

Stats with Code Maat

Code maat

Code Maat is a command line tool used to mine and analyze data from version-control systems (VCS).

web site:

<https://github.com/adamtornhill/code-maat>

Code maat

statistic,value

number-of-commits,400

number-of-entities,2987

number-of-entities-changed,4471

number-of-authors,31

entity,n-authors,n-revs

AbstractEntityPersister.java,6,12

Loader.java,6,7

libraries.gradle,5,13

hibernate-core.gradle,5,11

build.gradle,4,33

Configuration.java,4,22

AnnotationBinder.java,4,13

AvailableSettings.java,4,9

Dialect.java,4,9

InfinispanRegionFactory.java,4,6

QueryBinder.java,4,5

CollectionBinder.java,4,5

...

date,added,deleted,commits

2012-03-08,693,0,1

2012-05-09,61,4,3

2012-10-16,1236,49,1

2012-10-19,551,139,1

2012-10-23,271,47,1

2012-10-25,1373,339,1

2012-12-13,18,0,1

2012-12-14,25,22,1

2012-12-17,145,54,4

2012-12-30,1,0,1

2012-12-31,40,1,1

2013-01-03,243,28,3

...

Generating input data

```
git log --pretty=format:'[%h] %an %ad %s' --date=short  
--numstat --no-merges --no-renames  
--after=YYYY-MM-DD --before=YYYY-MM-DD
```

example:

```
git log --pretty=format:'[%h] %an %ad %s' --date=short  
--numstat --no-merges --no-renames  
--after=2018-01-01 --before=2018-07-01 > data/evo.log
```

Running Code Maat

```
java -jar code-maat-[version].jar -h
```

This is Code Maat, a program used to collect statistics from a VCS.

Version: 1.0-SNAPSHOT

Usage: program-name -l log-file [options]

Options:

-l, --log LOG		Log file with input data
-c, --version-control VCS		Input vcs module type: supports svn, git, git2, hg, p4, or tfs
-a, --analysis ANALYSIS		authors The analysis to run (abs-churn, age, author-churn, authors, communication, coupling, entity-churn, entity-effort, entity-ownership, fragmentation, identity, main-dev, main-dev-by-revs, messages, refactoring-main-dev, revisions, soc, summary)
--input-encoding INPUT-ENCODING		Specify an encoding other than UTF-8 for the log file
-r, --rows ROWS		Max rows in output
-o, --outfile OUTFILE		Write the result to the given file name
-g, --group GROUP		A file with a pre-defined set of layers. The data will be aggregated according to the group of layers.
-p, --team-map-file TEAM-MAP-FILE		A CSV file with author,team that translates individuals into teams.
-n, --min-revs MIN-REVS	5	Minimum number of revisions to include an entity in the analysis
-m, --min-shared-revs MIN-SHARED-REVS	5	Minimum number of shared revisions to include an entity in the analysis
-i, --min-coupling MIN-COUPLING	30	Minimum degree of coupling (in percentage) to consider
-x, --max-coupling MAX-COUPLING	100	Maximum degree of coupling (in percentage) to consider
-s, --max-changeset-size MAX-CHANGESET-SIZE	30	Maximum number of modules in a change set if it shall be included in a coupling analysis
-e, --expression-to-match MATCH-EXPRESSION		A regex to match against commit messages. Used with -messages analyse
s		Instructs Code Maat to consider all commits during the same day as a single, logical commit
-t, --temporal-period TEMPORAL-PERIOD		Specify a date as YYYY-MM-dd that counts as time zero when doing a co
de age analysis		
-h, --help		

Please refer to the manual page for more information.

Code Maat Command

Generating a summary

```
java -jar code-maat-[version].jar -l evo.log -c git -a summary
```

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l evo.log -c git -a summary
```

Mining organizational metrics

```
java -jar code-maat-[version].jar -l evo.log -c git
```

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l evo.log -c git
```

```
java -jar code-maat-[version].jar -l evo.log -c git -a revisions
```

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l evo.log -c git -a revisions
```

Code Maat Command

Calculate code age

```
java -jar code-maat-[version].jar -l evo.log -c git -a age
```

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l evo.log -c git -a age
```

Mining logical coupling

```
java -jar code-maat-[version].jar -l evo.log -c git -a coupling
```

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l evo.log -c git -a coupling
```

Code Maat Command

Code churn measures

absolute churn

```
java -jar code-maat-[version].jar -l evo.log -c git -a abs-churn
```

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l evo.log -c git -a abs-churn
```

churn by author

```
java -jar code-maat-[version].jar -l evo.log -c git -a author-churn
```

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l evo.log -c git -a author-churn
```

churn by entity

```
java -jar code-maat-[version].jar -l evo.log -c git -a entity-churn
```

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l evo.log -c git -a entity-churn
```

Code Maat Command

Ownership patterns

entity ownership

```
java -jar code-maat-[version].jar -l evo.log -c git -a entity-ownership
```

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l evo.log -c git -a abs-churn
```

entity effort

```
java -jar code-maat-[version].jar -l evo.log -c git -a entity-effort
```

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l evo.log -c git -a author-churn
```

main developer

```
java -jar code-maat-[version].jar -l evo.log -c git -a main-dev
```

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l evo.log -c git -a main-dev
```

Workshop

Clone Code Maat

```
git clone git@github.com:adamtornhill/code-maat.git
```

Change Directory

```
cd code-maat
```

Create Directory data

```
mkdir data
```

Generating input data

```
git log --pretty=format:'[%h] %an %ad %s' --date=short --numstat --no-merges --no-renames  
--after=2017-01-01 > data/evo.log
```

Copy code-maat.jar to code-maat directory

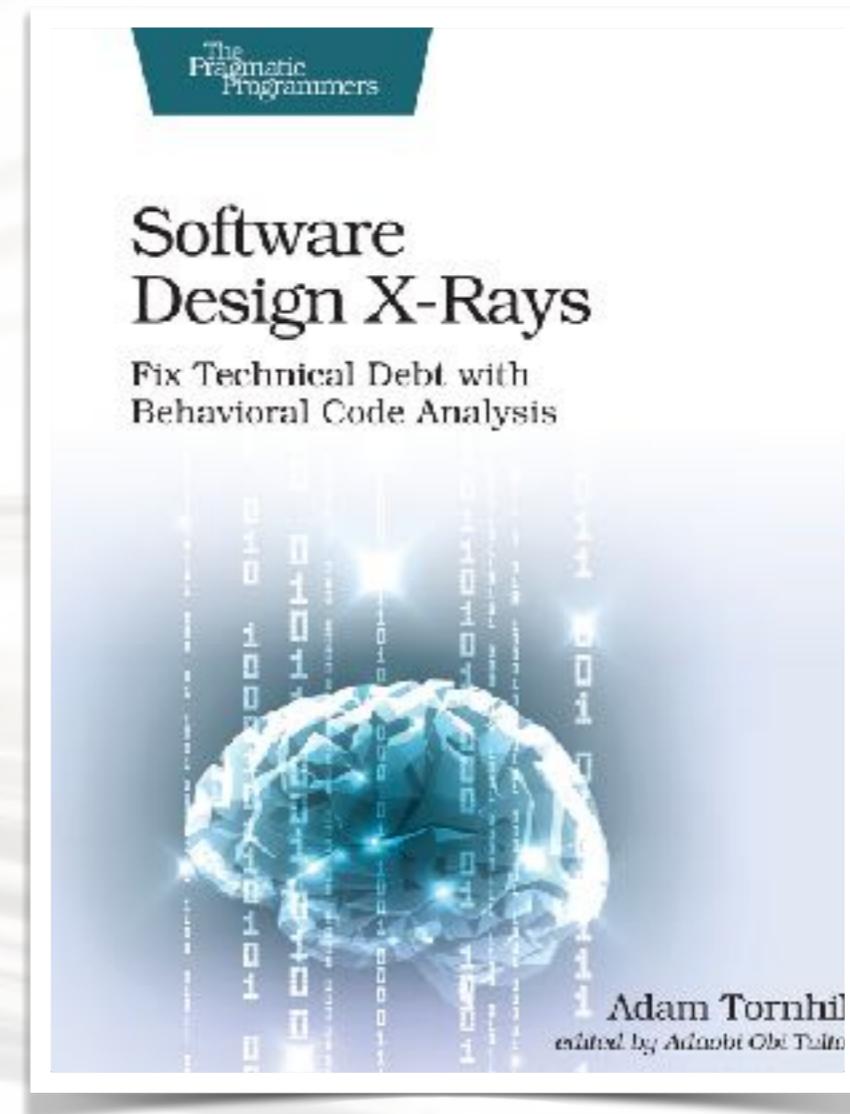
Mining with code maat

```
java -jar code-maat-1.1-SNAPSHOT-standalone.jar -l data/evo.log -c git
```

Books to Read and Practice



[your-code-as-a-crime-scene](#)



[software-design-x-rays](#)