7th Central and Eastern European Software Engineering Conference in Russia - CEE-SECR 2011

October 31 – November 3, Moscow

Mining source code changes from software repositories

Črt Gerlec, Andrej Krajnc, Marjan Heričko, Jan Božnik

crt.gerlec(at)uni-mb.si

Univerza v Mariboru

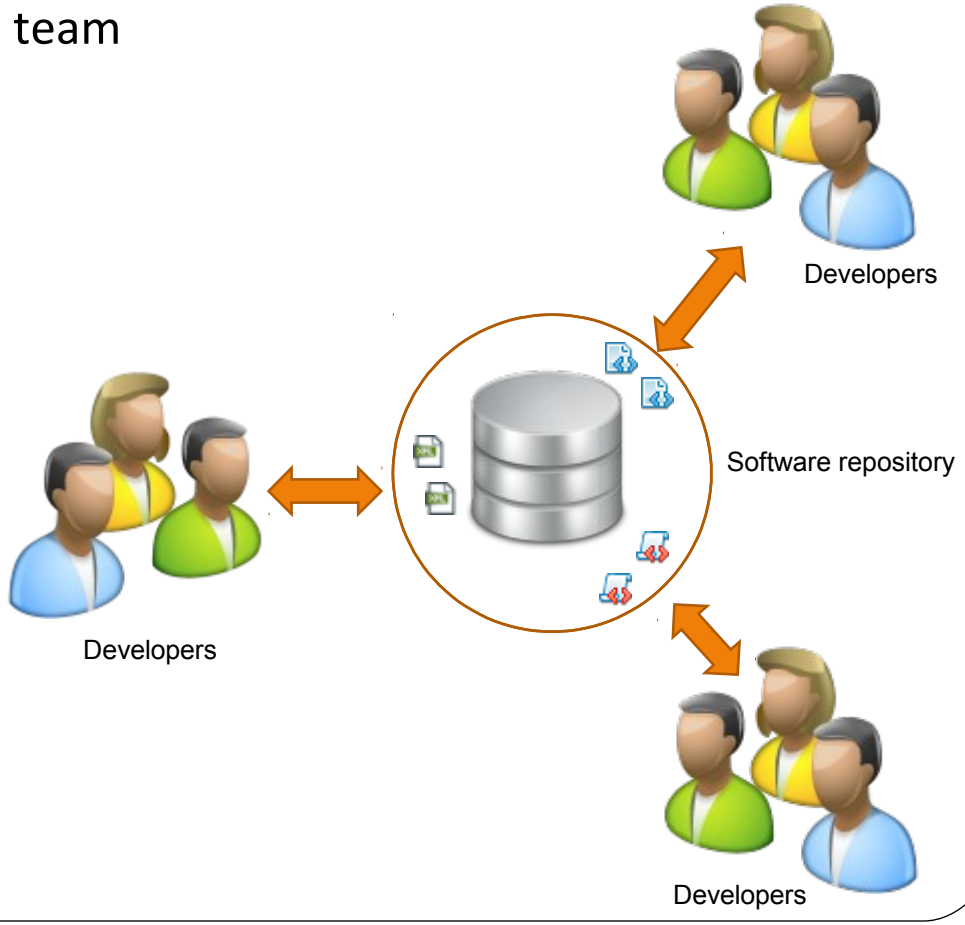
Fakulteta za elektrotehniko, računalništvo in informatiko

Agenda

- Mining software repositories
- Motivation
- Structural source code changes
- The tool for mining software repositories
- Source code change detection process
- Projects' analysis
- Results
- Conclusion

Mining software repositories

- Software repositories
  - Store the source code of software during its development
    - Share it over the development team
  - Store the documentation
- Store information about:
  - Software processes
  - Development processes
- Revision control
- CVS and Subversion
  - Check out
  - Commit



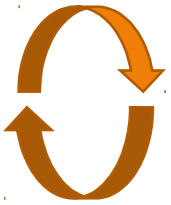
Motivation

- Find patterns in software development -> bugs
  - A tool for analyzing software evolution
    - Identification of structural source code changes between versions
  - Analyzing structural source code changes
    - Correlation with the number of bugs

Version n

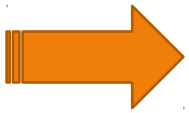
```
using System;

namespace Higuchi.Sample
{
    /// <summary>
    ///
    /// </summary>
    public partial class MUser
    {
        /// <summary>
        ///
        /// </summary>
        public enum UserType
        {
            /// <summary>
            /// General user
            /// </summary>
            Generic = 0,
            /// <summary>
            /// Special user
            /// </summary>
            Special = 1,
            /// <summary>
            /// Temporary user
            /// </summary>
            Temporary = 2
        }
    }
}
```

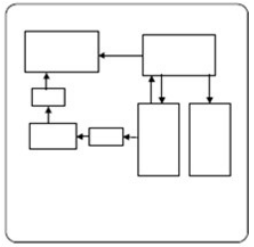


Version n + 1

```
public static void EnableResizeSupport(this Form form)
{
    string key = form.Handle.ToString();
    if (!state.ContainsKey(key)) {
        state.Add(
            form.Handle.ToString(),
            new FormResizeExtensionStateEntry
            {
                Form = form,
                MovableControls = new MovableControls(),
                PreviousSize = form.Size
            }
        );
        form.ResizeBegin += new EventHandler(form_ResizeBegin);
        form.SizeChanged += new EventHandler(form_SizeChanged);
        form.Resize += new EventHandler(form_Resize);
    }
}
```



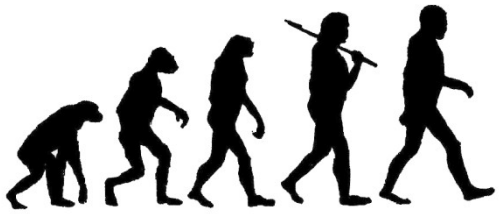
Structural source code changes



Basic terms and definitions

- Software repository (software archive)
- Version / revision
  - Developers
    - Commit action (e.g. file modification)
    - Change state
      - Added, modified, deleted
- Source code change
  - Defined as object-oriented changes on a class
    - When adding new functionality
    - Bug resolving and refactoring processes

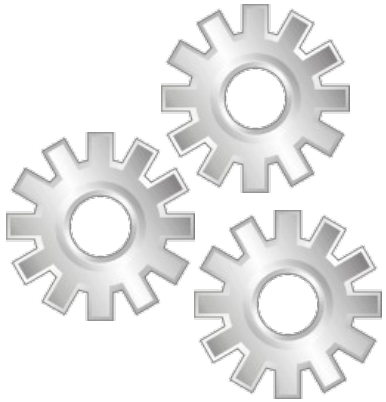
A tool for analyzing software evolution



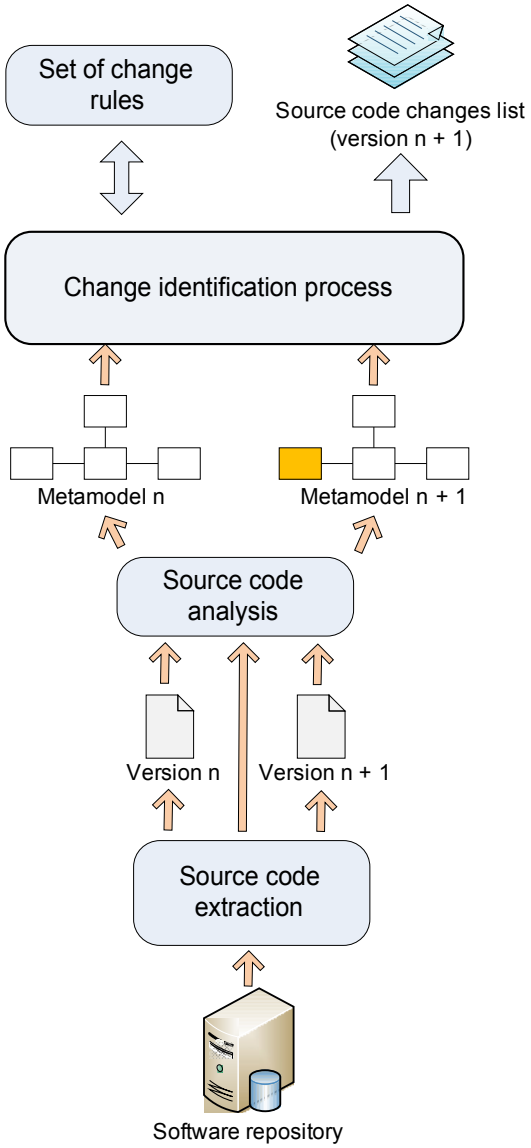
- Features
  - Mining software repositories
    - input: repository url
    - e.g. iterating through revisions in Subversion
  - Transformation process
    - Reshaping the data
  - Analysis
    - Comparing two subsequent revisions
    - Detection of structural changes between revisions

The tool's architecture

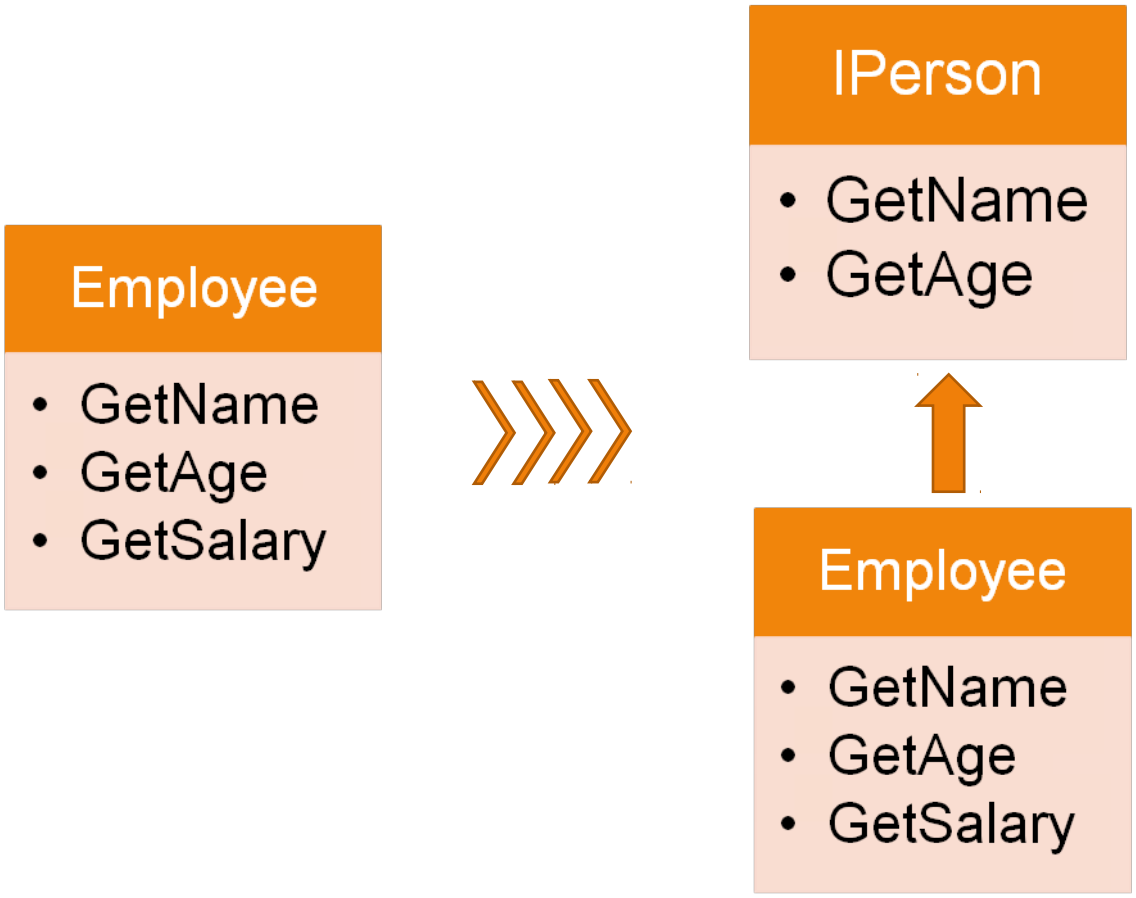
- Modules
  - SvnManager
    - Extracting the data from software repositories
      - List of source files (text)
    - Parsing a source code files to get object-oriented data
      - Metamodel for a class
      - StyleCop
        - Framework for style and consistency rules
        - Custom rule
  - RulesManager
    - Contains several rules
      - Identifying structural source code changes
    - Rules -> Metamodels -> source code changes



The tool and the identification process



Example: Extract interface change type



Rule example: extract interface

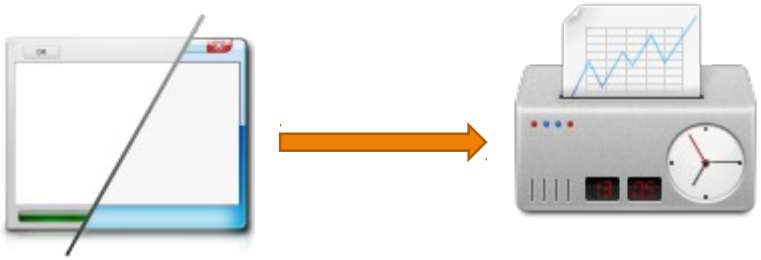
- Two classes in version *n* and *n+1*
  - C_n and C_{n+1}
- A new interface (implemented in C_{n+1})
  - I_{n+1}
- Rule definition
  - C_{n+1} implements the interface I_{n+1} & not implemented in C_n
  - the interface I_{n+1} has at least one method with the same name and parameters that exists in class C_n

Supported change types

- Simple changes
  - Add parameter, field and method
  - Remove parameter, field and method
  - Hide and unhide method
  - Rename method
  - Move attribute, method and class
  - Method body change
- Complex changes
  - Extract superclass, interface and class
  - Pull up field and method
  - Push down field and method
  - Inline class

Analyzing source code changes

- Open source projects
  - Nhibernate
    - Object-Relational mapper
  - CCNet
    - Automated continuous integration server
  - Lucene.Net
    - High-performance, full-featured text search engine library



NHibernate size metrics

- Analysis
  - 4 versions
    - LOC & # classes increased
    - # assemblies decreased
  - ~ 100k LOC
  - ~ 3500 classes

Version	NHibernate		
	Lines of code	Number of assemblies	Number of classes
1.2.1	63.079	28	1.795
2.0.1	64.550	7	2.083
2.1.0	97.977	11	3.337
2.1.2	99.392	11	3.473

* Measured with NDepend

CCNet size metrics

- Analysis
  - 7 versions
    - LOC, # classes & # assemblies increased
  - ~ 52k LOC
  - ~ 1700 classes

Version	CCNet		
	Lines of code	Number of assemblies	Number of classes
1.3	21.687	9	755
1.4	25.350	9	821
1.4.1	26.835	9	836
1.4.2	26.935	9	839
1.4.3	28.105	10	854
1.4.4	40.126	11	1.228
1.5	51.829	16	1.691

* Measured with NDepend

Lucene.Net size metrics

- Analysis
  - 5 versions
    - LOC & # classes increased
    - # assemblies stayed the same
  - ~ 30k LOC
  - ~ 730 classes

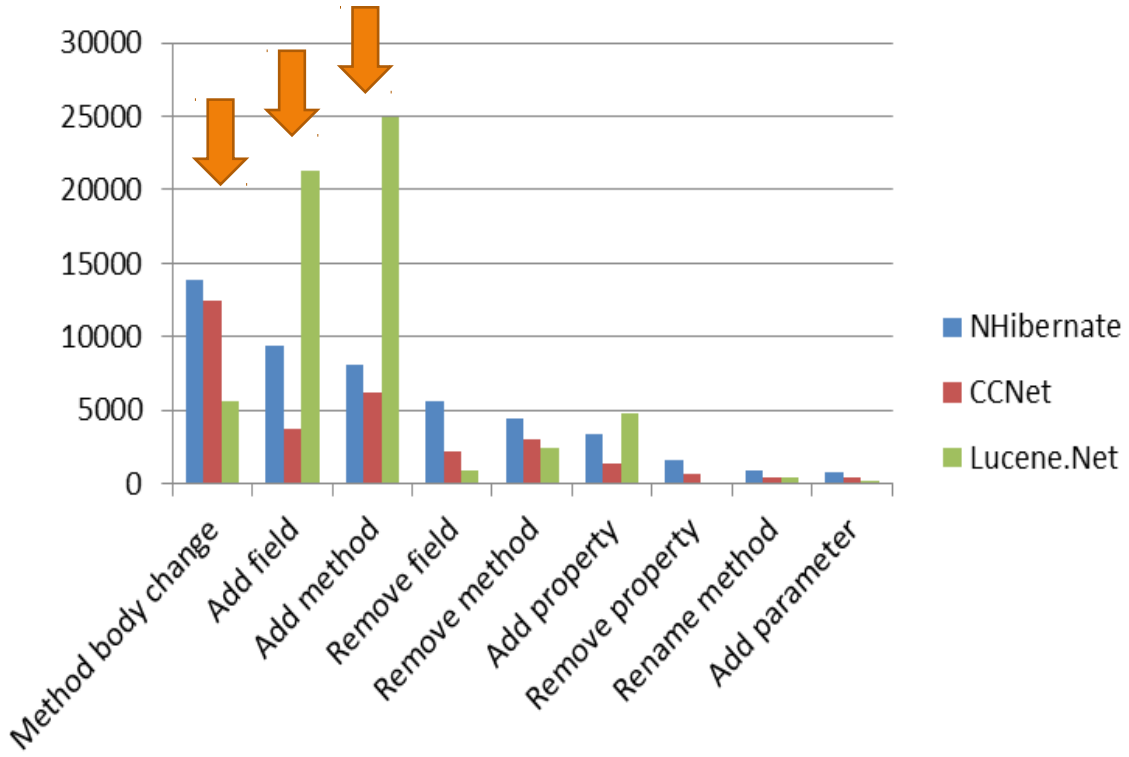
Version	Lucene.Net		
	Lines of code	Number of assemblies	Number of classes
2.0	11.937	1	276
2.1	13.889	1	310
2.4	23.233	1	525
2.9.1	29.401	1	716
2.9.2	29.452	1	726

* Measured with NDepend

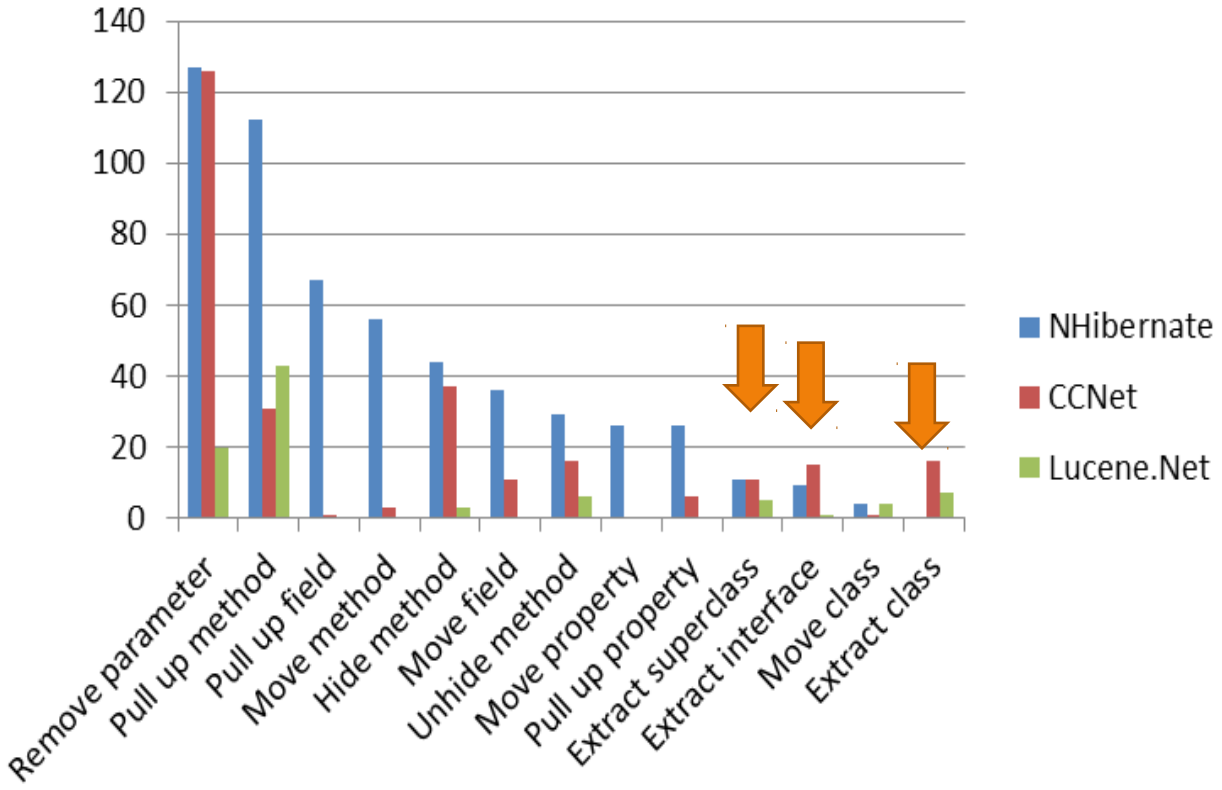
Change types	Analyzed projects		
	NHibernate	CCNet	Lucene.Net
Add parameter	772	390	140
Remove parameter	127	126	20
Add field	9.432	3.681	21.260
Remove field	5.574	2.201	906
Add property	3.344	1.407	4.817
Remove property	1.587	605	65
Add method	8.086	6.185	24.920
Remove method	4.448	3.055	2.436
Hide method	44	37	3
Unhide method	29	16	6
Rename method	944	396	417
Move field	36	11	0
Move property	26	0	0
Move method	56	3	0
Move class	4	1	4
Extract superclass	11	11	5
Extract interface	9	15	1
Extract class	0	16	7
Pull up field	67	1	0
Pull up property	26	6	0
Pull up method	112	31	43
Method body change	13.877	12.489	5.609
Sum	48.611	30.683	60.659

The largest!

Frequently used source code changes



Less frequently source code changes



Limitations

- Projects written in C#
- Software repository problems
  - Files that are not committed into the repository
    - New files
    - Modified/deleted files
  - A file is not syntactically correct
- Precision and accuracy of the tool
  - Not evaluated

Conclusion

- The tool for mining software repositories
  - The tool‘s architecture
  - The source code change identification process
    - Rules for identifying structural changes
  - List of supported structural source code changes
- Empirical analysis
  - Evaluation of open source projects
  - Successful identification of code changes
  - The method body change, add field and add method types were used frequently
  - More complex change types (e.g extract superclass, extract interface) were used rarely

Future work

- First step
  - Detection of code changes
- Next step
  - Correlation between
    - Structural source code changes
    - The number of bugs
  - Add more complex changes
  - The tool’s precision and accuracy

Thank you

Спасибо

It’s a lunch time ☺