

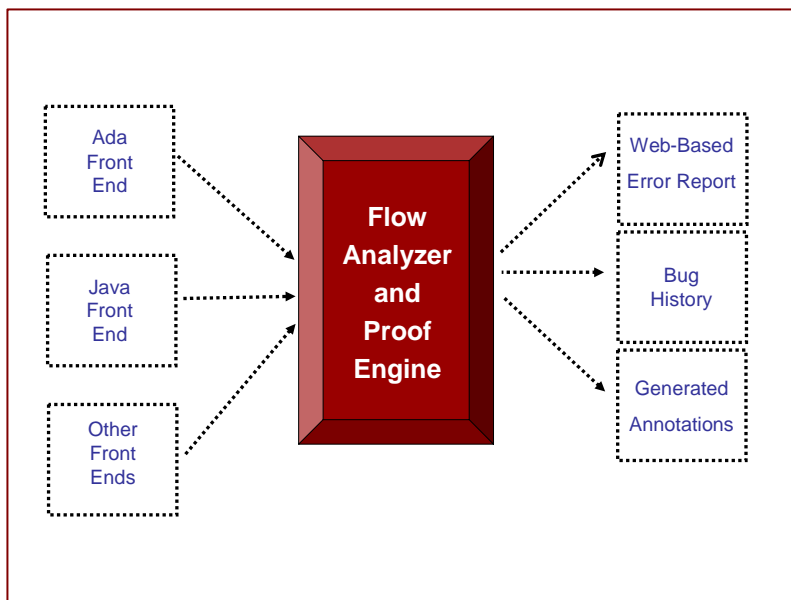
Find Logic Flaws in Ada Programs Automatically

- Achieve 100% path coverage without manual test-scripts or harnesses
- Identify and document component limitations and side effects (generate preconditions and postconditions)
- Pinpoint the root cause of each error, down to the module and line number
- Inspect software components as well as entire applications
- Supports Ada 83 and 95 *

SofCheck Inspector has unique capabilities. It uses static control-flow, data-flow, and possible-value-set propagation techniques to identify places where run-time errors could occur. This automated software quality technique provides 100% path coverage and allows flaws to be identified and eliminated very early in the software life cycle, before run time. SofCheck Inspector considers all combinations of program input across all paths within the program. It can evaluate software components in the absence of a complete system. It automatically generates

both human-readable and machine-readable specifications for component preconditions, postconditions, inputs, outputs, and heap allocations which are displayed in-line with program source code to help immediately pinpoint the root cause of any defect. SofCheck Inspector identifies race conditions that might occur in multi-threaded systems and it finds a number of program anomalies such as “dead code” or “unused variables” that could result in problems later in development. It also maintains an historical error database, which tracks trends across multiple inspections during the software development life cycle of your active code base. * Supports Ada 95 except: Protected Objects, Rendezvous, and Access-to-Subprogram

“SofCheck Inspector uses static analysis of cross-compiled Ada source code— not labor-intensive run time testing—to identify programming errors automatically.”



SofCheck Inspector looks beyond “style” and finds logic flaws. It

scrutinizes complex Ada-based systems for a wide range of logic flaws such as: misuse of pointers, indexing out of arrays (i.e., buffer overflows, a recognized source of security breaches), misuse of numerics (e.g. numeric overflow or wraparound, division by zero), dead code, unused variables, and concurrency hazards (race conditions). SofCheck Inspector finds logic flaws in software *without executing the program*. It works by mathematically analyzing every line of software, considering every possible

input, and every path through the program, in order to find any place where the program might crash or produce a meaningless result.

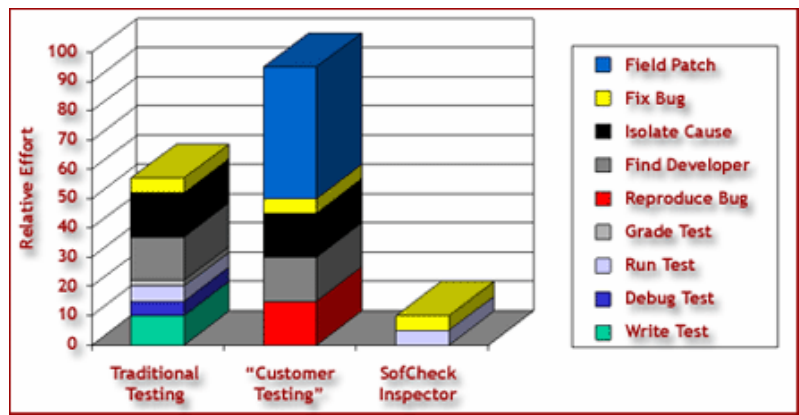
SofCheck Inspector automatically generates preconditions and postconditions. It provides a thorough characterization of every component of the system, even in the absence of detectable logic flaws. It identifies the inputs, outputs, and heap object creations of every component, and the preconditions on the inputs necessary to preclude run-time exceptions or other failures. Presuming the preconditions are satisfied, it determines the postconditions that characterize the system state after the component finishes. The automatically-determined preconditions and postconditions are displayed as comments in-line with source code to help the developer, quality assurance engineer, or architect walk through the code and make explicit assertions that can be guaranteed during run time.

SofCheck Inspector detects race conditions. Ada uses synchronization to guard against improper access to data that is shared by multiple threads. SofCheck Inspector identifies places where synchronization is used incorrectly (such as using different locks for the same shared object) or not used at all. Improper synchronization can cause intermittent problems that are next to impossible to track down with run time testing. Make sure you do not have a synchronization time bomb lurking in your application. SofCheck Inspector can verify the absence of improper synchronization at compile time and give you the confidence you need.

Benefit from Automatic Error Detection

- Detect logic flaws early—when they are least expensive to fix
- Speed time-to-market
- Guard against budget overruns
- Increase team productivity

“Save 90% of your find-and-fix costs... and be more effective.”



SofCheck Inspector results in a faster delivery to market, with a higher quality product, and lower costs. It finds logic flaws wherever they occur, as early as the coding phase—when they are only 10% as costly to fix as flaws found during integration. And for flaws that wouldn't have been found until after delivery, the savings can be 100 to 1000 times greater. SofCheck Inspector, by early identification and precise pinpointing of each problem, can reduce the find-and-fix effort by 90% or more while delivering higher quality code sooner and with lower risk.

