

Visual Studio **LIVE!** EXPERT SOLUTIONS FOR .NET DEVELOPERS | San Diego

Improving code quality with Static Analysis

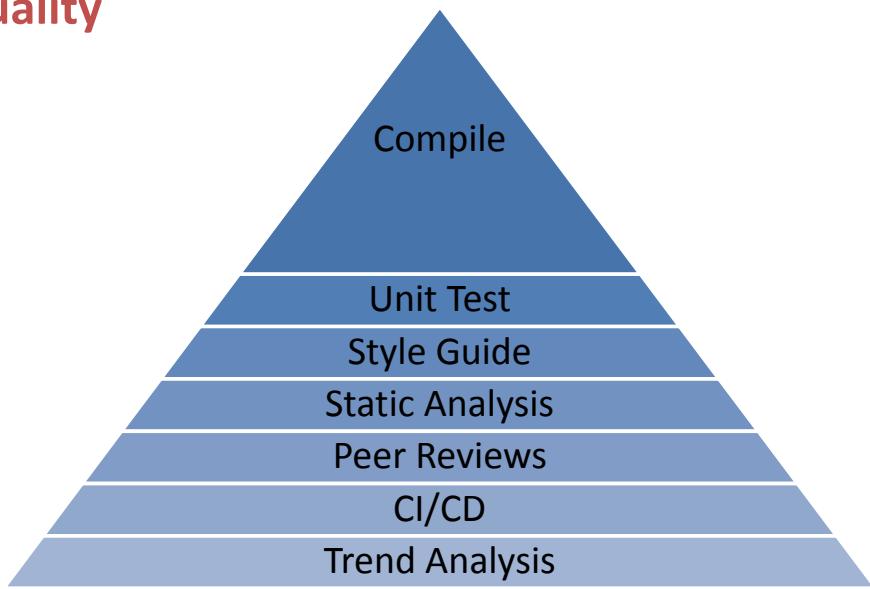
Jim Wooley
Solution Architect
Slalom

Level: Intermediate, etc.

Code Again for the First Time!

Visual Studio 25 YEARS OF CODING INNOVATION

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



Visual Studio Live! San Diego 2018





Fixing Cod~~e~~ Smelts



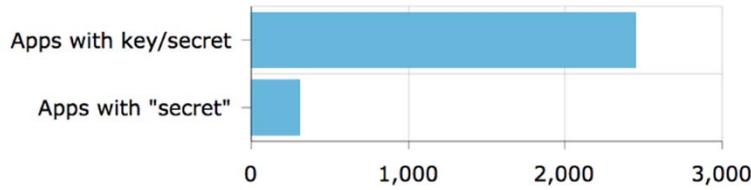
```
void ProcessCustomers(dynamic customers)
{
    if (customers != null) {
        foreach(var c in customers)
        {
            DoSomething();
            foreach (var o in c.Orders)
            {
                DoSomethingComplex(o);
                foreach (var o in c.Orders)
              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

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

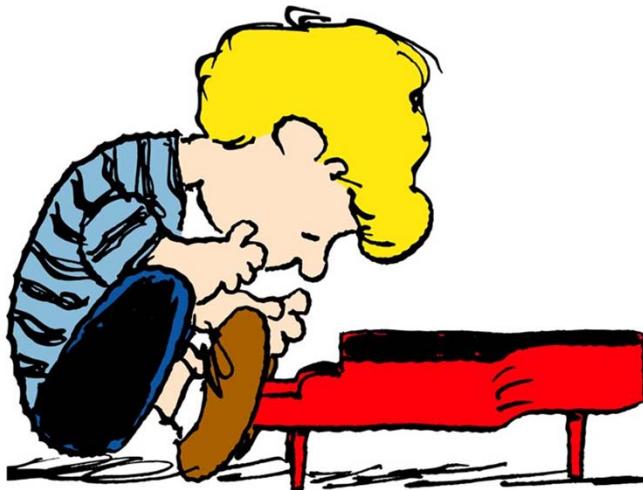
```
public class TestControllerMisnamed : Controller
{
    public TestControllerMisnamed()
    {

    }

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



Recommend best practices

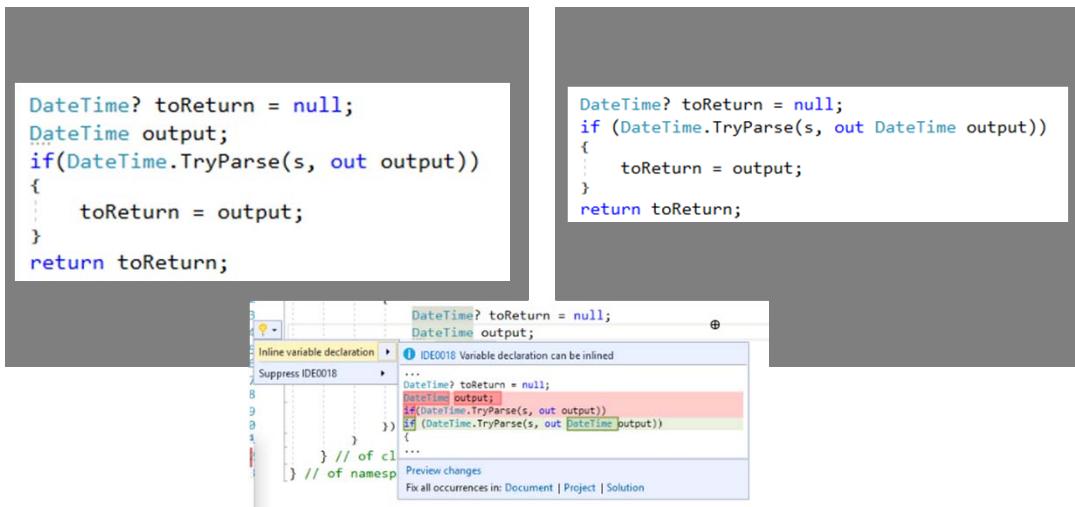


Detect maintainability issues

Code Metrics Results						
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#, Visual Basic, and F#.						
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\Modeling (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



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

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

13

Tools

- JSLint/JSHint
- CssLint/ScssLint/PostCSS/StyleLint
- SonarQube/Sonargraph/SonarLint
- Roslyn/FxCop/StyleCop/Code Cracker
- Code Rush/Resharper
- Your favorite tool?

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



JavaScript – JsLint/JsHint

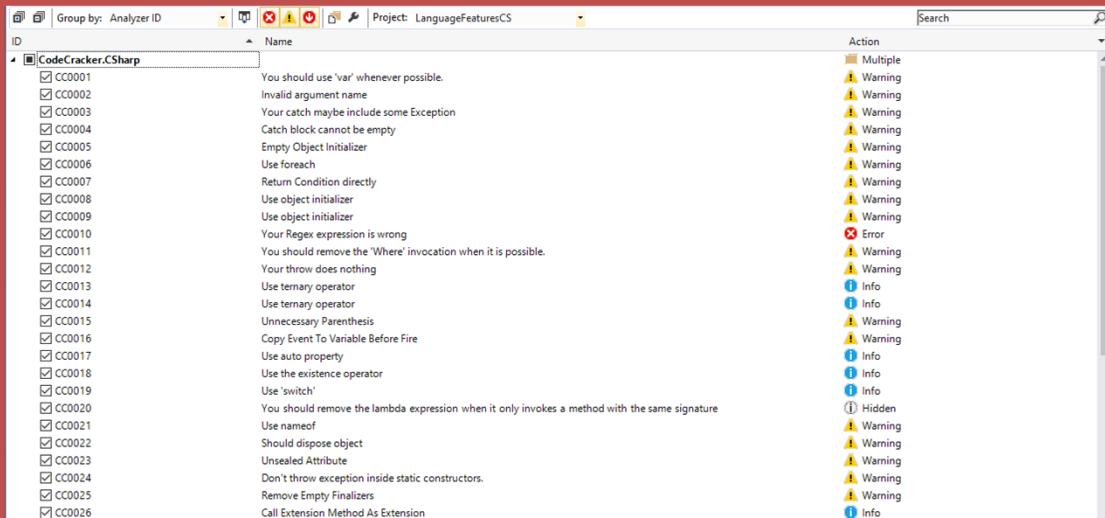
```
function main() {          1  function main() {          function main() {
    return 'Hello, World!';  2    return 'Hello, World!';      "use strict";
    var x = 1;              3    var x = 1;                  var x = 1;
    if (x == "2")           4    if (x == "2")            if (x === 2) {
        {                   5    {                      console.log(x);
            console.log(X)   6    console.log(X)         }
        }                   7    }                     }
    }                      8  }                     }
```

YEARS OF CODING INNOVATION

Demo

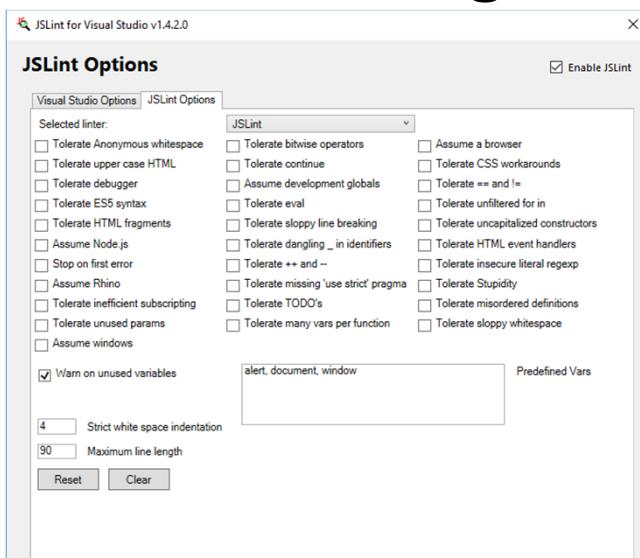
JSParser, JSLint, CSSLint, Code Cracker

Extensible and Customizable



slalom.com

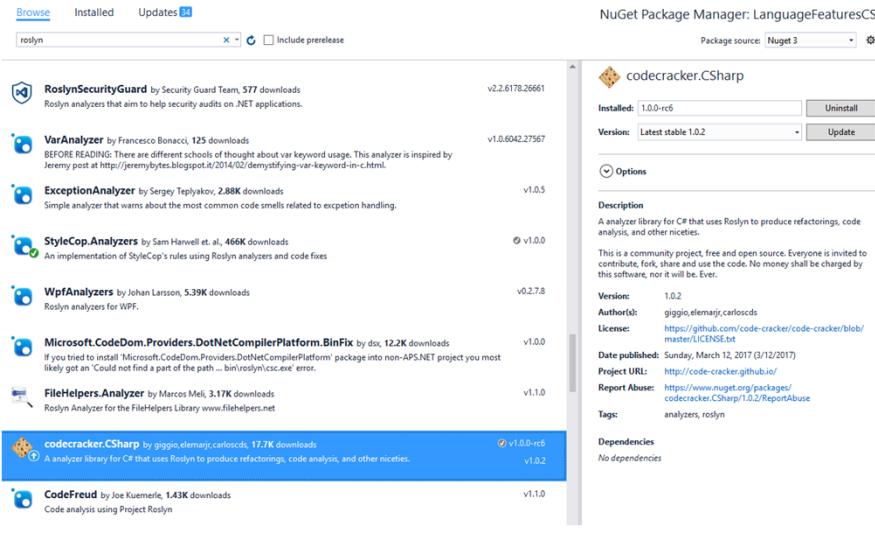
Configuring jsLint



```
.jshintrc
{
  "undef": true,
  "unused": true,
  "globals": {
    "MY_GLOBAL": true
  }
}
```



Adding Analyzer from Nuget

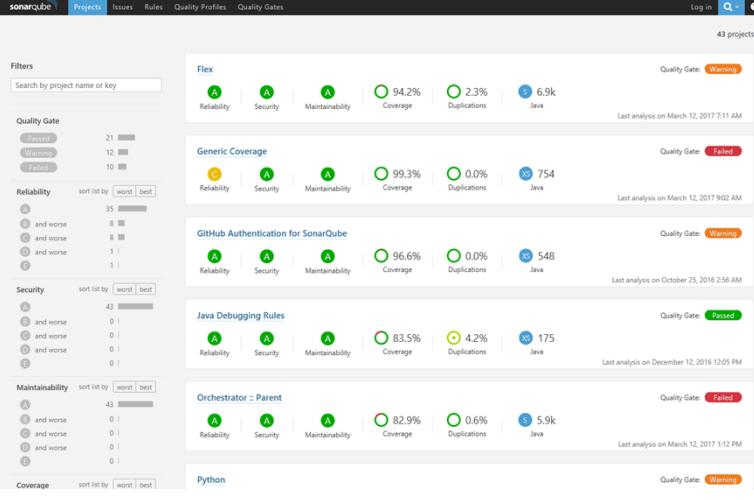


The screenshot shows the NuGet Package Manager interface. The search bar at the top has 'roslyn' typed into it. Below the search bar, there are three tabs: 'Browse', 'Installed', and 'Updates'. The 'Updates' tab is selected. On the right side of the interface, there is a detailed view of the 'codecracker.CSharp' package. The package details include:

- 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):** gigio, 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/codecracker.CSharp/1.0.2/reportabuse>
- Tags:** analyzers, roslyn

The package details also mention that it has no dependencies. At the bottom right of the interface, there is a Visual Studio 25th anniversary logo.

SonarQube



The screenshot shows the SonarQube interface. The top navigation bar includes 'sonarqube', 'Projects', 'Issues', 'Rules', 'Quality Profiles', and 'Quality Gates'. The main area displays various quality gate metrics for different projects:

- Flex:** Quality Gate: Warning. Metrics: Reliability (A), Security (A), Maintainability (A), Coverage (94.2%), Duplications (2.3%), Java (6.9k). Last analysis on March 12, 2017 7:11 AM.
- Generic Coverage:** Quality Gate: Failed. Metrics: Reliability (B), Security (A), Maintainability (A), Coverage (99.3%), Duplications (0.0%), Java (754). Last analysis on March 12, 2017 9:02 AM.
- GitHub Authentication for SonarQube:** Quality Gate: Warning. Metrics: Reliability (A), Security (A), Maintainability (A), Coverage (96.6%), Duplications (0.0%), Java (548). Last analysis on October 25, 2016 2:56 AM.
- Java Debugging Rules:** Quality Gate: Passed. Metrics: Reliability (A), Security (A), Maintainability (A), Coverage (83.5%), Duplications (4.2%), Java (175). Last analysis on December 12, 2016 12:05 PM.
- Orchestrator - Parent:** Quality Gate: Failed. Metrics: Reliability (A), Security (A), Maintainability (A), Coverage (82.9%), Duplications (0.6%), Java (5.9k). Last analysis on March 12, 2017 1:12 PM.
- Python:** Quality Gate: Warning.

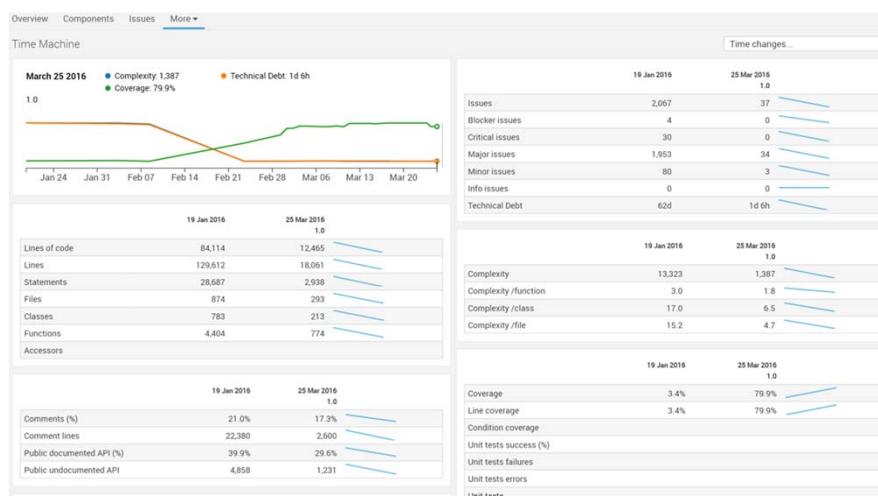
On the left side, there are filters for 'Projects' (43 projects), 'Quality Gate' (Passed: 21, Warning: 12, Failed: 10), 'Reliability' (sort by worst/best), 'Security' (sort by worst/best), 'Maintainability' (sort by worst/best), and 'Coverage' (sort by worst/best).

Sonar Rules

The screenshot shows the SonarQube Rules interface. At the top, there are tabs for sonarqube, Projects, Issues, Rules (which is selected), Quality Profiles, and Quality Gates. Below the tabs, there's a search bar and a sidebar with filters for Language (Java, Groovy, C++, C, Objective-C, C#, JavaScript), Type (Bug, Vulnerability, Code Smell), and Tag (Repository, Default Severity, Status, Available Since, Template, Quality Profile, Inheritance, Activation Severity). The main area displays a list of rules, each with a title, count, language, type, and severity. For example, there are 414 Java rules, 353 Groovy rules, and 1,161 Code Smell rules.



Sonar improvements



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);
    }
}
```



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;
}
```



Create ESLint rule

```
module.exports = {
    meta: {
        docs: {
            description: "disallow unnecessary semicolons",
            category: "Possible Errors",
            recommended: true
        },
        fixable: "code",
        schema: [] // no options
    },
    create: function(context) {
        return {
            // callback functions http://eslint.org/docs/developer-guide/working-with-rules
        };
    }
};
```

<http://eslint.org/docs/developer-guide/working-with-rules>



Key Principals

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



“The key takeaway is to use the tools available to find any defect you can – for even if you don’t, your adversaries will.”

- Travis Smith

Q & A

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

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

<https://jwooley.github.io>

<http://www.thingling.com>

<http://www.twitter.com/jimwooley>

