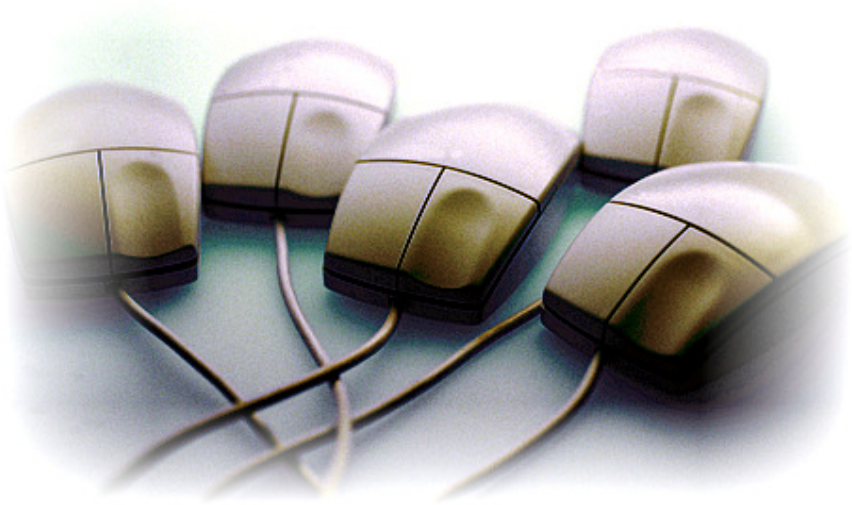
Whitepaper



10 BENEFITS OF INTEGRATED ALM

By

Atanu Majumdar
VP Engineering

Kovair Software, Inc.
4699 Old Ironsides Drive,
Unit 190,
Santa Clara, CA 95054
Ph.: +1.408.262.0200
Fax: +1.408.904.4794
Support:1.408.262.0200 Extn. 2100
Sales: 1.408.262.0200 Extn. 1
www.kovair.com

Application development organizations spend millions of dollars a year on multitude of siloed point function tools encompassing requirement management, architecture, coding, build, testing, tracking, release management and more. These tools live in isolation, rely mostly on traditional broken manual procedures to synchronize data of one tool with the other and as a result do not play well together. You may find some point-to-point fragile hardwired integrations between the tools if an organization commits to a single provider for all the tools. Integration is barely present for organizations choosing best-of-breed tools from different vendors suited to different lifecycle phases. Here are ten compelling reasons why an organization must connect these multiple lifecycle tools for an optimum application development environment.

1. Gain Greater Insight

Lack of visibility into day-to-day activities in application development 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.

Business Analysts don't get answers to their questions –

- ***"Are there test cases covering the performance aspect of this customer requirement?"***
- ***"Which customer requirements are designed, which ones coded and which are tested by now?"***

Developers struggle to get their questions answered –

- ***"What was the detail customer discussion and business perspective for this functional requirement that I am developing?"***
- ***"How many test cases 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?"***

Testers find it difficult to get answers for their questions –

- ***"How do I know which other parts of the application need retesting when I am testing the fix for a bug?"***
- ***"Is somebody working on fixing this bug and what is the current status? Is this recreated by the developer?"***

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 and is not based on any objective data.

- ***"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?"***

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 let all the project stakeholders share the same pool of up-to-date real-time information. Connected ecosystem of tools allows every individual, team and organization to gain greater insight into the application development helping them to do their job much better in an informed way. Efficiency can only be achieved when the right people are on the right projects at the right time, doing the right things on the right items. This is impossible without a full visibility into projects, people, costs and items.

2. Enact Best Practice Processes

Application development organizations find it very difficult to setup an integrated and uniform engineering process across SDLC stages when the tools used by stakeholders at different stages are not connected together.

- ***"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 complete 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 multiple locations. Paper based processes are commonly used to control hand-offs between functional areas. Integrated engineering process automates these hand-offs and ensures proper communication, synchronization and feedback between these stage level processes which is so essential for 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 through 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 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 overall traceability across all application lifecycle items a necessity which is practical only when lifecycle tools are connected. Organizations need an environment where these compliance efforts can be automated and built into the process throughout the application development stages to achieve sustainable compliance.

3. Overcome Challenges for Globally Distributed Development

Application development is more global than ever. Projects increasingly involve multiple teams in disparate locations, often from multiple companies and countries. Enterprise organizations often encounter a situation where even for the same lifecycle domain people from different companies/countries with different cultures participate in application development. 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. Challenge here for an enterprise is to accommodate different culture and tools, yet remain connected in such a way that work is not duplicated, productivity is not sacrificed and cross-culture miscommunication is minimized. This is crucial especially with lot of global business consolidations happening for enterprises.

- ***"How is your development team in Bangalore using Eclipse for developing code getting 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 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 with the fact that people using different lifecycle tools are not sharing the same network but their tools are behind the firewall in separate 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 may be 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?"***

All of the above collaboration is possible only when stakeholders' tools are connected. This integration ensures that all stakeholders in application development are in sync and perfect synergy. They share equal visibility into the development artifacts, activities and status. Their individual and team activities are synchronized with others and aligned with the ultimate project goals. They don't waste time on manual sync up tasks and bloated meeting schedules. They feel unified and take responsibility and 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 by decreasing inter-stage transition time.
- Teams can collaborate more efficiently and effectively, eliminating human interface errors and time delays.
- Business Analysts, architects, developers, testers and project managers are able to carry out most of their application development activities entirely using their preferred tools environment without requiring to learn and use new tools – this saves lot of training time for stakeholders.

- Reduce inflated meetings, travel times, phone calls and rework effort by keeping all the stakeholders in sync and sharing the up-to-date information.
- Automation of repetitive and tedious tasks saves lot of time and effort.
- Enable productivity improvement by analyzing trends and implementing metrics based management best 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 to rely upon.
- 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 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 HelpDesk tools provides complete transparency on the progress of customer cases to all lifecycle stakeholders including customer support managers, sales and the entire software development team. This helps organizations to be on top of customer issues. 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. It improves quality by reducing the number of defects due to miscommunication, catching inconsistencies between requirements, enabling efficient testing, and generally ensuring that the final application meets the needs and expectations of users. Quality is a result of effectiveness, efficiency and visibility.

- Accomplishing tasks with quality outcome, often requires wider vision and extended accessibility to related information/knowledgebase. In an IT environment, linked information from disparate tools needs to be exposed to right stakeholders at the right time – for them to come up with 'Quality', and this plays a big role behind the success of IT projects. With integration; architects, developers and testers have ready real-time access to the lifecycle artifacts from within their tool, and hence

become more informed to do better design, coding and testing activities. All these result in better software quality.

- Integration ensures code quality and performance throughout the lifecycle process. 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 detail level test results/defects to analyze and take smarter decisions to improve quality. Strong analytics help anticipate problems before they arrive.
- Better change management practices help to minimize quality issues arising out of changes in requirements, designs and code files. It is key to understand change dependencies and impacts to optimize quality.

8. Collect Actionable Metrics & Intelligence

Organizations are finding it increasingly hard to get the facts needed to build an up-to-date and reliable picture of application development projects, 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 results are inconsistent, imprecise, subjective, static, not granular and hard to collate. Absence of factual insights lead to missed deadlines, budget creep and fundamental erosion of trust and goodwill in relationship. Companies waste millions of dollars each year in project overruns, cancellations and missed business goals.

- ***"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 percent of code that has been tested?"***
- ***"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 rework percentage?"***
- ***"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?"***
- ***"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 application development. Embracing change and managing it properly is a major challenge. Organizations find it difficult to keep all stakeholders in sync with the latest changes in application development and ensure smooth change propagation through lifecycle silos. In the process, inconsistent information creeps in, resulting in wasted effort, productivity loss, incorrect delivery and customer dissatisfaction.

- ***"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 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 those test cases only?"***
- ***"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 which top five bug fixes required code corrections in multiple code files? Do we have time to retest those risky bug fixes before we release?"***
- ***"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 inter-relate application lifecycle items when they are maintained in isolated tools by people from different functional areas. Without end-to-end traceability, enterprises will continue to deliver applications that diverge through the lifecycle from the original requirements. 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 on training people on these tools. But they have not been able to harness the full potential of these tools and get most value out of their investments. Integrating all or most of these tools you achieve an End-to-End ALM solution resulting in increased ROI and process improvement. You need an integration framework that can accommodate your existing tools and processes and don't require to retool your current processes or to commit to a particular tool vendor.

- ***"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 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?"***

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 Integrations Middleware

Kovair Omnibus is the leading ESB (Enterprise Service Bus) integration technology available 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 tools specific Omnibus adapters, one can create one's own tools ecosystem. These tools can be from any vendor or any legacy data 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.

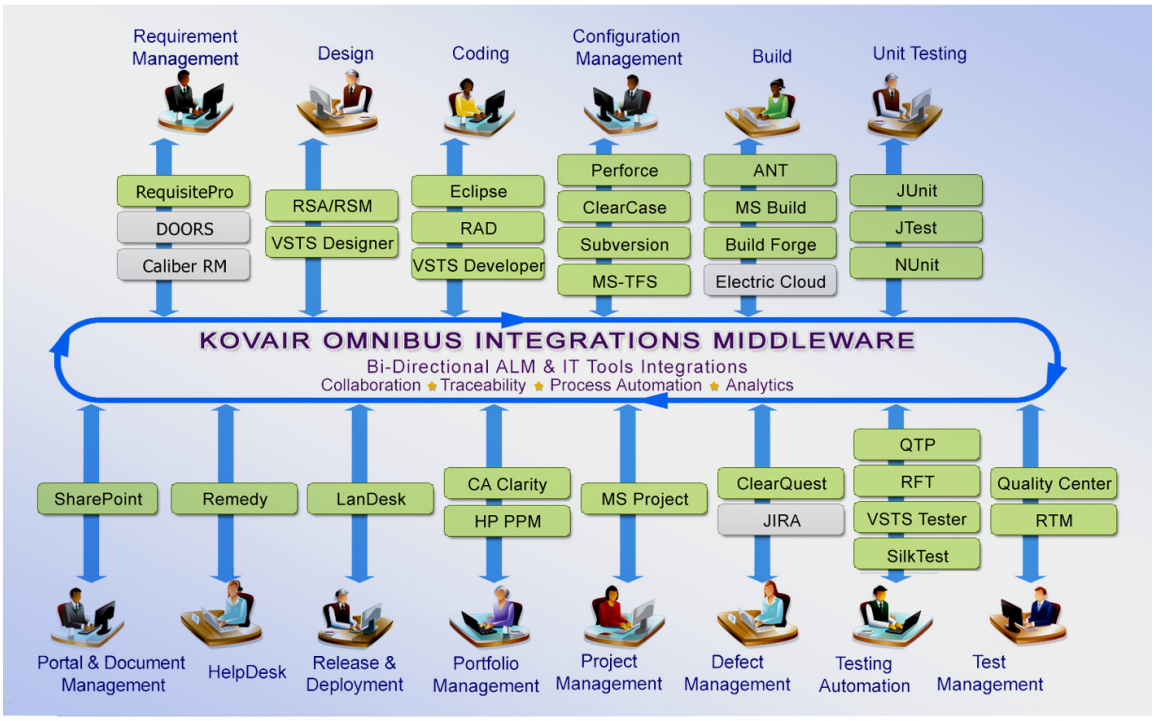
Bus or Hub versus point to point integrations bringing considerable savings in development and maintenance costs	Leverages existing tools investment for customers – no need to rip and replace tools
Supports multi-directional Synchronization and Federation	Synchronization of data and relations between lifecycle artifacts in different tools
Data validation and transformation	Conflict Detection and Mediation
Customizable Integration Rules & Data Mapping independent of Adapters – No rule hardcoded in the Adapter	Supports integrations of multiple tools for the same functional area
Augment Integrated Tools' capabilities for Process & Traceability	Connects tools behind firewalls through SOA – needed for globally distributed team

We have shown below our entire portfolio of all multi-vendor tools for which integrations are currently available or expected to be available over the next 90 days across all functions of the software development lifecycle. It shows a very compelling picture for you to explore and benefit from.

Enhance your ROI on current development tools

Do not rip and replace anything you own

Let your engineers work in their natural tools habitat



Integrate them with the Kovair Omnibus Middleware ™

Make better business decisions in your Lifecycle with Kovair

Kovair brings true End to End ALM and ITSM

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. You can have one unified, automatically synchronized software engineering and collaboration system rather than silos of tools and inconsistent information and processes. The net result of all these benefits is greater synchronization between IT and your business to deliver enhanced business value to your customer and to provide an additional competitive advantage.

Kovair Software, Inc.
4699 Old Ironsides Drive,
Unit 190,
Santa Clara, CA 95054
Ph.: +1.408.262.0200
Fax: +1.408.904.4794
Support:1.408.262.0200 Extn. 2100
Sales: 1.408.262.0200 Extn. 1
www.kovair.com