

**Introduction à**  
**FACTORY SCHEMES™**  
*Des schémas d'usine logicielle  
adaptables*

**FACTORY SCHEMES™ 3**  
**Standard Edition**  
*Community & Enterprise*  
*Principaux avantages et fonctionnalités*

**GECKO Software**

<http://consulting.byGecko.com>

Email: [Info@gecko.fr](mailto:Info@gecko.fr)

Tél: (33) 04 42 26 06 08



## **Index**

<b>AVANTAGES DE FACTORY SCHEMES™</b> .....	<b>3</b>
<b>FONCTIONNALITES DE FACTORY SCHEMES™</b> .....	<b>4</b>
<b>GESTION DU CHANGEMENT</b> .....	<b>4</b>
<b>CONTROLE DE SOURCE</b> .....	<b>6</b>
<b>INTEGRATION D'ENVIRONNEMENT</b> .....	<b>8</b>
<b>GESTION DE DEPOT</b> .....	<b>10</b>
<b>ADMINISTRATION DE PLATEFORME</b> .....	<b>12</b>
<b>ANALYSE DE LA QUALITE</b> .....	<b>14</b>
<b>BACKBONE DU CYCLE DE VIE</b> .....	<b>16</b>
<b>INTEGRATION EN CONTINU</b> .....	<b>18</b>

## Avantages de **FACTORY SCHEMES™**

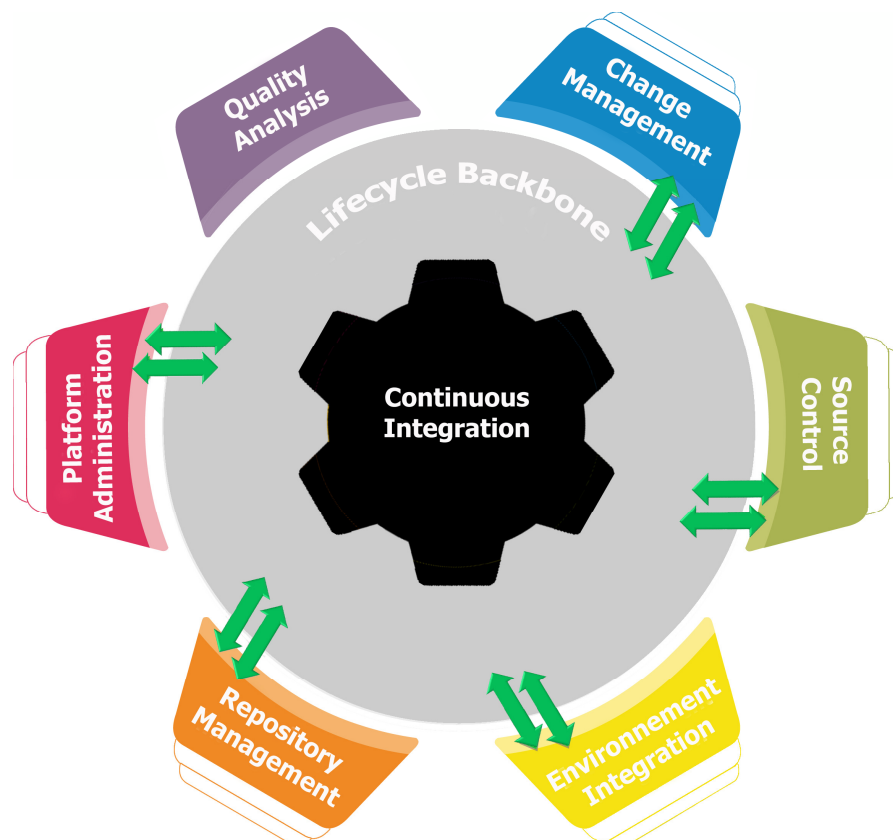
Les technologies ainsi que les « standards » de développement des applications sont toujours plus nombreux et font l'objet de changements de plus en plus fréquents. Si la prise en compte de ces changements dans la **construction des applications** est réalisée de manière manuelle l'augmentation des coûts s'avère alors très vite exponentielle. Ceci est vrai même si les référentiels de code, les plateformes de développement, les environnements d'exécution ou les besoins d'intégration demeurent identiques. Il devient donc nécessaire **d'industrialiser** le processus de fabrication des applications. Il est aussi question « **d'Usine logicielle** ».



**FACTORY SCHEMES™** fournit littéralement des plans « d'Usine Logicielle » prenant en compte les multiples dimensions que sont les acteurs, les processus et les outils mobilisés dans la fabrication d'applications.

Ces « **FACTORY SCHEMES™** » sont personnalisables au vu des sensibilités et des habitudes de l'entreprise.

**FACTORY SCHEMES™** tire parti des avantages retirés lors d'expériences de mises en œuvre similaires en environnement Grands Comptes Banques/Assurance/Services.



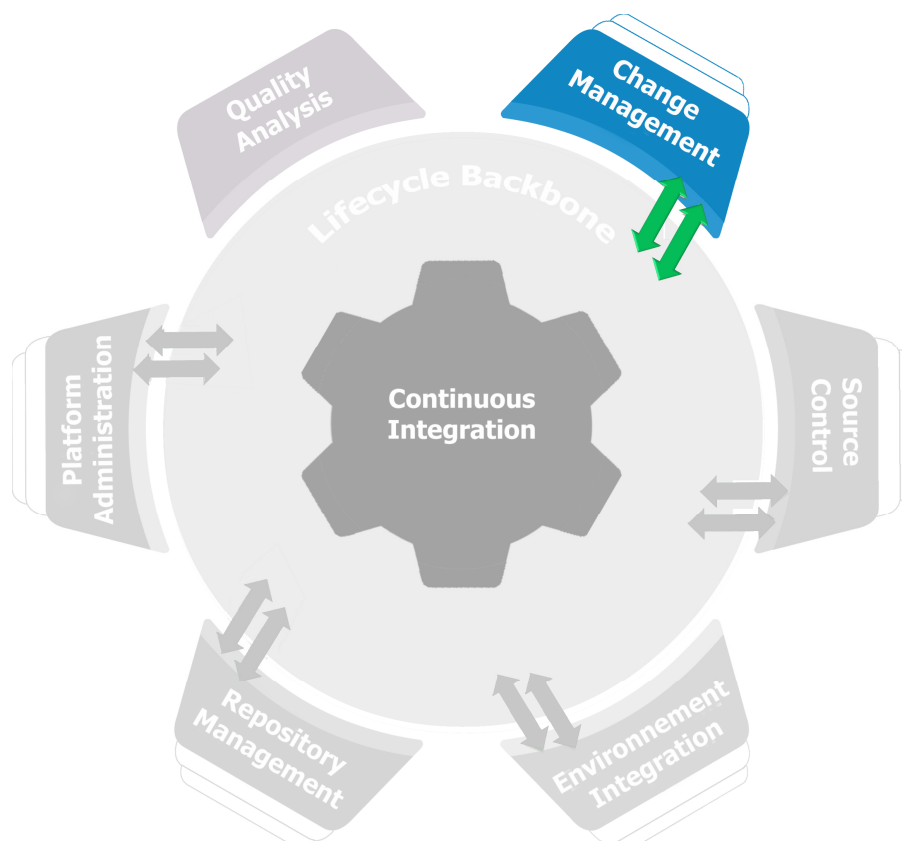
## Fonctionnalités de FACTORY SCHEMES™

**FACTORY SCHEMES™** agrège les meilleures pratiques de l'industrie issues de plateformes d'intégration continue concrètes réunissant tous les concepts d'une « Usine logicielle ».

Prenez connaissance ci-après des fonctionnalités clés de **FACTORY SCHEMES™** (**bientôt disponibles en version OpenSource**, **déjà disponibles en version OpenSource**, **et déjà disponibles en version entreprise**), en partant des sujets les plus globaux pour parvenir aux points précis de la *qualité interne du code*, du procédé de *construction en parallèle*, du mécanisme de *maintien des dépendances* applicatives etc... Il vous sera ainsi possible d'analyser notre solution au vu des problématiques qui vous sont propres.

### Gestion du Changement

*Sujet Global*

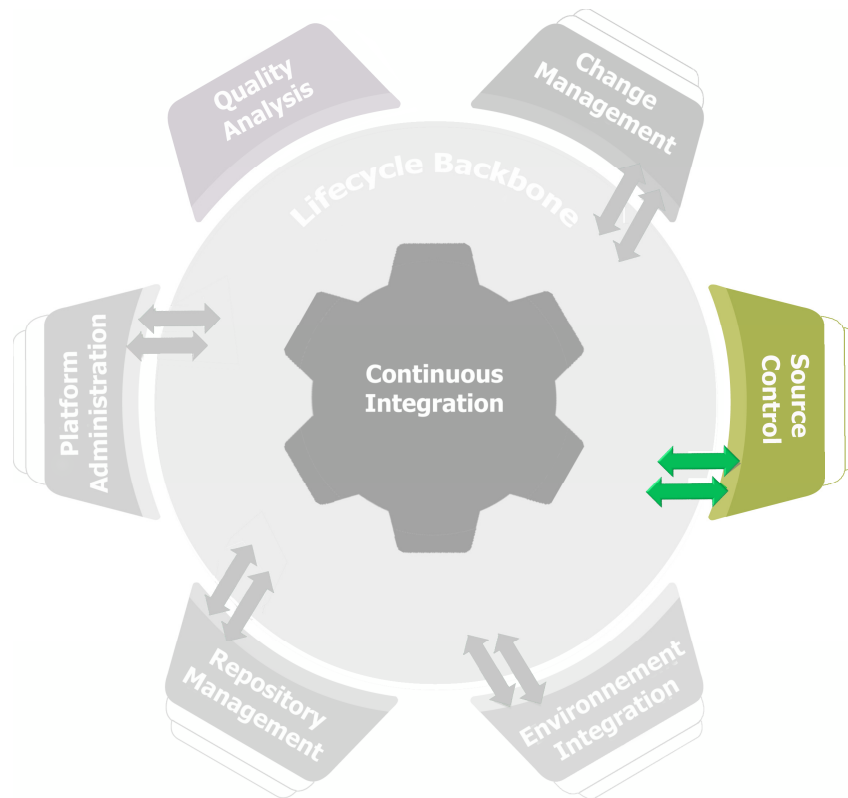


*Fonctionnalités Supportées*

<p><b>Respect des standards</b></p>	<ul style="list-style-type: none"> <li>- Génération de la documentation d'un projet par défaut supportée par le réacteur Maven et le système de suivi (JIRA, Trac, GitHub...) :             <ul style="list-style-type: none"> <li>o Issue Management</li> <li>o Release Announcement</li> <li>o Release Notes</li> <li>o Changes Reports...</li> </ul> </li> </ul>	
<p><b>Conduite des changements</b></p>	<ul style="list-style-type: none"> <li>- Normalisation du processus de développement autour de types de changements prédéfinis pour améliorer la visibilité entre les équipes et les projets :             <ul style="list-style-type: none"> <li>o changes</li> <li>o issues</li> <li>o improvements</li> <li>o features</li> <li>o feedbacks</li> <li>o incidents...</li> </ul> </li> </ul>	
<p><b>Cadrage des tickets</b></p>	<ul style="list-style-type: none"> <li>- Modélisation du processus de développement dans un workflow de suivi de ticket qui, pour s'assurer de prendre les bonnes mesures au bon moment, peut être configuré à partir d'un existant ou entièrement redéfini avec :             <ul style="list-style-type: none"> <li>o des champs personnalisés</li> <li>o des notifications par email</li> <li>o des contrôles d'accès</li> <li>o des autorisations...</li> </ul> </li> </ul>	
<p><b>Pilotage des ressources</b></p>	<ul style="list-style-type: none"> <li>- Définition d'une feuille de route (roadmap) organisant chaque jalon (milestone) afin de montrer la voie à suivre</li> <li>- Planification des ressources (teams) à travers les mêmes éléments de travail (work items) définis précédemment</li> </ul>	
<p><b>Suivi des avancements</b></p>	<ul style="list-style-type: none"> <li>- Intégration de chaque nouvel événement dans le processus de développement afin de collaborer, de rester à jour avec l'activité des équipes et de produire une vue d'ensemble (timeline) du projet et le suivi d'avancement</li> </ul>	

## Contrôle de Source

*Sujet Global*



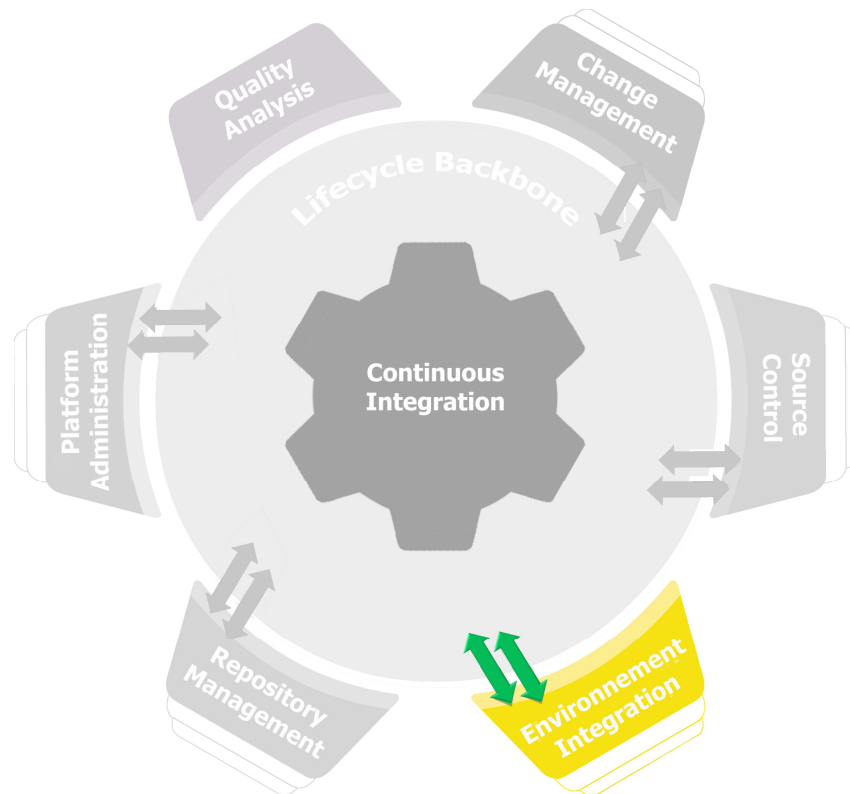
### *Fonctionnalités Supportées*

<b>Respect des standards</b>	<ul style="list-style-type: none"><li>- Gestion de version par défaut supportée par le réacteur Maven en tant que partie intégrante du projet et fournissant un mécanisme standard pour le versionnement des changements de configuration logicielle (également appelé contrôle de source) :<ul style="list-style-type: none"><li>o SubVersion</li><li>o CMSynergy</li><li>o ClearCase</li><li>o TFS</li><li>o GIT...</li></ul></li></ul>
------------------------------	---

<p><b>Indépendance des commandes</b></p>	<ul style="list-style-type: none"> <li>- Intégration indépendante de toute implémentation spécifique, des commandes courantes offrant les fonctionnalités de base pour gérer chacune des révisions de ressource :             <ul style="list-style-type: none"> <li>○ add</li> <li>○ changelog</li> <li>○ checkin</li> <li>○ checkout</li> <li>○ update...</li> </ul> </li> <li>- Intégration indépendante de toute implémentation spécifique, des commandes étendues offrant les fonctionnalités de base pour intégrer chacune de ces révisions :             <ul style="list-style-type: none"> <li>○ diff</li> <li>○ merge</li> <li>○ conflict...</li> </ul> </li> </ul>	
<p><b>Modularité des fonctionnalités</b></p>	<ul style="list-style-type: none"> <li>- Délégation des fonctionnalités via la mise en œuvre d'implémentations spécifiques</li> <li>- Extension spécifique de fonctions avancées fournissant un support étendu de certains usages de versionnement de projet :             <ul style="list-style-type: none"> <li>○ release</li> <li>○ bootstrap</li> <li>○ branch</li> <li>○ tag</li> <li>○ snapshot...</li> </ul> </li> </ul>	

## Intégration d'Environnement

*Sujet Global*



### Fonctionnalités Supportées

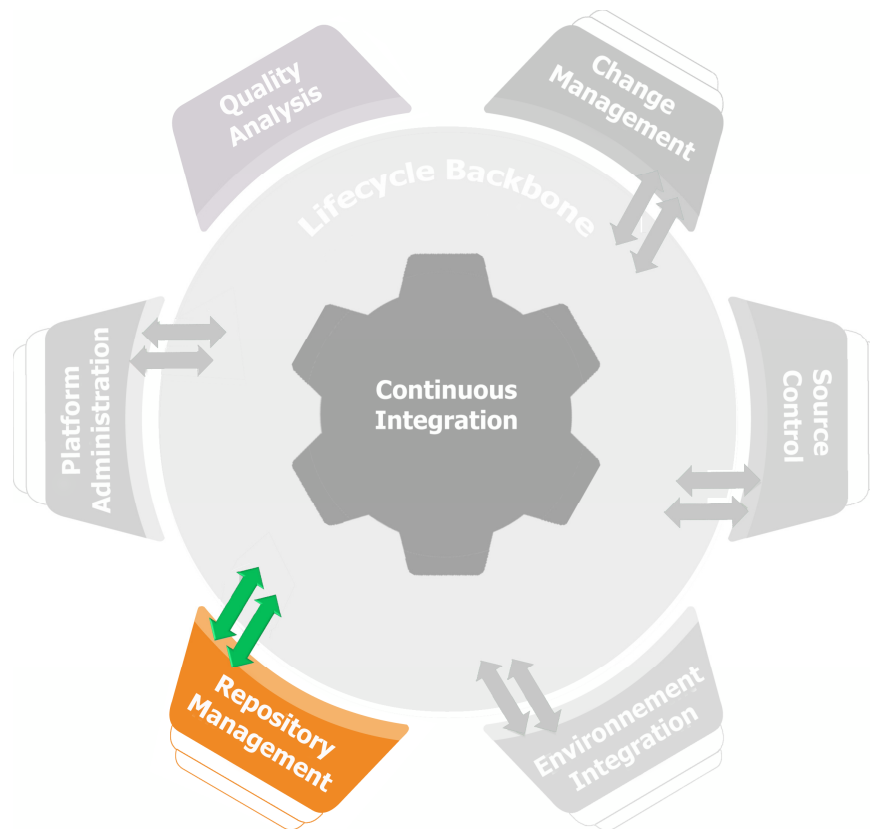
<p><b>Respect des standards</b></p>	<ul style="list-style-type: none"> <li>- Transformation de méta-modèles génériques pour un environnement d'exécution spécifique :             <ul style="list-style-type: none"> <li>o OMG</li> <li>o MDA</li> <li>o OSGi</li> <li>o JPA...</li> </ul> </li> <li>- Extraction de méta-données génératives pour un environnement d'exécution au sein d'une plateforme d'intégration spécifique :             <ul style="list-style-type: none"> <li>o OS</li> <li>o SGBD</li> <li>o JEE</li> <li>o NET...</li> </ul> </li> <li>- Génération de méta-données descriptives (files, folders, classpath, settings...) pour un environnement de travail spécifique :             <ul style="list-style-type: none"> <li>o IDE</li> <li>o PDE</li> <li>o RAD</li> <li>o WAS...</li> </ul> </li> </ul>	
-------------------------------------	--	--



<b>Cadrage des archétypes</b>	<ul style="list-style-type: none"><li>- Urbanisation des projets à travers une conception adaptée d'archétypes de projets, appliquant de façon cohérente un héritage et une agrégation à chaque type d'artefact</li></ul>	
<b>Portabilité des artéfacts</b>	<ul style="list-style-type: none"><li>- Normalisation de la portabilité (sous certaines conditions) de la configuration d'un ensemble de dépendances (potentiellement différentes) afin d'ajuster l'artefact du projet (par exemple, chemins, utilisateurs et profils locaux)</li></ul>	
<b>Cadrage des environnements</b>	<ul style="list-style-type: none"><li>- Généralisation du concept de profil de construction donnant des paramètres équivalents mais différents pour un ensemble d'environnements cibles (par exemple, développement, test et production)</li></ul>	

## Gestion de Dépôt

*Sujet Global*



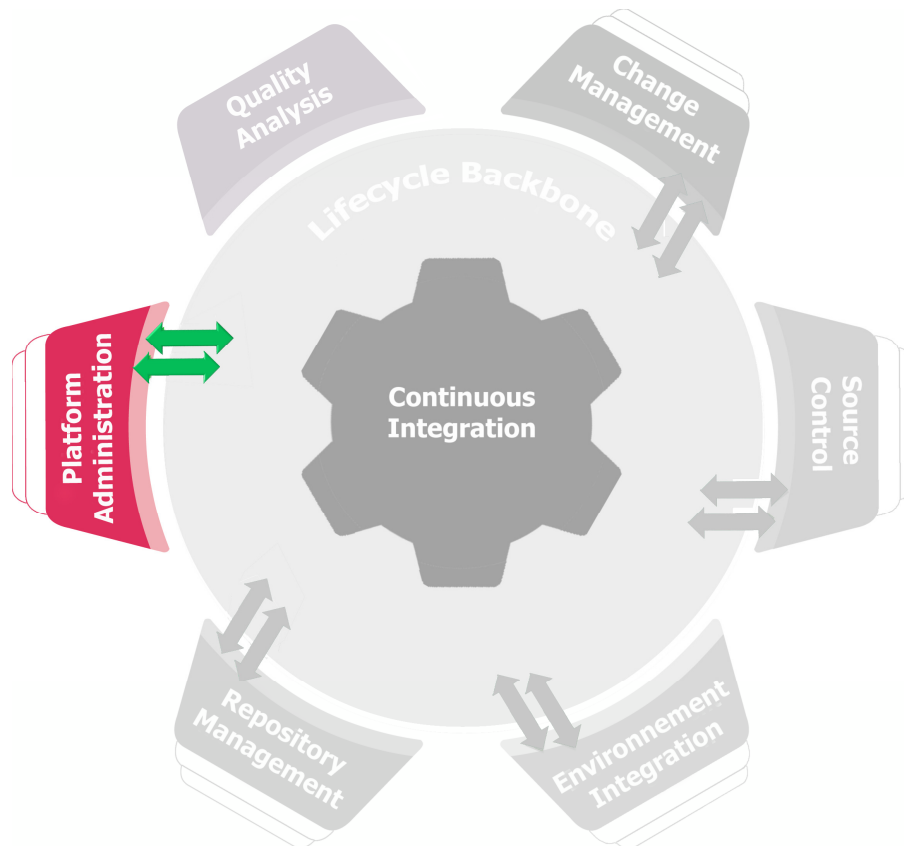
*Fonctionnalités Supportées*

<p><b>Respect des standards</b></p>	<ul style="list-style-type: none"> <li>- Organisation des dépôts d'artefacts par défaut supportée par le réacteur Maven et le gestionnaire de dépôt :             <ul style="list-style-type: none"> <li>o Archiva</li> <li>o Artifactory</li> <li>o Nexus...</li> </ul> </li> </ul>	
<p><b>Stabilité des téléchargements</b></p>	<ul style="list-style-type: none"> <li>- Notion de dépôt distant, mandaté pour refléter les téléchargements d'artefacts, afin d'assurer la stabilité au sein d'une organisation</li> </ul>	
<p><b>Efficacité des développements</b></p>	<ul style="list-style-type: none"> <li>- Notion de référentiel local, hébergé pour gérer les téléchargements d'artefacts, afin de favoriser l'efficacité et la collaboration du cycle de vie de développement</li> </ul>	

<b>Sécurité des déploiements</b>	- Notion de référentiel de déploiement, mis en scène pour promouvoir les téléchargements d'artefacts, afin d'accompagner les décisions avant d'aller en production	
<b>Uniformité des environnements</b>	- Notion de référentiel central, fusionné pour consolider les téléchargements d'artefacts afin de pointer sur un groupe de stockage unique	

## Administration de Plateforme

*Sujet Global*



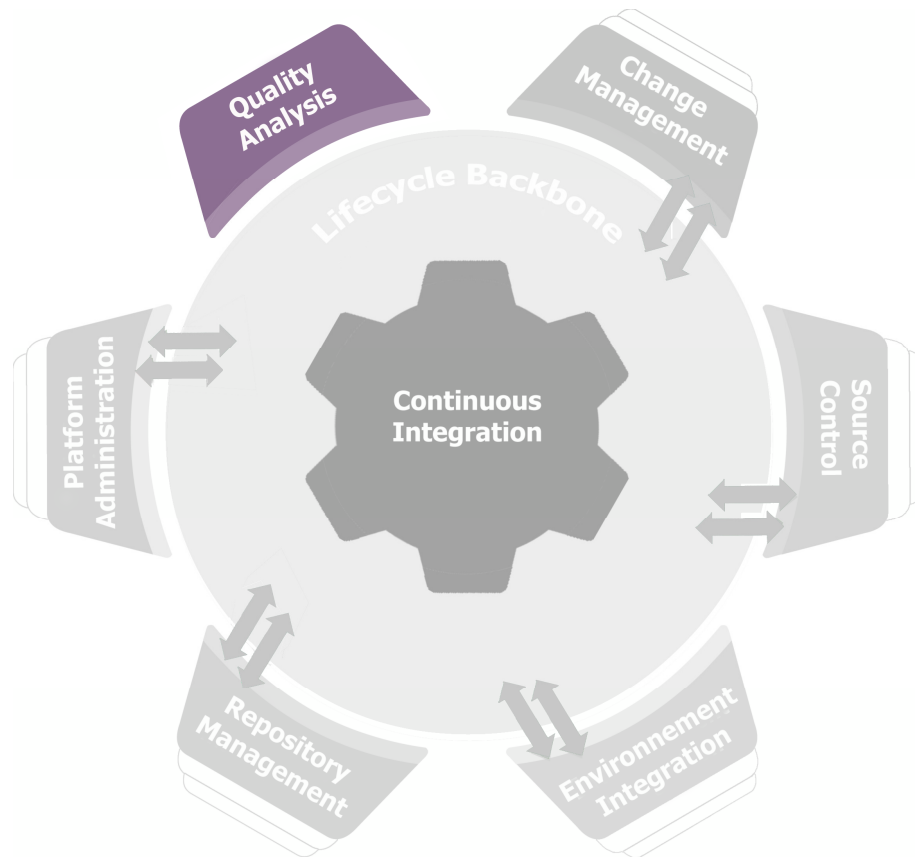
*Fonctionnalités Supportées*

<p><b>Respect des standards</b></p>	<ul style="list-style-type: none"> <li>- Administration de plateformes par défaut supportées par le réacteur Maven fournissant une interface standard pour l'installation, la configuration, le déploiement :                             <ul style="list-style-type: none"> <li>o Weblogic</li> <li>o Websphere</li> <li>o Tomcat...</li> </ul> </li> </ul>	
<p><b>Administrabilité des environnements</b></p>	<ul style="list-style-type: none"> <li>- Manipulation de l'environnement de façon standard en supportant les tâches d'administration complexes telles que :                             <ul style="list-style-type: none"> <li>o redémarrage d'un conteneur</li> <li>o l'exécution d'un plan de déploiement</li> <li>o l'installation d'un plugin...</li> </ul> </li> </ul>	

<p><b>Portabilité des variables</b></p>	<ul style="list-style-type: none"> <li>- Limitation de la portabilité sur une plateforme cible, au sein même de la solution native, par à une combinaison de variables connues telles que :             <ul style="list-style-type: none"> <li>○ le système d'exploitation</li> <li>○ le serveur web</li> <li>○ le conteneur d'application</li> <li>○ la grappe de serveurs</li> <li>○ l'équilibrage de charge</li> <li>○ le système de base de données...</li> </ul> </li> </ul>	
<p><b>Maintenabilité des binaires</b></p>	<ul style="list-style-type: none"> <li>- Distribution d'un livrable basé sur une plateforme spécifique cible et assurant, dans une certaine mesure, la maintenabilité de la solution binaire :             <ul style="list-style-type: none"> <li>○ centralization</li> <li>○ externalization</li> <li>○ variabilization</li> <li>○ versioning</li> <li>○ populating...</li> </ul> </li> </ul>	

## Analyse de la Qualité

*Sujet Global*



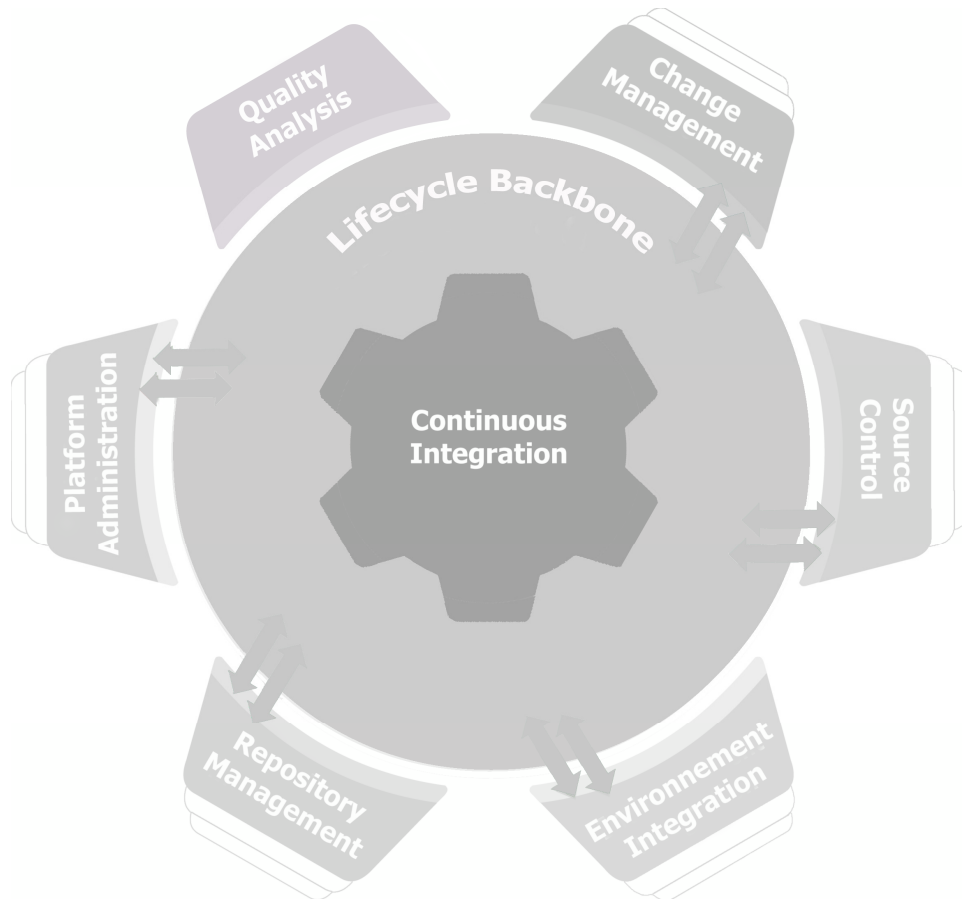
*Fonctionnalités Supportées*

<b>Respect des standards</b>	<ul style="list-style-type: none"><li>- Agrégation du reporting qualité d'un projet par défaut supportée par le réacteur Maven et le système d'analyse (Hudson, Dashboard, Sonar, Squale, XRadar...) :<ul style="list-style-type: none"><li>o CheckStyle</li><li>o PMD</li><li>o JDepend</li><li>o FindBugs...</li></ul></li></ul>
------------------------------	--

<p><b>Conformité des indicateurs</b></p>	<ul style="list-style-type: none"> <li>- Instrumentation automatique des indicateurs de chaque modèle de conception (method, class, package, module, language, architecture...) et des métriques d'acceptance d'un projet :             <ul style="list-style-type: none"> <li>o complexity</li> <li>o coupling</li> <li>o cohesion</li> <li>o cycle...</li> </ul> </li> <li>- Évaluation automatique de la dette technique et du coût de remédiation des non-conformités (clean, refactoring, rewriting...) du patrimoine applicatif de l'entreprise :             <ul style="list-style-type: none"> <li>o portability, maintainability, security, efficiency...</li> <li>o bloquant, critique, majeur, mineur...</li> </ul> </li> </ul>	
<p><b>Conformité des exigences</b></p>	<ul style="list-style-type: none"> <li>- Introduction automatique d'une revue de code statique (duplication, violation, documentation, convention...) afin de conformer l'application à un ensemble d'exigences techniques prédéfinies (approche dite boîte blanche)</li> <li>- Introspection de la façon dont l'application a été construite (à savoir sa qualité interne) réduisant le risque d'une application plus difficile à maintenir au fil du temps</li> </ul>	
<p><b>Conformité des spécifications</b></p>	<ul style="list-style-type: none"> <li>- Exécution automatique d'une couverture de test dynamique (unitaire, intégration, régression, qualification...) afin de s'assurer que l'application est conforme aux spécifications fonctionnelles (approche dite boîte noire)</li> <li>- Interaction avec l'application et observation de son comportement (à savoir sa qualité externe) réduisant le risque de régression au fil du temps</li> </ul>	

## Backbone du Cycle de Vie

*Sujet Global*



*Fonctionnalités Supportées*

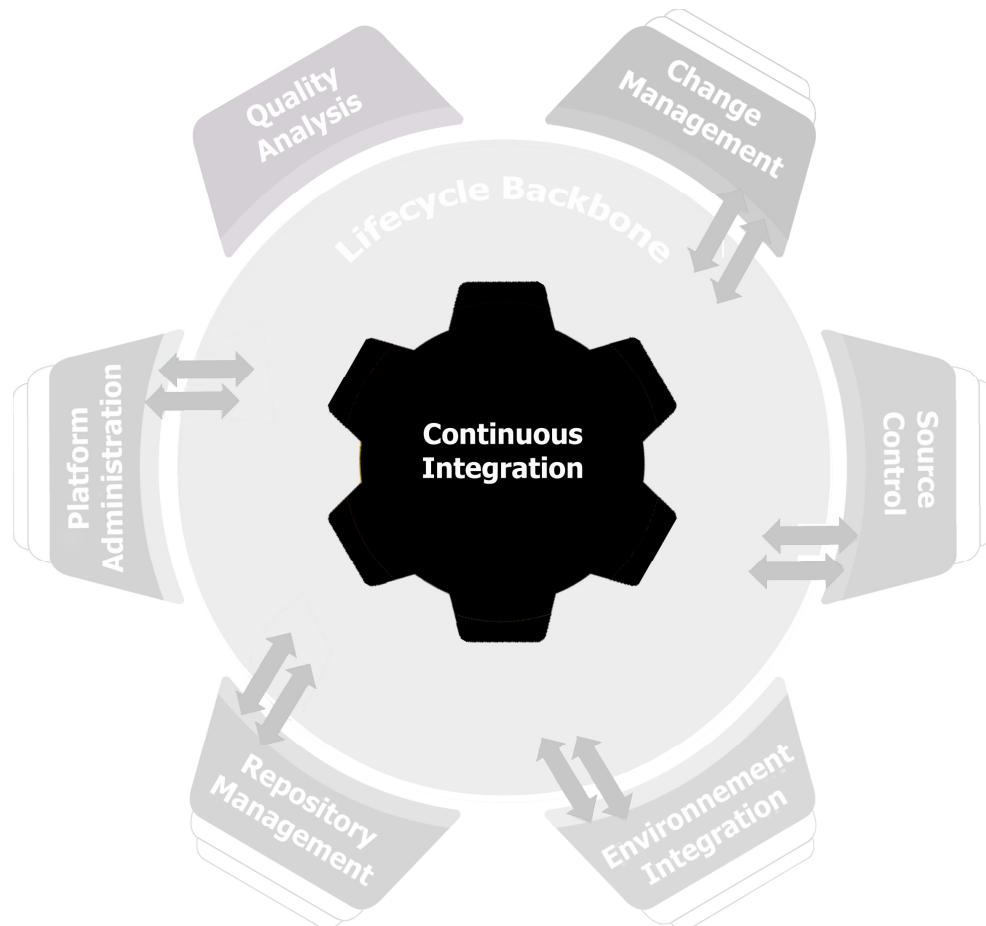
<p><b>Respect des standards</b></p>	<ul style="list-style-type: none"> <li>- Modélisation du processus de construction et de distribution d'un projet (ou « artefact ») en accord avec <u>tous</u> les concepts Maven :             <ul style="list-style-type: none"> <li>○ Build Lifecycle</li> <li>○ Convention Over Configuration</li> <li>○ Universal Reuse</li> <li>○ POM, MOJO...</li> </ul> </li> </ul>	
<p><b>Découpage en étapes (stages)</b></p>	<ul style="list-style-type: none"> <li>- Normalisation de la logique de construction autour des étapes prédéfinies du cycle de vie Maven et par défaut supportées par le noyau de construction Maven (ou « reactor ») :             <ul style="list-style-type: none"> <li>○ clean</li> <li>○ compile</li> <li>○ package</li> <li>○ install</li> <li>○ deploy</li> <li>○ site...</li> </ul> </li> </ul>	



<p><b>Indépendance des phases</b></p>	<ul style="list-style-type: none"> <li>- Définition de séquences de phases de construction, dans laquelle chaque phase est responsable d'une étape spécifique dans le cycle de vie</li> </ul>	
<p><b>Autonomie des objectifs (goals)</b></p>	<ul style="list-style-type: none"> <li>- Factorisation des responsabilités du cycle de vie au travers d'objectifs (dits « goals ») liés à chacune des phases de construction</li> <li>- Isolation de ces objectifs de construction pouvant varier en fonction de chacune des phases</li> </ul>	
<p><b>Modularité des capacités (capabilities)</b></p>	<ul style="list-style-type: none"> <li>- Délégation des capacités du cycle de vie via l'intégration et/ou l'implémentation de plugins (ou « artefact ») spécifiques</li> </ul>	
<p><b>Découplage des dépendances</b></p>	<ul style="list-style-type: none"> <li>- Externalisation du maintien, du contrôle et de la stabilité des dépendances de la logique de construction :             <ul style="list-style-type: none"> <li>o override</li> <li>o transitivity</li> <li>o cycle</li> <li>o scope...</li> </ul> </li> </ul>	
<p><b>Cadrage des normes</b></p>	<ul style="list-style-type: none"> <li>- Urbanisation transverse du cycle de vie par branches d'héritages (disposed, aggregated, managed, combined...) reflétant le modèle d'entreprise (hierarchical metadata organization policies) :             <ul style="list-style-type: none"> <li>o company</li> <li>o business unit</li> <li>o project</li> <li>o team...</li> </ul> </li> </ul>	

## Intégration en Continu

*Sujet Global*



*Fonctionnalités Supportées*

<p><b>Respect des standards</b></p>	<ul style="list-style-type: none"> <li>- Intégration en continu au sens Maven réalisée à travers des notifications d'état de construction dans le cadre d'une exigence de traçabilité et/ou une exigence de responsabilité immédiate :             <ul style="list-style-type: none"> <li>o notification en cas de succès, d'avertissement...</li> <li>o notification en cas d'erreur, d'échec...</li> </ul> </li> </ul>	
<p><b>Evolutivité des environnements</b></p>	<ul style="list-style-type: none"> <li>- Consolidation d'une version suivant un modèle de propagation dans les environnements cibles :             <ul style="list-style-type: none"> <li>o développement</li> <li>o intégration</li> <li>o qualification</li> <li>o production...</li> </ul> </li> </ul>	

<p><b>Disponibilité des modifications</b></p>	<ul style="list-style-type: none"> <li>- Propagation des modifications résultant d'une construction automatique complète :             <ul style="list-style-type: none"> <li>○ quotidienne</li> <li>○ nocturne</li> <li>○ hebdomadaire...</li> </ul> </li> </ul>	
<p><b>Effcience des constructions</b></p>	<ul style="list-style-type: none"> <li>- Construction automatique parallèle pour satisfaire une exigence de productivité en fonction de la structure des projets et de la compatibilité des écosystèmes</li> </ul>	
<p><b>Rendement des développements</b></p>	<ul style="list-style-type: none"> <li>- Construction automatique incrémentielle pour satisfaire une exigence d'agilité en fonction de la structure des projets et de la compatibilité des écosystèmes</li> </ul>	