

Customer Profile

One of the world's largest communication companies - with a range of products that serve consumer, business and communication infrastructure needs. The product lines deliver similar functionality across a range of technologies and form factors - through extensive use of embedded software.

Customer Challenges

This company relies on component based development and parallel development to achieve cost, quality and time to market benefits. These industry "best practices" create significant development leverage, but lead to important cost saving and quality "opportunities" that cannot be addressed with current applications during maintenance. The key opportunities / challenges:

Ineffective defect containment. Software problems can quickly and unknowingly propagate across projects via shared code that is under development. With no easy or reliable way to coordinate defect identification or fixes, project teams handle specific common issues differently - in some cases, the defect may go "undetected". This results in inconsistent quality, code divergence and significant, unnecessary maintenance expenses.

Poor change management of shared code and code components. When maintenance requires modifications to existing code, it is often the case that those changes will impact other projects. The impact of such change is currently unknown across all projects. As a result, the planned scope of work (to address the change) may be inadequate, and separate changes are made across the projects. Opportunities for leveraged development, testing and support are lost. Solutions are implemented inconsistently across projects. The resulting redundancy and inconsistency are estimated to cost the organization as much as 10% of the overall development effort.

Inability to establish defect root cause and defect propagation. When a defect is found, it is important to understand as much as possible about the problem. In addition to identifying the downstream projects that are affected by this problem, it is essential to accurately identify the defect injection point and understand why the defect occurred. Once the injection situation is known, future problems can be mitigated through training, tooling or process improvements.

Key Challenges

- » Managing evolution and productive use of shared code and code components
- » Defect management across projects
- » Bringing order and efficiency to corrective, adaptive and preventative software maintenance
- » Monitoring and enforcing coding standards

Why Krugle Enterprise

The company had considered enterprise search solutions and even funded an internal pilot project to address the issues mentioned above, but ultimately decided on Krugle Enterprise.

Enterprise search was interesting because it offered keyword search, but suffered from two significant problems. The first problem was inability of Enterprise search solutions to automatically access and index ever changing code files from the SCM repositories (a combination of CVS, SVN and ClearCase). The second problem was that enterprise search solutions can't distinguish between class/function definitions, calls and comments; as a result enterprise search generated too many false positives. Both issues require significant human involvement in the process - creating cost, error and automation issues that made enterprise search categorically unacceptable.

This assessment led to an internal effort to build a code search engine. Although the Lucene open source project was identified as the most appropriate and capable base technology, a number of complex, interdependent technology issues made this homegrown solution impractical. These issues include code parsing, robust access control, scalability to 10GB of source code, a service level integration architecture, and ongoing maintenance of the above.

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, and provided *accurate* code aware searches needed to solve the customer's problems.

Tangible Improvements with Krugle Enterprise

Krugle Enterprise reduces measureable development, testing and support costs associated with software maintenance and parallel development - savings eclipse \$200,000 annually per GB of source code managed. Faster cross project defect detection, better impact analysis and improved cross-project coordination during development greatly reduce overall effort and time to deployment - and significantly improve customer satisfaction.

Krugle Enterprise Benefits

- » Centralized, searchable access to ALL of organization's code
- » Code optimized search reliably finds replicated code fragments across the organization
- » Ability to pinpoint defect sources and instances across all of the organization's code
- » API level integration and compatibility with existing software development tools