## C# static code analysis

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

### [HTTP responses should not be vulnerable to session fixation](https://rules.sonarsource.com/csharp/RSPEC-6287)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-6287)

### [Extracting archives should not lead to zip slip vulnerabilities](https://rules.sonarsource.com/csharp/RSPEC-6096)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-6096)

### [Dynamic code execution should not be vulnerable to injection attacks](https://rules.sonarsource.com/csharp/RSPEC-5334)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5334)

### [HTTP request redirections should not be open to forging attacks](https://rules.sonarsource.com/csharp/RSPEC-5146)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5146)

### [Deserialization should not be vulnerable to injection attacks](https://rules.sonarsource.com/csharp/RSPEC-5135)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5135)

### [Endpoints should not be vulnerable to reflected cross-site scripting (XSS) attacks](https://rules.sonarsource.com/csharp/RSPEC-5131)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5131)

### ["CoSetProxyBlanket" and "CoInitializeSecurity" should not be used](https://rules.sonarsource.com/csharp/RSPEC-3884)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-3884)

### [Database queries should not be vulnerable to injection attacks](https://rules.sonarsource.com/csharp/RSPEC-3649)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-3649)

### [XML parsers should not be vulnerable to XXE attacks](https://rules.sonarsource.com/csharp/RSPEC-2755)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-2755)

### [A secure password should be used when connecting to a database](https://rules.sonarsource.com/csharp/RSPEC-2115)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-2115)

### [XPath expressions should not be vulnerable to injection attacks](https://rules.sonarsource.com/csharp/RSPEC-2091)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-2091)

### [I/O function calls should not be vulnerable to path injection attacks](https://rules.sonarsource.com/csharp/RSPEC-2083)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-2083)

### [LDAP queries should not be vulnerable to injection attacks](https://rules.sonarsource.com/csharp/RSPEC-2078)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-2078)

### [OS commands should not be vulnerable to command injection attacks](https://rules.sonarsource.com/csharp/RSPEC-2076)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-2076)

### [Classes should implement their "ExportAttribute" interfaces](https://rules.sonarsource.com/csharp/RSPEC-4159)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-4159)

### [Neither "Thread.Resume" nor "Thread.Suspend" should be used](https://rules.sonarsource.com/csharp/RSPEC-3889)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3889)

### ["SafeHandle.DangerousGetHandle" should not be called](https://rules.sonarsource.com/csharp/RSPEC-3869)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3869)

### [Type inheritance should not be recursive](https://rules.sonarsource.com/csharp/RSPEC-3464)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3464)

### ["IDisposables" should be disposed](https://rules.sonarsource.com/csharp/RSPEC-2930)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2930)

### [SQL keywords should be delimited by whitespace](https://rules.sonarsource.com/csharp/RSPEC-2857)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2857)

### [Composite format strings should not lead to unexpected behavior at runtime](https://rules.sonarsource.com/csharp/RSPEC-2275)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2275)

### [Recursion should not be infinite](https://rules.sonarsource.com/csharp/RSPEC-2190)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2190)

### [Destructors should not throw exceptions](https://rules.sonarsource.com/csharp/RSPEC-1048)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-1048)

### [Hard-coded credentials are security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-2068)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-2068)

### [Exceptions should not be thrown from unexpected methods](https://rules.sonarsource.com/csharp/RSPEC-3877)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3877)

### ["operator==" should not be overloaded on reference types](https://rules.sonarsource.com/csharp/RSPEC-3875)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3875)

### [Type should not be examined on "System.Type" instances](https://rules.sonarsource.com/csharp/RSPEC-3443)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3443)

### [Test method signatures should be correct](https://rules.sonarsource.com/csharp/RSPEC-3433)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3433)

### [Method overloads with default parameter values should not overlap](https://rules.sonarsource.com/csharp/RSPEC-3427)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3427)

### ["value" parameters should be used](https://rules.sonarsource.com/csharp/RSPEC-3237)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3237)

### ["is" should not be used with "this"](https://rules.sonarsource.com/csharp/RSPEC-3060)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3060)

### [Methods named "Dispose" should implement "IDisposable.Dispose"](https://rules.sonarsource.com/csharp/RSPEC-2953)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2953)

### [Tests should include assertions](https://rules.sonarsource.com/csharp/RSPEC-2699)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2699)

### [Silly bit operations should not be performed](https://rules.sonarsource.com/csharp/RSPEC-2437)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2437)

### [Public methods should not have multidimensional array parameters](https://rules.sonarsource.com/csharp/RSPEC-2368)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2368)

### ["async" and "await" should not be used as identifiers](https://rules.sonarsource.com/csharp/RSPEC-2306)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2306)

### [TestCases should contain tests](https://rules.sonarsource.com/csharp/RSPEC-2187)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2187)

### [Short-circuit logic should be used in boolean contexts](https://rules.sonarsource.com/csharp/RSPEC-2178)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2178)

### [JWT should be signed and verified with strong cipher algorithms](https://rules.sonarsource.com/csharp/RSPEC-5659)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5659)

### [Cipher algorithms should be robust](https://rules.sonarsource.com/csharp/RSPEC-5547)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5547)

### [Encryption algorithms should be used with secure mode and padding scheme](https://rules.sonarsource.com/csharp/RSPEC-5542)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5542)

### [Insecure temporary file creation methods should not be used](https://rules.sonarsource.com/csharp/RSPEC-5445)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5445)

### [Server certificates should be verified during SSL/TLS connections](https://rules.sonarsource.com/csharp/RSPEC-4830)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-4830)

### [LDAP connections should be authenticated](https://rules.sonarsource.com/csharp/RSPEC-4433)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-4433)

### [Cryptographic keys should be robust](https://rules.sonarsource.com/csharp/RSPEC-4426)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-4426)

### [Weak SSL/TLS protocols should not be used](https://rules.sonarsource.com/csharp/RSPEC-4423)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-4423)

### [Cipher Block Chaining IVs should be unpredictable](https://rules.sonarsource.com/csharp/RSPEC-3329)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-3329)

### [Regular expressions should not be vulnerable to Denial of Service attacks](https://rules.sonarsource.com/csharp/RSPEC-2631)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-2631)

### [Hashes should include an unpredictable salt](https://rules.sonarsource.com/csharp/RSPEC-2053)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-2053)

### [Non-async "Task/Task<T>" methods should not return null](https://rules.sonarsource.com/csharp/RSPEC-4586)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-4586)

### [Calls to delegate's method "BeginInvoke" should be paired with calls to "EndInvoke"](https://rules.sonarsource.com/csharp/RSPEC-4583)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-4583)

### ["Shared" parts should not be created with "new"](https://rules.sonarsource.com/csharp/RSPEC-4277)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-4277)

### [Getters and setters should access the expected fields](https://rules.sonarsource.com/csharp/RSPEC-4275)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-4275)

### [Right operands of shift operators should be integers](https://rules.sonarsource.com/csharp/RSPEC-3449)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3449)

### [Shared resources should not be used for locking](https://rules.sonarsource.com/csharp/RSPEC-2551)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2551)

### [Locks should be released](https://rules.sonarsource.com/csharp/RSPEC-2222)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2222)

### [Using publicly writable directories is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-5443)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-5443)

### [Using clear-text protocols is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-5332)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-5332)

### [Expanding archive files without controlling resource consumption is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-5042)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-5042)

### [Configuring loggers is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-4792)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-4792)

### [Using weak hashing algorithms is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-4790)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-4790)

### [Disabling CSRF protections is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-4502)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-4502)

### [Using non-standard cryptographic algorithms is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-2257)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-2257)

### [Using pseudorandom number generators (PRNGs) is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-2245)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-2245)

### [Parameter names should match base declaration and other partial definitions](https://rules.sonarsource.com/csharp/RSPEC-927)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-927)

### ["ValueTask" should be consumed correctly](https://rules.sonarsource.com/csharp/RSPEC-5034)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-5034)

### [String offset-based methods should be preferred for finding substrings from offsets](https://rules.sonarsource.com/csharp/RSPEC-4635)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4635)

### ["default" clauses should be first or last](https://rules.sonarsource.com/csharp/RSPEC-4524)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4524)

### [Unread "private" fields should be removed](https://rules.sonarsource.com/csharp/RSPEC-4487)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4487)

### [Base class methods should not be hidden](https://rules.sonarsource.com/csharp/RSPEC-4019)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4019)

### [Inherited member visibility should not be decreased](https://rules.sonarsource.com/csharp/RSPEC-4015)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4015)

### [Threads should not lock on objects with weak identity](https://rules.sonarsource.com/csharp/RSPEC-3998)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3998)

### [A conditionally executed single line should be denoted by indentation](https://rules.sonarsource.com/csharp/RSPEC-3973)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3973)

### [Conditionals should start on new lines](https://rules.sonarsource.com/csharp/RSPEC-3972)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3972)

### [Assemblies should have version information](https://rules.sonarsource.com/csharp/RSPEC-3904)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3904)

### [Exception types should be "public"](https://rules.sonarsource.com/csharp/RSPEC-3871)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3871)

### [Cognitive Complexity of methods should not be too high](https://rules.sonarsource.com/csharp/RSPEC-3776)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3776)

### ["params" should not be introduced on overrides](https://rules.sonarsource.com/csharp/RSPEC-3600)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3600)

### ["[DefaultValue]" should not be used when "[DefaultParameterValue]" is meant](https://rules.sonarsource.com/csharp/RSPEC-3451)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3451)

### ["[Optional]" should not be used on "ref" or "out" parameters](https://rules.sonarsource.com/csharp/RSPEC-3447)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3447)

### [Non-flags enums should not be used in bitwise operations](https://rules.sonarsource.com/csharp/RSPEC-3265)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3265)

### [Inner class members should not shadow outer class "static" or type members](https://rules.sonarsource.com/csharp/RSPEC-3218)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3218)

### ["Explicit" conversions of "foreach" loops should not be used](https://rules.sonarsource.com/csharp/RSPEC-3217)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3217)

### [Instance members should not write to "static" fields](https://rules.sonarsource.com/csharp/RSPEC-2696)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2696)

### ["IndexOf" checks should not be for positive numbers](https://rules.sonarsource.com/csharp/RSPEC-2692)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2692)

### [Whitespace and control characters in string literals should be explicit](https://rules.sonarsource.com/csharp/RSPEC-2479)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2479)

### [Properties should not make collection or array copies](https://rules.sonarsource.com/csharp/RSPEC-2365)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2365)

### [Flags enumerations zero-value members should be named "None"](https://rules.sonarsource.com/csharp/RSPEC-2346)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2346)

### [Overflow checking should not be disabled for "Enumerable.Sum"](https://rules.sonarsource.com/csharp/RSPEC-2291)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2291)

### [Field-like events should not be virtual](https://rules.sonarsource.com/csharp/RSPEC-2290)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2290)

### [Non-constant static fields should not be visible](https://rules.sonarsource.com/csharp/RSPEC-2223)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2223)

### [Inappropriate casts should not be made](https://rules.sonarsource.com/csharp/RSPEC-1944)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1944)

### [Constructors should only call non-overridable methods](https://rules.sonarsource.com/csharp/RSPEC-1699)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1699)

### ["GC.Collect" should not be called](https://rules.sonarsource.com/csharp/RSPEC-1215)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1215)

### [Methods should not be empty](https://rules.sonarsource.com/csharp/RSPEC-1186)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1186)

### [Exceptions should not be thrown in finally blocks](https://rules.sonarsource.com/csharp/RSPEC-1163)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1163)

### [Method overrides should not change parameter defaults](https://rules.sonarsource.com/csharp/RSPEC-1006)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1006)

### [Types allowed to be deserialized should be restricted](https://rules.sonarsource.com/csharp/RSPEC-5773)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5773)

### [Server-side requests should not be vulnerable to forging attacks](https://rules.sonarsource.com/csharp/RSPEC-5144)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5144)

### [Members should not have conflicting transparency annotations](https://rules.sonarsource.com/csharp/RSPEC-4211)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-4211)

### ["PartCreationPolicyAttribute" should be used with "ExportAttribute"](https://rules.sonarsource.com/csharp/RSPEC-4428)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-4428)

### ["ConstructorArgument" parameters should exist in constructors](https://rules.sonarsource.com/csharp/RSPEC-4260)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-4260)

### [Windows Forms entry points should be marked with STAThread](https://rules.sonarsource.com/csharp/RSPEC-4210)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-4210)

### [Collection elements should not be replaced unconditionally](https://rules.sonarsource.com/csharp/RSPEC-4143)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-4143)

### [Exceptions should not be created without being thrown](https://rules.sonarsource.com/csharp/RSPEC-3984)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3984)

### [Collection sizes and array length comparisons should make sense](https://rules.sonarsource.com/csharp/RSPEC-3981)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3981)

### [Serialization event handlers should be implemented correctly](https://rules.sonarsource.com/csharp/RSPEC-3927)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3927)

### [Deserialization methods should be provided for "OptionalField" members](https://rules.sonarsource.com/csharp/RSPEC-3926)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3926)

### [All branches in a conditional structure should not have exactly the same implementation](https://rules.sonarsource.com/csharp/RSPEC-3923)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3923)

### [Types should be defined in named namespaces](https://rules.sonarsource.com/csharp/RSPEC-3903)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3903)

### [Empty nullable value should not be accessed](https://rules.sonarsource.com/csharp/RSPEC-3655)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3655)

### [Nullable type comparison should not be redundant](https://rules.sonarsource.com/csharp/RSPEC-3610)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3610)

### [Methods with "Pure" attribute should return a value](https://rules.sonarsource.com/csharp/RSPEC-3603)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3603)

### [One-way "OperationContract" methods should have "void" return type](https://rules.sonarsource.com/csharp/RSPEC-3598)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3598)

### [Optional parameters should be passed to "base" calls](https://rules.sonarsource.com/csharp/RSPEC-3466)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3466)

### [Classes should not have only "private" constructors](https://rules.sonarsource.com/csharp/RSPEC-3453)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3453)

### [Expressions used in "Debug.Assert" should not produce side effects](https://rules.sonarsource.com/csharp/RSPEC-3346)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3346)

### [Caller information parameters should come at the end of the parameter list](https://rules.sonarsource.com/csharp/RSPEC-3343)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3343)

### [Static fields should appear in the order they must be initialized](https://rules.sonarsource.com/csharp/RSPEC-3263)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3263)

### [Classes directly extending "object" should not call "base" in "GetHashCode" or "Equals"](https://rules.sonarsource.com/csharp/RSPEC-3249)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3249)

### [Anonymous delegates should not be used to unsubscribe from Events](https://rules.sonarsource.com/csharp/RSPEC-3244)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3244)

### [Delegates should not be subtracted](https://rules.sonarsource.com/csharp/RSPEC-3172)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3172)

### ["async" methods should not return "void"](https://rules.sonarsource.com/csharp/RSPEC-3168)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3168)

### ["ThreadStatic" should not be used on non-static fields](https://rules.sonarsource.com/csharp/RSPEC-3005)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3005)

### ["IDisposables" created in a "using" statement should not be returned](https://rules.sonarsource.com/csharp/RSPEC-2997)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2997)

### ["ThreadStatic" fields should not be initialized](https://rules.sonarsource.com/csharp/RSPEC-2996)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2996)

### ["Object.ReferenceEquals" should not be used for value types](https://rules.sonarsource.com/csharp/RSPEC-2995)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2995)

### [Doubled prefix operators "!!" and "~~" should not be used](https://rules.sonarsource.com/csharp/RSPEC-2761)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2761)

### ["=+" should not be used instead of "+="](https://rules.sonarsource.com/csharp/RSPEC-2757)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2757)

### ["NaN" should not be used in comparisons](https://rules.sonarsource.com/csharp/RSPEC-2688)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2688)

### [Conditionally executed code should be reachable](https://rules.sonarsource.com/csharp/RSPEC-2583)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2583)

### [Null pointers should not be dereferenced](https://rules.sonarsource.com/csharp/RSPEC-2259)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2259)

### [For-loop conditions should be true at least once](https://rules.sonarsource.com/csharp/RSPEC-2252)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2252)

### [A "for" loop update clause should move the counter in the right direction](https://rules.sonarsource.com/csharp/RSPEC-2251)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2251)

### ["ToString()" method should not return null](https://rules.sonarsource.com/csharp/RSPEC-2225)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2225)

### [Return values from functions without side effects should not be ignored](https://rules.sonarsource.com/csharp/RSPEC-2201)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2201)

### [Values should not be uselessly incremented](https://rules.sonarsource.com/csharp/RSPEC-2123)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2123)

### [Collections should not be passed as arguments to their own methods](https://rules.sonarsource.com/csharp/RSPEC-2114)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2114)

### [Related "if/else if" statements should not have the same condition](https://rules.sonarsource.com/csharp/RSPEC-1862)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-1862)

### [Objects should not be created to be dropped immediately without being used](https://rules.sonarsource.com/csharp/RSPEC-1848)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-1848)

### [Identical expressions should not be used on both sides of a binary operator](https://rules.sonarsource.com/csharp/RSPEC-1764)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-1764)

### [Loops with at most one iteration should be refactored](https://rules.sonarsource.com/csharp/RSPEC-1751)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-1751)

### [Variables should not be self-assigned](https://rules.sonarsource.com/csharp/RSPEC-1656)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-1656)

### [Constructing arguments of system commands from user input is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-6350)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-6350)

### [Deserializing objects without performing data validation is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-5766)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-5766)

### [Disabling ASP.NET "Request Validation" feature is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-5753)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-5753)

### [Allowing requests with excessive content length is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-5693)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-5693)

### [Setting loose file permissions is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-2612)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-2612)

### [Formatting SQL queries is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-2077)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-2077)

### [Using hardcoded IP addresses is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-1313)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-1313)

### ["goto" statement should not be used](https://rules.sonarsource.com/csharp/RSPEC-907)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-907)

### ["new Guid()" should not be used](https://rules.sonarsource.com/csharp/RSPEC-4581)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4581)

### [Parameter validation in "async"/"await" methods should be wrapped](https://rules.sonarsource.com/csharp/RSPEC-4457)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4457)

### [Parameter validation in yielding methods should be wrapped](https://rules.sonarsource.com/csharp/RSPEC-4456)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4456)

### [Events should have proper arguments](https://rules.sonarsource.com/csharp/RSPEC-4220)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4220)

### ["P/Invoke" methods should not be visible](https://rules.sonarsource.com/csharp/RSPEC-4214)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4214)

### [Native methods should be wrapped](https://rules.sonarsource.com/csharp/RSPEC-4200)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4200)

### [Methods should not have identical implementations](https://rules.sonarsource.com/csharp/RSPEC-4144)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4144)

### [Non-flags enums should not be marked with "FlagsAttribute"](https://rules.sonarsource.com/csharp/RSPEC-4070)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4070)

### [Classes implementing "IEquatable<T>" should be sealed](https://rules.sonarsource.com/csharp/RSPEC-4035)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4035)

### ["GC.SuppressFinalize" should not be called](https://rules.sonarsource.com/csharp/RSPEC-3971)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3971)

### [Objects should not be disposed more than once](https://rules.sonarsource.com/csharp/RSPEC-3966)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3966)

### [Parameter names used into ArgumentException constructors should match an existing one](https://rules.sonarsource.com/csharp/RSPEC-3928)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3928)

### ["ISerializable" should be implemented correctly](https://rules.sonarsource.com/csharp/RSPEC-3925)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3925)

### ["Assembly.Load" should be used](https://rules.sonarsource.com/csharp/RSPEC-3885)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3885)

### ["IDisposable" should be implemented correctly](https://rules.sonarsource.com/csharp/RSPEC-3881)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3881)

### ["ServiceContract" and "OperationContract" attributes should be used together](https://rules.sonarsource.com/csharp/RSPEC-3597)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3597)

### [Composite format strings should be used correctly](https://rules.sonarsource.com/csharp/RSPEC-3457)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3457)

### [Exceptions should not be explicitly rethrown](https://rules.sonarsource.com/csharp/RSPEC-3445)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3445)

### ["abstract" classes should not have "public" constructors](https://rules.sonarsource.com/csharp/RSPEC-3442)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3442)

### [Assertion arguments should be passed in the correct order](https://rules.sonarsource.com/csharp/RSPEC-3415)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3415)

### [Ternary operators should not be nested](https://rules.sonarsource.com/csharp/RSPEC-3358)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3358)

### [Events should be invoked](https://rules.sonarsource.com/csharp/RSPEC-3264)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3264)

### ["params" should be used on overrides](https://rules.sonarsource.com/csharp/RSPEC-3262)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3262)

### [Generic type parameters should be co/contravariant when possible](https://rules.sonarsource.com/csharp/RSPEC-3246)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3246)

### [Multiple "OrderBy" calls should not be used](https://rules.sonarsource.com/csharp/RSPEC-3169)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3169)

### [Reflection should not be used to increase accessibility of classes, methods, or fields](https://rules.sonarsource.com/csharp/RSPEC-3011)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3011)

### [Static fields should not be updated in constructors](https://rules.sonarsource.com/csharp/RSPEC-3010)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3010)

### ["IEnumerable" LINQs should be simplified](https://rules.sonarsource.com/csharp/RSPEC-2971)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2971)

### [Fields that are only assigned in the constructor should be "readonly"](https://rules.sonarsource.com/csharp/RSPEC-2933)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2933)

### [Static fields should not be used in generic types](https://rules.sonarsource.com/csharp/RSPEC-2743)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2743)

### [Multiline blocks should be enclosed in curly braces](https://rules.sonarsource.com/csharp/RSPEC-2681)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2681)

### [Boolean expressions should not be gratuitous](https://rules.sonarsource.com/csharp/RSPEC-2589)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2589)

### [Types and methods should not have too many generic parameters](https://rules.sonarsource.com/csharp/RSPEC-2436)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2436)

### [Write-only properties should not be used](https://rules.sonarsource.com/csharp/RSPEC-2376)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2376)

### [Exceptions should not be thrown from property getters](https://rules.sonarsource.com/csharp/RSPEC-2372)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2372)

### [Unused type parameters should be removed](https://rules.sonarsource.com/csharp/RSPEC-2326)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2326)

### [Parameters should be passed in the correct order](https://rules.sonarsource.com/csharp/RSPEC-2234)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2234)

### [Two branches in a conditional structure should not have exactly the same implementation](https://rules.sonarsource.com/csharp/RSPEC-1871)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1871)

### [Unused assignments should be removed](https://rules.sonarsource.com/csharp/RSPEC-1854)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1854)

### [Tests should not be ignored](https://rules.sonarsource.com/csharp/RSPEC-1607)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1607)

### ["switch" statements should not have too many "case" clauses](https://rules.sonarsource.com/csharp/RSPEC-1479)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1479)

### [Sections of code should not be commented out](https://rules.sonarsource.com/csharp/RSPEC-125)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-125)

### [Unused method parameters should be removed](https://rules.sonarsource.com/csharp/RSPEC-1172)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1172)

### [Empty arrays and collections should be returned instead of null](https://rules.sonarsource.com/csharp/RSPEC-1168)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1168)

### [Unused private types or members should be removed](https://rules.sonarsource.com/csharp/RSPEC-1144)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1144)

### [Track uses of "FIXME" tags](https://rules.sonarsource.com/csharp/RSPEC-1134)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1134)

### ["Obsolete" attributes should include explanations](https://rules.sonarsource.com/csharp/RSPEC-1123)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1123)

### [Assignments should not be made from within sub-expressions](https://rules.sonarsource.com/csharp/RSPEC-1121)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1121)

### [General exceptions should never be thrown](https://rules.sonarsource.com/csharp/RSPEC-112)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-112)

### [Utility classes should not have public constructors](https://rules.sonarsource.com/csharp/RSPEC-1118)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1118)

### [Local variables should not shadow class fields](https://rules.sonarsource.com/csharp/RSPEC-1117)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1117)

### [Redundant pairs of parentheses should be removed](https://rules.sonarsource.com/csharp/RSPEC-1110)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1110)

### [Inheritance tree of classes should not be too deep](https://rules.sonarsource.com/csharp/RSPEC-110)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-110)

### [Nested blocks of code should not be left empty](https://rules.sonarsource.com/csharp/RSPEC-108)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-108)

### [Methods should not have too many parameters](https://rules.sonarsource.com/csharp/RSPEC-107)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-107)

### [Collapsible "if" statements should be merged](https://rules.sonarsource.com/csharp/RSPEC-1066)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1066)

### [OS commands should not be vulnerable to argument injection attacks](https://rules.sonarsource.com/csharp/RSPEC-5883)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5883)

### [Logging should not be vulnerable to injection attacks](https://rules.sonarsource.com/csharp/RSPEC-5145)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5145)

### [Empty collections should not be accessed or iterated](https://rules.sonarsource.com/csharp/RSPEC-4158)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-4158)

### [Mutable, non-private fields should not be "readonly"](https://rules.sonarsource.com/csharp/RSPEC-3887)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3887)

### ["string.ToCharArray()" should not be called redundantly](https://rules.sonarsource.com/csharp/RSPEC-3456)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3456)

### ["base.Equals" should not be used to check for reference equality in "Equals" if "base" is not "object"](https://rules.sonarsource.com/csharp/RSPEC-3397)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3397)

### [Property assignments should not be made for "readonly" fields not constrained to reference types](https://rules.sonarsource.com/csharp/RSPEC-2934)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2934)

### [Flags enumerations should explicitly initialize all their members](https://rules.sonarsource.com/csharp/RSPEC-2345)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2345)

### ["GetHashCode" should not reference mutable fields](https://rules.sonarsource.com/csharp/RSPEC-2328)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2328)

### [Results of integer division should not be assigned to floating point variables](https://rules.sonarsource.com/csharp/RSPEC-2184)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2184)

### [Integral numbers should not be shifted by zero or more than their number of bits-1](https://rules.sonarsource.com/csharp/RSPEC-2183)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2183)

### ["Equals(Object)" and "GetHashCode()" should be overridden in pairs](https://rules.sonarsource.com/csharp/RSPEC-1206)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-1206)

### [Having a permissive Cross-Origin Resource Sharing policy is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-5122)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-5122)

### [Delivering code in production with debug features activated is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-4507)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-4507)

### [Searching OS commands in PATH is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-4036)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-4036)

### [Creating cookies without the "HttpOnly" flag is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-3330)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-3330)

### [Creating cookies without the "secure" flag is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-2092)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-2092)

### [Literal suffixes should be upper case](https://rules.sonarsource.com/csharp/RSPEC-818)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-818)

### [Null checks should not be used with "is"](https://rules.sonarsource.com/csharp/RSPEC-4201)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4201)

### [Method overloads should be grouped together](https://rules.sonarsource.com/csharp/RSPEC-4136)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4136)

### ["params" should be used instead of "varargs"](https://rules.sonarsource.com/csharp/RSPEC-4061)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4061)

### ["static" fields should be initialized inline](https://rules.sonarsource.com/csharp/RSPEC-3963)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3963)

### [Classes that provide "Equals(<T>)" should implement "IEquatable<T>"](https://rules.sonarsource.com/csharp/RSPEC-3897)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3897)

### [Jump statements should not be redundant](https://rules.sonarsource.com/csharp/RSPEC-3626)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3626)

### [Member initializer values should not be redundant](https://rules.sonarsource.com/csharp/RSPEC-3604)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3604)

### [Unassigned members should be removed](https://rules.sonarsource.com/csharp/RSPEC-3459)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3459)

### [Empty "case" clauses that fall through to the "default" should be omitted](https://rules.sonarsource.com/csharp/RSPEC-3458)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3458)

### [Parameters with "[DefaultParameterValue]" attributes should also be marked "[Optional]"](https://rules.sonarsource.com/csharp/RSPEC-3450)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3450)

### [Interfaces should not simply inherit from base interfaces with colliding members](https://rules.sonarsource.com/csharp/RSPEC-3444)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3444)

### [Variables should not be checked against the values they're about to be assigned](https://rules.sonarsource.com/csharp/RSPEC-3440)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3440)

### [Methods should not return constants](https://rules.sonarsource.com/csharp/RSPEC-3400)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3400)

### [Attribute, EventArgs, and Exception type names should end with the type being extended](https://rules.sonarsource.com/csharp/RSPEC-3376)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3376)

### [Loops should be simplified with "LINQ" expressions](https://rules.sonarsource.com/csharp/RSPEC-3267)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3267)

### [Namespaces should not be empty](https://rules.sonarsource.com/csharp/RSPEC-3261)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3261)

### [Non-derived "private" classes and records should be "sealed"](https://rules.sonarsource.com/csharp/RSPEC-3260)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3260)

### ["string.IsNullOrEmpty" should be used](https://rules.sonarsource.com/csharp/RSPEC-3256)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3256)

### [Implementations should be provided for "partial" methods](https://rules.sonarsource.com/csharp/RSPEC-3251)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3251)

### [Duplicate casts should not be made](https://rules.sonarsource.com/csharp/RSPEC-3247)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3247)

### [Methods should not return values that are never used](https://rules.sonarsource.com/csharp/RSPEC-3241)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3241)

### [Caller information arguments should not be provided explicitly](https://rules.sonarsource.com/csharp/RSPEC-3236)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3236)

### [Method calls should not resolve ambiguously to overloads with "params"](https://rules.sonarsource.com/csharp/RSPEC-3220)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3220)

### ["catch" clauses should do more than rethrow](https://rules.sonarsource.com/csharp/RSPEC-2737)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2737)

### [Generic exceptions should not be ignored](https://rules.sonarsource.com/csharp/RSPEC-2486)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2486)

### [Mutable fields should not be "public static"](https://rules.sonarsource.com/csharp/RSPEC-2386)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2386)

### [Enumeration type names should not have "Flags" or "Enum" suffixes](https://rules.sonarsource.com/csharp/RSPEC-2344)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2344)

### [Enumeration types should comply with a naming convention](https://rules.sonarsource.com/csharp/RSPEC-2342)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2342)

### [Trivial properties should be auto-implemented](https://rules.sonarsource.com/csharp/RSPEC-2292)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2292)

### [Runtime type checking should be simplified](https://rules.sonarsource.com/csharp/RSPEC-2219)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2219)

### [Boolean checks should not be inverted](https://rules.sonarsource.com/csharp/RSPEC-1940)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1940)

### [Inheritance list should not be redundant](https://rules.sonarsource.com/csharp/RSPEC-1939)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1939)

### [Redundant casts should not be used](https://rules.sonarsource.com/csharp/RSPEC-1905)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1905)

### [Strings should not be concatenated using '+' in a loop](https://rules.sonarsource.com/csharp/RSPEC-1643)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1643)

### [Unused local variables should be removed](https://rules.sonarsource.com/csharp/RSPEC-1481)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1481)

### [Private fields only used as local variables in methods should become local variables](https://rules.sonarsource.com/csharp/RSPEC-1450)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1450)

### [A "while" loop should be used instead of a "for" loop](https://rules.sonarsource.com/csharp/RSPEC-1264)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1264)

### ["Equals" and the comparison operators should be overridden when implementing "IComparable"](https://rules.sonarsource.com/csharp/RSPEC-1210)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1210)

### [Nested code blocks should not be used](https://rules.sonarsource.com/csharp/RSPEC-1199)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1199)

### [Overriding members should do more than simply call the same member in the base class](https://rules.sonarsource.com/csharp/RSPEC-1185)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1185)

### ["Any()" should be used to test for emptiness](https://rules.sonarsource.com/csharp/RSPEC-1155)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1155)

### [Boolean literals should not be redundant](https://rules.sonarsource.com/csharp/RSPEC-1125)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1125)

### [Empty statements should be removed](https://rules.sonarsource.com/csharp/RSPEC-1116)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1116)

### [Fields should not have public accessibility](https://rules.sonarsource.com/csharp/RSPEC-1104)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1104)

### [URIs should not be hardcoded](https://rules.sonarsource.com/csharp/RSPEC-1075)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1075)

### [Types should be named in PascalCase](https://rules.sonarsource.com/csharp/RSPEC-101)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-101)

### [Track uses of "TODO" tags](https://rules.sonarsource.com/csharp/RSPEC-1135)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1135)

### [Classes with "IDisposable" members should implement "IDisposable"](https://rules.sonarsource.com/csharp/RSPEC-2931)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2931)

### [Calls to "async" methods should not be blocking](https://rules.sonarsource.com/csharp/RSPEC-4462)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4462)

### [Child class fields should not shadow parent class fields](https://rules.sonarsource.com/csharp/RSPEC-2387)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2387)

### [Track lack of copyright and license headers](https://rules.sonarsource.com/csharp/RSPEC-1451)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1451)

### [Exit methods should not be called](https://rules.sonarsource.com/csharp/RSPEC-1147)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1147)

### [Classes should "Dispose" of members from the classes' own "Dispose" methods](https://rules.sonarsource.com/csharp/RSPEC-2952)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2952)

### [Reading the Standard Input is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-4829)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-4829)

### [Using command line arguments is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-4823)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-4823)

### [Using Sockets is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-4818)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-4818)

### [Encrypting data is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-4787)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-4787)

### [Using regular expressions is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-4784)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-4784)

### [Interface methods should be callable by derived types](https://rules.sonarsource.com/csharp/RSPEC-4039)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4039)

### [Child class fields should not differ from parent class fields only by capitalization](https://rules.sonarsource.com/csharp/RSPEC-4025)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4025)

### [Pointers to unmanaged memory should not be visible](https://rules.sonarsource.com/csharp/RSPEC-4000)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4000)

### [Number patterns should be regular](https://rules.sonarsource.com/csharp/RSPEC-3937)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3937)

### ["out" and "ref" parameters should not be used](https://rules.sonarsource.com/csharp/RSPEC-3874)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3874)

### [Unchanged local variables should be "const"](https://rules.sonarsource.com/csharp/RSPEC-3353)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3353)

### ["ConfigureAwait(false)" should be used](https://rules.sonarsource.com/csharp/RSPEC-3216)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3216)

### ["interface" instances should not be cast to concrete types](https://rules.sonarsource.com/csharp/RSPEC-3215)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3215)

### [Literal boolean values should not be used in assertions](https://rules.sonarsource.com/csharp/RSPEC-2701)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2701)

### [Optional parameters should not be used](https://rules.sonarsource.com/csharp/RSPEC-2360)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2360)

### [Public constant members should not be used](https://rules.sonarsource.com/csharp/RSPEC-2339)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2339)

### [Array covariance should not be used](https://rules.sonarsource.com/csharp/RSPEC-2330)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2330)

### ["nameof" should be used](https://rules.sonarsource.com/csharp/RSPEC-2302)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2302)

### [Modulus results should not be checked for direct equality](https://rules.sonarsource.com/csharp/RSPEC-2197)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2197)

### ["for" loop increment clauses should modify the loops' counters](https://rules.sonarsource.com/csharp/RSPEC-1994)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1994)

### ["switch" statements should not be nested](https://rules.sonarsource.com/csharp/RSPEC-1821)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1821)

### [Methods and properties should not be too complex](https://rules.sonarsource.com/csharp/RSPEC-1541)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1541)

### [Control flow statements "if", "switch", "for", "foreach", "while", "do" and "try" should not be nested too deeply](https://rules.sonarsource.com/csharp/RSPEC-134)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-134)

### ["switch/Select" statements should contain a "default/Case Else" clauses](https://rules.sonarsource.com/csharp/RSPEC-131)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-131)

### ["if ... else if" constructs should end with "else" clauses](https://rules.sonarsource.com/csharp/RSPEC-126)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-126)

### [Control structures should use curly braces](https://rules.sonarsource.com/csharp/RSPEC-121)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-121)

### [Expressions should not be too complex](https://rules.sonarsource.com/csharp/RSPEC-1067)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1067)

### [ASP.NET HTTP request validation feature should not be disabled](https://rules.sonarsource.com/csharp/RSPEC-4564)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-4564)

### [Serialization constructors should be secured](https://rules.sonarsource.com/csharp/RSPEC-4212)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-4212)

### [Calculations should not overflow](https://rules.sonarsource.com/csharp/RSPEC-3949)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-3949)

### [Floating point numbers should not be tested for equality](https://rules.sonarsource.com/csharp/RSPEC-1244)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-1244)

### [Increment (++) and decrement (--) operators should not be used in a method call or mixed with other operators in an expression](https://rules.sonarsource.com/csharp/RSPEC-881)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-881)

### [Use a testable date/time provider.](https://rules.sonarsource.com/csharp/RSPEC-6354)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-6354)

### [Property names should not match get methods](https://rules.sonarsource.com/csharp/RSPEC-4059)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4059)

### [Locales should be set for data types](https://rules.sonarsource.com/csharp/RSPEC-4057)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4057)

### [Literals should not be passed as localized parameters](https://rules.sonarsource.com/csharp/RSPEC-4055)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4055)

### [Operators should be overloaded consistently](https://rules.sonarsource.com/csharp/RSPEC-4050)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4050)

### [Method signatures should not contain nested generic types](https://rules.sonarsource.com/csharp/RSPEC-4017)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4017)

### [Enumeration members should not be named "Reserved"](https://rules.sonarsource.com/csharp/RSPEC-4016)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4016)

### ["System.Uri" arguments should be used instead of strings](https://rules.sonarsource.com/csharp/RSPEC-4005)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4005)

### [Collection properties should be readonly](https://rules.sonarsource.com/csharp/RSPEC-4004)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4004)

### [Disposable types should declare finalizers](https://rules.sonarsource.com/csharp/RSPEC-4002)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4002)

### [String URI overloads should call "System.Uri" overloads](https://rules.sonarsource.com/csharp/RSPEC-3997)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3997)

### [URI properties should not be strings](https://rules.sonarsource.com/csharp/RSPEC-3996)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3996)

### [URI return values should not be strings](https://rules.sonarsource.com/csharp/RSPEC-3995)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3995)

### [URI Parameters should not be strings](https://rules.sonarsource.com/csharp/RSPEC-3994)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3994)

### [Custom attributes should be marked with "System.AttributeUsageAttribute"](https://rules.sonarsource.com/csharp/RSPEC-3993)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3993)

### [Assemblies should explicitly specify COM visibility](https://rules.sonarsource.com/csharp/RSPEC-3992)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3992)

### [Assemblies should be marked as CLS compliant](https://rules.sonarsource.com/csharp/RSPEC-3990)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3990)

### ["Generic.List" instances should not be part of public APIs](https://rules.sonarsource.com/csharp/RSPEC-3956)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3956)

### [Collections should implement the generic interface](https://rules.sonarsource.com/csharp/RSPEC-3909)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3909)

### [Generic event handlers should be used](https://rules.sonarsource.com/csharp/RSPEC-3908)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3908)

### [Event Handlers should have the correct signature](https://rules.sonarsource.com/csharp/RSPEC-3906)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3906)

### ["Assembly.GetExecutingAssembly" should not be called](https://rules.sonarsource.com/csharp/RSPEC-3902)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3902)

### [Arguments of public methods should be validated against null](https://rules.sonarsource.com/csharp/RSPEC-3900)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3900)

### [Value types should implement "IEquatable<T>"](https://rules.sonarsource.com/csharp/RSPEC-3898)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3898)

### [Finalizers should not be empty](https://rules.sonarsource.com/csharp/RSPEC-3880)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3880)

### ["[ExpectedException]" should not be used](https://rules.sonarsource.com/csharp/RSPEC-3431)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3431)

### ["this" should not be exposed from constructors](https://rules.sonarsource.com/csharp/RSPEC-3366)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3366)

### [Types should not have members with visibility set higher than the type's visibility](https://rules.sonarsource.com/csharp/RSPEC-3059)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3059)

### [Fields should be private](https://rules.sonarsource.com/csharp/RSPEC-2357)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2357)

### ["try" statements with identical "catch" and/or "finally" blocks should be merged](https://rules.sonarsource.com/csharp/RSPEC-2327)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2327)

### [NullReferenceException should not be caught](https://rules.sonarsource.com/csharp/RSPEC-1696)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1696)

### [Functions should not have too many lines of code](https://rules.sonarsource.com/csharp/RSPEC-138)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-138)

### ["for" loop stop conditions should be invariant](https://rules.sonarsource.com/csharp/RSPEC-127)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-127)

### [Statements should be on separate lines](https://rules.sonarsource.com/csharp/RSPEC-122)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-122)

### [Classes should not be coupled to too many other classes (Single Responsibility Principle)](https://rules.sonarsource.com/csharp/RSPEC-1200)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1200)

### ["switch case" clauses should not have too many lines of code](https://rules.sonarsource.com/csharp/RSPEC-1151)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1151)

### [Magic numbers should not be used](https://rules.sonarsource.com/csharp/RSPEC-109)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-109)

### [Standard outputs should not be used directly to log anything](https://rules.sonarsource.com/csharp/RSPEC-106)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-106)

### [Files should not have too many lines of code](https://rules.sonarsource.com/csharp/RSPEC-104)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-104)

### [Lines should not be too long](https://rules.sonarsource.com/csharp/RSPEC-103)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-103)

### [HTTP response headers should not be vulnerable to injection attacks](https://rules.sonarsource.com/csharp/RSPEC-5167)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-5167)

### [Console logging should not be used](https://rules.sonarsource.com/csharp/RSPEC-2228)

[Vulnerability](https://rules.sonarsource.com/csharp/RSPEC-2228)

### [Generic parameters not constrained to reference types should not be compared to "null"](https://rules.sonarsource.com/csharp/RSPEC-2955)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2955)

### [The length returned from a stream read should be checked](https://rules.sonarsource.com/csharp/RSPEC-2674)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-2674)

### [Method parameters, caught exceptions and foreach variables' initial values should not be ignored](https://rules.sonarsource.com/csharp/RSPEC-1226)

[Bug](https://rules.sonarsource.com/csharp/RSPEC-1226)

### [Controlling permissions is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-4834)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-4834)

### [Writing cookies is security-sensitive](https://rules.sonarsource.com/csharp/RSPEC-2255)

[Security Hotspot](https://rules.sonarsource.com/csharp/RSPEC-2255)

### [Methods should be named according to their synchronicities](https://rules.sonarsource.com/csharp/RSPEC-4261)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4261)

### [Extensions should be in separate namespaces](https://rules.sonarsource.com/csharp/RSPEC-4226)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4226)

### [Extension methods should not extend "object"](https://rules.sonarsource.com/csharp/RSPEC-4225)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4225)

### [Operator overloads should have named alternatives](https://rules.sonarsource.com/csharp/RSPEC-4069)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4069)

### [Non-abstract attributes should be sealed](https://rules.sonarsource.com/csharp/RSPEC-4060)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4060)

### [Overloads with a "StringComparison" parameter should be used](https://rules.sonarsource.com/csharp/RSPEC-4058)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4058)

### [Overloads with a "CultureInfo" or an "IFormatProvider" parameter should be used](https://rules.sonarsource.com/csharp/RSPEC-4056)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4056)

### [Types should not extend outdated base types](https://rules.sonarsource.com/csharp/RSPEC-4052)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4052)

### [Properties should be preferred](https://rules.sonarsource.com/csharp/RSPEC-4049)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4049)

### [Generics should be used when appropriate](https://rules.sonarsource.com/csharp/RSPEC-4047)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4047)

### [Type names should not match namespaces](https://rules.sonarsource.com/csharp/RSPEC-4041)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4041)

### [Strings should be normalized to uppercase](https://rules.sonarsource.com/csharp/RSPEC-4040)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4040)

### [Exceptions should provide standard constructors](https://rules.sonarsource.com/csharp/RSPEC-4027)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4027)

### [Assemblies should be marked with "NeutralResourcesLanguageAttribute"](https://rules.sonarsource.com/csharp/RSPEC-4026)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4026)

### [Interfaces should not be empty](https://rules.sonarsource.com/csharp/RSPEC-4023)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4023)

### [Enumerations should have "Int32" storage](https://rules.sonarsource.com/csharp/RSPEC-4022)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4022)

### [Generic methods should provide type parameters](https://rules.sonarsource.com/csharp/RSPEC-4018)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-4018)

### [Multidimensional arrays should not be used](https://rules.sonarsource.com/csharp/RSPEC-3967)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3967)

### ["static readonly" constants should be "const" instead](https://rules.sonarsource.com/csharp/RSPEC-3962)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3962)

### [Strings or integral types should be used for indexers](https://rules.sonarsource.com/csharp/RSPEC-3876)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3876)

### [Parameter names should not duplicate the names of their methods](https://rules.sonarsource.com/csharp/RSPEC-3872)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3872)

### [Track use of "NotImplementedException"](https://rules.sonarsource.com/csharp/RSPEC-3717)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3717)

### [Empty "default" clauses should be removed](https://rules.sonarsource.com/csharp/RSPEC-3532)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3532)

### [Redundant property names should be omitted in anonymous classes](https://rules.sonarsource.com/csharp/RSPEC-3441)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3441)

### [Declarations and initializations should be as concise as possible](https://rules.sonarsource.com/csharp/RSPEC-3257)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3257)

### [Default parameter values should not be passed as arguments](https://rules.sonarsource.com/csharp/RSPEC-3254)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3254)

### [Constructor and destructor declarations should not be redundant](https://rules.sonarsource.com/csharp/RSPEC-3253)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3253)

### [Method parameters should be declared with base types](https://rules.sonarsource.com/csharp/RSPEC-3242)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3242)

### [The simplest possible condition syntax should be used](https://rules.sonarsource.com/csharp/RSPEC-3240)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3240)

### [Redundant parentheses should not be used](https://rules.sonarsource.com/csharp/RSPEC-3235)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3235)

### ["GC.SuppressFinalize" should not be invoked for types without destructors](https://rules.sonarsource.com/csharp/RSPEC-3234)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3234)

### [Members should not be initialized to default values](https://rules.sonarsource.com/csharp/RSPEC-3052)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-3052)

### [Sequential tests should not check the same condition](https://rules.sonarsource.com/csharp/RSPEC-2760)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2760)

### [Redundant modifiers should not be used](https://rules.sonarsource.com/csharp/RSPEC-2333)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2333)

### [Methods and properties that don't access instance data should be static](https://rules.sonarsource.com/csharp/RSPEC-2325)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2325)

### ["Exception" should not be caught when not required by called methods](https://rules.sonarsource.com/csharp/RSPEC-2221)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2221)

### ["sealed" classes should not have "protected" members](https://rules.sonarsource.com/csharp/RSPEC-2156)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2156)

### [Underscores should be used to make large numbers readable](https://rules.sonarsource.com/csharp/RSPEC-2148)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-2148)

### ["ToString()" calls should not be redundant](https://rules.sonarsource.com/csharp/RSPEC-1858)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1858)

### ["==" should not be used when "Equals" is overridden](https://rules.sonarsource.com/csharp/RSPEC-1698)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1698)

### [An abstract class should have both abstract and concrete methods](https://rules.sonarsource.com/csharp/RSPEC-1694)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1694)

### [Multiple variables should not be declared on the same line](https://rules.sonarsource.com/csharp/RSPEC-1659)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1659)

### [Culture should be specified for "string" operations](https://rules.sonarsource.com/csharp/RSPEC-1449)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1449)

### ["switch" statements should have at least 3 "case" clauses](https://rules.sonarsource.com/csharp/RSPEC-1301)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1301)

### [break statements should not be used except for switch cases](https://rules.sonarsource.com/csharp/RSPEC-1227)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1227)

### [String literals should not be duplicated](https://rules.sonarsource.com/csharp/RSPEC-1192)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1192)

### [Files should contain an empty newline at the end](https://rules.sonarsource.com/csharp/RSPEC-113)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-113)

### [Unused "using" should be removed](https://rules.sonarsource.com/csharp/RSPEC-1128)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1128)

### [A close curly brace should be located at the beginning of a line](https://rules.sonarsource.com/csharp/RSPEC-1109)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1109)

### [Tabulation characters should not be used](https://rules.sonarsource.com/csharp/RSPEC-105)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-105)

### [Methods and properties should be named in PascalCase](https://rules.sonarsource.com/csharp/RSPEC-100)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-100)

### [Track uses of in-source issue suppressions](https://rules.sonarsource.com/csharp/RSPEC-1309)

[Code Smell](https://rules.sonarsource.com/csharp/RSPEC-1309)