KLOCWORK

Continuous static code analysis + SmartRank

Software is fundamental to virtually everything developers interact with on a daily basis. Whether a developer is creating applications to tap into the power of advanced medical devices, or a director of a development team is contributing to the automotive supply chain, the challenges have changed over the past several years.

- Include more features
- Release on shorter timelines
- Adhere to stricter standards than ever before
- Create code free of bugs and security flaws

Managing everything can be overwhelming for developers, and that’s where static code analysis helps.

Accelerate development with continuous static code analysis

The adoption of Agile methods, DevOps, and continuous integration (CI) has taken the velocity of software development to unprecedented levels. Klocwork accelerates the pace by introducing continuous static code analysis (CSCA).

Continuous static code analysis is where comprehensive security, safety, reliability, and performance checkers meet continuous integration tools to provide rapid feedback on the health of incremental check ins across the development team. This is made possible by the unique architecture of the Klocwork analysis engine, designed to maximize scalability and performance for multiple concurrent analysis at a time. In other words, if you’re delivering many releases a day, only Klocwork CSCA handles the frequency and complexity of code changes to give every developer accurate results without the analysis engine getting confused or slowing down.

With native plugin in support, Klocwork integrates with popular CI tools, such as Jenkins and TeamCity, and supports any system that uses scripting and the command line.

More secure code analysis, faster

Klocwork accurately identifies critical security and reliability issues, and supports compliance with key coding standards through sophisticated whole program analysis of C, C++, Java, and C# code. The latest analysis engine features support for C++ language features up to and including C++17.
Prioritize and tackle issues faster with SmartRank

Based on a sophisticated analysis of code, including factors such as analysis complexity, Klocwork identifies and applies a SmartRank recommendation to a subset of detected issues in each build. These are issues that developers can be highly confident are true issues, and worth tackling first.

Continuous delivery of secure, reliable, and conformant code

Klocwork allows developers to identify code defects before they happen, at the desktop. With implementations for the desktop and the enterprise, Klocwork identifies critical safety issues, increases code reliability, and improves coding standards well before code is checked in — saving developers time and organizations money.

Development teams using Klocwork recognize significant productivity gains, while shortening cycle times and mitigating risk due to code defects. Organizations around the world trust Klocwork to help develop the most secure and reliable code possible.

Improve team productivity

With extensive support for the programming languages, platforms, and development environments used by top organizations, Klocwork delivers serious productivity gains to the entire development process.

With on-the-fly desktop analysis, Klocwork functions like a spell-checker for developers. Klocwork provides instant, accurate, and continuous feedback on the critical defects and security vulnerabilities introduced into code, as a developer is writing it.

Drag and drop reporting allows development leads to answer complex questions about the security, reliability, and maintainability of an entire code base in minutes, and makes it easier to get the needed data in the required format quickly. Being able to identify whether an issue occurs in just one code branch, or in other streams, creates a faster time-to-market for organizations.

And, development teams can extend the static analysis capabilities of Klocwork by writing their own security, reliability, or coding standard checkers to meet their organization’s unique needs. These customizable checkers can be tailored to enforce the rules for compliance with each standard by enabling or disabling individual checkers or full checker groups to meet the specific needs of your software development environment and processes.

Built-in refactoring simplifies the time-consuming task of code maintenance for C/C++. Developers can automatically clean up their code and make it easier to understand within Visual Studio or Eclipse. With a consistent refactoring discipline that helps developers improve code structure, code is automatically abstracted into re-usable and understandable segments, ensuring things are expressed as few times as possible, saving time on future code modifications.