The Pragmatic Programmers

Investigator:
Date:
Case #:
Location:

Your Code as a Crime Scene

Use Forensic Techniques
to Arrest Defects, Bottlenecks, and
Bad Design in Your Programs

1. Evolving Software

1. Evolving Software
2. Dissect your Architecture
3. Master the Social Aspects of Code

Richard Wettel

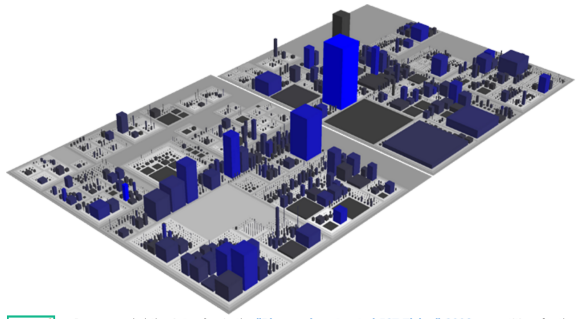
Home | Publications | Research | CodeCity | DuDe | Activities | Contact

Description | Download | Video Tutorials | Movies | FAQ | Release History | Wall of Fame

Welcome to CodeCity!

CodeCity is an integrated environment for software analysis, in which software systems are visualized as interactive, navigable 3D cities. The classes are represented as buildings in the city, while the packages are depicted as the districts in which the buildings reside. The visible properties of the city artifacts depict a set of chosen software metrics, as in the polymetric views of CodeCrawler.

You can read more about my approach in the ICPC 2007, VISSOFT 2007, Softvis 2008, WASDett 2008, WCRE 2008, FAMOOSr 2008 papers and the ICSE 2008 tool demo on my **Publications** page. If you would rather prefer a hands-on learning process, **download** CodeCity, run it, and tell me what you think. I appreciate every piece of feedback I get. CodeCity is programmed in VisualWorks Smalltalk on top of the **Moose** platform, uses OpenGL for rendering, and runs on every major platform. To see CodeCity in action visit the **Wall of Fame**. Here is a sneak preview with a CodeCity visualization of JDK (Java Development Kit) v1.5:



I was awarded the **1st prize** in the **"Riconoscimento ated-ICT Ticino" 2008** competition, for the project: "CodeCity: Immersive Software Visualization". The 2008 competition was disputed among 18 projects, including 5 from the IT industry and 11 from the Academia. The **ATED-ICT association** promotes competencies in the field of IT in the Ticino canton in Switzerland.

CodeCity appeared in a guest column of IEEE Software Journal (Volume 26, Issue 1, Jan.-Feb. 2009, pp 22-23), called **Tool Building on the Shoulders of Others**.

Last modified on: 19 November 2015

Pro & Cons ???

```
git clone

https://github.com/hibernate/hibernate-orm.git

https://github.com/adamtornhill/code-maat.git
```

```
Cristinas-MacBook-Pro:code-maat cristina$ lein run -l hib_evo.log -c git -a summary
statistic,value
number-of-commits,6561
number-of-entities,34988
number-of-entities-changed,124308
number-of-authors,251
```

```
Cristinas-MacBook-Pro:code-maat cristina$ lein run -l hib_evo.log -c git -a revisions
entity,n-revs
build.gradle,294
hibernate-core/src/main/java/org/hibernate/persister/entity/AbstractEntityPersister.java,137
parent/pom.xml,127
hibernate-core/src/main/java/org/hibernate/internal/SessionFactoryImpl.java,111
pom.xml,106
hibernate-core/src/main/java/org/hibernate/cfg/Configuration.java,101
core/pom.xml,96
hibernate-core/hibernate-core.gradle,94
testsuite/pom.xml,91
hibernate-core/src/main/java/org/hibernate/internal/SessionImpl.java,86
hibernate-core/src/main/java/org/hibernate/dialect/Dialect.java,86
entitymanager/pom.xml,78
cache-jbosscache/pom.xml,77
testing/pom.xml,76
```

```
git log –numstat

. . .
commit 97dc286a67fe342ccbfe4ecbad8e47f0086916f3
Author: Adam Petersen <adam@adampetersen.se>
Date:   Fri Aug 9 15:38:50 2013 +0200

    Introduced entities

4   8   src/code_maat/analysis/authors.clj
6   0   src/code_maat/analysis/entities.clj
3   11  test/code_maat/analysis/authors_test.clj
8   0   test/code_maat/analysis/entities_test.clj
10  0   test/code_maat/analysis/test_data.clj
```

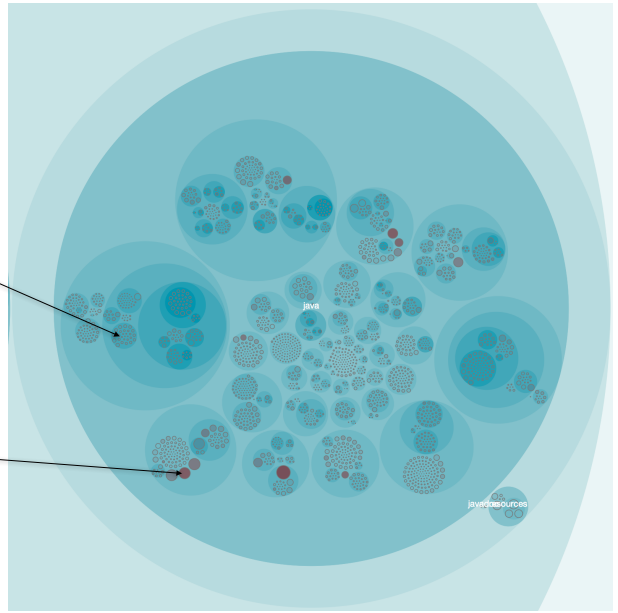
Added lines

Removed lines

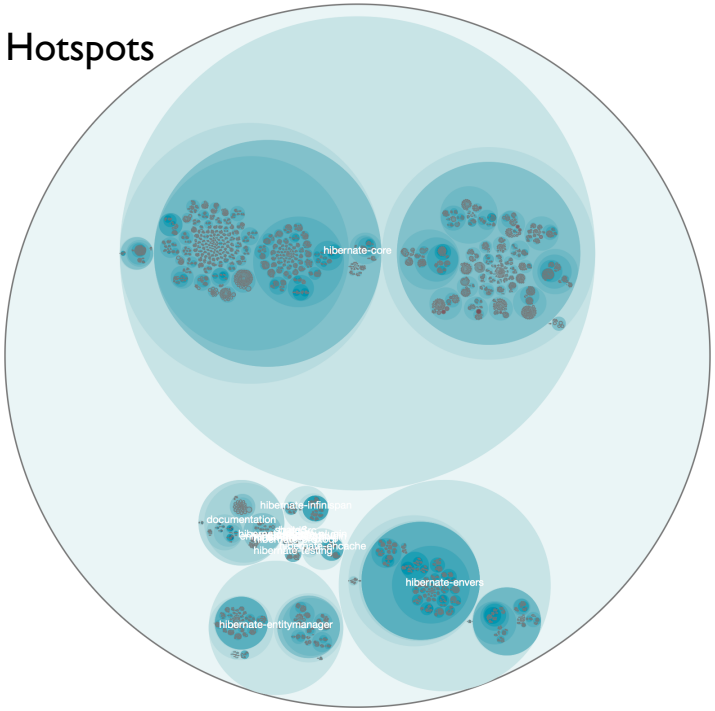
```
cloc #counting LOC

language,filename,blank,comment,code
Java,./hibernate-entitymanager/src/main/java/org/hibernate/jpa/spi/AbstractEntityManagerImpl.java,182,74,1512
Java,./hibernate-core/src/test/java/org/hibernate/test/readonly/ReadOnlyProxyTest.java,166,30,1507
Java,./hibernate-core/src/main/java/org/hibernate/cfg/annotations/CollectionBinder.java,126,96,1452
XML,./documentation/src/main/docbook/manual-old/en-US/content/performance.xml,331,8,1415
XML,./documentation/src/main/docbook/devguide-old/en-US/chapters/query_ql/HQL_JPQL.xml,58,6,1385
```

```
python merge_comp_freqs.py hibernate_freqs.csv
hibernate_lines.csv

module,revisions,code
build.gradle,294,233
hibernate-core/src/main/java/org/hibernate/persister/entity/AbstractEntityPersister.java,137,4187
hibernate-core/src/main/java/org/hibernate/internal/SessionFactoryImpl.java,111,1209
hibernate-core/src/main/java/org/hibernate/cfg/Configuration.java,101,407
hibernate-core/hibernate-core.gradle,94,96
hibernate-core/src/main/java/org/hibernate/dialect/Dialect.java,86,1062
hibernate-core/src/main/java/org/hibernate/internal/SessionImpl.java,86,2550
libraries.gradle,73,64
```

Visualising Hotspots



Visualising Hotspots



Heuristics

Hotspots account for 4 to 6 percent of the total codebase (measured using LOC).

Search for Bad Names - identify true/false positives!

Complexity measured as Indentation

```
/**
 * deposit the specified amount in the specified account
 */
private boolean depositAndTransfer(int accountP, int amountP, boolean opP)
{
int i=positionAccount(accountP);
if (i!=-1)
{
    accountList[i][1]=accountList[i][1]+amountP;
    if (opP == true)
    {
        operations[i].append("\nDeposit to " + accountP + " " + amountP);
        operationsCancel.append("\nDeposit to " + accountP + " " + amountP);
    }
    try
    {
        Statement statement = c.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                ResultSet.CONCUR_UPDATABLE);
        String update = "UPDATE bank SET amount = " +
                accountList[i][1] +
                " WHERE userId = " +
                userId +
                " AND account = " +
                accountP;
        statement.executeUpdate( update );
        statement.close();

        if(opP == true)
        {
            String op = new String(operations[i].toString());
            Statement statementOp = c.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                ResultSet.CONCUR_UPDATABLE);
            String updateOp = "UPDATE bank SET oper = '" +
                op + "'" +
                " WHERE userId = " +
                userId +
                " AND account = " +
                accountP;
            statementOp.executeUpdate( updateOp );
            statementOp.close();
        }
        c.commit();
    }
    catch ( SQLException sqlex )
    {
        sqlex.printStackTrace();
    }
}
else
{
    //nu e un cont pt. acest userID, poate fi un cont de la aceasta banca
    //sau un cont de la alta banca; contul pt. alta banca are 5 cifre
    String accountDeposit = new String(Integer.toString(accountP));
    if (accountDeposit.length()==4)
    {
        //e cont din aceasta banca
        try
        {
            /*Statement statementBank = c.createStatement();
            String queryBank = "SELECT * FROM bank WHERE account = " + accountP;
            ResultSet rsBank = statementBank.executeQuery(queryBank);
            rsBank.next();

            int amount = rsBank.getInt("amount");*/
            int amount = transferAmount + amountP;

            Statement statementOp = c.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                ResultSet.CONCUR_UPDATABLE);
            String updateOp = "UPDATE bank SET amount = " +
                amount +
                " WHERE account = " +
                accountP;
            statementOp.executeUpdate( updateOp );
            statementOp.close();
```

?

```
private KeyPad()
{
    JPanel panel = new JPanel();
    JPanel menu = new JPanel();
    JPanel panel1 = new JPanel();
    JPanel panel2 = new JPanel();
    JPanel panel3 = new JPanel();
    JPanel panel4 = new JPanel();

    panel.setLayout(new BorderLayout());
    panel1.setLayout(new FlowLayout(FlowLayout.LEFT));
    panel2.setLayout(new FlowLayout(FlowLayout.LEFT));
    panel3.setLayout(new FlowLayout(FlowLayout.LEFT));
    panel4.setLayout(new FlowLayout(FlowLayout.LEFT));

    menu.setLayout(new GridLayout(0,1,-5,-5));

    zero = new JButton("0");
    unu = new JButton("1");
    doi = new JButton("2");
    trei = new JButton("3");
    patru = new JButton("4");
    cinci = new JButton("5");
    sase = new JButton("6");
    sapte = new JButton("7");
    opt = new JButton("8");
    noua = new JButton("9");

    tripleZero = new JButton(".000");

    okButton = new JButton(Messages.messages[Messages.OK_BUTTON]);
    cancelButton = new JButton(Messages.messages[Messages.CANCEL_BUTTON]);
    deleteButton = new JButton(Messages.messages[Messages.DEL_BUTTON]);

    Insets ins = cancelButton.getMargin();

    okButton.setMargin(new Insets(ins.top, ins.left+5, ins.bottom, ins.right+5));
    cancelButton.setMargin(new Insets(ins.top, 8, ins.bottom, 8));
    deleteButton.setMargin(new Insets(ins.top, 9, ins.bottom, 9));
    tripleZero.setMargin(new Insets(ins.top, 5, ins.bottom, 5));

    Messages.addComponent(okButton, Messages.OK_BUTTON);
    Messages.addComponent(cancelButton, Messages.CANCEL_BUTTON);
    Messages.addComponent(deleteButton, Messages.DEL_BUTTON);

    Messages.addComponent(status, Messages.PRESS_OK);

    unu.addActionListener(digitListener);
    doi.addActionListener(digitListener);
    trei.addActionListener(digitListener);
    patru.addActionListener(digitListener);
    cinci.addActionListener(digitListener);
    sase.addActionListener(digitListener);
    sapte.addActionListener(digitListener);
    opt.addActionListener(digitListener);
    noua.addActionListener(digitListener);
    zero.addActionListener(digitListener);

    okButton.addActionListener( new ActionListener()
```

Complexity - how did it evolve?

```
python ../../maat-scripts/miner/git_complexity_trend.py --start ccc087b --end 46c962e --file hibernate-core/src/main/java/org/hibernate/cfg/Configuration.java
```

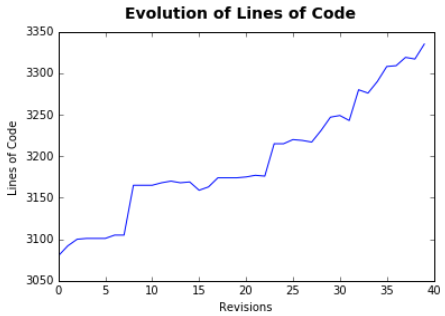
```
rev,n,total,mean,sd
e75b8a7,3080,7610,2.47,1.76
23a6280,3092,7649,2.47,1.76
8991100,3100,7658,2.47,1.76

5ea40ce,3319,8022,2.42,1.63
1825a47,3317,8022,2.42,1.63
580a713,3335,8072,2.42,1.63
```

```
python complexity_analysis.py hibernate/hibernate-orm/hibernate-core/src/main/java/org/hibernate/cfg/Configuration.java
```

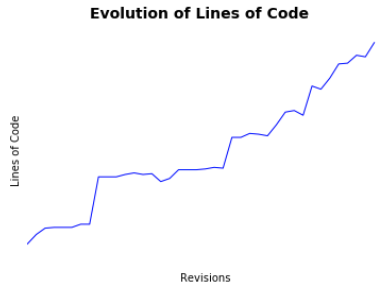
```
n,total,mean,sd,max
769,933,1.21,0.76,6
```

```
import pandas as pd

ev = pd.read_csv('/Users/cristina/Desktop/CodeMaat/configuration-complexity-evolving.csv', header = 0, delimiter = ",")
```

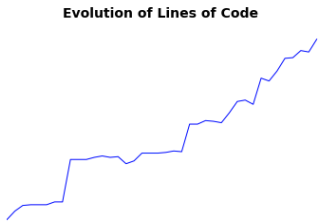
```
import matplotlib.pyplot as plt

%matplotlib inline

plt.plot(range(0,ev.shape[0]), ev['n'])
plt.suptitle('Evolution of Lines of Code', fontsize=14, fontweight='bold')
plt.ylabel('Lines of Code')
plt.xlabel('Revisions')

plt.show()
```



```
def format_plot_axes(currentPlot):
    currentPlot.spines["top"].set_visible(False)
    currentPlot.spines["bottom"].set_visible(False)
    currentPlot.spines["right"].set_visible(False)
    currentPlot.spines["left"].set_visible(False)

    for xlabel_i in currentPlot.axes.get_xticklabels():
        xlabel_i.set_visible(False)
    for xlabel_i in currentPlot.axes.get_yticklabels():
        xlabel_i.set_visible(False)
    for tick in currentPlot.axes.get_xticklines():
        tick.set_visible(False)
    for tick in currentPlot.axes.get_yticklines():
        tick.set_visible(False)

plt.plot(range(0,ev.shape[0]), ev['n'])
plt.suptitle('Evolution of Lines of Code', fontsize=14, fontweight='bold')
plt.ylabel('Lines of Code')
plt.xlabel('Revisions')

format_plot_axes(plt.gca())
plt.show()
```

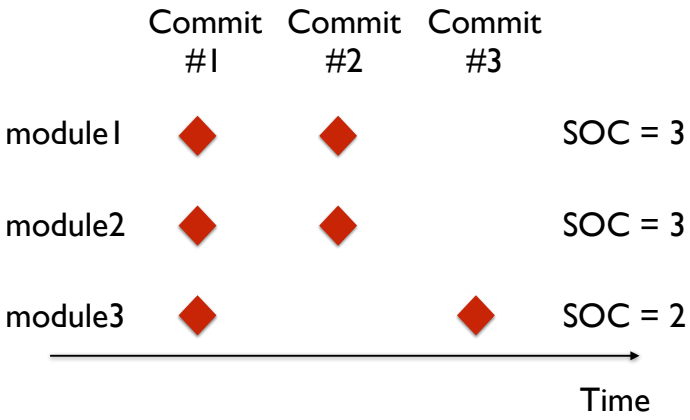


2. Dissect your Architecture



Temporal Coupling
implicit dependency
modules change together!

Temporal Coupling - SOC (Sum of Coupling)



SOC - ∑how many times each module has been coupled to another one in a commit

Measure Temporal Coupling

```
maat -l hib_evo.log -c git -a coupling
```

```
entity,coupled,degree,average-revs
hibernate-core/src/main/java/org/hibernate/cfg/HbmBinder.java,hibernate-core/src/main/java/org/hibernate/cfg/annotations/EntityBinder.java,30,30
```

degree - percent of shared commits

average-revs - weighted number of total revisions for the involved modules

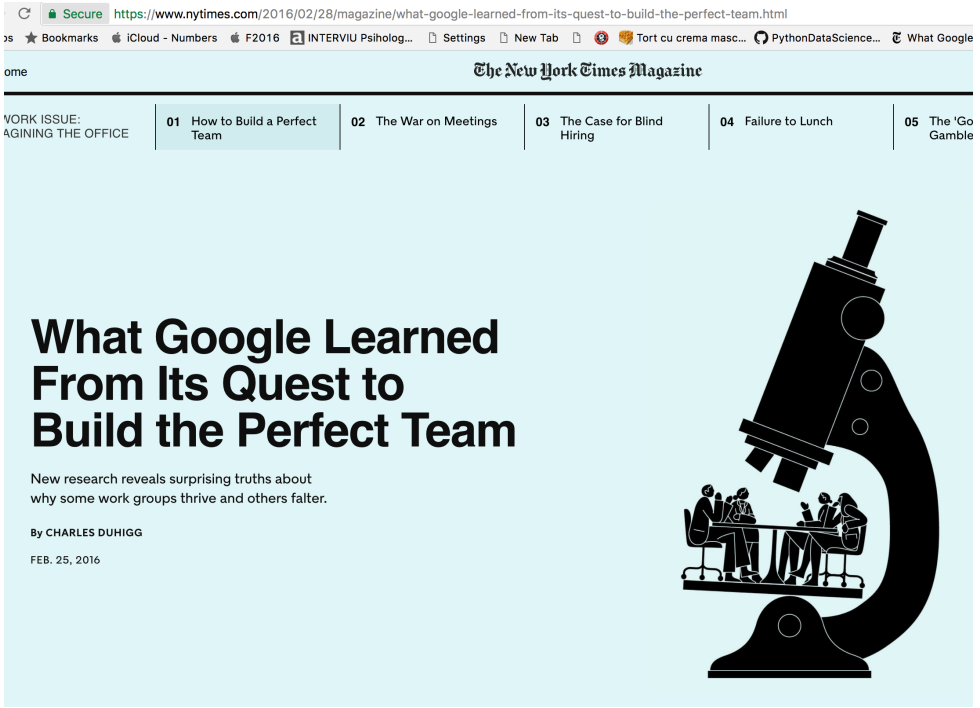
```
maat -l hib_evo.log -c git -a soc
```

```
entity,soc
hibernate-core/src/test/java/org/hibernate/test/legacy/FooBarTest.java,31204
hibernate-core/src/test/java/org/hibernate/test/filter/DynamicFilterTest.java,31000
hibernate-core/src/test/java/org/hibernate/test/jpa/AbstractJPATest.java,30799

hibernate-core/src/test/java/org/hibernate/test/sql/check/ResultCheckStyleTest.java,27554
hibernate-core/src/test/java/org/hibernate/test/orphan/one2one/pk/unidirectional/DeleteOneToOneOrphansTest.java,27554
hibernate-core/src/test/java/org/hibernate/test/optlock/OptimisticLockTest.java,27554

hibernate-core/src/test/java/org/hibernate/test/collection/dereferenced/One.java,19
hibernate-core/src/test/java/org/hibernate/test/annotations/onetoone/hhh9798/OneToOneJoinTableTest.java,19
hibernate-core/src/main/java/org/hibernate/metamodel/source/annotations/AttributeColumnRelationalState.java,19
```

3. Master the Social Aspects of Code

Conway's Law

Any organization that designs a system will produce a design whose structure is a copy of the organization's communication structure.



Social Biases

Process loss = losses in coordination and motivation

Pluralistic ignorance = everyone privately rejects a norm but think everyone else supports it! (Ask questions, Talk to people, Support decisions with data)

Commit Cloud (hibernate)

```
git log –pretty=format:'%s'
```

https://www.jasondavies.com/wordcloud/

Diffusion of responsibility = Who is responsible?

Increased group size

more quality problems are left unattended

Identify Ownership

add info regarding temporal coupling

```
maat -c git -l hib_evo.log -a main-dev
```

```
entity,main-dev,added,total-added,ownership
annotations/src/main/docbook/en/modules/entity.xml,Hardy Ferentschik,4094,6373,0.64
annotations/src/main/docbook/en/modules/setup.xml,Hardy Ferentschik,251,453,0.55

annotations/doc/reference/support/docbook-dtd/calstblx.dtd,Steve Ebersole,205,205,1.0
annotations/doc/reference/support/docbook-dtd/catalog.xml,Steve Ebersole,115,115,1.0

working-5.0-migration-guide.md,Steve Ebersole,103,105,0.98
wrapper/gradle-wrapper.jar,Steve Ebersole,0,0,0.0
wrapper/gradle-wrapper.properties,Steve Ebersole,9,10,0.9
```

Owner has the most added lines.

Shared Modules

```
maat -c git -l hib_evo.log -a authors
```

```
entity,n-authors,n-revs
hibernate-core/src/main/java/org/hibernate/persister/entity/AbstractEntityPersister.java,23,137
hibernate-core/src/main/java/org/hibernate/dialect/Dialect.java,18,86
hibernate-core/src/main/java/org/hibernate/cfg/annotations/CollectionBinder.java,18,58

hibernate-ehcache/src/main/java/org/hibernate/cache/EhCacheRegionFactory.java,4,4
hibernate-core/src/test/java/org/hibernate/test/annotations/id/sequences/HibernateSequenceTest.java,4,4
hibernate-core/src/test/java/org/hibernate/test/cache/User.java,4,4
hibernate-core/src/test/java/org/hibernate/test/sql/refcursor/NumValue.java,4,4
```