

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C# C#
- CSS
- Flex
- Go
- HTML
- Java
- JavaScript
- Kotlin
- Objective C
- PHP
- PL/I
- PL/SQL
- Python
- RPG
- Ruby
- Scala
- Swift
- Terraform
- Text
- TypeScript
- T-SQL
- VB.NET
- VB6
- XML



C# static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C# code

All rules 409

Vulnerability 34

Bug 76

Security Hotspot 28

Code Smell 271

Quick Fix 52

Tags

Search by name...

Collections should implement the generic interface

Code Smell

Generic event handlers should be used

Code Smell

Event Handlers should have the correct signature

Code Smell

"Assembly.GetExecutingAssembly" should not be called

Code Smell

Arguments of public methods should be validated against null

Code Smell

Value types should implement "IEquatable<T>"

Code Smell

Finalizers should not be empty

Code Smell

"[ExpectedException]" should not be used

Code Smell

"this" should not be exposed from constructors

Code Smell

Types should not have members with visibility set higher than the type's visibility

Code Smell

Fields should be private

Code Smell

"try" statements with identical "catch" and/or "finally" blocks should be

Types and methods should not have too many generic parameters

Analyze your code

Code Smell Major brain-overload

A method or class with too many type parameters has likely aggregated too many responsibilities and should be split.

Noncompliant Code Example





With the default parameter value of 2:

```
<S, T, U, V> void foo() {} // Noncompliant; not really reada
<String, Integer, Object, String>foo(); // especially on inv
```

Available In:

sonarlint sonarcloud sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved. Privacy Policy

merged  Code Smell
NullPointerException should not be caught  Code Smell
Functions should not have too many lines of code  Code Smell
"for" loop stop conditions should be invariant  Code Smell
Statements should be on separate