

Improving Software Quality with Static Code Analysis

Name Fredrik Håbring

Abstract text Engineers in automotive, aerospace, and other industries must ensure the reliability and quality of increasingly complex high-integrity software systems. To minimize software defects that can lead to catastrophic system failures, injuries, or fatalities, development of many of these systems is governed by standards such as DO-178C, ISO 26262, and IEC 61508.

To improve code quality, many development and testing teams complement traditional software verification activities with static code analysis, which use formal methods with abstract interpretation to verify C, C++, or Ada code. This approach enables engineering teams to:

- Measure code complexity and verify compliance with coding standards
- Prove, via formal methods, that the software will not fail due to certain types of run-time errors
- Perform change impact analysis to determine how changing code in one area may affect the reliability or functionality of code in another