

Static Code Analysis with Visual Studio 2010

Allen Hurst
Ben Floyd

Allen Hurst

Ben Floyd

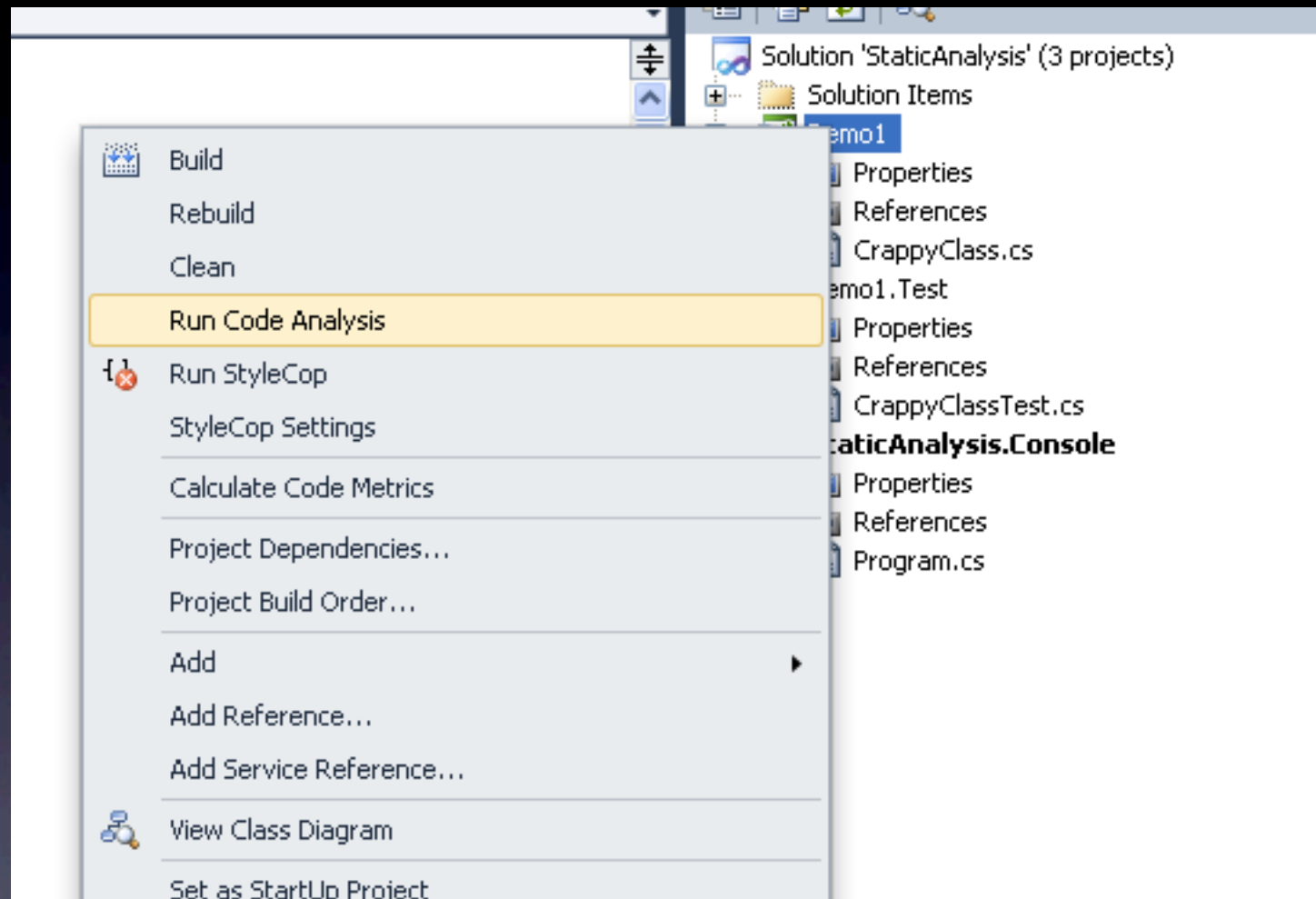
improving

Today

- Slides and Demos
- Agenda
 - FxCop Intro
 - StyleCop Intro
 - Going Further
 - Other Static Analysis Tools
- Ask Questions!

FxCop

- Analyzes managed assemblies
- Developed by Microsoft
- Free - As in Beer
- Included in VS 2010 Premium and Ultimate



Visual Studio Code Analysis Demo

Technical Debt



Technical Debt

Reckless	Prudent
<i>“We don’t have time for design”</i>	<i>“We must ship now and deal with consequences”</i>
Deliberate	
Inadvertent	
<i>“What’s Layering?”</i>	<i>“Now we know how we should have done it”</i>

FxCop Rules

- Design
- Globalization
- Interoperability
- Mobility
- Naming
- Performance
- Portability
- Security
- Usage
- Maintainability
- Reliability

FxCop Rule Sets Demo

FxCop vs. Code Review

Disposable Demo

Custom Dictionary Demo

Custom Rules

StyleCop





FASHION POLICE

Dat belt don't go
with dem shoes Fool!

StyleCop

- Static Code Analysis - C# only
- Written by Microsoft
- Different than FxCop
- Free (as in speech)
- Not built in to Visual Studio
- Enforces style guidelines

StyleCop Demo

StyleCop

- Documentation
- Layout
- Maintainability
- Naming
- Ordering
- Readability
- Spacing



StyleCop

I pity the fool that
don't document his methods



Why StyleCop?

More StyleCop Demo

Next Steps

StyleCop/FxCop Build Integration Demo

Check In Rules

FxCop/StyleCop

Other Static Analysis Tools



- Build
- Rebuild
- Clean
- Run Code Analysis
- Run StyleCop
- StyleCop Settings
- Calculate Code Metrics**
- Project Dependencies...
- Project Build Order...
- Add
- Add Reference...
- Add Service Reference...
- View Class Diagram
- Set as StartUp Project
- Debug
- Add Solution to Source Control...
- Cut (Ctrl+X)
- Paste (Ctrl+V)
- Remove (Del)

- Properties
- References
- CrappyClass.cs
- Demo1.Test
- Properties
- References
- CrappyClassTest.cs
- omeCompany.Utilities
- Properties
- References
- MyUtility.cs
- StaticAnalysis.Console**
- Properties
- References
- Program.cs

Code Metrics Demo

NDepend

A Little About Coupling

- Efferent Coupling - Outgoing Coupling
- Afferent Coupling - Incoming Coupling
- High Coupling requires stability
- Achieve stability through abstractions and good OO design

Coverity

CAT.NET

Discussion

Thanks!

- Allen Hurst
- <http://ahurst.com>
- Ben Floyd
- <http://dataplex.org>