Custom Roslyn Tool for Static Code Analysis

Zuzana Dankovčíková

Spring 2017

Goal

- Static code analysis tool for Kentico Software
- Check compliance to company’s internal guidelines
- Real-time static code analysis in Visual Studio
- Provide automated code fixes
- Performance optimization

Motivation

- Need for static code analysis tool
  - Many internal conventions and guidelines
  - Complicated code reviews
- Replace existing console application (“BugHunter”)
  - Dummy string matching
  - Many false negatives and positives
  - External application
  - Obscure and not granular configuration

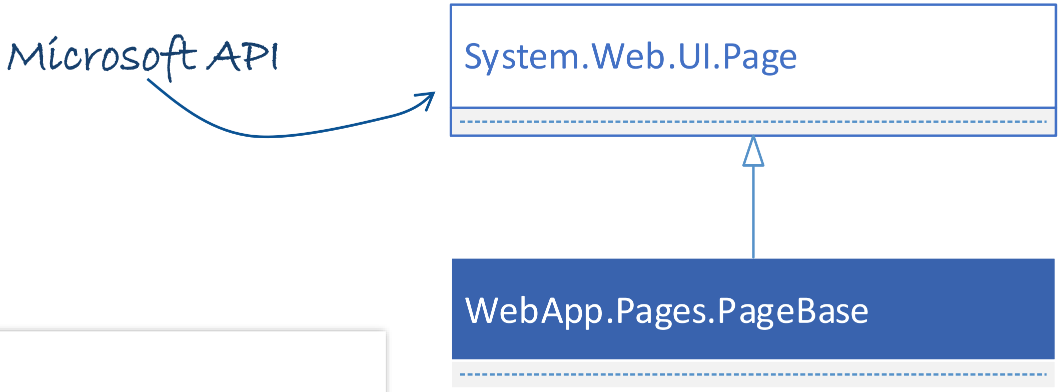
.NET Compiler Platform (“Roslyn”)

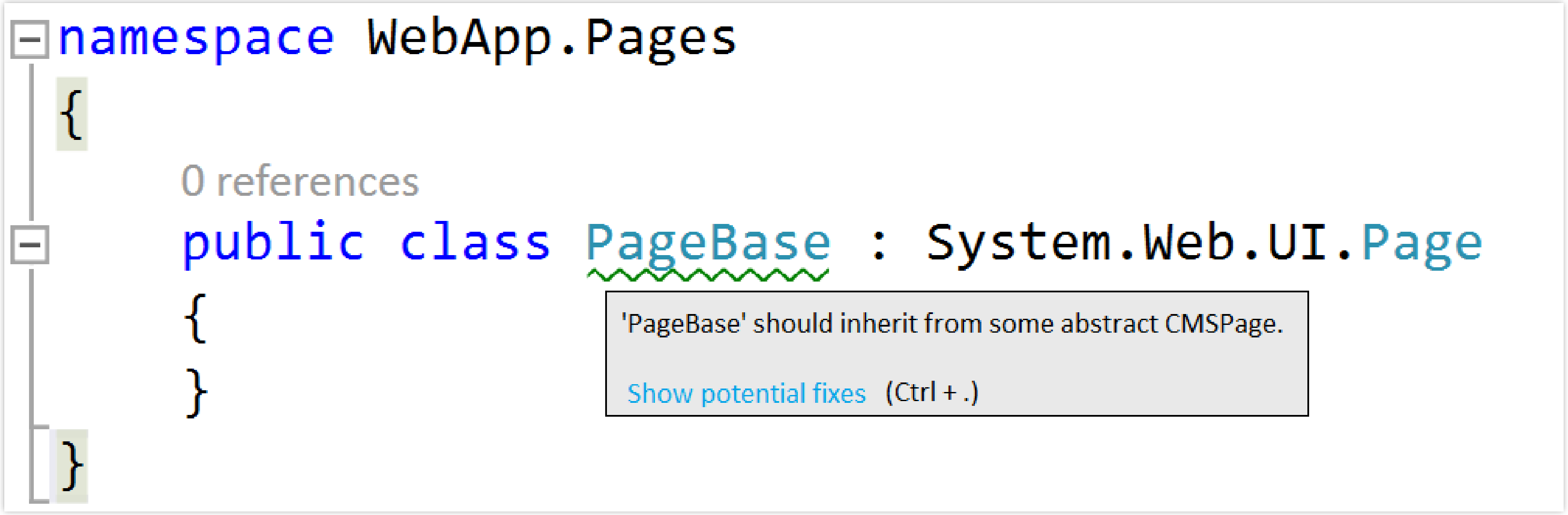
- Compiler-as-a-service
  - Internals of the compiler pipeline as API
- Analysis APIs
- Refactoring APIs

Thesis development

- Analyzing the old tool
- Implementation
    - Analyzers
    - Code fixes
- Unit tests
- Documentation
- Performance optimization

Analyzer



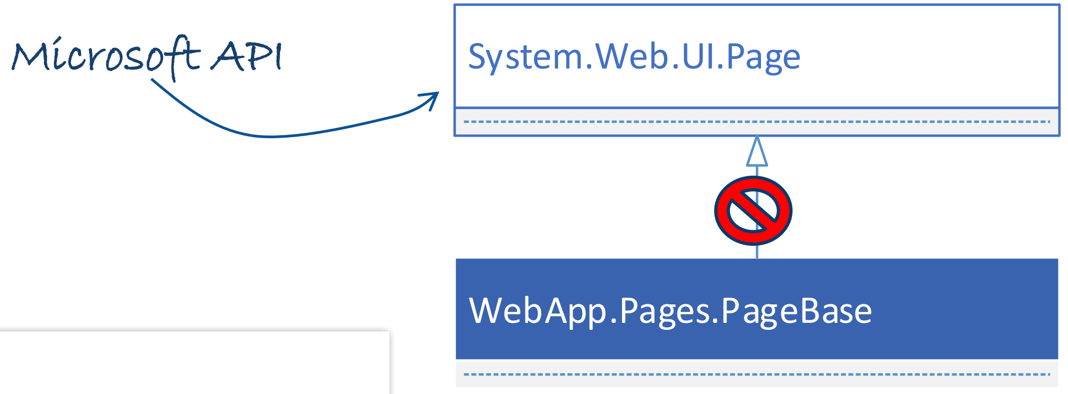


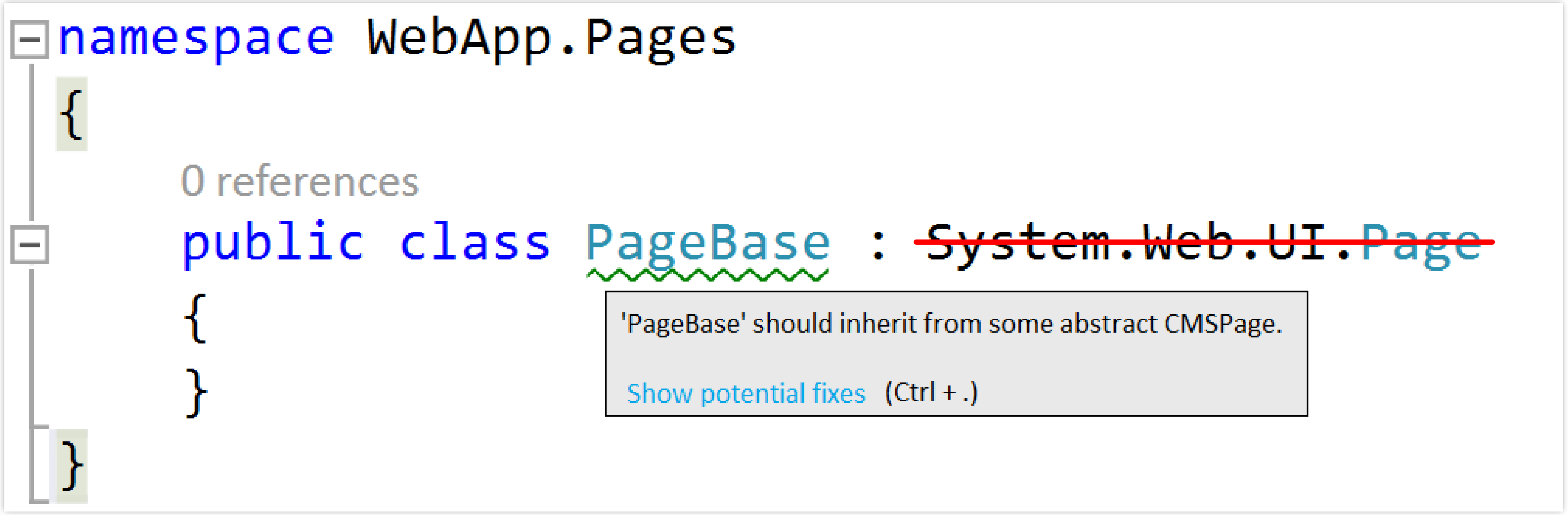
```
namespace WebApp.Pages
{
    0 references
    public class PageBase : System.Web.UI.Page
    {
    }
}
```

'PageBase' should inherit from some abstract CMSPage.

Show potential fixes (Ctrl + .)

Analyzer



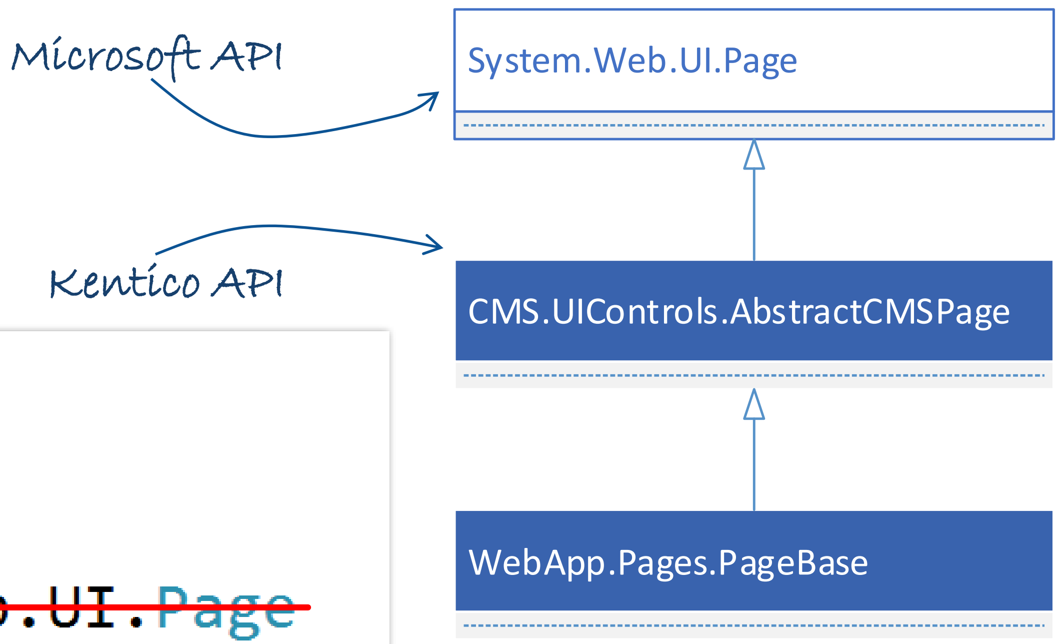


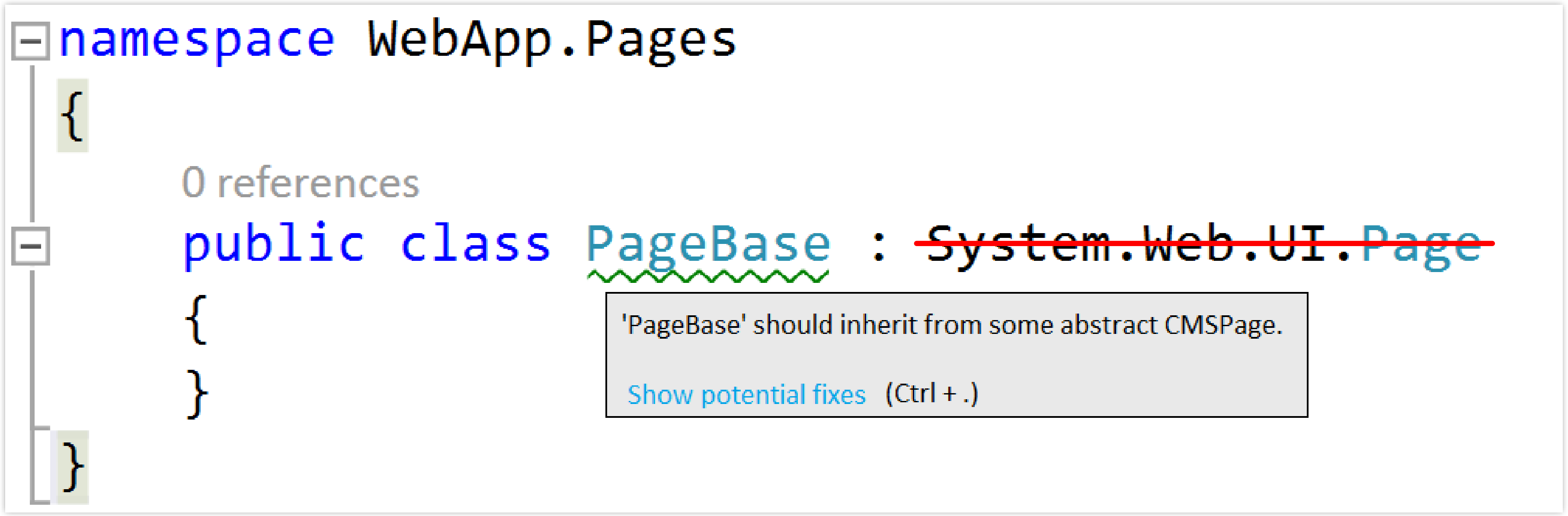
```
namespace WebApp.Pages
{
        0 references
        public class PageBase : ~~System.Web.UI.Page~~
        {
        }
}
```

'PageBase' should inherit from some abstract CMSPage.

Show potential fixes (Ctrl + .)

Analyzer



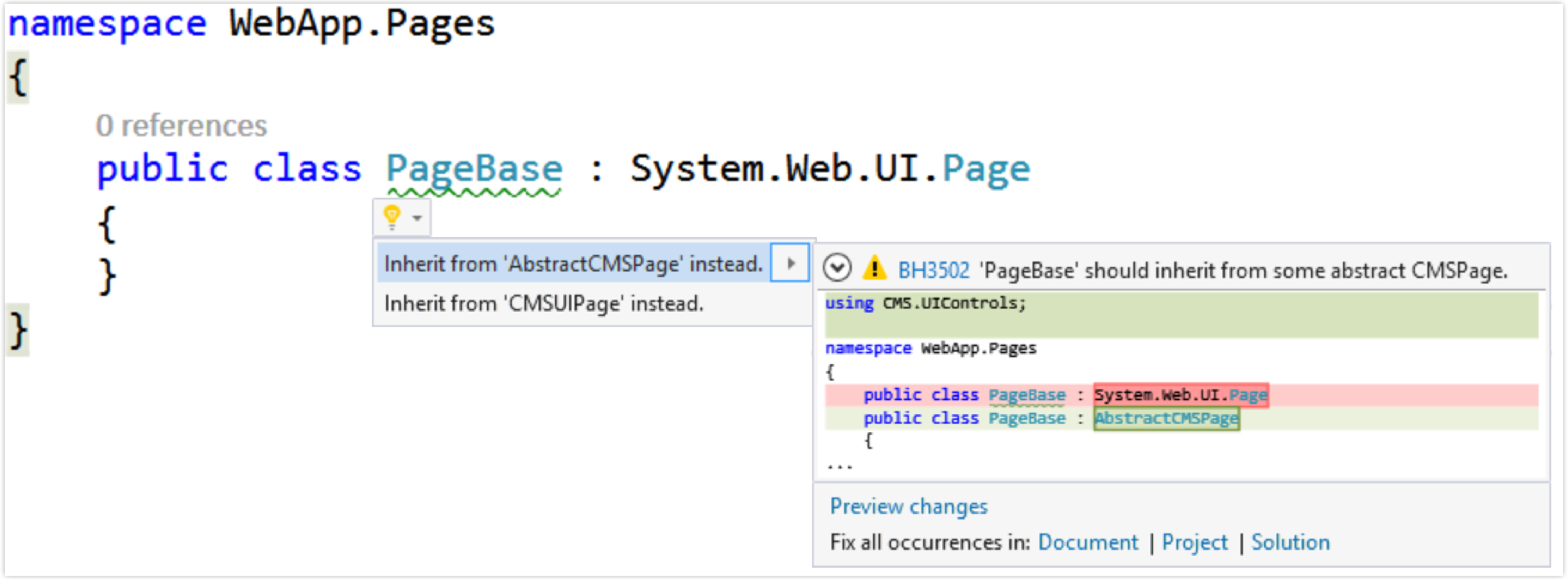


```
namespace WebApp.Pages
{
    0 references
    public class PageBase : ~~System.Web.UI.Page~~
    {
    }
}
```

'PageBase' should inherit from some abstract CMSPage.

Show potential fixes (Ctrl + .)

Code fix



```
namespace WebApp.Pages
{
    0 references
    public class PageBase : System.Web.UI.Page
    {
    }
}
```

Inherit from 'AbstractCMSPage' instead.
Inherit from 'CMSUIPage' instead.

BH3502 'PageBase' should inherit from some abstract CMSPage.

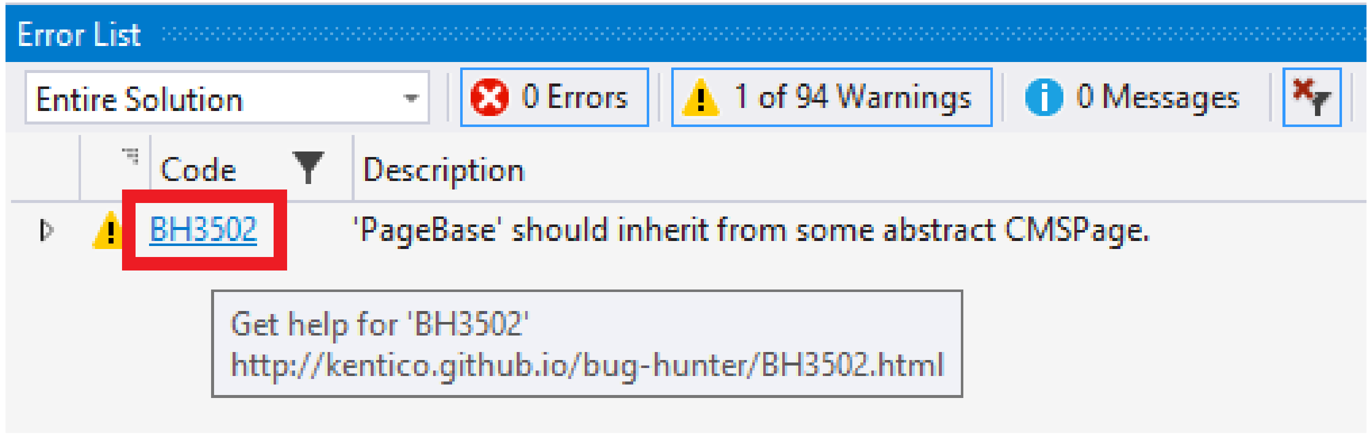
```
using CMS.UIControls;

namespace WebApp.Pages
{
    public class PageBase : System.Web.UI.Page
    public class PageBase : AbstractCMSPage
    {
...
```

Preview changes

Fix all occurrences in: Document | Project | Solution

Online documentation

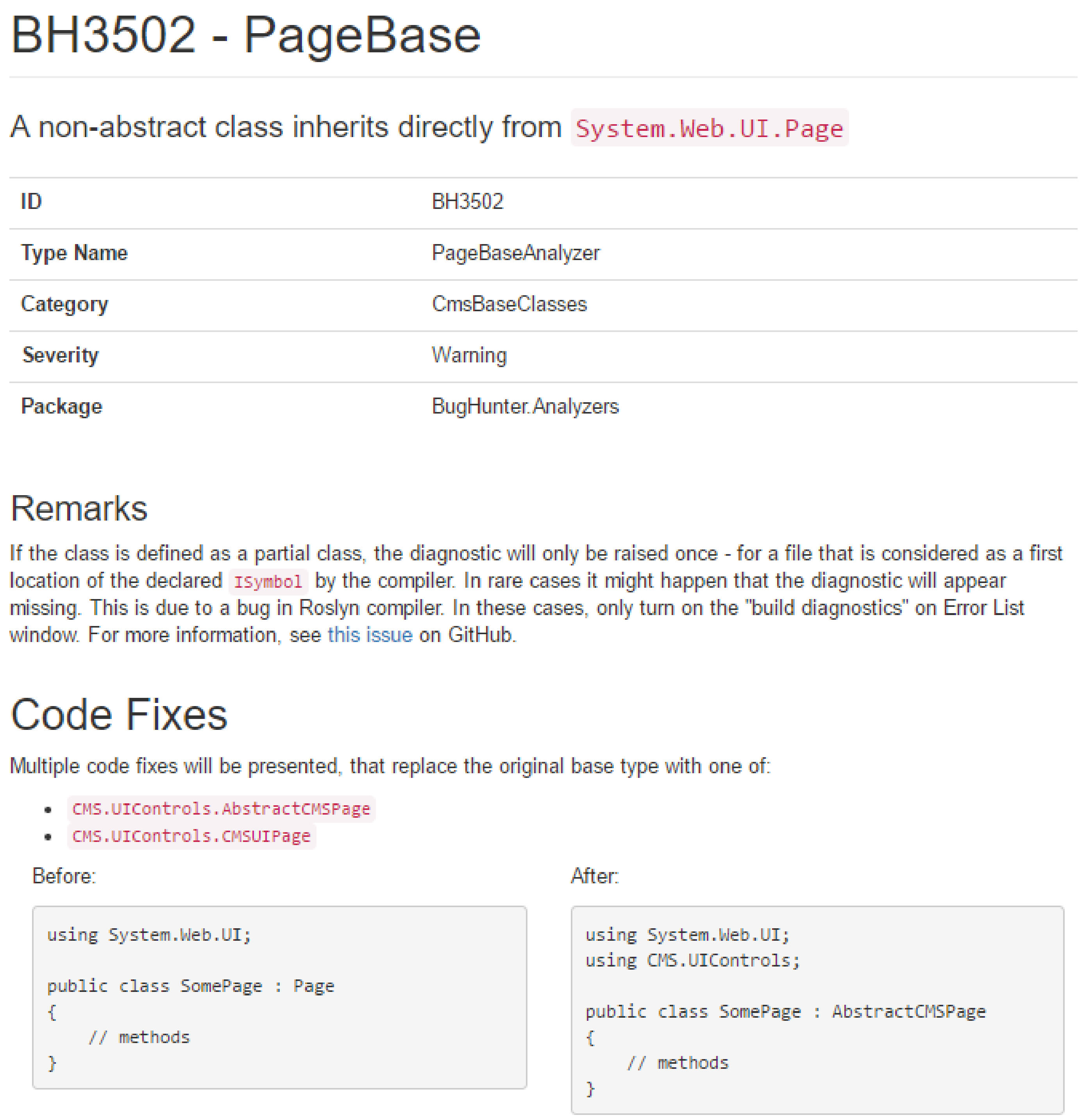
Clicking on the analyzer ID in Error list window



BH3502 'PageBase' should inherit from some abstract CMSPage.

Get help for 'BH3502'
http://kentico.github.io/bug-hunter/BH3502.html

Online documentation



BH3502 - PageBase

A non-abstract class inherits directly from `System.Web.UI.Page`

ID	BH3502
Type Name	PageBaseAnalyzer
Category	CmsBaseClasses
Severity	Warning
Package	BugHunter.Analyzers

Remarks

If the class is defined as a partial class, the diagnostic will only be raised once - for a file that is considered as a first location of the declared `ISymbol` by the compiler. In rare cases it might happen that the diagnostic will appear missing. This is due to a bug in Roslyn compiler. In these cases, only turn on the "build diagnostics" on Error List window. For more information, see this issue on GitHub.

Code Fixes

Multiple code fixes will be presented, that replace the original base type with one of:

- `CMS.UIControls.AbstractCMSPage`
- `CMS.UIControls.CMSUIPage`

Before:

```
using System.Web.UI;

public class SomePage : Page
{
    // methods
}
```

After:

```
using System.Web.UI;
using CMS.UIControls;

public class SomePage : AbstractCMSPage
{
    // methods
}
```

Unit tests

- Rewriting Microsoft utilities for testing analyzers and code fixes
- Discovered hidden bugs in existing BugHunter
- Tracking Kentico API changes

Performance of the analyzers

- Analyzed solution of Kentico's CMS contains over 240 projects
- Performance of the analyzers is crucial
- Tune performance of the slowest analyzers
- Analyze total impact on build times

Analyzers’ execution times

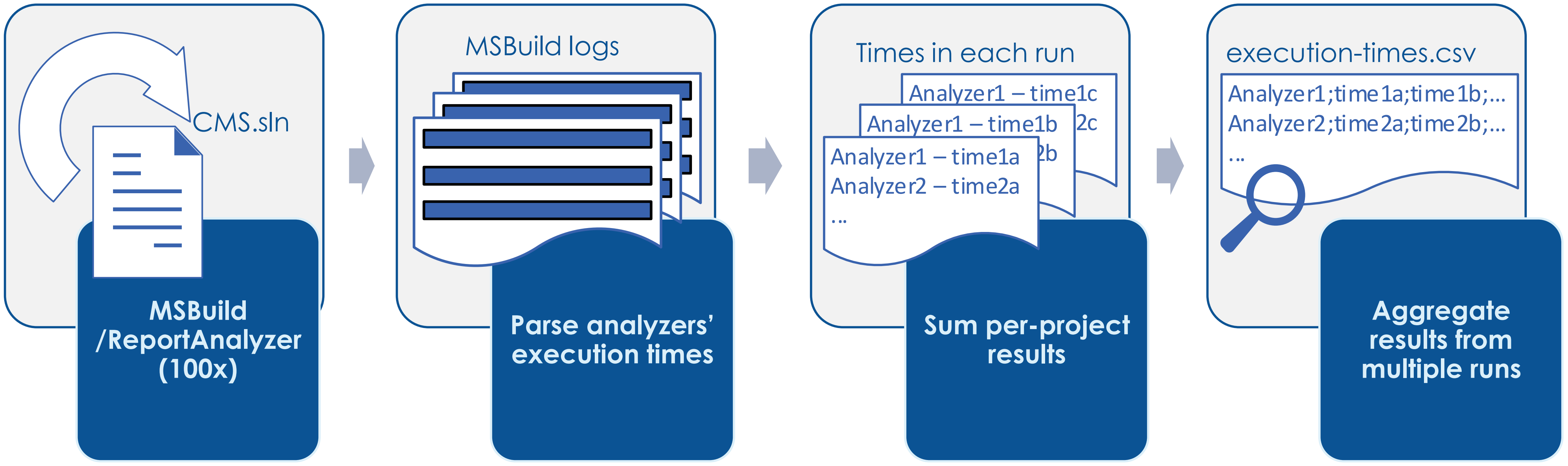
- NuGet package with analyzers + MSBuild with /ReportAnalyzer switch

Per-project execution time

Analyzer name

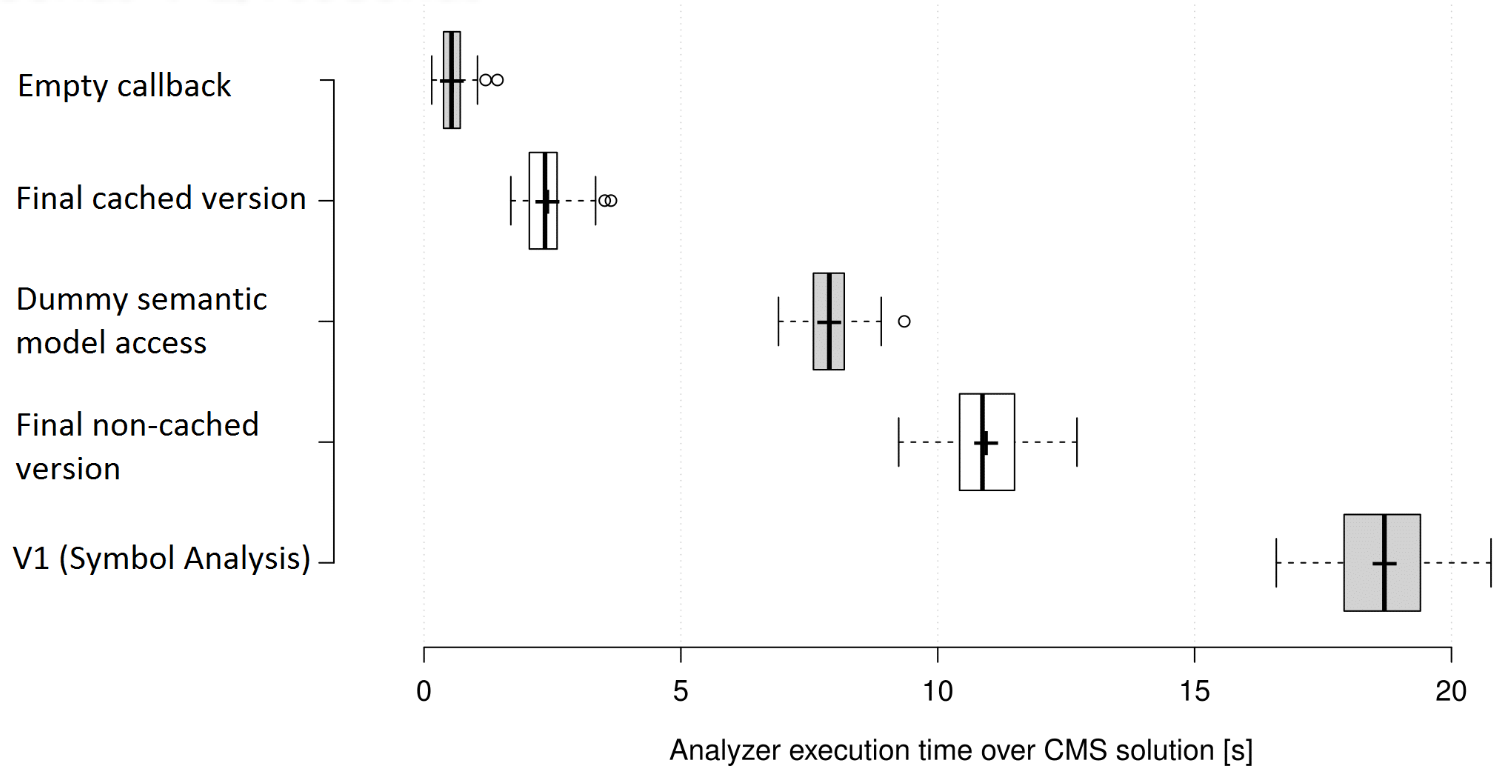
```
Total analyzer execution time: 0.017 seconds. (TaskId:3718)
  NOTE: Elapsed time may be less than analyzer execution time because analyzers can run concurrently. (TaskId:3718)
  Time (s)    %   Analyzer (TaskId:3718)
  0.017   100   BugHunter.Analyzers, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null (TaskId:3718)
  0.009    51     BugHunter.Analyzers.CmsApiReplacementRules.Analyzers.SystemIOAnalyzer (TaskId:3718)
  0.002    10     BugHunter.Analyzers.CmsBaseClassesRules.Analyzers.ModuleRegistrationAnalyzer (TaskId:3718)
  0.001     7     BugHunter.Analyzers.CmsApiReplacementRules.Analyzers.HttpSessionElementAccessAnalyzer (TaskId:3718)
  <0.001     3     BugHunter.Analyzers.StringAndCultureRules.Analyzers.StringCompareToMethodAnalyzer (TaskId:3718)
  <0.001     3     BugHunter.Analyzers.AbstractionOverImplementation.Analyzers.LuceneSearchDocumentAnalyzer (TaskId:3718)
  <0.001     2     BugHunter.Analyzers.CmsApiReplacementRules.Analyzers.HttpRequestQueryStringAnalyzer (TaskId:3718)
  <0.001     1     BugHunter.Analyzers.CmsApiReplacementRules.Analyzers.ClientScriptMethodsAnalyzer (TaskId:3718)
  <0.001     1     BugHunter.Analyzers.CmsApiGuidelinesRules.Analyzers.EventLogArgumentsAnalyzer (TaskId:3718)
  <0.001     1     BugHunter.Analyzers.CmsApiGuidelinesRules.Analyzers.WhereLikeMethodAnalyzer (TaskId:3718)
  <0.001     1     BugHunter.Analyzers.CmsApiReplacementRules.Analyzers.FormsAuthenticationSignOutAnalyzer (TaskId:3718)
  <0.001     1     BugHunter.Analyzers.StringAndCultureRules.Analyzers.StringEqualsMethodAnalyzer (TaskId:3718)
```

Analyzers' execution times



Tuning the performance

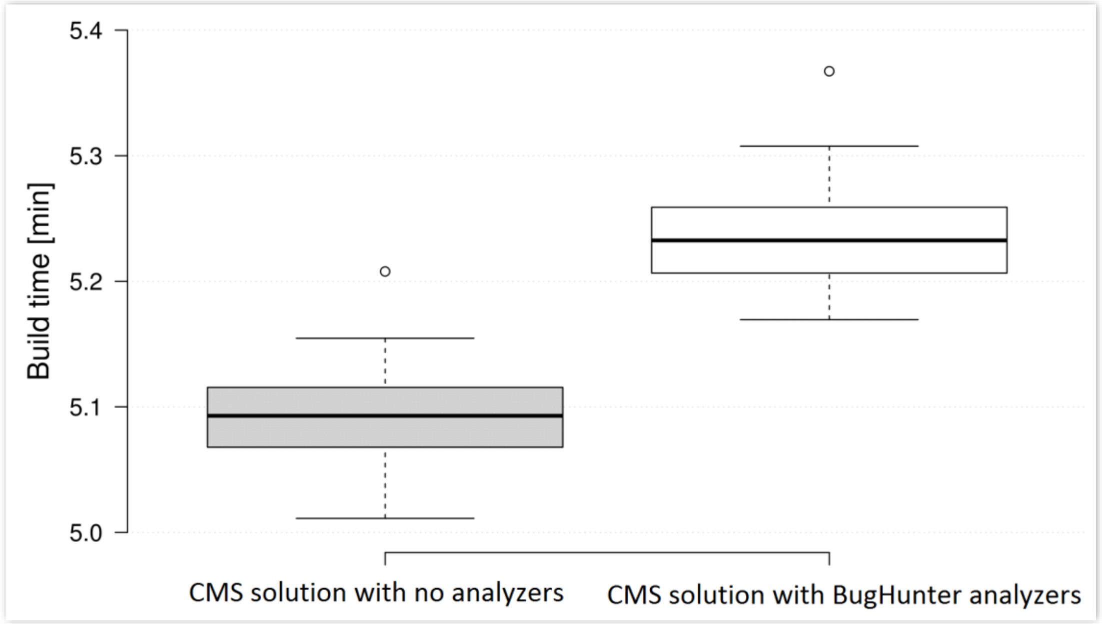
- Different versions of SystemIOAnalyzer
- 18 seconds → 2,4 seconds



Total impact on performance

- 100 build runs
- 5:05 minutes without analyzers
- 5:14 minutes with analyzers

- +2,9% (9 seconds) slowdown



Benefits

- Checks for types, not strings
- Runs inside of Visual Studio
  - Immediate feedback
  - Provides code fixes
  - Teaching new developers the internal rules
- Online documentation
- Transparent configuration
- Extensive unit testing
- Tracks Kentico API changes

Current state

- The new BugHunter has been deployed in April
- Available via company's internal NuGet feed
- The questionnaire did not show any performance nor correctness issues

Future work

- More rules that were not possible in the old version of the tool
- Upgrade to .NET Compiler Platform v2.0
- Making BugHunter available to Kentico’s customers via Microsoft NuGet Gallery

Thank you for your attention

Question

“As stated in the thesis, at the time of writing there are no tools for performance analysis of the analyzers.

- *Would a more generic solution than the one provided for the validation of the thesis make sense?*
- *What would be the challenges of such development?“*

“*More generic*” in terms of **running the analysis** itself?

- Problems obtaining the “real” execution time
  - Running analyzers manually does not give same results
  - Many internal optimizations performed by Roslyn
  - Experiments with measuring using Benchmark .NET were not successful
- Only viable option is /ReportAnalyzer switch in csc.exe
  - The proposed process of running the analysis multiple times, parsing and aggregating the results can be run on any solution with any set of analyzers

Would a more generic solution than the one provided for the validation of the thesis make sense?

What would be the challenges of such development?

“More generic” in terms of **running on different source codes** than Kentico’s CMS solution?

- Cannot not run on any solution
  - References to Kentico.Libraries are necessary
- Generating lot’s of code is possible but does not make sense
- Running on artificial generated source codes poses many questions
  - Test the analyzers separately?
  - What about invocations on unintended nodes?
  - Provide only code with diagnostics or test also on compliant code?

Would a more generic solution than the one provided for the validation of the thesis make sense?

What would be the challenges of such development?