r7k

Lessons from Closure

Michael Feathers
R7K Research & Conveyance


Clojure


Clojure

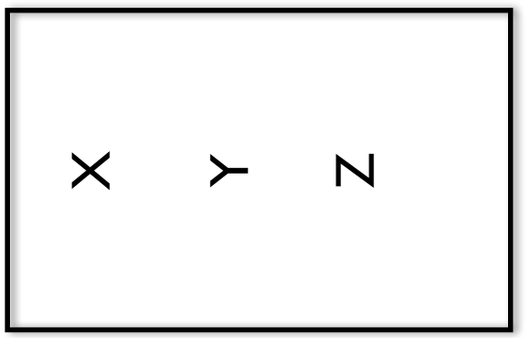
Open/Closed Principle

SOFTWARE ENTITIES (CLASSES, MODULES, FUNCTIONS, ETC.)
SHOULD BE OPEN FOR EXTENSION, BUT CLOSED FOR MODIFICATION.

Prescription or Description?

A codebase is well designed when we don’t have to change it much to add new features

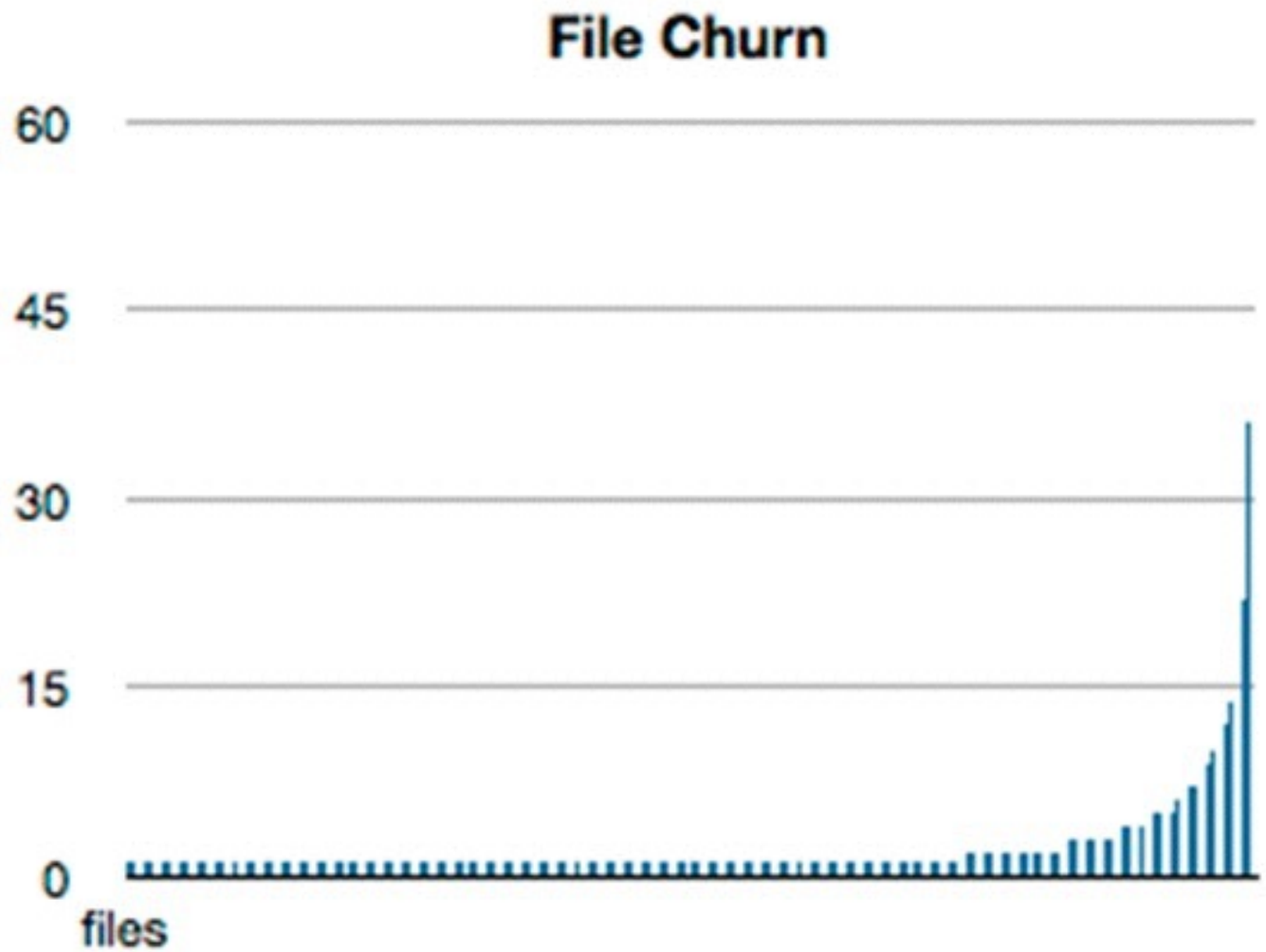
The Open/Closed Principle and Single Responsibility Principle are related


X
Y
Z

X

Y

Z


File Churn
60
45
30
15
0
files

Git

Git

Commit

Git

Commit

commit hash (sha1)
time/date stamp
committer
files
actual change

Git

Method Event

commit hash (sha1)
time/date stamp
committer
method name
method body
add/change/delete

Commit

commit hash (sha1)
time/date stamp
committer
files
actual change

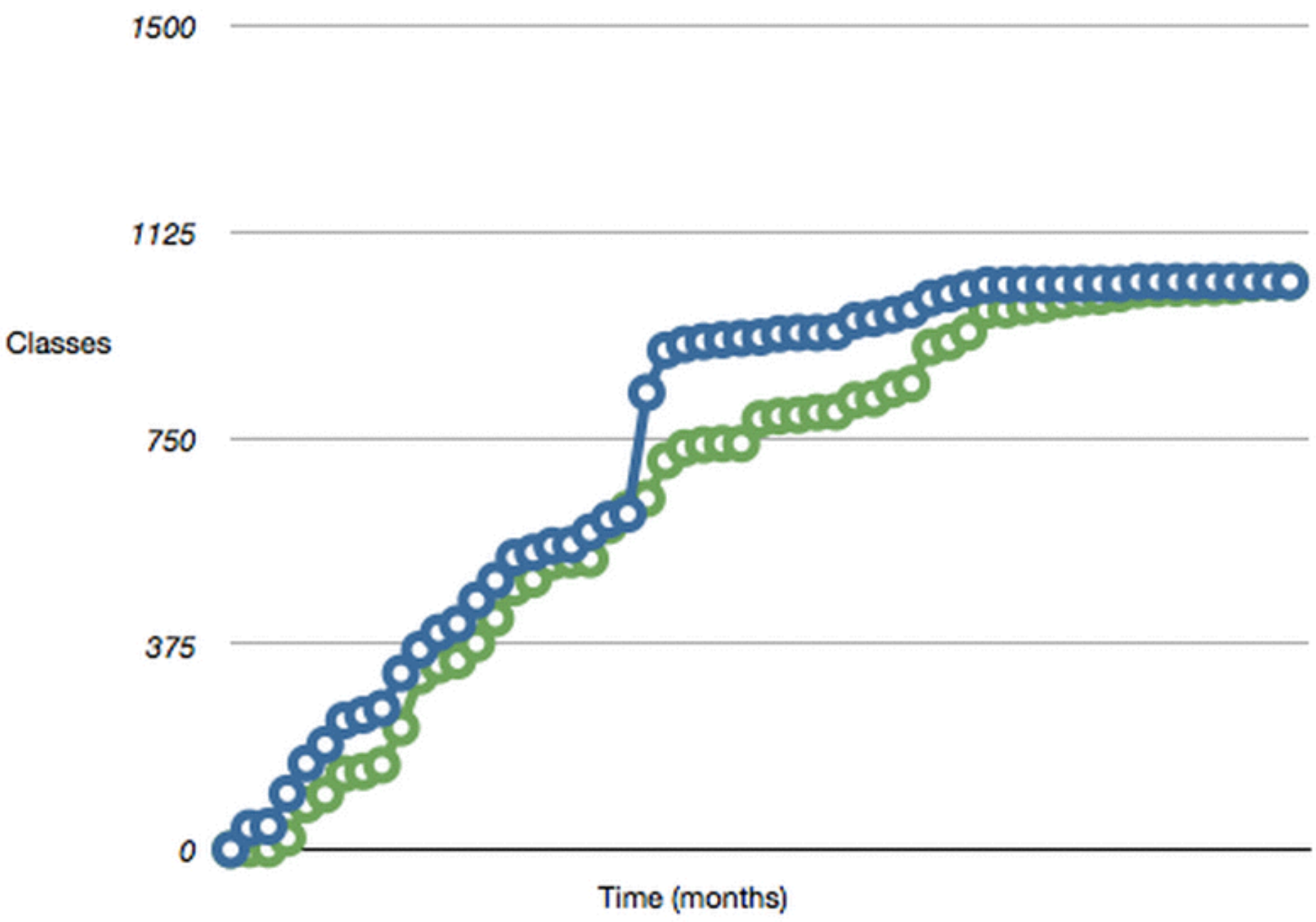
Classes By Closure Date

[["DummiesController", 2008-04-21 13:03:08 -0700],
["Core::ActiveRecord::AttributeDefaults::ClassMethods", 2008-04-22 16:02:54 -0700],
["Legacy::Database", 2008-04-24 15:37:51 -0700],
["Core::ActiveRecord::AttributeDelegation::ClassMethods", 2008-04-24 20:46:58 -0700],
["Core::ActiveRecord::SkipValidationForHasOnes", 2008-04-29 21:54:32 -0700]]

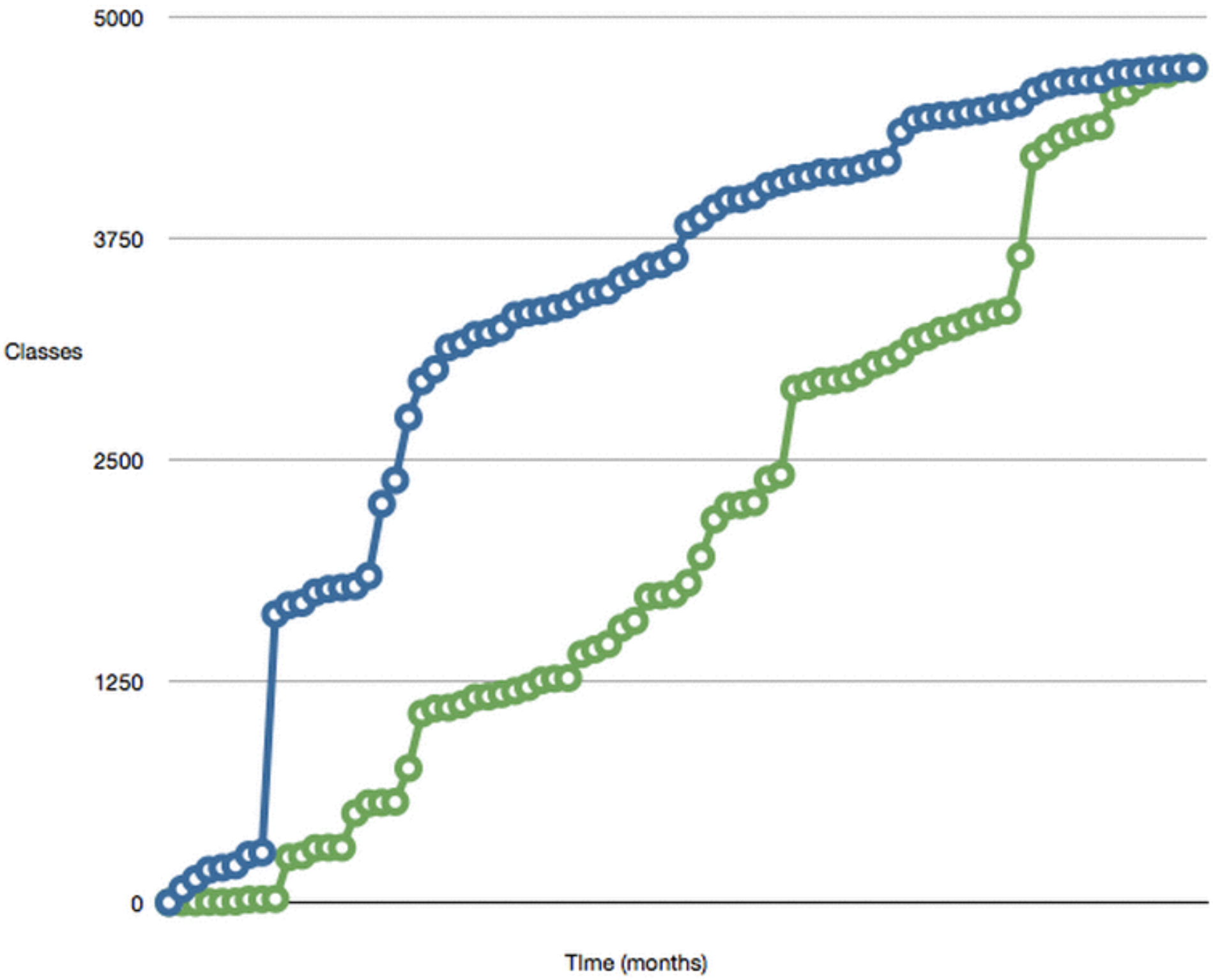
Classes By Closure Date

```
def classes_by_closure events
  class_names = method_events(events).map(&:class_name).uniq
  classes = Hash[class_names.zip([Time.now] * class_names.length)]
  method_events(events).each {|e| classes[e.class_name] = e.date }
  classes.to_a.sort_by {|_,date| date }
end
```

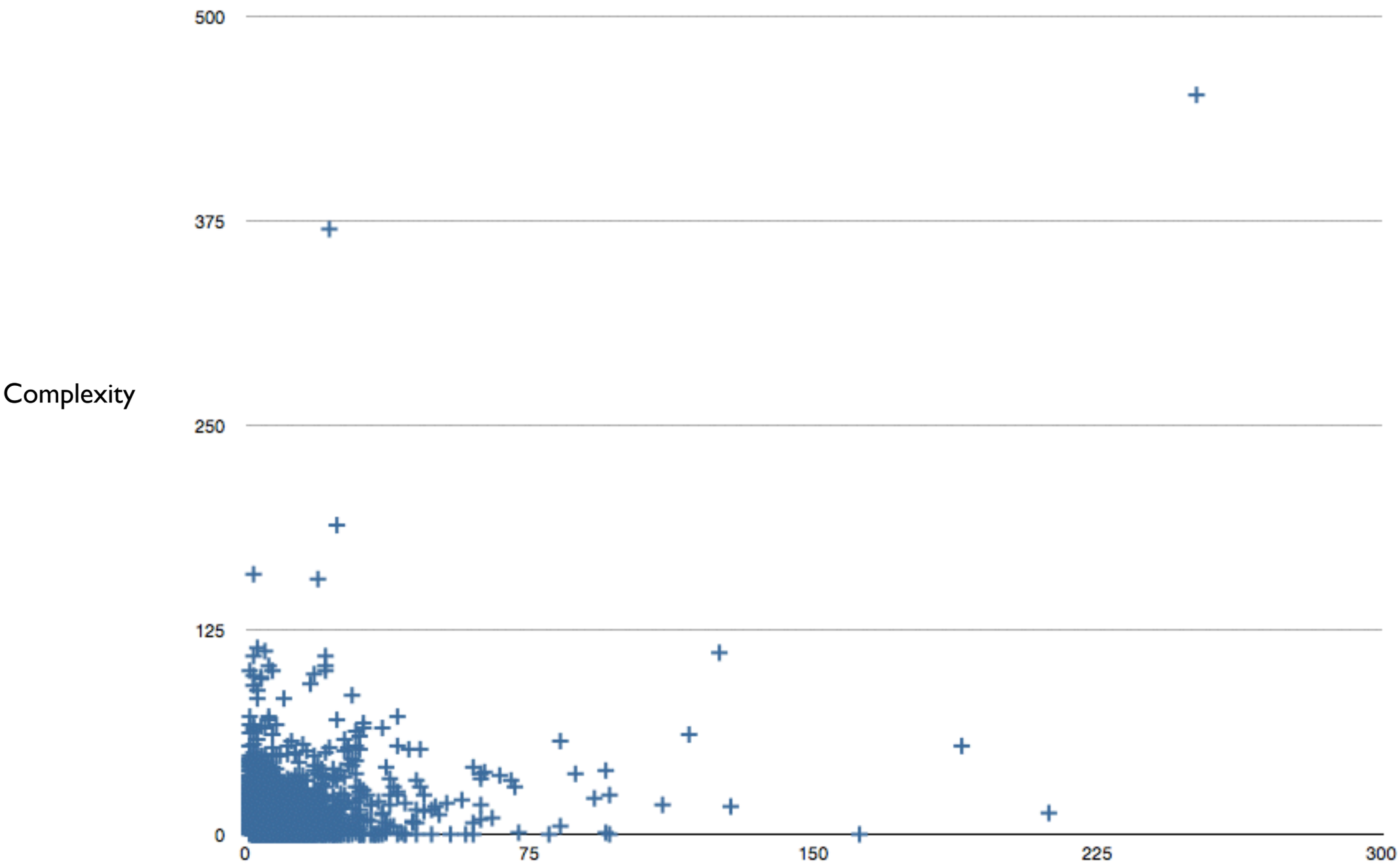
Active Set of Classes



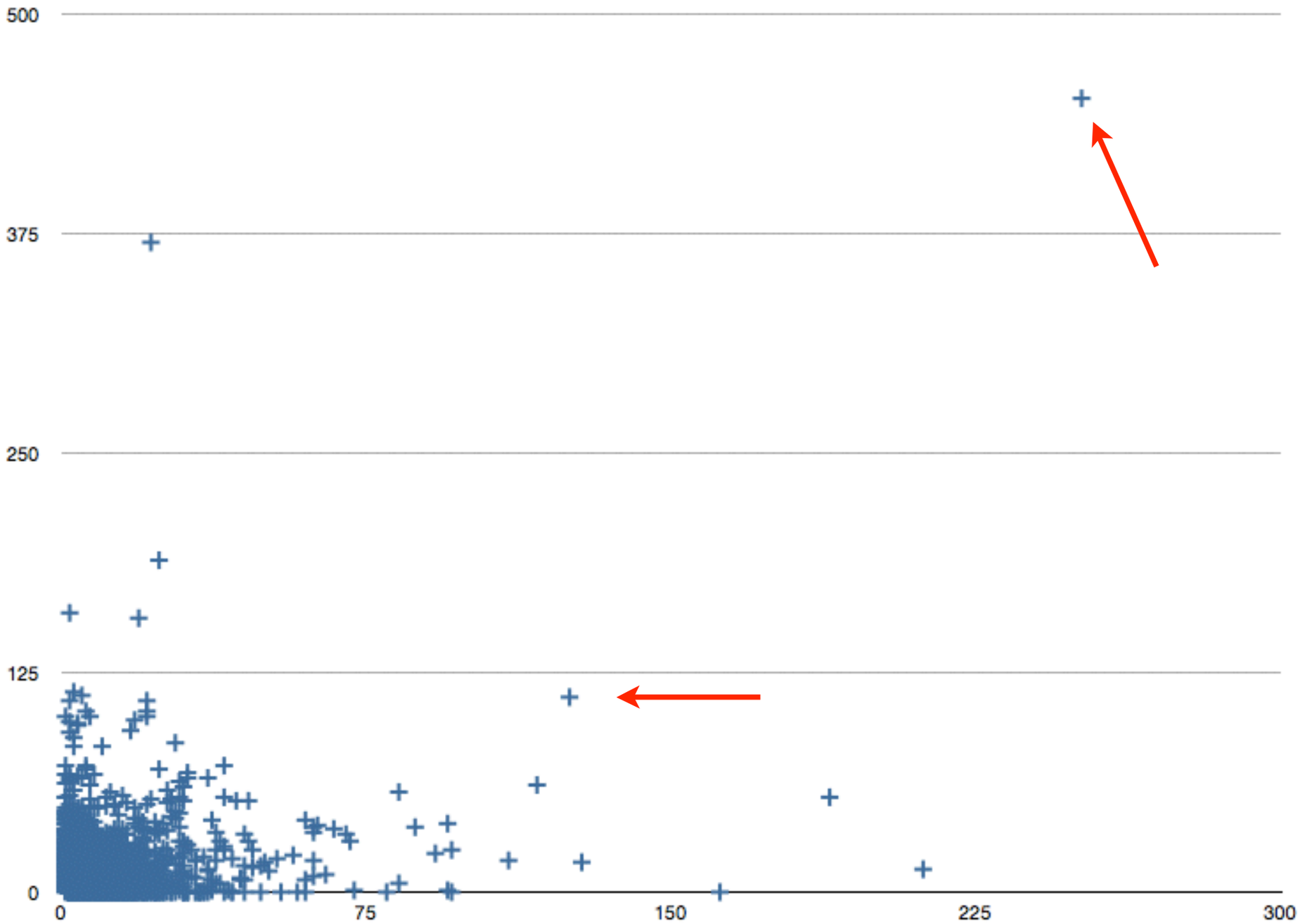
Active Set of Classes



Turbulence



Turbulence



The Big Challenge is Understanding What Causes "Openness"

Is Churn all we need?

2014-01-20

BTH-Blekinge Institute of Technology
Uppsats inlämnad som del av examination i
DV1446 Kandidatarbete i datavetenskap.

Kandidatuppsats

Where do you save most money on refactoring?

Susanne Siverland

Use of Relative Code Churn Measures to Predict System Defect Density

Nachiappan Nagappan*
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
nnagapp@ncsu.edu

Thomas Ball
Microsoft Research
Redmond, WA 98052
tball@microsoft.com

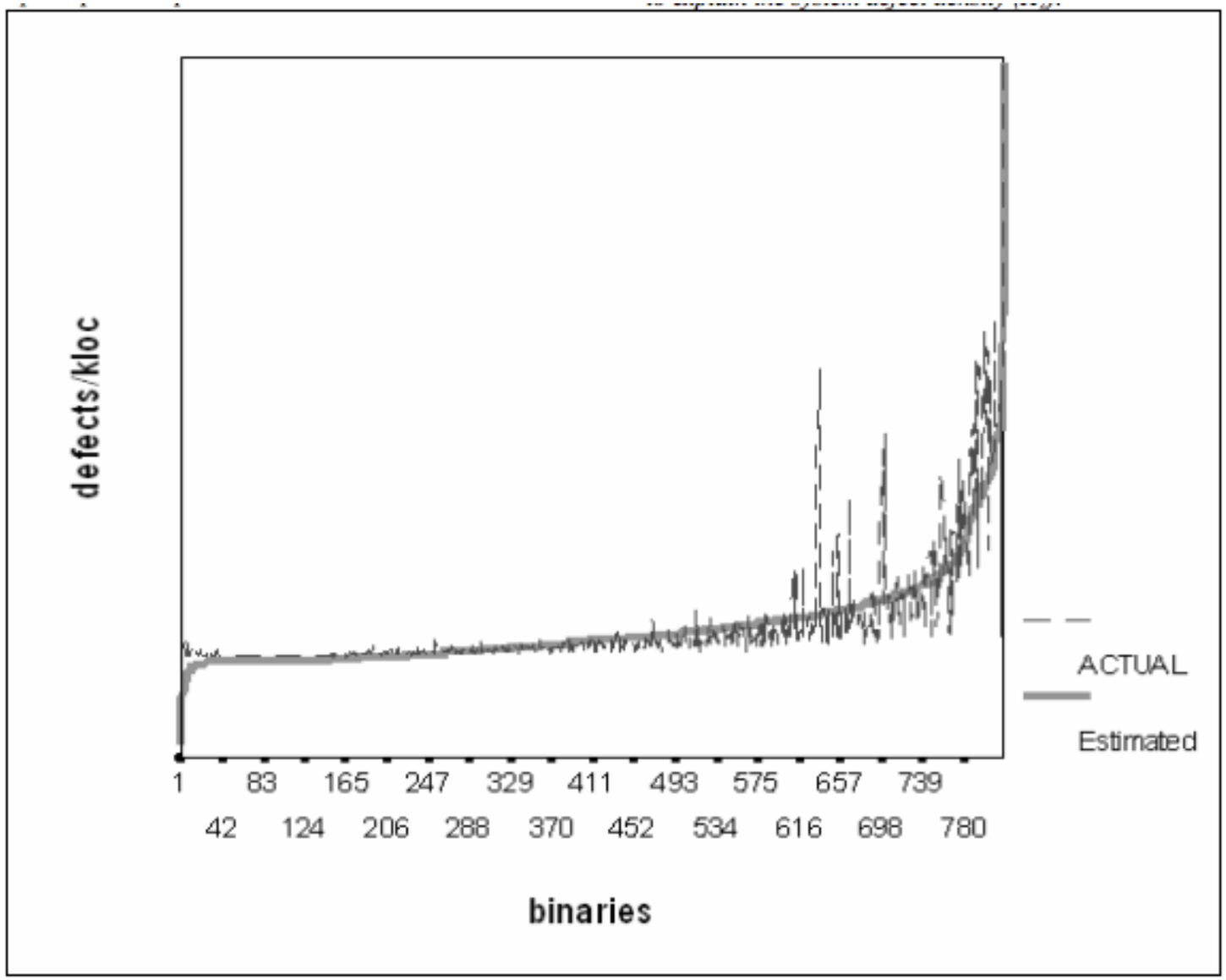
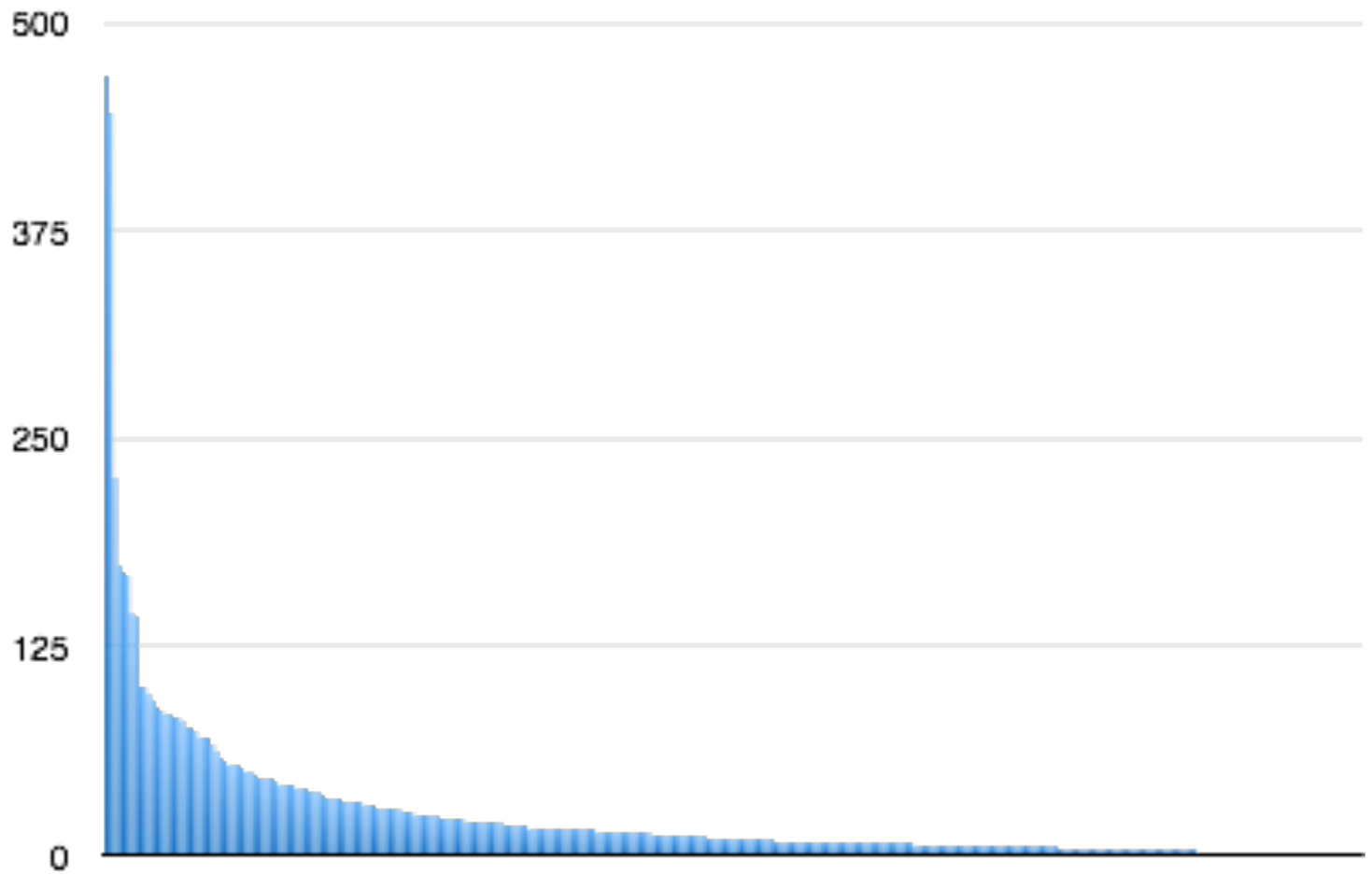
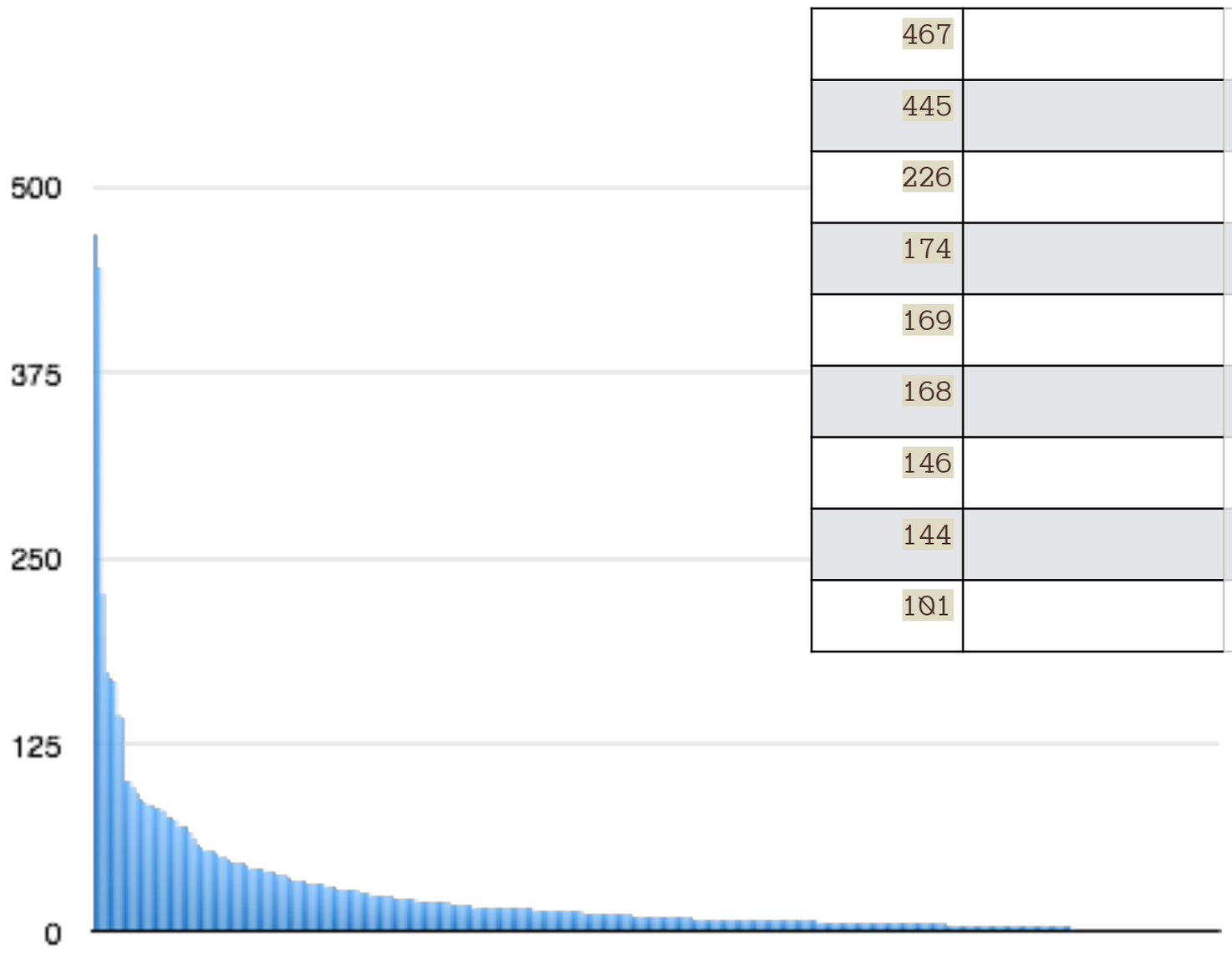


Figure 3: Actual vs. Estimated System Defect Density


500
375
250
125
0

How Effective Is Refactoring?



467		6.57283603096411
445		12.8360309641098
226		16.0168895144265
174		18.4658691062632
169		20.844475721323
168		23.2090077410274
146		25.2638986629134
144		27.2906403940887
101		28.7121745249824

The Big Challenge is Understanding What Causes "Openness"