

# REX DSI Inra



9 années d'évolution des pratiques de développement

Thomas Lallart, DSI Inra

Ecole Informatique IN2P3, Lyon, 28/09 - 03/10 2015



# Unité DSI-PILS



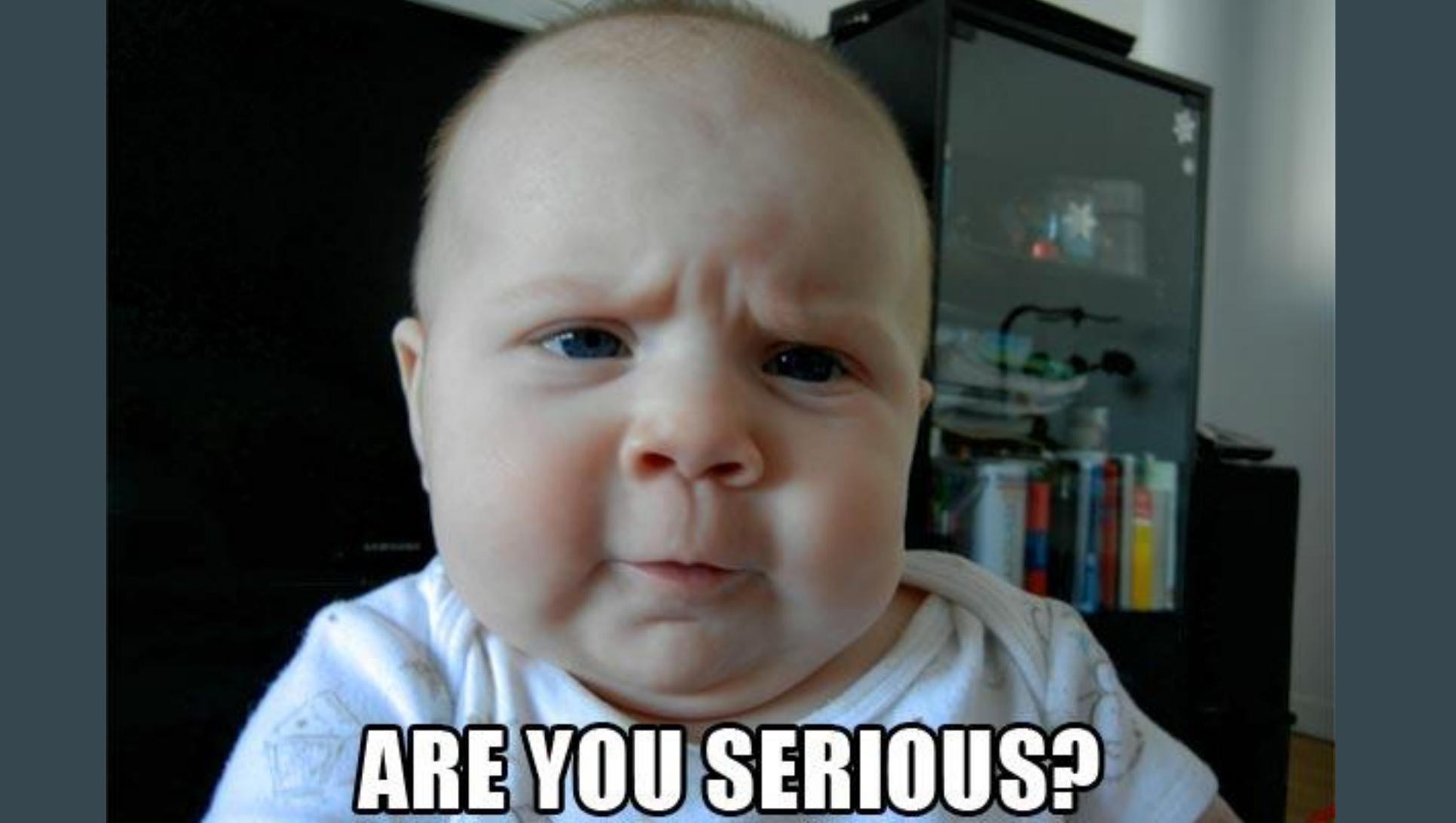
- Actuellement 10 personnes internes + 4 à 12 prestataires externes
- Activités :
  - Responsable des référentiels institutionnels (Personnes, Structures, Activités, Productions)
  - Urbanisation du SI et interopérabilité : création de services (Web, fichiers, Messages) d'exposition, transformation ou échanges de données
  - Autres petits projets périphériques : annuaires, applications de gestion de l'annuaire Ldap, Mantis,...
  - Conseils et accompagnement en architecture et développement pour les autres unités de la DSI
- Projets en cours :
  - Services REST de recherche, composition et agrégations de données référentielles (Elastic Search)
  - API Manager

# 2006

- 4 chefs de projets
- 4 développeurs
- Aucune harmonisation
  - Chacun développe comme il veut
  - Chacun livre comme il peut
  - Chacun exploite à sa façon
  - Chacun documente pour lui
- Pas de gestionnaire de version
  - On s'envoie les codes en zip ou jar par mail
- Pas ou peu de tests
  - Une méthode main dans certaines classes qui println des choses... seul celui qui l'a écrit sait ce qui est attendu

# 2006

- Pas de bugtracker sur le projet
- Pas de qualimétrie
- Aucune pratique collective, pas même de Javadoc
- Spécialisation
  - Un développeur développe le modèle... et seul lui connaît ce code
  - Un développeur développe la BD et les DAO... et seul lui les comprend
  - Un développeur développe l'IHM... et personne d'autre ne sait le faire
- Packaging : clic droit dans Eclipse, Export Jar/War
- Versionnement : `mv projet.jar projetX.Y.jar`
- Pas d'archivage, pas de traçabilité des déploiements
- Planification : chacun à un TODO.txt à la racine de son projet

A close-up photograph of a baby with a serious, slightly pouting expression. The baby has light skin and blue eyes. The background is dark, with a bookshelf visible on the right side. The text "ARE YOU SERIOUS?" is overlaid at the bottom in a bold, white, sans-serif font with a black outline.

**ARE YOU SERIOUS?**

# 2006 - 2008

- Mise en place de Subversion
  - 2 développeurs parmi les 4 l'utilisent (!)
  - discussion méta-physiques sur le développement : inertie, peur du changement
- Bug déclarés dans Mantis
  - Tracés, corrigés, testés, clôturés
- Passage ponctuels d'outils de contrôles de qualité de code
  - PMD, Checkstyle, Findbugs
  - Réactions sporadiques mais pas intégré en tant que pratique
- Premiers tests JUnit et Cactus
  - Parcelles et pas automatisés

A close-up photograph of a ginger cat's face. The cat has large, dark, expressive eyes and a white patch on its muzzle. The background is a soft, out-of-focus green. The text "I hope you" is overlaid in white serif font at the top, and "feel better soon" is overlaid at the bottom.

I hope you

feel better soon

# 2009-2010

- Très gros projet confié à l'unité
- +2 développeurs CDD
- +2 développeurs titulaires
- Nouveau responsable d'équipe, Big Bang :
  - Mode projet Agile (Scrum)
  - Tests automatisés
  - Intégration continue
  - Build (Ant)
  - Repository d'artefacts
  - Revue de code et revue de conception

**FAIL**



Failure is not  
falling down but  
refusing to get  
up. ~Chinese proverb

# 2010-2012

- Amélioration progressive et avec accompagnement
  - a. Ajout de tests unitaires
  - b. Remplacement des builds Ant par Maven
  - c. Intégration continue (Hudson puis Jenkins)
  - d. Mise en place de Sonar
  - e. Mise en place d'Archiva
  - f. Ajout de tests d'intégration
- En parallèle
  - a. Atelier technique mensuel de partage sur
    - Les principes de conception
    - Les tests, les builds, les outils
  - b. Un wiki décrivant l'utilisation des outils, les pratiques d'équipe, les processus (release, développements), les procédures (installation, mise à jour, logs)
  - c. Un transition agile plus en douceur

Success.



# 2012-2015

- Amélioration continue des fonctionnalités de l'usine logicielle
  - Tests de performance (JMeter)
  - Déploiement automatisé (plugin Jenkins pour Tomcat, scripts shell pour les autres)
  - Tests d'intégration directement sur serveur d'application embarqué (Arquillian)
  - Quelques tests IHM
  - Pipeline Jenkins
- Nouveaux outils :
  - TestLink : référentiels des tests fonctionnels
  - IceScrum : gestion de backlog et de sprint
  - Sélénium : tests interfaces Web

# Les échecs

- Intégration BugTracker - SCM : intégration à faire manuellement, plus complexe et moins intéressant que dans un outil déjà intégré (forge)
- Tests avec BD mémoire : difficulté à créer un schéma et des données sous base mémoire (H2, Derby) qui corresponde au schéma réel (Oracle, Postgres) et coût de maintenance des scripts
- Maîtrise d'outils (Maven) encore délicat pour certains

# Mode de fonctionnement

- Agile :
  - Auto-organisation
  - Responsabilisation individuelle et collective
  - Amélioration continue
- Rétrospective mensuelle sur nos pratiques
- Les décisions sont prises par l'équipe qui décide d'elle-même quoi améliorer en fonction des difficultés rencontrées
- Fédération des équipiers autour des pratiques facilitée
- Usine logicielle avec droits d'administration pour tout le monde

# Gains

- Sérénité!
  - Grâce à la fiabilisation et la sécurisation de plusieurs activités du développement
  - Serein lors des modifications de code, même sur de “vieux” projets
  - Serein lors des livraisons
- Motivation
  - Equipiers fiers de ce qu’ils font ; les indicateurs qualité et rapport de tests sont les résultats de leur travail
  - Envie de faire un peu mieux à chaque projet
- Réduction des bugs remontés après mise en production
- Facilite l’intégration d’un nouvel équipier
- Facilite les diagnostics : cadre de test défini, détecter l’origine d’un problème est simplifié

# Gains

- Capacité de livrer tôt et souvent
  - d'impliquer l'utilisateur régulièrement, de recueillir du feedback
  - de rapidement changer les priorités d'un développement
  - réduction drastique de l'effet tunnel
- Communication vers l'utilisateur et vers les décideurs
  - disposer des rapports de tests
  - montrer la qualité du logiciel produit
  - justifier des actions de refactoring ou de mise à jour
- Régressions de plus en plus rares
- Dynamique d'amélioration continue
- Diminution du ratio de dette technique sur les projets

# Risques

- Communication
  - Ne pas fédérer les personnes autour de ces concepts
  - Ne pas accompagner les mutations
- Essentiellement humain
  - Tirer l'excellence technique suppose que toute l'équipe suit le rythme
  - Pragmatiquement, équipe hétérogènes, compétences diverses
  - Attention à ne pas laisser quelqu'un derrière
- Technique
  - Rester dans la zone de maîtrise, si trop complexe, arrêter, prendre du recul et recommencer autrement
- Opérationnel
  - L'objectif avant tout reste de livrer un logiciel opérationnel ; ne pas confondre les priorités
  - Le meilleur indicateur de qualité reste la satisfaction utilisateur, pas les métriques



# Perspectives

- Déploiement automatisé par invocation de jobs Rundeck
- Traçabilité des déploiements
- Passage de Subversion vers Git/Gitlab
- Automatisation côté systèmes et middleware : en relation avec les équipes  
Systèmes autour de Puppet, Rundeck et Docker
- Intégration des tests d'acceptation / tests fonctionnels dans les développements et dans l'usine : Fitness? Cucumber?
- Tests sur la sécurité logicielle : audit de code sécurité et/ou passage automatique de scanners de vulnérabilités
- Passage de Maven à Gradle

**I HAZA QUESTION**



19037.jpg