

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- 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... 🔍

Exceptions should not be explicitly rethrown

Code Smell

"abstract" classes should not have "public" constructors

Code Smell

Assertion arguments should be passed in the correct order

Code Smell

Ternary operators should not be nested

Code Smell

Events should be invoked

Code Smell

"params" should be used on overrides

Code Smell

Generic type parameters should be co/contravariant when possible

Code Smell

Multiple "OrderBy" calls should not be used

Code Smell

Reflection should not be used to increase accessibility of classes, methods, or fields

Code Smell

Static fields should not be updated in constructors

Code Smell

"IEnumerable" LINQs should be simplified

Code Smell

Fields that are only assigned in the constructor should be "readonly"

Exceptions should not be thrown in finally blocks

Analyze your code

Code Smell Critical ? error-handling suspicious

Throwing an exception from within a finally block will mask any exception which was previously thrown in the try or catch block, and the masked's exception message and stack trace will be lost.

Noncompliant Code Example

```
try
{
    /* some work which end up throwing an exception */
    throw new ArgumentException();
}
finally
{
    /* clean up */
    throw new InvalidOperationException(); // Noncompliant
}
```

Compliant Solution

```
try
{
    /* some work which end up throwing an exception */
    throw new ArgumentException();
}
finally
{
    /* clean up */ // Compliant
}
```

Available In:


sonarlint | sonarcloud | sonarqube

 Code Smell

Static fields should not be used in generic types

 Code Smell

Multiline blocks should be enclosed in curly braces

 Code Smell

Boolean expressions should not be gratuitous

 Code Smell

Types and methods should not have too many generic parameters