Eyeballing the Cyclomatic Complexity Metric

Jurgen J. Vinju, Michael W. Godfrey

Centrum Wiskunde & Informatica, Amsterdam
David R. Cheriton School of Computer Science, Waterloo

Contribution

- "Control flow patterns" to analyze control flow
- Used to eyeball the CC metric
- Extremely simple
- Good for other research

Disclaimer: we introduce a metric to debunk another, not to introduce another metric

A “Control Flow Pattern” is an abstract syntax tree where all nodes that are not control flow statements are replaced by nothing. $\perp$

```
while (x >= 0) {
  if (x % 2 == 0)
    print("even");
  x--;
}
return 1;
```

```
while (⊥) {
  if (⊥)
    ⊥

}
return ⊥;
```

A "Compressed Control Flow Pattern" is a control flow pattern where all empty lists have been removed and consecutive repetition has been contracted (based on structural equality), in a fixed point computation.

```
switch(⊥) {
  case ⊥ : return ⊥;
  case ⊥ : return ⊥;
  case ⊥ : return ⊥;
  case ⊥ : return ⊥;
  case ⊥ : return ⊥;
  case ⊥ : return ⊥;
  case ⊥ : return ⊥;
  case ⊥ : return ⊥;
}
```

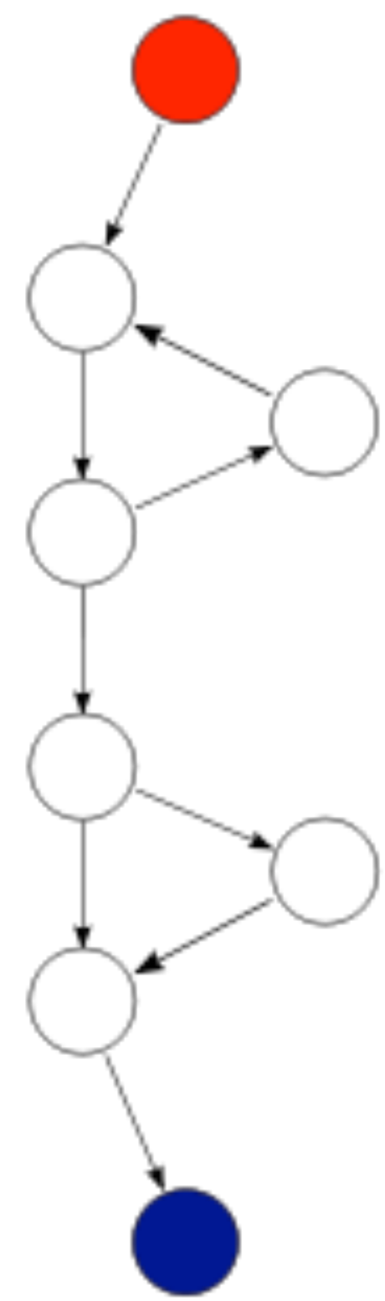
```
switch (⊥) {
  ℛ (case ⊥ : return ⊥;)
}
```

Control flow patterns matter

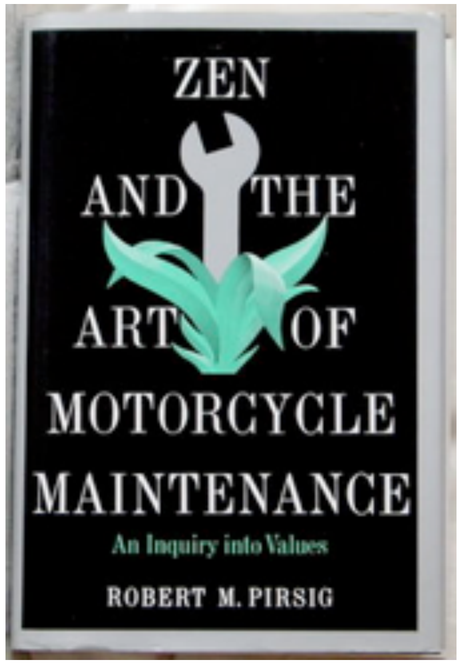
Project	#Meth	#Pat	$\#Pat^{comp}$	#Comp
compendium	7,736	1,271 (16%)	1,234 (15%)	455 (36%)
Tomcat70	16,018	2,211 (13%)	2,158 (13%)	931 (43%)
dsbudget	306	64 (20%)	64 (20%)	18 (28%)
xml-commons-external	3,346	91 (2%)	89 (2%)	30 (33%)
apache-ant	10,278	1,391 (13%)	1,349 (13%)	555 (41%)
bcel	3,076	286 (9%)	268 (8%)	120 (44%)
hsqldb	5,326	1,013 (19%)	969 (18%)	438 (45%)
smallsql	2,556	353 (13%)	332 (12%)	158 (47%)
Merged	48,642	5,633 (11%)	5,434 (11%)	2,455 (43%)

Cyclomatic Complexity

- Is defined on control flow graphs
- Measures the number of linear independent paths
- Estimates #tests to cover code
- CC = #conditions + 1



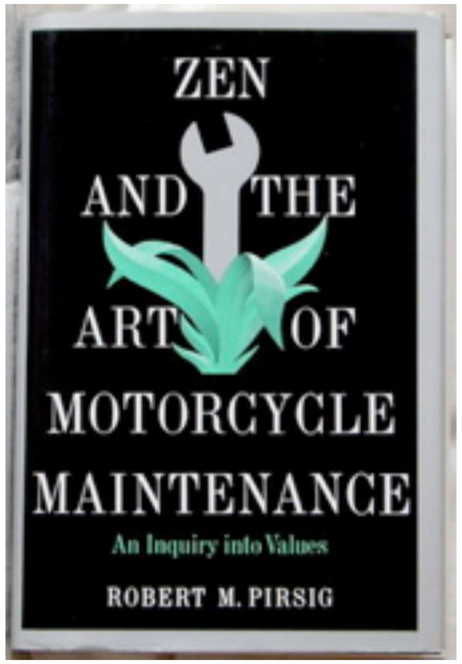
[wikipedia]



Measuring Understandability

Cyclomatic Complexity

Correlation with Badness

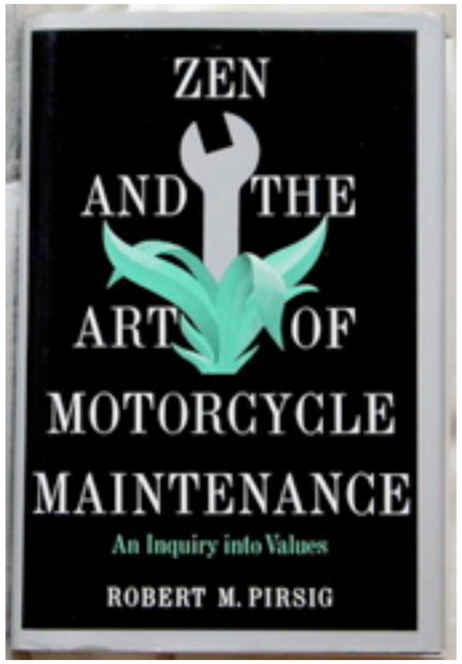


Measuring Understandabi

gimme!

Cyclomatic Complexity

Correlation with Badness



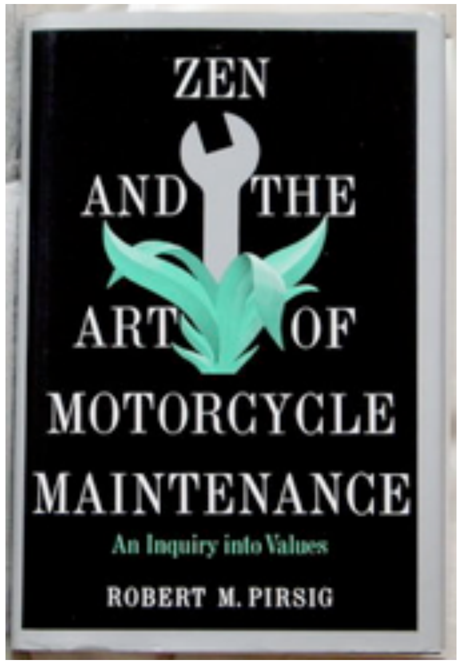
Measuring Understandabil

gimme!

Cyclomatic Complexity

easy!

Correlation with Badness



Measuring Understandabil

gimme!

Cyclomatic Complexity

easy!

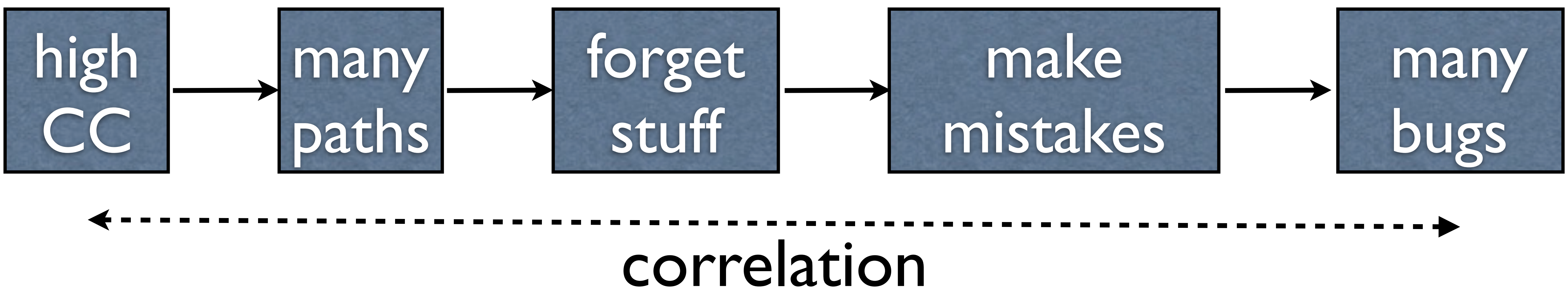
great!

Correlation with Badness

The Beef: Causality

- Does high CC cause bugs?
- Is high CC bad? Low CC good? Thresholds?
- What about CC-bug correlation in 5-10 years?
- Does CC work “in the small”, method by method?

Right?


high CC
many paths
forget stuff
make mistakes
many bugs
correlation

Not the whole story

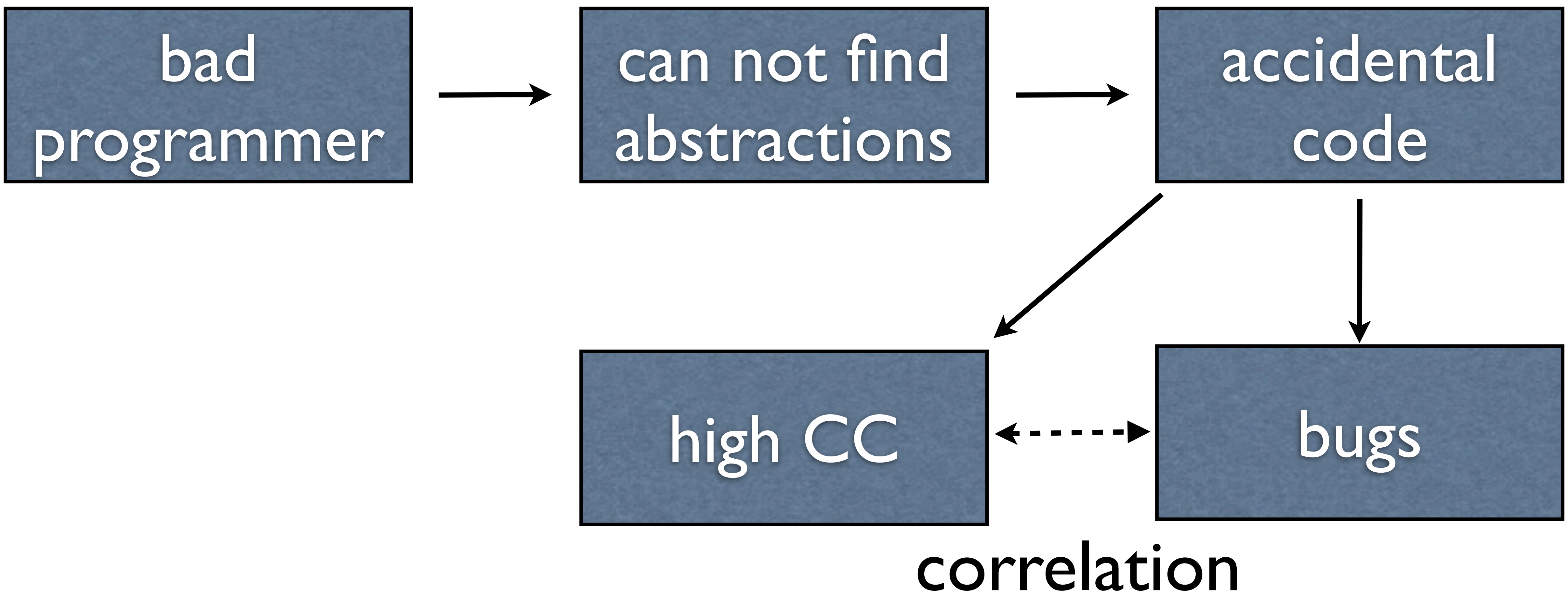
```
  i = 0;
  goto body;
loop:
  if (i == 10)
    goto done;
  i++;
body:
  print(i);
  goto loop;
done:
```

```
i = 0;
do
  print(i);
while (i++ != 10);
```

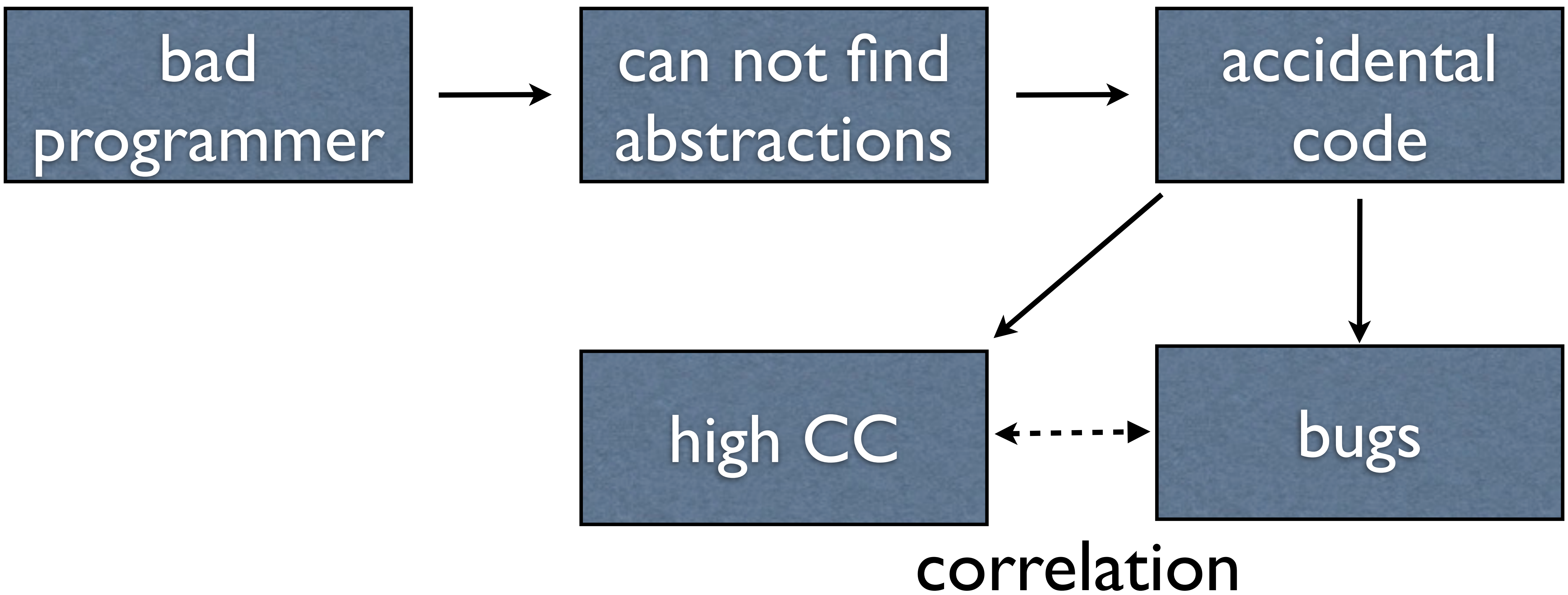
More than bargained for

```
switch(state) {
  case A: return aap();
  case N: return noot();
  case M: return mies();
  case V: return vuur();
  case G: return gijs();
  case H: return hok();
  case D: return does();
  case S: return schaap();
  // and 50 more
}
```

Alternative story



Alternative story



DUH!

To say high CC causes bugs is not like saying high cholesterol causes cardiovascular disease.

It's like saying that driving a BMW causes more visits to personal therapists.

Question

Does all that matter in real software?

That the CC metric both under and over appreciates the complexity of control flow?

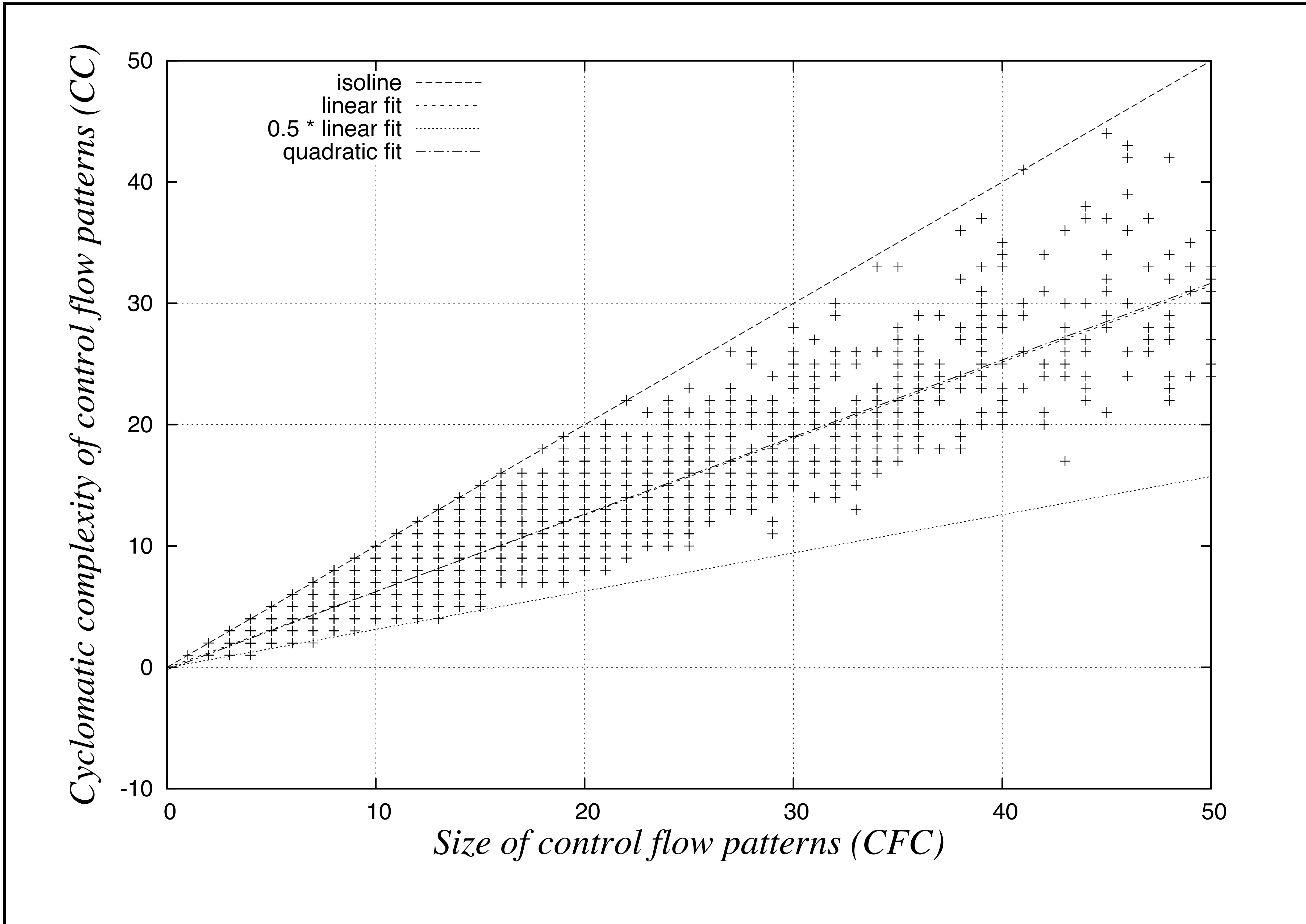
Eyeballing CC

We assume that to appreciate the full complexity of the control flow of a method, you need to know all of it: all nodes and all edges.

Does CC predict the size of the full control flow of any given method or not?

Control flow patterns to correct for skew caused by frequently used programming idioms

Can CC predict size of control flow?



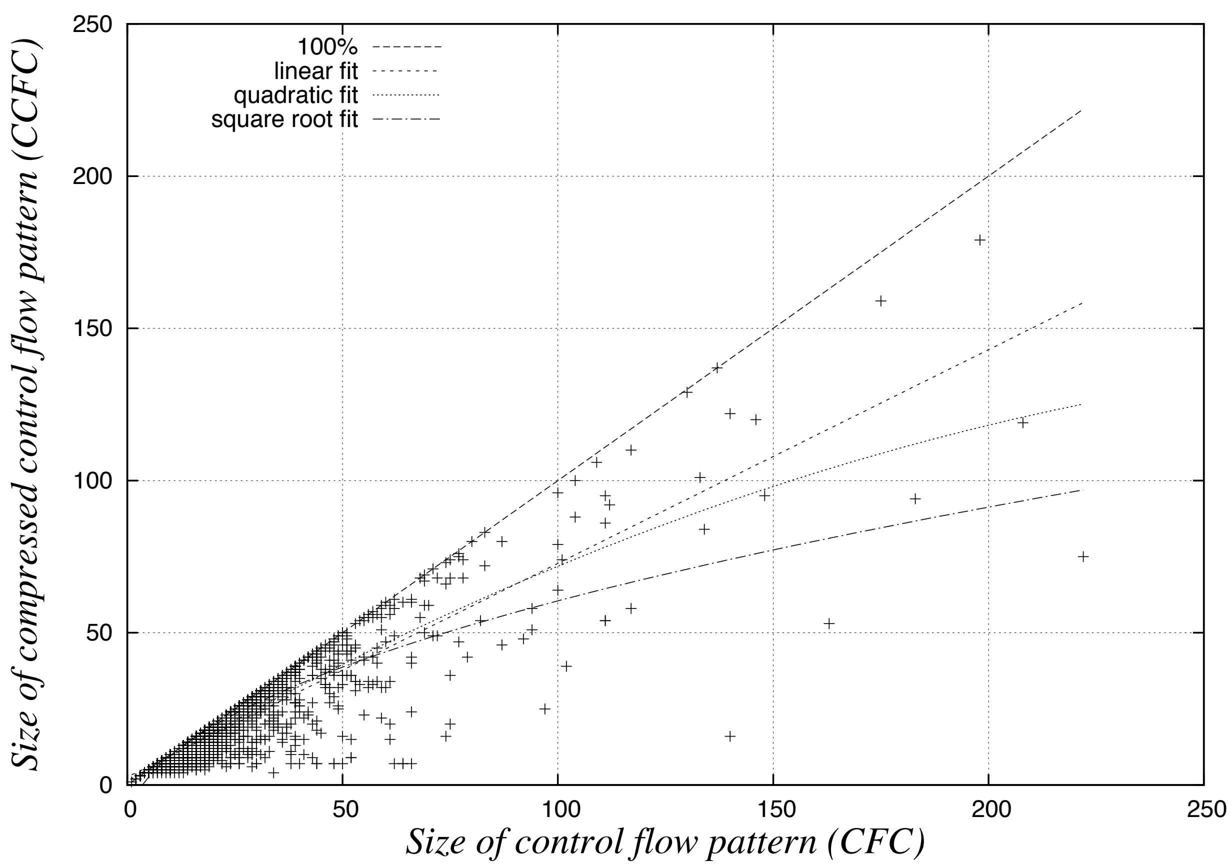
- CC does not predict the size of a complete control flow graph (at all).

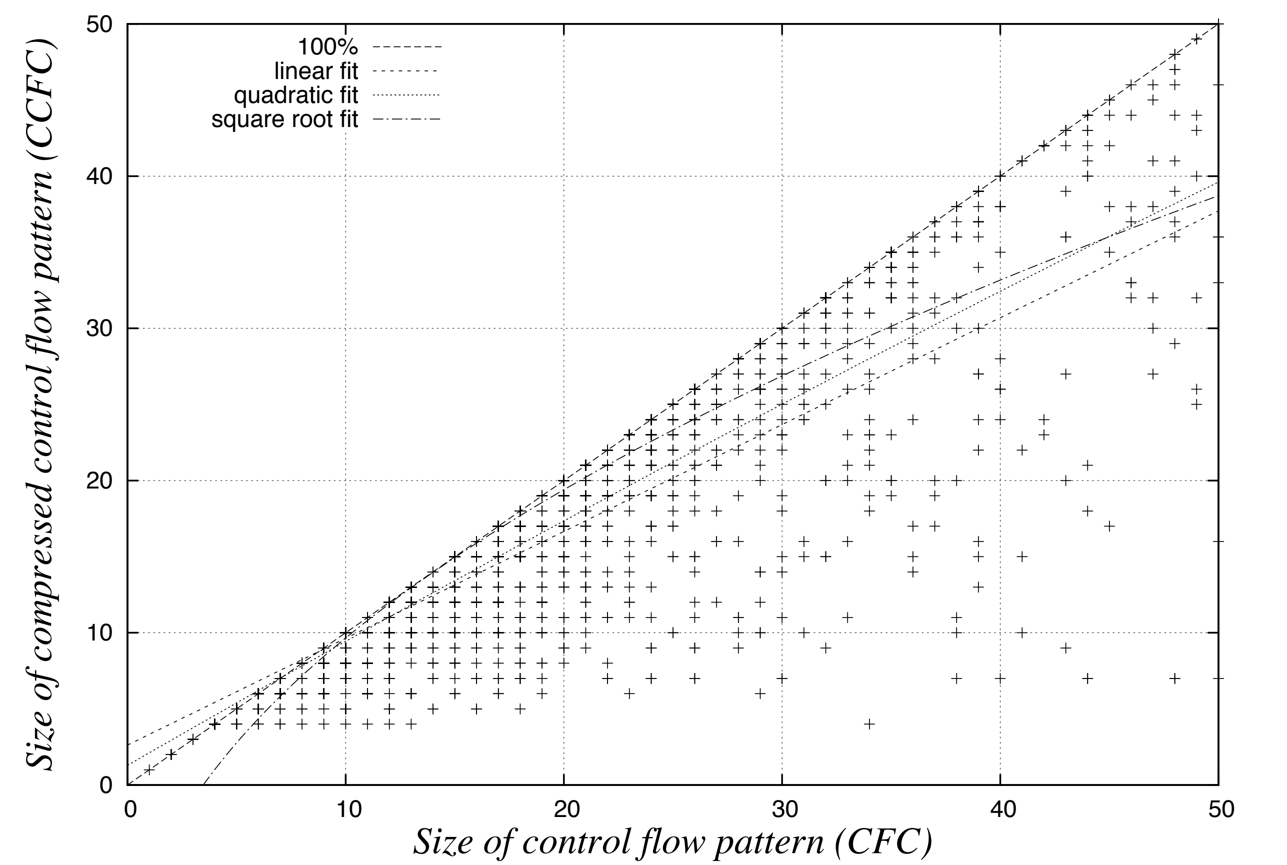
- CC does not predict the size of a complete control flow graph (at all).
- The published linear correlation between CC and method size is contentious.

- CC does not predict the size of a complete control flow graph (at all).
- The published linear correlation between CC and method size is contentious.
- CC practically always underrepresents the complexity of understanding a method.

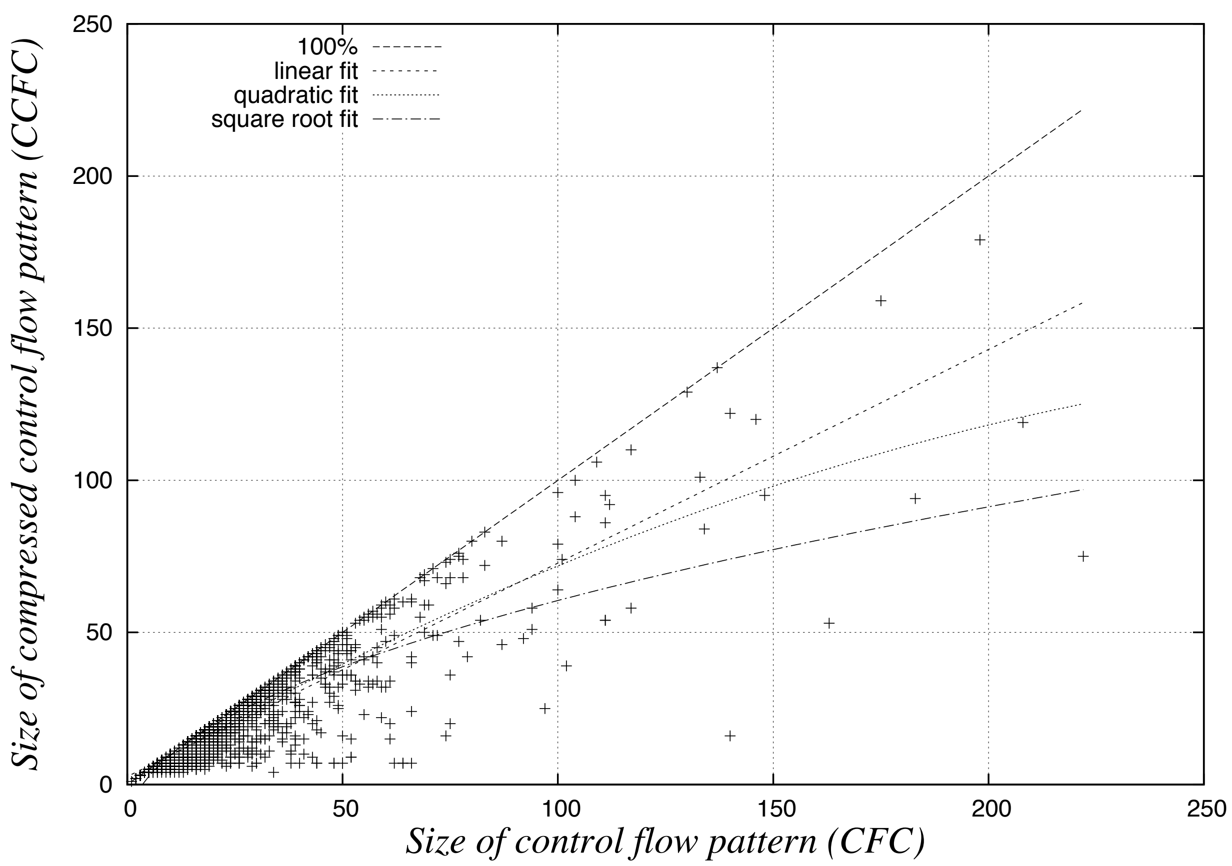
- CC does not predict the size of a complete control flow graph (at all).
- The published linear correlation between CC and method size is contentious.
- CC practically always underrepresents the complexity of understanding a method.
- Additional edges do matter in real Java

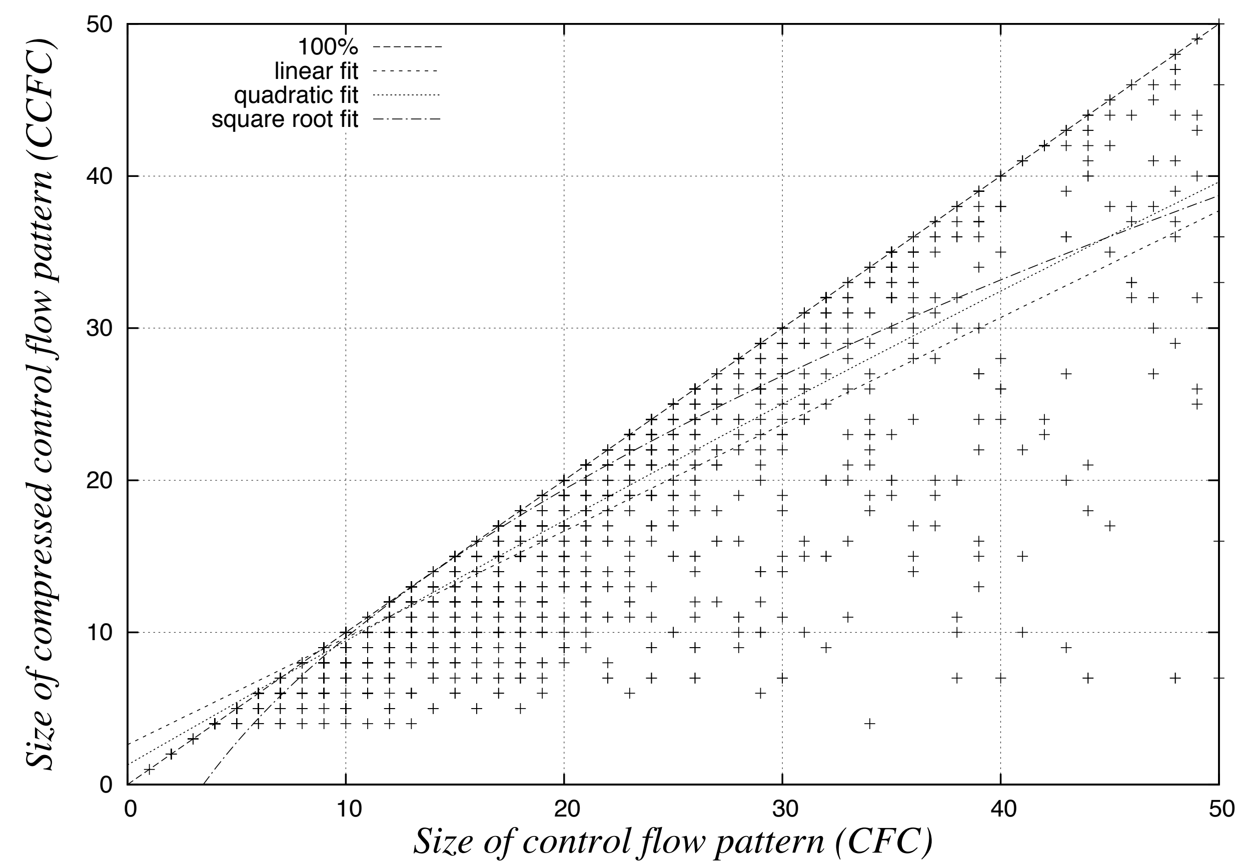
How many times do repeated patterns occur? How much repetition in patterns?





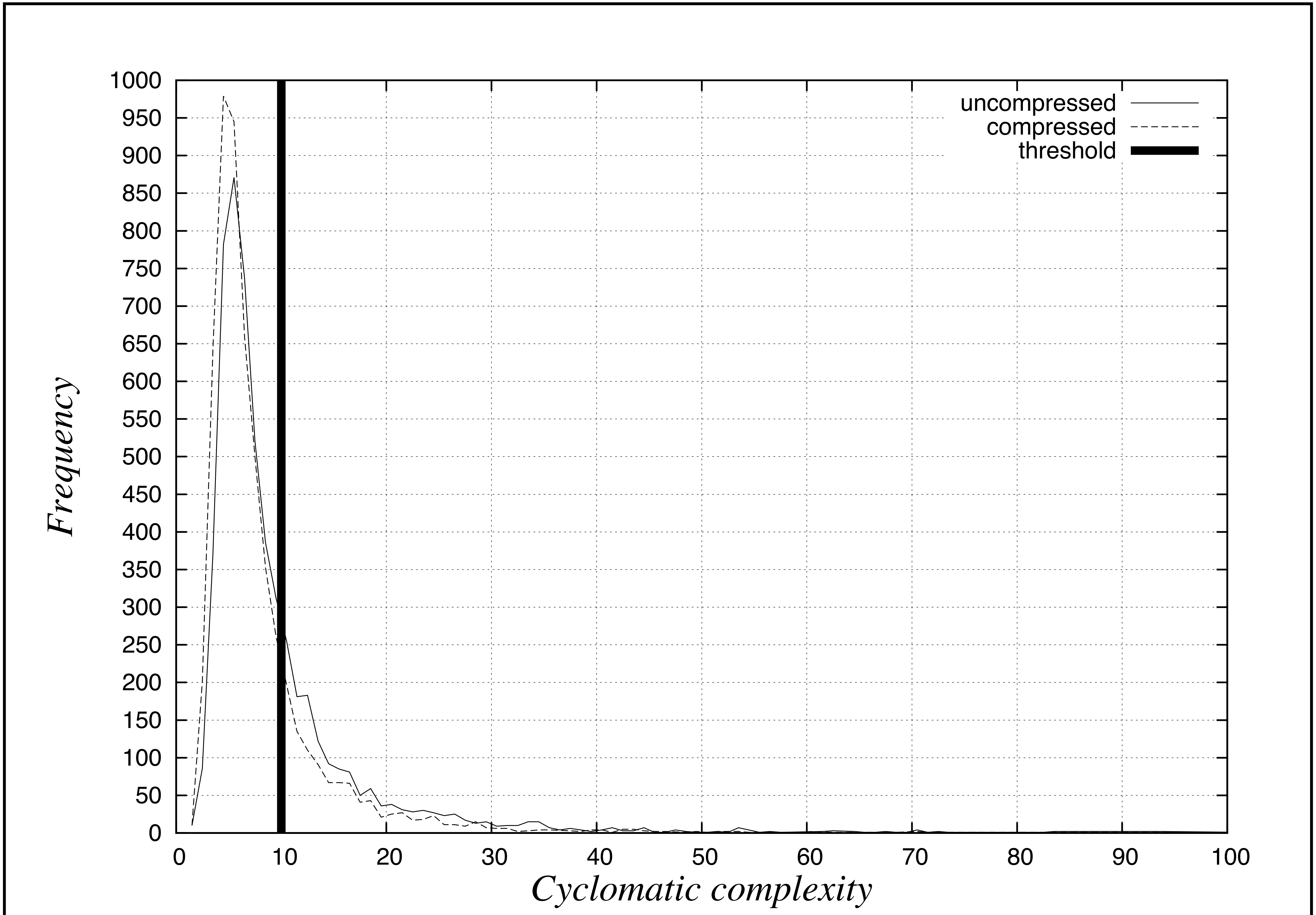
How many times do repeated patterns occur? How much repetition in patterns?



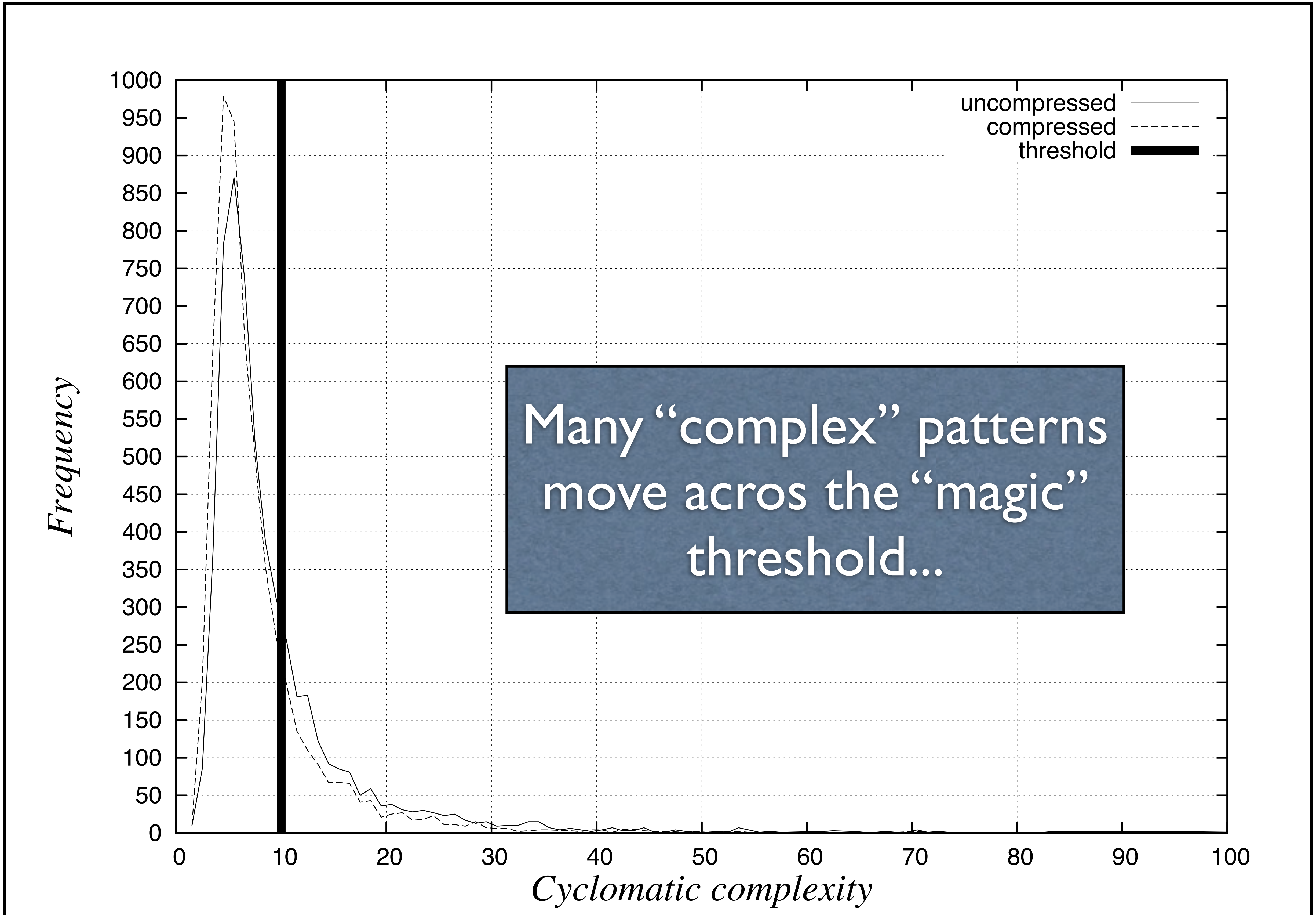


Quite a lot
and more aggressively for the larger patterns

How are the compressed patterns distributed?



How are the compressed patterns distributed?



http://www.rascal-mpl.org

```
public AstNode compress(AstNode body) = innermost visit(body) {
  case [*a, repeated([*n]), n, *b] => [*a, repeated([*n]), *b]
  case [*a, x, *c, x, c, *d] => [*a, repeated([x,*c]), *d]
  case block(repeated(n)) => repeated(n)
};
```

```
public AstNode compress(AstNode body) = innermost visit(body) {
  case [*a, repeated([*n]), n, *b] => [*a, repeated([*n]), *b]
  case [*a, x, *c, x, c, *d] => [*a, repeated([x,*c]), *d]
  case block(repeated(n)) => repeated(n)
};
```

Cyclomatic Complexity is not good for indicating understandability

http://www.rascal-mpl.org

```
public AstNode compress(AstNode body) = innermost visit(body) {
  case [*a, repeated([*n]), n, *b] => [*a, repeated([*n]), *b]
  case [*a, x, *c, x, c, *d] => [*a, repeated([x,*c]), *d]
  case block(repeated(n)) => repeated(n)
};
```

Cyclomatic Complexity is not good for indicating understandability

Control flow patterns are useful

```
public AstNode compress(AstNode body) = innermost visit(body) {
  case [*a, repeated([*n]), n, *b] => [*a, repeated([*n]), *b]
  case [*a, x, *c, x, c, *d] => [*a, repeated([x,*c]), *d]
  case block(repeated(n)) => repeated(n)
};
```

Cyclomatic Complexity is not good for indicating understandability

Control flow patterns are useful

Watch out with linear correlation

Industry is starting to believe some of our results. Now, this is a time to be very very careful, because our models will be used to judge the past and predict the future.

Just because you can fit a plot with a power function does not mean you have a scale-free distribution. You can fit it with anything.

Just because least squares produces a nice linear fit does not mean that you have a linear relation.

Always show the plots.