

Project Codeviolation

<http://www.codeviolation.org>

Presenter: Petr Zajac

(Petr.Zajac@sun.com)

Petr Hřebejk

(Petr.Hrebejk@sun.com)

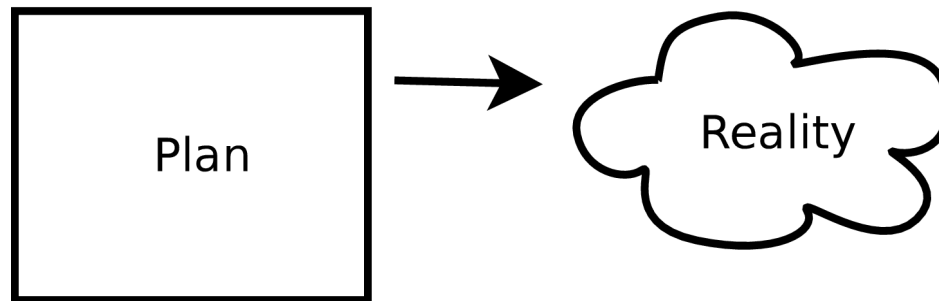
Date: 24.5.2007



Contents

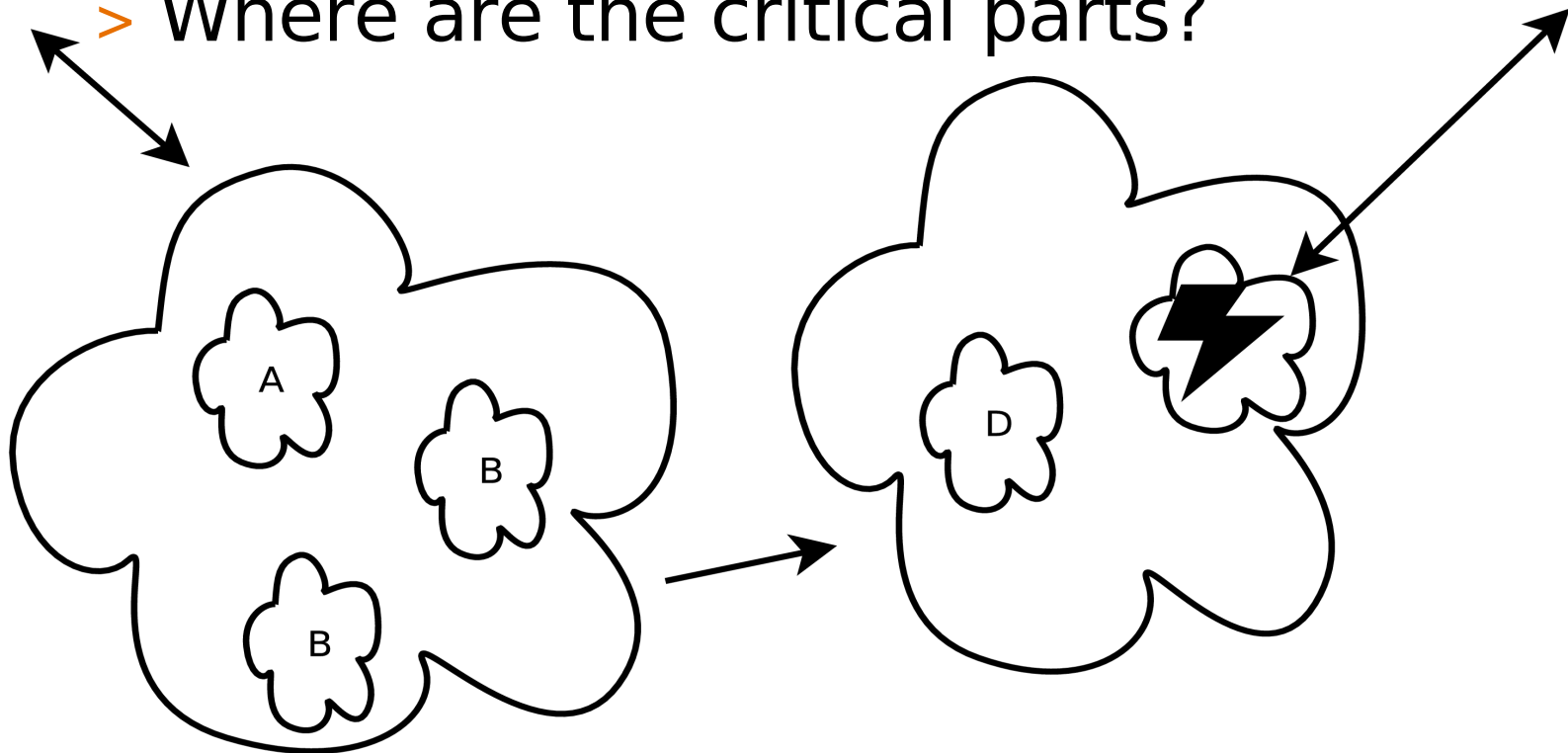
- Introduction to the problem
- Codeviolation project
- Mining the project's history
- Statistics
- Code importance by Google Page Rank
- Software reliability
- Demo

Simple program



Complex programs & evolution

- > More developers
- > Growing amount of code
- > Which library to use?
- > Maintenance problem
- > Where are the critical parts?



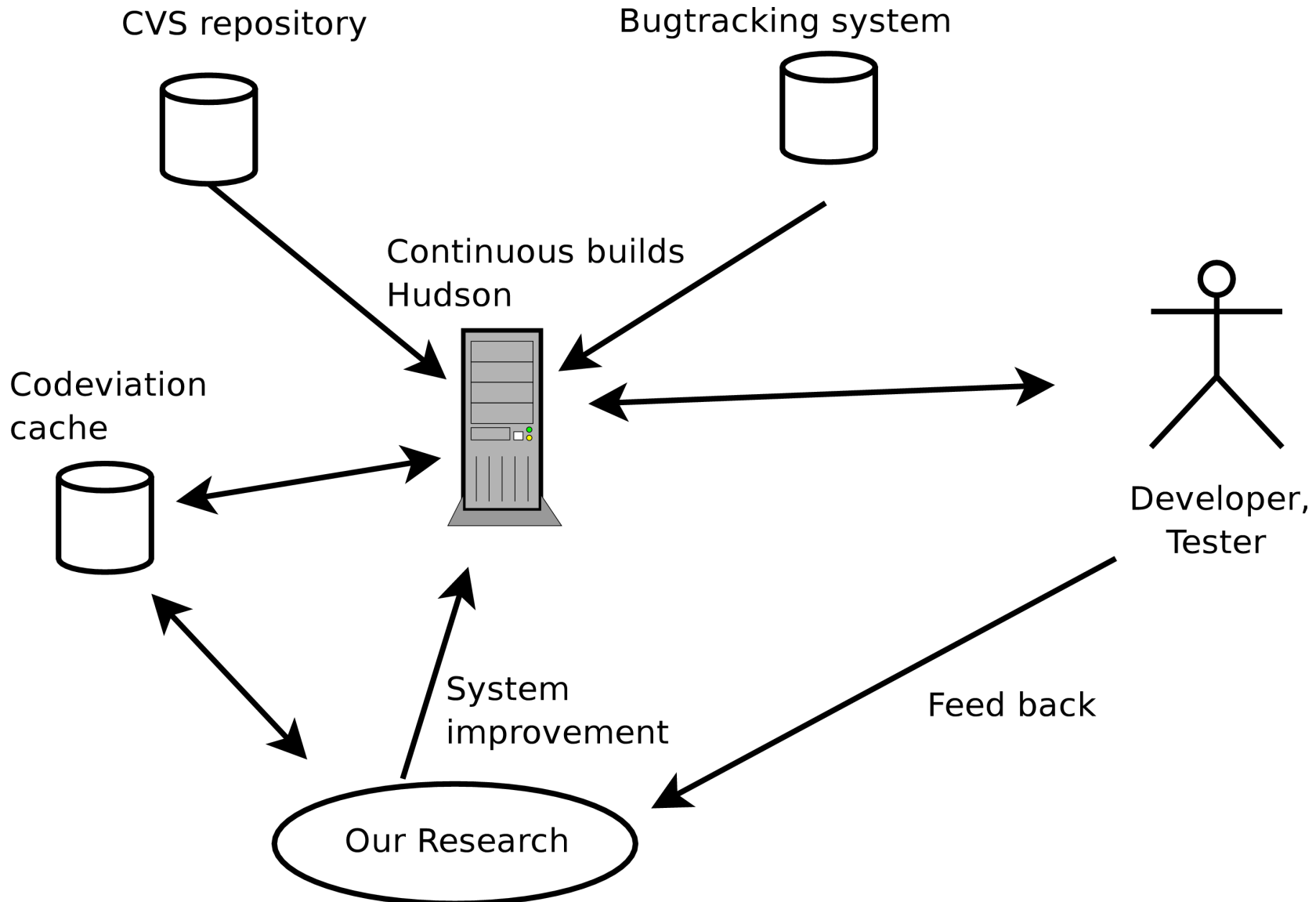
Project Codeviation

- Analyzing software metrics
- Connection between CVS and Bugs
- Mining semantics from history
- Predicting software reliability
- Participants
 - > Founders: Petr Hrebejk, Petr Zajac
 - > Few Sun employees
 - > Czech universities (CVUT, VSB)

Similar projects

- Software Evolution project
 - > <http://www.st.cs.uni-sb.de/softevo/>
- Hackystat
 - > <http://csdl.ics.hawaii.edu/Research/Hackystat/>
- Software Reliability Research
 - > <http://research.microsoft.com/srr/>

Architecture



Prototype

- Modified compiler
 - > Allows to mine metrics from java code
- Connecting with the history
 - > Who, what, when implemented/used
 - > Where testers found bugs
 - > How the bugs were fixed
 - > Evolution of complexity
- Integrated with Hudson cont. build system

Mining changes from CVS

- Execution cvs diff :
 - > cvs diff RequestPoster.java

Index: RequestPoster.java

=====

RCS file: /module/Request.java,v

retrieving revision 1.1

retrieving revision 1.2

diff -r1.1 -r1.2

44c44

< // empty comment

> String getV() {

> return v.trim();

> }

Reconstruction of lines' history

- We know a lot from history:
 - > who
 - > what
 - > when

1.1 (john 06-Jun-04):

1.1 (john 06-Jun-04): void main()

1.1 (john 06-Jun-04): { }

1.1 (john 06-Jun-04): String v;

1.2 (oleg 30-Jun-06): String getV(){

1.2 (oleg 30-Jun-06): return v.trim();

1.2 (oleg 30-Jun-06): }

Sources <-> Bugtracking system

- Making connection from cvs log
- Two rules
 - > Number exists in the bug database
 - > Similar log's and resolution's date
- Example of log:

revision 1.7

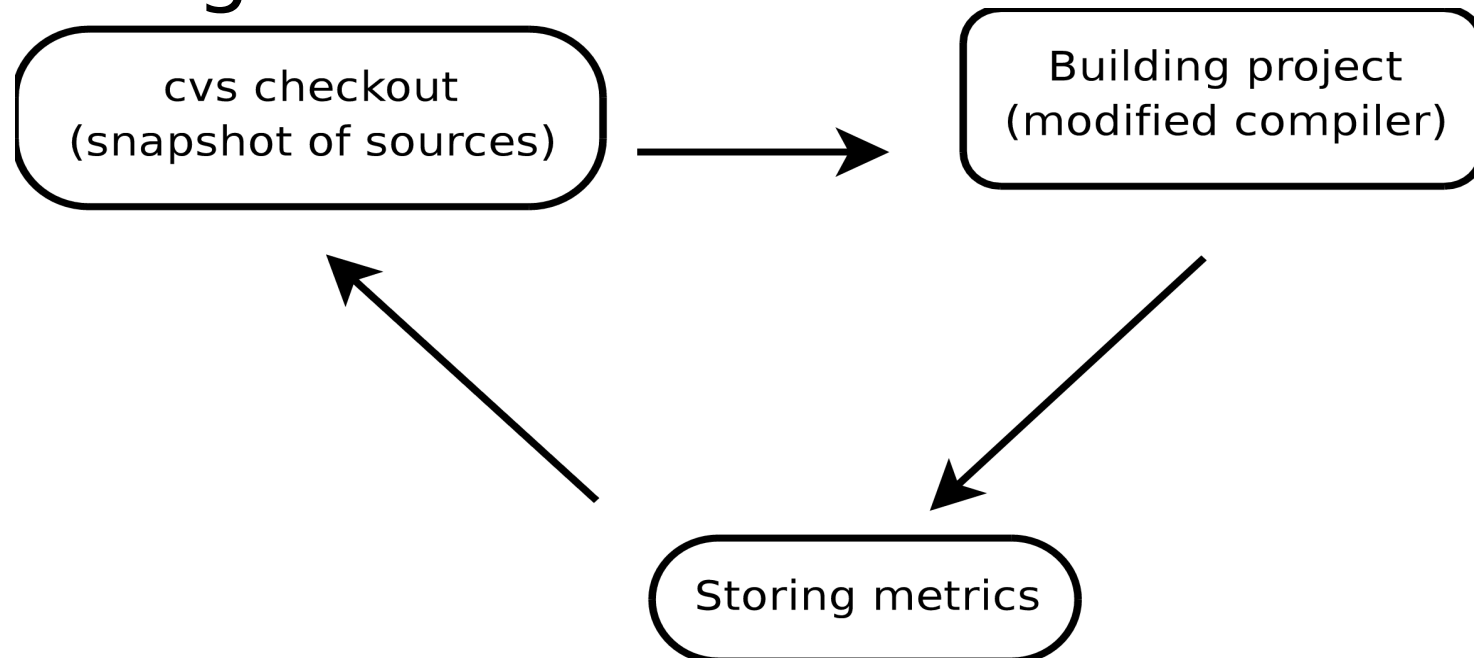
date: 2006/05/26 08:35:28;

author: john;

#74869 EventDispatchThread hang

Mining semantics from history

- building of projects in loop for different cvs snapshots
- storing metrics



Revisions & metrics – whole file

- For example you want to store number classes, methods and fields
- Example

Version 1.2:
1 class, 1 method
0 fields

```
class Hello {  
    void hello() {  
        print("hello");  
    }  
}
```

Version 1.3
1 class 2 methods,
1 field

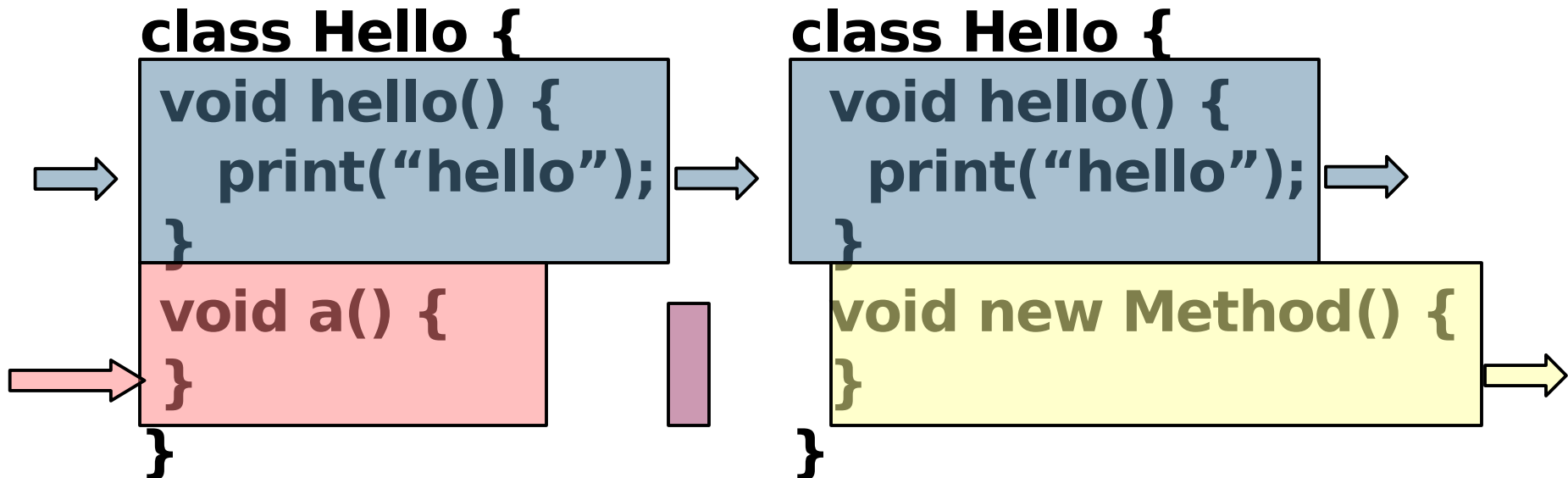
```
class Hello {  
    int a;  
    void hello() {  
        print("hello")  
    }  
    void secondMethod()  
}
```

Versioning metrics - blocks

- Developers change small ranges
- Saves disk space
- Monitoring evolution

Version 1.2

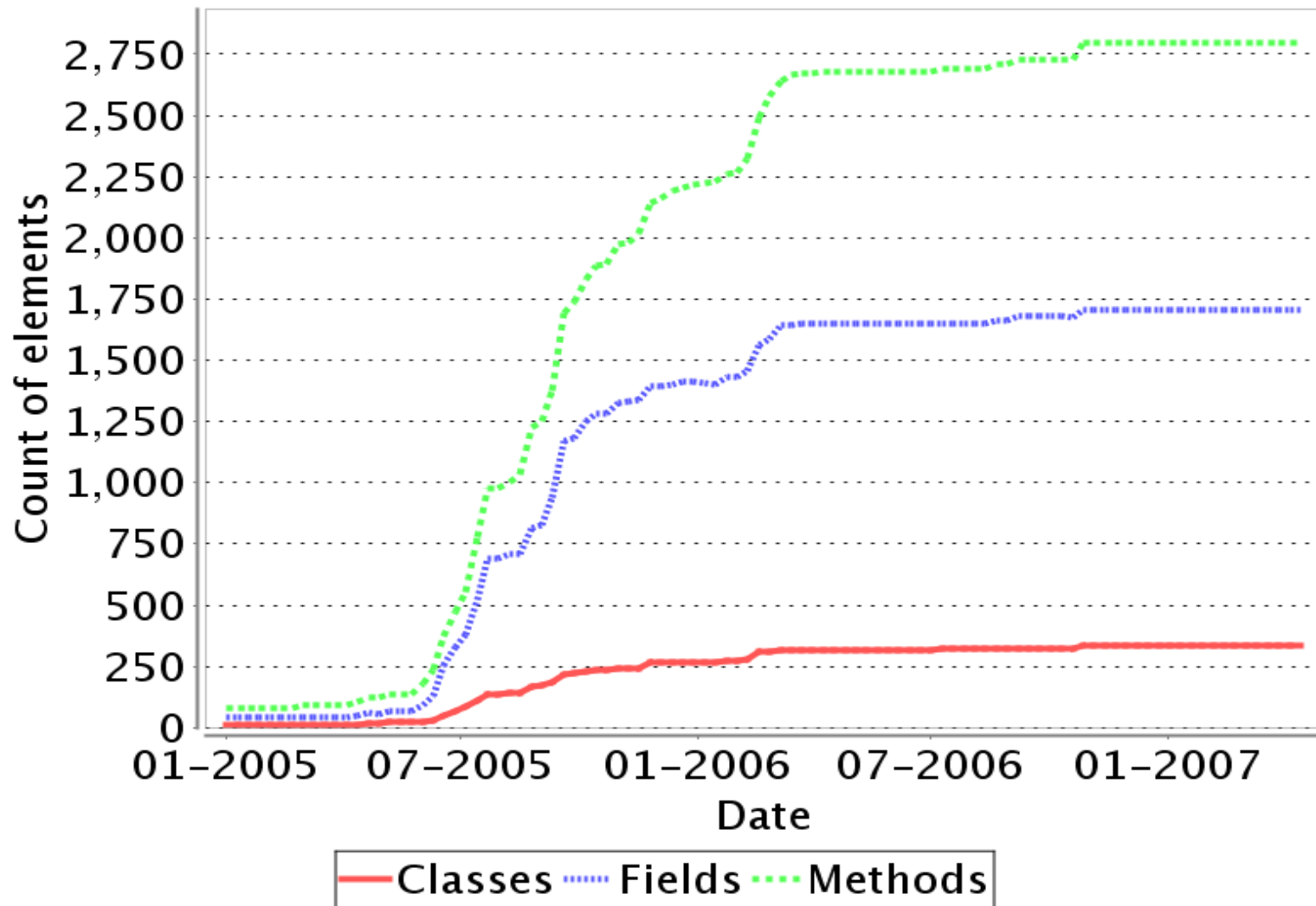
Version 1.3



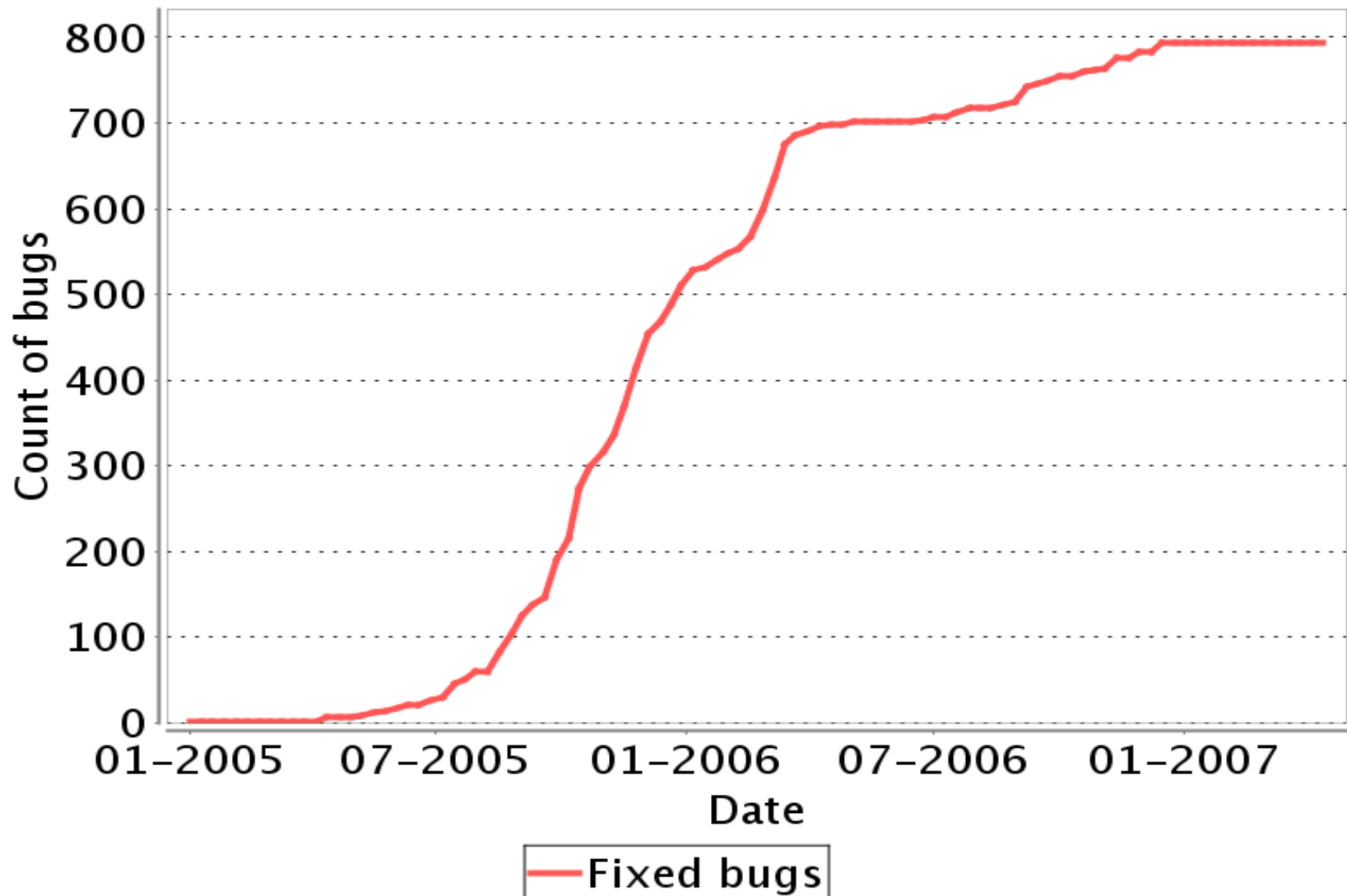
First Statistics

- Java elements counts
- Count of fixed bugs
- Lines count
 - > new features
 - > fixing of bugs
- Integrated with Hudson

History of elements



Bugs fixing history



Code importance by ClassRank

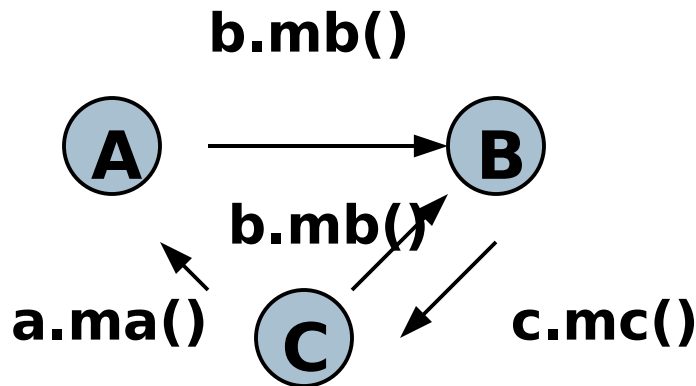
- Google's PageRank Algorithm
 - > Measures importance of web page
 - > Web pages are connected by links - graph
- Application on software
 - > Classes are connected by methods usages and inheritance
 - > ClassRank

How to construct the graph

```
class A {
  B b;
  void ma() {
    b.mb();
  }
}
```

```
class B {
  C c;
  void mb() {
    c.mc();
  }
}
```

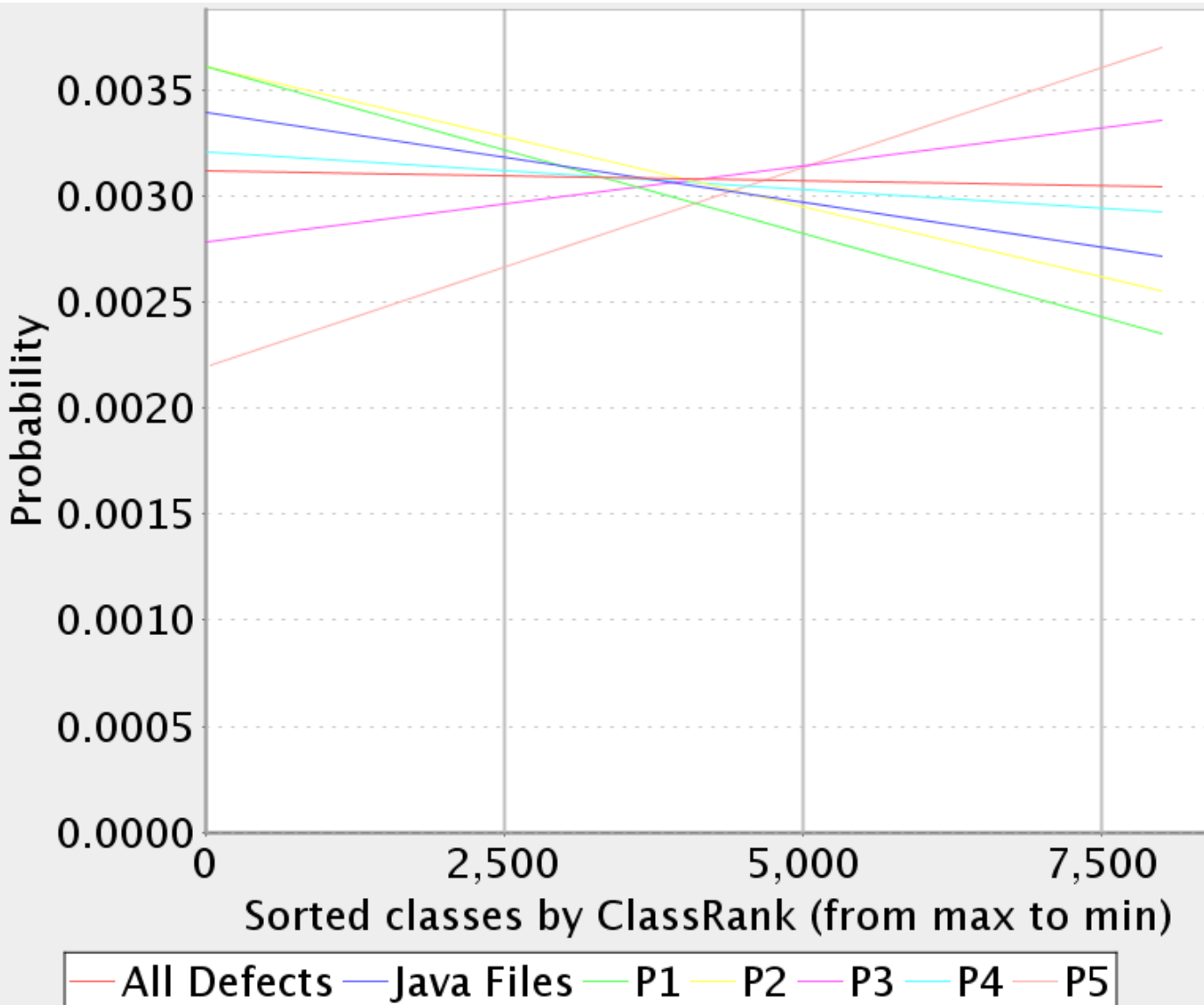
```
class C {
  A a;
  B b;
  void mc() {}
  void mc2() {
    b.mb();
    a.ma();
  }
}
```



Bugs and code importance

- Bugs importance (P1, P2, P3, P4, P5)
- Code importance – sorted by ClassRank
- Important bugs are in important code
 - > More P1 bugs in classes with highest ClassRank than P5 bugs
 - > More P5 bugs in classes with lowest ClassRank than P1 bugs

Linear regression



Future Plans(1)

- Analyzing software reliability
- Mixing All information
 - > History of metrics
 - > NetBeans Exceptions reporter
 - > Issuezilla
 - > NetBeans UI Gestures logger
- Similar projects have result from textual representation of sources
- Expected better results

Future Plans (2)


- More Metrics
- Measuring more projects
- Comparing the projects
- Integration with Development environment (NetBeans)


Practical Applications


- Evaluation of commits
- Comparing projects/choosing the right library or service to use
- Continuous monitoring of software quality
- Evaluating software produced by external company before buying
- Due diligence before SW companies acquisitions
- ...


Demo – project's setup in Hudson


Build Environment


Codeviation metrics 

First tag's date 

Last tag's date 


Step in days 

Repository mapping 

ANT_OPTS 

ANT_OPTS environment variable for ant

Build

Execute shell 

Command ant -f pantexamples/build.xml clean jar"/>

See [the list of available environment variables](#)

Demo – metrics report in Hudson

Hudson

Hudson » [c](#) » [cvsexamples](#) » [db/src](#)

[ENABLE AUTO REFRESH!](#)

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Delete Project](#)

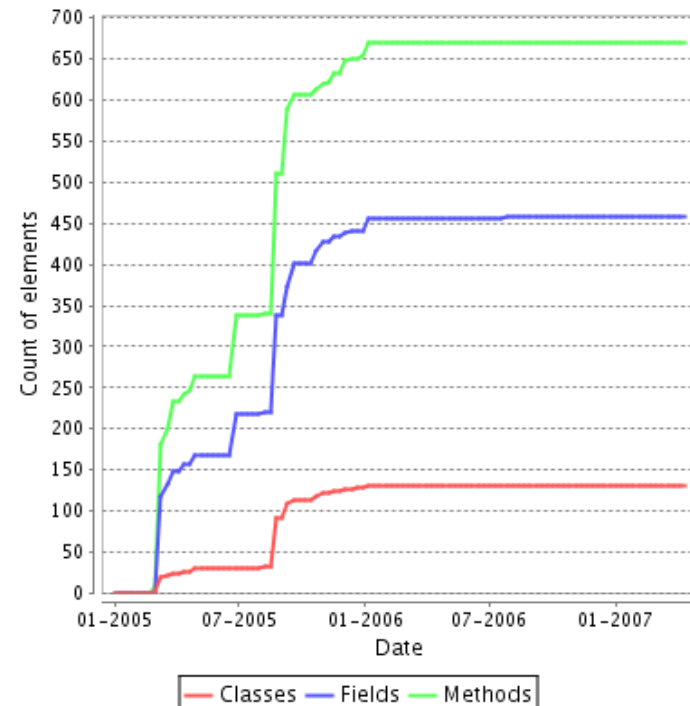
[Configure](#)

[Projects Metrics](#)

Module (source root) view: db/src

Package name
org.netbeans.api.db.explorer
org.netbeans.api.db.explorer.support
org.netbeans.modules.db
org.netbeans.modules.db.explorer
org.netbeans.modules.db.explorer.actions
org.netbeans.modules.db.explorer.dataview
org.netbeans.modules.db.explorer.dlg
org.netbeans.modules.db.explorer.driver
org.netbeans.modules.db.explorer.infos
org.netbeans.modules.db.explorer.nodes
org.netbeans.modules.db.explorer.sql.editor
org.netbeans.modules.db.runtime
org.netbeans.modules.db.spi.sql.editor
org.netbeans.modules.db.util
org.netbeans.spi.db.explorer

View Type



Build History (trend)

- #486 [May 11, 2007 2:07:50 PM](#)
- #485 [May 11, 2007 2:07:41 PM](#)
- #484 [May 11, 2007 2:07:35 PM](#)
- #483 [May 11, 2007 2:07:29 PM](#)
- #482 [May 11, 2007 2:07:23 PM](#)
- #481 [May 11, 2007 2:07:17 PM](#)
- #480 [May 11, 2007 2:07:11 PM](#)
- #479 [May 11, 2007 2:07:04 PM](#)
- #478 [May 11, 2007 2:06:57 PM](#)
- #477 [May 10, 2007 10:58:41 AM](#)
- #476 [May 10, 2007 10:58:35 AM](#)
- #475 [May 10, 2007 10:58:29 AM](#)
- #474 [May 10, 2007 10:58:23 AM](#)
- #473 [May 10, 2007 10:58:17 AM](#)

Q&A

References

- <http://www.codeviolation.org>
- Hudson, <https://hudson.dev.java.net/>