



Guide

Déployer un outil ALM

Les bénéfices, les pièges à éviter et le ROI à attendre



Sommaire

1. Introduction	03
2. Qu'est-ce que l'Application Lifecycle Management (ALM) ?	04
3. ALM et conformité : un enjeu industriel	06
4. Les fonctionnalités clés d'un ALM	08
5. Déployer un ALM : étapes et bonnes pratiques	10
6. ALM et PLM : convergence pour le développement produit dans l'industrie	12
7. Checklist : êtes-vous prêt pour l'ALM agile ?	14
8. Conclusion	17

Introduction

À l'heure où la transformation digitale s'accélère, la capacité à concevoir, livrer et maintenir des applications logicielles de qualité est devenue un levier stratégique. Les attentes clients évoluent, les cycles produits raccourcissent, et la pression réglementaire s'intensifie, notamment dans les secteurs comme l'aéronautique, l'automobile, le médical ou la défense.

Dans ce contexte, l'ALM (Application Lifecycle Management) s'impose comme un pilier de performance et d'innovation. Il ne se limite plus à la gestion de projets, mais couvre l'ensemble du cycle de vie logiciel : conception, développement, validation, déploiement et maintenance. Cette approche intégrée vise à garantir qualité, traçabilité, conformité et agilité.

Les défis sont nombreux : aligner exigences métiers et développement, assurer traçabilité et conformité, favoriser la collaboration d'équipes dispersées, ou encore livrer plus vite tout en maîtrisant risques et coûts.

Les solutions ALM répondent à ces enjeux en centralisant les activités du cycle de vie logiciel, en automatisant les processus critiques et en facilitant la collaboration. Elles s'adaptent aux méthodes agiles tout en respectant les contraintes des secteurs réglementés.

Ce guide vous aidera à comprendre les enjeux de l'ALM, à en mesurer les bénéfices, à identifier les fonctionnalités clés d'une solution, et à réussir son déploiement. Vous y trouverez aussi une checklist d'évaluation et un éclairage sur la convergence ALM/PLM, moteur de la R&D industrielle.

Destiné aux décideurs, chefs de projet, responsables qualité ou développeurs, ce guide vous accompagnera dans votre démarche ALM, au service de la transformation et de la compétitivité de votre entreprise.



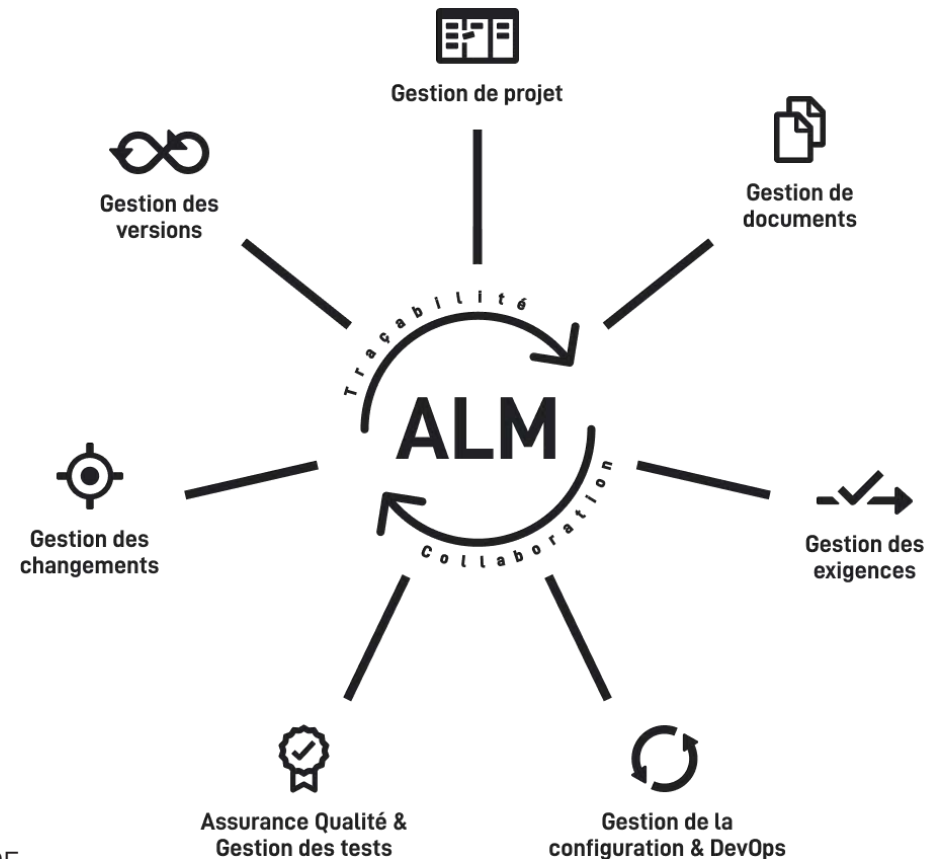
Qu'est-ce que l'Application Lifecycle Management (ALM) ?

L'Application Lifecycle Management est l'ensemble des processus, outils et méthodes qui permettent de gérer le cycle de vie d'une application, de sa conception initiale à sa mise hors service.

Il regroupe toutes les phases suivantes :

- La capture et la gestion des exigences
- La planification et la gestion de projet
- Le développement logiciel
- Les tests
- Le déploiement
- La maintenance et l'évolution

Contrairement au SDLC (Software Development Life Cycle), centré sur le développement pur, l'ALM inclut aussi les processus de gouvernance, de conformité, et de collaboration entre toutes les parties prenantes : MOA, MOE, qualité, dév, tests, etc.



Pourquoi un ALM ?



Un outil ALM comme Tuleap répond aux besoins actuels des entreprises :

- Centralisation des informations et des flux de travail
- Adaptabilité aux méthodes agiles
- Réduction du shadow IT et des outils dispersés
- Traçabilité pour la conformité et les audits
- Collaboration renforcée et visibilité accrue sur les projets

Des entreprises comme Dassault Systèmes ou STMicroelectronics ont choisi Tuleap pour sa capacité à soutenir l'agilité à l'échelle, la conformité et l'innovation, tout en favorisant la convergence entre équipes hétérogènes, telles que hardware et software.



STMicroelectronics gère le cycle de développement de logiciels complexes avec **Tuleap**



Tuleap et Dassault Systèmes : une collaboration entre ALM-PLM pour révolutionner le développement produit

ALM et conformité : un enjeu industriel

Dans de nombreux secteurs industriels avec des systèmes embarqués – tels que l'aéronautique, l'automobile, le médical, la défense – la conformité réglementaire et la qualité sont des impératifs majeurs. L'ALM (Application Lifecycle Management) joue un rôle central pour répondre à ces exigences strictes, en offrant une traçabilité complète et en facilitant les audits.

Les défis de la conformité

Multiplicité des normes :

ISO 9001, ISO 26262, DO-178C, IEC 62304, CMMI, SPICE, FDA, etc.

Chaque secteur impose ses propres référentiels, qui exigent une documentation rigoureuse, la gestion des exigences et la traçabilité des modifications.

Complexité des processus :

Les projets industriels impliquent souvent de nombreuses parties prenantes, des cycles longs et des processus complexes, où chaque étape doit être justifiée et vérifiable.

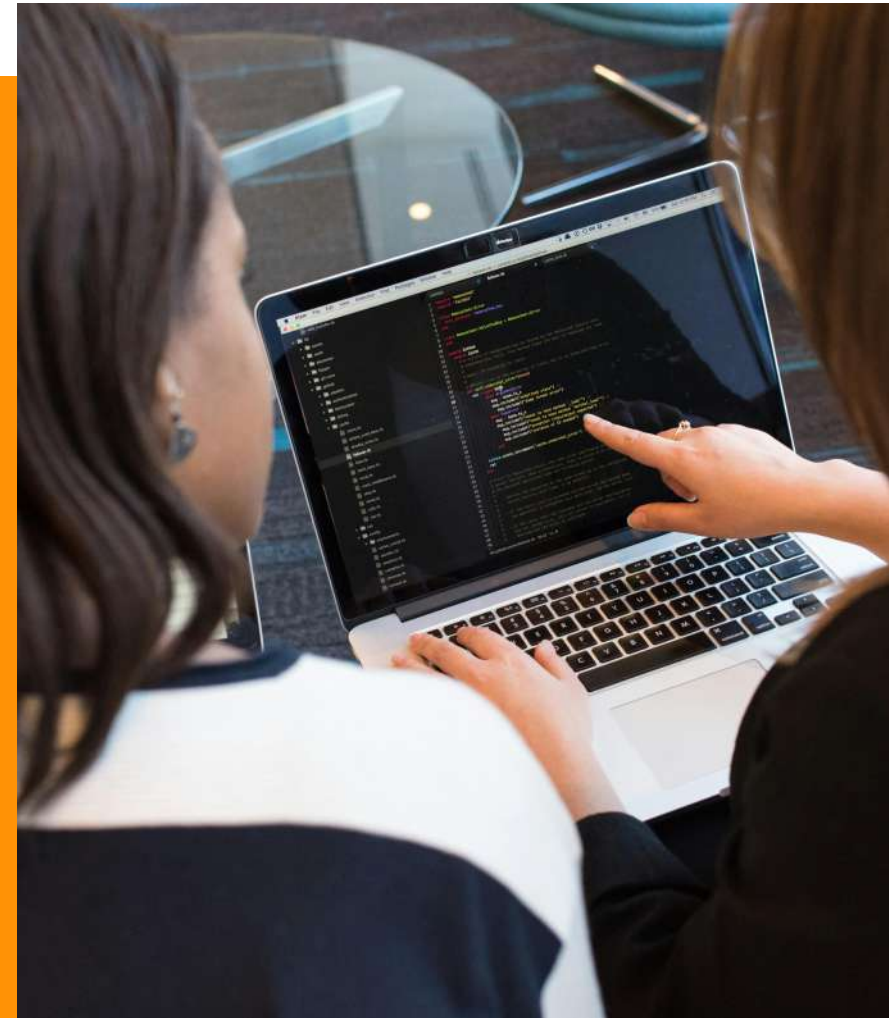
Pression des audits :

Les entreprises doivent prouver à tout moment que leurs processus sont maîtrisés et que chaque exigence a été prise en compte, testée et validée.



Un outil ALM permet de :

- Centraliser et tracer toutes les exigences, modifications, validations et livraisons dans un référentiel unique.
- Automatiser la génération des rapports nécessaires pour les audits, en fournissant des preuves de conformité à chaque étape du cycle de vie.
- Relier les exigences aux tests et aux livrables, pour garantir que chaque besoin métier ou réglementaire est bien couvert et validé.
- Gérer les workflows de validation (relectures, approbations, signatures électroniques) afin de respecter les processus qualité internes et externes.
- Réduire les risques d'erreurs ou d'oublis grâce à une visibilité complète sur l'avancement et la couverture des exigences.



Les fonctionnalités clés d'un ALM

Comme évoqué précédemment, un outil ALM centralise et structure l'ensemble des processus nécessaires à la gestion du cycle de vie d'une application, de l'idée initiale à la maintenance. Voici les fonctionnalités essentielles à rechercher dans une solution ALM :

Gestion des exigences

Centralisation, formalisation et suivi des besoins métiers et techniques. Possibilité de relier chaque exigence à des critères d'acceptation, des tests et des livrables pour assurer la traçabilité complète.

Planification et gestion de projet

Outils pour planifier les sprints (Agile) ou les jalons (cycle en V), affecter les tâches, suivre l'avancement, visualiser la charge et anticiper les risques grâce à des diagrammes de Gantt ou des tableaux Kanban.

Suivi des activités et gestion des tickets

Système de suivi des tâches, bugs, incidents et demandes d'évolution, permettant de centraliser toutes les informations et de garantir l'enregistrement et la traçabilité des opérations

Gestion du code source et contrôle des versions

Intégration native ou interconnexion avec les outils de gestion de versions (Git, SVN, etc.) et permettant d'historiser chaque modification pour une traçabilité de bout en bout.

Gestion des tests et de la qualité

Modules pour concevoir, exécuter et suivre les tests (unitaires, fonctionnels, d'intégration...), automatiser les campagnes de tests et relier les résultats aux exigences.

Gestion des changements et de la configuration

Suivi des évolutions, gestion des versions applicatives, des configurations et des livrables pour garantir la cohérence et la conformité du produit tout au long du cycle de vie.





Tableaux de bord et reporting

Tableaux de bord personnalisables pour visualiser en temps réel l'état d'avancement, la couverture des exigences, la qualité et les risques du projet. Génération de rapports pour faciliter le pilotage et les audits.

Collaboration et communication

Espaces de travail partagés, gestion documentaire, notifications, commentaires et intégration avec les outils collaboratifs pour fluidifier les échanges entre toutes les parties prenantes.

Automatisation et intégration

Intégration avec les outils DevOps (CI/CD, gestion des builds, déploiement automatisé), interopérabilité avec d'autres systèmes (PLM, CRM, ERP...) pour un écosystème logiciel cohérent et efficace.

Ces fonctionnalités permettent d'industrialiser la gestion des projets logiciels, d'améliorer la collaboration, de garantir la qualité et la conformité, et d'accélérer la mise sur le marché des applications.

Déployer un ALM : étapes et bonnes pratiques

Déployer une solution ALM dans une organisation ne se limite pas à installer un outil. C'est un projet structurant qui implique la refonte des processus, l'alignement des équipes et l'adoption de bonnes pratiques. Voici les étapes et recommandations pour réussir cette démarche :

Définir les objectifs et le périmètre

1

- Clarifiez les attentes: quels sont les enjeux (qualité, conformité, agilité, traçabilité...)?
- Identifiez les processus à couvrir et les équipes concernées

2

Recueillir et formaliser les exigences

- Rassemblez les besoins métiers et techniques.
- Documentez les exigences fonctionnelles et non fonctionnelles de l'outil ALM à déployer

3

Concevoir l'architecture et les workflows

- Définissez l'architecture cible: intégration avec les outils existants, choix du mode (Cloud, On-premise, hybride).
- Paramétrez les workflows (gestion des exigences, des tâches, des tests, des validations) selon vos processus internes.

4

Développer, configurer et intégrer

- Configurez l'outil ALM et intégrez-le à l'écosystème existant (gestion de code source, CI/CD, PLM, etc.).
- Développez des connecteurs ou des automatisations si nécessaire

Tester et valider

5

- Menez des tests fonctionnels et d'intégration pour valider la conformité aux besoins.
- Impliquez les utilisateurs clés dans la phase de recette.

6

Déployer et accompagner le changement

- Déployez progressivement (par équipe, projet pilote, puis généralisation).
- Formez les utilisateurs, accompagnez le changement et récoltez les retours pour ajuster les processus.

7

Piloter et améliorer en continu

- Mettez en place des indicateurs de suivi (KPI, tableaux de bord).
- Organisez des rétrospectives régulières pour améliorer les pratiques et l'outil.

Bonnes pratiques clés :

- Adoptez une méthodologie agile pour permettre l'adaptation continue.
- Centralisez la gestion des exigences, tests, tâches et livrables.
- Automatisez les processus critiques (tests, déploiement, reporting).
- Favorisez la collaboration interdisciplinaire et la transparence.
- Sécurisez les accès et les données sensibles.

ALM et PLM : convergence pour le développement produit dans l'industrie

Dans la R&D industrielle, la frontière entre logiciel et matériel s'estompe : les produits sont de plus en plus pilotés par logiciel, rendant indispensable la convergence entre ALM (gestion du cycle de vie des applications) et PLM (gestion du cycle de vie des produits physiques).

Définitions :

ALM: Gère le développement, les tests, le déploiement et la maintenance des applications logicielles.

PLM: Gère la conception, la fabrication, l'utilisation et la fin de vie des produits physiques.





Pourquoi intégrer ALM et PLM ?

- Collaboration renforcée: Aligne les équipes hardware et software, réduit les silos et favorise la gestion unifiée des projets complexes.
- Traçabilité de bout en bout: Permet de relier exigences, modifications et tests logiciels aux composants matériels, garantissant la conformité réglementaire et la qualité produit.
- Innovation accélérée: Synchronise les cycles de développement, réduit les délais de mise sur le marché et facilite la gestion des changements dans des environnements multidisciplinaires.
- Gestion des impacts: Évalue plus facilement l'impact des évolutions logicielles sur les produits physiques et inversement.

Checklist : êtes-vous prêt pour l'ALM agile ?

Utilisez cette checklist pour évaluer la maturité de votre organisation et sa capacité à adopter efficacement l'ALM agile. Cochez chaque point qui correspond à votre situation actuelle.

1. Gouvernance & Organisation

- ☐ Les rôles et responsabilités sont clairement définis pour chaque étape du cycle de vie logiciel.
- ☐ Les processus de gestion des exigences, des changements et des versions sont formalisés et connus de tous.
- ☐ La gestion des accès, des droits utilisateurs et la traçabilité des actions sont assurées.

2. Gestion des exigences

- ☐ Les exigences sont documentées, analysées, priorisées et suivies tout au long du projet.
- ☐ Les parties prenantes collaborent activement à la définition et à l'évolution des besoins.
- ☐ Les exigences sont reliées aux tests, aux tâches de développement et aux livrables.

3. Collaboration & Communication

- ☐ Les équipes disposent d'outils favorisant la collaboration en temps réel et la transparence.
- ☐ Les informations projet (avancement, décisions, blocages) sont partagées et accessibles à tous.
- ☐ Les réunions de suivi (daily, revues, rétrospectives) sont régulières et efficaces.

4. Développement & Intégration

- ☐ Le développement suit des principes agiles : itérations courtes, feedbacks fréquents, adaptation continue.
- ☐ L'intégration continue et la gestion de configuration logicielle sont en place (CI, SCM).
- ☐ Les équipes sont capables de livrer des incréments fonctionnels à chaque itération.

5. Qualité & Tests

- ☐ Les tests (unitaires, intégration, système, acceptation) sont automatisés et intégrés au cycle de développement.
- ☐ Les critères d'acceptation sont définis pour chaque exigence ou user story.
- ☐ Les anomalies sont tracées, priorisées et résolues rapidement.

6. Déploiement & Maintenance

- ☐ Les processus de déploiement sont automatisés et reproductibles.
- ☐ Les procédures de rollback sont définies en cas d'échec de mise en production.
- ☐ La maintenance corrective et évolutive est organisée, avec un suivi des incidents et des demandes d'évolution.

7. Pilotage & Reporting

- ☐ Des tableaux de bord permettent de suivre l'avancement, la qualité et les risques projet en temps réel.
- ☐ Les indicateurs clés (KPIs) sont définis et suivis pour piloter la performance de l'équipe et du projet.
- ☐ Les audits internes et externes sont facilités grâce à la traçabilité et à la centralisation des informations.

8. Amélioration continue

- ☐ Des rétrospectives sont menées à la fin de chaque itération pour identifier les axes d'amélioration.
- ☐ Les processus sont régulièrement ajustés en fonction des retours d'expérience et des évolutions du contexte.
- ☐ L'organisation favorise la montée en compétences et la formation aux pratiques agiles et ALM.

Interprétation des résultats :

Plus de 20 cases cochées

Votre organisation est prête à adopter ou renforcer l'ALM agile.

Entre 12 et 20 cases cochées

De bonnes bases, mais des axes d'amélioration existent.

Moins de 12 cases cochées

Un accompagnement et une montée en maturité sont recommandés avant de déployer un ALM agile à grande échelle.

Conclusion

L'ALM s'impose aujourd'hui comme un levier stratégique pour toutes les organisations qui souhaitent accélérer leur transformation digitale, garantir la qualité de leurs produits et répondre aux exigences de conformité, en particulier dans les secteurs industriels. Bien plus qu'un simple outil de gestion de projet, l'ALM permet d'aligner les équipes, de centraliser l'information, d'automatiser les processus et de renforcer la collaboration à chaque étape du cycle de vie logiciel.

Adopter une solution ALM, intégrée et agile, c'est offrir à ses équipes la capacité de **s'adapter rapidement aux changements**, d'**innover** en continu et de **livrer** des applications fiables, traçables et conformes, tout en maîtrisant les risques et les coûts. L'intégration croissante entre ALM et PLM ouvre la voie à une gestion unifiée des projets complexes, indispensable à l'ère de l'industrie 4.0.

La réussite d'un projet ALM repose sur une démarche structurée : définition des objectifs, implication des parties prenantes, choix d'une solution adaptée, accompagnement du changement et amélioration continue.

Grâce à ce guide, vous disposez des clés pour évaluer votre maturité, identifier les bonnes pratiques et réussir la mise en place d'un ALM agile, moteur de performance et d'innovation pour votre organisation.

N'attendez plus pour franchir le cap : l'ALM est un atout décisif pour relever les défis d'aujourd'hui et de demain.



Nous contacter

Demandez une démo



Déployer un outil ALM :

Les bénéfices, les pièges à éviter et le ROI à attendre