

Slovenská Technická Univerzita v Bratislave  
Fakulta Informatiky a Informačných Technológií

Jakub Šimko

[jsimko@fiit.stuba.sk](mailto:jsimko@fiit.stuba.sk)

# MERANIE SOFTVÉRU

9.10.2012

MSI

# Meranie a metriky

---

- Kto by mal dávať pozor?
- Predsa všetci 😊 najmä však
  - ▣ Manažéri kvality
  - ▣ Manažéri monitorovania

# Prečo meriame?



- Chceme vedieť či
  - Niečo nekazíme
    - Hacky a badsmelly všetkých druhov
  - Či stíhame
  - A kto za to môže
  - Či si nevytvárame budúci problém
  
- Chceme vedieť reagovať

# Čo sa dá meraním zistiť?

- Kvalita softvéru (či skôr jej odhad)
  - Výpočtová efektívnosť
  - Rozšíriteľnosť
  - Čitateľnosť
  - Použiteľnosť a používanosť
- Informácie o procese tvorby softvéru
  - Stav plnenia plánu
  - Produktivita členov tímu
- To všetko meniace sa v čase
  - „Dôležité sú trendy, nie čísla“

# Čo merať?



- Zdrojové kódy
- Binárky
- Používanie softvéru
- Dokumentáciu
- Doručenú funkcionálnosť
- Čas
- Minuté peniaze

# Jednoduché metriky



- CLOC – comment lines of code
- ELOC – executable lines of code
- LOC – lines of code
- NCLOC – non comment lines of code
- NOP – number of packages
- NOC – number of classes
- NOM – number of methods
  
- Hlavný nedostatok: chýba interpretácia

# Cyklomatická zložitost' (CYCLO)



- Väčšinou vypočítaná pre metódu
- Počet rozhodovacích miest
  - ▣ Podmienky
  - ▣ Cykly
  
- Už lepšie, vypovedá o zložitosti kódu
- Stále však chýba interpretácia

# Je to vel'a, či málo?

## Java

Metric	Low	Avg	High	V.High
CYCLO/Line of code	0.16	0.20	0.24	0.36
LOC/Method	7	10	13	19.5
NOM/Class	4	7	10	15

## C++

Metric	Low	Avg	High	V.High
CYCLO/Line of code	0.20	0.25	0.30	0.45
LOC/Method	5	10	16	24
NOM/Class	4	9	15	22.5

Source: *Object-Oriented Metrics in Practice* (based on 45 Java projects)



# WMC and AMW

- Weighted Method Count – total complexity of a class

$$WMC = \frac{CYCLO}{LOC} \times \frac{LOC}{METHOD} \times \frac{NOM}{CLASS}$$

- Average Method Weight – average complexity of a method

$$AMW = \frac{CYCLO}{LOC} \times \frac{LOC}{METHOD}$$

# JAVA

## Java

Metric	Low	Avg	High	V.High
WMC	5	14	31	47
AMW	1.1	2.0	3.1	4.7
LOC/Class	28	70	130	195

## C++

Metric	Low	Avg	High	V.High
WMC	4	23	72	108
AMW	1.0	2.5	4.8	7.0
LOC/Class	20	90	240	360



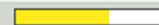





















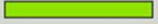
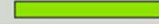







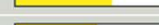

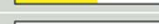


Source: Object-Oriented Metrics in Practice (based on 45 Java projects)

# Coverage report






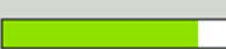


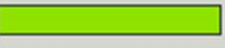
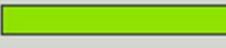







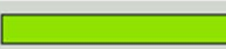

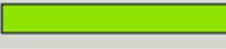


## PHP\_CodeCoverage

Current file: [/Volumes/git/private/php-code-coverage/PHP/CodeCoverage.php](#)

Legend:   executed   not executed   dead code

	Coverage									
	Classes			Functions / Methods			Lines			
Total		0.00%	0 / 1		50.00%	8 / 16	CRAP		64.66%	161 / 249
<b>PHP_CodeCoverage</b>		0.00%	0 / 1		50.00%	8 / 16			64.66%	161 / 249
<code>__construct(PHP_CodeCoverage_Driver \$driver = NULL, PHP_CodeCoverage_Filter \$filter = NULL)</code>		0.00%	0 / 1	8.14		71.43%	15 / 21			
<code>getInstance()</code>		100.00%	1 / 1	2		100.00%	4 / 4			
<code>start(\$id, \$clear = FALSE)</code>		100.00%	1 / 1	3		100.00%	8 / 8			
<code>stop(\$append = TRUE)</code>		100.00%	1 / 1	3		100.00%	8 / 8			
<code>append(array \$data, \$id = NULL, array \$filterGroups = array('DEFAULT'))</code>		0.00%	0 / 1	6.06		88.00%	22 / 25			
<code>merge(PHP_CodeCoverage \$that)</code>		0.00%	0 / 1	26.41		53.57%	15 / 28			
<code>getSummary()</code>		0.00%	0 / 1	14.03		94.44%	34 / 36			
<code>clear()</code>		100.00%	1 / 1	1		100.00%	5 / 5			
<code>filter()</code>		100.00%	1 / 1	1		100.00%	1 / 1			
<code>setForceCoversAnnotation(\$flag)</code>		100.00%	1 / 1	2		100.00%	4 / 4			
<code>setMapTestClassnameToCoveredClassname(\$flag)</code>		100.00%	1 / 1	2		100.00%	4 / 4			
<code>setProcessUncoveredFilesFromWhitelist(\$flag)</code>		100.00%	1 / 1	2		100.00%	4 / 4			
<code>applySelfFilter(&amp;\$data)</code>		0.00%	0 / 1	32.71		57.14%	16 / 28			
<code>applyListsFilter(&amp;\$data, \$filterGroups)</code>		0.00%	0 / 1	3.33		66.67%	4 / 6			
<code>applyCoversAnnotationFilter(&amp;\$data, \$id)</code>		0.00%	0 / 1	13.21		56.67%	17 / 30			
<code>processUncoveredFilesFromWhitelist()</code>		0.00%	0 / 1	156		0.00%	0 / 37			

# Coverage report

Coverage						
Functions / Methods				Lines		
	<b>50.00%</b>	8 / 16	CRAP		<b>64.66%</b>	161 / 2
	<b>50.00%</b>	8 / 16			<b>64.66%</b>	161 / 2
	<b>0.00%</b>	0 / 1	8.14		<b>71.43%</b>	15 /
	<b>100.00%</b>	1 / 1	2		<b>100.00%</b>	4
	<b>100.00%</b>	1 / 1	3		<b>100.00%</b>	8
	<b>100.00%</b>	1 / 1	3		<b>100.00%</b>	8
	<b>0.00%</b>	0 / 1	6.06		<b>88.00%</b>	22 /
	<b>0.00%</b>	0 / 1	26.41		<b>53.57%</b>	15 /
	<b>0.00%</b>	0 / 1	14.03		<b>94.44%</b>	34 /
	<b>100.00%</b>	1 / 1	1		<b>100.00%</b>	5
	<b>100.00%</b>	1 / 1	1		<b>100.00%</b>	1

# Change Risk Analysis and Predictions

$$C.R.A.P_m = CYCLO^2 \times \left(1 - \frac{COV_m}{100}\right)^3 + CYCLO$$

**Code coverage = 100%**

$$C.R.A.P_m = CYCLO_m$$

**Code coverage = 0%**

$$C.R.A.P_m = CYCLO_m^2 + CYCLO_m$$

# NPATH

- Počet necyklických prechodov metódou
- Biela skrinka

# OO – Ďalšie charakteristiky kódu



- Dedenie
  - ▣ Nie je dobré ak je príliš hlboké
  - ▣ Niektorí ho zatračujú úplne
- Súdržnosť komponentov
  - ▣ Čo najvyššia
- Previazanosť komponentov
  - ▣ Čo najnižšia

# Nástroje

- Sonar <http://www.sonarsource.org/>
- PHP Mess detector <http://phpmd.org/>
- PMD (Java) <http://pmd.sourceforge.net/>
  
- Štýl, dobré vzory písania, chyby, refaktoring
  - StyleCop, FindBugs, FxCop, CheckStyle (CodeRush), PHP Mess Detector
  
- Zložitosť, súdržnosť, rozsah
  - SourceMonitor, Eclipse Metrics, LOC Metrics
  
- Pokrytie kódu testami
  - Testovacie nástroje (JUnit)



# Ukážka: SourceMonitor

SourceMonitor - [Files in C# Project 'Test-SourceMonitor', Checkpoint 'Checkpoint1' [Base Directory: 'F:\Programs\NET\NextGen Semantic Services\Meia\]]

File Edit View Window Help

File Name	Lines	Statements	% Comments	% Docs	Classes	Methods/Cl...	Calls/Method	Stmts/Method	Max Complexity	Max Depth	Avg Depth	Avg Complexity	
Steltek.PhotoGallery\Display\CommonPage\PhotoBrowser.xaml.cs	1210	391	2,6	26,8	1	72,00	1,01	3,88	11	4	2,06	1,76	
Meia.Repository\RepositoryFacade.cs	592	178	1,5	25,2	1	41,00	1,85	3,02	5	5	1,88	1,67	
GraphToolkit.Templates\PhotoBrowser\PhotoNodeData.cs	234	145	2,1	0,0	1	37,00	0,51	1,54	2	3	1,92	1,03	
SessionLogAnalysis\AOLAnalyzer.cs	978	525	9,0	8,6	1	34,00	6,94	13,62	20	6	2,76	4,56	
Steltek\Factic\Browser.cs	498	225	5,6	9,0	1	34,00	2,00	4,47	10	6	2,35	1,84	
Steltek\ResourceEditor\ResourceEditorWidget.xaml.cs	722	349	6,4	1,1	1	30,00	2,93	9,43	18	7	2,41	3,87	
SearchSupport.SearchHistoryTree\HistoryMap\HMMetadata.cs	686	387	7,0	10,5	1	26,00	5,31	12,31	40	7	2,91	5,23	
GraphToolkit.Core\...	...	...	...	...	...	25,00	0,36	2,04	3	4	1,75	1,40	
Steltek.GraphVizNo...	...	...	...	...	...	25,00	0,60	1,76	4	3	1,89	1,38	
Steltek\FacetedBro...	...	...	...	...	...	24,00	1,67	6,33	30	6	2,60	3,71	
Steltek.GraphVizBr...	...	...	...	...	...	24,00	1,25	5,17	7	4	1,95	1,96	
GraphToolkit.Core\...	...	...	...	...	...	24,00	0,63	1,75	4	3	1,50	1,14	
Steltek.PhotoGaller...	...	...	...	...	...	23,00	0,52	1,52	2	3	1,79	1,04	
Steltek.GraphVizUC...	...	...	...	...	...	23,00	3,13	9,78	60	9+	3,87	5,92	
GraphToolkit.Core\...	...	...	...	...	...	20,67	0,45	1,81	4	4	1,78	1,26	
Steltek\Factic\Face...	...	...	...	...	...	20,00	2,35	5,35	5	4	2,24	1,86	
Steltek\Factic\View...	...	...	...	...	...	20,00	0,50	1,75	1	3	1,85	1,00	
Steltek\Factic\View...	...	...	...	...	...	2,1	20,00	0,50	1,50	1	3	1,79	1,00
GraphToolkit.Core\...	...	...	...	...	...	0,0	19,00	1,32	3,30	13	5	2,21	1,86
GraphToolkit.Core\...	...	...	...	...	...	1	19,00	2,26	6,58	17	5	2,24	3,74
SearchSupport.HM\...	...	...	...	...	...	37,00	19,00	3,00	6,00	4	4	1,86	1,84
Steltek.GraphVizGr...	...	...	...	...	...	0,51	19,00	4,79	15,53	51	9+	4,11	9,21
GraphToolkit.Templ...	...	...	...	...	...	1,54	19,00	0,00	1,84	4	3	1,82	1,26
SearchSupport.Sea...	...	...	...	...	...	221	18,00	3,28	7,78	16	6	2,81	3,39
SearchSupport.Ser...	...	...	...	...	...	Name of Most Complex Method	18,00	3,28	7,78	16	6	2,81	3,39
SessionLogAnalysis	...	...	...	...	...	Maximum Complexity	18,00	3,28	7,78	16	6	2,81	3,39
Meia.Repository\Da...	...	...	...	...	...	Line Number of Deepest Block	26	17,67	2,68	12	6	2,56	2,75
SearchSupport.Sea...	...	...	...	...	...	Maximum Block Depth	3	17,00	1,82	13	6	2,33	3,18
Steltek.GraphVizZo...	...	...	...	...	...	Average Block Depth	1,92	17,00	1,24	16	5	2,14	2,22
Meia\Factic\Deprec...	...	...	...	...	...	Average Complexity	1,03	16,00	1,25	4	3	1,65	1,32
Meia.Test\Service R...	...	...	...	...	...	15,38	0,85	1,56	2	4	1,70	1,30	
GraphToolkit.TestA...	...	...	...	...	...	15,00	1,53	4,27	2	3	1,46	1,13	
SearchSupport.Ses...	...	...	...	...	...	Most Complex Methods in 1 Class(es):	Complexity, Statements, Max Depth, Calls	14,00	4,11	38	6	2,48	3,57
Meia\Factic\Core.cs	...	...	...	...	...	PhotoNodeData.Author.get()	1, 1, 3, 0	14,00	5,43	3	4	1,79	1,53
GraphToolkit.Core\...	...	...	...	...	...	PhotoNodeData.Author.set()	1, 2, 3, 1	14,00	1,79	11	3	1,67	1,67
SessionLogAnalysis	...	...	...	...	...	PhotoNodeData.Bmp.get()	1, 1, 3, 0	14,00	0,14	2	3	1,77	1,07
SearchSupport.Res...	...	...	...	...	...	13,00	0,92	4,31	6	5	2,00	1,57	
GraphToolkit.Core\...	...	...	...	...	...	13,00	1,08	6,38	14	6	2,79	3,46	
GraphToolkit.Core\...	...	...	...	...	...	13,00	0,54	2,46	4	3	1,50	1,62	
GraphToolkit.Templ...	...	...	...	...	...	13,00	0,54	1,62	2	3	1,65	1,08	
SearchSupport.Res...	...	...	...	...	...	12,50	0,84	2,28	4	4	1,76	1,48	
SearchSupport.Sea...	...	...	...	...	...	12,50	0,84	2,28	4	4	1,76	1,48	
SearchSupport.Sea...	...	...	...	...	...	12,20	0,85	2,34	4	4	1,77	1,46	
GraphToolkit.Core\...	...	...	...	...	...	12,00	1,42	4,58	8	4	1,84	2,25	
Steltek\Service Ref...	...	...	...	...	...	11,80	0,86	2,42	4	4	1,82	1,49	
SearchSupport.Sea...	...	...	...	...	...	11,50	2,22	6,04	13	6	2,38	2,70	
Meia\Steltecia src...	...	...	...	...	...	11,00	2,75	5,70	16	6	2,33	2,59	

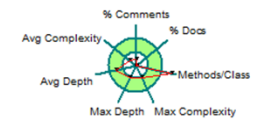
Metrics Details For File 'GraphToolkit.Templates\PhotoBrowser\PhotoNodeData.cs'

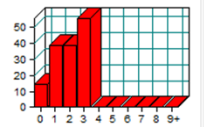
Parameter	Value
Project Directory	F:\Programs\NET\NextGen Semantic Services\Meia\
Project Name	Test-SourceMonitor
Checkpoint Name	Checkpoint1
File Name	GraphToolkit.Templates\PhotoBrowser\PhotoNodeData.cs
Lines	234
Statements	145
Percent Comment Lines	2,1
Percent Documentation Lines	0,0
Classes, Interfaces, Structs	1
Methods per Class	37,00
Calls per Method	0,51
Statements per Method	1,54
Line Number of Most Complex Method	221
Name of Most Complex Method	PhotoNodeData.OnPropertyChanged(string name)
Maximum Complexity	2
Line Number of Deepest Block	26
Maximum Block Depth	3
Average Block Depth	1,92
Average Complexity	1,03

-----

Most Complex Methods in 1 Class(es): Complexity, Statements, Max Depth, Calls

Method	Complexity	Statements	Max Depth	Calls
PhotoNodeData.Author.get()	1	1	3	0
PhotoNodeData.Author.set()	1	2	3	1
PhotoNodeData.Bmp.get()	1	1	3	0

Kviat Graph: 

Block Histogram (statements vs. depth): 

Buttons: Help, Copy, Print, Done

# Ukážka: FindBugs (Java)

The screenshot shows the FindBugs application window titled "FindBugs: Test". The interface includes a menu bar (File, Edit, View, Navigation, Designation, Help), a search field for class search strings, and a list of bugs categorized by type and rank. The "Bad practice (25)" category is expanded, showing sub-categories like "Confusing method name (6)". One bug is selected: "Class edu.umd.cs.findbugs.detect.DroppedException is not derived from an Exception, even though it is named as such".

The detailed view of this bug shows the following text:

```
Class edu.umd.cs.findbugs.detect.DroppedException is not derived from an Exception, even though it is named as such
In DroppedException.java
In class edu.umd.cs.findbugs.detect.DroppedException
```

Below the detailed view, the bug is highlighted in red with the following text:

**Class is not derived from an Exception, even though it is named as such**  
This class is not derived from another exception, but ends with 'Exception'. This will be confusing to users of this class.

# Ukážka: StyleCop (.NET)

The screenshot displays the Visual Studio IDE with a C# program and the StyleCop tool window. The program, `FindPrimes.Program`, defines a `Program` class with a `Main` method that finds prime numbers. The StyleCop tool window shows the rule `SA1400: AccessModifierMustBeDeclared` selected in the tree view. The rule details include its type name, check ID, category, cause, rule description, and how to fix violations. The error list at the bottom shows 58 warnings, with the first 12 related to StyleCop rules.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace FindPrimes
{
    class Program
    {
        const int Max = 20000000;
        static bool[] isPrime = new bool[Max];

        static void Main(string[] args)
        {
            // init
            isPrime[0] = false;
            isPrime[1] = false;
            for (int i = 2; i < Max; i++)
            {
                isPrime[i] = true;
            }

            // start with 2, continue
            for (int i = 0; i < Max; i++)
            {
                if (isPrime[i] == false)
                    continue;

                // found a prime, unprime its
                for (int j = i+1; j < Max; j++)
                {
                    isPrime[j] = false;
                }

                // find k-th prime
                int k = 4000, result = -1;

                for (int i = 0; i < Max; i++)
                {
                    if (isPrime[i])
                    {
                        if (--k == 0)
                        {
                            result = i;
                            break;
                        }
                    }
                }

                Console.Out.WriteLine(String.Format("The {0}th prime is {1}.", k, result));
            }
        }
    }
}
```

**StyleCop 4.3**  
**SA1400: AccessModifierMustBeDeclared**

TypeName	AccessModifierMustBeDeclared
CheckId	SA1400
Category	Maintainability Rules

**Cause**  
The access modifier for a C# element has not been explicitly defined.

**Rule Description**  
C# allows elements to be defined without an access modifier. Depending upon the type of element, C# will automatically assign an access level to the element in this case.

This rule requires an access modifier to be explicitly defined for every element. This removes the need for the reader to make assumptions about the code, improving the readability of the code.

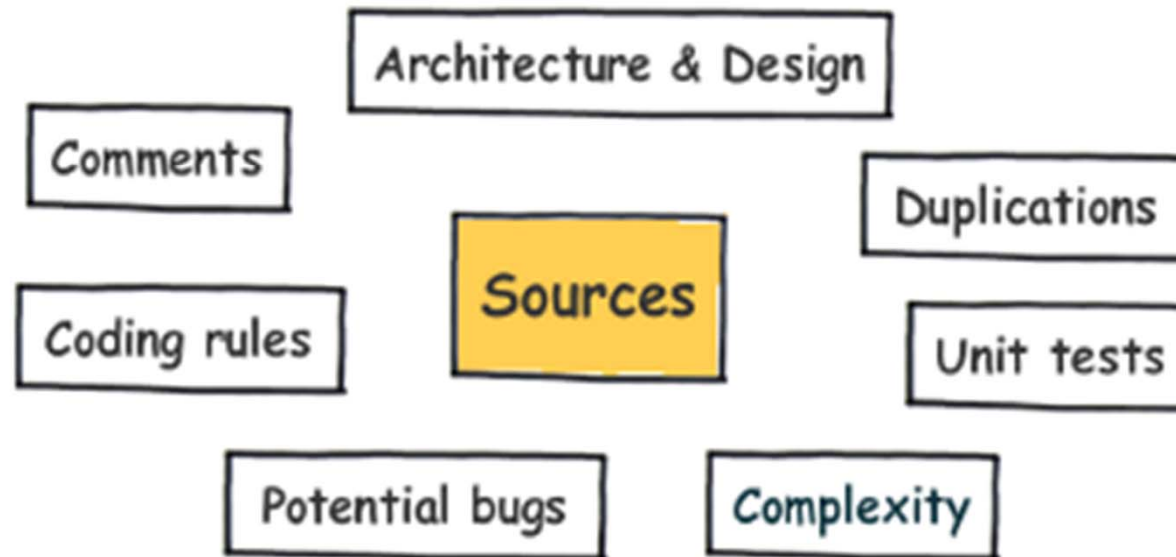
**How to Fix Violations**  
To fix a violation of this rule, add an access modifier to the declaration of the element.

© Microsoft Corporation. All Rights Reserved.

**Error List**  
0 Errors | 58 Warnings | 0 Messages

Description	File	Line	Column	Project
SA1600: The method must have a documentation header.	Program.cs	13	1	Program.cs
SA1633: The file has no header, the header Xml is invalid, or the header is not located at the top of the file.	Program.cs	1	1	Program.cs
SA1503: The body of the if statement must be wrapped in opening and closing curly brackets.	Program.cs	27	1	Program.cs
SA1515: A single-line comment must be preceded by a blank line or another single-line comment, or must be the first item in its scope. To ignore this error when commenting out a line of code, begin the comment with '////' rather than '//'. // init	Program.cs	35	1	Program.cs
SA1513: Statements or elements wrapped in curly brackets must be followed by a blank line.	Program.cs	48	1	Program.cs
SA1400: The class must have an access modifier.	Program.cs	8	1	Program.cs
SA1400: The field must have an access modifier.	Program.cs	10	1	Program.cs
SA1400: The field must have an access modifier.	Program.cs	11	1	Program.cs

# Sonar



# Ukážka: Sonar

## Lines of code

**1,910**

3,391 lines  
10 files

## Classes

**11**

98 methods

## Comments

**38.9%**

1,214 lines  
0 commented LOCs

## Duplications

**7.7%**

262 lines  
54 blocks  
7 files

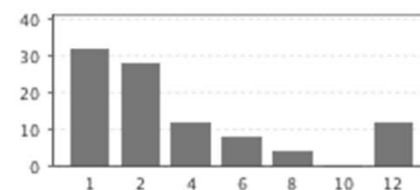
## Complexity

**4.5** /method

**40.0** /class

**44.0** /file

Total: 440



Methods  Classes

## Code coverage

**51.5%**

51.5% line coverage




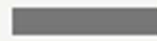


## Test success

**100.0%**

0 failures  
0 errors  
73 tests  
435 ms

# Ukážka: Sonar

## Severity

 <u>Blocker</u>	3
 <u>Critical</u>	14
 <u>Major</u>	254 
 <u>Minor</u>	6
 <u>Info</u>	6

## Rule

 <u>Method length exceeds maximum</u>	3
 <u>Class cyclomatic complexity exceed maximum</u>	10
 <u>Class NPath Complexity exceeds maximum</u>	4
 <u>FunctionCallSignatureIndent</u>	221 
 <u>ValidVariableNamePrivateNoUnderscore</u>	26 
 <u>LineLengthMaxExceeded</u>	4

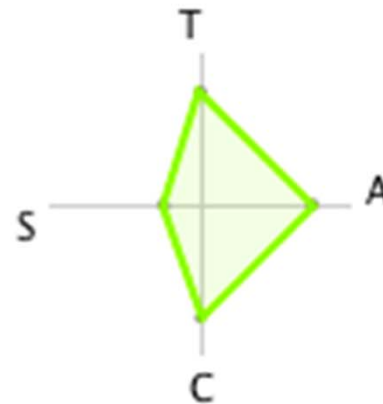
 <u>CodeCoverage.Report</u>	96
 <u>CodeCoverage.TextUI</u>	55
 <u>CodeCoverage</u>	49
 <u>CodeCoverage.Report.HTML</u>	44
 <u>(default)</u>	36
 <u>CodeCoverage.Report.HTML.Node</u>	3

 <u>Clover</u>	57
 <u>Command</u>	55
 <u>Node</u>	44
 <u>Util</u>	39
 <u>HTML</u>	39
 <u>CodeCoverage</u>	36

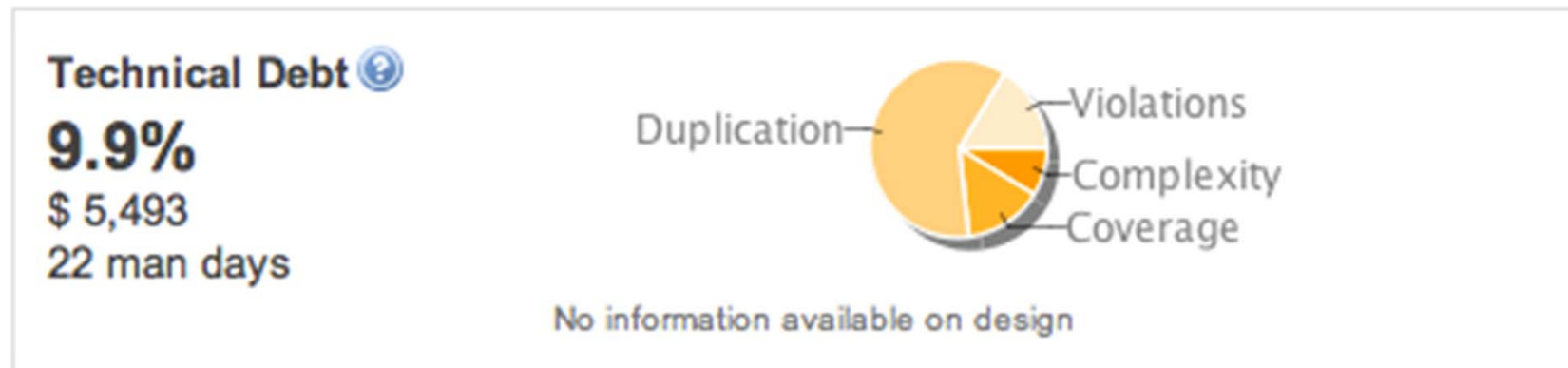
# Sonar: Model udržiavateľnosti (SIG Plugin)

## SIG Maintain. Model

(A)nalyzability	+
(C)hangeability	+
(S)tability	-
(T)estability	+



# Sonar: Technical Debt





# Záver k metrikám zdrojového kódu



- Nemeňte kód preto aby ste dosiahli lepšie čísla, meňte ho preto aby bol lepší
  - ▣ Meranie a metriky vás majú upozorniť na to, že by lepší mal byť

# Dynamické meranie programov



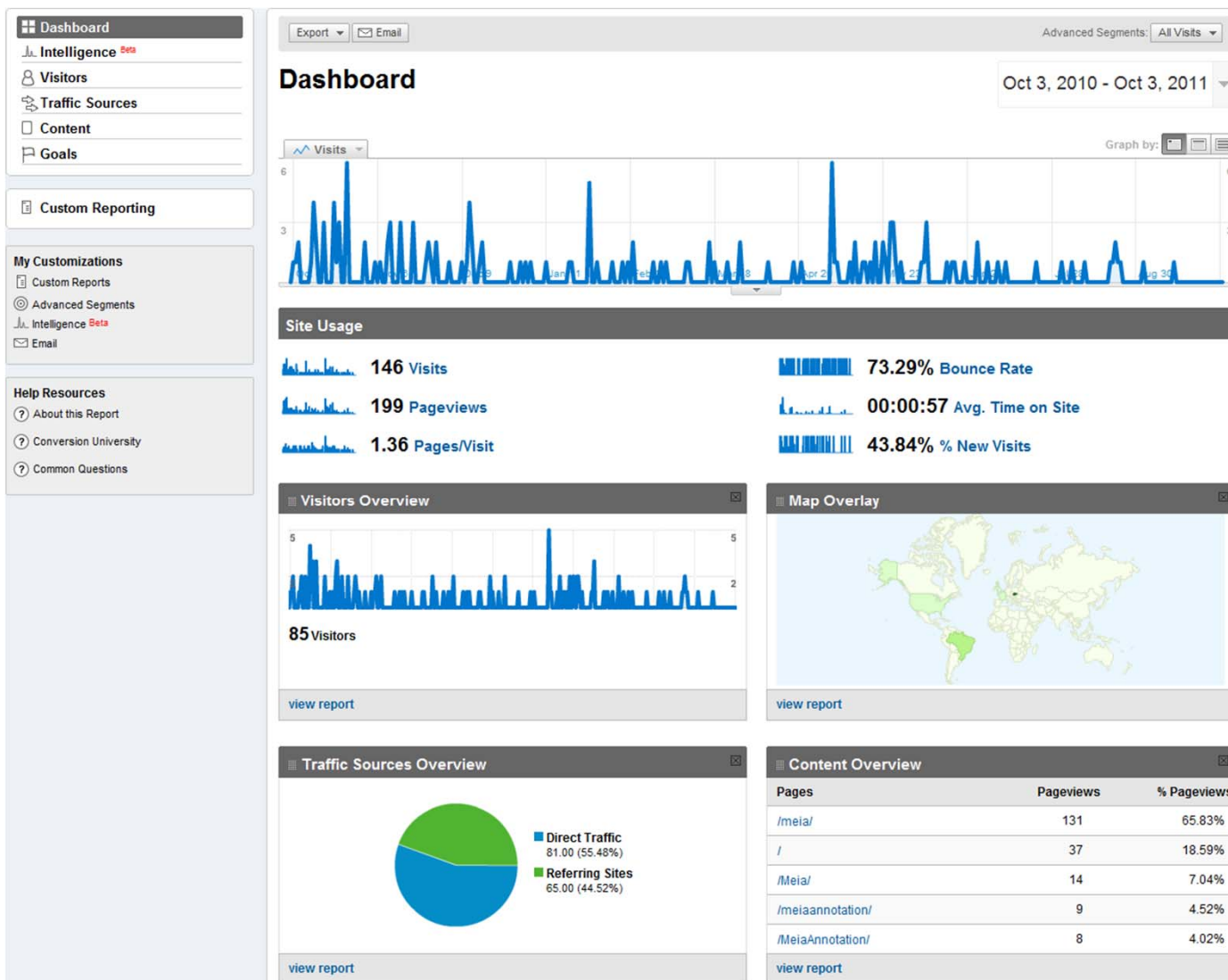
- Metriky pre spustiteľný kód
  - Spotreba pamäti a cyklov procesora (po riadkoch)
  - Doba odozvy, priepustnosť, zát'az', návštevnosť
  - Miera súbežnosti procesov a vláken
  - Počet chýb, zraniteľností, otvorených portov
- **... a priebeh týchto metrík v čase!**

# Meranie programov



- Robustnosť, pamäť, bezpečnosť
  - Nástroje od MS, HP, IBM (napr. Rational AppScan)
- **Výkonnosť, optimalizácia**
  - VS Profiler, Netbeans Profiler (DotTrace, JProfiler)
- Návštevnosť, záťaž, priepustnosť
  - Google Analytics, IIS / Apache monitorovanie

# Ukážka: Google Analytics



# Google Analytics – ako na to?



- Registrovať sa na Google
- Získať kľúč pre svoju stránku
- Pridať jednoduchý JavaScript kód ku stránke
- Prehliadať štatistiky

# Čo treba pridať?

```
<script type="text/javascript">
```

```
var gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." :  
  "http://www.");
```

```
document.write(unescape("%3Cscript src=\"" + gaJsHost + "google-  
  analytics.com/ga.js' type='text/javascript'%3E%3C/script%3E"));
```

```
</script>
```

```
<script type="text/javascript">
```

```
try {
```

```
var pageTracker = _gat._getTracker("XXX-KLUC-XXX");
```

```
pageTracker._trackPageview();
```

```
} catch(err) {}
```

```
</script>
```

# Odkazy



- Magic behind the numbers

- [http://www.youtube.com/watch?v=fxAb6\\_wRqg](http://www.youtube.com/watch?v=fxAb6_wRqg)

- <http://www.slideshare.net/proofek/magic-behind-the-numbers-software-metrics-in-practice>

- Software metrics and quality

- <http://www.youtube.com/watch?v=KqDIDubS-OU>

The ONLY VALID MEASUREMENT  
OF CODE QUALITY: WTFs/MINUTE

