

# Gestion du Cycle de vie des Applications (ALM)

Le meilleur des solutions open source

**Smile**  
OPEN SOURCE SOLUTIONS

[www.smile.fr](http://www.smile.fr) • +33 (0)1 41 40 11 00 • [contact@smile.fr](mailto:contact@smile.fr)  
[www.smile-oss.com](http://www.smile-oss.com) • [blog.smile.fr](http://blog.smile.fr) • twitter: @GroupeSmile



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## PREAMBULE

### SMILE

Smile est une **société d’ingénieurs experts** dans la mise en œuvre de **solutions open source** et l’intégration de systèmes appuyés sur l’open source. Smile est membre de l’**APRIL**, l’association pour la promotion et la défense du logiciel libre, du **PLOSS** – le réseau des entreprises du Logiciel Libre en Ile-de-France et du **CNLL** – le conseil national du logiciel libre.

**Smile compte près de 600 collaborateurs dans le monde**, dont plus de 500 en France (décembre 2011), ce qui en fait *le premier intégrateur français et européen de solutions open source*.

Depuis 2000, environ, **Smile mène une action active de veille technologique** qui lui permet de découvrir les produits les plus prometteurs de l’open source, de les qualifier et de les évaluer, de manière à proposer à ses clients les produits les plus aboutis, les plus robustes et les plus pérennes.

Cette démarche a donné lieu à **toute une gamme de livres blancs** couvrant différents domaines d’application. La gestion de contenus (2004), les portails (2005), la business intelligence (2006), la virtualisation (2007), la gestion électronique de documents (2008), les PGI/ERPs (2008), les VPN open source (2009), les Firewall et Contrôle de flux (2009), les Middleware orientés messages (2009), l’e-commerce et les Réseaux Sociaux d’Entreprise (2010) et plus récemment, le Guide de l’open source, NoSQL (2011). Chacun de **ces ouvrages présente une sélection des meilleures solutions open source** dans le domaine considéré, leurs qualités respectives, ainsi que des retours d’expérience opérationnels.

Au fur et à mesure que des solutions open source solides gagnent de nouveaux domaines, Smile sera présent pour proposer à ses clients d’en bénéficier sans risque. Smile apparaît dans le paysage informatique français comme **le prestataire intégrateur de choix** pour **accompagner** les plus grandes entreprises dans l’adoption des meilleures solutions open source.

Ces dernières années, Smile a également étendu la gamme des services proposés. Depuis 2005, un département consulting accompagne nos clients, tant dans les phases d’avant-projet, en recherche de solutions, qu’en accompagnement de projet. Depuis 2000, Smile dispose d’un studio graphique, devenu en 2007 Smile Digital – agence interactive, proposant outre la création graphique, une expertise e-marketing, éditoriale, et interfaces riches. Smile dispose aussi d’une agence spécialisée dans la TMA (support et exploitation des applications) et d’un centre de formation complet, Smile Training. **Enfin, Smile est implanté à Paris, Lille, Lyon, Grenoble, Nantes, Bordeaux, Marseille et Montpellier ; et présent également en Espagne, en Suisse, au Benelux, en Ukraine et au Maroc.**

## QUELQUES REFERENCES DE SMILE

SMILE est fier d’avoir contribué, au fil des années, aux plus grandes réalisations Web françaises et européennes. Vous trouverez ci-dessous quelques clients nous ayant adressé leur confiance.

### Sites Internet

EMI Music, Salon de l’Agriculture, Mazars, Areva, Société Générale, Gîtes de France, Patrice Pichet, Groupama, Eco-Emballage, CFnews, CEA, Prisma Pub, Véolia, NRJ, JCDecaux, 01 Informatique, Spie, PSA, Boiron, Larousse, Dassault Systèmes, Action Contre la Faim, BNP Paribas, Air Pays de Loire, Forum des Images, IFP, BHV, ZeMedical, Gallimard, Cheval Mag, Afssaps, Beneteau, Carrefour, AG2R La Mondiale, Groupe Bayard, Association de la Prévention Routière, Secours Catholique, Canson, Veolia, Bouygues Telecom, CNIL...

### Portails, Intranets et Systèmes d’Information

HEC, Bouygues Telecom, Prisma, Veolia, Arjowiggins, INA, Primagaz, Croix Rouge, Eurosport, Invivo, Faceo, Château de Versailles, Eurosport, Ipsos, VSC Technologies, Sanef, Explorimmo, Bureau Veritas, Région Centre, Dassault Systèmes, Fondation d’Auteuil, INRA, Gaz Electricité de Grenoble, Ville de Niort, Ministère de la Culture, PagesJaunes Annonces...

### E-Commerce

Krys, La Halle, Gibert Joseph, De Dietrich, Adenclassifieds, Macif, Furet du Nord, Gîtes de France, Camif Collectivité, GPdis, Projectif, ETS, Bain & Spa, Yves Rocher, Bouygues Immobilier, Nestlé, Stanhome, AVF Périmédical, CCI, Pompiers de France, Commissariat à l’Energie Atomique, Snowleader, Darjeeling...

### ERP et Décisionnel

Veolia, La Poste, Christian Louboutin, Eveha, Sun’R, Home Ciné Solutions, Pub Audit, Effia, France 24, Publicis, iCasque, Nomadvantage, Gets, Nouvelles Frontières, Anevia, Jus de Fruits de Mooréa, Espace Loggia, Bureau Veritas, Skyrock, Lafarge, Cadremploi, Meilleurmobilite.com, Groupe Vinci, IEDOM (Banque de France), Carrefour, Jardiland, Trésorerie Générale du Maroc, Ville de Genève, ESCP, Sofia, Faiveley Transport, INRA, Deloitte, Yves Rocher, ETS, DGAC, Generalitat de Catalunya, Gilbert Joseph, Perouse Médical...

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## Gestion documentaire

Primagaz, UCFF, Apave, Géoservices, Renault F1 Team, INRIA, CIDJ, SNCD, Ecureuil Gestion, CS informatique, Serimax, Véolia Propreté, NetasQ, Corep, Packetis, Alstom Power Services, Mazars...

## Infrastructure et Hébergement

Agence Nationale pour les Chèques Vacances, Pierre Audoin Consultants, Rexel, Motor Presse, OSEO, Sport24, Eco-Emballage, Institut Mutualiste Montsouris, ETS, Ionis, Osmoz, SIDEL, Atel Hotels, Cadremploi, SETRAG, Institut Français du Pétrole, Mutualité Française...

Consultez nos références, en ligne, à l’adresse : <http://www.smile.fr/clients>.

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## CE LIVRE BLANC

Assez bien implantée dans les pays anglo-saxons, la gestion du cycle de vie des applications (ALM) et la valeur ajoutée qu’elle apporte sont encore mal connus des entreprises en France.

Une prise de conscience a commencé à émerger lorsque l’on a constaté les échecs récurrents des projets informatiques. Dès 1994, l’étude Chaos Report du Standish Group montrait que le pilotage des projets logiciels laissait à désirer et on enregistrait plus d’échecs que de succès : **à peine 16 % des projets logiciels pouvaient être considérés comme réussis** dans la mesure où ils étaient livrés dans les délais, sans dépassement de budget et répondant globalement aux exigences utilisateurs.

En 2006, la même étude illustrait que la situation était un peu moins catastrophique, puisque **le taux de succès s’élevait à 35 %**. Les raisons qui pouvaient expliquer ces échecs étaient pourtant connues : besoins mal exprimés, spécifications incomplètes ou inadaptées, manque de communication, éléments de travail du projet sans liens, sans traçabilité, manque de réactivité, outils disparates, mal intégrés ou trop chers, méthodes en inadéquation avec les outils et les hommes...

En 2008, selon une enquête internationale du Cabinet Forrester, **près de la moitié des entreprises utilisent déjà des processus ALM et ont pris conscience de leur nécessité**, et près de l’autre moitié n’a pas encore perçu l’enjeu qu’elle représente. Les entreprises des pays anglo-saxons et d’Allemagne se situent principalement dans la première moitié, mais la France a un retard important sur le sujet : 26 % des entreprises ne sont même pas au courant de l’existence de cette technologie. Pourtant, on note que la France est friande des nouvelles technologies (Ajax, Web 2.0...) et les utilise rapidement, plus vite même que les Etats-Unis. Par contre les USA accordent plus d’importance à l’optimisation de leurs processus et aux outils y contribuant.

Le marché de l’ALM est donc un marché jeune, en pleine croissance. Aujourd’hui, **l’enjeu est toujours le même et il devient de plus en plus incontournable : industrialiser les développements, c’est-à-dire de gérer le cycle de vie des applications**. C’est tout l’objet de ce livre blanc. Attardons-nous sur cette notion et essayons de mieux la cerner.

N’hésitez pas à nous transmettre vos avis et évaluations sur les produits présentés dans ce livre blanc. Une seule adresse : [contact@smile.fr](mailto:contact@smile.fr)



## SOMMAIRE

<b>PREAMBULE.....</b>	<b>2</b>
SMILE .....	2
QUELQUES REFERENCES DE SMILE.....	3
CE LIVRE BLANC.....	5
SOMMAIRE .....	6
<b>MIEUX COMPRENDRE L’ALM .....</b>	<b>8</b>
QU’EST-CE QUE LA GESTION DU CYCLE DE VIE DES APPLICATIONS ? .....	8
LES GRANDS CONCEPTS.....	10
GESTION DES EXIGENCES.....	10
GESTION DES CHANGEMENTS.....	12
GESTION DE VERSION .....	14
GESTION DE PROJET .....	16
GESTION DE LA QUALITE : TESTS ET INTEGRATION CONTINUE .....	19
GESTION DES LIVRABLES.....	20
GESTION DOCUMENTAIRE .....	21
COLLABORATION SOCIALE .....	22
<b>CE QUE L’ALM VA VOUS APPORTER .....</b>	<b>24</b>
POUR QUI ? POUR QUOI FAIRE? .....	24
FAIRE COLLABORER TOUS LES SERVICES DE L’ENTREPRISE A L’INNOVATION.....	27
CREER UNE CHAINE CENTRALISEE D’INGENIERIE LOGICIELLE ET DE GESTION DE PROJET.....	29
APPLIQUER LES PROCESSUS D’ENTREPRISE ET OBTENIR LA TRAÇABILITE .....	32
INDUSTRIALISER LE PROCESSUS DE DEVELOPPEMENT .....	35
<b>POURQUOI CHOISIR UNE SOLUTION OPEN-SOURCE POUR VOTRE ALM ? .</b>	<b>37</b>
4 BONNES RAISONS.....	37
INNOVATION .....	37
FLEXIBILITE-EXTENSIBILITE.....	37
INTEROPERABILITE .....	37
LIBERTE ET INDEPENDANCE.....	38
MISE EN GARDE .....	38
LOGICIELS LIBRES VS LOGICIELS OUVERTS, SACHEZ FAIRE LA DIFFERENCE !.....	38
QU’EST-CE QU’UN LOGICIEL OUVERT ? .....	38
QU’EST-CE QU’UN LOGICIEL LIBRE ? .....	39
LES AVANTAGES D’UN LOGICIEL LIBRE POUR L’ENTREPRISE? .....	39
ET L’OPEN-SOURCE DANS TOUT ÇA ? .....	40

<b>LES COMPOSANTS DE L’ALM .....</b>	<b>41</b>
DEVELOPPEMENT .....	41
ECLIPSE .....	41
GESTION DE VERSION .....	41
SUBVERSION .....	41
GIT .....	42
AUTRES .....	42
GESTION DES CHANGEMENTS/TRACKING .....	43
MANTISBT .....	43
AUTRES .....	43
OUTILS DE BUILDS .....	44
ANT .....	44
PHING .....	44
MAVEN .....	45
OUTILS DE TESTS/INTEGRATION CONTINUE .....	45
CONTINUUM .....	45
JENKINS / HUDSON .....	46
AUTRES .....	46
<b>LES SOLUTIONS ALM INTEGREES OPEN SOURCE .....</b>	<b>47</b>
3 SOLUTIONS ALM INTEGREES OPEN-SOURCE .....	47
FUSIONFORGE .....	48
REDMINE .....	52
TULEAP .....	56
SYNTHESE DES COUVERTURES .....	65
<b>ANALYSE DES 3 SOLUTIONS INTEGREES .....</b>	<b>66</b>
CRITERES D’EVALUATION .....	66
SYNTHESE ET RESULTATS .....	68
<b>SYNTHESE .....</b>	<b>71</b>

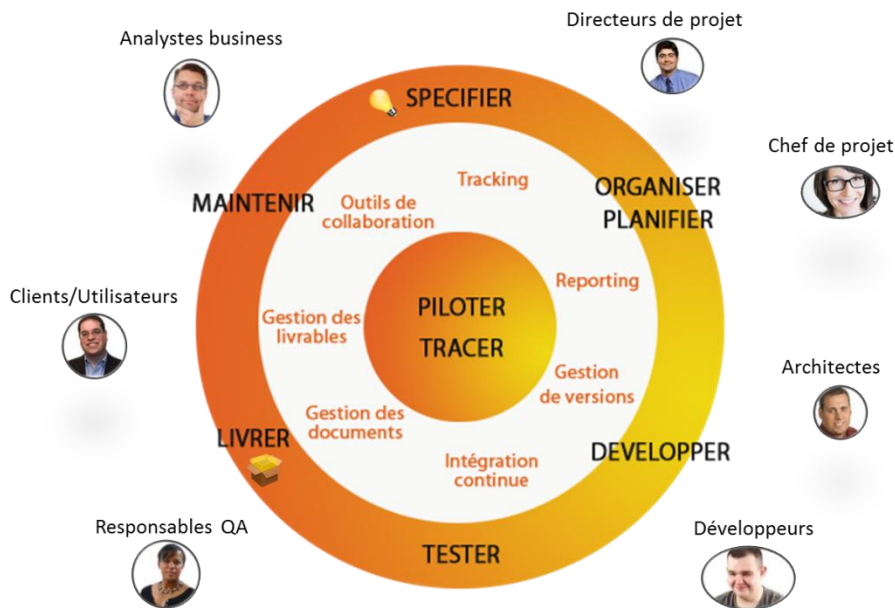
# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

## MIEUX COMPRENDRE L'ALM

### QU'EST-CE QUE LA GESTION DU CYCLE DE VIE DES APPLICATIONS ?

La Gestion du cycle de vie des Applications- ou ALM pour Application Lifecycle Management en anglais- peut être définie de plusieurs façons. Nous retiendrons la définition suivante qui comprend des mots clefs très importants. La Gestion du cycle de vie des Application, c'est :

- La gestion **coordonnée** du **cycle de développement** d'une application logicielle...
- ...de son **idée** initiale à sa **livraison** et sa **maintenance**...
- ..impliquant **tous** les **acteurs** participant de près ou de loin au projet...
- ...où qu'ils soient...
- ...tout en appliquant les **processus** propres à l'entreprise et les **politiques** de sécurité.



L'ALM : couvrir le cycle de vie continu du projet applicatif de la collecte des besoins à la livraison du logiciel final



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

L’ALM ne se contente pas de fournir des outils aux équipes d’ingénieurs pour le développement logiciel (Software Development Lifecycle). L’ALM doit aller plus loin et fournir des outils adaptés également pour les autres personnes impliquées dans le projet : les services marketing et commerciaux qui font remonter des informations importantes sur les besoins actuels des marchés, les services méthodes et outils qui vont mettre en place des processus qualité, les services de système d’information qui vont vérifier la sécurité, les utilisateurs finaux qui font remonter des incidents et requêtes d’amélioration du logiciel, etc.

Voilà un projet bien ambitieux nous direz-vous mais l’enjeu est de taille et il existe des solutions matures qui permettent de répondre à ces objectifs.

WWW.SMILE.FR



**L’ALM apporte des réponses à tous les participants au projet logiciel**

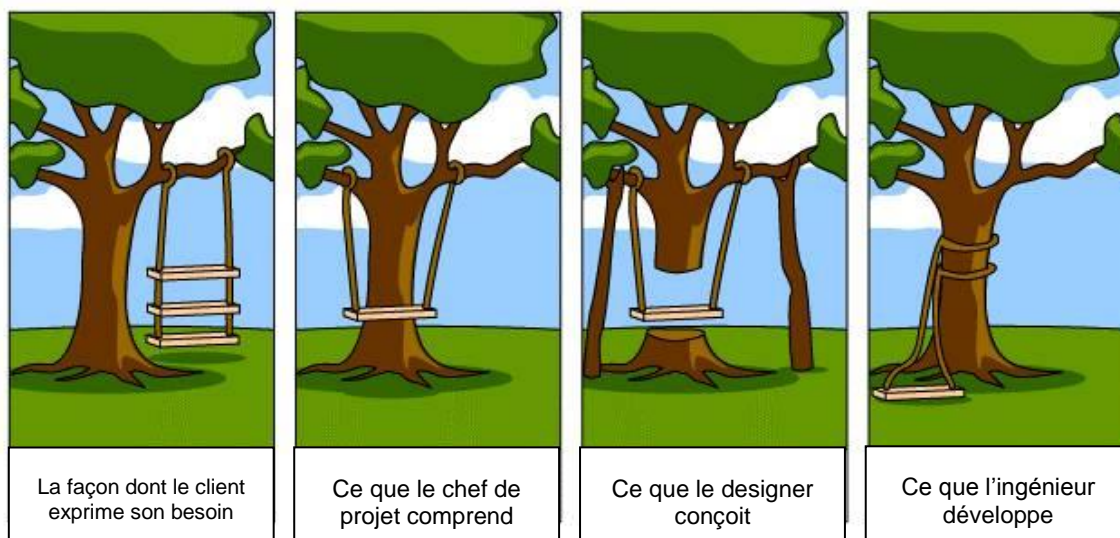
## LES GRANDS CONCEPTS

La Gestion du Cycle de vie des Applications recouvre plusieurs grandes étapes. Dans l’idéal, la solution ALM doit proposer un outil pour chacun d’eux.

### Gestion des exigences

La gestion des exigences intervient dans les phases amont du projet logiciel. La prédictibilité des projets étant devenue indispensable, la gestion des exigences a pour objectif de mettre d’accord le client et le fournisseur sur le périmètre fonctionnel du logiciel à produire.

S’outiller pour la gestion des exigences devient d’autant plus nécessaire lorsque la complexité augmente, que ce soit en terme de nombre d’exigences ou bien du nombre de personnes impliquées. L’illustration ci-dessous souligne ce qu’on souhaite éviter en gérant les exigences :



Il s’agit donc d’identifier et gérer les exigences d’un projet, c’est-à-dire, les besoins/les demandes et les hiérarchiser. En fonction de l’entreprise et de son approche de travail, les exigences peuvent prendre plusieurs formes :

- Selon leur niveau de précision: objectifs, spécifications générales, spécifications détaillées, mode d’utilisation, etc.
- Selon la nature de l’exigence: fonctionnalité, ergonomie, sûreté, robustesse, performance, sécurité, environnement, etc.

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

- Selon les personnes qui les formulent: exigences utilisateurs, exigences métier, exigences techniques, etc.

L'outil ALM va permettre de suivre l'évolution des exigences qui peuvent être amenées à changer au cours du projet. L'objectif est également de détecter s'il y a des incohérences entre elles et s'assurer de leur traçabilité. La **traçabilité** des exigences est le fait de connaître l'origine et les liens entre les exigences, ainsi qu'entre les exigences et les autres éléments du projet.

Par exemple, une traçabilité : Exigence Métier><Exigence Produit><Test><Résultat de Test><Bug, permettra de suivre dans quelle mesure l'exigence est correctement satisfaite. Si ce n'est pas le cas, cela permettra de savoir quelles anomalies ont été rencontrées.

WWW.SMILE.FR

Cross Trackers Search						
Title (Semantic)		Status (Semantic)	Product	Nature	Releases	Complexity
<input type="text"/>		Any	Any None Product A Product B Product C Product D Tuleap	All Robustness Functional Security Hardware Security	Any 0.1 0.2 0.3	Any 1 2 3 4 5 6
<input type="button" value="Submit"/>						
id	Title (Semantic)	Status (Semantic)	Product	Nature	Complexity	
requirement #225	Anticipated error conditions	Closed	Product A	Functional	2	
business request #235	Compatible with mobile device	Closed	Product A	Hardware	1	
requirement #226	Screen layout constraints	Open	Product A	Functional	3	
specification #236	Interoperability	Open	Product A	Security	2	
test #237	Check system requirements	Open	Product A	Security	2	
requirement #227	Application programming interface	Open	Product A	Functional	2	
specification #239		Open	Product A	Security	1	
requirement #234	History	Closed	Product A	Security	1	
use case #348	Supported device types	Closed	Product a	Hardware	1	
test #347	Manage HDR images	Open	Product A			
requirement #233	Shapes	Open	Product A		4	
business request #229	Draw with pencils	Closed	Product A		3	
testcases #232	Pencils should have size	Closed				
testcases #238	Pencils with size 12	Closed				

**Exemple : organiser les exigences et les lier les unes aux autres**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

## Gestion des changements

Un projet est toujours amené à évoluer. Il faut pouvoir stocker et **tracer** tous les changements effectués par les membres de l'équipe sur les artefacts (également appelé *ticket* ou *demande*) du projet. Les artefacts sont tous les éléments de travail du projet qu'il va falloir identifier, créer, assigner et suivre car ils vont être amenés à être modifiés tout au long du projet. On peut avoir :

- Des artefacts d'ingénierie: exigence, risque, bug/incident, demande de support, patch...
- Des artefacts de gestion : tâche, user story, congé, fournisseur, budget...

Le système utilisé pour gérer le changement en ingénierie logicielle est souvent dénommé *tracker*. Le tracker sera lié a minima à un logiciel de gestion de version et encore mieux, à d'autres outils complémentaires.

Demands

Filtres

☒ Statut  Ajouter le filtre:

☒ Appliquer

#	Tracker	Statut	Priorité	Sujet	Mis à jour	Catégorie
<input type="checkbox"/> 3480	Defect	New	Normal	Bot filter plugin crashes when missing useragent	2009-06-11 08:50	Plugins
<input type="checkbox"/> 3479	Defect	New	High	SEGFault in Mysql	2009-06-11 07:15	Tickets
<input type="checkbox"/> 3477	Patch	New	Normal	Fix access handler to remove the need for a separate svn-private/git-private	2009-06-10 21:54	
<input type="checkbox"/> 3476	Defect	New	Normal	Right-floating TOC funky in the roadmap	2009-06-10 19:52	UI
<input type="checkbox"/> 3473	Feature	New	Normal	Can Redmine support notification when a issue will be overdue?	2009-06-10 07:17	Emails
<input type="checkbox"/> 3472	Defect	New	Normal	Role given to a non-admin user who creates a project	2009-06-10 06:37	
<input type="checkbox"/> 3471	Defect	Resolved	Normal	Project managers should be able to assign "sub-project of"	2009-06-10 05:49	Projects
<input type="checkbox"/> 3470	Feature	New	Normal	Projects should inherit documents and files from sub-projects	2009-06-10 23:08	Projects
<input type="checkbox"/> 3469	Defect	New	Normal	senAS	2009-06-10 04:48	Administration
<input type="checkbox"/> 3468	Defect	Resolved	Normal	Subversion : View Differences 500 error	2009-06-09 21:21	SCM
<input type="checkbox"/> 3467	Feature	New	Normal	Due date sort order should sort issues with no due date to the end of the list	2009-06-09 08:12	Tickets
<input type="checkbox"/> 3466	Patch	New	Normal	ja label for text_status_changed_by_changeset	2009-06-09 05:06	
<input type="checkbox"/> 3465	Feature	New	Urgent	Default project	2009-06-09 04:46	
<input type="checkbox"/> 3464	Feature	New	Normal	columns "user_id" & "created_on" in tables like projects, documents & custom_fields...	2009-06-08 21:29	Administration
<input type="checkbox"/> 3463	Feature	New	Low	Export (all) Wiki-Pages to PDF/DOC	2009-06-09 09:02	Wiki
<input type="checkbox"/> 3462	Defect	New	High	CVS path encoding problems	2009-06-08 16:24	SCM
<input type="checkbox"/> 3461	Patch	New	Normal	Manage permission on issue assignment	2009-06-10 20:41	Permissions
<input type="checkbox"/> 3457	Defect	New	Normal	Default value on log text fields	2009-06-08 11:01	Custom fields
<input type="checkbox"/> 3454	Defect	New	Normal	Mercurial Repository Browsing Disappearing	2009-06-06 13:20	SCM
<input type="checkbox"/> 3453	Feature	New	Normal	Issue creation via email by anonymous	2009-06-07 05:51	Emails
<input type="checkbox"/> 3452	Feature	New	Normal	Per project email notifications settings panel	2009-06-07 16:38	Projects
<input type="checkbox"/> 3451	Defect	Resolved	Normal	Issue Creation Via Email not Working	2009-06-10 12:34	Emails
<input type="checkbox"/> 3450	Feature	New	Normal	Project Workflow	2009-06-07 00:31	
<input type="checkbox"/> 3449	Defect	New	High	Redmine Takes Too Long On Large Mercurial Repository	2009-06-10 12:35	SCM
<input type="checkbox"/> 3448	Feature	New	Normal	Add issue watcher that isn't a maintainer	2009-06-10 11:31	

1 2 3 ... 55 Suivant » (1-25/1365) | Par page: 25, 50

Formats disponibles: ☒ Atom | CSV | PDF

### Exemple de trackers suivant les changements de différents types de demandes

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

WWW.SMILE.FR

**Nouvelle demande**

Tracker \*  ☐ Privée

Sujet \*

Description  [Formatage du texte](#)

Statut \*  Tâche parente

Priorité \*  Début

Assigné à  Echéance

Temps estimé  Heures

% réalisé

Impact technique  Impact business

Confiance estimation  Catégorie

Fichiers  Aucun fi... choisi Description facultative

[Ajouter un autre fichier \(Taille maximale: 5 Mo\)](#)

Observateurs

[Prévisualisation](#)

## Exemple de création d'une nouvelle demande



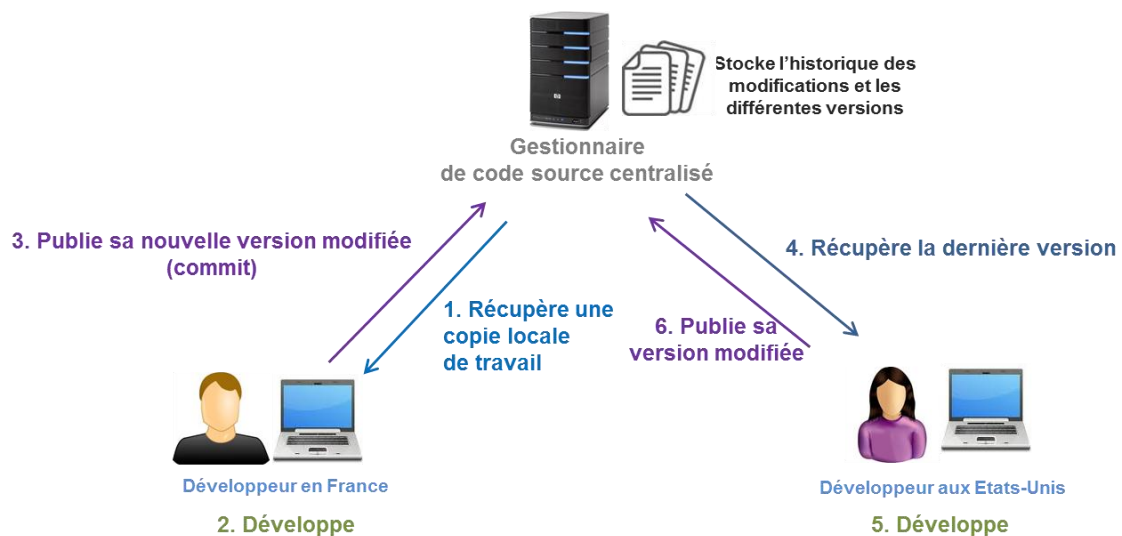
## Gestion de version

La Gestion de version, également appelée gestion de configuration ou gestion de code source, va permettre aux ingénieurs logiciels de **développer ensemble sous contrôle**. Elle permet de **stocker les différentes versions** du code source, **suivre les modifications** apportées et les **comparer entre elles**. L'objectif est de conserver une trace de toutes les modifications qui ont été apportées sur chaque composant logiciel.

Si deux développeurs travaillent sur une même version de code source, l'outil de gestion de version va fusionner les versions et non écraser l'une ou l'autre.

On distingue les outils de gestion de version centralisée, des outils décentralisés.

Le gestionnaire de versions centralisé ne va contenir qu'un seul dépôt des versions. Cela simplifie la gestion des versions mais cela peut devenir contraignant pour certains usages comme le travail sans connexion au réseau ou lorsque l'on travaille sur des branches expérimentales.

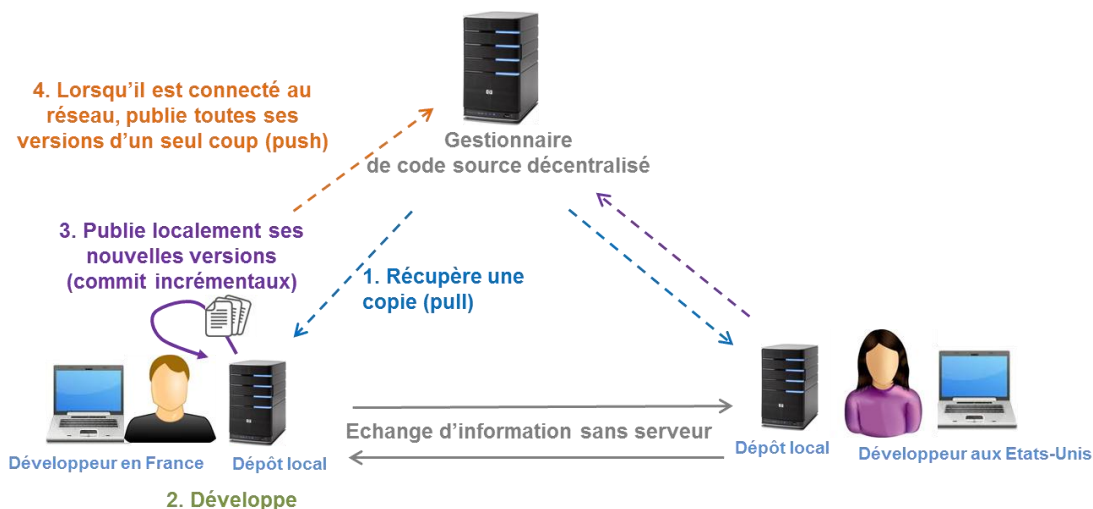


### Fonctionnement d'un gestionnaire de version centralisé

D'où la montée en puissance des gestionnaires de version décentralisée qui apportent bien des avantages. Le principe est d'avoir plusieurs dépôts ce qui permet de travailler sans être connecté au gestionnaire de version. Les opérations sont ainsi plus rapides puisque réalisées en local, sans accès réseau. Le travail peut être fait de façon privée pour réaliser des essais sans devoir publier ses modifications et gêner ses collègues.

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

WWW.SMILE.FR



## Fonctionnement d'un gestionnaire de version décentralisé

Information Subversion | Afficher l'arbre SVN | Mes commits SVN | Recherche SVN | Administration SVN | Notification e-mail | Aide

[Playground]

Index of /

Files shown: 1

Directory revision: 5 (of 5)

Sticky Revision:  Set

File	Rev.	Age	Author	Last log entry
SomePHP/	5	7 months	garon	First step for tasks #75
helloworld.sh	2	7 months	jwitren	Complete fix of bugs #45

### Exemple de navigation dans un dépôt de code source

et

### Exemple de visualisation des différences entre versions de code source

#### Dépôt de sources

[fusionforge] / tags / Branch\_3\_1-merge\_2 / gforge / contrib / beta1\_install\_from\_scratch.txt

Diff of /tags/Branch\_3\_1-merge\_2/gforge/contrib /beta1\_install\_from\_scratch.txt

Parent Directory | Revision Log | Patch

trunk/gforge/contrib /beta1_install_from_scratch.txt revision 2037, Sun May 18 01:18:46 2003 UTC		tags/Branch_3_1-merge_2/gforge/contrib /beta1_install_from_scratch.txt revision 13879, Thu Jul 28 11:07:56 2011 UTC	
#	Line 15	Line 15	
15	1) Downloads	1) Downloads	
16	a) postgres 7.3.2 -- (for the database)	a) postgres 7.3.2 -- (for the database)	
17	ftp://ftp.at.postgresql.org/db/www.postgresql.org/pub/source/v7.3.2/postgresql-7.3.2.tar.gz	ftp://ftp.at.postgresql.org/db/www.postgresql.org/pub/source/v7.3.2/postgresql-7.3.2.tar.gz	
18	b) apache 2.0.44 -- (for the webserver)	b) apache 1.3.27 -- (for the webserver)	
19	http://apache.towardex.com/httpd/httpd-2.0.44.tar.gz	http://ftp.epix.net/apache/httpd/apache_1.3.27.tar.gz	
20	c) php 4.3.1 -- (for the programming language)	c) php 4.3.1 -- (for the programming language)	
21	http://www.php.net/get/php-4.3.1.tar.gz/from/us3.php.net/mirror	http://www.php.net/get/php-4.3.1.tar.gz/from/us3.php.net/mirror	
22	d) gzip 1.1.4 -- (for php, needed by gd library, not sure why)	d) gzip 1.1.4 -- (for php, needed by gd library, not sure why)	

**open  
source**

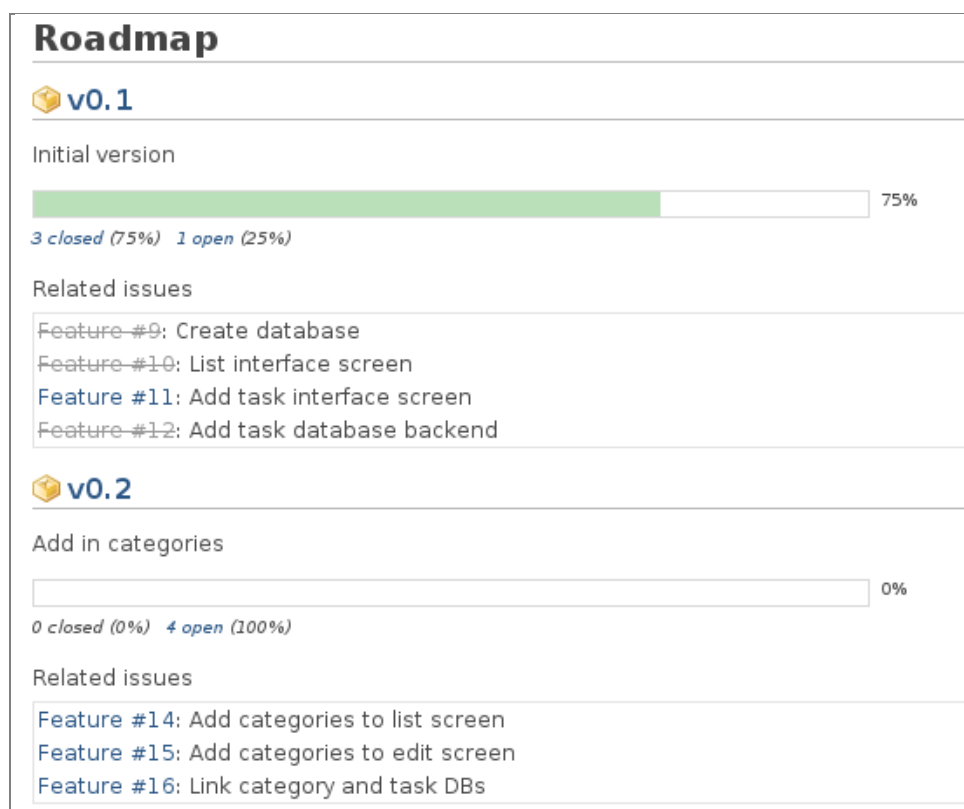
# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## Gestion de projet

La Gestion de projet, incontournable, **permet d’organiser et suivre** les projets applicatifs. Elle englobe également la définition et **la mise en place des processus** de l’entreprise et l’application des **politiques de sécurité**.

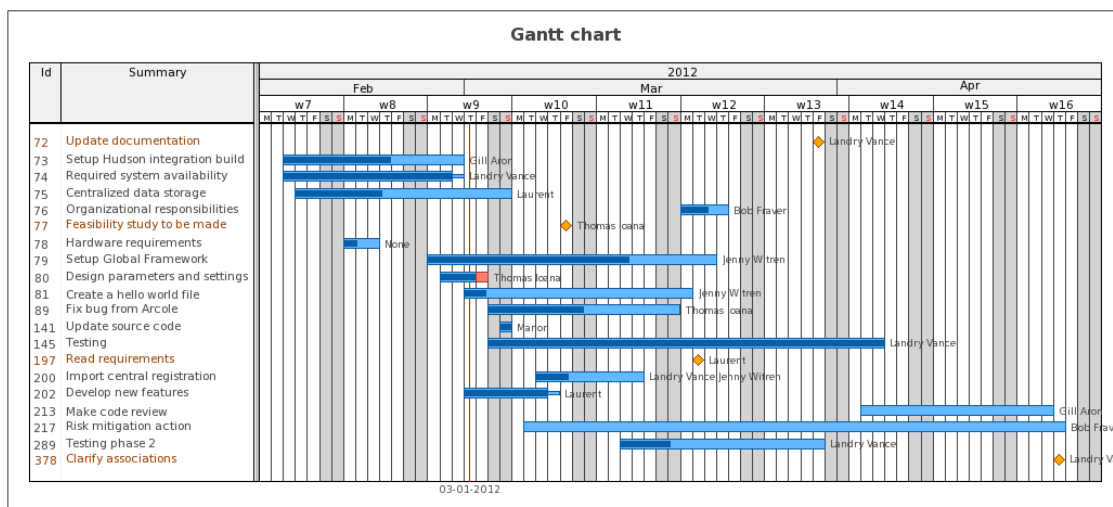
La solution ALM va donc fournir les fonctionnalités pour gérer les tâches, les planifier et suivre l’ensemble du projet.

Exemples d’outils de gestion de projet :

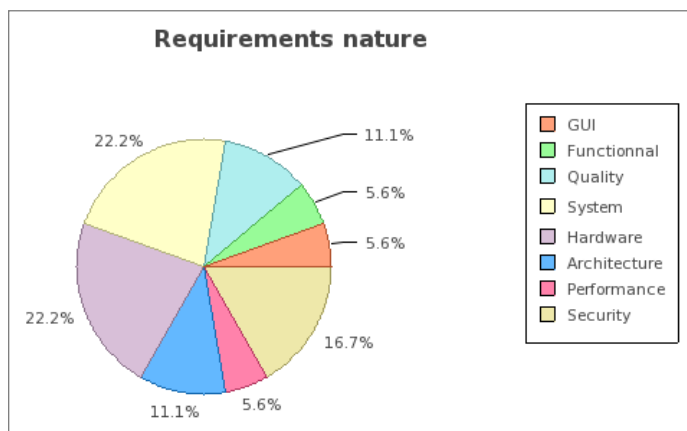


**Progression de l’avancement d’une version**  
**par rapport à la roadmap**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”



**Graphique de Gantt permettant de suivre l'avancement des tâches**



**Graphique représentant la répartition des exigences selon leur nature**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

**Kanban**

Each list is a Pane of issues. The issues can be dragged and dropped onto other panes based on Roles and Permissions settings.

Incoming	Quick Tasks	User	Active	Testing	Finished Requests
<p>#10 - Feature - TSM Data Adapter - UI can display and filter separately</p> <p>#12 - Feature - Accounting Manager - Accounting requires an accrual reconciliation view</p> <p>#24 - Feature - Accounting Manager - Accountability of assets held</p> <p>#25 - Feature - Accounting Manager - Load assets/overhead on demand</p> <p>#34 - Feature - Accounting Manager - Manage intermediate session management</p> <p>#35 - Feature - Accounting Manager - Implement point to point message queue</p> <p>#36 - Feature - General Tasks - Build support</p>	<p>(No issues)</p> <p>Selected Requests</p> <p>#14 - Feature - Accounting Manager - Add billable category for UI policies</p> <p>#33 - Feature - TSM Data Adapter - FOP Gateway, Low add key and call transactions for balance</p>	Chris Nicola	<p>#39 - Feature - Accounting Manager - David's changes to commission report</p> <p>#43 - Feature - TSM Data Adapter - Create UI for loading cash transactions</p> <p>#5 - Feature - General Tasks - Environment Setup</p> <p>#42 - Feature - Insurance Manager - Display records field in policy list</p>	(No issues)	(No issues)
		Adam Dymitrak	#46 - Feature - Dynamics CRM - Evaluate format for FOP and call data integration	(No issues)	(No issues)
		Admin (strator)	#47 - Bug - Insurance - Tracking in system	(No issues)	(No issues)
		Jennifer Keates	#49 - Technical Debt - Unsorted - Test Technical Debt for Kanban Board	#47 - Feature - TSM Data Adapter - Import a new cash transaction	(No issues)
		Lee Toppett-Aylmer	#48 - Architecture - Unsorted - Test Architecture for Kanban Board	(No issues)	(No issues)
		Paul Newton	#22 - Feature - Database - Access Log Page	(No issues)	(No issues)
		Richard Hardinger	(No issues)	(No issues)	(No issues)
		Tim Low	(No issues)	(No issues)	(No issues)
		Unknown User	(No issues)	(No issues)	(No issues)

**Tableau Kanban**

On s'attachera aussi à automatiser les processus avec un système de workflow. Les systèmes de *workflow*, littéralement, « flux de travail » permettent d'automatiser les processus et d'en contrôler le suivi. Ils permettent de définir les conditions de transition d'un état à un autre.

**Définir les transitions**

DE	A L ETAT	A L ETAT	A L ETAT	A L ETAT	A L ETAT	A L ETAT
	Nouveau	Prise en compte	En attente d'info	Contournement	Résolu	Réouvert
(New Artifact)	[Details]					
Nouveau	-	[Details]	[Details]	[Details]	[Details]	
Prise en compte		-	[Details]	[Details]	[Details]	
En attente d'info		[Details]	-	[Details]	[Details]	
Contournement		[Details]	[Details]	-	[Details]	
Résolu					-	[Details]
Réouvert		[Details]	[Details]	[Details]	[Details]	-

**Valider**

**Configuration du workflow d'un tracker d'incident**



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

## Gestion de la qualité : tests et intégration continue

La gestion de la qualité du logiciel a pour objectif de vérifier à chaque modification de code source que le résultat des modifications ne produit pas de **régression** de l'application en cours de développement, c'est-à-dire l'introduction de nouveaux bugs. Il s'agit de détecter au plus tôt les éventuels problèmes d'intégration du nouveau code dans le logiciel existant.

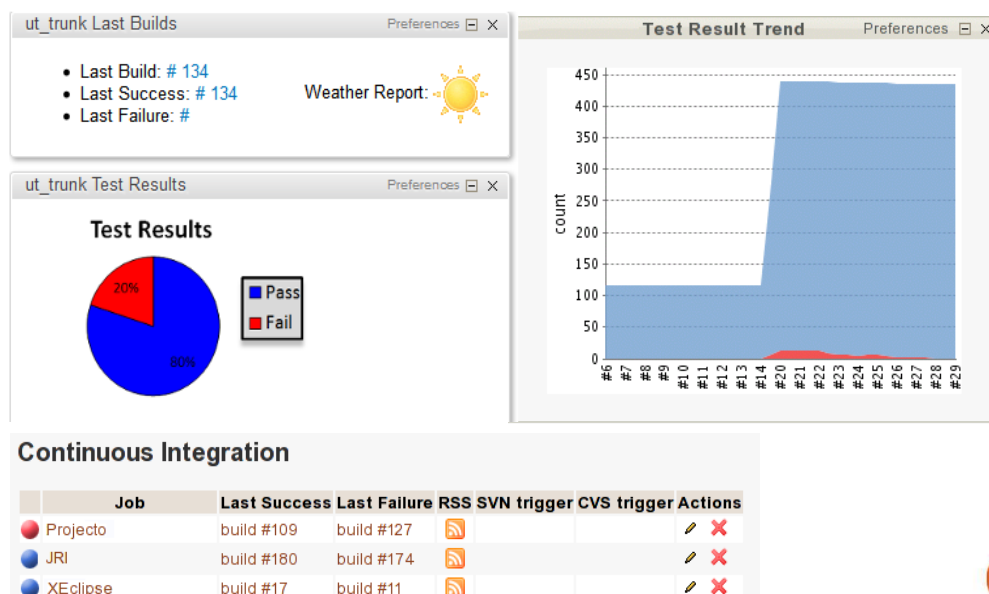
Pour cela il faut automatiser très fréquemment les tests par des revues de codes, des compilations, des tests automatiques. Cela requiert des outils de tests et d'intégration continue.

L'intégration continue présente de nombreux avantages :

- les problèmes d'intégration sont détectés et réparés de façon continue
- on évite les problèmes de dernière minute
- on est rapidement prévenu en cas de code incompatible ou manquant
- une version du nouveau logiciel produit est toujours disponible pour test, démonstration ou distribution
- tout le monde peut accéder aux développements en cours, on peut donc plus facilement récolter du feedback.

Elle permet de partager l'avancement du projet plus facilement par la création de versions intermédiaires et de démonstrations. Le suivi du processus permet de revenir en arrière à tout moment si le développement se révèle non satisfaisant. Ce processus d'amélioration continue permet de prendre en compte des changements à tout moment, sans crainte qu'ils mettent en danger le reste du logiciel.

On peut synthétiser le principe de l'intégration continue par : développer souvent, tester souvent et intégrer souvent, c'est-à-dire industrialiser le développement logiciel.



Exemples de restitution des résultats de l'intégration continue affichables sur le tableau de suivi du projet

## Gestion des livrables

A chaque jalon du projet applicatif, il va falloir mettre à disposition les résultats attendus, c'est-à-dire les **livrables** du projet. Les livrables peuvent être des composants logiciels, des documents, images, vidéos...

L'outil ALM permettra d'organiser ces livrables, de les rendre accessibles à tel ou tel membre du projet, d'expliquer les évolutions par rapport aux précédentes versions (release note) et d'informer les membres de la mise à disposition de nouvelles données. Une bonne pratique est d'organiser les livrables sous forme de *package*, c'est-à-dire un lot, un ensemble de livrables ayant une cohérence, un lien entre eux à une étape du projet.

On parle de création de *baseline* lorsqu'on crée un livrable de référence à un moment donné. La baseline englobe tous les éléments du projet (artefacts, documents, etc.) qui ont permis d'élaborer l'image de référence. Ceci facilite le retour à une version initiale en cas de dysfonctionnement ou d'effet de bord imprévu et doit permettre de reconstruire l'environnement à l'identique.

**Livrables** (Version imprimable)

Sélectionner l'icône à côté du nom de la version pour voir les notes et les modifications correspondantes. Sélectionner un fichier pour démarrer le téléchargement.

[\[Créer un nouveau paquet\]](#)

- PWDev**
- Hello\_World**
- Camellia**

[\[Créer une version\]](#)

v6.34 2011-10-12							
Nom de fichier	Taille (Ko)	D/L	Arch.	Type	Date	Md5sum du fichier	Propriétaire
camellia-1.1.msi	4 593	5	i386	Other Source File	2011-10-12	da7ed721bc994b9f689429681782a024	Erwin

v6.20 2011-10-12							
Nom de fichier	Taille (Ko)	D/L	Arch.	Type	Date	Md5sum du fichier	Propriétaire
camellia-presentation.odt	31	1	PPC	text	2011-10-12	e45dcaa144476c7a84f6b78d06dc4492	Helene
Tracker_Patches.xml	11	1	PPC	text	2011-11-23	a09f3c6b531b7cc07745ed3f0bda1531	Thomas

**Exemple de gestionnaires livrables qui organise les paquets et les différentes versions des fichiers à publier**

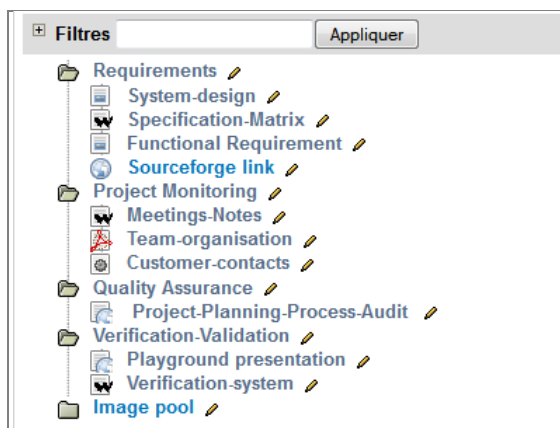
# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

## Gestion documentaire

La gestion de la documentation est un élément essentiel dans un projet ; cela peut vite devenir le chaos si elle n'est pas bien organisée. Il faut donc **centraliser l'espace de documentation, le structurer** en fonction du type de projet. Le gestionnaire de document permettra de **stocker** les documents et conserver les différentes versions du document.

La solution ALM doit permettre de définir finement qui a le droit d'accéder à quel document et qui a le droit de le modifier. Un wiki peut être complémentaire. Il est utile pour faciliter **l'écriture de document de façon collaborative et évolutive**.

WWW.SMILE.FR



Exemple de gestionnaire de documents

Revision as of 19:16, 25 April 2011 (view source) Alberto (Talk   contribs) (cleanup) ← Older edit	Latest revision as of 17:20, 18 November 2011 (view source) EricG (Talk   contribs) (→Debian and derivatives)
(4 intermediate revisions not shown)	
<b>Line 3:</b> There are several versions currently maintained or developed : * 4.8.3 (old stable release) - * <b>""5.0.3"" (recommended for production sites)</b> - * <b>5.1 (not released yet, testers welcome)</b> * trunk (future version, only for developers).	<b>Line 3:</b> There are several versions currently maintained or developed : * 4.8.3 (old stable release) + * <b>5.0.3 (old stable)</b> + * <b>""5.1"" (recommended)</b> * trunk (future version, only for developers).
<b>Line 11:</b> == Installation == - <b>TBD : recommended way to install... depending on context.</b> If official packages are available for Distributions that suit you, it is advised to install with packages (i.e. using apt on Debian-like systems).	<b>Line 11:</b> == Installation == If official packages are available for Distributions that suit you, it is advised to install with packages (i.e. using apt on Debian-like systems).

Exemple de visualisation des différences entre deux pages wiki

## Collaboration Sociale

Avec l’augmentation des équipes distribuées, il est nécessaire de faciliter la **communication et le partage d’expérience** entre les membres d’un projet. Une bonne solution ALM propose nativement des outils de collaboration. Ceci favorise les **échanges** entre les membres des équipes, les clients, les fournisseurs et autres partenaires extérieurs.

Heure Participant	Message
13/10/2011	
09:32	<i>LVance a rejoint le salon de discussion</i>
<lvance>	About the new QA processes now.Please look at the <a href="#">doc #65</a>
09:37 <jwitren>	Yes, I read the document and add few information.
<lvance>	There's an issue with the Hello World software: see <a href="#">bug #45</a> . Someone is working on it?
09:40 <jwitren>	Yes, I already fixed it. See <a href="#">svn #2</a> .
<lvance>	Great, good work!
09:42 <jwitren>	Yes, it is available for our customer TyruAutomotive in <a href="#">file #7</a> .
<lvance>	Thank, see you sson
<jwitren>	Bye!

### Exemple de discussion dans une messagerie instantanée

#### No PFO\_RoleAnonymous role in the database

Surveiller ce forum | Commencer une nouvelle discussion

No PFO\_RoleAnonymous role in the database [ répondre ]

[ forum:415 ]

Par : François on 18/10/2011 13:47

Bonjour,

je voudrais faire migrer notre forge vers fusionforge-5.1.1 et j'obtiens le message cité en sujet pour résultat d'un login ou register.

Bref, si quelqu'un a déjà vu ça ...

François

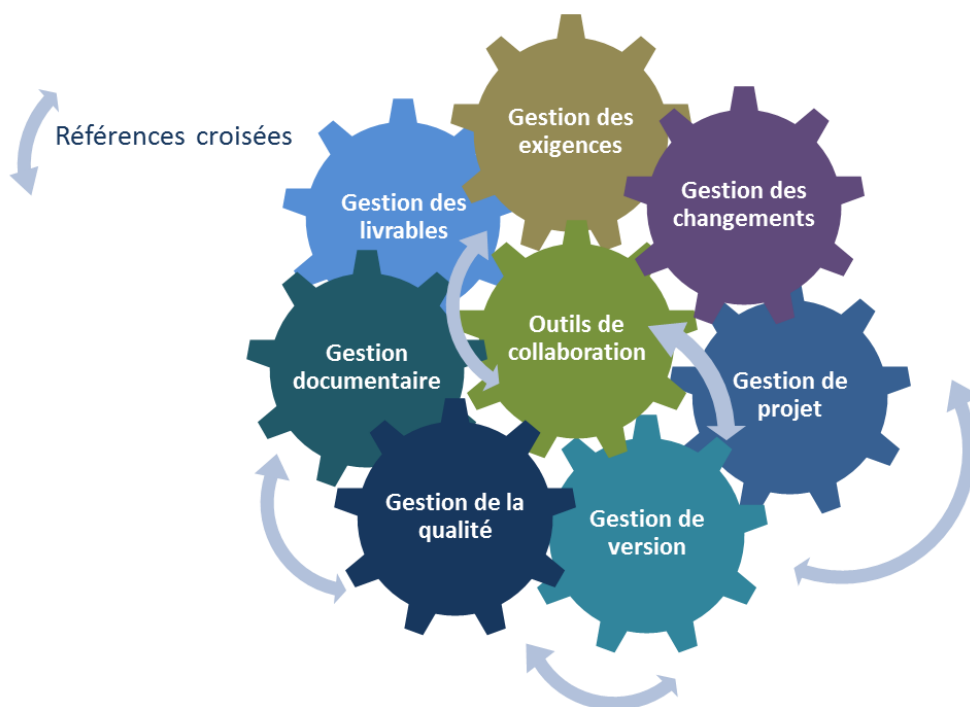
#### Voir les fils de discussion

Discussion	Auteur	Date
No PFO_RoleAnonymous role in the database	François	18/10/2011 13:47
RE: No PFO_RoleAnonymous role in the database	Elodie	21/11/2011 12:52
RE: No PFO_RoleAnonymous role in the database	NickP	21/11/2011 14:26

### Exemple de discussion dans un forum

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

Pour synthétiser cette partie sur les grands concepts de la Gestion du cycle de vies des Applications, voici les différents outils coordonnés, indispensables à la solution ALM:



Les différents outils de l’ALM et leur intégration



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## CE QUE L’ALM VA VOUS APPORTER

### POUR QUI ? POUR QUOI FAIRE ?

De nos jours, c’est clair, **l’innovation par le logiciel est devenue un facteur clef de compétitivité**. La bonne gestion du portefeuille de projets logiciels prend donc une place capitale pour ces entreprises. **Leur capacité à développer des produits innovants et à rester compétitif est accélérée avec l’industrialisation de leurs processus d’ingénierie logicielle.**

Les entreprises « traditionnellement » industrielles sont **en pleine transition** : elles doivent passer d’une fabrication traditionnelle à une fabrication où le composant logiciel est phare.

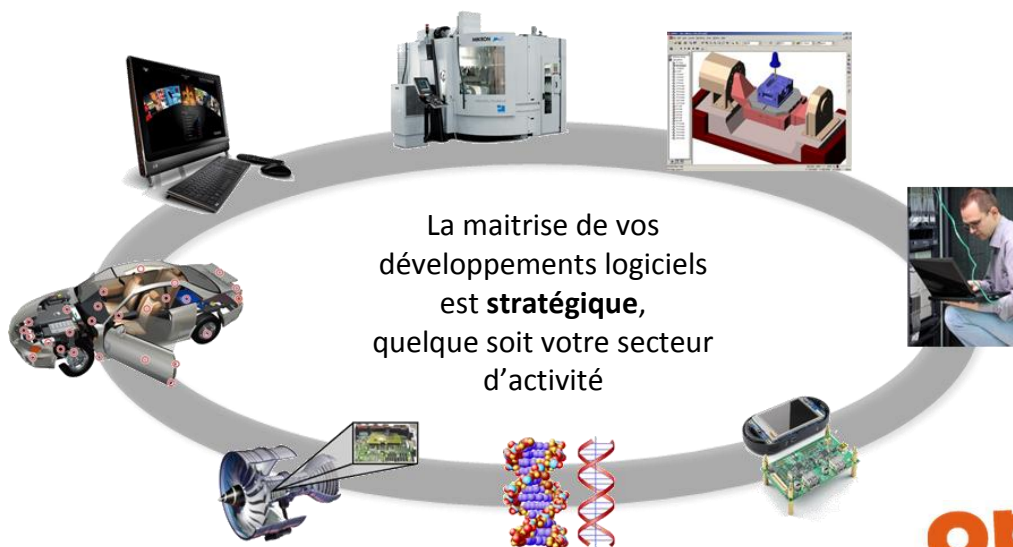
Regardez le secteur de l’automobile par exemple : auparavant une voiture était un ensemble de composants mécaniques. Aujourd’hui c’est également un ensemble de composants électroniques dépendants les uns des autres.

Idem pour le monde médical ou aéronautique, où les chercheurs utilisent des logiciels hautes technologies pour créer de nouveaux produits et développent des logiciels pour les embarquer dans leurs produits.

Autre exemple dans l’industrie des télécoms, avec l’explosion des mobiles smartphone, le développement d’applications est devenu une nouvelle manne pour les entreprises du secteur.

Quelques exemples pour vous montrer qu’à l’heure actuelle **aucun secteur d’activité ne peut échapper au challenge de la gestion de ses projets logiciels.**

**L’ALM est devenu un outil de pilotage du patrimoine applicatif, stratégique pour l’entreprise.**



**open  
source**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

Si vous vous demandez encore si l’ALM est pour vous. Voici quelques questions qui vont vous permettre d’auto-évaluer votre situation :

- ☐ Le logiciel est au cœur du développement de mes nouveaux produits
- ☐ Mon entreprise développe des logiciels embarqués ou fait ses propres développements « maison »
- ☐ Dans mon entreprise, il y a une équipe de développeurs internes (quelque soit sa taille)
- ☐ Il faudrait que je normalise mes processus et réponde aux référentiels qualité
- ☐ Je suis perdu, je n’arrive pas à savoir où en sont mes projets logiciels
- ☐ Je dois montrer que je sais tout ce qu’il s’est passé tout au long de mon projet
- ☐ Ça serait bien d’avoir un outil qui facilite le travail avec mes partenaires
- ☐ J’ai *plusieurs* outils de développement et de gestion de projet

Si vous avez répondu OUI à au moins une de ces questions, alors vous avez besoin d’une solution ALM. **Plus vous vous implémenterez une solution ALM tôt et efficacement, plus vous accélerez votre capacité d’innovation.**

En effet, nous allons voir les grands challenges que l’ALM va relever avec vous.



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## Résumé des grands challenges de l’ALM :

WWW.SMILE.FR

CHALLENGES	BENEFICES APPORTES PAR LA SOLUTION ALM
<p><b>Les équipes sont réparties sur plusieurs sites</b></p> <p><b>Il y a différents profils de personnes impliqués dans le projet</b></p> <p><b>Il faut gérer plusieurs projets à la fois... avec des méthodes différentes</b></p>	<p>Partager les informations de façon sécurisée</p> <p>Suivre en temps réel l’état du projet</p> <p>Améliorer la productivité</p> <p>Faciliter la collaboration/communication entre les membres</p>
<p><b>Je note une multiplication du nombre d’outils indépendants... et sans intégration</b></p> <p><b>Oulala, les difficultés, le temps et le coût de maintenance</b></p>	<p>Solution intégrée, tout-en-un, prête-à-l’emploi : rassembler tous les outils d’ingénierie logicielle en une seule plateforme</p> <p><b>Diminuer le TCO</b> (coût total d’exploitation)</p> <p>Être plus <b>efficace</b> au quotidien</p> <p>Obtenir la <b>traçabilité</b> des changements</p>
<p><b>Les exigences qualité ne cessent de croître... et je dois y répondre</b></p> <p><b>Il faut que j’uniformise et organise les processus</b></p>	<p>Obtenir la traçabilité des changements</p> <p>Industrialiser les développements</p> <p>Améliorer la qualité des livrables</p> <p>Mettre en place les méthodes qualité et agiles</p> <p>Accélérer le «time-to-market»</p>

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## FAIRE COLLABORER TOUS LES SERVICES DE L’ENTREPRISE A L’INNOVATION

L’objet global de l’ALM est d’accélérer la mise sur le marché de produits innovants et de qualité. Cela nécessite l’implication de toute l’entreprise. En effet, l’ALM ne se contente pas de fournir des outils de développement (SDLF, Software Development Lifecycle), l’ALM va plus loin. Elle va faire travailler ensemble tous les services, toutes les personnes, internes et externes à l’entreprise, qui jouent un rôle, à un moment donné, dans la création d’un nouveau produit.

L’ALM doit donc coordonner des hommes, des informations, des processus. L’ALM, c’est comme un ERP des données techniques et logicielles de l’entreprise.

### Sans Suite ALM :

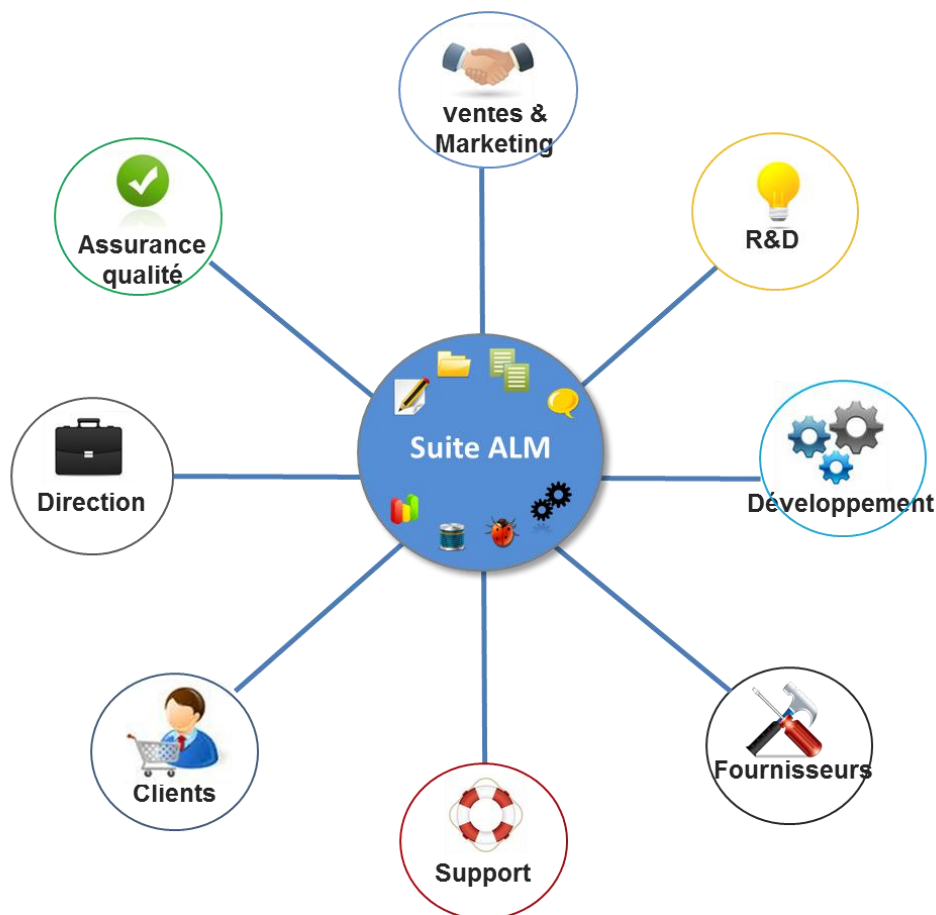
- la collaboration ne se fait pas entre tous les services,
- les échanges de fichiers sont manuels, mal gérés,
- le cercle de développement de produits innovants n’est pas optimisé.



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## Avec la Suite ALM :

- « les silos » entre services sont supprimés,
- les données techniques et de gestion de projet sont centralisées et partagées,
- la collaboration entre les membres est facilitée et plus efficace



## Exemple de retours terrain :

« Avant les services commerciaux ne faisaient pas remonter les demandes clients au service d'innovation. Maintenant ils ont compris l'intérêt de travailler ensemble et ils ont les moyens de le faire. »



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

**CREER UNE CHAÎNE  
CENTRALISÉE D’INGÉNIERIE  
LOGICIELLE ET DE GESTION DE  
PROJET**

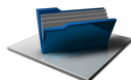
Centraliser les outils techniques de développement ainsi que les outils de suivi d’avancement du projet logiciel est absolument capital. Cela va apporter une vraie valeur ajoutée à l’entreprise qui va percevoir instantanément les avantages que l’ALM apporte.

## Sans Suite ALM :

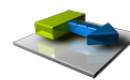
- multiplication d’outils indépendants,
- non intégration des outils donc manque de liens, de traçabilité
- gestion du projet logiciel minimaliste



Gestion des exigences



Gestion des documents



Gestion de tests



Gestion de version



Gestion de livrables



Outils de collaboration



Gestion de projet



Gestion des tâches



Gestion des bugs



Gestion des builds

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## Avec l’ALM :

- Suite prête-à-l’emploi avec des modules intégrés les uns aux autres
- gestion et administration des projets centralisées sur une seule application
- amélioration de la productivité grâce à la facilité d’accès aux informations
- augmentation de la collaboration entre les équipes distribuées et avec les fournisseurs



**Avec l’ALM, centraliser tous les outils  
en une unique plateforme**



**Avec l’ALM, augmenter la collaboration pour  
les équipes distribuées**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

Un autre intérêt de l'ALM repose sur l'administration de l'application qui va se voir grandement facilitée. En effet, le fait que tout soit regroupé en une seule et même application va faciliter la vie des administrateurs systèmes :

## Sans l' ALM

- autant d'installation que d'outils ou modules indépendants
- autant de mises à jour mineures et majeures
- la configuration d'autant d'outils lors de la création de chaque projet, avec la création des nouveaux membres etc.

WWW.SMILE.FR

### OUTILS

  
Gestion des changements

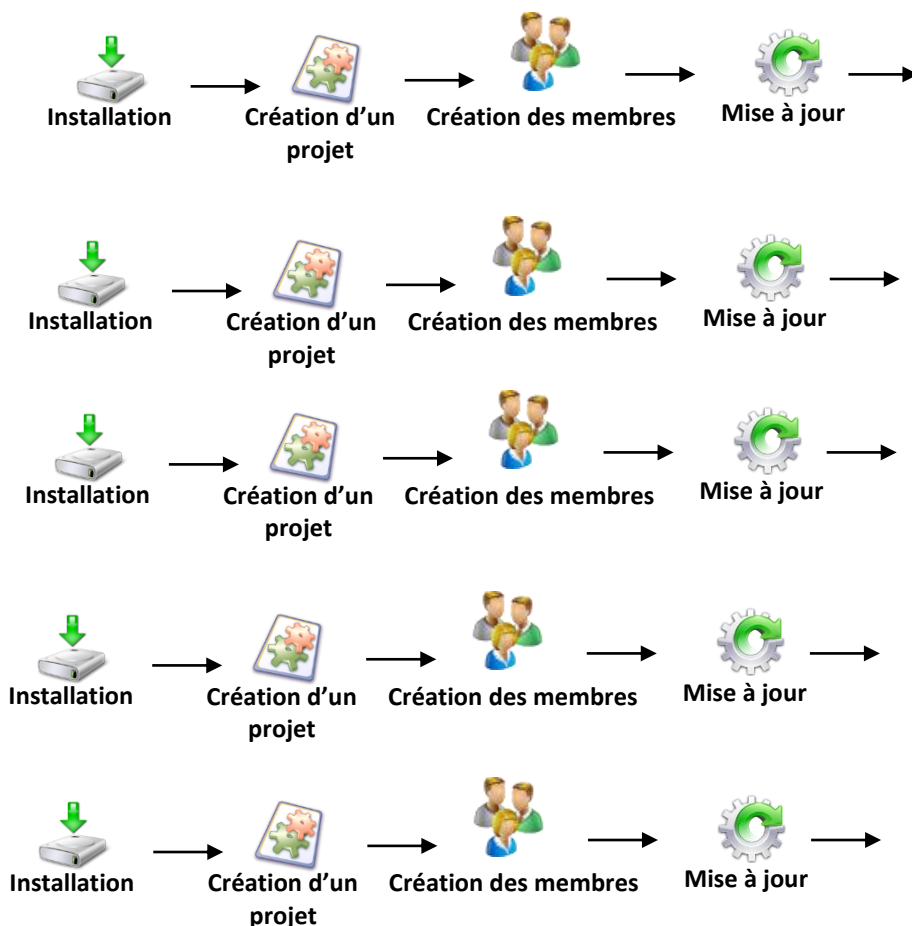
  
Gestion de version

  
Gestion des documents

  
Gestion de projet

  
Outils de collaboration

### ADMINISTRATION ET MAINTENANCE

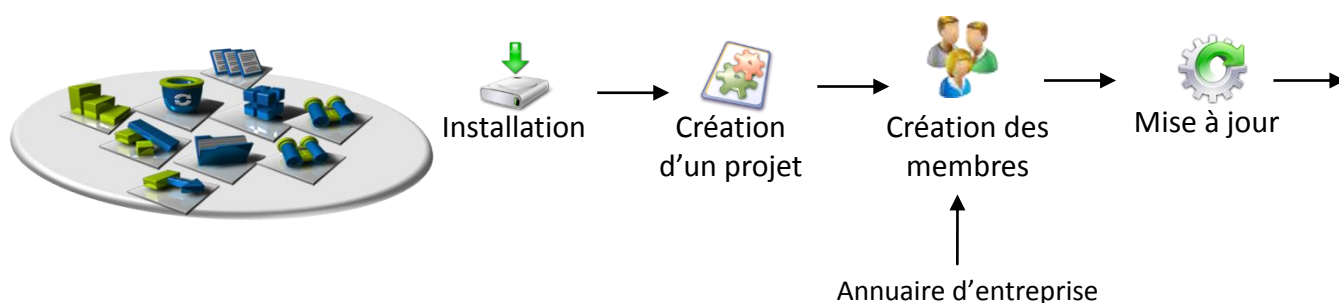


**open  
source**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## Avec l’ALM :

- une seule installation,
- une seule maintenance,
- la création d’un nouveau projet depuis une seule application avec la possibilité d’ajouter simplement des membres au projet si il y a une connexion à l’annuaire d’entreprise par exemple (LDAP, Active Directory...)
- la diminution des coûts de maintenance : TCO (coût total d’exploitation)



Précisons que cela fonctionne parfaitement sous condition que les modules soient intégrés et compatibles en cas de changements de version

**APPLIQUER LES PROCESSUS  
D’ENTREPRISE ET OBTENIR LA  
TRAÇABILITE**

En fonction de son secteur et de ces spécificités, chaque entreprise doit mettre en place des processus qui lui sont propres. S’outiller d’une solution appropriée pour normaliser les processus est indispensable. Elle permettra d’être en conformité avec les normes, certifications et référentiels. L’ALM va s’adapter aux méthodes de travail en configurant les outils en conséquence.

Voici des exemples de méthodes de gestion de projet qui peuvent être mises en place avec une Suite ALM :

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## Cycle en V

Le cycle en V est à l’origine une méthode d’organisation de l’industrie et qui a été adaptée à l’informatique dans les années 80. C’est l’une des premières méthodes de développement. On appelle « cycle en V » parce que l’on descend la réalisation du plus général au plus détaillé, puis qu’on remonte les tests du plus détaillé au plus général.

Les équipes de développement commencent à identifier les limites de cette méthode mais elle reste toujours d’actualité et est encore largement utilisée par les équipes de développement.

## CMMI

Le Capability Maturity Model Integration est un référentiel d’évaluation portant sur la capacité à gérer et à faire aboutir un projet correctement. Il propose un certain nombre de bonnes pratiques liées à la gestion, au développement et à la maintenance d’applications et de systèmes. Il est maintenant largement employé par les entreprises d’ingénierie informatique, les Directeurs des systèmes informatiques et les industriels pour évaluer et améliorer leurs propres développements de produits. Le modèle CMMI définit une échelle de mesure de la maturité à 5 niveaux, ainsi que les indicateurs nécessaires pour évaluer les activités menées par une équipe par rapport à cette échelle.

Plus d’information : <http://www.sei.cmu.edu/cmmi/>



## Méthodes agiles

Les méthodes agiles sont une démarche récente de gestion de projet dans l’ingénierie logicielle. Elles se caractérisent par une interactivité permanente avec le client et la livraison régulière des nouveaux développements pour s’assurer qu’ils répondent bien à ce que souhaite le client. Elles visent ainsi à réduire le cycle de production du logiciel et accélérer son développement, en livrant des versions minimales régulièrement.

On retrouve plusieurs déclinaisons de l’agilité : le Scrum, l’eXtreme Programming, le Kanban, le Lean ou le RAD.

Plus d’information : <http://www.agilealliance.org/>

## CobiT

Le **CobiT** (Control Objectives for Information and related Technology , en français Objectifs de contrôle de l’Information et des Technologies Associées) est un outil fédérateur qui permet de mettre en place un référentiel commun sur la Gouvernance des systèmes d’information. CobiT fournit aux gestionnaires, auditeurs et utilisateurs de TIC, des indicateurs, des processus et des bonnes pratiques pour les aider à maximiser les avantages issus du recours à des techniques informatiques. Le modèle CobiT se focalise sur ce que l’entreprise a besoin de faire et non sur la façon dont elle doit le faire.

Plus d’information : <https://www.isaca.org/>

## ITIL

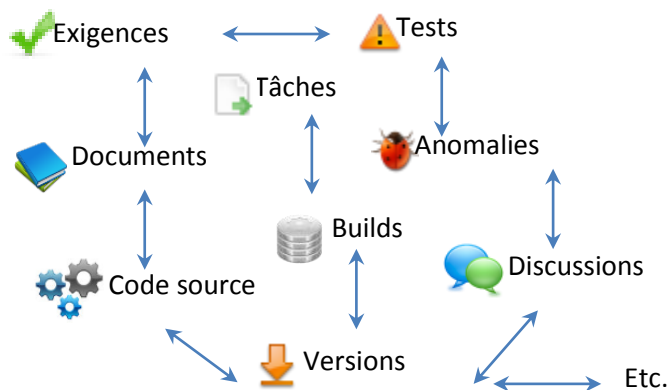
**ITIL** (IT Information Library, en français bibliothèque de l’infrastructure des technologies de l’information) est un cadre de référence proposé par l’OGC (Office of Government Commerce) du Royaume-Uni rassemblant les meilleures pratiques en matière de management des services informatiques. L’objectif d’ITIL est de doter les directions des systèmes informatiques (DSI) d’outils et de documents leur permettant d’améliorer la qualité de leurs prestations, c’est-à-dire améliorer la satisfaction de leurs clients, tout en répondant au mieux aux objectifs stratégiques de l’organisation. Ce référentiel d’origine britannique permet de répondre à 4 types de questions :

- Comment organiser un système d’information ?
- Comment améliorer l’efficacité du système d’information ?
- Comment réduire les risques ?
- Comment augmenter la qualité des services informatiques ?

Plus d’information : <http://www.itil-officialsite.com>

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

La solution de Gestion du Cycle de vie des Applications va favoriser la mise en place de ses processus avec l’assurance de retrouver, pour une version donnée, la trace de toutes les étapes de son développement :



**La traçabilité améliore la visibilité, le contrôle des changements et le cycle de vie du projet logiciel**

## INDUSTRIALISER LE PROCESSUS DE DEVELOPPEMENT

De la même façon que nous avons vécu l’ère de l’industrialisation des processus de fabrication de produits manufacturés, il est l’heure d’industrialiser les étapes de fabrication d’applications logicielles. L’ALM va permettre l’automatisation des processus « d’assemblage logiciel ».

Le développement d’un composant logiciel passe par plusieurs étapes de codage, de tests, de compilation... En fournissant aux équipes de développeurs des outils qui leur permettent d’automatiser des tâches répétitives, ils peuvent se concentrer sur des activités à plus forte valeur ajoutée et produire des applications de meilleure qualité.

### Sans Suite ALM :

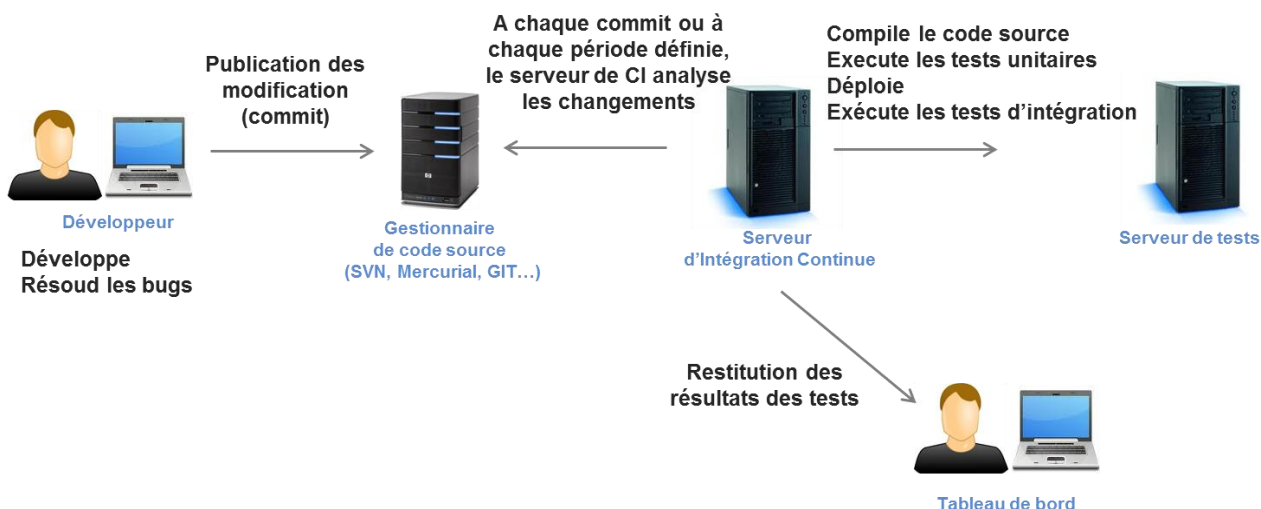
- le code est simplement poussé vers le gestionnaire de code source.
- les tests sont réalisés séparément et considérés comme une étape supplémentaire de travail.

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”



## Avec Suite ALM :

- on met en place une véritable industrialisation des développements
- les processus de test et de déploiement sont automatisés
- la qualité et la fiabilité des développements sont grandement améliorées



## Exemple de retours terrain :

« Nous avons réduit à 30 minutes et de façon automatique, une étape qui pouvait demander à l'équipe plusieurs jours. »

« Nos utilisateurs sont confiants dans nos nouvelles versions. Ils savent que le nombre d'incidents qu'ils rencontreront sera très limité »

## POURQUOI CHOISIR UNE SOLUTION OPEN-SOURCE POUR VOTRE ALM ?

### 4 BONNES RAISONS

Nous avons vu l’importance d’une bonne solution ALM et de sa pérennité pour les entreprises d’aujourd’hui. Voici 4 bonnes raisons de choisir une solution open-source pour votre ALM.

#### Innovation

La **grande majorité des innovations en ingénierie logicielle sont impulsées par les communautés libres**. En réalité, ce sont les communautés, souvent constituées d’experts et de passionnés, qui ont lancé les grandes tendances de l’ALM : nouveaux outils (ex : CVS puis Subversion), nouvelles méthodes (ex : méthodes agiles). La dynamique positive créée par le partage du code accélère les innovations et les bonnes pratiques de développement.

En choisissant une solution ALM open-source, **vous êtes les premiers à bénéficier de cette création de valeur tant technologique que méthodologique**.

#### Flexibilité-Extensibilité

Les solutions ALM open-source sont souvent basées sur une architecture en *plugin*, c’est-à-dire un module d’extension, qui offre un potentiel d’extensibilité et de modularité illimité. Vous pouvez donc ne déployer que les fonctionnalités qui vous sont nécessaires et surtout les **adapter très précisément à vos besoins**.

Seule une solution open-source peut s’adapter aussi finement aux **spécificités de votre métier** et vous donne la possibilité d’étendre votre solution avec les standards ouverts et les *API* (interface de programmation qui favorise l’interaction des programmes les uns avec les autres).

#### Interopérabilité

L’*interopérabilité* d’un logiciel, c’est sa capacité à fonctionner avec d’autres technologies existantes ou futures. L’interopérabilité passe souvent par l’usage de *standards ouverts* qui sont le meilleur moyen de garantir l’accessibilité des données et leur pérennité. Les logiciels open-source sont par nature interopérables grâce à leur code source ouvert.

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

Un logiciel ALM open-source est donc le plus ouvert, non seulement en terme de code source, mais également au niveau des formats de fichiers et des protocoles de communication. Cela garanti une interopérabilité optimale.

## Liberté et Indépendance

La compétitivité de votre entreprise repose sur sa capacité d’innovation. N’enfermez pas vos innovations logicielles et vos savoirs-techniques dans un logiciel dépendant d’un éditeur.

Gardez votre liberté et votre indépendance. Restez libres de changer d’avis. Soyez sûr que vous pouvez conserver votre solution ALM en l’état, quoi qu’il advienne.

### MISE EN GARDE

## Logiciels libres vs Logiciels ouverts, sachez faire la différence !

Vous avez bien compris l’intérêt des outils libres, mais attention, il persiste une confusion courante entre les termes « libre » et « ouvert » qui induisent souvent les décideurs en erreur.

Certains produits d’ailleurs profitent de cette ambiguïté qui joue en leur faveur. Les logiciels libres et les logiciels ouverts ont certains objectifs communs, mais ces deux concepts sont tout de même bien différents : schématiquement, le logiciel libre concerne l’accès et l’utilisation du code source alors que les standards ouverts traitent de l’échange d’information.

## Qu’est-ce qu’un logiciel OUVERT ?

Un logiciel ouvert signifie un logiciel utilisant des standards ou formats ouverts. La définition légale en France d’un standard ou format « ouvert » est la suivante : « un logiciel ouvert offre des formats de données interopérables et dont les spécifications techniques sont publiques et sans restriction d’accès ni de mise en œuvre » par opposition à un format fermé ou propriétaire.

Les formats ouverts sont intéressants pour leur interopérabilité. Mais attention, **un logiciel ouvert ne signifie pas qu’il est libre. Il apporte des garanties différentes pour l’entreprise. C’est à ce niveau que se crée la confusion.** Revenons sur ces deux notions pour bien comprendre leurs avantages et/ou limites.



## Qu’est-ce qu’un logiciel LIBRE ?

Selon la Free Software Foundation, un logiciel est considéré comme libre si sa licence garantit à l'utilisateur les 4 libertés suivantes:

- accéder au code source,
- étudier le fonctionnement du programme et l'adapter,
- redistribuer des copies,
- améliorer le programme et publier les améliorations pour en faire profiter toute la communauté.

L'accès au code-source seul, ne suffit pas pour qu'un logiciel puisse être dit « libre ». Logiciel libre ne signifie pas gratuit. Un logiciel libre n'est pas un freeware ou shareware.

La licence GNU General Public License (GPL) concrétise ces quatre libertés sous la forme d'une licence juridique.

## Les avantages d'un logiciel LIBRE pour l'ENTREPRISE?

Au-delà des considérations philosophique ou juridique, le logiciel libre apporte plusieurs garanties pour l'entreprise, qu'aucun autre logiciel ne peut apporter:

- Utiliser librement le logiciel pour une évaluation ou un usage en production
- Apporter des modifications au logiciel, l'adapter aux besoins et processus de l'entreprise en accédant au code source
- Profiter plus rapidement d'innovations : parce qu'un logiciel libre dévoile et permet la modification du code, les contributeurs peuvent participer à son amélioration et à son développement.
- **Conserver le logiciel, en cas d'arrêt de contrat par exemple, sans aucune limitation technique ou de temps. Ceci confère au logiciel libre une garantie d'indépendance sur le long terme qu'aucun autre logiciel ne peut apporter.**

Ce dernier point fait une vraie différence avec un logiciel ouvert qui ne peut garantir cela, ce qui réduit l'indépendance de l'entreprise vis-à-vis de l'éditeur.

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## Et l’open-source dans tout ça ?

Les désignations Free Software (logiciel libre en français) et Open Source sont deux désignations concurrentes pour un type de licence de logiciel assez proches. Dans les usages, le terme Logiciel libre tient à insister sur la finalité philosophique et politique de la licence. La désignation Open Source fournit une approche plus pragmatique des avantages du logiciel libre afin de n'en conserver que les avantages sur le plan de l'ingénierie.

**Retenons qu’un logiciel ouvert offre des caractéristiques intéressantes mais il n’apporte pas les garanties de liberté et d’indépendance comme le fait un logiciel libre (ou open-source).**

## LES COMPOSANTS DE L’ALM

Voici des outils libres d’ingénierie logicielle qui ne couvrent qu’une partie de la Gestion du cycle de vie des Applications.

### DEVELOPPEMENT

#### Eclipse

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

Eclipse est un environnement de développement , écrit en Java, extensible et polyvalent, initié par IBM en 2001, et aujourd’hui porté par une fondation Eclipse qui en est le porteur officiel.

Conçu autour d’une plateforme commune à laquelle s’agrègent des composants dérivatifs, le projet est ainsi constitué de nombreux sous projets spécifiques aux technologies sous jacentes.

L’objet de la solution Eclipse est de fournir des outils favorisant la productivité, mais pas seulement celle qui concerne le codage logiciel. On y trouve des environnements de développement intégré mais également de conception, de modélisation, de tests, de reporting, etc. Eclipse a beau être écrit en Java, il peut être utilisé pour développer sous n’importe quel langage de programmation.

Eclipse est publié sous la licence EPL (Eclipse Public License).

### GESTION DE VERSION

#### Subversion

<http://subversion.apache.org>

Subversion (SVN) est un système de gestion de version centralisé écrit en C.

Issu de CVS, son développement est initialisé en 2000 par la société Collabnet. Il est devenu officiellement un projet de la fondation Apache en 2010.

Standard et populaire, il a été choisi par de nombreuses communautés du logiciel libre. De nombreux outils et ressources sont disponibles pour l'exploiter au mieux. Apache Subversion a été écrit pour combler les manques de CVS dont seul l'implémentation avait été remise en cause (i.e pas son concept). Certaines fonctionnalités ont été ajoutées : les répertoires et les métadonnées sont versionnés, les numéros de révision sont globaux pour l'ensemble du dépôt, il est possible de renommer ou de déplacer des fichiers sans perte de l'historique, etc.

Subversion est distribué sous licence Apache et BSD.

## Git

<http://git-scm.com>

Git est un système de gestion de version décentralisé (DVCS), porté par une communauté. Il est écrit en C, Bourne Shell et Perl.

Il est notamment utilisé pour le noyau Linux ou pour PHP. C'est un logiciel libre créé par Linus Torvalds en 2005.

Git permet notamment de "commiter" localement puis de pousser aux autres développeurs un ensemble de commits locaux. Il permet également d'utiliser un workflow de développement en soumettant par exemple l'envoi de code à l'approbation d'un des développeurs. La faculté de Git à créer des branches facilement ainsi que de permettre leur administration de façon simple en fait un outil de choix dans le cadre de développement de projets open source.

Git est distribué sous la licence GPL v2.

## Autres

Nom	Url/Site web
CVS	<a href="http://cvs.nongnu.org">http://cvs.nongnu.org</a>
Bazarr	<a href="http://bazaar.canonical.com/en/">http://bazaar.canonical.com/en/</a>
Mercurial	<a href="http://mercurial.selenic.com">http://mercurial.selenic.com</a>

## GESTION DES CHANGEMENTS/TRACKING

### MantisBT

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

MantisBT est un outil web de suivi de tickets (anomalies, demandes d'évolutions, demandes d'intervention, ...), adapté à tous types de projets. Il est supporté par une communauté. MantisBT est développé en PHP et nécessite l'utilisation d'une base de données (MySQL, PostgreSQL, MS SQL ou DB2). Il est principalement testé pour les serveurs Web Apache et IIS.

Kenzaburo Ito, initiateur du projet en novembre 2000, est rapidement rejoint en 2002 par Jeroen Latour, Victor Boctor et Julian Fitzell pour constituer le cœur de l'équipe de développement.

MantisBT permet une répartition des tickets par projet. Il est également possible de qualifier les demandes suivant différents critères (type, catégorie, sévérité, priorité, privé/public).

L'outil dispose notamment de fonctionnalités comme un workflow d'enchaînement d'états paramétrables, un système de notification, un formulaire de recherche de tickets avec filtres, une page personnalisée pour chaque utilisateur listant les tickets dont il est l'auteur ainsi que les tickets qui lui sont assignés, une gestion de droits utilisateurs, une rubrique d'administration par projet permettant d'ajouter/supprimer des utilisateurs au projet, la création de sous projets, etc.

MantisBT est distribué sous licence GNU General Public License 2 (GPL v2).

### Autres

Nom	Url/Site web
Bugzilla	<a href="http://www.bugzilla.org/">http://www.bugzilla.org/</a>
Trac	<a href="http://trac.edgewall.org/">http://trac.edgewall.org/</a>



## OUTILS DE BUILDS

### Ant

<http://ant.apache.org>

Ant ("Another Neat Tool") est un outil d'automatisation de tâches tout au long du cycle de développement logiciel. Ant a été conçu par James Duncan Davidson en 2000 (sortie de la première version). Ant est portée par la fondation Apache. Il est écrit en Java.

Ant permet donc d'automatiser des opérations répétitives tout au long d'un projet, à l'instar de ce que peuvent faire les commandes Make. Ant est principalement utilisé dans le cadre de projets écrits en Java même s'il peut être utilisé dans des projets faisant appels à d'autres technologies. Parmi les tâches que l'on confie souvent à Ant, on peut lister : l'exécution d'outils externes comme Checkstyle ou FindBugs, la génération des documentations (Javadoc), la compilation des programmes, l'archivage, la sauvegarde, etc.

Ant est distribué sous la licence Apache, et soutenu par la fondation du même nom.

### Phing

[www.phing.info/trac](http://www.phing.info/trac)

Phing est un outil de build, inspiré de Ant et développé à l'origine par Andreas Aderhold. Il est supporté par une communauté. Phing est développé en PHP et est compatible avec les plateformes Windows, Linux et Mac OS X.

Il est notamment utilisé pour automatiser les tâches récurrentes lors du développement de projets informatiques tels que lancement de tests unitaires ou d'outils d'analyse de code ou encore pour la gestion des livraisons.

Phing utilise, comme Ant, un fichier XML pour définir une liste d'actions ("target"), comportant chacune une ou plusieurs tâches (copie de fichiers, lancement d'une commande), pouvant être lancées en ligne de commande. Il intègre une large liste de tâches en natif (exécution de tests PHPUnit, analyse de code avec PHP CodeSniffer, PHP Mess Detector, manipulation de fichiers, travail avec des gestionnaires de versions de sources), qu'il est très facile d'étendre en fonction de ses besoins par l'intermédiaire de nouvelles classes PHP.

Distribué à travers le gestionnaire de paquets PEAR, Phing peut utiliser, pour certaines de ses tâches, des bibliothèques annexes telles que Xdebug, phpcs.

Phing est distribué sous licence LGPL (Lesser Public General License) en version 3.

## Maven

<http://maven.apache.org>

Maven est un outil de gestion de projet technique écrit en Java. Son développement est assuré principalement par la fondation Apache.

Maven permet de standardiser la forme d'un projet ainsi que son utilisation. Il permet également de gérer les dépendances d'un projet, d'effectuer une livraison complète et automatique, de déployer une application, et de faciliter le déploiement d'un projet au sein d'une plateforme d'intégration continue.

Maven offre également une intégration poussée de nombreux outils de reporting (Surefire, PMD, CheckStyle, NCSS, etc.). Maven est destiné aux projets Java en général et aux projets Java EE en particulier.

Maven est distribué sous la licence Apache.

OUTILS DE  
TESTS/INTEGRATION  
CONTINUE

## Continuum

<http://continuum.apache.org>

Continuum est l'outil d'intégration continue de la fondation Apache, écrit en Java.

Continuum offre toutes les fonctionnalités que l'on attend d'un tel outil. Automatisations configurables bien sûr, mais aussi distribution du build sur des machines esclaves, configuration de différents environnements d'exécution, gestion très fine des droits et reporting intégré pour les résultats des tests unitaires.

La fonctionnalité intéressante dans Continuum est la possibilité de regrouper des modules/projets au sein de groupes de projets. Au sein de ces groupes, les différentes configurations et droits sont mutualisés ce qui permet de gérer un ensemble de projets de façon simple même quand ce nombre augmente rapidement.

Continuum est distribué sous la licence Apache.

## Jenkins / Hudson

<http://hudson-ci.org>

Solution portée par un éditeur (Oracle)

Jenkins est un outil d'intégration continue. Hudson est historiquement l'outil d'intégration continue développé par Oracle/Sun, Jenkins est un fork de Hudson. On évalue ici Jenkins et Hudson, deux outils écrit en Java.

Jenkins permet d'automatiser la construction de projets et de générer des rapports de tests et de qualité. Jenkins est majoritairement utilisé dans le marché des solutions d'intégration continue. Le grand atout de Jenkins est son écosystème composé de centaines de plugins, ainsi que son interface plus simple et moins austère que celle de Continuum par exemple. Les générations de projets peuvent être initiées par différents moyens (mécanismes de planification similaires au cron, des systèmes de dépendances entre générations, ou par des requêtes sur certaines URL spécifiques).

Jenkins est distribué sous les licences MIT et Creative Commons.

## Autres

Nom	Url/Site web
Sonar	<a href="http://www.sonarsource.org">http://www.sonarsource.org</a>
Checkstyles	<a href="http://checkstyle.sourceforge.net">http://checkstyle.sourceforge.net</a>
JUnit	<a href="http://www.junit.org">http://www.junit.org</a>
PHPUnit	<a href="https://github.com/sebastianbergmann/phpunit">https://github.com/sebastianbergmann/phpunit</a>
CruiseControl	<a href="http://cruisecontrol.sourceforge.net">http://cruisecontrol.sourceforge.net</a>
FindBugs	<a href="http://findbugs.sourceforge.net">http://findbugs.sourceforge.net</a>

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## LES SOLUTIONS ALM INTÉGRÉES OPEN SOURCE

Nous avons vu quelques composants d’ingénierie logicielle du monde de l’open source. Il en existe une multitude. Ils sont intéressants mais ne couvrent qu’une partie de la Gestion du cycle de vie de développement. Ce qui est encore plus intéressant, ce sont les solutions plus intégrées qui couvrent plusieurs étapes du cycle de vie, apportant une réelle valeur ajoutée.

### 3 SOLUTIONS ALM INTEGREES OPEN-SOURCE

Le marché des solutions ALM étant encore en train de se structurer on trouve des solutions plus ou moins avancées dans le monde propriétaire. Dans l’open-source, nous avons sélectionné 3 solutions ALM qui nous apparaissent comme les plus pertinentes et prometteuses : Fusionforge, Redmine et Tuleap.



**open  
source**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## Fusionforge

Fusionforge ([www.fusionforge.org](http://www.fusionforge.org)), issu de Sourceforge a été l’une des premières forges libres disponibles. Dénommé auparavant GForge, l’outil a été renommé lorsque l’éditeur originel de GForge, le GForge Group a décidé de ne plus maintenir la version libre et de ne développer que sa version propriétaire GForge As.

Fusionforge est sous licence GNU GPL. La dernière version stable est la 5.1, elle date de Juillet 2011. L’outil est principalement développé pour les systèmes Linux : il est compatible avec Debian, CentOS et RedHat 5. Il nécessite également un serveur Apache, PHP, PostgreSQL.

Fusionforge est notamment utilisé par la Communauté d’innovation d’Alcatel-Lucent, l’INRIA ou l’Atelier, la forge pour les administrations et collectivités locales.

L’activité de la communauté a connu une phase descendante pendant plusieurs années mais un regain d’activité l’anime depuis 1 ou 2 ans, sans doute sous l’impulsion du projet Coclico, désormais terminé.

### Périmètre fonctionnel :



### Périmètre fonctionnel de Fusionforge



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

FusionForge

Bugs

Search

Advanced search

Log In

New Account

Home

My Page

Projects

FusionForge

Summary

Activity

Forums

Tracker

Lists

Tasks

Docs

News

SCM

Files

Mediawiki

Hudson

Bugs

View Trackers | Bugs | Download .csv | Monitor

Displaying results 1-25 out of 99 total.

Advanced queries | Simple Filtering and Sorting | Default

Viewing only opened records by default, use 'Advanced queries' or 'Simple Filtering and Sorting' to change.

ID	Summary	Open Date	State	Target Release	Assigned to	Submitted by
292	Mailman list admin should be granted to project admins	* 2011-04-06 10:03	Open	None	Nobody	Olivier Berger
36	[4.7.3] Trove tree never populated	* 2009-05-27 08:54	Open	None	Nobody	Benjamin David
50	SVN permissions	* 2009-09-29 17:31	Open	4.8.x	Nobody	Rob Dora
55	[TRUNK] Failing creation of SVN branches	* 2009-10-03 13:08	Open	5.0.x	Roland Mas	Laurence -
60	gforge-mta-postfix doesn't install	* 2009-11-01 15:54	Open	None	Nobody	Florian Effenberger
66	SOAP functions getUsersByName and getGroupsByName should exlude some input as invalid	* 2009-11-18 17:11	Open	4.8.x	Nobody	Olivier Berger
77	MediaWiki VMware img Bug	* 2009-12-24 08:28	Open	None	Nobody	fejzof fjehezzyuf
84	SVNTracker 4.8.2	* 2010-02-15 09:05	Open	4.8.x	Nobody	Chanh TRAN
85	CVSTracker 4.8.2	* 2010-02-15 09:09	Open	4.8.x	Nobody	Chanh TRAN
106	Assigned To: Button on trackers has no memory	* 2010-03-11 17:43	Open	None	Nobody	Jeff Schlaybach
107	Loggerhead seems to fail on some particular b2r revisions	* 2010-03-12 13:36	Open	None	Nobody	Olivier Berger
108	"make phpdoc" takes more memory than available	* 2010-03-12 13:46	Open	None	Olaf Lenz	Olaf Lenz
114	There are 2 Site News Admin project on a fresh install	* 2010-03-24 22:53	Open	None	Nobody	Christian Bayle
120	Installation on Debian 5 fails on postgres setup	* 2010-03-30 22:18	Open	None	Christian Bayle	Julian Golderer
128	Error message not very clear when attaching a file to an issue	* 2010-04-13 08:52	Open	None	Nobody	Sylvain Le Gall
134	Installation error on Ubuntu: update-inetd required	* 2010-04-28 15:05	Open	None	Nobody	Olaf Lenz
140	FF 5.0.1 : Group.class.php : mixture of 'sys_lang' & 'user lang' ...	* 2010-05-12 11:34	Open	5.0.x	Nobody	Chanh TRAN
143	FF 5.0.1 : group deletion not working ...	* 2010-05-27 10:01	Open	5.0.x	Nobody	Chanh TRAN
159	FF 5.0.1 : Tasks	* 2010-06-18 07:35	Open	5.0.x	Nobody	Chanh TRAN
174	bug in task browser	* 2010-06-22 15:27	Open	5.0.x	Nobody	Gael Merlin

## Vue du tracker

L'outil est intéressant par les aspects suivants :

- le support de **plusieurs outils de gestion de version** : Arch, Bazaar, CVS, Darcs, Git et Subversion
- **deux wikis différents** : PHP wiki et Mediawiki plus facile d'accès pour les non initiés

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

Page 50

Set \$wgLogo to the URL path to your own logo image.

navigation

- Main page
- Community portal
- Current events
- Recent changes
- Random page
- Help

search

Go Search

toolbox

- What links here
- Related changes
- Special pages
- Printable version
- Permanent link

**Main Page**  
(Difference between revisions)

**Revision as of 14:47, 7 October 2011 (view source)**  
Olberger (Talk | contribs)  
m (→Using or administrating FusionForge)  
← Older edit

**Latest revision as of 14:50, 7 October 2011 (view source)**  
Olberger (Talk | contribs)  
(→Getting in touch with "the project")

**Line 17:**  
== Getting in touch with "the project" ==  
The project is loosely organised, but we tried to document our  
-[[Organization]] nevertheless, as well as a list of [[Maintainers]]  
for subsystems and plugins.

**Line 17:**  
== Getting in touch with "the project" ==  
The project is ""loosely organised"", but we tried to document  
+our [[Organization]] nevertheless, as well as a list of  
[[Maintainers]] for subsystems and plugins.  
**I you need to talk to us, you'll then understand that your  
best bet (so far) is either to pick one member out of  
+random, or better contact the whole project "via the  
channels above".... until we decide we setup a PR crew for  
just that kind of contact enquiries.**  
We also sometimes organize gatherings, such as the [[2 days  
Hackers/users meeting in Paris, 2010-02-03 and 04|February  
2010 meeting]], where developers, users and other interested  
parties can meet face-to-face and discuss technical as well as  
non-technical questions.

## Exemple d'une page wiki visualisant les différences entre versions

- **l'intégration avec le tracker Mantis** en plus de celui fournie par défaut
- **plugin OSLC** pour les trackers. L'OSCLC pour Open Services for Lifecycle Collaboration est une communauté qui travaille à définir des standards communs aux solutions d'ALM et de PPM en vue de favoriser l'interopérabilité des outils et l'échange de données
- **authentification CAS, OpenID, OAuth**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

Page 51

WWW.SMILE.FR

The screenshot shows the FusionForge forum interface. At the top, there's a navigation bar with 'Home', 'My Page', 'Projects', and 'FusionForge'. Below this, there's a 'Forum: help' section with options to 'Monitor Forum' or 'Start New Thread'. A table lists various forum threads with columns for 'Thread', 'Author', and 'Date'.

Thread	Author	Date
SVN & trunk	christophe lefevre	2012-04-10 14:27
install on ubuntu - hudson plugin	manuel dahmen	2012-04-06 07:49
RE: install on ubuntu - hudson plugin	manuel dahmen	2012-04-06 10:36
RE: install on ubuntu - hudson plugin	Alain Peyrat	2012-04-06 10:52
Upgrading MediaWiki	Kaustuv Royburman	2012-04-05 12:06
How do you relate a bug to a revision in svn?	Alejandro Villamarin	2012-04-04 07:36
RE: How do you relate a bug to a revision in svn?	Alain Peyrat	2012-04-04 08:58
RE: How do you relate a bug to a revision in svn?	Alejandro Villamarin	2012-04-04 09:20
RE: How do you relate a bug to a revision in svn?	Franck VILLAUME	2012-04-04 12:01
RE: How do you relate a bug to a revision in svn?	Alejandro Villamarin	2012-04-04 13:16
RE: How do you relate a bug to a revision in svn?	Franck VILLAUME	2012-04-04 14:28
GIT : installation help	Chanh TRAN	2010-11-04 15:09
RE: GIT : installation help	Coiby Xu	2012-03-31 03:44
FusionForge VMware solution has no apache running	Markus Liebelt	2012-03-28 12:39
RE: FusionForge VMware solution has no apache running	Alain Peyrat	2012-03-28 13:18
RE: FusionForge VMware solution has no apache running	Markus Liebelt	2012-03-28 13:21
RE: FusionForge VMware solution has no apache running	Alain Peyrat	2012-03-28 18:22

## Exemple de forums dans Fusionforge

**open  
source**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

## Redmine

Redmine (<http://www.redmine.org/>) est une application web de gestion complète de projet, développée en Ruby, sur la base du framework Ruby on Rails. L'outil a été initié en 2006 par Jean-Philippe Lang ; depuis d'autres développeurs de la communauté y contribuent.

Redmine était considéré initialement comme le cousin de Trac. Cette comparaison est aujourd'hui réductrice car Redmine a intégré nativement des fonctionnalités pour lesquelles il était nécessaire d'ajouter des extensions sous Trac.

Redmine peut être utilisé avec une base MySQL ou PostgreSQL, sous Linux, Mac ou Windows. Il est distribué sous licence GNU GPL, avec plus de 30 traductions proposées. La dernière version est Redmine 2.0.2, datant de Juin 2012.

Redmine est utilisé par plusieurs centres de recherche en France et à l'étranger tels que le Laboratoire de Recherche Informatique ou le CNRS, par des projets open source et des entreprises. Une partie de la communauté, regrettant le manque de disponibilité et de dynamique de l'équipe de développement, a souhaité continuer le développement de son côté. Il existe désormais un fork de Redmine dénommé Chiliproject.

### Périmètre fonctionnel :

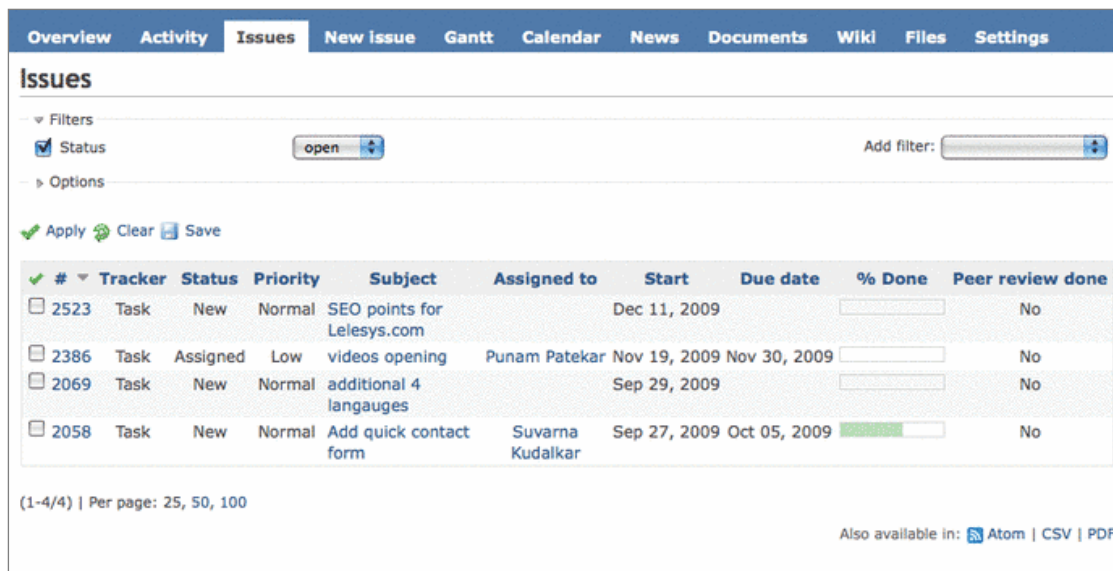


### Périmètre fonctionnel de Redmine

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

L'outil est intéressant par différents aspects. Nous avons sélectionné les suivants qui font la force de Redmine:

- **les 250 plugins proposés par la communauté** permettent de compléter la solution première qui ne couvre pas tous les besoins d'une bonne gestion des projets logiciels. La facilité d'installation des plugins permet de les tester et ainsi de sélectionner ceux qui répondent à vos besoins : agenda, messagerie instantanée, graphiques, gestion du budget, intégration continue, etc. Il faut évidemment vérifier leur stabilité et maintenance dans le temps, ainsi que leur compatibilité avec les prochaines versions de Redmine. Notez qu'il n'y aura pas forcément d'intégration avec les autres outils.
- **Annuaire des plugins Redmine** : <http://www.redmine.org/plugins>
- l'outil jouit d'une **popularité importante**. Le grand nombre de messages sur internet augmente les chances de trouver une réponse à ses questions.
- Ruby On Rails est un **environnement de développement récent** qui apporte le confort des outils modernes : MVC, REST, total objet, DRY, Système de plugins (gems), langage dynamique, etc. Il est donc **agréable de faire ses propres développements**



The screenshot shows the Redmine 'Issues' page. It features a navigation bar with tabs: Overview, Activity, Issues (selected), New Issue, Gantt, Calendar, News, Documents, Wiki, Files, and Settings. Below the navigation bar, there are filters (Status: open) and options (Apply, Clear, Save). The main content is a table of issues.

#	Tracker	Status	Priority	Subject	Assigned to	Start	Due date	% Done	Peer review done
2523	Task	New	Normal	SEO points for Lelesys.com		Dec 11, 2009			No
2386	Task	Assigned	Low	videos opening	Punam Patekar	Nov 19, 2009	Nov 30, 2009		No
2069	Task	New	Normal	additional 4 langauges		Sep 29, 2009			No
2058	Task	New	Normal	Add quick contact form	Suvarna Kudalkar	Sep 27, 2009	Oct 05, 2009		No

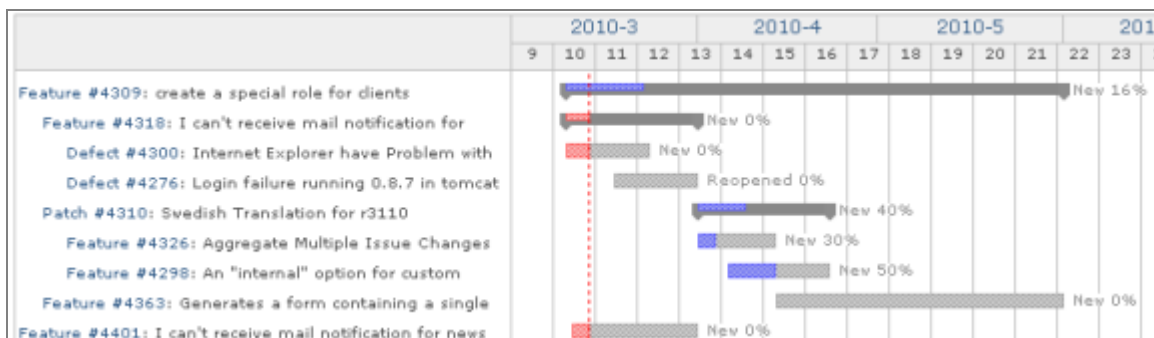
(1-4/4) | Per page: 25, 50, 100

Also available in: Atom | CSV | PDF

Exemple de tableau de tâches avec progression d'avancement



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”



Exemple de graphique de Gantt dans Redmine

All Projects » Developers »

### Spent time

▼ Date range

☒ all time

☐ From 2010-09-20 to 2010-09-24

✓ Apply

Details

Report

Details:

Days

Add:

Clear

Total: 5.67 hours

Issue	Activity	2010-09-20	2010-09-21	2010-09-22	2010-09-23	2010-09-24	Total
none			1.00				1.00
	Support		1.00				1.00
Bug #10: Bug Report from Test user				1.33	1.00	0.33	2.67
	Design			1.00			1.00
	Development				1.00		1.00
	Research			0.33			0.33
	Testing					0.33	0.33
Feature #11: Build a bridge					2.00		2.00
	Implementation				2.00		2.00
Total			1.00	1.33	3.00	0.33	5.67

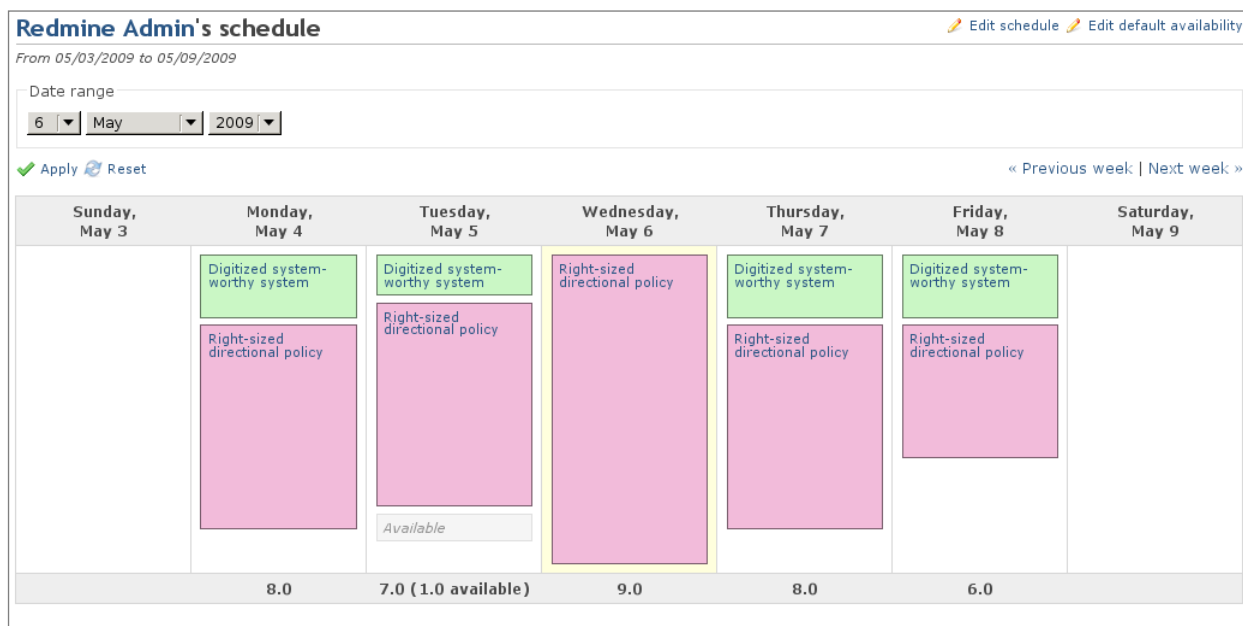
Also available in: CSV

Also available in: CSV

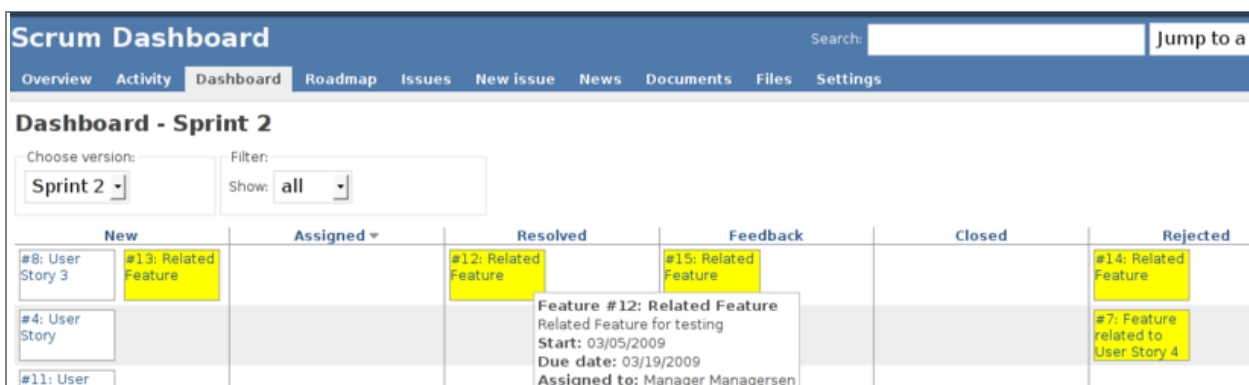
Exemple de rapport de temps passé dans Redmine

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

WWW.SMILE.FR



## Plugin pour la création de calendrier



## Plugin pour le tableau de post-it Scrum

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## Tuleap

Tuleap ([www.tuleap.com](http://www.tuleap.com)) est une suite logicielle de plusieurs modules intégrés. Tuleap est majoritairement développée en PHP et elle repose sur une architecture Linux Apache MySQL PHP. Pour l’installer, le serveur de l’entreprise doit être un CentOS5 ou Red Hat Enterprise Linux v 5. Comme il s’agit d’une application entièrement web, le système d’exploitation des utilisateurs n’importe pas. La dernière version majeure est Tuleap 5.0, sortie fin Avril 2012.

Tuleap est issue de l’externalisation du savoir-faire en ingénierie logicielle de STMicroelectronics, un des leaders de l’industrie des semi-conducteurs. Tuleap était un outil interne utilisé depuis 2004 par les équipes R&D de STMicroelectronics. En 2011, des responsables et ingénieurs de ST ont décidé de créer Enalean, une société dédiée au développement de Tuleap et sa maintenance.

Depuis, d’autres grandes entreprises industrielles sont devenues clientes de Tuleap, en plus de STMicroelectronics qui utilisent Tuleap dans 23 pays avec plus de 15 000 utilisateurs : Orange France Télécom avec plusieurs milliers d’utilisateurs ou Ericsson. Tuleap est principalement développé par Enalean, mais également par les clients et utilisateurs qui y contribuent.

La Suite est distribuée sous licence GPL. A l’heure d’aujourd’hui, c’est la seule solution ALM libre développée par un éditeur spécialisé en ingénierie logicielle et proposant une offre de services. Contrairement à ce qui est souvent fait par les éditeurs open-source, il n’existe qu’une seule version de Tuleap : clients et utilisateurs de la communauté utilisent et contribuent à une même version.

Tuleap est un outil puissant qui évolue vite. Pour en exploiter toute sa puissance, il faut prendre le temps de l’adapter à ses besoins.

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

## Périmètre fonctionnel :

WWW.SMILE.FR



Guide d'utilisation de Tuleap (en français) :

[https://tuleap.net/documentation/user\\_guide/html/fr\\_FR/User\\_Guide.html](https://tuleap.net/documentation/user_guide/html/fr_FR/User_Guide.html)

Guide d'utilisation de Tuleap (en anglais) :

[https://tuleap.net/documentation/user\\_guide/html/en\\_US/User\\_Guide.html](https://tuleap.net/documentation/user_guide/html/en_US/User_Guide.html)

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

Comparativement aux autres solutions, l’outil apporte des caractéristiques complémentaires intéressantes parmi lesquelles :

- bonne couverture fonctionnelle pour l’ensemble du processus de Gestion du cycle de vie des Applications
- forte traçabilité: par un système de références croisées, on peut lier entre eux les artefacts, documents, codes source, builds, messages dans un forum, dans un chat, etc.

WWW.SMILE.FR

The screenshot shows a 'Details' view of a bug report in Tuleap. An orange box labeled 'Création automatique de liens bi-directionnels' has arrows pointing to specific text elements in the 'Résumé' field: 'chat #20111013', 'le doc #68', 'bug #118', 'svn #3', and 'build #32'. These elements are also highlighted with orange boxes. The form includes fields for Organisation (ReccoMedical), Sévérité (2), Résumé (Hello World only works in console), Détails (La version de Hello World 5.2 ne fonctionne pas sur les environnements graphiques. Se référer à la discussion de messagerie instantanée chat #20111013 et le doc #68. Voir également bug #118, svn #3 et build #32), Priorité (Minor), Type (Incident), Catégorie (Other), and Version (4.0.18).

Illustration de la traçabilité dans Tuleap: extrait d’un artefact avec création automatique de liens bi-directionnels vers une discussion, un document, un bug, une révision et un build d’intégration  
continue

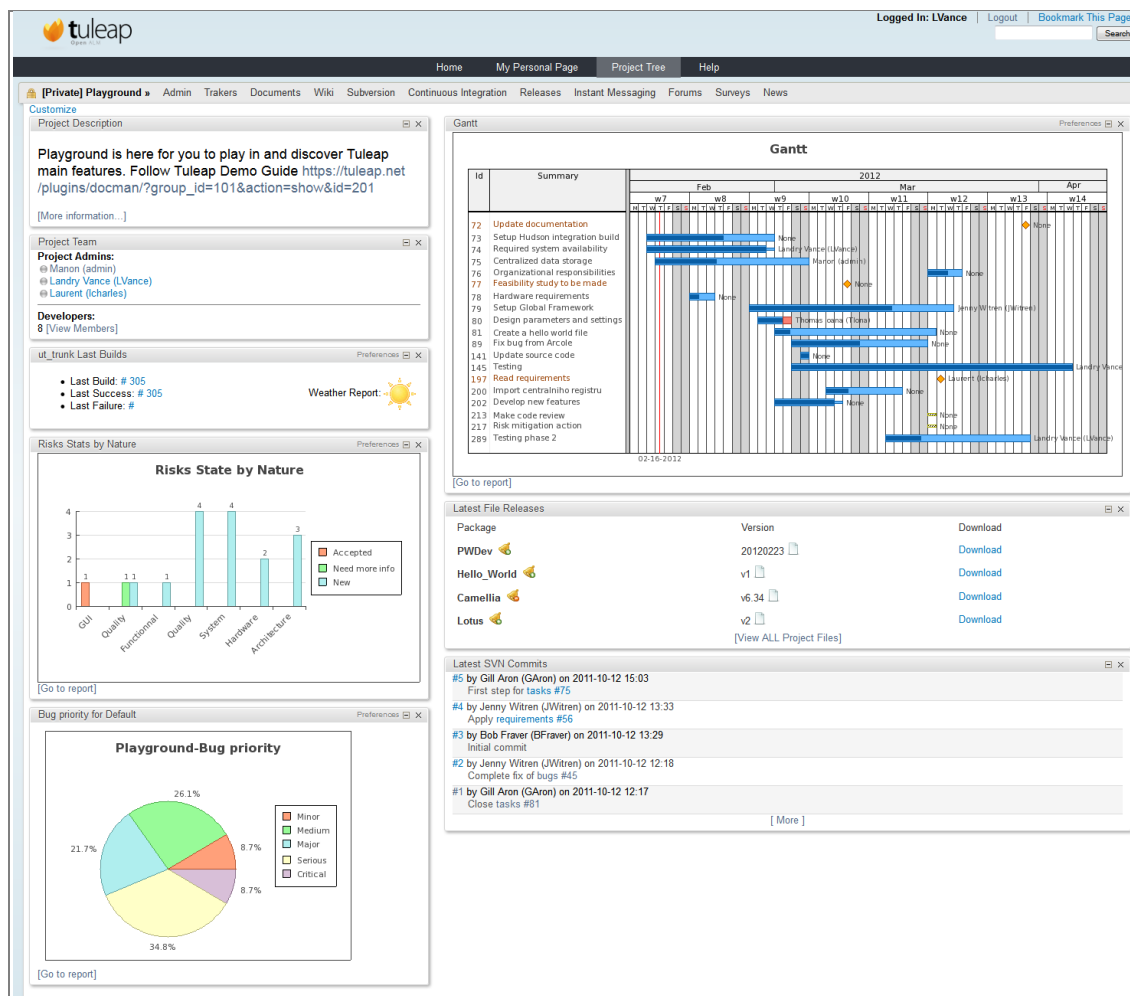


# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

Page 59

- les différents widgets des tableaux de bords, mis à jour en temps réel : derniers artefacts, graphiques, documents, forums, livrables, intégration continue, commits, actualités, etc.

WWW.SMILE.FR



**Exemple de tableau de bord personnel**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

Page 60

- **système de tracker nouvelle génération**, facilement **configurable** aux processus de l'entreprise, tous les champs sont personnalisables

WWW.SMILE.FR

The screenshot displays the Tuleap web application interface for configuring trackers. The top navigation bar includes links for Home, My Personal Page, Project Tree, and Help. The main content area is titled 'Manage Field Usage' and contains several sections for configuring fields and elements. The 'Fields' section lists various field types like String, Text, Selectbox, Multi Select Box, Date, File upload, Integer, Float, Open List, Artifact Link, Permissions on artifact, and Shared field. The 'Dynamic fields' section includes fields like Artifact ID, Last Update Date, Submitted By, Submitted On, and Cross References. The 'Structural elements' section includes Fieldset, Column, Line Break, Separator, and Static Text. The 'Unused elements' section lists References, Assigned to (multiple), Resolution, dependance, and End Date. The main configuration area shows a grid of fields for 'Identification' (Artifact ID, Product, Version, State, Type, Nature, Complexity, Priority) and a 'Summary' section with a 'Satisfy' field. The 'Historic' section shows fields for Author, Creation date, Last Modified On, and Assigned to. The 'Details' section is partially visible at the bottom.

## Configuration et personnalisation des trackers

**open**  
**source**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

- **intégration native** de tous les outils avec **gestion très fine des permissions** : contrôle des accès sur chaque outil, chaque document, chaque artefact, chaque champ des trackers, etc.
- **recherche d'artefacts multi-trackers** : dans un même projet, on peut faire des recherches dans plusieurs trackers à la fois, pour lister tous les artefacts répondant à une requête.

**Cross Trackers Search**

Title (Semantic)  Status (Semantic)  Component  Releases

id	Title (Semantic)	Status (Semantic)
requirement #443	1 – System Fields	Open
epic #578	Shared fields	
story #368	Select for mytracker a field from another tracker	Closed
story #373	Use the shared field in an artifact creation/update	Closed
task #369	Display shared field on artifact form	Closed
task #370	Record shared field value in db	Closed
task #371	Various verification	Closed
task #372	Mass Change and shared fields	Closed
story #380	Query tracker "B" with shared field	Closed
task #374	Shared field available in criteria	Closed
task #375	Shared field available in columns	Closed
task #376	Shared field available in graphs	Closed
task #377	Shared field in CSV export	Closed
task #378	Shared field available in aggregate functions	Closed
task #379	Shared field permissions	Closed
story #383	Manage shared fields XML export/import	Closed
task #381	On export, shared fields are "duplicated"	Closed
task #382	On import, no longer shared field	Closed
requirement #444	1.1 – Submit a new artifact	Open
requirement #445	2 – Query across Trackers	Open
requirement #446	2.1 – Hierarchical view	Open
epic #577	Hierarchical view	
story #478	have a hierarchical view to visualize my trackers	Closed
story #482	have some clarification about hirarchical view	Closed
story #513	Get search results with respect of hierarchy	Open
story #528	clarify "hierarchy" vs "association"	Closed
story #567	See the tracker hierarchy on tracker front page	Open
story #569	be able link artifacts bi-directionally as long as it respects the hierararchy defined by the Project Admin	Open

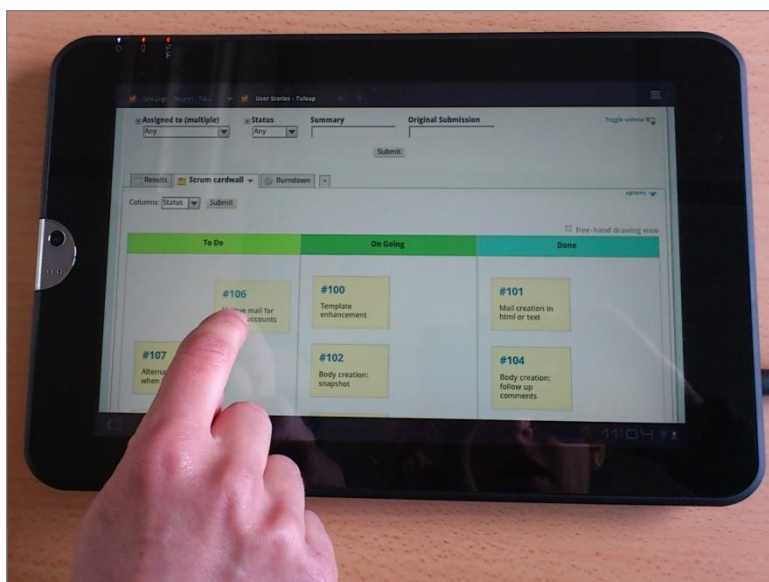
## Exemple de recherche multi-trackers

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

- **outils agiles intégrés:** burndown chart, cardwall de post-it avec cliquer-glisser, création d'un planning de version par sprint.



**Exemple de Burndown Chart (graphique de reste-à-faire en méthode agile)**



**Exemple de cardwall (tableau de post-it représentant des artefacts en méthode agile)**

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

- création **personnalisée** de **rapports et graphiques** actualisés en temps réel

**Monthly**

State: Tous, Aucun, Accepted, Rejected, Need more info, Satisfied, New  
 Product: MobileDevice  
 Nature: Tous  
 Complexity: Tous  
 Priority: Tous

Modifier les critères

Valider

Requirements Monthly report-Nature Risks New table +

18 artefact(s)

Cliquer sur une entête de colonne pour trier les résultats.

Modifier les colonnes

N°	Nature	Priority	Summary	Complexity	Cross references	State	Type
55	GUI	Medium	Screen layout constraints	Low	Bugs #7	Accepted	design
56	Quality	Medium	Anticipated error conditions or invalid inputs	Medium	requirements #61, tasks #81, tasks #77, svn #4	Need more info	design
57	Quality	Low	Interoperability	High	tasks #77	New	specification
58	System	Medium	Specify system requirements	Low	requirements #59, requirements #60, requirements #61, tests #82, tests #85, rev #23	New	test
59	System	Medium	Application programming interface protocols	Low	requirements #58	New	specification

Nombre (par groupe) :  
 NULL 1  
 High 3  
 Low 2  
 Medium 12

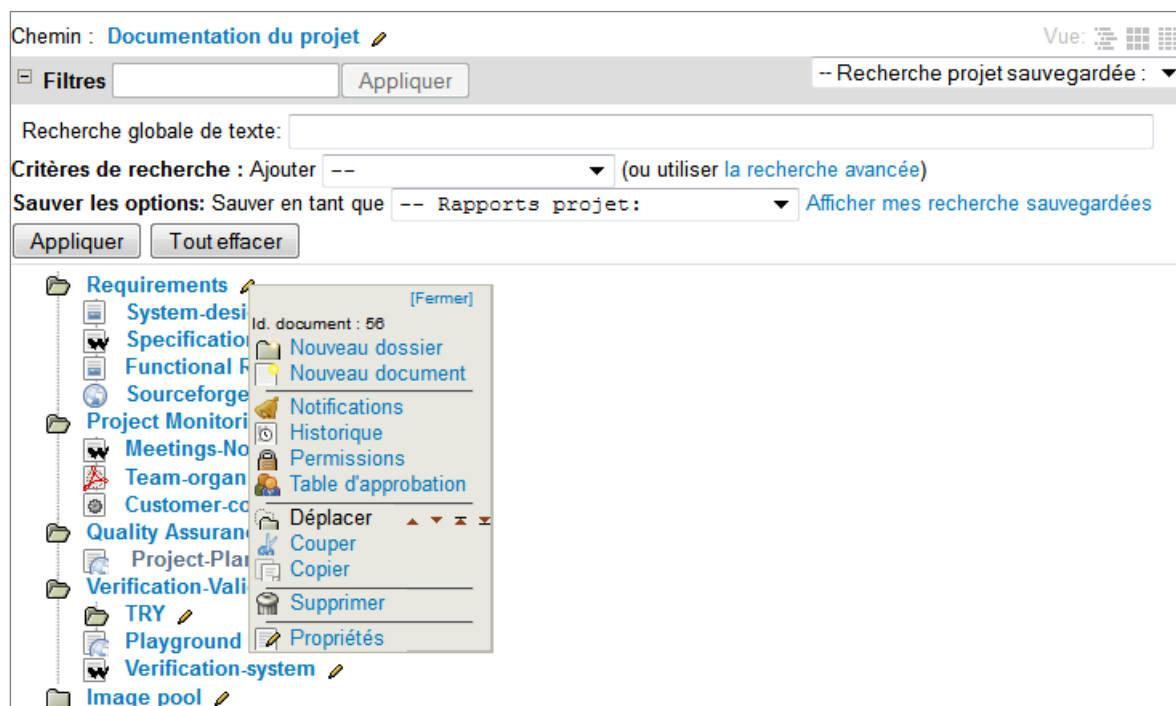
Nombre (par groupe) :  
 Accepted 1  
 Need more info 2  
 New 15

**Exemple de rapports tabulaires et zone de recherches personnalisable**



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

- un vrai mini gestionnaire électronique de documents avec gestion des différentes versions, workflow , gestion fine des permissions, recherche dans les documents

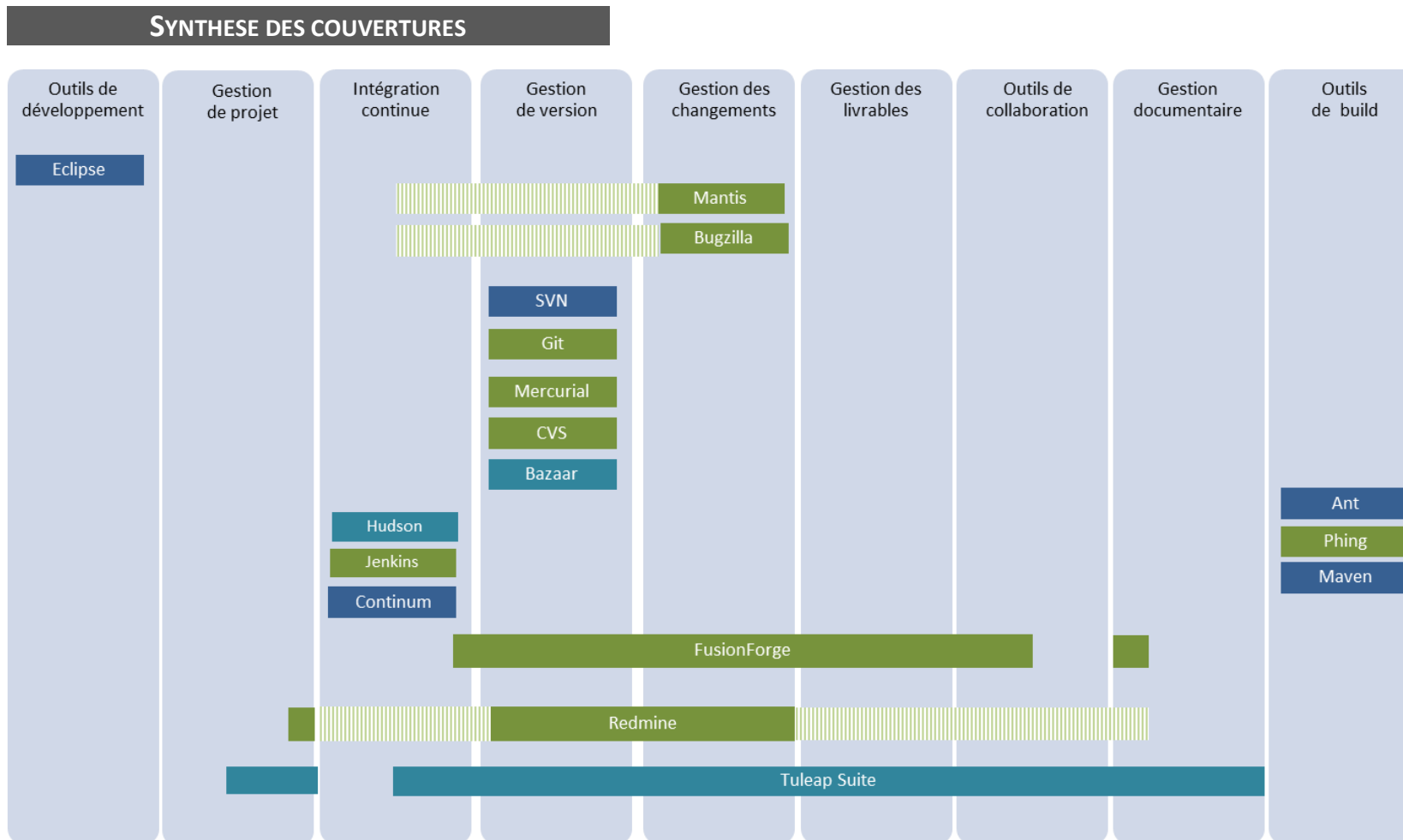


## Gestionnaire de document inclus dans Tuleap

- Au-delà de l’aspect fonctionnel, nous avons apprécié la documentation mise à jour à chaque version ainsi que la transparence de l’équipe sur les développements en cours et à venir : une fois par mois, une nouvelle version incrémentale est mise à disposition. Les discussions entre développeurs Tuleap et la communauté sont ouvertes.

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l'open source”

WWW.SMILE.FR



Outil libre propulsé  
par une communauté

Outil libre propulsé  
par une fondation

Outil libre propulsé  
par un éditeur



Il existe des extensions pour la majorité de ces outils. Elles peuvent étendre l'étendue fonctionnelle première. Avant déploiement, nous vous encourageons à les tester, à vérifier leur **stabilité** et **compatibilité** avec les prochaines versions.

## ANALYSE DES 3 SOLUTIONS INTEGREES

Nous synthétisons ci-dessous les forces et faiblesses de chacune d’elles selon les critères importants à prendre à compte lors de votre choix de solution ALM.

### CRITERES D’EVALUATION

#### Stabilité à grande échelle

Il s’agit ici d’évaluer dans quelle mesure la solution ALM supporte un grand nombre d’utilisateurs en même temps. Ceci est important lorsque votre solution ALM est amenée à être déployée de façon industrielle au sein de votre entreprise, sur un ou plusieurs sites.

De plus, on regarde aussi la cohérence des développements : les plugins sont-ils développés de façon décollée du « core » de l’outil (le cœur, le centre) ou en même temps ? Ceci apportera à l’entreprise une indication sur la stabilité et de pérennité de la forge.

#### Périmètre fonctionnel

Nous étudions ici la couverture fonctionnelle pour l’ensemble du processus de la gestion du cycle de vie des Applications. Nous regardons quels outils et quelles fonctionnalités sont proposées de façon **native** et **intégrée** dans la solution.

#### Mises à jour et Nouvelles versions

Sont considérés ici :

- le nombre et la qualité des patches correctifs
- la dynamique autour de la solution avec l’activité des commits
- la fréquence et régularité de nouvelles versions
- l’apport fonctionnel par les nouvelles versions

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

Page 67

## Qualité technique

Ceci englobe la cohérence de l’architecture, l’amélioration continue du code produit, les méthodes de développement.

## Ressources et Information disponibles

Il convient de prendre en compte tant la disponibilité de documentation, tutoriels, messages dans les forums que le nombre de développeurs réguliers à la solution.

## Réactivité aux demandes de support

Certes la popularité d’un logiciel est importante mais si vous n’obtenez pas de réponses lorsque vous avez une question d’usage ou que vous rencontrez une difficulté, cela apporte un vrai bémol. Nous avons donc regardé la rapidité et la qualité des réponses à nos questions.

## Souplesse-Extensibilité

Ici on met l’accent sur la modularité et la souplesse de la solution pour personnaliser les outils et les adapter aux besoins métiers, rajouter des plugins, faire ses propres développements et les intégrer au noyau.

## Notoriété

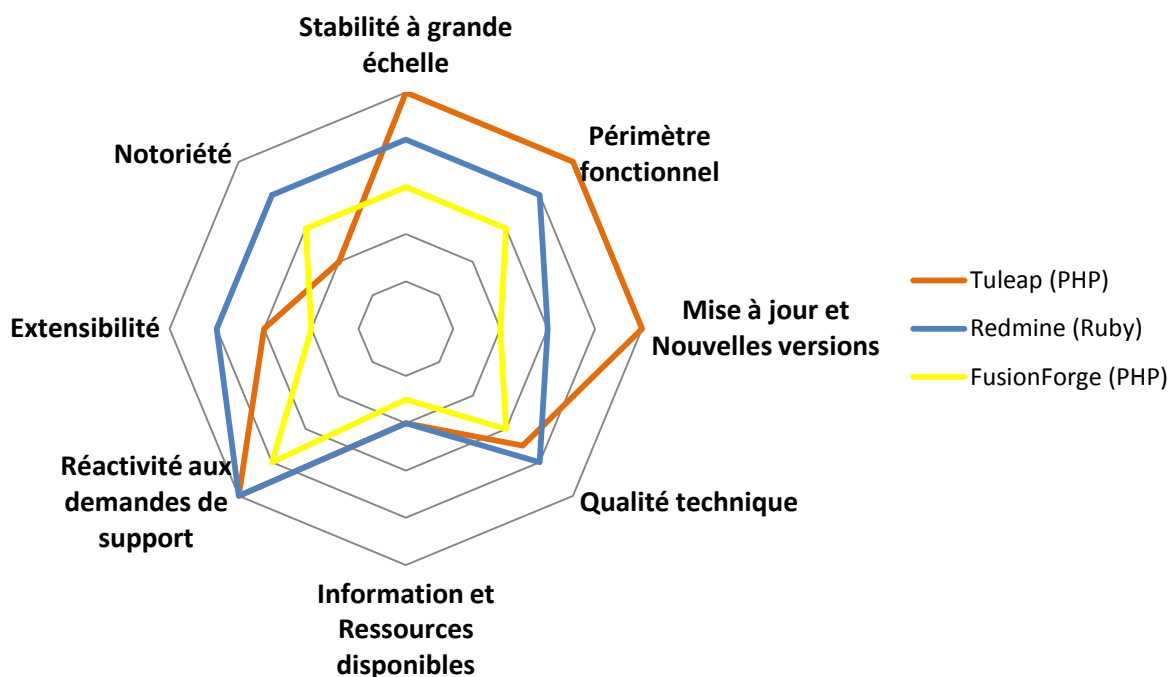
Nombre et importance des références clients, autres installations, sociétés (intégrateurs, SSLL, éditeurs) proposant de l’expertise autour de la solution, citations dans la presse professionnelle, etc. sont étudiés. Ce critère est à pondérer en fonction du niveau des autres puisque seul il ne peut garantir la qualité et la pérennité de l’application.

# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

Page 68

## SYNTHESE ET RESULTATS

Les résultats observés ont été repris ci-dessous sous la forme d’un graph radar :



Si nous analysons le radar ci-dessus et reprenons les critères pour faire une comparaison des outils, nous observons alors les points suivants :

### Notoriété

Redmine a une popularité importante qui fait son succès. Fusionforge, anciennement GForge, est connu depuis plusieurs années par les communautés. Tuleap est un produit mature mais encore peu connu. La Suite se forge petit à petit une notoriété car il convainc de par sa qualité professionnelle.





# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

Page 69

## Stabilité à grande échelle

Tuleap semble le mieux positionné puisqu’il est connu pour être utilisé par des dizaines de milliers d’utilisateurs en même temps et dans plusieurs pays.

Comme nous l’avons mentionné, le grand atout de Redmine est le grand nombre de plugins disponibles. Les entreprises qui souhaitent déployer cet outil à grande échelle et de façon industrielle doivent faire attention à sa sécurité et fiabilité. En effet, les plugins non intégrés peuvent être sources de failles de sécurité.

## Périmètre fonctionnel

Redmine, additionné à ses extensions, et Tuleap proposent le plus grand nombre de fonctionnalités et couvrent le plus largement le périmètre de la Gestion du cycle de vie des Applications. Néanmoins, les solutions ne proposent pas les mêmes outils, certains sont présents nativement dans Tuleap, sous forme de plugins pour Redmine. Fusionforge n’a pas beaucoup évolué fonctionnellement depuis son origine, ce qui le rend un peu plus en retard sur ce sujet mais il offre les fonctionnalités de base.

## Mise à jour et Nouvelles versions

Fusionforge et Redmine sont animés par des communautés donc par des volontaires qui veulent bien consacrer de leur temps, par opposition à Tuleap qui est impulsé par l’éditeur Enalean. Il en résulte forcément des différences de régularités de mises à jour et de nouvelles versions. Fusionforge publie en moyenne une nouvelle version tous les 4 mois, Redmine une fois par trimestre, Tuleap une version par mois.

## Qualité technique

Redmine est développé en ruby et son architecture est cohérente, avec de bons concepts. Ceci en fait un produit d’une bonne qualité technique. Par contre, les développements des extensions se font indépendamment du développement de Redmine.

L’équipe de développement de Tuleap travaille en Scrum et selon les nouvelles méthodes d’ingénierie logicielle qui prouvent leur pertinence : GIT, livraisons régulières, Test Driven Development, Emergent Design... Les nouveaux développements sont donc de bonne qualité et propres.



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

Page 70

## Information et Ressources disponibles

Fusionforge inclut de la documentation et un wiki d’aide intéressant. Le wiki documentaire de Redmine n’est pas souvent mis à jour mais en cherchant dans les forums on arrive à trouver des réponses. De plus, il existe plusieurs ouvrages explicatifs sur l’outil. Tuleap fournit un guide utilisateurs en français et anglais, actualisé à chaque nouvelle version.

## Réactivité aux demandes de support

Redmine, grâce à sa grande communauté, vous permet d’obtenir rapidement de l’aide, fournie par d’autres membres. Il faudra compter un peu plus de temps pour Fusionforge dont la réactivité est fluctuante. Pour Tuleap, vous obtiendrez des réponses rapides ; ce sont les développeurs qui répondent majoritairement aux questions.

## Extensibilité

Tous les outils proposent une architecture basée sur des standards ouverts et les plugins ainsi qu’une API. Fusionforge et Tuleap offrent une API SOAP. En complément, Fusionforge travaille aux standards OSLC. Redmine fournit une API REST et un très grand nombre de plugins proposés par la communauté.

WWW.SMILE.FR

**open**  
**source**



# “Gestion du Cycle de vie des Applications (ALM) : le meilleur de l’open source”

Page 71

## SYNTHESE

*« L’offre open source est particulièrement large avec des produits de grande qualité aux socles techniques et fonctionnels parmi les meilleurs du marché logiciel »*

C’est une phrase que l’on répète souvent dans nos conclusions. Elle s’applique tellement bien à des marchés tels les CMS, la GED, l’ECommerce, le Système, etc.

En ce qui concerne l’ALM, on peut également dire la même chose ; et sans aucun scrupule. Certes c’est un marché relativement jeune ; mais où l’offre est prometteuse et pleine d’avenir.

Au-delà des composants ALM tels que Eclipse, Subversion, Git, Mantis, ... on trouve également de véritables suites telles que FusionForge, Redmine ou Tuleap qui ont fait leurs preuves dans des contextes ambitieux.

Car c’est bien là la particularité de l’offre ALM open source. Ce sont d’abord de grands groupes tels que Orange, STMicroelectronics, CNRS, etc. qui les ont adoptés (et fait évoluer pour certains) ; et désormais ces nouveaux outils se répandent sur tous secteurs et toutes tailles d’entreprise.

Il faut dire que le périmètre fonctionnel et technique est déjà très important comme on a pu le voir à travers ce livre blanc : gestion des exigences, gestion de version, gestion de projet, gestion de la qualité, gestion des livrables, gestion documentaire, collaboration sociale, ... Le ROI direct apporté par ce type de solution n’est pas neutre, loin de là.

Et puis les bénéfices à l’utilisation de l’open source sont tellement énormes : pérennité, liberté de choix, respect des standards, dynamique d’évolution, standard de fait, coût, etc.

Ce livre, nous l’espérons, vous permettra de converger vers les meilleurs produits et vous aidera à bâtir des architectures sûres, pérennes et innovantes.

WWW.SMILE.FR

**open**  
**source**