krugle

How a Leading Health Care Company Improves Code Maintenance and Reuse with Krugle Enterprise

Customer Profile

A leader in health care information technology, with US $1.4 billion in annual revenue and over 300 software developers, this company develops, sells and supports a range of software products and services for the health care industry.

Customer Challenges

Source code is a core asset for this company. Although much discipline is used to coordinate the evolution of existing software and the development of new capabilities, this progressive software development organization saw areas for significant improvement. The key challenges:

Key Challenges

- Improving development efficiencies with legacy code
- Coordinating use and evolution of core code components
- Replacing a problematic internal code indexing project

Fragmented repositories of Source Code and Related Data. Agile development tools recently adopted by the company brought an integrated view of code and related information. Non – code artifacts including defect records, requirements and support tickets can be used to reference related code. This integration is very compelling to the company. Unfortunately, this visibility was restricted to code and code related information produced within the new development applications. In the wake of using the new development tools, extensive fragmentation of existing code storage (across legacy SCM systems) and development information is seen as a significant barrier to improved productivity.

Low productivity with Legacy Code. As much as 45% of the total development effort is involved with maintaining, evolving and integrating with legacy code. As the company grows, this work is being performed by developers and teams unfamiliar with the legacy code. Legacy code is typically visible only through an API – and there no easy access to source code. As a result, these "new" developers must seek out developers familiar with the existing code; this involves not only lost cost due to delays but significant opportunity costs in having senior developers continually involved in support of maintenance activities. If a knowledgeable reference can be found, an expensive learning curve results before work with the legacy code is completed. This is the "best" case, because if a knowledgeable reference can't be found, the learning takes longer and in some cases, developers may take the initiative of rewriting significant portions of the code they can't fully understand.

Poorly managed changes to shared code. The customer relies on several methodologies to leverage shared code across projects. Whenever possible, code is factored into reusable components and then shared among teams. Although the financial benefits associated with this are significant, important problems exist. Because components continually evolve, tracking and managing component versions across projects is important. But because these projects involve a variety of different SCM systems, the only way to check versions is to

manually inspect each active project for a known version and then cross reference the component version. Components "out of support" can then be identified and updated, but the cost of manual searching is prohibitive. A more fundamental problem occurs when planning component changes. Currently, changes may be initiated by an isolated project team. These changes may solve similar problems with other projects – or create problems that hadn't existed before. Because each project's code is stored and managed differently, there is no simple way to understand the impact of such changes before the change is planned or implemented.

Why Krugle Enterprise

These challenges motivated the company to develop its own system for creating a unified, searchable library of internal source code. The effort was terminated when the costs and development challenges associated with reliably supporting indexing/search across multiple SCM systems were understood.

Krugle Enterprise was chosen because it reliably handled code from a range of SCM systems and because of its code search capability across multiple languages.

The customer has used numerous development technologies, languages and tools to develop their current code base. Source Code Management (SCM) systems include CVS, SVN, and Clearcase. Non-code development information is stored in .doc and .pdf documents, wikis and a variety of records based issue tracking systems. Krugle's ability to reliably generate and maintain a current, single searchable index of this code and development information was a key factor in the purchase decision.

The ability of Krugle Enterprise to perform syntax related code search was also an important factor in the decision. By allowing search results to be limited to either definitions or calls, Krugle Enterprise greatly simplified the impact analysis that would be routinely performed.

Krugle Enterprise Benefits

- Centralized, searchable access to ALL of organization's code
- Integrated search of code and code related documentation and metadata
- Dramatically improved use of existing code and development knowledge among "new" developers
- Ability to manage code components effectively across the organization

Other factors that supported the decision were collaboration features and clone detection. Clone detection made it easy to track file level code duplication across the organizations code base. The ability to save permanent URLs to specific search results, along with the ability to create informative notes helped team communication across development and maintenance phases.

In summary, Krugle Enterprise was the only solution that: could be integrated across the development environment (without any changes to the existing development tools or methodologies), scaled to search all of the organization's code for the large developer population, and provided *accurate* searches on code and code related information.

Krugle, Inc. 200 Middlefield Road, Suite 104, Menlo Park, CA 94025 Tel 650-853-1986 Fax 650-853-1889 www.krugle.com

Tangible Improvements with Krugle Enterprise

Krugle Enterprise provides unprecedented visibility into the company's code and related information.

As a result, developers are able to quickly locate, understand, use and troubleshoot relevant code files – across computer language and SCM boundaries. As a result, "new" developers and development teams are more self-sufficient and senior level developers spend up to 50% less time on legacy code maintenance. This allows the senior developers to spend more time on strategically significant development work – improving their contribution, motivation and contentment.

Krugle Enterprise also makes it possible to manage and track code components across the organization – something that simply was not possible before. This is important, because it helps ensure that the components are used to maximum advantage across the company's code base.

Krugle, Inc. 200 Middlefield Road, Suite 104, Menlo Park, CA 94025 Tel 650-853-1986 Fax 650-853-1889 www.krugle.com