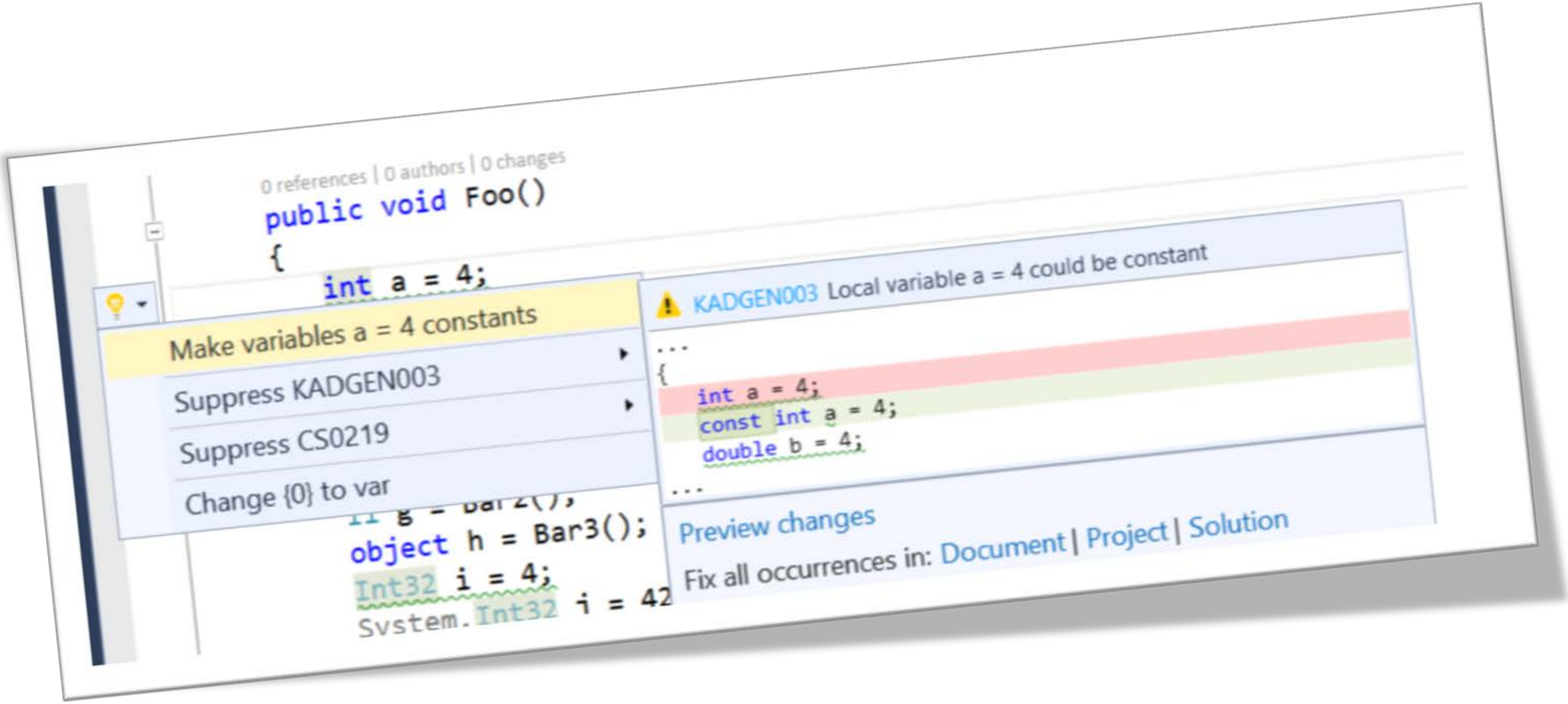
Diagnostic Analyzers in Visual Studio 2015: First Look

Using Diagnostic Analyzers

Kathleen Dollard

@kathleendollard | www.CodeRapid.com

Diagnostic Analyzer and Code Fix



```
0 references | 0 authors | 0 changes
public void Foo()
{
    int a = 4;
    ...
    object h = Bar3();
    Int32 i = 4;
    System.Int32 i = 42
```

⚠ KADGEN003 Local variable a = 4 could be constant

```
...
{
    int a = 4;
    const int a = 4;
    double b = 4;
...
```

Make variables a = 4 constants
Suppress KADGEN003
Suppress CS0219
Change {0} to var
Preview changes
Fix all occurrences in: Document | Project | Solution

What’s a diagnostic analyzer?

What’s a code fix?

What’s a refactoring?

How will these improve my code?

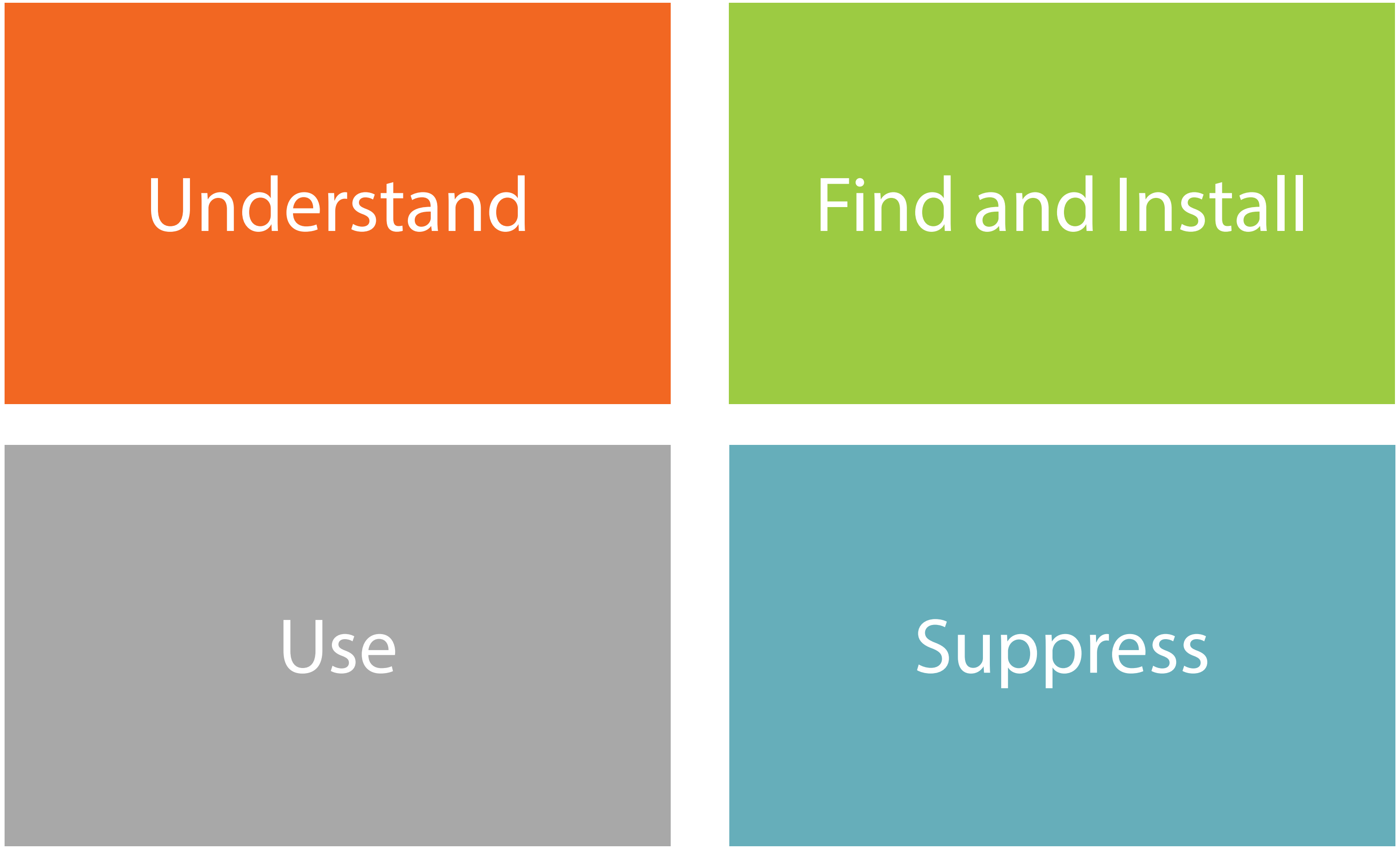
Can I really build them?

How hard are they to build?

Can I test them?

What guidelines should I follow?

VS 2015?



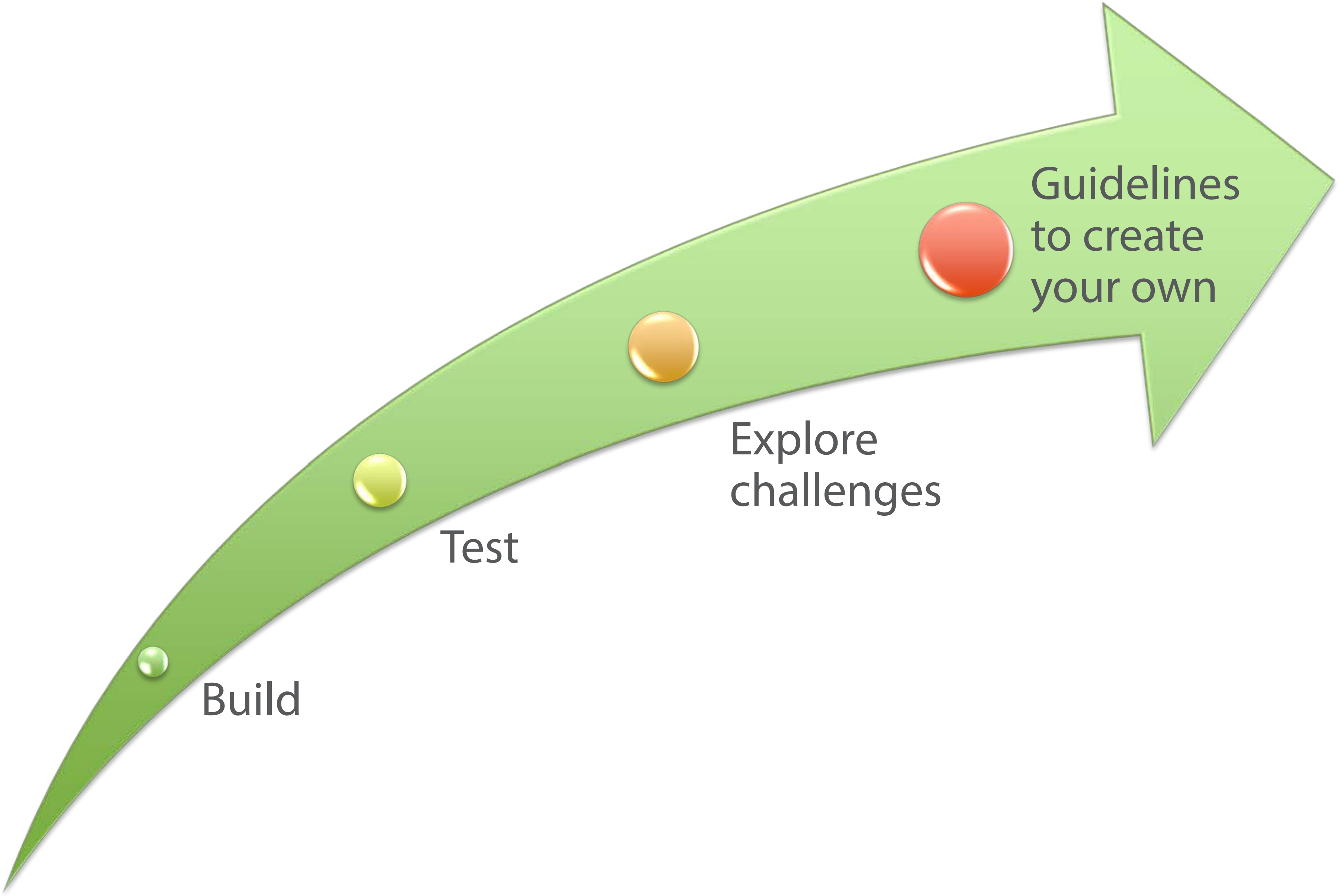
Understand

Find and Install

Use

Suppress

pluralsight



Build

Test

Explore challenges

Guidelines to create your own

Visual Studio®

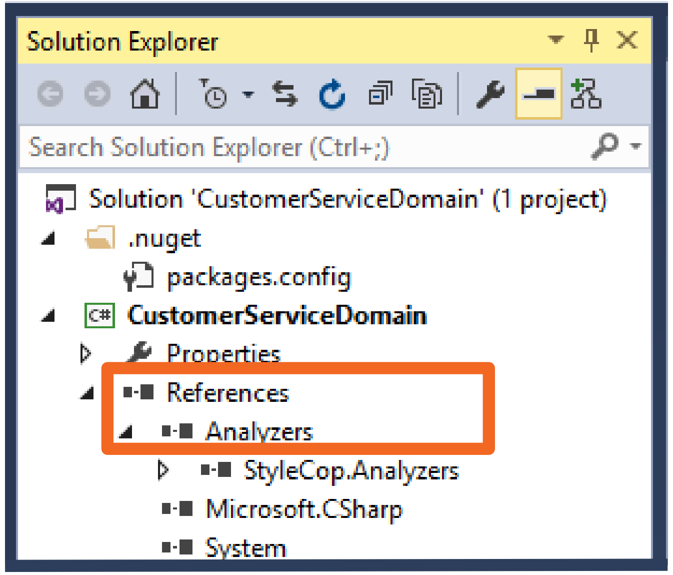
2015 CTP

Best practices

Team guidelines

pluralsight

Per Project



Solution Explorer

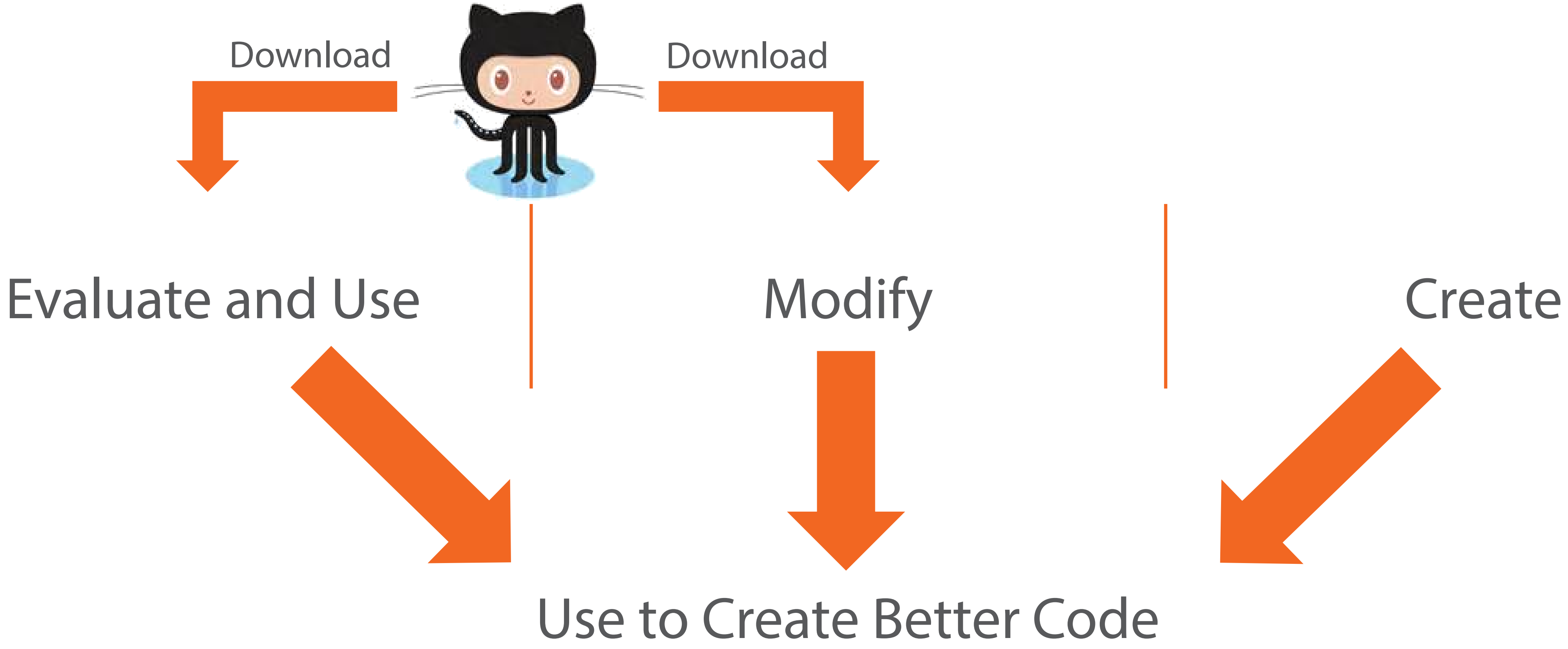
Search Solution Explorer (Ctrl+;)

```
Solution 'CustomerServiceDomain' (1 project)
  .nuget
    packages.config
  CustomerServiceDomain
    Properties
    References
      Analyzers
        StyleCop.Analyzers
      Microsoft.CSharp
      System
```

Per Machine

pluralsight

Learning About Diagnostics



Download

Download

Evaluate and Use

Modify

Create

Use to Create Better Code

pluralsight

Issues with Tools like FxCop

Enormous set

Mixed importance

Limited Customization

Run late in cycle

Too many messages

Manual step

No editor feedback

Diagnostic Analyzers and Refactorings

Under your control

Integrated with Visual Studio

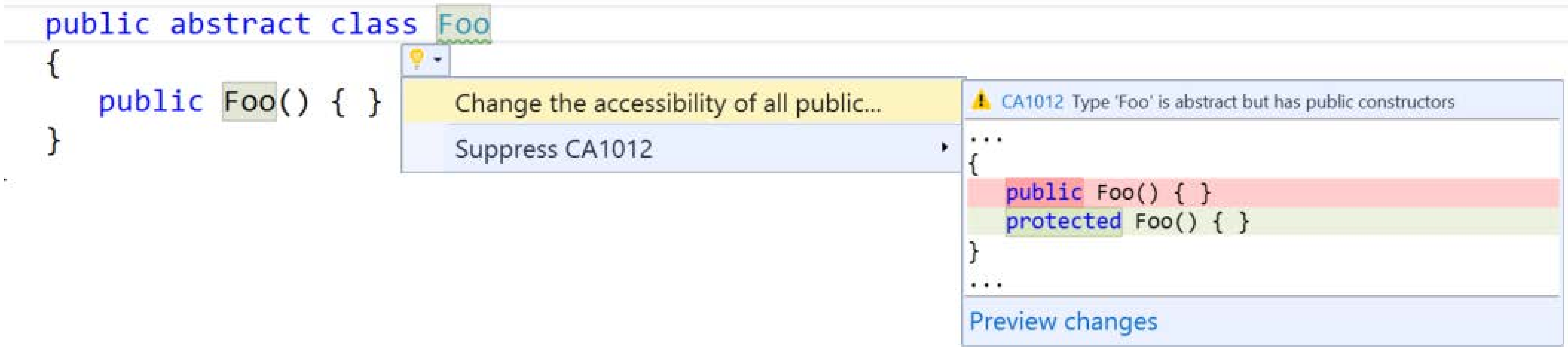
Uses Roslyn compiler API

Lightweight

pluralsight

Results Appear Familiar Locations

Project and Machine

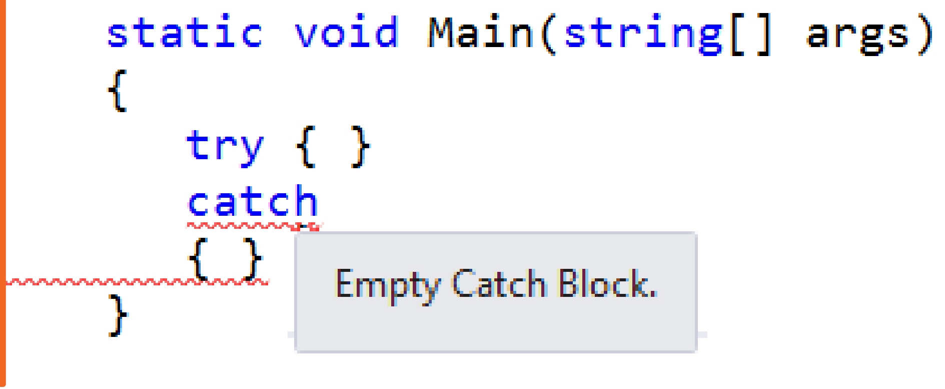


```
public abstract class Foo
{
    public Foo() { }
}
```

CA1012 Type 'Foo' is abstract but has public constructors

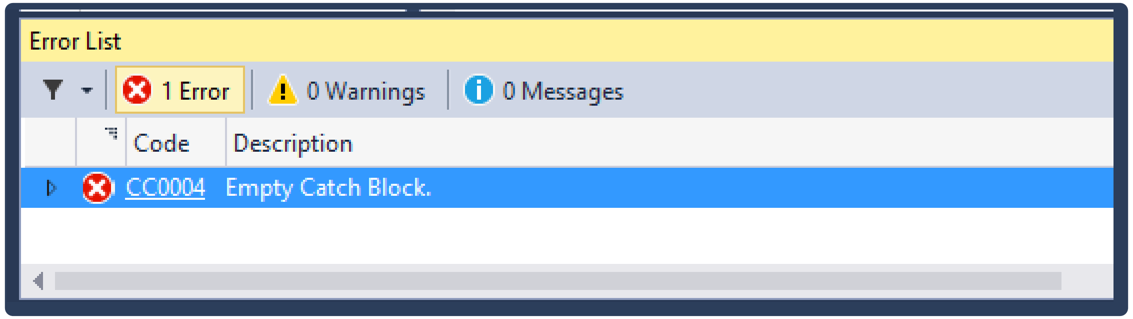
```
...
{
    public Foo() { }
    protected Foo() { }
}
...
```

Project



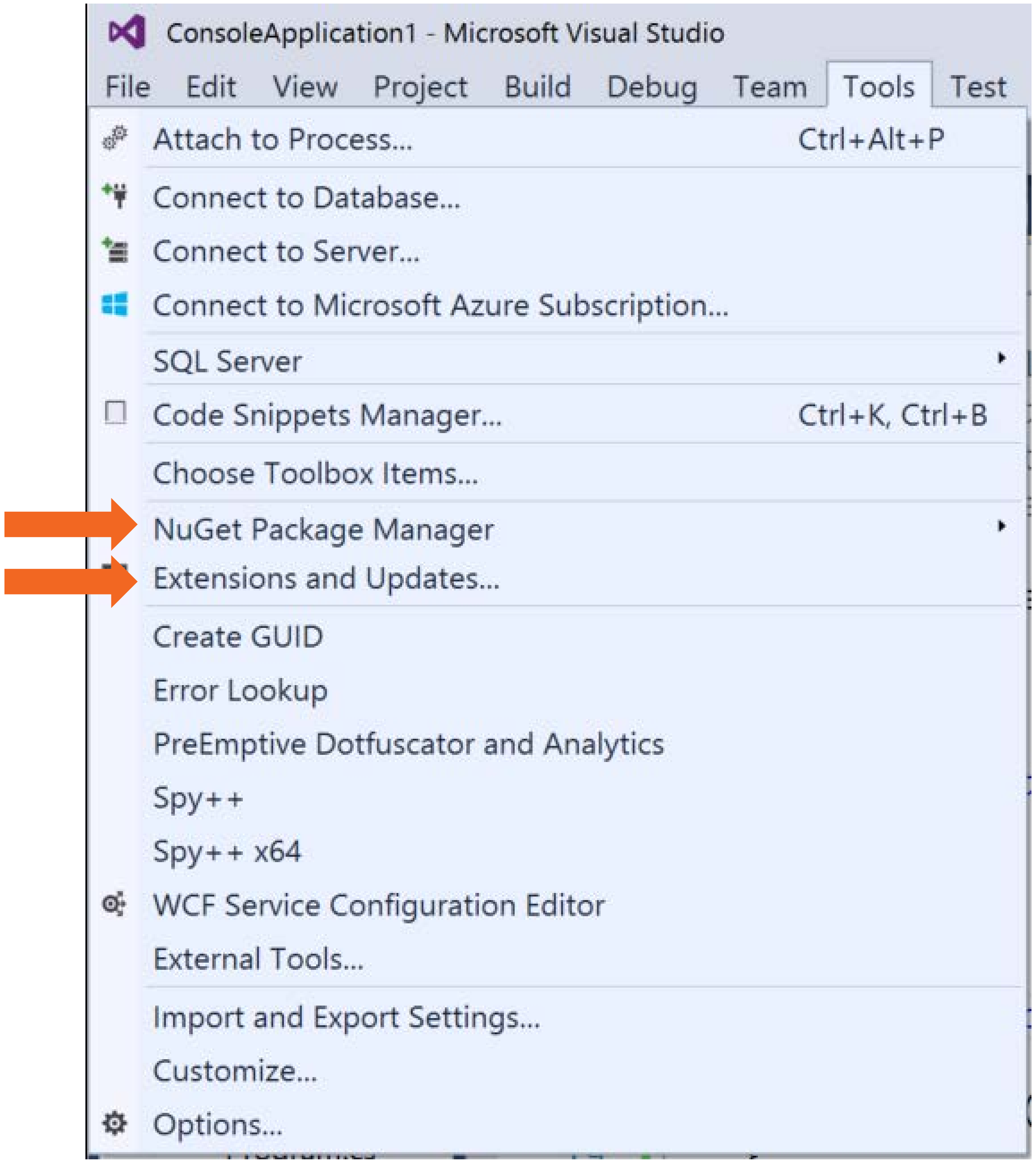
```
static void Main(string[] args)
{
    try { }
    catch
    { }
}
```

Empty Catch Block.



Error List

Code	Description
CC0004	Empty Catch Block.


ConsoleApplication1 - Microsoft Visual Studio
File
Edit
View
Project
Build
Debug
Team
Tools
Test
Attach to Process... Ctrl+Alt+P
Connect to Database...
Connect to Server...
Connect to Microsoft Azure Subscription...
SQL Server
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
NuGet Package Manager
Extensions and Updates...
Create GUID
Error Lookup
PreEmptive Dotfuscator and Analytics
Spy++
Spy++ x64
WCF Service Configuration Editor
External Tools...
Import and Export Settings...
Customize...
Options...

pluralsight

Desired Features for Analyzer Sets

- Cohesive
- Complete
- Thoughtful
- Tested

Current Sources of General Purpose Analyzers

https://github.com/DotNetAnalyzers

https://github.com/code-cracker/code-cracker

https://github.com/Wintellect/Wintellect.Analyzers

Microsoft.CodeAnalysis.CSharp.FxCopAnalyzers (NuGet)

Microsoft.CodeAnalysis.VisualBasic.FxCopAnalyzers

pluralsight

Per Machine

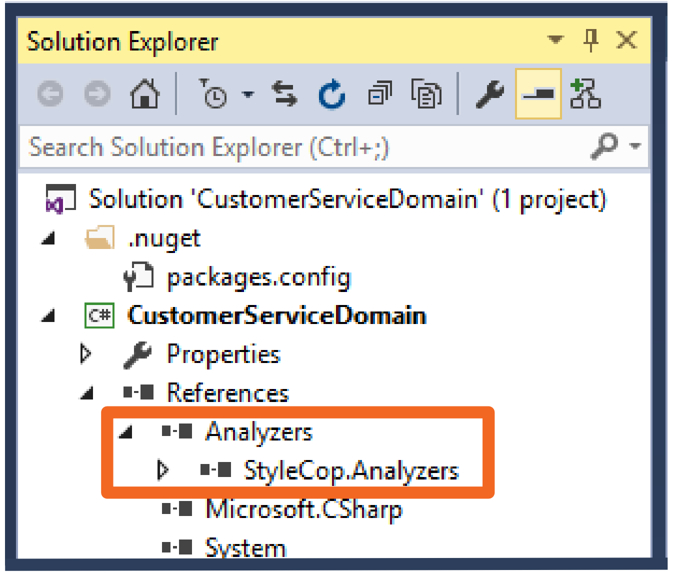
Single installation

Good for convenience analyzers

Installed and updated via extensions

?

Per Project



Solution Explorer

Search Solution Explorer (Ctrl+;)

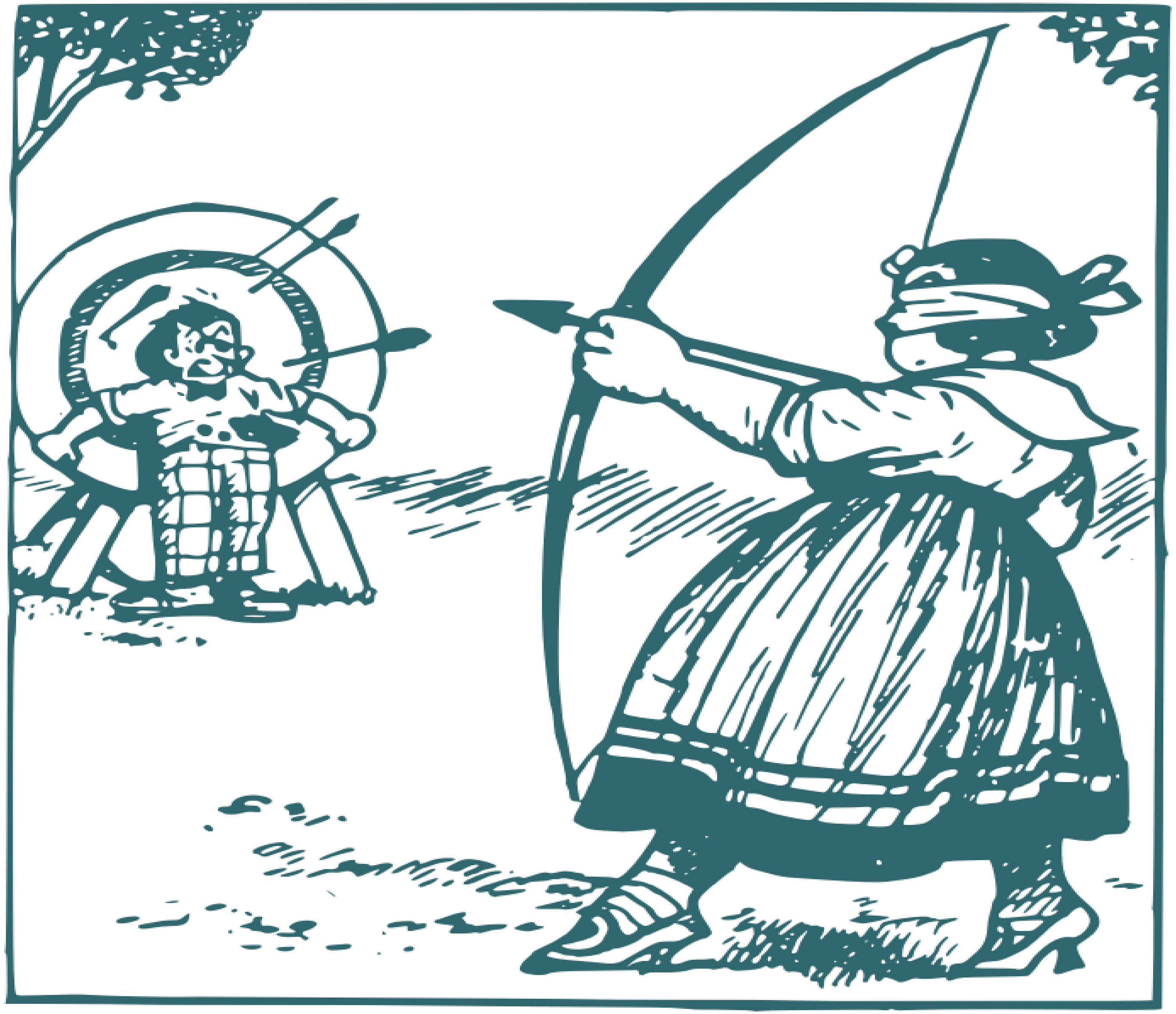
```
Solution 'CustomerServiceDomain' (1 project)
◢ .nuget
      packages.config
◢ CustomerServiceDomain
   ▷ Properties
   ◢ References
      ◢ Analyzers
         ▷ StyleCop.Analyzers
         Microsoft.CSharp
         System
```

Source control and build support

Good for project guidelines

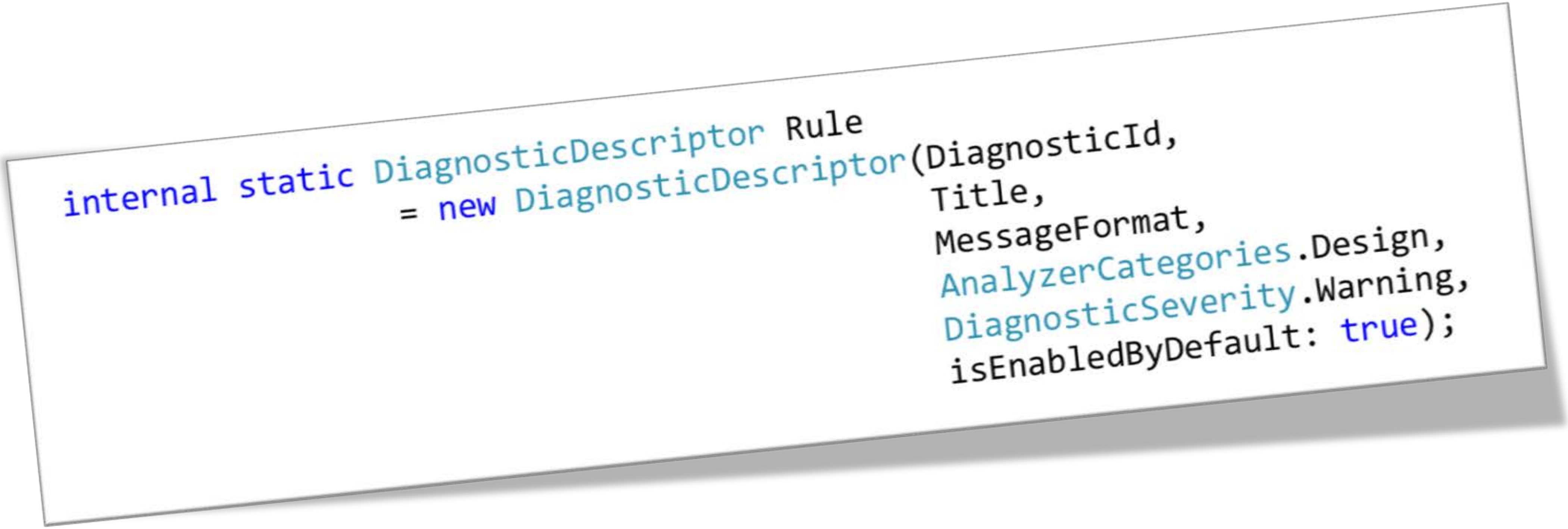
Installed and updated via NuGet

Diagnostic Analyzers and Refactorings Run in Full Trust



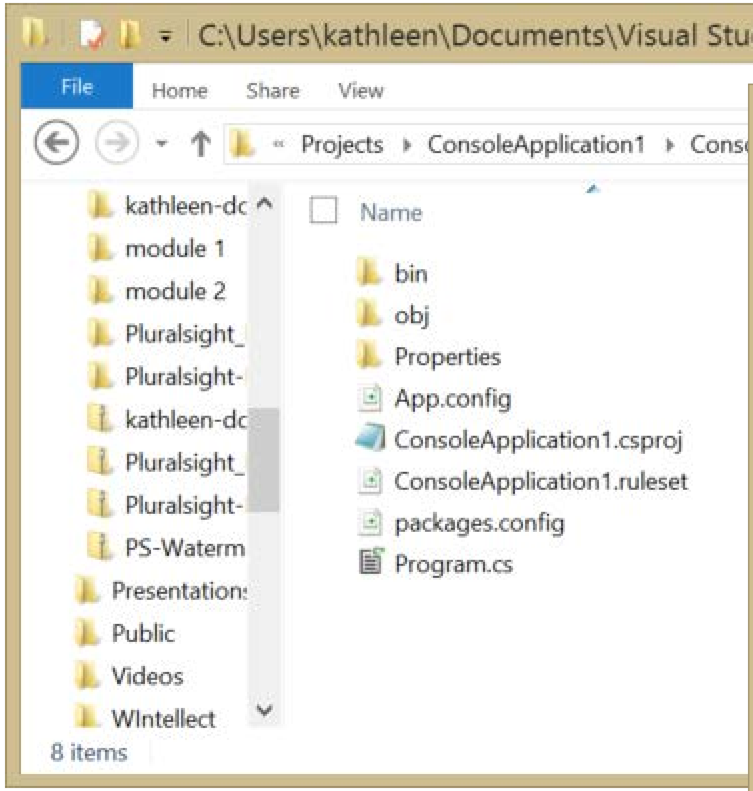
Library Specific Analyzers

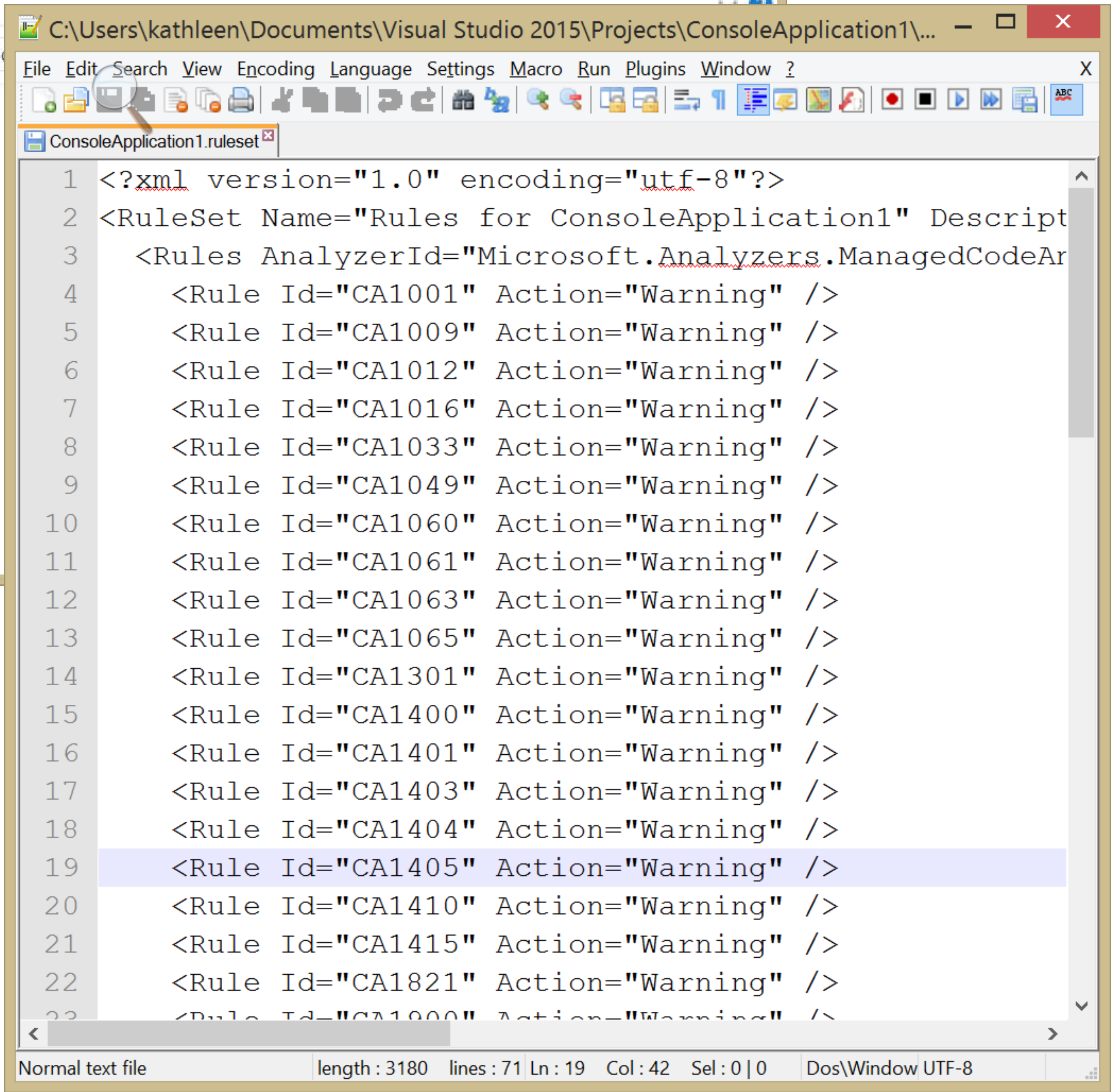
Each Analyzer Sets Defaults



```
internal static DiagnosticDescriptor Rule
        = new DiagnosticDescriptor(DiagnosticId,
                                   Title,
                                   MessageFormat,
                                   AnalyzerCategories.Design,
                                   DiagnosticSeverity.Warning,
                                   isEnabledByDefault: true);
```

Rulesets are XML Files




C:\Users\kathleen\Documents\Visual Studio 2015\Projects\ConsoleApplication1\...
File
Edit
Search
View
Encoding
Language
Settings
Macro
Run
Plugins
Window
?
ConsoleApplication1.ruleset

```xml
<?xml version="1.0" encoding="utf-8"?>
<RuleSet Name="Rules for ConsoleApplication1" Descript
  <Rules AnalyzerId="Microsoft.Analyzers.ManagedCodeAr
    <Rule Id="CA1001" Action="Warning" />
    <Rule Id="CA1009" Action="Warning" />
    <Rule Id="CA1012" Action="Warning" />
    <Rule Id="CA1016" Action="Warning" />
    <Rule Id="CA1033" Action="Warning" />
    <Rule Id="CA1049" Action="Warning" />
    <Rule Id="CA1060" Action="Warning" />
    <Rule Id="CA1061" Action="Warning" />
    <Rule Id="CA1063" Action="Warning" />
    <Rule Id="CA1065" Action="Warning" />
    <Rule Id="CA1301" Action="Warning" />
    <Rule Id="CA1400" Action="Warning" />
    <Rule Id="CA1401" Action="Warning" />
    <Rule Id="CA1403" Action="Warning" />
    <Rule Id="CA1404" Action="Warning" />
    <Rule Id="CA1405" Action="Warning" />
    <Rule Id="CA1410" Action="Warning" />
    <Rule Id="CA1415" Action="Warning" />
    <Rule Id="CA1821" Action="Warning" />
    <Rule Id="CA1900" Action="Warning" />
```

Normal text file length : 3180 lines : 71 Ln : 19 Col : 42 Sel : 0 | 0 Dos\Window UTF-8

pluralsight

Definitions for New and Subtle Phrases

My Own Definitions

Diagnostic analyzer

A rule for code quality supported in Visual Studio 2015 and builds based on the new compilers.

Depending on the severity, it may result in squiggles and compilation messages.

Squiggle

A wavy underline in response to an error (red by default) or warning (green by default) compiler message.

Code fix

An automatic fix linked to a specific diagnostic analyzer.

Code fixes appear in a popup while the user is positioned on the line.

Refactoring

A separate mechanism in Visual Studio 2015 to provide automatic code changes, even when a specific issue can't be identified and available for other scenarios, such as to switch between valid choices.

Ruleset

A mechanism for overriding the default enabling and severity of diagnostic analyzers.

Rulesets are accessed through the Code Analysis tab of Project Properties.

Roslyn

1) The new C# and Visual Basic compilers
2) The overall effort to create the new compilers and the associated changes to Visual Studio and the build system

Incremental compiler

A compiler that can limit compilation to altered code.

In Roslyn, this allows rapid recreation of valid structures, especially important for diagnostic analyzers.

Immutable

Data which cannot be altered.

Instead of altering immutable data, a new copy is returned with the changes

Tree

A representation of a hierarchical structure

Syntax Tree

In most compilers, the hierarchical data structure created from parsing code.

Nodes in this structure represent common programming elements like classes, methods, statements and expressions.

Semantic Model

In most compilers, a hierarchical data structure which contains resolved symbols and adequate information to emit executable code.

pluralsight

Full fidelity round-trip

A process can exactly recreate its input.

Syntax Visualizer

A Visual Studio 2015 add-in that displays the syntax tree and semantic model for code at the current cursor position.

Experimental hive (or experimental instance)

A second instance of Visual Studio that spins up when you debug a Visual Studio extension.

Called a hive because it contains an entirely independent copy of the registry settings, and thus maintains different settings.

Verbatim strings in C#

Strings prefixed with “@” appear exactly as stated, including new line characters.

Azure VM

A virtual machine in the Azure gallery that lets you play with Visual Studio 2015.

Summary

Understand

Find and Install

Use

Suppress

The new diagnostics let you create better code

- Suggesting and enforcing the guidelines of **your** team

Diagnostics can be installed per project via NuGet
or installed machine wide via Visual Studio Gallery

Diagnostics provide feedback where and when you expect

- Squiggles, lightbulbs, Ctl-period, and compiler output

You and your team retain complete control over analyzers

- Suppress or change severity as fits your guidelines

pluralsight