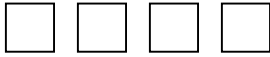


# Best of Breed vs. End-to-end: Solutions for Application Development

An MKS White Paper  
By David J. Martin  
Vice President  
Product Management

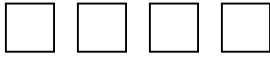


## **Best of Breed vs. End-to-End: Solutions for Application Development**

### **Introduction**

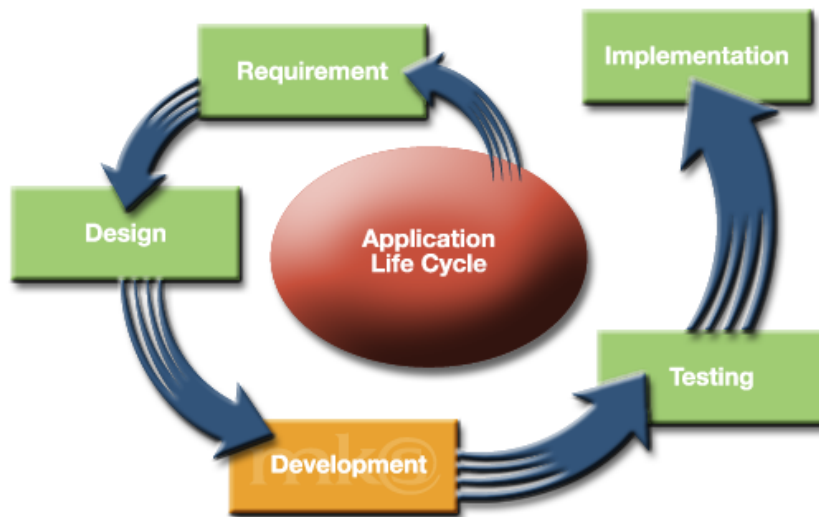
One of the most important decisions facing an organization as it prepares to acquire software development tools is the choice between an end-to-end solution versus a best of breed solution. End-to-end solutions are those offered by vendors who supply all the tools necessary to address the full application lifecycle. Best of breed solutions focus on one or a few elements of the application lifecycle, but integrate with other tools to provide full coverage. Both approaches have their advantages and disadvantages, but, as this paper will argue, best of breed solutions offer software development teams greater flexibility and better technology at a lower cost-of-ownership. Furthermore, best of breed vendors show an unflinching commitment to innovation and customer responsiveness. This commitment originates from a vendor's need to innovate and react to the intense competition present in the industry today.

By contrast, end-to-end solutions offer solid but average technology components at a more expensive cost of ownership. End-to-end solutions are frequently proprietary and often inflexible to customer needs. The monopolistic approach taken by end-to-end vendors stifles innovation and focuses efforts on maximizing revenue from "total account ownership". An end-to-end solution also "traps" customers by ingraining overly complex processes that link various tool components and users but reduce flexibility and productivity. Once a customer has committed to an end-to-end solution, the upfront investment required makes a customer extremely reluctant to replace these tools, out of a fear of total system collapse or highlighting a bad investment. This saddles them with a solution that may not meet their needs in the long run as software development technologies, methodologies and environments change at a torrid pace.



## The Application Lifecycle Management Market Today

Application lifecycle management consists of a set of software tools and practices that guide the development of software from the conceptual stage to the final product. It includes tools for requirements management, design, development, testing, and implementation. The processes and practices surrounding the use of these tools are flexible, from relaxed to complex, and can be governed by standards such as the Capability Maturity Model (CMM) or Six Sigma.

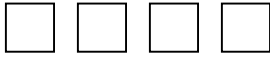


- Integrity Manager
- Source Integrity Enterprise & Standard Editions
- Implementer
- Code Integrity
- Engineer Integrity

The evolution of the ALCM market over the past three years has been remarkable. Before this time, numerous vendors offered unique solutions for software configuration management, testing, requirements management and so on. Development organizations faced a veritable smorgasbord of choices that differed according to functionality, scalability, adaptability and cost. The buzzword during these days was “integration”. Vendors raced to integrate their tools with others in the ALCM space so that they could claim to have broader coverage of the application lifecycle than their competitors.

It came as no surprise to anyone who paid attention to other industries that companies in the ALCM market began to consolidate and merge their operations. Certain companies saw the opportunity to put all the tools necessary for application lifecycle management under one roof. By providing a single, integrated solution, new economies of scale could be realized as well as new efficiencies and greater value for the customer, or so the argument went. These companies held up other industries as an example of why this was the right thing to do. If it was good for the media, transportation and telecommunications industries then it’s good for us and, by extension, our customers. Few questioned this wisdom.

Today there are three vendors that can credibly claim to offer “end-to-end lifecycle management” for software development projects. These companies point to the “natural” advantages of implementing an



integrated solution from a single vendor to handle requirements, design, development, testing and implementation. In truth, the actual integrations between the tools are little better now than they were three years ago when the products were offered by separate vendors. The difference today is in the complex and unifying process that surrounds their use.

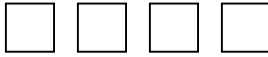
While consolidation has been occurring in the ALCM market, other vendors have resisted the overtures and temptations, choosing instead to focus their efforts on creating better technical solutions to meet their customers' needs. They are concentrating on building better and deeper integrations with other ALCM tools and forging partnerships within the industry that will ultimately result in a broader range of technically superior products. And they have not forgotten about process, either. Process and workflow management solutions, such as MKS Integrity Manager™, is the 'glue' that connects separate technologies and users in the development lifecycle and allows development teams to impose as much or as little process as needed in a particular environment. This is the best of breed approach to application lifecycle management, and MKS is leading the way in the software configuration management (SCM) and process and workflow management segment.

## **Best of Breed vs. End-to-End**

### **The Application Lifecycle – Perfect World vs. Real World**

Just as the ALCM market has matured over the past number of years to offer a multitude of developer productivity tools, so have the processes that define their use. Today, the application lifecycle (also known as the software development lifecycle) is a set of well-defined rules and processes for ensuring success in managing the development of any kind of software. Nobody denies that these rules and processes are the "right" thing to do when managing a software development team. And, if every company and development team were given the resources, skills and time to implement the application lifecycle from end to end, most, if not all, of them would do it. But such is not the case in the real world of software development. For the majority of development organizations, following the application lifecycle to the letter of the law is too complex and simply not practical. In fact, it is overkill. This is why solutions from vendors offering "end-to-end lifecycle management" should not be immediately accepted as the panacea they are made out to be. Often forgotten is the fact that to make the solution work as advertised, a set of overly complex processes must be implemented over an extended period. The truest test of customer satisfaction with a process heavy, end-to-end solution is to review implementation progress after a year. In many cases, acceptance of the tools by users is low because the associated processes are onerous and too complex. The best of breed approach offers much more flexibility because tools are introduced incrementally and integrated into existing processes that are then maintained or improved according to needs.

Unless a particular company is in a heavily regulated industry where software development processes are under scrutiny by government agencies, most companies adopt processes and tools that best fit their particular environment. Most often, that environment is characterized by the push to deliver high quality software on time. Developer productivity is the key ingredient to making this a reality. But productivity is stifled when developers are forced to deal with numerous tools on their desktop, as well as complex rules and processes governing their use. This usually happens when the developers are not a part of the selection process when the new tools are being reviewed and acquired. Too often, higher levels of management assume a greater level of understanding of their development organizations' needs than is really the case. They become persuaded that implementing an overarching process around an end-to-end solution will improve the productivity of their developers. They also perceive a benefit in managing a relationship with a single vendor or supplier. These are not necessarily true advantages. In reality, the frontline developer and his immediate managers have the best perspective and are most aware of the issues, obstacles and peculiarities in their own software development environment. And although most software development teams could benefit from process improvements, the answer is not always to add more complex processes to the mix. Further, while there is a single supplier contact, there is no assurance of responsiveness or continuity of relationship. A more practical solution is to leverage existing, productive processes and tools, and then complement them with new, best of breed tools that allow the team to build on these processes in a developer-



friendly environment. Only those tools that are critical to the developer's job need occupy their desktop. Best of breed vendors also bring more buying freedom and flexibility to the organization.

### **End-to-End Solutions – What are the Consequences?**

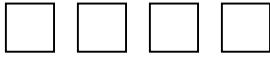
What are the consequences of implementing an end-to-end solution and accompanying complex process for managing software development? The three main consequences are cultural, financial and functional in nature. Culturally, implementing new application lifecycle tools into a software development team is a tricky proposition. It is well recognized that software developers are a unique breed of worker. Their creativity is heavily dependent on the comfort of their surroundings and they are finicky about their tools. They also dislike process. So, by implementing a new end-to-end solution you are taking two risks. First, you risk alienating developers by replacing tools that they have presumably come to accept and enjoy with a new and foreign tool that threatens their comfort. Second, given their distaste for controlling processes, they may be more likely to subvert these processes in an act of workplace protest or in a simple desire to be unencumbered (also known as the "work around" approach). In a best of breed implementation, popular tools remain in place while only the outdated or poorly functioning tools are replaced. Learning curves are reduced and developers remain happy.

Financially, the arguments in favour of best of breed ALCM tools are compelling. The most obvious benefit to best of breed solutions is that only those tools that are considered unsatisfactory are replaced. Tools that are working well and that developers like are left untouched. So are the investments in those successful tools. Often, significant dollars have been spent on implementing and maintaining these tools and training the developers in their use. Vendors offering end-to-end solutions ask organizations to throw away all or most of their tools and replace them with their suite of products. Previous investments, along with potentially successful and productive processes, are swept away into the dustbin. As the new tools are introduced, a new and more complex application development process begins to take shape, which is accompanied by hours of process consulting, implementation services and user training. And user training extends beyond the product to the new and complex process. Learning curves become steep, as do costs for the training.

Similarly, administration of an end-to-end solution is more complex and costly. Whereas it is possible to train individual "specialists" on ALCM tools in a best of breed environment, a single process manager or administrator is necessary in an end-to-end environment to oversee the entire operation. This exposes a critical weak link in the chain should this person be absent for a prolonged period or leave the company permanently. When this happens a new, skilled person must be hired, at significant expense if they can be found at all, or trained from within by a representative of the vendor, again at considerable expense. In a best of breed scenario, a new administrator or specialist is almost always found within the development team. Training is limited to a single tool, and the processes are generally familiar and less complex.

The final financial impact is more difficult to quantify but it deserves equal consideration. Most organizations do not have the luxury of taking many months to train their development teams, overcoming objections to process complexity and fine-tuning the solution and its processes. Slow ramp-up can lead to missed deadlines, discouraged managers and development teams and slow pay back on investment. With a best of breed solution, an organization can implement tools rationally and at their own pace, only progressing when a solid foundation for growth is in place. This allows for a more focused and manageable implementation.

Functionally, there are also important consequences to implementing an end-to-end solution. Vendors that offer this type of solution are generalists. They know something about most elements of the application lifecycle. Best of breed vendors are specialists who are intimately knowledgeable about their specific place in the application lifecycle. They are the true innovators who lead both technical and procedural advances in, for example, SCM. Their resources are devoted to providing the most technically sound product while providing deep and valuable integrations with other best of breed products that cover other parts of the application lifecycle. Providing this kind of value and functionality is the lifeblood of their business. In contrast, end-to-end vendors are less technical innovators than they are integrators. They acquire disparate



tools with disparate technologies and then concentrate on stitching them together to present an integrated solution. Some of the integrations they produce are solid and useful, while others are superficial and barely functional enough to fall under the integration moniker.

Development teams are at the mercy of the vendor when they purchase an end-to-end solution. Frequently, the goal of the vendor is to achieve “total account ownership” so that the customer’s investment is so large and his processes so closely tied to their technology that they wouldn’t think of replacing it, even if something better were available. This provides little motivation for the vendor to be responsive and innovative. When problems arise with their products, customers are often provided with a process “work around” or a script that merely masks the underlying problem. A true patch may or may not appear in a subsequent release. Their resources are devoted to making the disparate technologies work together more smoothly such that innovation becomes stagnant and secondary. Best of breed vendors cannot afford to take this approach. In order to keep their competitive edge they must innovate and provide value where their competitors do not. Most often this value is found in the quality of their technology and integrations. By providing solid and deep integrations with other ALCM tools they differentiate themselves and frequently provide greater functionality than end-to-end solutions. And, most importantly, best of breed vendors are motivated to work closely with their customers and be responsive to their needs because they know that they could be replaced during the next round of software acquisitions.

In the final analysis, the needs and goals of the organization will dictate which kind of ALCM tool or solution is right for them. Some will appreciate the strict and encompassing processes that end-to-end vendors provide while others will prefer to choose the side of process flexibility and vendor independence. Neither way is absolutely right, just in the same way that no software development project is absolutely perfect. However, it is undeniably true that vendors who offer best of breed solutions are on the cutting edge of technology and their integrations with other ALCM tools are delivering value far beyond what has been seen before.

The last section of this white paper will examine one vendor, MKS, and its approach to delivering value in the software configuration management and process and workflow management markets. MKS’s platform transparent, multi-tier architecture is the most advanced SCM platform ever introduced to the market and this vendor’s deep integrations with tools such as Microsoft Project are setting new standards for functionality and quality.

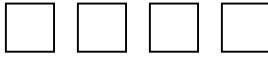
## **MKS Integrity Manager and MKS Source Integrity™ Enterprise Edition – Best of Breed Solutions for SCM and Process & Workflow Management**

### **Integrations**

Every software configuration management product offers integrations into other ALCM products. Most commonly, these integrations occur at the developer level where the SCM tool integrates with a developer’s IDE to allow him to perform simple SCM functions from within the IDE (i.e. check in/out). Most IDEs facilitate integration into SCM tools through Microsoft’s SCC standard interface, which provides the most basic integration functionality. Still other IDEs integrate with SCM tools through custom integrations that offer the promise of additional functionality beyond the standard check in/out, but all too rarely deliver.

As a best of breed provider of SCM and process and workflow management solutions, MKS approaches integrations with a different strategy, one that aims to deliver value and productivity gains to the user beyond what standard integrations offer. There are three main elements to the MKS integration strategy, which differentiates it from its competitors:

- MKS delivers deeper and better integrations than other vendors to offer greater levels of functionality and productivity. While other vendors try to deliver value through maximizing the number of integrations with other ALCM tools, MKS delivers the greatest value by providing integrations that offer real advancements in productivity and ease of use.



- MKS provides integrations into only the top best of breed ALCM tools to leverage their customers' investments in these tools. Some vendors ask customers to replace their current ALCM tools, whether they are working successfully or not, in an attempt to "unify" process and tools. MKS saves customers' investments and complements those well functioning, successful tools by building better integrations with the "best of the best".
- MKS integrations are well thought out and planned so as to minimize mouse-clicks and desktop clutter for developers. End-to-end solutions usually consist of multiple GUIs that encumber developers' desktops and actually inhibit productivity. MKS continues to develop integrations that minimize desktop intrusion while fostering productivity.

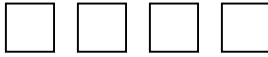
### **MKS Integrity Manager and Microsoft Project Integration**

The best example of this strategy is MKS Integrity Manager's integration into Microsoft Project. Integrity Manager is MKS's process and workflow management solution. It integrates with MKS Source Integrity Enterprise Edition to provide workflow-enabled SCM, but it has also recently been integrated with MS Project to provide advanced project management capabilities. MS Project is an excellent tool for helping project managers supervise the tasks involved in successfully completing a project, software or otherwise. What it is not good at, however, is facilitating collaboration among project contributors and guiding their workflow throughout the life of the project. What this means is that MS Project does not capture information about a task, or subtask, when a contributor or developer actually performs it. Project managers are provided with information only after the contributor or developer has completed the task and presents a subjective number for the time spent on it. By integrating MKS Integrity Manager with MS Project, the time spent on each task or subtask is reported objectively, without influence from the developer or contributor.

Other vendors offer integrations to add workflow and collaboration functionality to MS Project, but the integration with MKS Integrity Manager adds a level of sophistication and functionality that truly illustrates MKS's commitment to delivering greater value than its competitors through its integrations. For example, while other vendors integrate with MS Project to track a single task from inception to completion, there may be several related subtasks that go untracked. This places the project manager at a distinct disadvantage because he will not know how long it took each of the subtasks to be completed. This, in turn, hampers his ability to estimate resource usage for the next project or for later stages of the same project. And it is the accuracy of project timing estimates that separate the good from the bad project managers. MKS Integrity Manager makes the status and time spent on every subtask visible to the project manager. The effect is to equip the project manager with more information so that he makes more accurate estimates next time around and plans his resources appropriately. It also lets him identify bottlenecks or inefficiencies in the project workflow more readily and apply remedial actions quickly.

### **MKS Technology – Innovation in SCM and Process & Workflow Management**

Vendors of best of breed software solutions are the true innovators of the industry. MKS, in business since 1984, is a perfect example of this. Its original product, MKS Toolkit, set the standard for Windows/NT scripting and UNIX/NT interoperability. Today, it is still the de facto standard on developers and administrators desktops around the world. For the past seven years MKS has provided software configuration management tools to software development teams to become one of the leading companies in the market. Its Sandbox™ technology revolutionized parallel development and its most recent release of MKS Integrity Manager and MKS Source Integrity Enterprise Edition, with its advanced SCM architecture, solidifies its position as innovator and leader in the SCM and process management space.

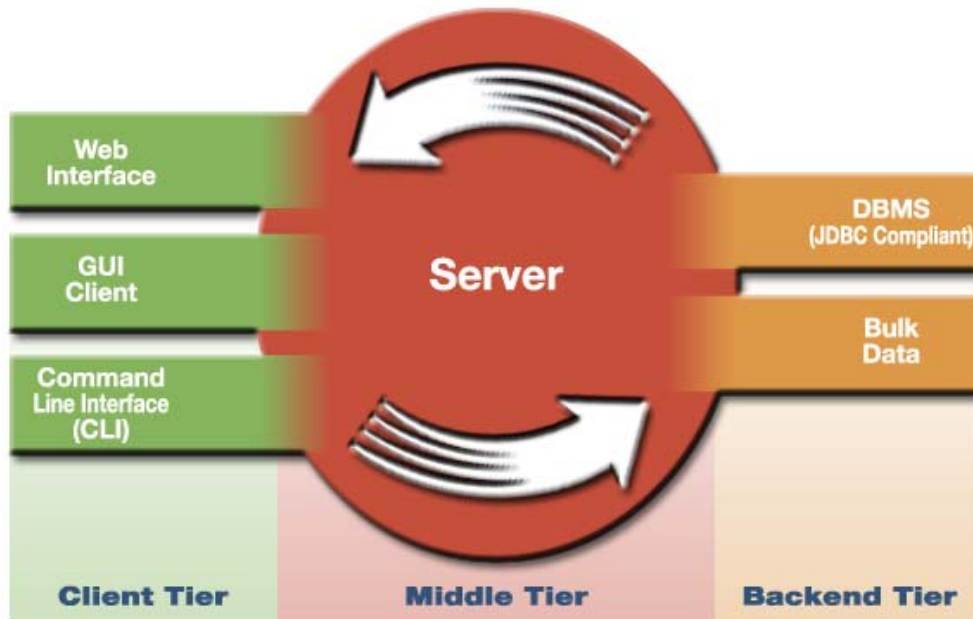


MKS Integrity Manager and Source Integrity Enterprise Edition are built on a platform transparent, multi-tier architecture. It is the most technically advanced platform for SCM and process management available today. It was designed not only to deliver superior functionality, such as distributed development, but also to support MKS's aggressive integration strategy, outlined above. The key to uniting these two elements lies in the open, standards-based architecture.

### Open Standards-Based Architecture

Vendors who offer end-to-end solutions for the application lifecycle frequently incorporate proprietary technology into their products. Their motivation may relate to functionality, but the effect is to further limit the flexibility of their customers. As a best of breed vendor of SCM and process management solutions, MKS is dedicated to building products that offer the greatest degree of openness and flexibility possible. It is a critical consideration because when customers choose best of breed solutions such as MKS's, they expect them to integrate with their own existing ALCM tools to deliver exceptional value. How does MKS deliver this flexibility? By using industry standard technology such as SCC and JDBC interfaces to facilitate integrations into IDEs and relational databases, and Enterprise Java Beans and J2EE to promote platform transparency and cross-platform functionality. MKS incorporates Enterprise Java Beans into their Integrity Manager and Source Integrity Enterprise Edition products to enable remote calls to other Java machines, taking maximum advantage of a Java environment. Its compliance with the J2EE standard ensures seamless integrations into other Java-based applications that reside in the customers' environment.

## Multi-Tier Architecture

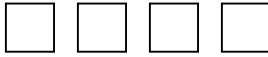


### Benefits of the Multi-Tier Architecture

The multi-tier architecture featured in MKS Integrity Manager and MKS Source Integrity Enterprise Edition offers four distinct benefits that differentiate it from other products in the SCM and process management space:

- **Facilitates integration and adoption of new technology.** The vast majority of the computing logic in MKS Integrity Manager and Source Integrity Enterprise Edition resides on the server, not





on the client. By isolating capabilities on the server side it makes it easier and more efficient to introduce new platforms, IDEs and other technologies. The benefit to the customer is the ability of MKS to introduce new integrations and technologies more quickly as they become available. The ability to react and adopt new technologies quickly in a software development environment can be a distinct competitive advantage when time-to-market pressures are great.

- **Real-time notification of project changes increases productivity.** Again, because almost all of the computing takes place on the server, developers and their managers receive instant notification of project changes as the central server “pushes” them out to the clients. The benefit to the customer is greater productivity and collaboration fostered by greater overall visibility of developers’ actions. Without real-time notifications of project changes developers become out of touch with their colleagues and, therefore, are unaware if another’s changes have impacted their files. By providing real time notifications you give developers greater control over the decisions they make about their own development tasks.
- **Thin client makes administration easier.** The MKS Integrity Manager and Source Integrity Enterprise Edition GUI clients are thin in terms of functionality. This means that if there is a bug or an enhancement to be made it is more likely to occur on the server, where the majority of the logic resides. When the patch or enhancement is applied it is immediately available to all the clients without having to perform lengthy downloads and installations. This makes administration of the MKS tools more simple and efficient, thereby reducing overall maintenance and maintaining productivity.
- **TCP/IP architecture facilitates distributed development in cross-platform environments.** The internet-based architecture used by MKS enables distributed teams to connect to a central server via a web browser to retrieve information in MKS Integrity Manager (N.B. a web interface for Source Integrity Enterprise Edition will be introduced in October, 2001). The web interface is platform transparent as are both the Integrity Manager and Source Integrity Enterprise client GUIs because they are Java-based. The benefit of platform transparent tools is that they leverage customers’ investments in disparate platforms and technologies.

## Conclusion

Ultimately, the unique needs and circumstances of a given software development organization is going to dictate which solution is right for them, best of breed or end-to-end. It is necessary to point out, however, that end-to-end solutions are only valuable when they are implemented in the context of a complex and enveloping process. Too often, this overarching process is overkill for the average development team and actually hampers productivity instead of facilitating it. Furthermore, companies can sometimes do themselves a disservice by adopting an end-to-end application lifecycle solution because they become beholden to the whims of the vendor, thereby limiting their flexibility and ability to adopt the latest technologies.

A best of breed solution, such as the type offered by MKS, is built on innovative ideas and technology. Software development teams are afforded a high degree of process flexibility and are assured that new technologies in software development will be available to them quickly for maximum competitive advantage.

For more information on MKS’s best of breed solutions, point your browser to [www.mks.com](http://www.mks.com).

MKS, Mortice Kern Systems Inc. and design, MKS Code Integrity, MKS Engineer Integrity, MKS Impact Integrity, MKS Integrity Manager, MKS Source Integrity, MKS Toolkit, CodeRover, Discover, Implementer, NuTCCRACKER, SDM and Software Manager are trademarks or registered trademarks of Mortice Kern Systems Inc. All other trademarks acknowledged. © 2001. All rights reserved.

IS08013WP