



LES JEUDIS DE L'OBJET



Squale

Le portail qualimétrie open-source

29 janvier 2009

- *Fabrice BELLINGARD* -



- Qualixo

- 2005, JEI spécialisée en qualité logicielle
- Activités principales : audits, démarche qualimétrique, expertise qualité logicielle
- Clientèle : Air France, PSA, GDF, EDF, BNP, Generali, ...
- <http://www.qualixo.com>

- Mon expérience

- 7 années en architecture et développement des SI
- ... et 7 années dans le monde de l'open-source
 - Eclipse Core
 - Eclipse Plugins
 - Apache Software Fondation
 - ... et Squale !



Plan

- La qualimétrie
- Introduction à Squale
- Focus sur les modèles
- Squale : le projet de recherche
- Squale : le projet open-source
- Synthèse





La qualimétrie

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

Tom DeMarco

qualixo

La qualimétrie ?

- qualimétrie = mesure de la qualité
- Une démarche qualimétrie logicielle, c'est :
 - Un référentiel qualimétrique composé :
 - De données qualité brutes unitaires
 - De pratiques avec leur formule de notation
 - De **critères et facteurs** avec formules d'agrégation
 - Un outillage
 - Pour la production de métriques et la récupération de données diverses
 - Pour l'**agrégation** et le calcul des facteurs/critères pratiques
 - Pour l'administration des audits et données qualimétriques
 - Pour la **restitution** aux différents profils utilisateurs
 - Un processus
 - De définition des **exigences** qualimétriques (avec éventuellement une annexe au CCTP pour la contractualisation des forfaits)
 - De **contrôles** qualimétriques intégrables dans la **méthodologie** interne SI et dans les processus qualité (ex. CMMI)
 - L'organisation correspondante





Introduction à Squale

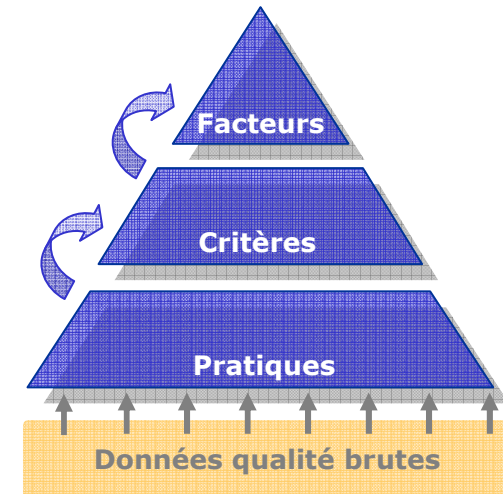
Qu'est-ce que Squale ?

- Un projet :
 - Open-source
 - Début 2009 : 10 committers et 10 contributeurs
 - De recherche
 - Pôle de compétitivité System@tic
 - Partenaires : Air France, PSA, INRIA, LIASD, Qualixo, Paqtigo
- Concrètement : *répond au métier de la qualimétrie !*
 - **Des modèles**
 - Qualimétriques multi-langages
 - Technico-économiques
 - **Un logiciel**
 - Récupération de données diverses (ex. : métriques de code)
 - Connecteurs vers outils commerciaux, open-source, autres, ...
 - Agrégation basée sur les modèles choisis
 - Portail Web de restitution des indicateurs qualité

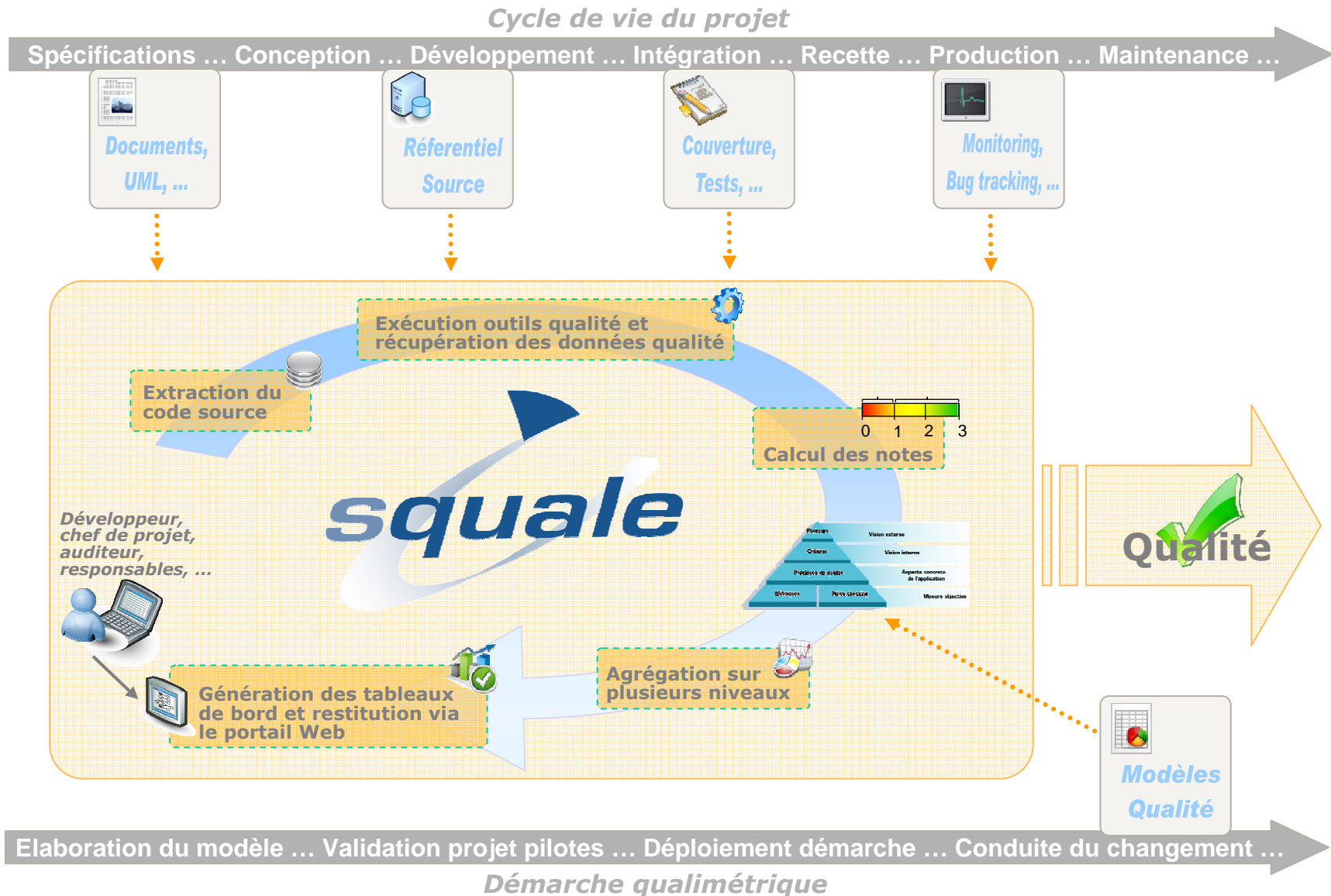


Squale – les modèles

- Issus initialement de l'expérience de Qualixo
 - Démarches mises en place et validées chez de 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 (25+) : ex. « Règles de nommage »
 - Données qualité brutes (ex. métriques de code)
- Notes des facteurs/critères/pratiques normalisée :
 - Situées dans l'intervalle continu [0 → 3]

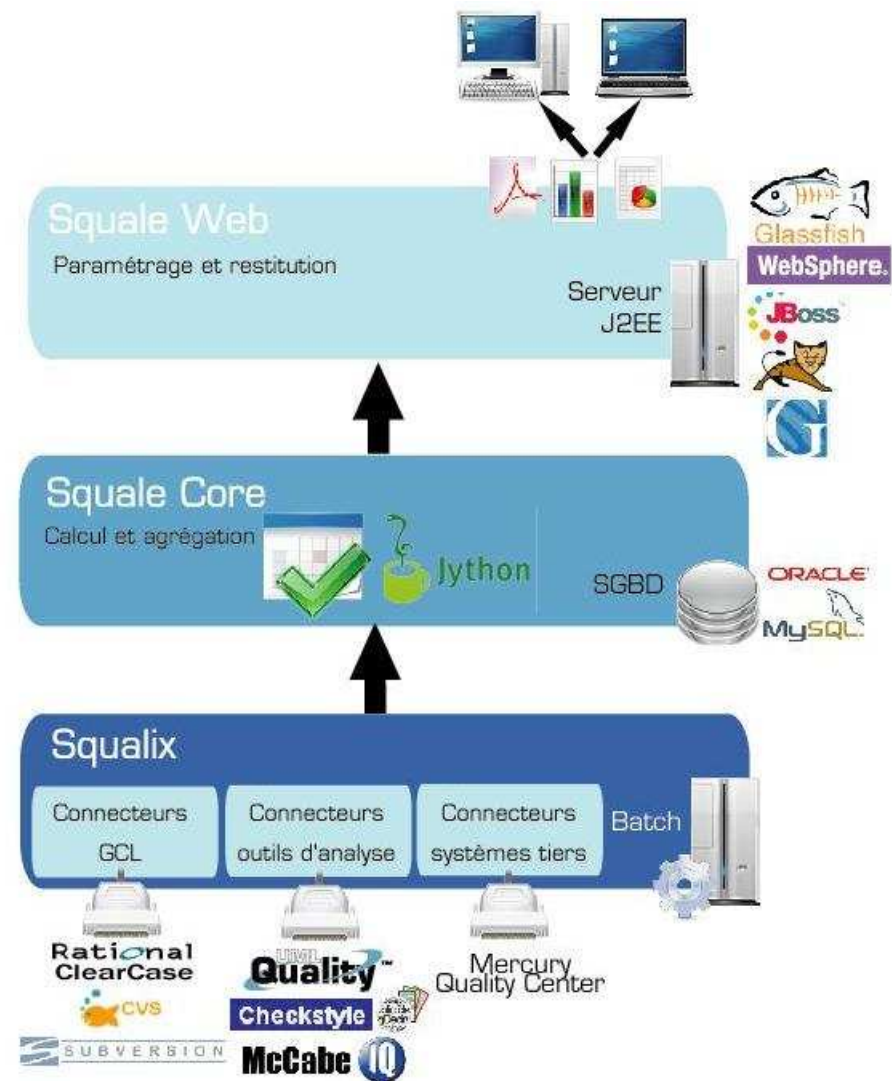


Squale – le logiciel : principe de fonctionnement

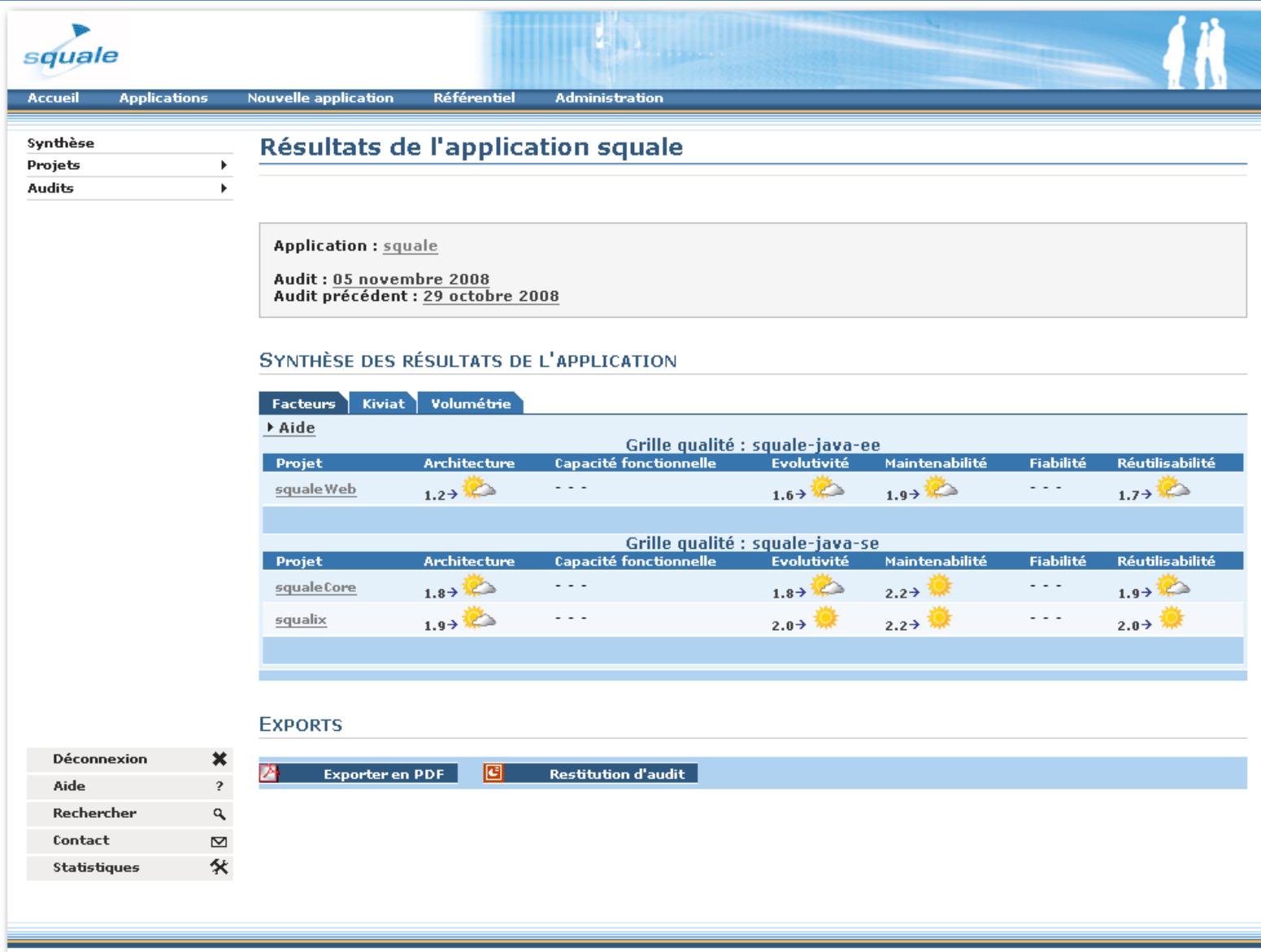


Squale : un logiciel ouvert et extensible

- **Adresse plusieurs langages**
 - Java, C/C++, Cobol
 - Autres à venir
- **Serveur compliant JEE 1.3**
- **Bases de données Oracle et MySql actuellement supportées**
 - Ajout simple d'autres SGBD
- **Mécanisme de connecteur extensible**
 - Connecteur GCL
 - CVS, SVN, ClearCase, Archive, ...
 - Connecteur outils génération de métriques
 - Commerciaux (ex. McCabe)
 - Open-source (ex. Checkstyle, JavaNCSS, CKJM, Macker, ...)
 - Connecteur systèmes tiers



Squale : une démonstration ?



The screenshot displays the Squale 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 shows the application name 'squale', the audit date '05 novembre 2008', and the previous audit date '29 octobre 2008'. Below this, a section titled 'SYNTHÈSE DES RÉSULTATS DE L'APPLICATION' contains two quality grids. The first grid, 'Grille qualité : squale-java-ee', shows scores for 'squakeWeb' across various categories. The second grid, 'Grille qualité : squale-java-se', shows scores for 'squakeCore' and 'squalix'. At the bottom, there are 'EXPORTS' options: 'Exporter en PDF' and 'Restitution d'audit'. A sidebar on the left contains links for 'Synthèse', 'Projets', 'Audits', 'Déconnexion', 'Aide', 'Rechercher', 'Contact', and 'Statistiques'.

Application : squale
Audit : 05 novembre 2008
Audit précédent : 29 octobre 2008

SYNTHÈSE DES RÉSULTATS DE L'APPLICATION

Facteurs Kiviati Volumétrie

Aide

Grille qualité : squale-java-ee

Projet	Architecture	Capacité fonctionnelle	Evolutivité	Maintenabilité	Fiabilité	Réutilisabilité
squakeWeb	1.2 → ☁	---	1.6 → ☁	1.9 → ☁	---	1.7 → ☁

Grille qualité : squale-java-se

Projet	Architecture	Capacité fonctionnelle	Evolutivité	Maintenabilité	Fiabilité	Réutilisabilité
squakeCore	1.8 → ☁	---	1.8 → ☁	2.2 → ☀	---	1.9 → ☁
squalix	1.9 → ☁	---	2.0 → ☀	2.2 → ☀	---	2.0 → ☀

EXPORTS

Déconnexion ✕
Aide ?
Rechercher 🔍
Contact ✉
Statistiques 📊

Exporter en PDF Restitution d'audit

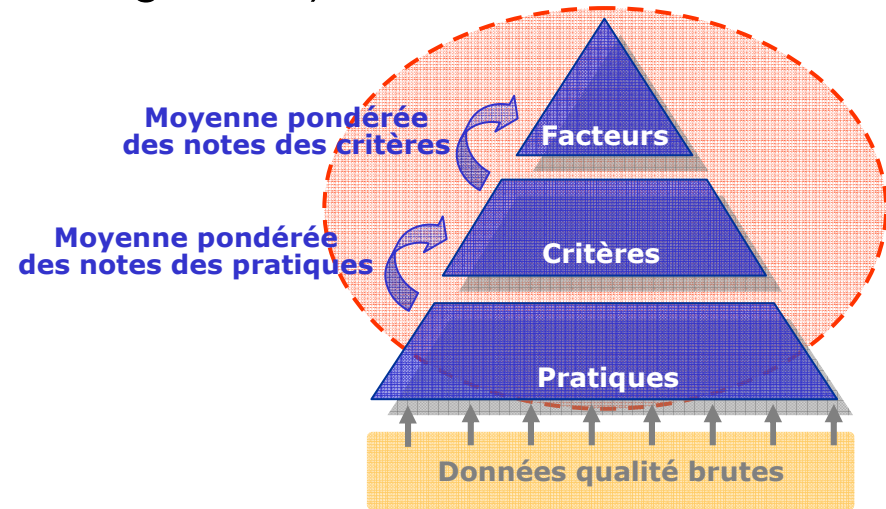


Focus sur les modèles

Modèle « Qualixo »

- Facteurs : haut niveau (~ top management)

- Capacité fonctionnelle
- Architecture
- Évolutivité
- Maintenabilité
- Réutilisabilité
- Fiabilité

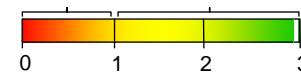


- Critères = vue boîte noire (~ chef de projet)

- Pratiques = vue boîte blanche (~ développeur)

- Automatisables
- Non automatisables

- Notes normalisées dans l'intervalle 0 → 3



Modèle qualité sous forme de grille : exemple

Facteurs		Critères		Pratiques	
Maintenabilité	Homogénéité		Pratique "Normes et standards" : mise en forme		
			Pratique "Normes et standards" : nommage		
			Pratique "Normes et standards" : programmation		
	Compréhension		Pratique "Documentation"		
			Pratique "Profondeur héritage"		
			Pratique "Spécialisation de la classe"		
			Pratique "Qualité de la documentation"		
			Pratique "Normes et standard" - documentation		
	Simplicité		Pratique "Spaghetti code"		
			Pratique "Nombre de méthodes"		
			Pratique "Taille procédure"		
	Niveau d'interdépendance		Pratique "Couplage classe efférent"		
			Pratique "Couplage classe afférent"		

Déclinaison et notation du facteur "Qualité" de la maintenabilité

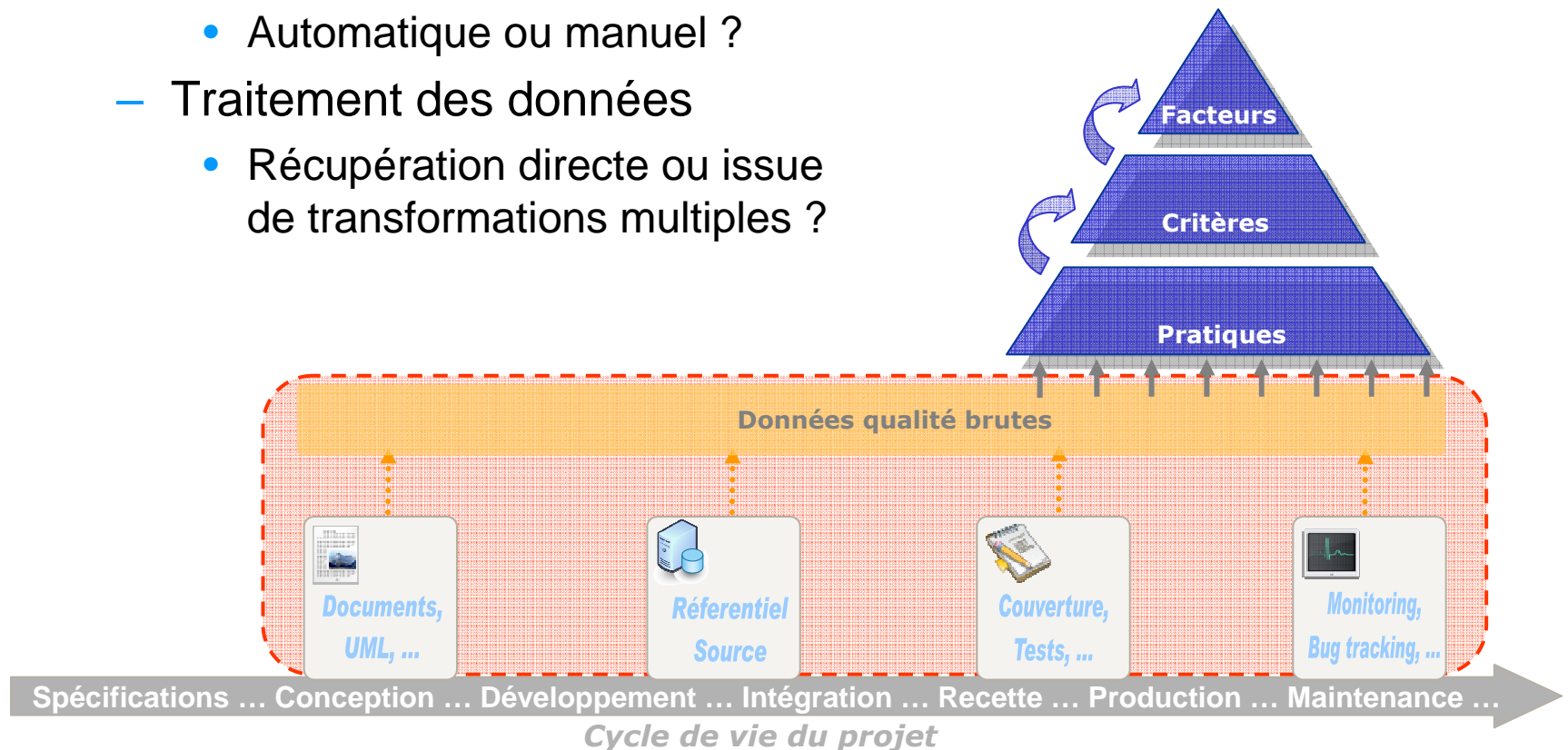
Facteurs		Critères		Pratiques	
Evolutivité	Homogénéité		Pratique "Normes et standards" : mise en forme		
			Pratique "Normes et standards" : nommage		
			Pratique "Normes et standards" : programmation		
	Compréhension		Pratique "Documentation"		
			Pratique "Profondeur héritage"		
			Pratique "Spécialisation de la classe"		
			Pratique "Qualité de la documentation"		
			Pratique "Normes et standard" - documentation		
	Modélisation		Pratique "Diagrammes de modélisation"		
			Pratique "Pré-détection d'antipatterns"		
			Pratique "Conformité entre modélisation et implémentation"		
			Pratique "Raisonnement par les modèles"		
	Modularité		Pratique "Couplage Classe efférent"		
			Pratique "Copier-coller"		
			Pratique "Cohésion Classe"		
			Pratique "Couteau suisse"		

Déclinaison et notation du facteur "Qualité" de l'évolutivité



Les données qualité brutes

- Caractéristiques
 - Accessibilité dans le temps
 - Cycle de vie du projet
 - Mode de mise à disposition
 - Automatique ou manuel ?
 - Traitement des données
 - Récupération directe ou issue de transformations multiples ?



Exemple de données issues du code

- Les pratiques liées à de l'analyse de code sont les plus automatisables
 - Analyse statique
 - Rule-checking
 - Vérification de règles (nommage, formatage, ...etc)
 - Métriques de code
 - Métriques structurelles
 - Métriques orientées objet
 - Métriques d'architecture
 - Analyse dynamique
 - Métriques de couverture des tests
- ... mais attention : elles n'arrivent qu'au commencement du développement !
 - Défaut détecté plus tôt (ex. conception) = coût moindre
 - Ex. : pratique « Conception – modèle UML »



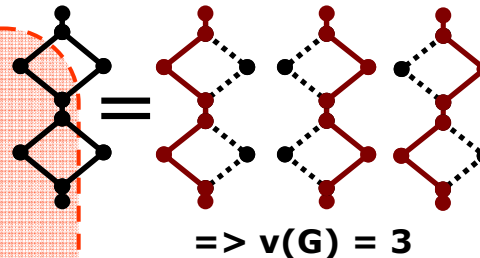
Exemple 1 : métriques structurelles

Métriques		Définition
$v(g)$	[M, C, P, A]	complexité cyclomatique de la méthode, déterminant le nombre de chemins de linéaires qu'il faut parcourir pour couvrir l'ensemble du code de la méthode, de la classe.
$eV(g)$	[M, C, P, A]	complexité essentielle de la méthode. Elle illustre le niveau de déstructuration du code
$IV(g)$	[M, C, P, A]	complexité de conception modulaire. Elle caractérise entre autre le nombre de chemins de tests nécessaires pour tester l'intégration du module avec les autres modules de l'application

M : Méthode, C : Classe, P : Package, A : Application

- **$v(g)$: Complexité cyclomatique**

- Nombre de chemins de tests indépendants
 - Effort de test à fournir
 - Poids de logique

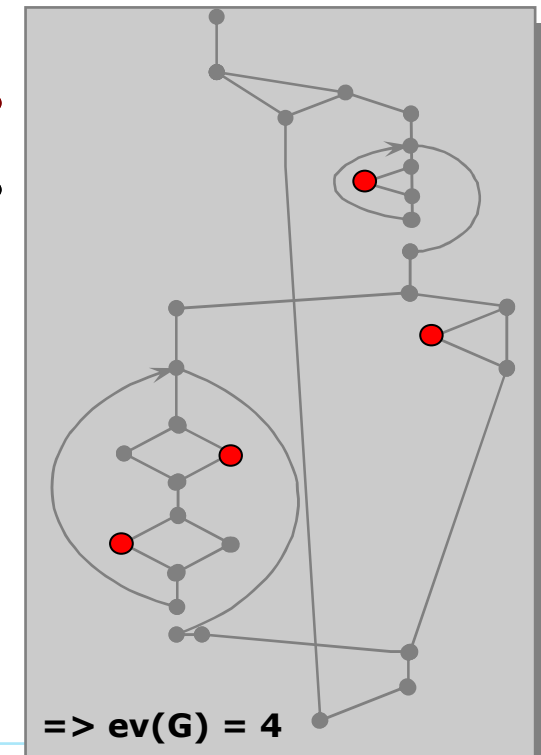


- **$ev(g)$: Complexité cyclomatique essentielle**

- $V(g)$ du graphe simplifié
 - Niveau de déstructuration
 - Corrélé à la maintenabilité

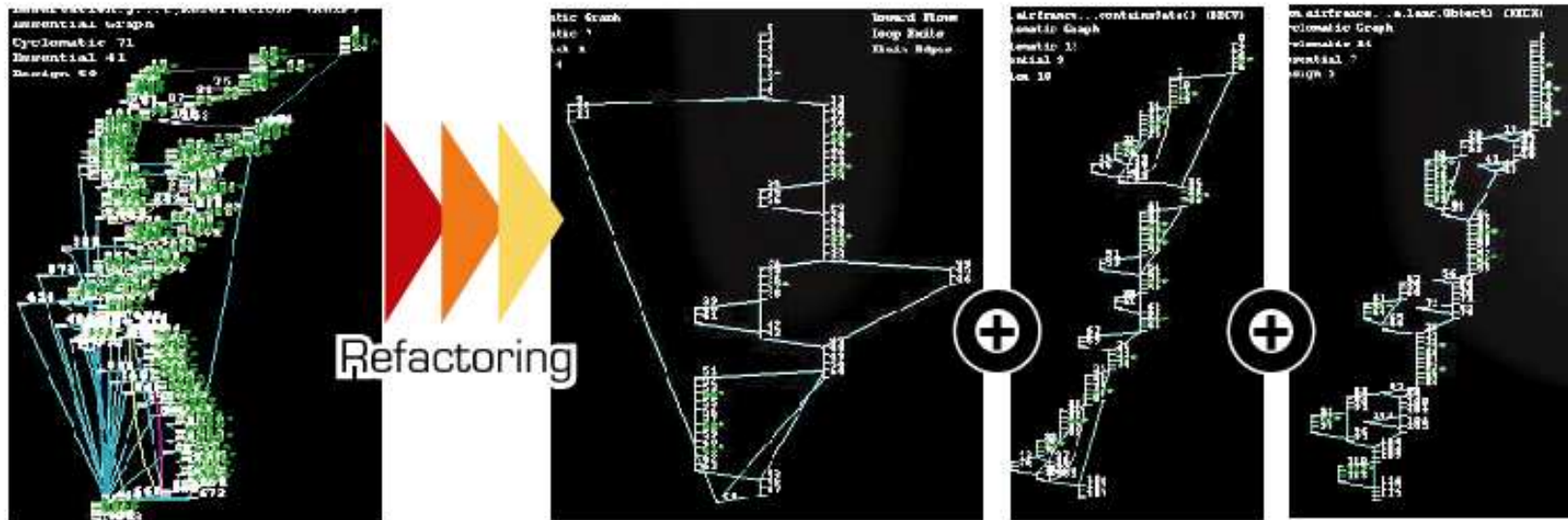
- **$iv(g)$: Complexité d'intégration**

- $V(g)$ du graphe simplifié hors appel externe
 - Niveau de déstructuration
 - Corrélé à l'effort d'intégration



Exemple 1-bis : le plat de spaghetti

- Exemple de méthode ayant donné lieu à refactoring

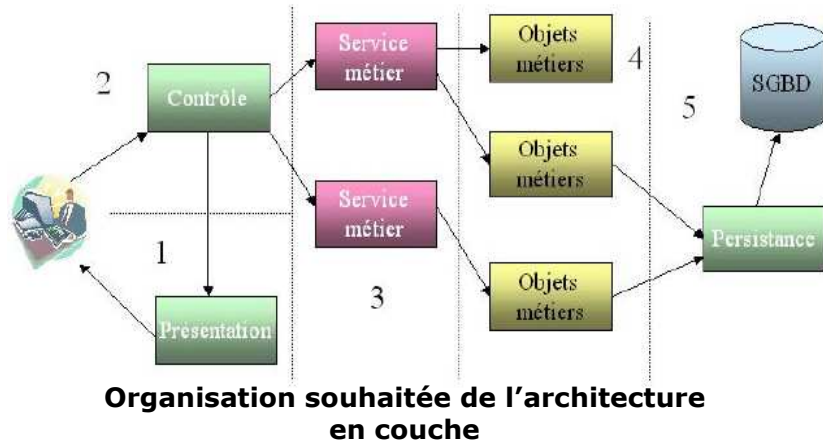


Avant : LOC : 167
v(g) : 71
ev(g) : 41

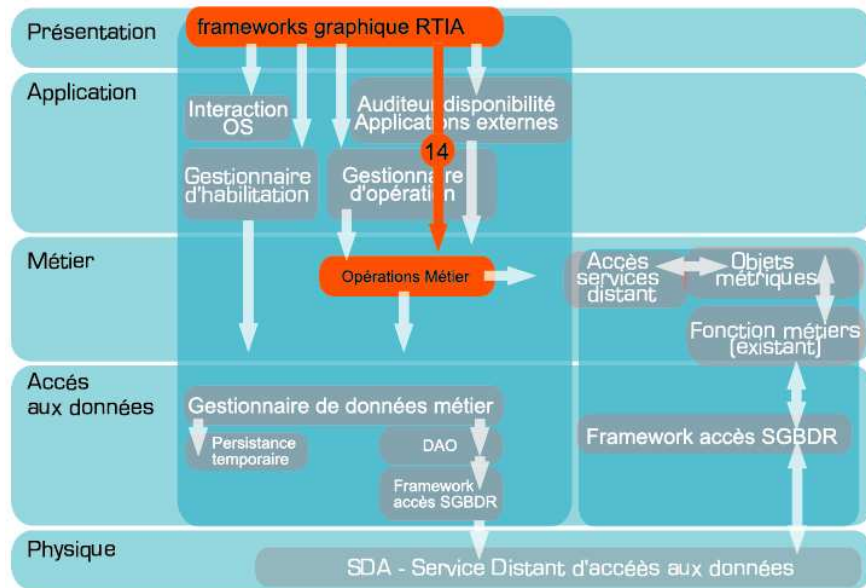
Après : LOC : 22+30+26
v(g) : 7+13+14
ev(g) : 1+9+7



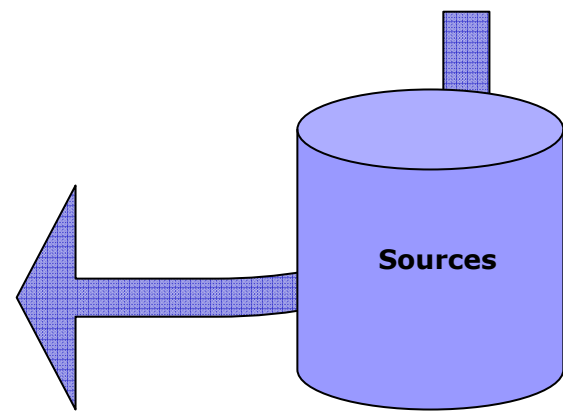
Exemple 2 : métriques d'architecture



Définition des contraintes d'architecture



Synthèse des violations détectées avec si besoin la référence aux méthodes incriminées



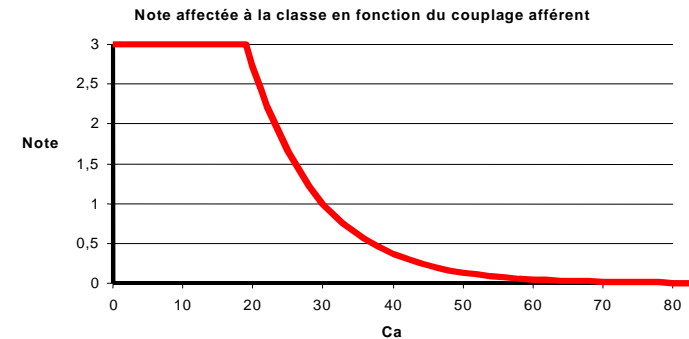
Analyse des sources



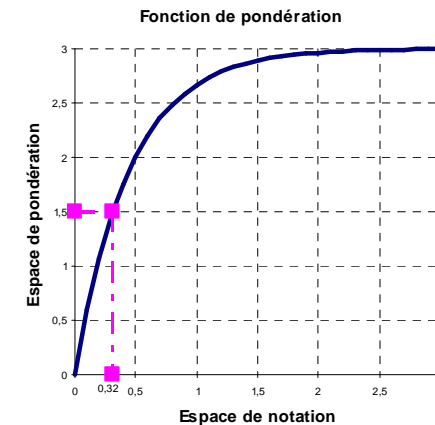
Passage des métriques aux pratiques

- Note d'une pratique = formule basée sur une ou plusieurs métriques

- Formules continues
 - Éviter les effets de seuils



- Formules paramétrables
 - Durcir ou assouplir le calcul d'une note
→ Espace de pondération



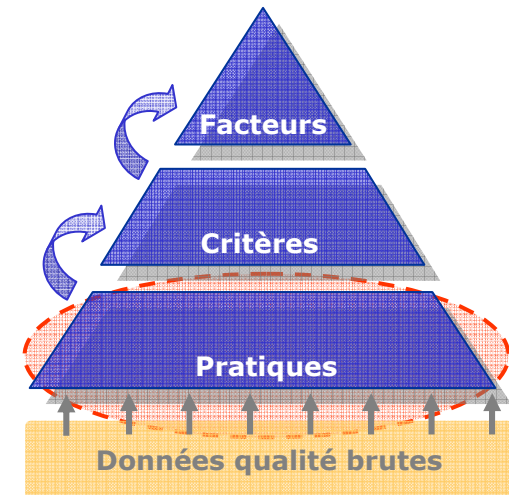
- Formules mathématiques élaborées
 - Limiter les effets de bords

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



La pratique = point fort du modèle

- Une pratique :
 - A un sens concret
 - Facile à comprendre par un programmeur
 - Très « pragmatique »
 - Tente de combiner plusieurs métriques pour aider à détecter des cas suspects
 - Exemple : on souhaite s'assurer que les parties les plus complexe d'un code soient aussi celles qui soient le plus testées
 - Pratique « Tests unitaires » combinant :
 - Taux de couverture de tests
 - Complexité cyclomatique ou essentielle
 - Utilise des formules mathématiques élaborées
 - Pas une simple moyenne pondérée !





Squale : le projet de recherche

Le FUI et les pôles de compétitivité



- **Fonds Unique Interministériel**
 - Fonds pour financer des projets de recherche labellisés par des pôles de compétitivité
 - Projets collaboratifs impliquant entreprises et labo de recherche
 - Coordination par la DGE (Direction Générale des Entreprises)
 - FUI doté de 720 M€ sur la période 2006-2008
- **Le Pôle de compétitivité System@tic Paris – Région**
 - 1 des 71 pôles de compétitivité labellisés par le FUI
 - Dimension internationale
 - 5 groupes thématiques, dont « Logiciels Libres »
 - Budget 2006 des projets de R&D du Pôle : > 280 M€/an



Et Squale dans tout ça ?

- Squale :
 - A été présenté au bureau exécutif du groupe thématique Logiciel Libre de System@tic, et a reçu son soutien
 - *début 11-2007*
 - Puis a été labellisé par le directoire du pôle System@tic
 - *28-11-2007*
 - Avant d'être analysé par les soins de la DGE
 - *Q1 2008*
 - Pour finalement être accepté pour financement dans le cadre du 5^{ème} appel à projets du FUI
 - *mai 2008*
 - Le projet de recherche Squale a officiellement démarré en juin 2008
 - Pour une période initiale de 2 ans

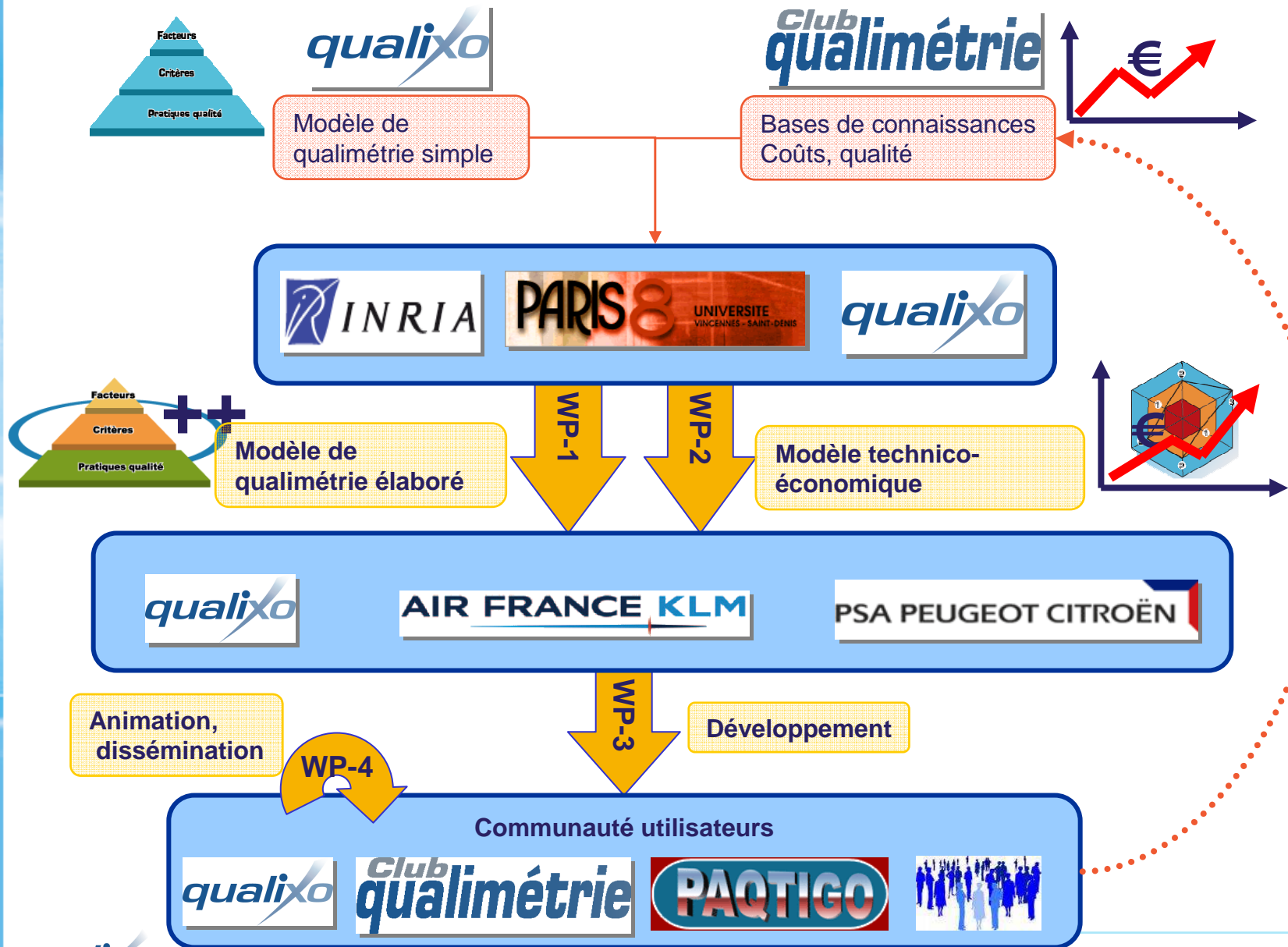


Les différents partenaires

- Industriels
 - Air France – DSI
 - Arnaud Poivre, responsable du centre de compétences Tests et qualité logicielle
 - PSA Peugeot Citroën – DSI
 - Thierry Bey, responsable entité Langages, Qualité et Processus de Développement
- PME
 - Qualixo (*pilote du projet*)
 - Philippe Vaillergues et Fabrice Bellingard, expertise en qualimétrie
 - Paqtigo
 - Amin Popote, expertise en gouvernance du SI
- Académiques
 - INRIA – Lille
 - Stéphane Ducasse, spécialiste en maintenance et génie logiciel
 - Laboratoire LIASD – Université Paris 8
 - Françoise Balmas, spécialiste en compréhension, représentation et animation de programmes



Squale : le projet de recherche System@tic



WP 1 : Modèles qualimétriques



- Acteurs : INRIA (*pilote*), LIASD, Qualixo
- Objectif :
Améliorer les modèles qualimétriques existants et leur mode de visualisation / représentation
- Découpage :
 - WP 1.1 : **Caractérisation des pratiques sensibles via métriques pour Java et C++**
 - Étendre la liste des métriques existantes et les caractériser
 - WP 1.2 : **Visualisation graphique pratiques**
 - Représentation donnant un sens immédiat à la pratique
 - WP 1.3 : **Modèle enrichi d'agrégation des pratiques pour Java et C++**
 - Améliorer les formules d'agrégation permettant de passer des métriques aux facteurs de haut niveau



WP 2 : Modèles technico-économiques



- Acteurs : INRIA (*pilote*), LIASD, Qualixo, Paqtigo
- Objectif :
Définir les modèles permettant de corrélérer les indicateurs techniques à des indicateurs économiques
- Découpage :
 - WP 2.1 : Modèle de « l'effort » de création/modification de composants à partir de métriques
 - WP 2.2 : Modèle d'estimation de l'effort d'amélioration de la qualité suite à un diagnostic qualimétrique
 - Permettant de déterminer des plans d'action de remédiation
 - WP 2.3 : Modèle d'estimation du retour sur investissement de la qualimétrie
 - Suffisamment générique pour devenir un standard mais suffisamment paramétrable pour s'adapter aisément



WP 3 : Développement en open-source



- Acteurs : Qualixo (*pilote*), AF, PSA, INRIA, LIASD
- Objectif :
Améliorer l'application Squale existante en se basant sur :
 - les travaux de recherche des WP 1 et 2
 - les besoins exprimés par les grands comptes
- Découpage :
 - WP 3.1 : Ordonnanceur et interfaces récupération de métriques
 - WP 3.2 : Implémentation modèles de calcul et d'agrégation
 - WP 3.3 : IHM administration paramétrage
 - WP 3.4 : IHM de restitution : navigation, tableaux de bords, ...
 - WP 3.5 : Moteur de visualisations graphiques des pratiques
 - WP 3.6 : Moteur de plan de remédiation avec charges
 - WP 3.7 : Interface avec le projet System@tic « Helios »
 - WP 3.8 : Animation communauté développeurs



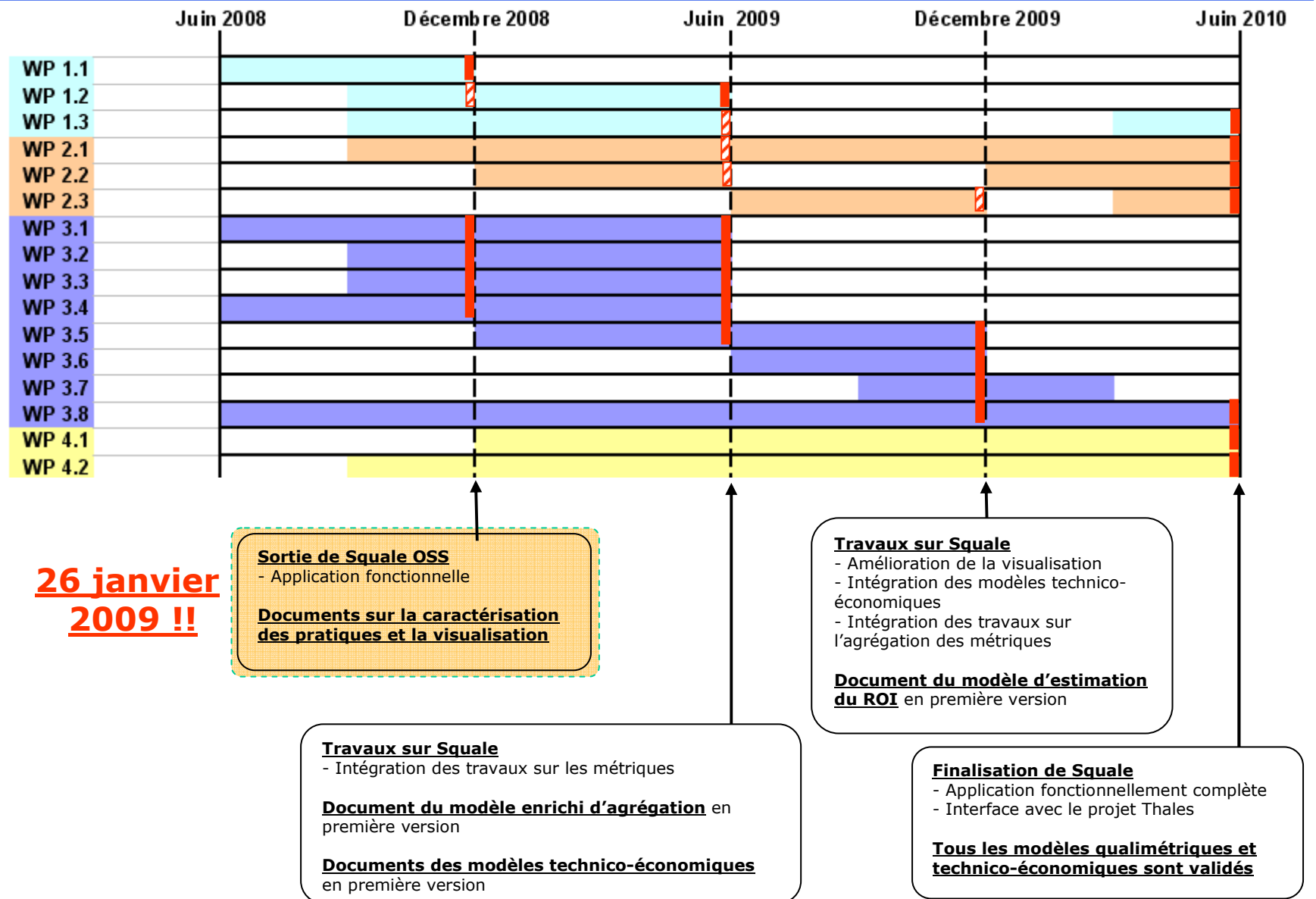
WP 4 : Dissémination et animation communauté utilisateurs



- Acteurs : Qualixo (*pilote*), Paqtigo
- Objectif :
Faire valider largement les résultats des 3 premiers WP par les utilisateurs, bénéficier de leur retour et de leurs données
- Découpage :
 - WP 4.1 : **Animation communauté**
 - Mettre en place les moyens de communication
 - Assurer la bonne dissémination des résultats du projet
 - WP 4.2 : **Constitution base de connaissances partagée**
 - Sauvegarder, dans une base de donnée centralisée, les données brutes et agrégées générées par Squale
 - Utiliser ces données pour :
 - comparer des applications entre elles
 - alimenter la recherche sur les modèles (WP1 et WP2)





Roadmap du projet Squale





Squale : le projet open-source

Le projet open-source et les licences

- Les livrables du projet de recherche doivent être disponibles sous licence de type « libre »
 - Imposé par la charte du groupe thématique « Logiciel Libre » du pôle System@tic
 - Pour Squale
 - Partie recherche (WP1 et WP2) : 
 - Creative Commons Attribution-Noncommercial-Share Alike 3.0
 - Partie logiciel Squale (WP3) : 
 - GNU Lesser General Public License version 3
 - Concrètement :
 - Squale peut être utilisé librement
 - Toute modification de Squale doit être redistribuée sous licence LGPL
- Le projet open-source est donc le réceptacle des résultats du projet de recherche
 - Mais il aura sa propre vie, comme tout projet open-source



La première version open-source de Squale

- Livrée le 26 janvier 2009 !
- Périmètres couverts par le code open-source
 - Java : OK
 - Basé sur des outils d'analyse de code open-source
 - C/C++ : OK
 - Nécessite des outils d'analyse de code commerciaux pour le moment
 - Cobol : en cours
 - Sera basé sur des outils d'analyse commerciaux
- Distribution disponible : optimale pour le périmètre Java
 - Embarque toutes les technologies open-source nécessaires
- Infos supplémentaires
 - Site officiel du projet open-source : <http://www.squale.org>
 - Démonstration : <http://saas.qualixo.com/demo-squale/>
 - user/user ou squaleadmin/admin



Squale : le projet OSS

- Equipe
 - Actuellement
 - Membres du projet de recherche
 - Gestion et décisions par Qualixo et Air France
 - Demain
 - Gouvernance « classique » d'un projet OSS « à la Apache »
 - Par vote
 - Décisions par les « PMC » de l'équipe
 - Adhésion à l'équipe Squale :
 - Ouverte à tous, mais requiert un vote interne des PMC
 - Basée sur la méritocratie
 - Evolution classique
 - Utilisateur → Contributeur → Committer → PMC

Périmètre équipe



Les ressources du projet Squale (1/2)

- Le site officiel du projet open-source
 - URL : <http://www.squale.org>
 - Point d'entrée pour tout le reste
- Référentiel de source Subversion
 - URL : <https://svn.squale.org/squale/>
- Outil de gestion de projet Trac
 - URL : <https://project.squale.org>
 - Contenu : Wiki, Bug and issue tracker, Timeline et roadmap
- Serveur d'intégration continue Hudson
 - URL : <http://hudson.squale.org>



Les ressources du projet Squale (2/2)

- Principales mailing lists :
 - users@squale.org
 - Liste de discussion et d'aide entre utilisateurs de Squalé
 - announce@squale.org
 - Liste de diffusion des principales informations du projet
 - Ex. : nouvelles releases, évènement autour de Squalé, ...
 - Autres listes destinées à l'équipe Squalé
- Une démo live :
 - URL : <https://saas.qualixo.com/demo-squale/>
 - user/user ou squaleadmin/admin
 - Exemple concret de l'offre Squalé as a Service de Qualixo





En synthèse...

qualixo

Ecosystème Squalo

Squalo – projet open-source

<http://www.squale.org>

- Projet communautaire ouvert à tous
 - Utilisateurs
 - Contributeurs et committers
- Infrastructure nécessaire au développement du projet open-source :
 - Bug tracker et Wiki
 - Serveur d'intégration continue
 - Référentiel Maven
 - Mailing-listes
 - ...etc.
- **Licences LGPL v3 et CC**
- **Première version officielle 26-01-2009**



Squalo – projet de recherche

<http://recherche.squale.org>

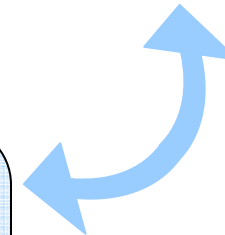
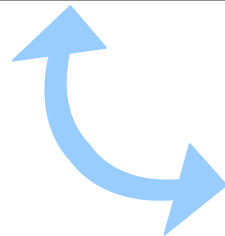
- Dédié aux partenaires du projet de recherche
- Espace de communication nécessaire à l'équipe
 - CR de réunions
 - Documents de gestion
 - Détails des WP
 - Planning
 - Livrables
 - Brainstorming
 - ...
- **Résultats déversés dans le projet open-source lors de leur livraison**



Squalo – produit Qualixo

<http://www.qualixo.com>

- Société offrant des services autour de Squalo :
 - Implantation du logiciel dans une démarche qualimétrique
 - Support et maintenance
 - Offre ASP / SaaS
- Assure l'évolution de Squalo en y implantant :
 - son expertise du domaine de la qualimétrie
 - les besoins des grands comptes



En synthèse...

- Squale, c'est :
 - Des modèles qualimétriques
 - issus de plusieurs années d'expérience chez des grands comptes
 - améliorés/validés par des travaux de recherche
 - Un logiciel « Portail qualimétrie »
 - open-source
 - disponible tout de suite

→ inscrivez-vous à announce@squale.org !
- Les modèles et le logiciel vont évoluer et s'enrichir en fonction des besoins exprimés par les grands comptes
 - Les efforts des uns seront récompensés par les apports des autres
 - Air France et PSA sont déjà entrés dans ce cercle vertueux
- Ambition affichée :
 - Devenir le standard « de facto » dans le domaine de la qualimétrie



Merci de votre attention !

**Des
questions ?**

