**Static Code Analysis**

[**https://www.milanjovanovic.tech/blog/improving-code-quality-in-csharp-with-static-code-analysis**](https://www.milanjovanovic.tech/blog/improving-code-quality-in-csharp-with-static-code-analysis)

It’s a way to examine your code without running it.

It reports any issue related to security, performance, coding style, or best practices.

* The best way I found to configure static code analysis is using **Directory.Build.props**
* You can place the **Directory.Build.props** file in the root folder so it will apply to all projects.
* Here are some properties we can configure:

**TreatWarningsAsErrors** - Treat all warnings as errors.

**CodeAnalysisTreatWarningsAsErrors** - Treat code quality (CAxxxx) warnings as errors.

**EnforceCodeStyleInBuild** - Enables code-style analysis ("IDExxxx") rules.

**AnalysisLevel** - Specifies which analyzers to enable. The default value is the **latest**.

**AnalysisMode** - Configures the predefined code analysis configuration.

* We can install another Nuget package

**SonarAnalyzer.CSharp** contains an additional code analyzer to help us write clean, safe, and reliable code.

<Project>

<PropertyGroup>

<TargetFramework>net8.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

<!-- Configure code analysis. -->

<AnalysisLevel>latest</AnalysisLevel>

<AnalysisMode>All</AnalysisMode>

<TreatWarningsAsErrors>true</TreatWarningsAsErrors>

<CodeAnalysisTreatWarningsAsErrors>true</CodeAnalysisTreatWarningsAsErrors>

<EnforceCodeStyleInBuild>true</EnforceCodeStyleInBuild>

</PropertyGroup>

<ItemGroup Condition="'$(MSBuildProjectExtension)' != '.dcproj'">

<PackageReference Include="SonarAnalyzer.CSharp" Version="\*">

<PrivateAssets>all</PrivateAssets>

<IncludeAssets>

runtime; build; native; contentfiles; analyzers; buildtransitive

</IncludeAssets>

</PackageReference>

</ItemGroup>

</Project>