

Improving code quality with Roslyn Static Analysis

Jim Wooley
Solution Architect,
Slalom

About Jim



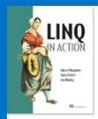
Solution Architect

Slalom Consulting



Microsoft MVP

Visual Studio Tools



Author

LINQ in Action



@JimWooley

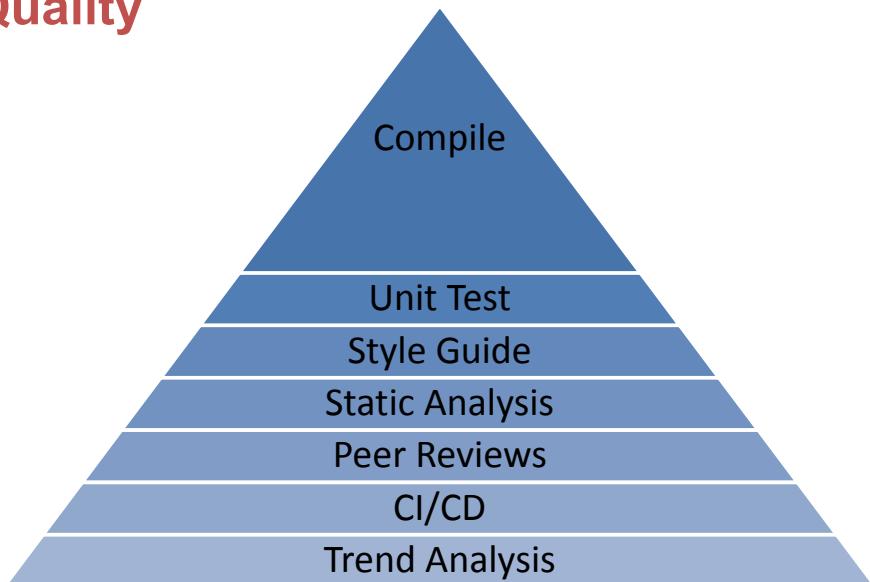
<https://jwooley.github.io>

<https://github.com/jwooley/Analyzers>

<http://www.thinqlinq.com>

2

Code Quality

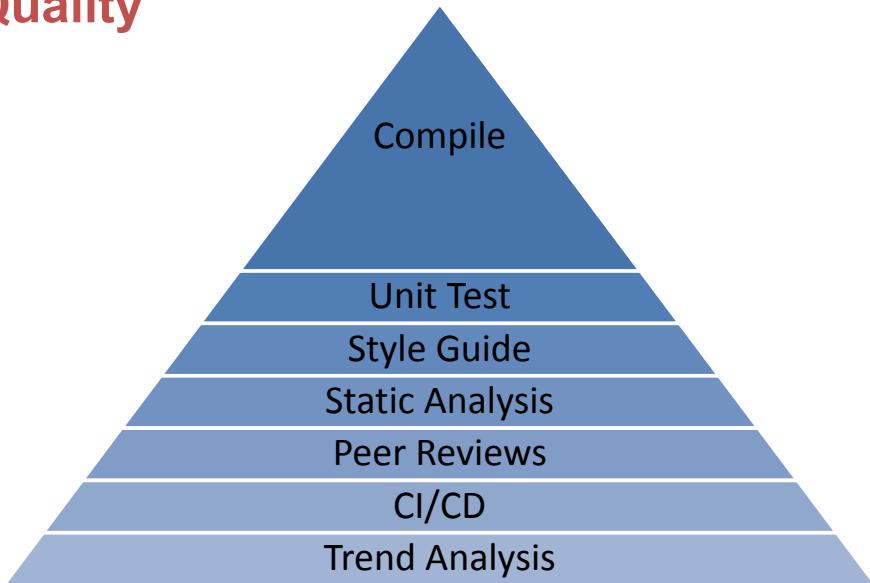


Visual Studio Live! Chicago 2017



TH18 - Improving Code Quality with Roslyn Code Analyzers - Jim Wooley

Code Quality



“

“Static program analysis is the analysis of computer software that is performed without actually executing programs.”

Wikipedia

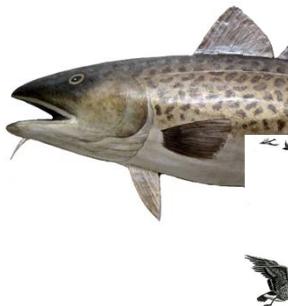
https://en.wikipedia.org/wiki/Static_program_analysis

Why use

- Standardize within team
- Improve maintainability
- Allow code review to focus on domain/logic issues
- Detect common errors not caught by compiler
- Fix code smells



Fixing Cod~~e~~ Smelts



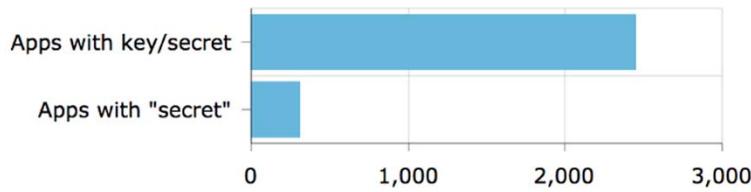
```
void ProcessCustomers(dynamic customers)
{
    if (customers != null) {
        foreach(var c in customers)
        {
            DoSomething();
            foreach (var o in c.Orders)
            {
                DoSomethingComplex(o);
                foreach (var d in o.OrderDetails)
                {
                    ProcessOrder(d);
                }
            }
        }
    }
}

void ProcessBetter(dynamic customers)
{
    if (customers == null) return;
    foreach (var c in customers)
    {
        DoSomethingComplex(c);
        foreach (var o in c.Orders)
        {
            ProcessOrder(o);
        }
    }
}

void ProcessOrder(dynamic order)
{
    DoSomethingComplex(order);
    if (order.total <= 1000)
    {
        DoSomethingComplex(order.OrderDetails);
        return;
    }
}
```



Detect security vulnerabilities



Out of 16,000 apps—apps with keys or secrets

<https://hackernoon.com/we-reverse-engineered-16k-apps-heres-what-we-found-51bdf3b456bb#.eechvyizf>



Detect domain specific errors

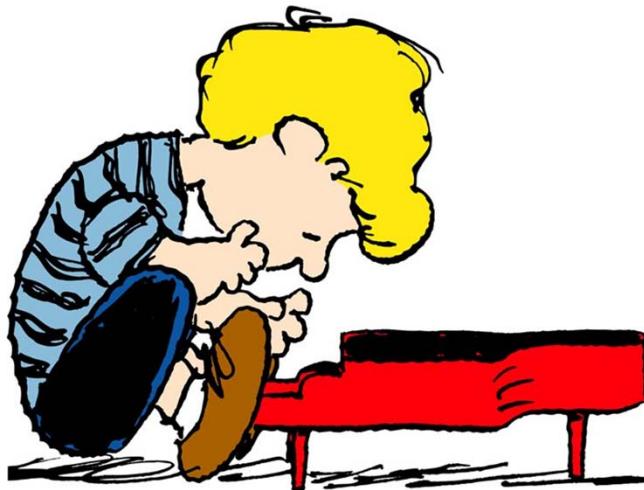
```
public class TestControllerMisnamed : Controller
{
    public TestControllerMisnamed()
    {

    }

    // GET: Test
    public ActionResult Index()
    {
        return View();
    }
}
```

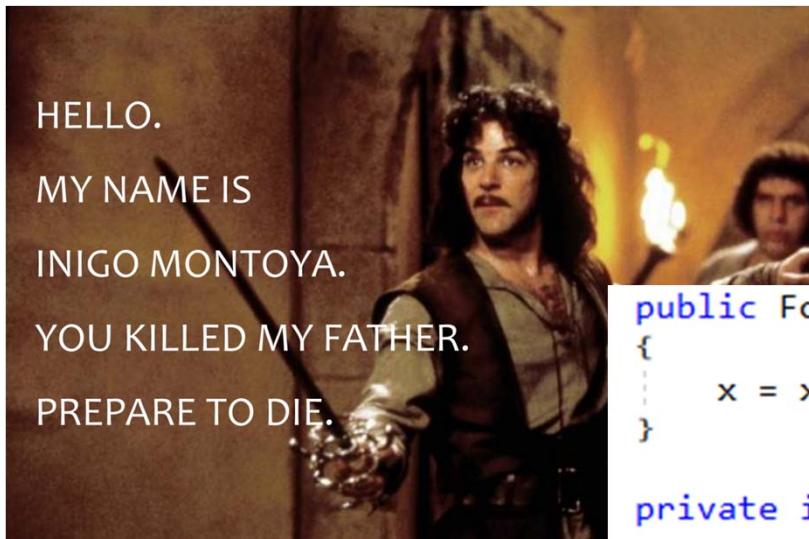


Recommend best practices



Visual Studio **LIVE!**
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Names matter



Visual Studio **LIVE!**
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Detect maintainability issues

Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
One or more projects were skipped. Code metrics are available only for C#, Visu					
ApiDll (Debug)	69	1,945	4	1,233	12,341
APITests (Debug)	75	1,009	5	533	3,918
Common (Debug)	73	660	3	263	1,232
CommonControls (Debug)	84	223	1	104	471
Data (Debug)	92	2,837	2	211	3,412
Documentation\Modelling (Debug)	84	3,249	2	80	5,104
Email (Debug)	68	25	1	28	53
InternalManagementAppWeb (Debug)	86	1,427	4	648	3,187
KofaxImport\AzureUpload (Debug)	85	234	9	126	573
KofaxImport\AzureUpload.Core (Debug)	76	99	1	34	174
KofaxImport\FileImport (Debug)	75	97	1	80	319
LetterModels (Debug)	93	129	1	13	131
Models (Debug)	93	3,159	2	95	3,271
Tests (Debug)	83	234	1	99	483
WebJobs\AuditJob (Debug)	71	93	1	114	713
WebJobs\NivDocCleanup (Debug)	86	26	1	39	36
WebJobs\ScheduledTasks (Debug)	86	20	1	23	38
WebJobs\SendEmails (Debug)	89	16	1	15	21



Recommend modernized syntax

The screenshot shows two snippets of C# code. The left snippet is:

```
DateTime? toReturn = null;
DateTime output;
if(DateTime.TryParse(s, out output))
{
    toReturn = output;
}
return toReturn;
```

The right snippet is the recommended modernized syntax:

```
DateTime? toReturn = null;
if (DateTime.TryParse(s, out DateTime output))
{
    toReturn = output;
}
return toReturn;
```

Below the code, a tooltip from the IDE indicates a warning: "IDE0018 Variable declaration can be inlined". The tooltip also provides a preview of the refactored code.

16

Demo

Using Visual Studio built in analyzers

Extensible and Customizable

ID	Name	Action
CC0001	You should use 'var' whenever possible.	Multiple
CC0002	Invalid argument name	Warning
CC0003	Your catch maybe include some Exception	Warning
CC0004	Catch block cannot be empty	Warning
CC0005	Empty Object Initializer	Warning
CC0006	Use foreach	Warning
CC0007	Return Condition directly	Warning
CC0008	Use object initializer	Warning
CC0009	Use object initializer	Warning
CC0010	Your Regex expression is wrong	Error
CC0011	You should remove the 'Where' invocation when it is possible.	Warning
CC0012	Your throw does nothing	Warning
CC0013	Use ternary operator	Info
CC0014	Use ternary operator	Info
CC0015	Unnecessary Parenthesis	Warning
CC0016	Copy Event To Variable Before Fire	Warning
CC0017	Use auto property	Info
CC0018	Use the existence operator	Info
CC0019	Use 'switch'	Info
CC0020	You should remove the lambda expression when it only invokes a method with the same signature	Hidden
CC0021	Use nameof	Warning
CC0022	Should dispose object	Warning
CC0023	Unsealed Attribute	Warning
CC0024	Don't throw exception inside static constructors.	Warning
CC0025	Remove Empty Finalizers	Warning
CC0026	Call Extension Method As Extension	Info

.editorconfig

```
[*]
end_of_line = crlf

[*.{cs,py,xmlm}]
indent_style = tab

[*.js]
indent_style = space
indent_size = 2

[*.cs]
dotnet_style_qualification_for_field = false:suggestion
dotnet_style_qualification_for_property = false:suggestion
dotnet_style_qualification_for_method = false:suggestion
dotnet_style_qualification_for_event = false:suggestion
dotnet_style_predefined_type_for_locals_parameters_members = true:suggestion
dotnet_style_predefined_type_for_member_access = true:suggestion
dotnet_style_object_initializer = true:suggestion
dotnet_style_collection_initializer = true:suggestion
dotnet_style_explicit_tuple_names = true:suggestion
dotnet_style_coalesce_expression = true:suggestion
dotnet_style_null_propagation = true:suggestion
csharp_style_var_for_builtin_types = true:suggestion

https://docs.microsoft.com/en-us/visualstudio/ide/editorconfig-code-style-settings-reference
```

Adding Analyzer from Nuget

The screenshot shows the NuGet Package Manager interface. On the left, a search bar contains 'roslyn'. Below it, a list of packages includes:

- RoslynSecurityGuard by Security Guard Team, 577 downloads
- VarAnalyzer by Francesco Bonacci, 125 downloads
- ExceptionAnalyzer by Sergey Teplyakov, 2,88K downloads
- StyleCop.Analyzers by Sam Harwell et. al., 466K downloads
- WpfAnalyzers by Johan Larsson, 5,39K downloads
- Microsoft.CodeDom.Providers.DotNetCompilerPlatform.BinFix by dos, 12.2K downloads
- FileHelpers.Analyzer by Marcos Meli, 3,17K downloads
- codecracker.CSharp** by gigo, elemarj, carloscds, 17.7K downloads
- CodeFreud by Joe Kuermerle, 1,43K downloads

On the right, the details page for 'codecracker.CSharp' is shown. It includes the following information:

- codecracker.CSharp**
- Installed: 1.0.0-rc6
- Version: Latest stable 1.0.2
- Description: A analyzer library for C# that uses Roslyn to produce refactorings, code analysis, and other niceties.
- Author(s): gigo, elemarj, carloscds
- License: <https://github.com/code-cracker/code-cracker/blob/master/LICENSE.txt>
- Date published: Sunday, March 12, 2017 (3/12/2017)
- Project URL: <http://code-cracker.github.io/>
- Report Abuse: <https://www.nuget.org/packages/codcracker.CSharp/1.0.2/ReportAbuse>
- Tags: analyzers, roslyn
- Dependencies: No dependencies

Demo

Using third party analyzers

Detect domain specific errors

```
public class TestControllerMisnamed : Controller
{
    public TestControllerMisnamed()
    {

    }

    // GET: Test
    public ActionResult Index()
    {
        return View();
    }
}
```



Creating your own Roslyn analyzer and Code Fix

```
private void Analyzer(SymbolAnalysisContext context)
{
    var symbol = (INamedTypeSymbol)context.Symbol;
    if (symbol.BaseType == null) return;

    if ((symbol.BaseType.Name == "Controller"
        || symbol.BaseType.Name == "ApiController")
        && !symbol.Name.EndsWith("Controller"))
    {
        var diagnostic = Diagnostic.Create(
            Rule, symbol.Locations[0], symbol.Name);
        context.ReportDiagnostic(diagnostic);
    }
}
```

© 2017 Microsoft Corporation. All rights reserved.

Custom Diagnostic - CodeFixProvider

```
private async Task<Document> MakeEndInControllerAsync(
    Document document,
    TypeDeclarationSyntax typeDecl,
    CancellationToken cancellationToken)
{
    var identifierToken = typeDecl.Identifier;
    var originalName = identifierToken.Text;

    var nameWithoutController = Regex.Replace(
        originalName, "controller", String.Empty, RegexOptions.IgnoreCase);
    var newName = nameWithoutController + "Controller";

    var newIdentifier = SyntaxFactory.Identifier(newName)
        .WithAdditionalAnnotations(Formatter.Annotation);

    var newDeclaration = typeDecl.ReplaceToken(identifierToken, newIdentifier);

    var root = await document.GetSyntaxRootAsync();
    var newRoot = root.ReplaceNode(typeDecl, newDeclaration);
    var newDocument = document.WithSyntaxRoot(newRoot);
    return newDocument;
}
```

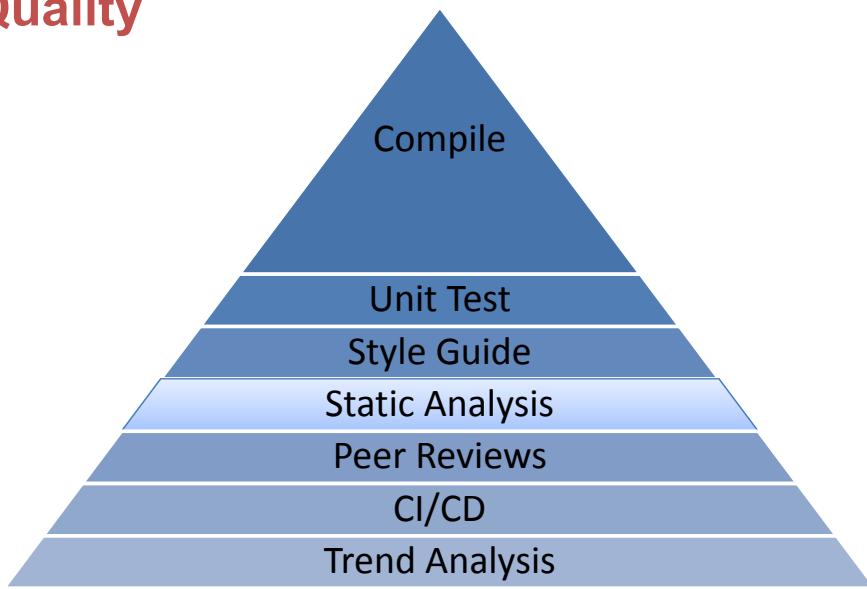


Key Principles

- Start Early
- Run Often
- Customize for your needs
- Augment SDLC
- Focus Code Review on domain issues not code smells

Visual Studio **LIVE!**
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Code Quality





ROCK YOUR CODE
TOUR 2017

CHICAGO

Improving code quality with Roslyn Static Analysis

[HTTPS://GITHUB.COM/JWOOLEY/ANALYZERS](https://github.com/jwooley/analyzers)

Jim Wooley
www.ThinqLinq.com
<https://jwooley.github.io/>
Twitter: @JimWooley