

Inflectra: Software Built For You



Our one goal is to help you succeed. We care deeply about giving you the best quality service and support you've ever had.



As many users, projects, tests, releases, items, API calls as you want. All pricing is based on concurrent users.



Flexible options to make your life easier. Use on desktop or mobile; your servers or our cloud, sensible add-ons, fairly priced.

How To Choose An Application Lifecycle Management Suite

Why Choose SpiraTeam as Your ALM Tool?

SpiraTeam is the premier Application Lifecycle Management (ALM) system that manages your requirements, releases, tests, issues and tasks in one integrated environment.

- SpiraTeam is a **complete solution** that includes requirements management, test case management, release planning, iteration planning, task estimation and defect tracking all fully integrated from day one.
- SpiraTeam provides reporting dashboards of key project quality and progress indicators – requirements test coverage, task progress, project velocity, test execution and top risk and issues – in one consolidated view that is **tailor-made for agile methodologies** as well as supporting your legacy/hybrid waterfall projects.
- SpiraTeam is methodology agnostic and can be used equally well for any agile methodology – including Scrum, AUP, XP, DSDM
- SpiraTeam helps you leverage your existing technology investments, as it integrates with many automated testing solutions, software configuration management (SCM) and third-party defect-management systems.

12 Top Things to Look for in an ALM Platform

1. develop requirements & user-stories, then decompose into tasks for resource tracking
2. prioritize & schedule requirements for each release/iteration
3. assign requirements, tasks & incidents using Agile planning board
4. track bugs, risks & issues linked to changes in your source code repository
5. create, edit, assign & execute test-cases and test sets
6. map tests to requirements to track coverage
7. record bugs linked to test steps for full test traceability
8. manage manual & automated testing
9. personalized dashboards & customizable reporting
10. ability to attach documents, screenshots & URLs to all artifacts
11. full audit history of all changes made in the system
12. customizable graphs & reports in multiple formats.

What Makes Up Application Lifecycle Management?

Application Lifecycle Management (ALM) is an umbrella term that covers several different disciplines that traditionally were considered separate, including project management, requirements management, development, testing and quality assurance (QA) as well as customer support and IT service delivery. In essence, ALM tools provide a standardized environment for communication and collaboration between software development teams and related departments, such as test and operations. They also automate the process of software development and delivery.

Why is ALM So Important Now?

Before the inefficiencies of waterfall approach to software development were thoroughly understood, each of the disciplines that made up the overall software process was completely separate. Having understood that, it made sense to put together integrated teams that collaboratively defined the requirements, planned the releases and sprints, tested the product during development and deployed the latest update in a seamless way. Thus, **ALM was born by fusing together of the disciplines concerned with all aspects of the software delivery process.** With ALM, systems and apps are updated continually, rather than having a major new release every 18 months.

Let's consider some of the components of application lifecycle management:

Requirements Management

A good ALM tool comes with functionality that helps you make sense of your requirements and is adaptable to your methodology and processes.

Some ALM tools support the capture and management of system use-cases as part of the requirements definition process. Use cases can be stored in the system along with their associated scenarios and links to the other requirements that the use case defines. Depending on your needs (do you have complicated multi-step processes or simple CRUD data entry needs) this may be an important feature. Even better if your ALM tool lets you define and manage workflows that help orchestrate your processes. If you are working in regulated or process-driven industries this may be a critical feature.

Estimation and Planning

ALM tools typically provide functionality to help you estimate and plan your projects. It is wise to choose a tool that is designed to manage both – the traditional waterfall projects and agile development projects. Depending on the level of planning you need, consider an ALM tool that lets you estimate individual technical tasks that are then aggregated

back against the original requirements that were specified. If you have a complex review and oversight requirements, you may need to have workflow functionality that ensures all of the assigned tasks follow a specific set of steps with reviews and quality gates.

Software Development

Often you need to have source code management functionality integrated closely within the ALM suite. Today, most ALM tools will offer this functionality. Some will provide proprietary source code management solutions, others will use industry-standard tools, such as Git or Subversion. Ideally, the ALM tool should provide flexibility to support different branching and merging models.

Ideally, even though the code writing is not strictly part of the ALM suite itself, the best ALM tools on the market provide integration with the tools used by developers to write the code. For example, the Microsoft Visual Studio IDE has a plug-in architecture that lets ALM vendors provide plugins to manage work items. The popular open-source Eclipse IDE has gone one step further with its Mylyn task-based interface that lets developers connect to their ALM tool of choice and use the same user interface for all the different supported tools.

Testing and Quality Assurance

One of the key attributes that distinguishes ALM suites from mere project management tools or issue tracking systems is that they include the QA part of the software development process. Additionally, the ALM tools should let you create and manage your test cases in folders, with sorting and filtering capabilities. Users should be able to display different columns in their own customized view, and allow the manager to define custom properties and lists that can be used in the project.

When managing a larger QA team or have testers working in different geographies and time zones, your ALM tool should offer some kind of test suite or test set concept that lets you group test cases for assignment and tracking.

Some ALM suites offer support for automated testing.

As applications become increasingly complex with user interfaces and published APIs to test, with the testing needs covering a wide variety of platforms and technologies (mobile, web, desktop) it is not feasible to manually test everything each time you release a new version or an update. Thus, choose a tool that provides the ability to connect to different testing tools to manage your automated testing, or include native test automation capabilities.

Deployment and DevOps

Although the compilation and packaging of the software being developed is often outside of the scope of ALM tools themselves, most ALM tools will, at the very least,

let you integrate with continuous integration (aka CI) servers, such as Jenkins. Depending

on your needs, choose the ALM tool that can integrate with the different CI servers that your development

teams are using and make sure that the reporting

within the ALM tool can fuse the information from

other sources with the build information. That

will allow you to see all of the changes in each build: features that

have been added and issues that have been resolved.

Related to deployment is the concept of DevOps.

DevOps is the blending of tasks performed by a company's application development

and systems operations teams into a coherent whole. Now

that more applications are being cloud-hosted and release cycles are measured

in days, **you cannot develop and test in isolation.** For testing new updates with real data in

production-ready environments, DevOps is the key.

Seeing which code changes have made it into production and being able to tie customer issues with specific builds becomes increasingly important, requiring that ALM tools have the ability to perform this level of analysis and traceability.

Perhaps the most overlooked part of the software lifecycle is the support and maintenance of the product after it has been released. Traditionally this was done right at

Support and Maintenance

the end when the development team delivered the finished product to the support team and again when the next version was developed. Agile development, continuous delivery and DevOps made support an integral part of the process. Customer support is the feedback mechanism that guides the new user stories in the next sprint. It is therefore advisable to choose the ALM suite that has a customer support capability, or at least integrates with a help desk tool.

the end when the development team delivered the finished product to the support team and again when the next version was developed. Agile development, continuous delivery and DevOps made support an integral part of the process. Customer support is the feedback mechanism that guides the new user stories in the next sprint. It is therefore advisable to choose the ALM suite that has a customer support capability, or at least integrates with a help desk tool.

Project and Portfolio Management

Since ALM tools provide such rich and powerful functionality, they also need to provide good tools for consuming the information and for reporting across the entire system with information summarized and fused together to allow better decision-making.

A good ALM tool will allow you to display 'dashboards' or commonly accessed graphs and charts and/or will have Project Portfolio Management (PPM) functionality for managing hierarchies of projects, each with their own requirements, features and milestones. Look for tools that let you visualize the current status of your projects and/or manage the dependencies and tasks across projects.

Collaboration and Communication

Today's ALM tools should include an integrated instant-messenger that lets users collaborate and communicate in real-time regardless of geographic location, plus other asynchronous methods, such as RSS feeds and email notifications.

What is the Future of ALM?

We are seeing a convergence happening in the Application Lifecycle Management (ALM) and Product Lifecycle Management (PLM) markets. However, as the lines between software and hardware products increasingly blur, agile development dictates that the **hardware and software cannot be developed independently.** Thus, the tools used to manage software and hardware products need to become more integrated.

In a parallel to what happened with ERP, the standard business line functions of HR, Finance and other enterprise functions are also becoming intertwined with ALM / PLM systems. The timesheets entered by a developer to mark the time spent fixing a bug or implementing a feature may need to be billed to a client, or used to determine that quarter's hiring needs. Thus, we believe that in the future, customers will be using Enterprise Lifecycle Management (ELM) solutions, containing elements of ALM, PLM and will either integrate with ERP systems or even include such functions.

