



Guide

Deploying an ALM Tool

Benefits, Pitfalls to Avoid,
and Expected ROI



Summary

1. Introduction	03
2. What is Application Lifecycle Management (ALM)?	04
3. ALM and compliance: An industrial challenge	06
4. Key features of an ALM	08
5. Deploying an ALM: Steps and best practices	10
6. ALM and PLM: Convergence for product development in industry	12
7. Checklist: Are you ready for agile ALM?	14
8. Conclusion	17

Introduction

As digital transformation accelerates, the ability to design, deliver, and maintain high-quality software applications has become a strategic driver. Customer expectations are evolving, product cycles are shortening, and regulatory pressure is increasing, particularly in sectors such as aerospace, automotive, medical, and defense.

In this context, Application Lifecycle Management (ALM) has become a cornerstone of performance and innovation. It is no longer limited to project management but now covers the entire software lifecycle: design, development, validation, deployment, and maintenance. This integrated approach aims to ensure quality, traceability, compliance, and agility.

The challenges are significant: aligning business requirements with development, ensuring traceability and regulatory compliance, enabling collaboration across distributed teams, and delivering faster while controlling risks and costs.

ALM solutions address these challenges by centralizing software lifecycle activities, automating critical processes, and facilitating collaboration. They are fully compatible with agile methodologies while meeting the strict requirements of regulated industries.

This guide will help you understand the key issues of ALM, assess its benefits, identify the essential features of an ALM solution, and successfully deploy it. You will also find an evaluation checklist and insights on ALM/PLM convergence, a driving force for industrial R&D.

Intended for decision-makers, project managers, quality managers, and developers, this guide will support you in your ALM journey, serving the transformation and competitiveness of your company.



What is Application Lifecycle Management?

Application Lifecycle Management encompasses all the processes, tools, and methods used to manage the entire lifecycle of an application, from its initial design to its decommissioning.

It includes all the following phases:

- Requirements capture and management
- Project planning and management
- Software development
- Testing
- Deployment
- Maintenance and evolution

Unlike the SDLC (Software Development Life Cycle), which focuses solely on software development, ALM also incorporates governance processes, compliance management, and collaboration among all stakeholders: business owners, project managers, quality assurance, developers, testers, etc.



Why an ALM?



An ALM tool like Tuleap addresses the current needs of companies by providing:

- Centralization of information and workflows
- Adaptability to agile methodologies
- Reduction of shadow IT and scattered tools
- Traceability for compliance and audits
- Enhanced collaboration and increased project visibility

Companies such as Dassault Systèmes and STMicroelectronics have chosen Tuleap for its ability to support agility at scale, ensure compliance, foster innovation, and promote convergence between diverse teams, such as hardware and software.



STMicroelectronics gains full control over complex software systems application lifecycle



Tuleap and Dassault Systèmes
A collaboration between ALM and PLM to revolutionize product development

ALM and compliance: An industrial challenge

In many industrial sectors with embedded systems - such as aerospace, automotive, medical, and defense - regulatory compliance and quality are critical imperatives. Application Lifecycle Management (ALM) plays a central role in meeting these stringent requirements by providing complete traceability and facilitating audits.

The challenges of compliance

Multiple standards:

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

Each sector imposes its own frameworks, requiring rigorous documentation, thorough requirements management, and traceability of changes.

Process complexity:

Industrial projects often involve many stakeholders, long development cycles, and complex processes where each step must be justified and verifiable.

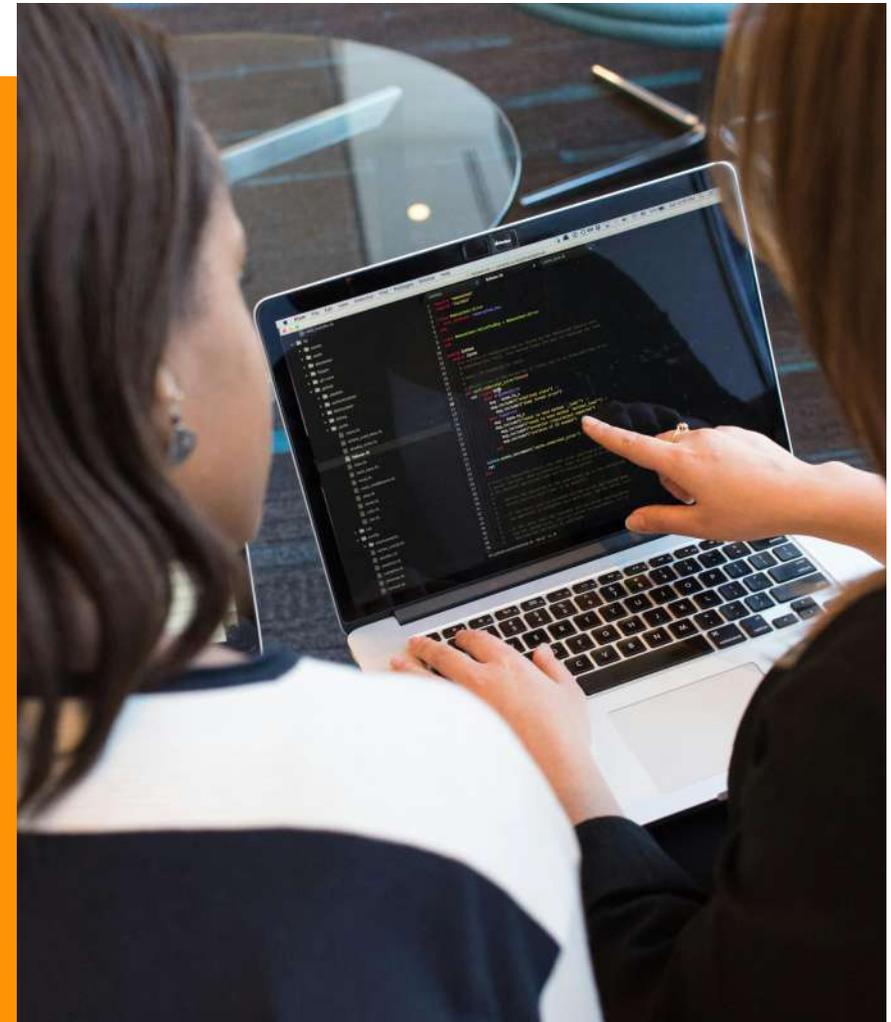
Audit pressure:

Companies must be able to demonstrate at any time that their processes are under control and that every requirement has been addressed, tested, and validated.



An ALM tool makes it possible to:

- Centralize and track all requirements, changes, validations, and deliveries in a single, unified repository.
- Automate the generation of reports required for audits, providing evidence of compliance at every stage of the lifecycle.
- Link requirements to tests and deliverables, ensuring that every business or regulatory need is fully covered and validated.
- Manage validation workflows (reviews, approvals, electronic signatures) to comply with both internal and external quality processes.
- Reduce the risk of errors or omissions thanks to complete visibility over progress and requirement coverage.



Key features of an ALM

As mentioned earlier, an ALM tool centralizes and structures all the processes required to manage the lifecycle of an application, from the initial idea to maintenance. Here are the essential features to look for in an ALM solution:

Requirements management

Centralization, formalization, and tracking of business and technical needs. The ability to link each requirement to acceptance criteria, tests, and deliverables to ensure complete traceability.

Project planning and management

Tools to plan sprints (Agile) or milestones (V-cycle), assign tasks, monitor progress, visualize workload, and anticipate risks using Gantt charts or Kanban boards.

Activity tracking and ticket management

A system to track tasks, bugs, incidents, and change requests, allowing centralized information management and ensuring that all operations are recorded and traceable.

Source code management and version control

Native integration or connectivity with version control tools (Git, SVN, etc.), enabling the history of every modification for end-to-end traceability.

Test and quality management

Modules to design, execute, and track tests (unit, functional, integration, etc.), automate test campaigns, and link results to requirements.

Change and configuration management

Tracking changes, managing application versions, configurations, and deliverables to ensure product consistency and compliance throughout the lifecycle.





Dashboards and reporting

Customizable dashboards to visualize, in real time, project progress, requirements coverage, quality, and risk levels. Report generation to facilitate project oversight and audits.

Collaboration and communication

Shared workspaces, document management, notifications, comments, and integration with collaborative tools to streamline communication among all stakeholders.

Automation and integration

Integration with DevOps tools (CI/CD, build management, automated deployment), and interoperability with other systems (PLM, CRM, ERP...) to create a coherent and efficient software ecosystem.

These features help industrialize software project management, enhance collaboration, ensure quality and compliance, and accelerate time-to-market for applications.

Deploying an ALM: Steps and best practices

Deploying an ALM solution within an organization goes beyond simply installing a tool. It is a transformative project that involves redesigning processes, aligning teams, and adopting best practices. Here are the key steps and recommendations for a successful implementation:

1 Define objectives and scope

- Clarify expectations: what are the main goals (quality, compliance, agility, traceability...)?
- Identify the processes to be covered and the teams involved.

2

Gather and formalize requirements

- Collect both business and technical needs.
- Document the functional and non-functional requirements of the ALM tool to be deployed.

3

Design the architecture and workflows

- Define the target architecture: integration with existing tools, choice of deployment model (Cloud, On-premises, Hybrid).
- Configure workflows (requirements management, task tracking, testing, validation) according to your internal processes.

4

Develop, configure, and integrate

- Set up the ALM tool and integrate it into the existing ecosystem (source code management, CI/CD, PLM, etc.).
- Develop connectors or automation scripts if needed.

Test and validate

5

- Perform functional and integration testing to ensure compliance with requirements.
- Involve key users during the acceptance testing phase.

6

Deploy and Support Change Management

- Roll out progressively (by team, pilot project, then full deployment).
- Train users, support change adoption, and gather feedback to fine-tune processes.

7

Monitor and continuously improve

- Establish tracking indicators (KPIs, dashboards).
- Hold regular retrospectives to improve both practices and the tool.

Key best practices:

- Adopt an agile methodology to enable continuous adaptation.
- Centralize the management of requirements, tests, tasks, and deliverables.
- Automate critical processes (testing, deployment, reporting).
- Promote cross-functional collaboration and transparency.
- Secure access and sensitive data.

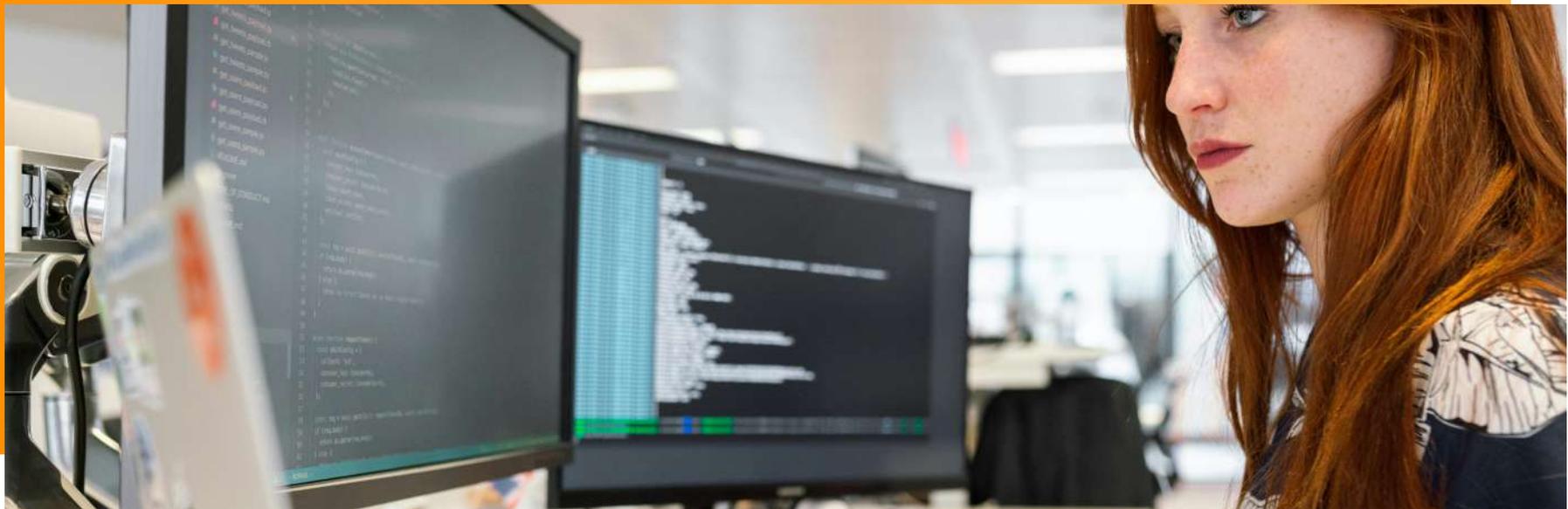
ALM and PLM: Convergence for product development in industry

In industrial R&D, the line between software and hardware is increasingly blurred: products are more and more software-driven, making the convergence of ALM (Application Lifecycle Management) and PLM (Product Lifecycle Management) essential.

Definitions:

ALM: Manages the development, testing, deployment, and maintenance of software applications.

PLM: Manages the design, manufacturing, usage, and end-of-life of physical products.





Why Integrate ALM and PLM?

- **Enhanced Collaboration:** Aligns hardware and software teams, breaks down silos, and supports unified management of complex projects.
- **End-to-End Traceability:** Links requirements, changes, and software tests to hardware components, ensuring regulatory compliance and product quality.
- **Accelerated Innovation:** Synchronizes development cycles, shortens time-to-market, and simplifies change management in multidisciplinary environments.
- **Impact Management:** Facilitates assessment of how software changes affect physical products—and vice versa.

Checklist: Are you ready for agile ALM?

Use this checklist to assess your organization's maturity and its ability to successfully adopt agile ALM. Check each item that reflects your current situation.

1. Governance & Organization

- Roles and responsibilities are clearly defined for each stage of the software lifecycle.
- Requirements, change, and release management processes are formalized and well understood by all.
- Access control, user permissions, and action traceability are properly managed.

2. Requirements Management

- Requirements are documented, analyzed, prioritized, and tracked throughout the project.
- Stakeholders actively collaborate on the definition and evolution of needs.
- Requirements are linked to tests, development tasks, and deliverables.

3. Collaboration & Communication

- Teams have access to tools that support real-time collaboration and transparency.
- Project information (progress, decisions, blockers) is shared and accessible to all.
- Follow-up meetings (daily stand-ups, reviews, retrospectives) are held regularly and are effective.

4. Development & Integration

- Development follows agile principles: short iterations, frequent feedback, continuous adaptation.
- Continuous integration and software configuration management are in place (CI, SCM).
- Teams are capable of delivering functional increments at each iteration.

5. Quality & Testing

- Tests (unit, integration, system, acceptance) are automated and integrated into the development cycle.
- Acceptance criteria are defined for each requirement or user story.
- Defects are tracked, prioritized, and resolved promptly.

6. Deployment & Maintenance

- Deployment processes are automated and repeatable.
- Rollback procedures are defined in case of production failures.
- Corrective and adaptive maintenance is organized, with tracking of incidents and change requests.

7. Monitoring & Reporting

- Dashboards provide real-time visibility into project progress, quality, and risks.
- Key performance indicators (KPIs) are defined and monitored to guide team and project performance.
- Internal and external audits are facilitated through traceability and centralized information.

8. Continuous Improvement

- Retrospectives are held at the end of each iteration to identify areas for improvement.
- Processes are regularly adjusted based on feedback and evolving context.
- The organization encourages skill development and training in agile practices and ALM.

Interpreting the results:

More than 20 boxes checked:

Your organization is ready to adopt or strengthen agile ALM.

Between 12 and 20 boxes checked:

A solid foundation is in place, but there are areas for improvement.

Less than 12 boxes checked:

Support and a maturity ramp-up are recommended before rolling out agile ALM on a large scale.

Conclusion

ALM has become a strategic lever for organizations aiming to accelerate their digital transformation, ensure product quality, and meet compliance requirements—especially in industrial sectors.

Far beyond a simple project management tool, ALM aligns teams, centralizes information, automates processes, and strengthens collaboration at every stage of the software lifecycle.

Adopting an integrated and agile ALM solution empowers teams to **quickly adapt to change**, drive continuous **innovation**, and **deliver** reliable, traceable, and compliant applications—all while managing risks and controlling costs. The growing convergence of ALM and PLM paves the way for unified management of complex projects, which is essential in the era of Industry 4.0.

The success of an ALM initiative depends on a structured approach: setting clear objectives, involving stakeholders, choosing the right solution, supporting change, and committing to continuous improvement.

With this guide, you now have the tools to assess your maturity, adopt best practices, and successfully implement agile ALM—a true driver of performance and innovation for your organization.

Don't wait to take the next step: ALM is a decisive asset for meeting the challenges of today and tomorrow.



[Contact Us](#)

[Get a Demo](#)

