRAIMENT ?

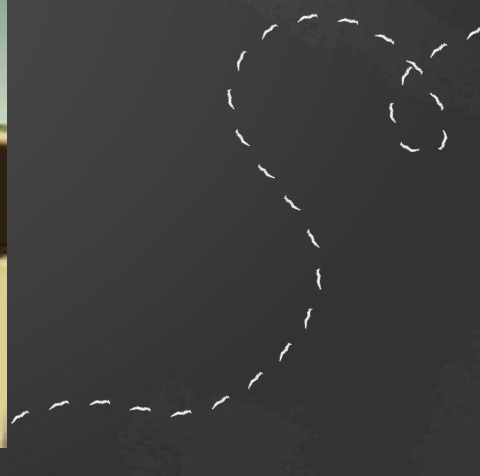
- Remplacer JSF / jQuery
- Avoir des applications Web riches
- Emancipation des développeurs
- Angular*



SEEDS & FRAMEWORKS



L'EST COOL QUAND L'EST SIMPLE



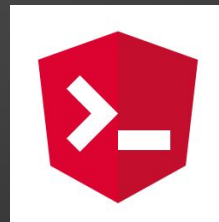
ANGULAR*



1.X



~~2.X, 4.X,~~
~~6.X, 8.X, 9.X~~
et 10.X



DTS & BO ... GOODBYE

- N'est plus géré par notre usine logicielle
- BO est géré par une équipe décisionnelle
- DTS par l'équipe historique



PHASE 4 : 2019 - 20??

AUTOMATISATION

Jarvis

VERS L'INFINI ET L'AU-DELA

L'avenir

⇒ AUTOMATISATION

Jarvis

DEPLOIEMENT CONTINU

- Intégration Continue
- Livraison Continue
- Déploiement continu jusqu'en recette
- MEP instrumentée (mais non continue)



LA PIC EVOLUE

2ème version des shared lib Jenkins:

- plus petites
- plus modulaires
- plus évolutives
- plus testables

Sonarqube 7.x LTS

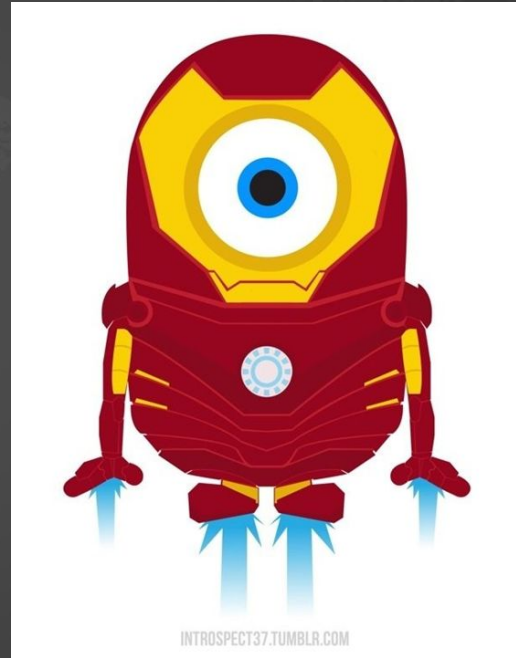
- Analyse multi-branches
- Décorations des PR

xRay



JARVIS ... RENDRE LA PIC AUX DEVS

- Développement interne
- Bootstrap de projets:
 - Jenkins
 - GitHub
 - Sonar
 - Seed
- Indicateurs projets
- Modification de projets
- Tâches administratives



Bilan de la phase 4 (2019 - 20???)



- PLUSIEURS SHARED LIBs
- JARVIS
- AUTOMATISATION
- LIBERTÉS AUX DEVs



- ANGULAR ?
- ???

PANORAMA DES TECHNOLOGIES HARMONIE MUTUELLE 2019

BASES DE DONNÉES & STOCKAGE



MESSAGING



ANNUAIRES



DATA MANAGEMENT

TRAITEMENT DE LA DONNÉE



MOBILE



LANGAGES & FRAMEWORKS



BACK & API

OpenSource

SOA, SCRUM, CI/CD, API, DevOps, IaaS, PeerReviewing, StandardsWeb, Microservices, Architecture, Stateless, LogManagement, JSON, Conteneurisation, InnerSource, Modularité, TestUnitaires, RESTful

FRONT



SOCLES & INFRASTRUCTURE

SOLUTIONS D'INFRASTRUCTURE



SOCLES LOGICIELS



USINE LOGICIELLE





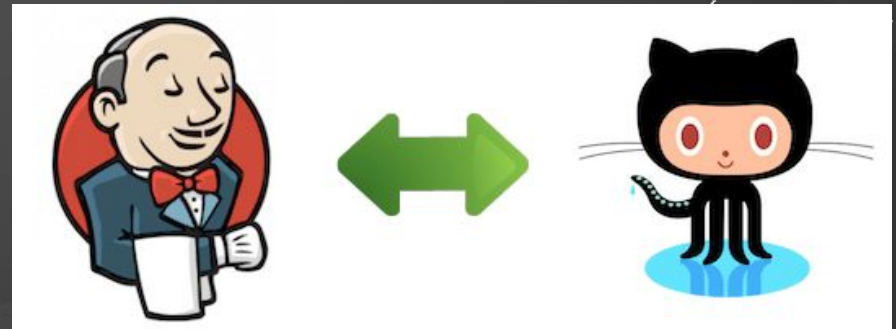


➡ VERS L'INFINI ET L'AU DELA

L'avenir

DEV to PROD

- Dev to production
- Qualimétrie et tests sur pull request
- GitHub Actions ?



OPENSIFT

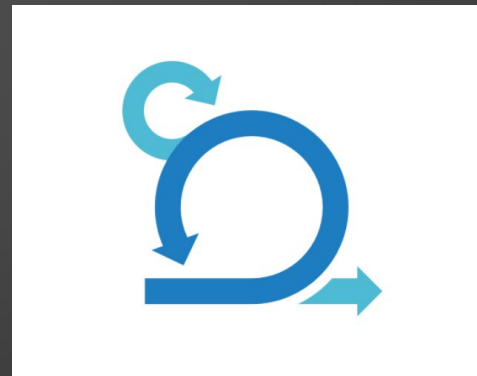
- Simplifier la PIC elle-même
- Simplifier le packaging & déploiement des applications
- Simplifier l'utilisation de nouvelles technos
- Rationaliser les machines
- Unifier les déploiements, l'administration et la supervision



RED HAT®
OPENSIFT

DEVOPS

- Fluidifier les MEP et les relations dev et production
- Approche tool first, une bonne idée ?
- Cohabitation legacy et nouveaux projets
- DevSecOps ?



OUVERTURE DES TECHNOS

- La techno la plus adaptée au besoin ou au développeur
- Dans un environnement cadré et responsable



À grand pouvoir, grandes responsabilités.

Spiderman (Ben Parker)

www.citation-celebre.com

Siil ne fallait retenir que 2 ou 3 choses ...

- Eviter de trop cadrer et restreindre les dévs au risque de passer son temps à jouer au gendarme et au voleur
- Impliquer les développeurs (inner source)
- Casser les silos
- Attention au [“hype driven development”](#)
- S’outiller pour les tâches ingrates (qualimétrie par exemple)
- Rationaliser les builds (sharedlib Jenkins)



MERCI !

