CS405

Merrik Wright

Cppcheck found significantly more issues than Visual Studio, especially logic flaws and dangerous behavior not caught at compile time. Visual Studio primarily caught assertion failures at runtime, while Cppcheck flagged memory misuse, unsafe recursion, and iterator problems. I see why it is recommended to integrate Cppcheck into the development process alongside Visual Studio to improve code safety and maintainability.

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue Description** | **Risk** | **Found By** | **Explanation** |
| Recursion in C::is\_type() | RISK | Cppcheck only | Infinite recursion causes stack overflow. Visual Studio missed it. |
| Assignment inside assert(z = 2) | RISK | Visual Studio | Logic bug; should be comparison (==). |
| Exception thrown in noexcept function | RISK | Cppcheck only | DontThrow() violates noexcept. |
| Invalid iterator erase in vector\_test() | RISK | Cppcheck only | Iterator becomes invalid after erase. |
| Use of pointer to stack variable (foo) | RISK | Cppcheck only | Causes undefined behavior. |
| Infinite while(tok); loop | NOT RISK | Cppcheck only | Harmless here, but usually logic bug. |
| Shadowing variables inside try block | NOT RISK | Cppcheck only | Local variables shadow outer scope. |
| Unused variables (x, y, z) | NOT RISK | Cppcheck only | Code clarity issue, no runtime impact. |
| Return of global a in my\_function() | RISK | Visual Studio (runtime) | Dangerous if logic depends on a staying constant. |



