

15/09/2009



Paris  
JUG

[www.parisjug.org](http://www.parisjug.org)

[www.parisjug.org](http://www.parisjug.org)





15/09/2009

## Squale - Software QUALity Enhancement

<http://www.squale.org>

Fabrice Bellingard

- *Squale Project Lead* -



[www.parisjug.org](http://www.parisjug.org)



Copyright © 2008 ParisJug. Licence CC – Creative Commons 2.0 France – Paternité – Pas d'Utilisation Commerciale – Partage des Conditions Initiales à l'Identique



« You can't control what you can't measure »

*Tom DeMarco, 1986*

# Intervenant

## • Fabrice Bellingard

- 7 années en architecture et développement des SI
- ... et 8 années dans le monde de l'open-source
  - Eclipse Core
  - Eclipse plugins
  - Apache Maven
  - Apache Archiva
  - ... et maintenant Squale, chez Qualixo !
- Une autre passion : l'autre « SI » – Solidarité Internationale –
  - WWF, Planète Urgence



# Sommaire

- **Introduction en questions**
- **Les grands principes de Squale**
- **Une démo !**
- **Squale aujourd'hui et demain**

# Outils qualité : pour qui, pour quoi ?

## Exemples de points de vue différents

(source : Club Qualimétrie - <http://www.clubqualimetrie.fr>)

- Développeur
  - Détecter des parties suspectes de son code, potentiellement zones à risque
- Chef de projet
  - Contractualiser un niveau d'exigences avec ses fournisseurs
- Top-manager
  - Maîtriser la complexité de son SI, donc les coûts

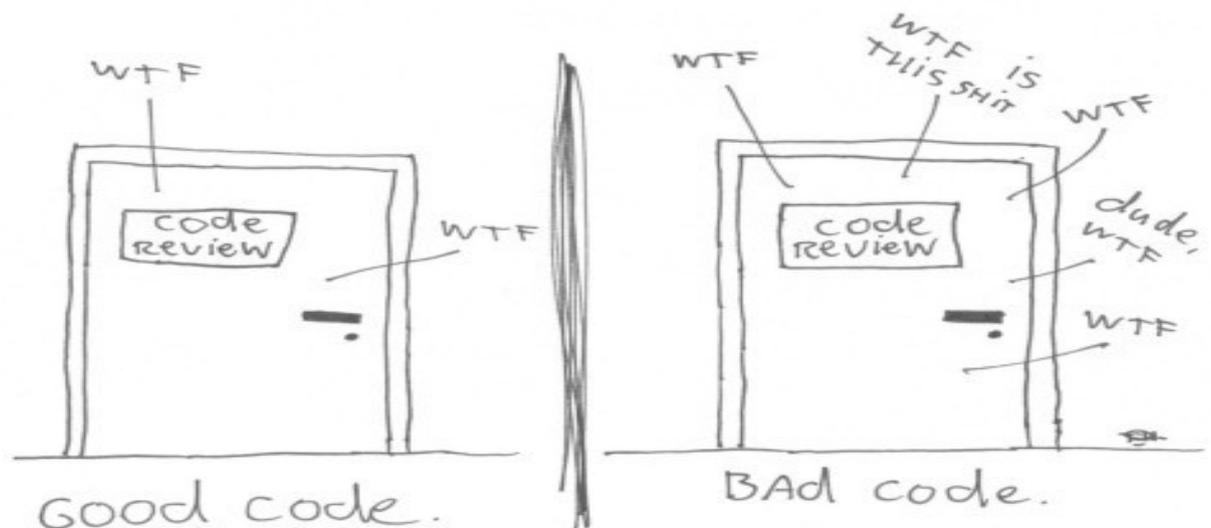
→ **Analyser, contrôler, maîtriser**

# Des métriques ?

## • Pléthores de métriques existantes

- Sens ?
- Justesse de l'algorithme ?
- Applicabilité ?
- Seuils ?

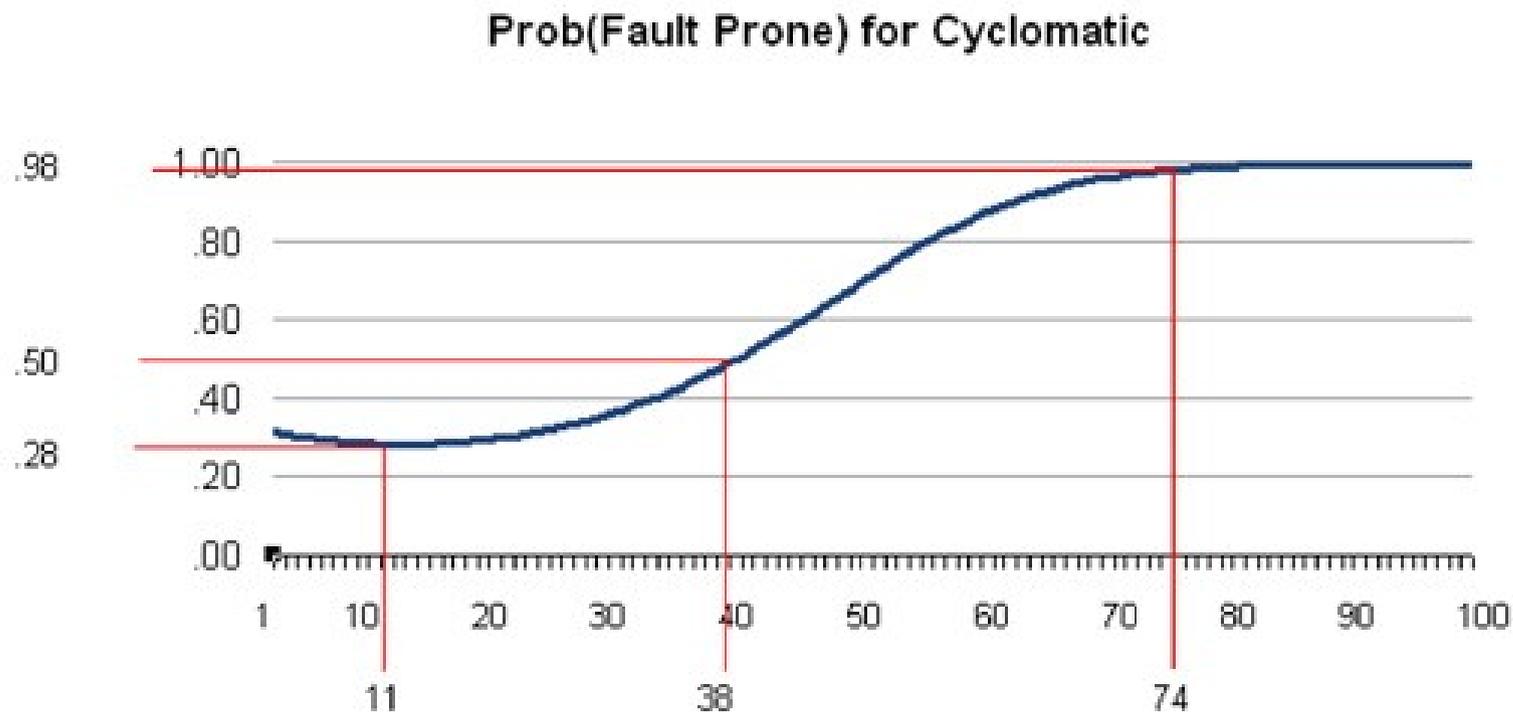
The ONLY VALID MEASUREMENT  
OF CODE QUALITY: WTFs/MINUTE



(c) 2008 Focus Shift/OSNews/Thom Holwerda - <http://www.osnews.com/comics>

# Qualité : des dogmes faciles ?

- « La probabilité de bug est proportionnelle à la complexité cyclomatique »



<http://www.enerjy.com/blog/?p=198>

# Trop haut-niveau ?

## • Modèle de la norme ISO-9126

- Capacité fonctionnelle
- Fiabilité
- Facilité d'utilisation
- Rendement
- Maintenabilité
- Portabilité



→ **Comment calculer ces indicateurs ??**

# Sommaire

- Introduction en questions
- **Les grands principes de Squale**
- Une démo !
- Squale aujourd'hui et demain

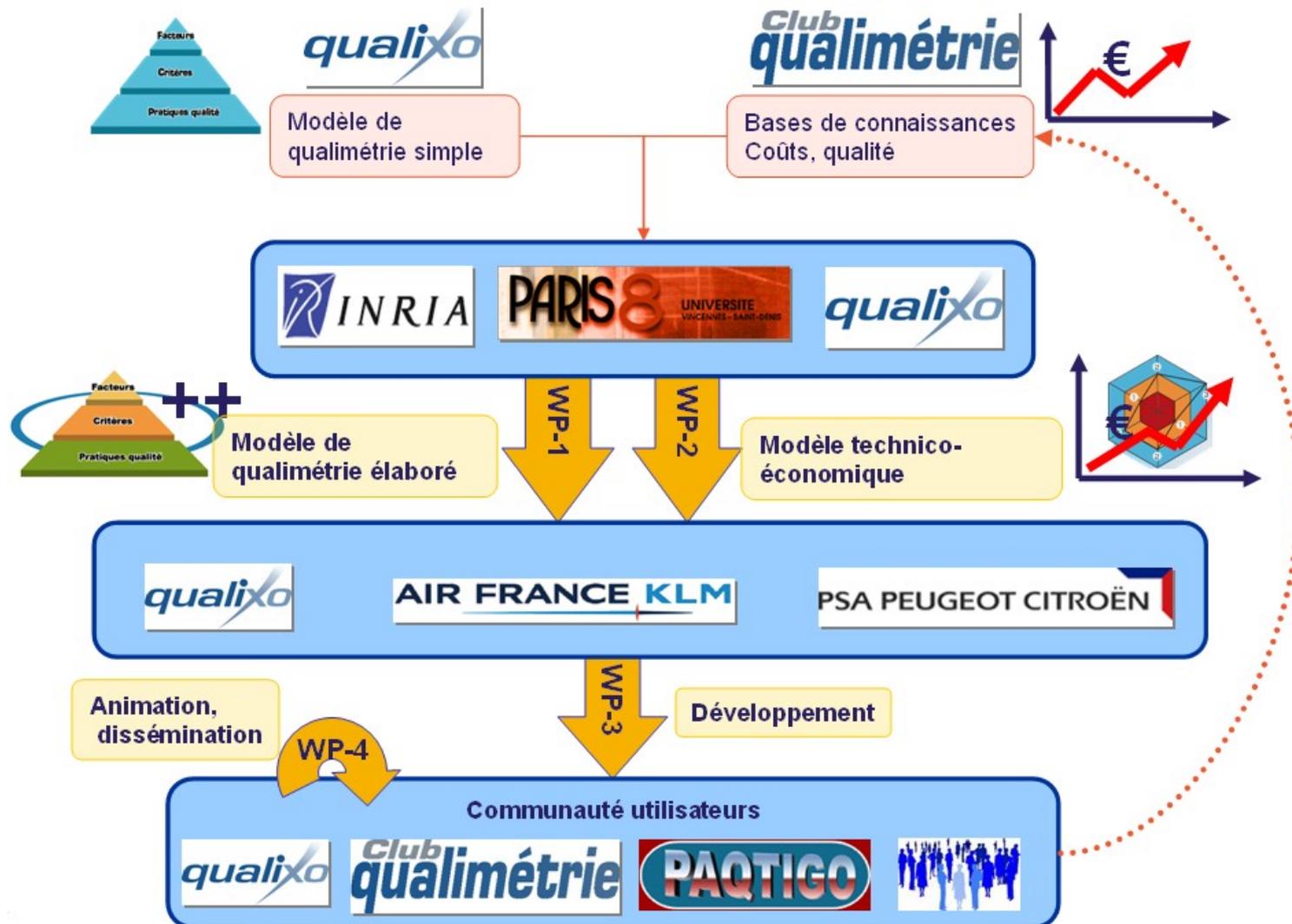
# Software QUALity Enhancement

- **Squale, une plateforme qualimétrie open-source offrant :**
  - Des modèles issus de travaux de recherche
    - Modèles qualité élaborés
    - Modèles technico-économiques
  - Une application d'analyse et de restitution de la qualité
    - Basée sur ces modèles
    - Ciblant plusieurs langages et plateformes
    - Ouverte et extensible
    - Vision de la qualité adaptée à différents profils



# Pôle de compétitivité System@tic

- Squale est un projet du groupe thématique « Logiciel Libre »
- Organisation et partenaires :



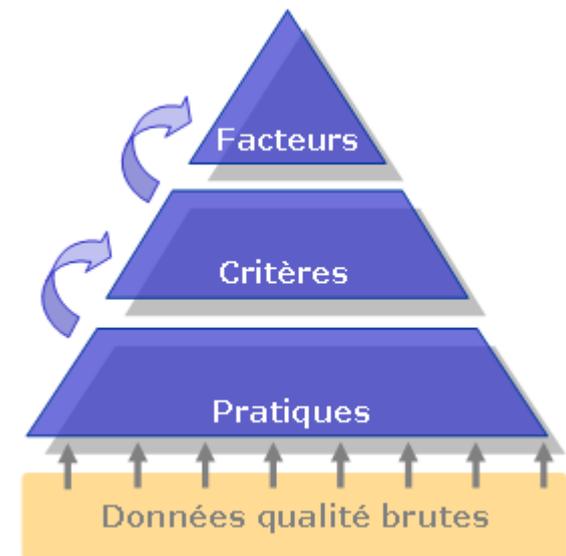
# Squale – Les modèles

## ■ Travaux de recherche avec INRIA Lille et Paris 8

- Issus de l'expérience de Qualixo et validés chez les grands comptes
- S'inspire de l'état de l'art : Standard ISO 9126, Travaux McCall, Approche GQM

## ■ Approche sur 4 niveaux :

- Facteurs (6) : ex. « Maintenabilité »
- Critères (13+) : ex. « Homogénéité »
- Pratiques : ex. « Règles de nommage »
- Données qualité brutes (ex. métriques de code)



## ■ Notes des facteurs/critères/pratiques normalisées :

- Situées dans l'intervalle continu [ 0 – 3 ]
- Définissent un niveau d'exigence



# Squale – Les modèles (2)

## ■ Modèles mathématiques customisables

### ■ Formules basées sur une ou plusieurs données brutes

- Pour une meilleure identification

### ■ Formules continues

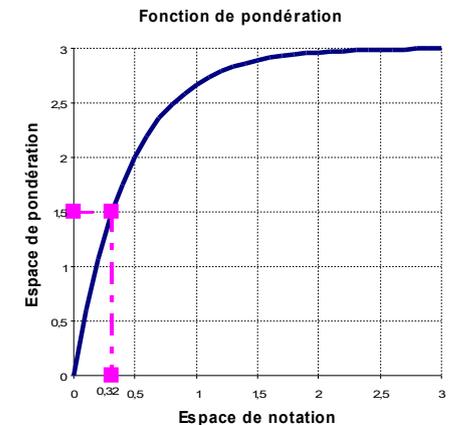
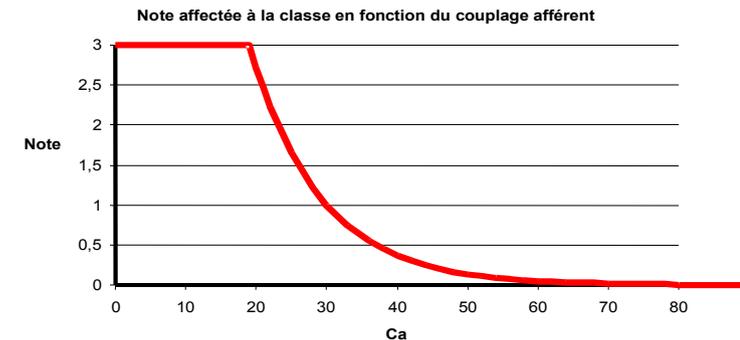
- Pour éviter les effets de seuils

### ■ Formules paramétrables

- Durcir ou assouplir le calcul d'une note

### ■ Formules complexes

- Permettant de limiter les effets de bords



$$Note = P^{-1} \left( \frac{1}{N} \sum_{n=1}^N P(NC_n) \right)$$

# Squale – L'application

Cycle de vie du projet

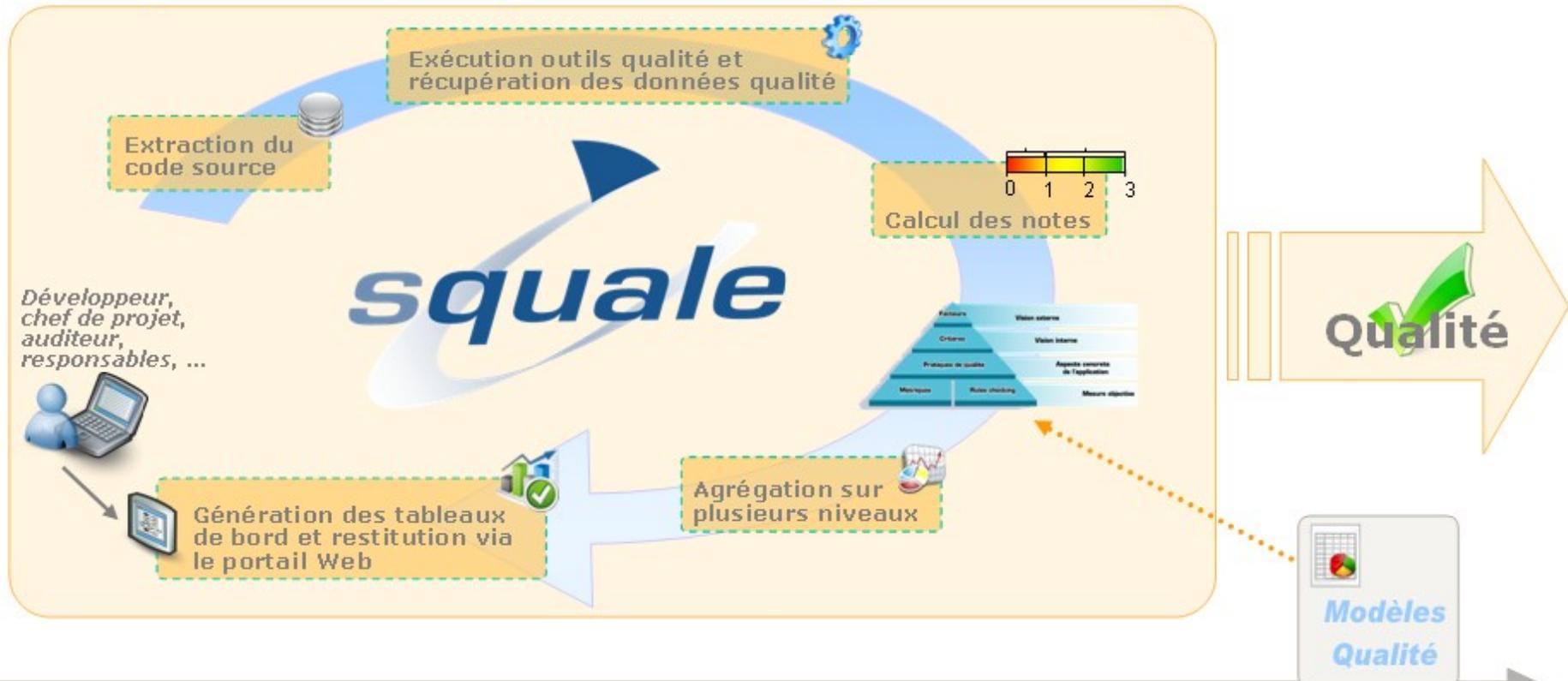
Spécifications ... Conception ... Développement ... Intégration ... Recette ... Production ... Maintenance ...

Documents,  
UML, ...

Référentiel  
Source

Couverture,  
Tests, ...

Monitoring,  
Bug tracking, ...



Elaboration du modèle ... Validation projet pilotes ... Déploiement démarche ... Conduite du changement ...

Démarche qualimétrique

# Squale – L'application (2)

## • Grandes fonctionnalités :

- Support de plusieurs langages
  - Java : out-of-the-box avec les Checkstyle & Co.
  - C/C++/Cobol : connecteur McCabe
  - PHP et autres à venir
- Support de plusieurs modèles
  - Pour les applications hétérogènes
  - Modification en ligne des grilles qualité
  - Insertion de notations manuelles
- Notion d'audits
  - de suivi – exécution régulière, purge automatique
  - de jalon – on demand, pas de purge

# Squale – L'application (3)

- IHM Web
  - Gestion des différents profils
  - Home page personnalisable
  - Navigation guidée, selon plusieurs axes :
    - Drill-down niveau modèle
    - Drill-down niveau source
    - Temporelle
    - Tops
    - Evolution
  - Proposition de plans de remédiation
- Référentiel des applications analysées
- Statistiques et exports divers

# Sommaire

- Introduction en questions
- Les grands principes de Squale
- **Une démo !**
- Squale aujourd'hui et demain

# Une démo !

The screenshot displays the Squalle application interface. At the top, there is a navigation menu with 'Accueil', 'Applications', 'Nouvelle application', 'Référentiel', and 'Administration'. The main content area is titled 'Résultats de l'application squale'. It includes a sidebar with 'Synthèse', 'Projets', and 'Audits'. The main content shows the application name 'squale', the audit date '05 novembre 2008', and the previous audit date '29 octobre 2008'. Below this is a section for 'SYNTHÈSE DES RÉSULTATS DE L'APPLICATION' with tabs for 'Facteurs', 'Kiviat', and 'Volumétrie'. The 'Kiviat' tab is active, showing two quality grids. The first grid is for 'squale-java-ee' and the second is for 'squale-java-se'. Each grid has columns for 'Projet', 'Architecture', 'Capacité fonctionnelle', 'Evolutivité', 'Maintenabilité', 'Fiabilité', and 'Réutilisabilité'. The 'EXPORTS' section at the bottom includes a 'Déconnexion' button and two buttons: 'Exporter en PDF' and 'Restitution d'audit'. A sidebar on the left contains links for 'Aide', 'Rechercher', 'Contact', and 'Statistiques'.

Application : [squale](#)

Audit : 05 novembre 2008  
Audit précédent : 29 octobre 2008

### SYNTHÈSE DES RÉSULTATS DE L'APPLICATION

Facteurs Kiviat Volumétrie

► Aide

#### Grille qualité : squale-java-ee

Projet	Architecture	Capacité fonctionnelle	Evolutivité	Maintenabilité	Fiabilité	Réutilisabilité
<a href="#">squaleWeb</a>	1.2 → ☀️	---	1.6 → ☀️	1.9 → ☀️	---	1.7 → ☀️

#### Grille qualité : squale-java-se

Projet	Architecture	Capacité fonctionnelle	Evolutivité	Maintenabilité	Fiabilité	Réutilisabilité
<a href="#">squaleCore</a>	1.8 → ☀️	---	1.8 → ☀️	2.2 → ☀️	---	1.9 → ☀️
<a href="#">squalix</a>	1.9 → ☀️	---	2.0 → ☀️	2.2 → ☀️	---	2.0 → ☀️

### EXPORTS

Déconnexion ✕  
Aide ?  
Rechercher 🔍  
Contact ✉️  
Statistiques ✕

Exporter en PDF Restitution d'audit

# Sommaire

- Introduction en questions
- Les grands principes de Squale
- Une démo !
- Squale aujourd'hui et demain

# Squale – Aujourd'hui

- **En production chez Air France et PSA**

- 100+ applications, ~ 13 millions de ligne de code
  - Java, C/C++, Cobol
- Modèles customisés et validés
- Application mûre, production-ready
  - Architecture distribuée chez Air France



- **En téléchargement sur le site officiel :**

- Distribution full open-source permettant l'analyse d'applications Java sur un modèle simple
  - Basée sur Checkstyle, PMD, JavaNCSS & Co.
- Nécessite des outils commerciaux pour C/C++/Cobol

# Squale – A venir

## ■ Modèles :

- Amélioration des modèles qualité existants (→ *standardisation* ?)
- Amélioration des plans de remédiation
- Modèle de calcul du ROI d'un processus qualimétrique

## ■ Application :

- Amélioration de l'ergonomie et de l'IHM (visualisations)
- Extensions à d'autres langages
- Extensions à d'autres données

## ■ Référentiel mutualisé

- Centralisation de données d'analyse pour :
  - Valider plus largement les modèles et les améliorer
  - Fournir des statistiques globales



# ***Conclusion***

[www.parisjug.org](http://www.parisjug.org)



Copyright © 2008 ParisJug. Licence CC – Creative Commons 2.0 France – Paternité – Pas d'Utilisation Commerciale – Partage des Conditions Initiales à l'Identique



# Conclusion

- « You can't control what you can't measure »
  - However, recently...
    - Software Engineering: An Idea Whose Time Has Come and Gone?
- **Mots clés de Squale :**
  - Vision de la qualité adaptée à différents profils : du suivi haut-niveau pour top-managers à une utilisation pragmatique pour le développeur
  - Modèles qualimétriques évolués
  - Analyse multi-langages, Java en full open-source
  - Recherche constante (plans de remédiation, ROI, ...)

# Bibliographie / liens

## ■ Site Squale

- <http://www.squale.org>

## ■ 25<sup>th</sup> IEEE International Conference on Software Maintenance

- <http://icsm2009.cs.ualberta.ca/schedb.php#sept23>

## ■ Le Club Qualimétrie

- <http://www.clubqualimetrie.fr>

## ■ Tom DeMarco

- 1986. Controlling Software Projects: Management, Measurement, and Estimates. Prentice Hall, ISBN 0131717111

- 2009. Software Engineering: An Idea Whose Time Has Come and Gone? IEEE Software, Viewpoints.

## ■ McCabe Cyclomatic Complexity: the proof in the pudding

- <http://www.enerjy.com/blog/?p=198>



# *Questions / Réponses*

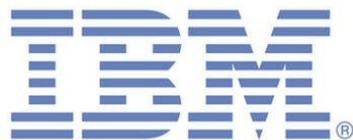
[www.parisjug.org](http://www.parisjug.org)



Copyright © 2008 ParisJug. Licence CC – Creative Commons 2.0 France – Paternité – Pas d'Utilisation Commerciale – Partage des Conditions Initiales à l'Identique



# Sponsors



# ***Merci de votre attention!***



[www.parisjug.org](http://www.parisjug.org)



# Licence



Paternité-Pas d'Utilisation Commerciale-Partage des Conditions Initiales à l'Identique  
2.0 France

- <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>