

# White Paper - 10 Benefits of Integrated ALM



**KOVAIR**

By Kovair Marketing

## US Office

**Kovair Software, Inc.**  
2603 Camino Ramon  
STE 200, San Ramon, CA 94583  
**Tel:** 1.408.262.0200 Extn.1  
**Email:** [sales@kovair.com](mailto:sales@kovair.com)

## India Office

**Kovair Software Pvt. Ltd.**  
PTI Building, 6th Floor, DP-9, Sector – V  
Salt Lake City, Kolkata – 700091, India  
**Tel:** 91-33-4065 7016/17/18/19 Extn. 107  
**Email:** [sales@kovair.com](mailto:sales@kovair.com)



# Table of Contents

Introduction.....	3
1. Gain Greater Insight.....	3
2. Enact Best Practice Processes .....	5
3. Overcome Challenges for Globally Distributed Development.....	6
4. Enable Collaboration between Stakeholders.....	7
5. Increase Productivity & Deliver Faster .....	8
6. Enhance Customer Satisfaction.....	8
7. Improve Quality .....	9
8. Collect Actionable Metrics & Intelligence.....	10
9. Manage Change with Confidence .....	11
10. Plug and Play tools.....	12
About Kovair .....	15

# Introduction

Application development organizations spend millions of dollars a year on a multitude of siloed point function tools encompassing requirement management, design/architecture management, coding, configuration management, build, testing, defect tracking, release management, and more. These tools live in isolation which makes users rely mostly on traditional and broken manual procedures while synchronizing data of one tool with the other. These isolated tools do not play well together, therefore, cannot provide a coherent development setup.

You may also find some point-to-point fragile and hardwired integrations between tools in organizations that commit to a single provider for sourcing all their required tools. Organizations that choose best-of-breed tools from different vendors, suited to different lifecycle phases also lack an effective integration mechanism in their tool setup. Here are the ten compelling reasons why an organization must connect these multiple lifecycle tools and achieve an optimum application development environment for its teams.

## 1. Gain Greater Insight

Lack of data visibility into day-to-day activities of an application development project is a big problem for all stakeholders. Business Analysts, Architects, Developers, Testers, and Managers – all work in a very isolated manner using different tools and have limited information about the overall application development activities and status. It is important that developers and testers have the ability to ensure code quality and performance throughout the lifecycle process. This needs quality of work to be validated at every stage and defects to be identified and removed early in the process, not at the end of it.

In a non-synchronized development environment, achieving an end-to-end visibility of tool-specific information is a big challenge.

Business Analysts don't get easy answers to these questions:

- **“Have all the latest customer requirements been successfully incorporated into the final build?”**
- **“Are there test cases covering the performance aspects of this customer requirement?”**
- **“Which customer requirements are designed, which ones are coded and which are tested by now?”**

Developers struggle to get these questions answered:

- **“What was the detailed customer discussion and business perspective of this functional requirement that I am developing?”**
- **Am I developing codes based on the latest design diagrams?**
- **“How many test cases have failed for my code?”**
- **“How do I find which components are using this part of the code that I need to change to implement this new requirement?”**
- **“Which other requirements can reuse my piece of code?”**
- **“Can I see how similar requirements were developed in the past?”**
- **“Which defects or change requests need urgent fixing?”**

Testers find it difficult to get answers to certain questions:

- **“How do I know which other parts of the application need retesting when I am testing the fix for a particular bug?”**
- **“Is somebody working on fixing this bug and what is the current status? Is this recreated by the developer?”**
- **How do I know the source requirements for which these test cases were created and have failed now?**

Project Managers and Executives have limited visibility into the progress and quality of development projects. Whatever visibility they have is typically gleaned from subjective testimonials which are not based on any objective data. They often come up with following questions:

- **“Are we on track to meet the delivery schedule? If not, what are the bottlenecks?”**
- **“Are we doing enough reviews at every stage of the development? What are we finding?”**
- **“What is the quality of the current release that we are working on? Which part is not stable yet? What corrective actions are needed?”**
- **Are we using all the resources as per delivery priorities and time availabilities?**

The information required to answer all the above questions are spread across disparate tools and locked in. All these tools need to be connected to unlock the hidden data and enable all the project stakeholders to share the same pool of up-to-date information real time. A connected ecosystem of tools allows every individual, team, and organization to gain greater insight into the application development, thus helping them to do their job in a much better and informed way. Efficiency can only be achieved through full visibility into projects, people, costs, and items across the lifecycle.

## 2. Enact Best Practice Processes

Organizations, developing software applications find it very difficult to set up an integrated and uniform engineering process across SDLC stages, when the tools used by stakeholders at different stages are disconnected. This raises a series of questions for users as follows:

- **“How do you trigger Requirement review and approval process in your Requirements tool as soon as your customer submits a new feature or change request from a Portal tool?”**
- **“How does the architect get to know from within the Designer tool that he/she needs to start the design process, once the requirements are approved and assigned a Release in the Requirements tool?”**
- **“How does the developer begin the development process using the IDE tool as soon as the requirements pass through the design process in the Designer tool?”**
- **“How do you trigger a build process in a Build tool once the requirements for a Release are developed?”**
- **“How does the tester start testing the requirements for a Release in the Testing tool once the build is available?”**
- **“How do you initiate the defect resolution process in your Defect tool once the tester records test results in the Testing tool?”**
- **“How do you commence your Release and Deployment process in the Release Management tool once the verification process for your Release is completed?”**

Without an interconnected set of tools, it is almost impossible to integrate the above engineering processes carried out in different practitioner tools and more than likely, distributed at the multiple locations. Paper-based processes are commonly used to control handoffs between functional areas. Integrated engineering process automates these handoffs and ensures proper communication,

synchronization, and feedback exchange between these stage level processes. These are essential for any successful application development.

Organizations establish several best-practice processes like Requirements Validation, Code Review and Verification, Test Coverage for Requirements, Build Verification, Risk identification and mitigation, Check for Release Readiness, Change Management, and more spanning one or more lifecycle stages and tools.

- **“How do you ensure that your corporate methodology and processes are adopted across your organization and maintained throughout the application development lifecycle stages?”**
- **“How do you ascertain that your development partner in Bangalore and testing partner in Shanghai are following the same best-practice processes?”**

Connecting the tools is a must for an organization to disseminate, implement, and automate its methodology and best practice processes universally across the lifecycle tools so that the methodology becomes a natural part of the organization’s business process. Automated and executable process models used across these integrated set of tools ensure strict process adherence and achieve consistency, repeatability, and predictability.

Corporations face the daunting task of ensuring effective compliance requirements like Sarbanes-Oxley, CMMI, ITIL, ISO, Six Sigma, and others. Compliance requirements make the overall traceability across all application lifecycle items a necessity, which is practical only if lifecycle tools are connected to each other. Organizations need an environment, where these compliance efforts can be automated and built into the process throughout the application development stages in order to achieve sustainable compliance.

### 3. Overcome Challenges for Globally Distributed Development

Today, the Application development process is more global than ever. Projects increasingly involve multiple teams in disparate locations, often from multiple companies and countries. Enterprise level organizations often encounter a situation, where even for the same lifecycle domain people from different companies/countries with different cultures participate in application development process. It is often the case that one company using DOORS for Requirements definition and Perforce for SCM needs to work with another partner company using RequisitePro for Requirements and ClearCase for SCM. The challenge here for an enterprise is to accommodate different cultural and tools, yet remain connected in such a way that work is not duplicated, productivity is not sacrificed,

and cross-cultural miscommunication is minimized. This is crucial, especially with many global business consolidations happening for enterprises.

- 
- **“How is your development team in Bangalore using Eclipse for developing code remains updated real time on the business requirements managed by your Product team in Boston?”**
- **“How do you control and gain visibility into your development process in Bangalore, testing activities in Shanghai and delivery process in LA happening around the clock 24X7?”**
- **“How do you synchronize your stakeholders’ activities such that the developers in India start development after the requirements are designed in the US and testers in China begin testing immediately after development is complete?”**
- **“How do you measure individuals’ skills and experience with tools, technologies and activities in the application development lifecycle, when they are physically distributed?”**

Here the situation is even more complicated by the fact that people using different lifecycle tools are not sharing the same network, and their tools are behind the firewall in different networks. You will need a modern framework integrating tools behind firewalls across different networks that overcomes the structural challenges of working across remote locations, and ensures project outcome – on time, on budget, and on target with business goals.

## 4. Enable Collaboration between Stakeholders

**“A development effort can still fail miserably, even if analysts document business requirements perfectly, architects build flawless models, developers write defect-free code, and testers execute thousands of tests.”**

Organizations need to provide real-time collaboration and coordination between the various functional teams, including customer, product team, development team, QA team and the testing team by seamlessly integrating different development tools and processes that best fit their needs.

- **“How do the business analysts and customers collaborate real-time on exploring and analyzing business requirements when they are using different tools?”**

- **“How can I start a discussion thread with the tester to recreate a bug in my development environment?”**
- **“How do I see my tasks and work items from within my tool, update my progress status, and finally complete those?”**
- **“How do we notify the right developer immediately after a build is failed for his/her part of code in a Build tool?”**

Performing all these above activities in collaboration is possible, only when stakeholders' tools are connected together. The integration ensures that all stakeholders during application development are in sync and perfect synergy. They share equal visibility into the development artifacts, activities, and project status. Their individual and team activities are synchronized with others and aligned with the ultimate project goals. They don't waste time on manual syncing up of tasks and bloated meeting schedules. They feel unified and take accountability for the success or failure of the whole application development.

## 5. Increase Productivity & Deliver Faster

Connecting the tools used by different stakeholders throughout the application lifecycle stages brings huge opportunities for increasing the productivity of individuals and teams.

- Integration enables implementing and enforcing best practice processes and proven methodologies that drive improvement in productivity.
- Shorter development cycle and quicker software delivery decrease inter-stage transition time.
- Teams can collaborate more efficiently and effectively, eliminating human errors and delivery delays.
- Business analysts, architects, developers, testers, and project managers are able to carry out most of their application development activities using their preferred tools environment without requiring them to learn and use new tools. This saves a lot of training time for stakeholders.
- Reduce inflated meetings, travel time, phone calls and unwanted rework by keeping all the stakeholders in sync and sharing the up-to-date information.
- Automation of repetitive and tedious tasks saves a lot of time and effort.
- Enable productivity improvement by analyzing trends and implementing metrics-based best management practices.

## 6. Enhance Customer Satisfaction

Integrated Application Lifecycle Management (ALM) breaks down the developer/customer barrier, enables rapid delivery of customer-driven value, and results in better customer satisfaction.

- Integration enables business stakeholders and developers to collaborate seamlessly with the customers around common shared artifacts. This provides the opportunity to understand customer needs and expectations better.
- Best-practice processes adopted throughout the lifecycle stages constantly ensure that all the development activities are aligned with the ultimate customer requirements. Customers get exactly what they have asked for.
- Improved quality achieved through the integration helps to deliver a better product to the customer that works for their business.
- Customer gets real-time updates on the application development progress and status because of the increased transparency and visibility into the lifecycle stages. Software quality and reliability are the lifelines to customer loyalty.
- Integration facilitates quick response to customer requests and requirements with the support of an end-to-end traceability of lifecycle artifacts across the stages, and integrated change management process.
- Integration with Customer Relationship Management (CRM) and Help Desk tools provides a transparent view of the progress of customer cases to all stakeholders, including customer support managers, sales, and the entire software development team. This helps organizations to be on the top of every customer issue.
- Customer will instantly know the status of issues and when the fixes will be deployed.

## 7. Improve Quality

Integrating the application development lifecycle tools has direct and indirect influences on the overall quality of a software product. It improves quality by reducing the number of defects due to miscommunication, identifying inconsistencies between requirements, enabling efficient testing, and generally ensuring that the final application meets the needs and expectations of users. Quality is the sum total of effectiveness, efficiency, and visibility.

- Accomplishing tasks with a quality outcome often requires wider vision and an extended access to related information/knowledgebase. In an IT environment, linked information from disparate tools needs to be exposed to the right stakeholders at the right time – for them to come up with 'Quality'. This plays a big role behind the success of IT projects. With integration in place, architects, developers, and testers have ready and real-time access to the lifecycle artifacts from within their own tool. Hence, they become more informed to do better design, coding, and testing activities. All these result in a better quality software.
- Integration ensures code quality and performance throughout the lifecycle processes. Quality of work is validated at every stage, and defects are caught during the process, instead of at the end of it.
- Enforce and automate some of the best practice quality verification processes like code analysis, unit testing, checking code coverage, continuous build verification, root cause

analysis, and more by integrating the relevant tools. Capture these detailed test results/defects to analyze and take smarter decisions to improve quality. Strong analytics help users to anticipate problems before they actually arrive.

- Better change management practices help to minimize quality issues arising out of changes in requirements, designs, and code files. Understanding change dependencies and impacts are the key to optimize quality.

## 8. Collect Actionable Metrics & Intelligence

Organizations are finding it increasingly difficult to get the facts needed to build an up-to-date and reliable picture of application development projects, and then obtain accurate measures against business goals, and deliver simple actionable intelligence. Typically, in the absence of an automated metrics collection program across the disconnected lifecycle tools, managers rely on team members to provide these inputs, and the result is a bunch of inconsistent, imprecise, subjective, static, non-granular, and hard-to-collate information. The absence of factual insights leads to missed deadlines, budget crunch, and fundamental erosion of trust and goodwill in a business relationship. Companies waste millions of dollars each year in project overruns, cancellations, and missed business goals. Senior managers often come across situations where finding answers to the following questions becomes a thing of nightmare.

- **“What is the cost per feature development? Is it going up or down?”**
- **“Are we spending enough time on proper requirements analysis and design?”**
- **“How much time is spent by developers in testing versus developing the code?”**
- **“What is the percentage of code that has been tested so far?”**
- **“What is the productivity and quality gain after we introduced best practice processes like static code analysis, code coverage test, unit testing, and code reviews?”**
- **“What is the percentage of rework?”**
- **“Which tasks do we need to focus to avoid slipping delivery dates?”**
- **“Which resources exemplify the best practices resulting in better quality? Can we adopt those as models for others to follow?”**
- **“Are we properly allocating resources against the project priorities?”**
- **“Are we being able to estimate and plan properly? Are we deviating much from our goals?”**
- **“What is our build and release readiness? How is the code churn?”**

Do you feel that the answers are not easily forthcoming? You need an automated metrics collection system integrated across application development lifecycle tools. This will provide you a decision intelligence solution that delivers relevant, objective, dynamic and granular metrics needed to make smart decisions influencing cost, quality, and time.

## 9. Manage Change with Confidence

Change is inevitable in any application development project. Embracing and managing change properly is a major challenge. Organizations find it difficult to keep all stakeholders in sync with the latest changes in application development and ensure the smooth change propagation through lifecycle silos. In the process, inconsistent information creeps in, resulting in wasted effort, productivity loss, incorrect delivery, and customer dissatisfaction. The project leads often ask the following questions:

- **“What would be the impact to incorporate this customer feedback that requires a change in the top level business requirement? Where are we on developing the functional requirements? What would be the waste? How much can we reuse? Can we squeeze the change in the current release?”**
- **“How do we ensure that the developers are working on the changed scope of the requirement and testers are creating or updating test cases for the recent change?”**
- **“How do the testers easily find out the right sets of test cases that are either modified or newly created because of a change in business requirement, and execute only those test cases?”**
- **“How do we know which business requirements to clarify, which design models to check, which code files to reprogram, and which unit tests to rerun when a bug is detected during system testing?”**
- **“How do we find out the top five bug fixes that required code corrections in multiple code files? Do we have time to retest those risky bug fixes before we move to release step?”**
- **“Which code file has produced the maximum number of bugs?”**

To deal with the above scenarios, what we need is an end-to-end multi-directional traceability between the lifecycle artifacts and deliverables like requirements, models, source code, build scripts

and test cases getting generated throughout the application development stages by different stakeholders using different tools.

Organizations find it challenging to correlate application lifecycle items when they are maintained in isolated tools by people from different functional areas. Without an end-to-end traceability, enterprises will continue to deliver applications that diverge through the lifecycle from the original requirements. A connected set of lifecycle tools makes it easier for the application development teams to better assess the impact of changes, track the full history, automate change propagation, and thus keep everyone on the same page and reduce change reaction time.

## 10. Plug and Play tools

Organizations invest heavily on buying the best-of-breed tools for different lifecycle stages and for training people about these tools. However, this does not allow them to harness the full potential of these tools and get the most value out of their investments. Integrating all or most of these tools can help you achieve an end-to-end ALM solution resulting in increased ROI, and process improvement. You need a comprehensive integration framework that can accommodate your existing tools and processes and don't require you to retool your current processes, or to commit to a particular tool vendor. In a typical multi-tool integration scenario, integration managers often end up asking the following questions:

- **“How do we retire one tool and plug another tool easily in the same lifecycle domain without disturbing the current process, and avoiding the large overhead involved in data migration?”**
- **“How do we maintain multiple tools in the same lifecycle domain e.g. DOORS and Caliber RM for Requirements Management and keep these tools synchronized?”**
- **“How can we control and customize the information flow between the tools and avoid fixed integration flows?”**
- **“How do we incrementally build the integrated tools environment? Can we start with Requirements and Testing tools and then bring in CM tools?”**
- **“How do we automate workflows cutting across the tool boundaries?”**
- **“How do we integrate our existing homegrown tools with the newly adopted vendor tool?”**
- **“How much additional effort do we need to put to incorporate a new tool to our existing integrated system?”**

This integration framework has to be flexible enough to support all of the above integration requirements. You should be able to plug any tool from any vendor any time to your integrated tools ecosystem and follow your best practice process irrespective of the tools chosen.

## Kovair Omnibus Integration Platform

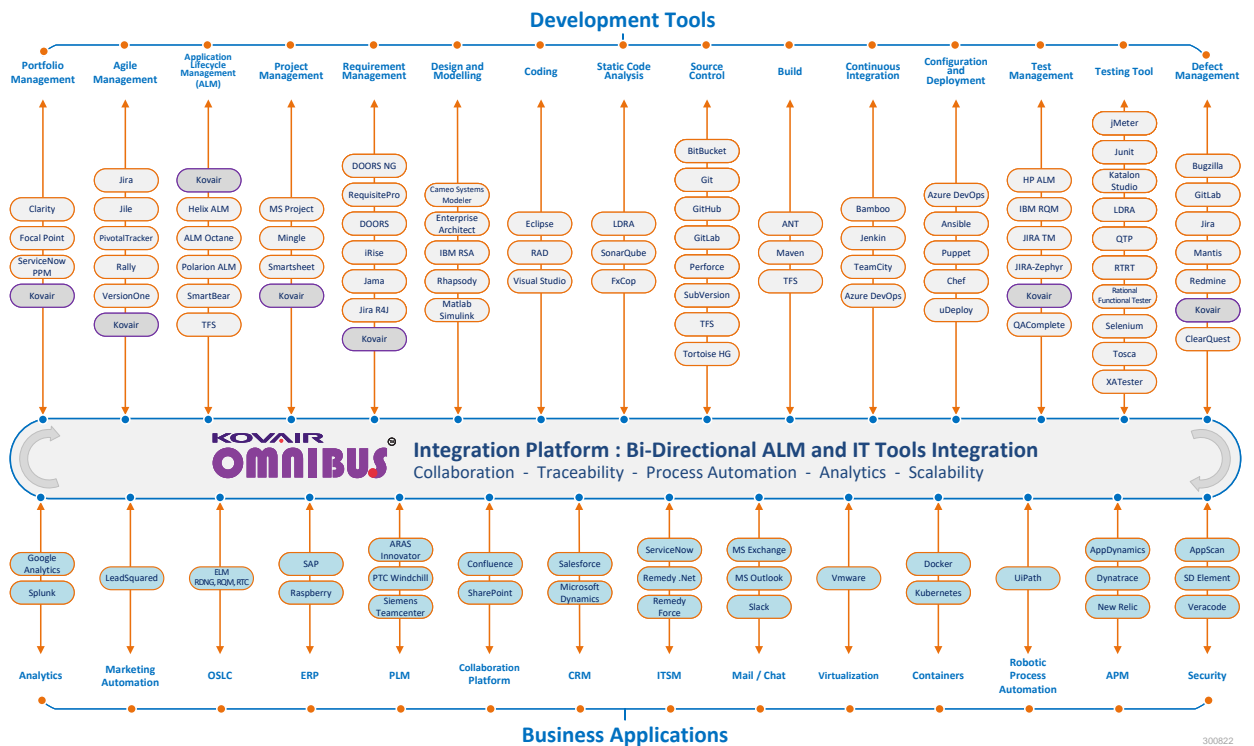
Kovair Omnibus deploys the leading ESB (Enterprise Service Bus) based integration technology for Application Development and IT tools. This technology offers an open and seamless integration middleware with all essential ALM services, like collaboration, traceability, process automation, security, reporting and analytics built on a single repository. With standardized SOA based tool-specific Omnibus adapters/connectors, one can create his or her own tools' ecosystem. These tools can be from any vendor, any legacy data, open sources, or any custom homegrown application development and IT tools.

Kovair's Omnibus integration technology has major advantages which are not typically found in the vendor-specific point tools or even a suite of tools.

Bus or Hub based integration as well as point-to-point integration when a pair of tools is getting considered thus supporting all integration and migration needs of an organization.	Leveraging existing tool investments for customers – no need to rip and replace tools
Support for multi-directional Synchronization, Migration and Federation	Synchronization of data and relations between lifecycle artifacts in different tools of same or different functional areas
Ability to define attribute mapping, integration flows, user and relationship mapping using simple mouse-click configuration	Integration of IBM Jazz based OSLC tools with OSLC non-compatible tools
Support for data validation and transformation	Conflict Detection and Mediation during integration flow
Customizable Integration Rules & Data Mapping independent of Adapters or Connectors – No rule hardcoded in the Adapter	Reuse of built-in configuration templates to support automated integration setup for hundreds and thousands of projects.

<p>An end-to-end traceability from Requirements to Code to Testing and Delivery ensuring compliance</p>	<p>Connects tools behind firewalls through SOA – needed for globally distributed teams</p>
<p>ALM tool interoperability, cross-repository linking and transparency across organizations</p>	<p>Automated end-to-end ALM processes with Task-based workflow management</p>

Below is the diagram showing a Kovair's current portfolio of multi-vendor tool integrations encompassing all functions of the software development lifecycle.



300822

## Conclusion

You have purchased the best tools money can buy to make your development process more effective. You have trained your people in these expensive tools. But, you are not getting the most value out of these investments. You don't have a consolidated picture of your whole development projects, but only the pieces of the puzzle.

You saw the great opportunities you have to achieve substantial savings in time and effort, and to ensure consistent quality by integrating the tools in your software engineering environment. With Kovair, you can have a single unified, automatically synchronized software engineering and collaboration system that avoids tool silos causing inconsistent information and inefficient processes. The net result is a better synchronization between IT and your business, delivering an enhanced business value to your customer and a competitive advantage.

# About Kovair

**Kovair Software** is a Silicon Valley based software product company specializing in the domain of Integrated Application Lifecycle Management (ALM) solutions and supports global software development and management. Kovair's focus on integrating third party best-of-breed ALM tools enables creation of applications in a synchronized tools environment.

Kovair has partnered with leading technology brands like Microsoft, IBM, CA, BMC and more to provide customers a wide range of integration solutions.

**Product Portfolio:** Kovair's flagship products **Omnibus Integration Platform**, **ALM Studio**, **QuickSync** and **Integrated Test Management** are highly preferred solutions by some of the major corporations globally.

**Recognitions:** The **SD Times 100** has recognized Kovair as one of the top 100 software innovators in the domain of Application Lifecycle Management. Kovair's Innovations in ALM Tools and ALM Integrations are well recognized both in the industry and by analysts at places like **Gartner** and **Forrester**.

**Business Focus:** Application Lifecycle Management Products and Services, Integration Platform

**Industry Verticals:** IT Consulting and Services, Banking and Financial Services, Telecom, Manufacturing, Networking, Healthcare, Defense and Government.

**Contact:** For more information about product and services contact [sales@kovair.com](mailto:sales@kovair.com). You may follow Kovair updates on [Facebook](#), [LinkedIn](#), [Twitter](#), [Google+](#), [Slideshare](#) and [YouTube](#).

**Important Links:** [Why Kovair](#) | [Management](#) | [Product Updates](#) | [Tool Integrations](#) | [Product Brochure](#) | [Videos](#) | [Datasheets](#) | [White Papers](#) | [Case Study](#) | [Technical Documents](#) | [Presentations](#) | [Services](#) | [Blog](#) | [Press Releases](#) | [Events](#) | [Customers](#) | [Partners](#) | [Support](#) | [Contact](#) | [Site Map](#)

## Global Technology Partners



Cognizant



## Memberships and Associations



Electronics and Computer Software  
Export Promotion Council

