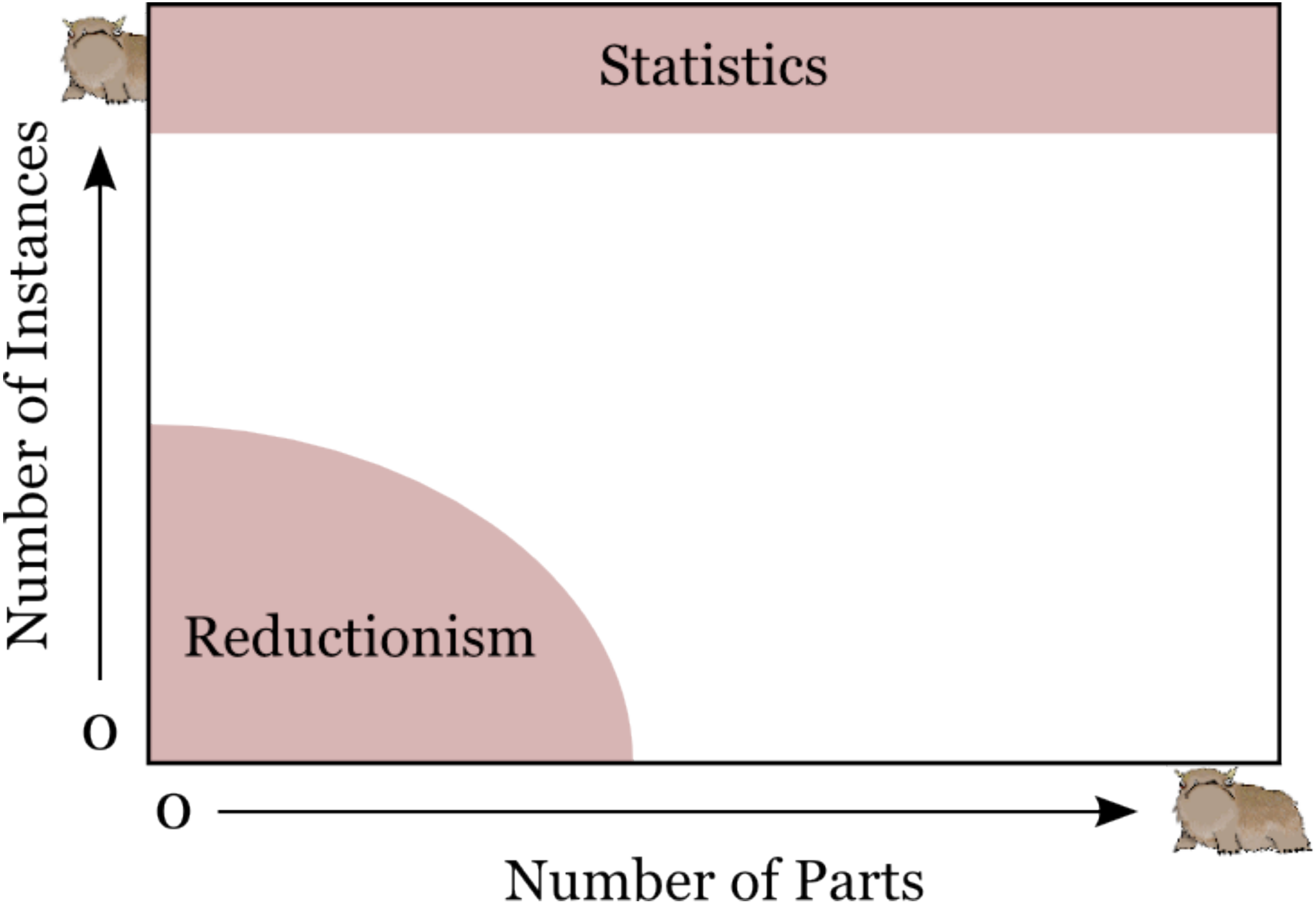
Complexity Multipliers

Jason Felice
@eraserhd

Act I: Models

How to Understand Things



Stolen from Gerald Weinberg

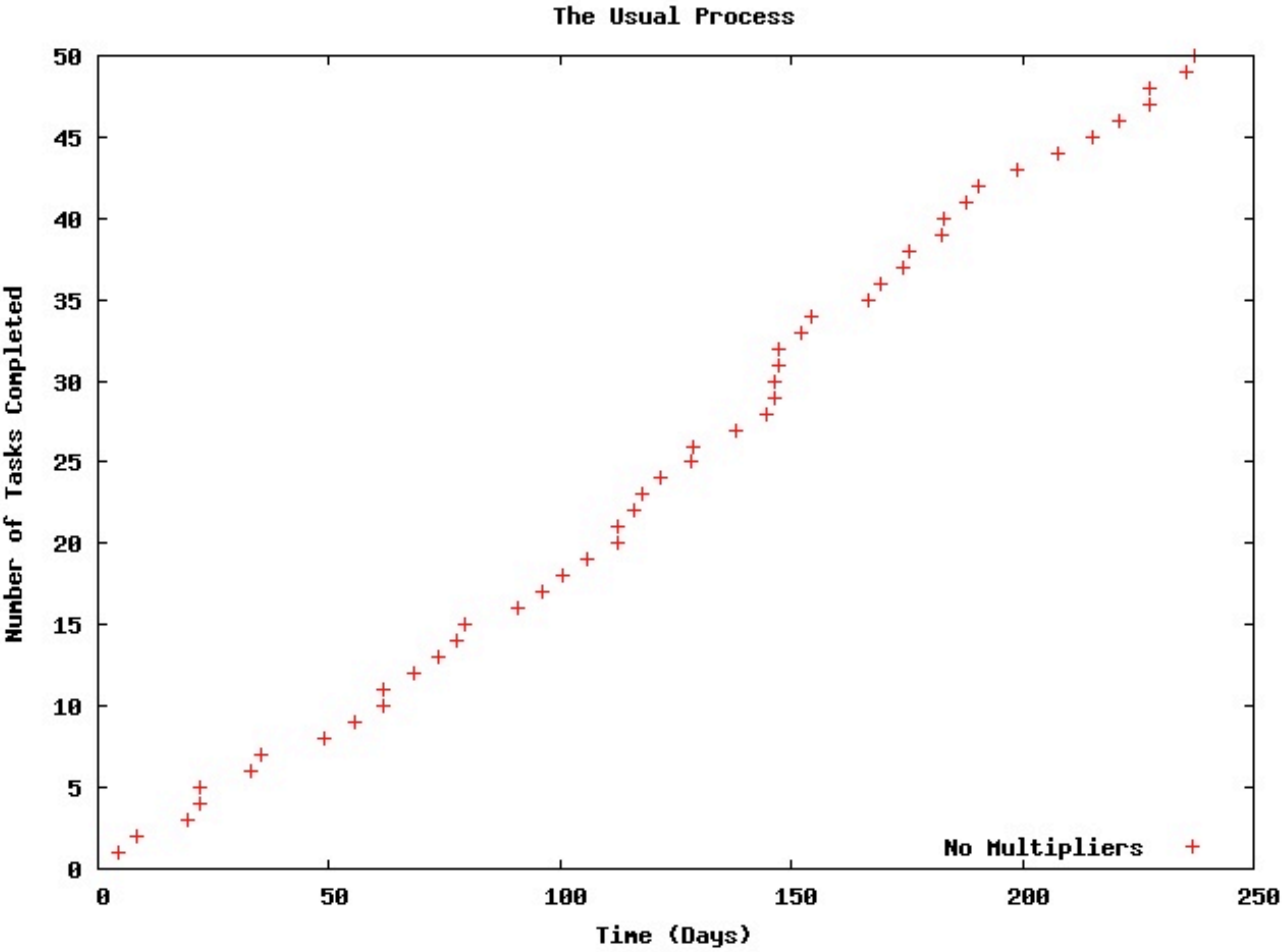
“Complexity”

Complexity Multiplier

- A change which increases the cost of future activities.

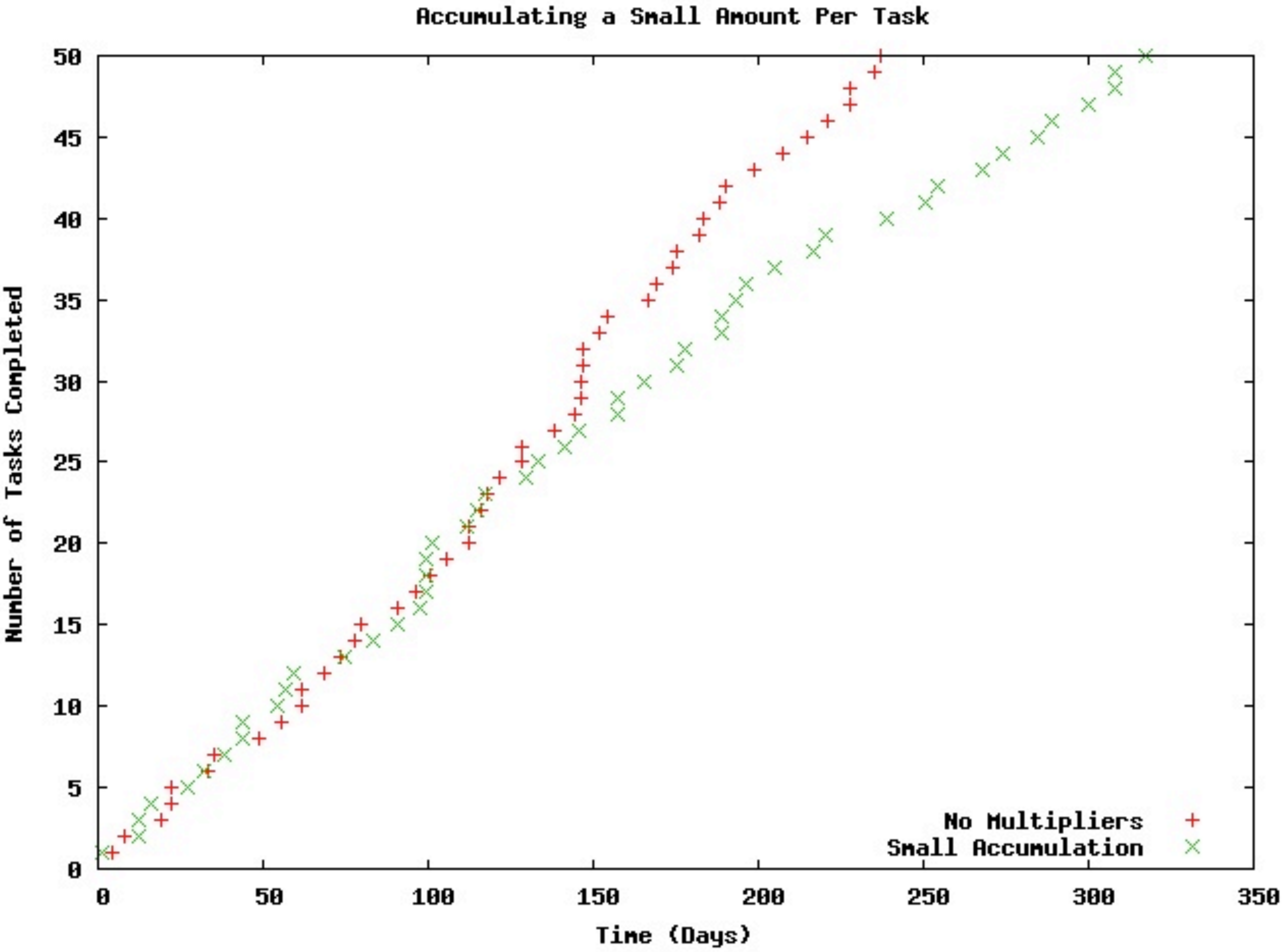
A Really Simple Model

- 50 Tasks
- Average task takes 5 days, normally distributed, with a deviation of 2 days


The Usual Process
Number of Tasks Completed
Time (Days)
0
5
10
15
20
25
30
35
40
45
50
0
50
100
150
200
250
No Multipliers +

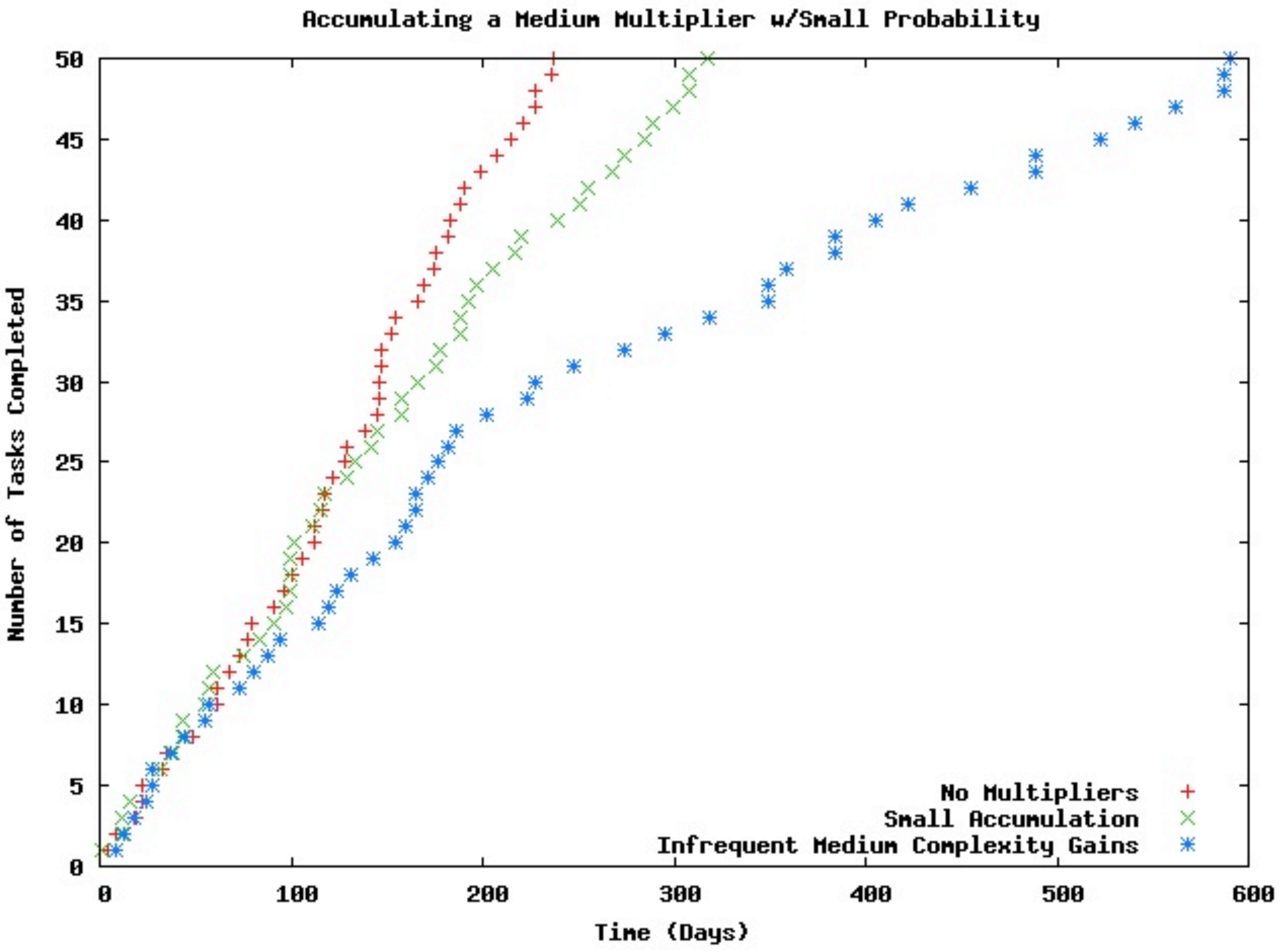
A Better Model

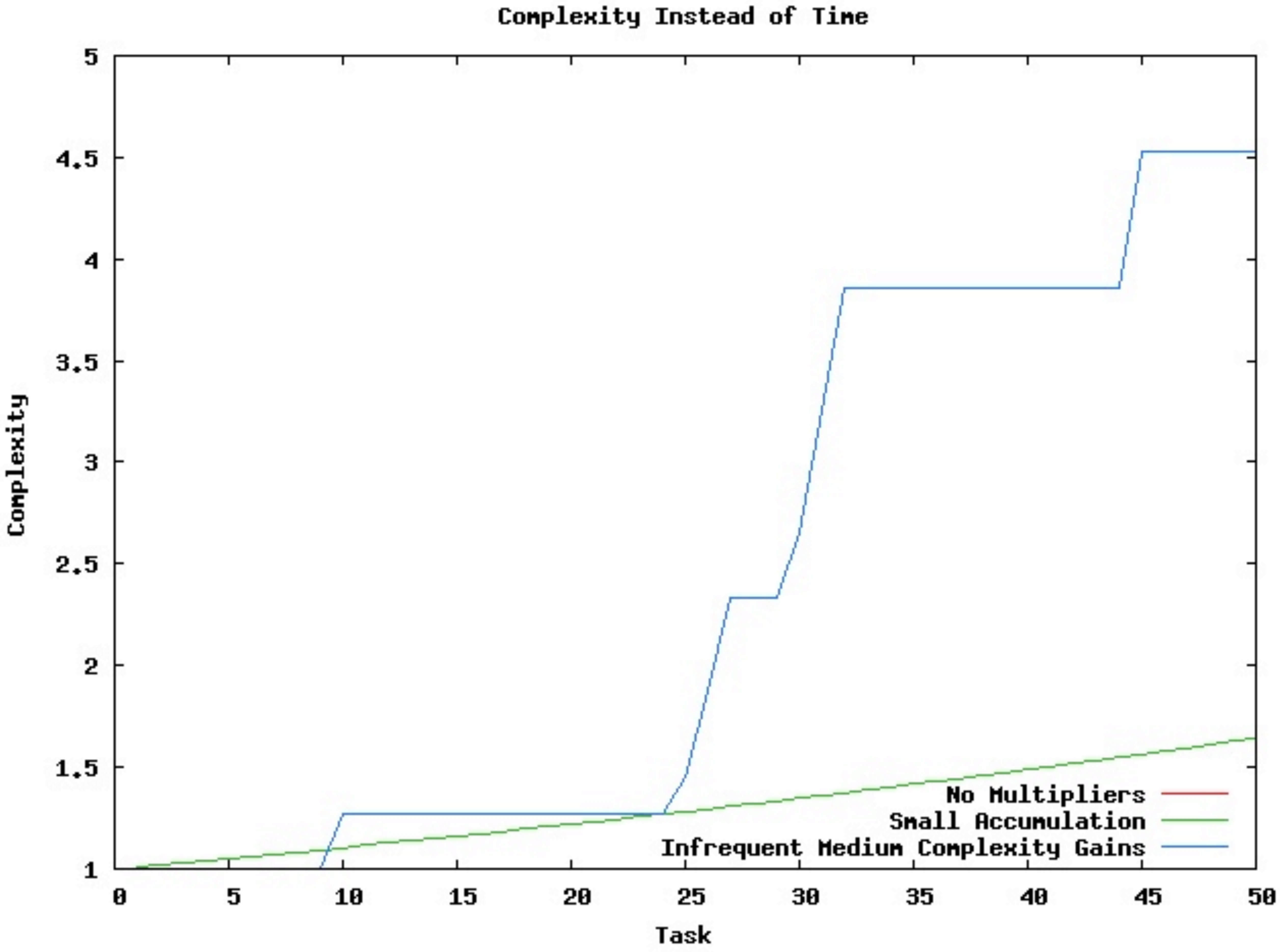
- 50 Tasks
- Tasks start as an average of 5 days with about 2 day deviation
- Each task accrues a 1% complexity multiplier - about 20 minutes on the first task

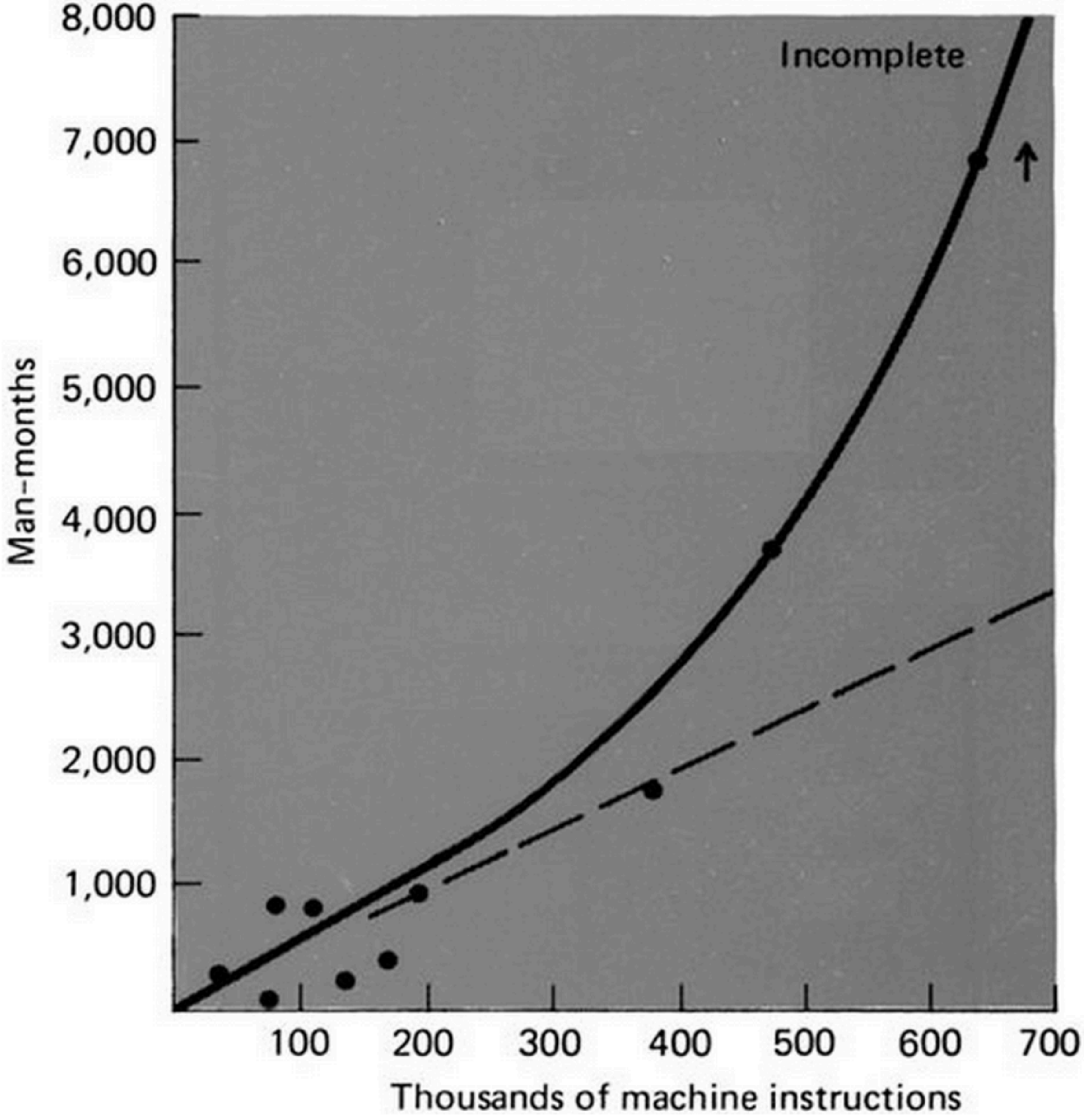

Accumulating a Small Amount Per Task
Number of Tasks Completed
Time (Days)
0
5
10
15
20
25
30
35
40
45
50
0
50
100
150
200
250
300
350
No Multipliers +
Small Accumulation ×

A "Realistic" Model

- 50 Tasks
- Tasks start as an average of 5 days with about 2 day deviation
- Each task has a 10% probability of incurring about 20% complexity (with a deviation of about 4%)


Accumulating a Medium Multiplier w/Small Probability
Number of Tasks Completed
Time (Days)
0
5
10
15
20
25
30
35
40
45
50
0
100
200
300
400
500
600
No Multipliers +
Small Accumulation ×
Infrequent Medium Complexity Gains ✱


Complexity Instead of Time
Complexity
Task
5
4.5
4
3.5
3
2.5
2
1.5
1
0
5
10
15
20
25
30
35
40
45
50
No Multipliers
Small Accumulation
Infrequent Medium Complexity Gains



From Mythical Man Month, Fred Brooks

Act II: Conflict

The Trade-Off

A task can *almost always* be completed more quickly by ignoring its effect on the complexity of the system.

“Let’s Maintain Two”

- Code modules
- Git repositories
- Code paths
- Config files

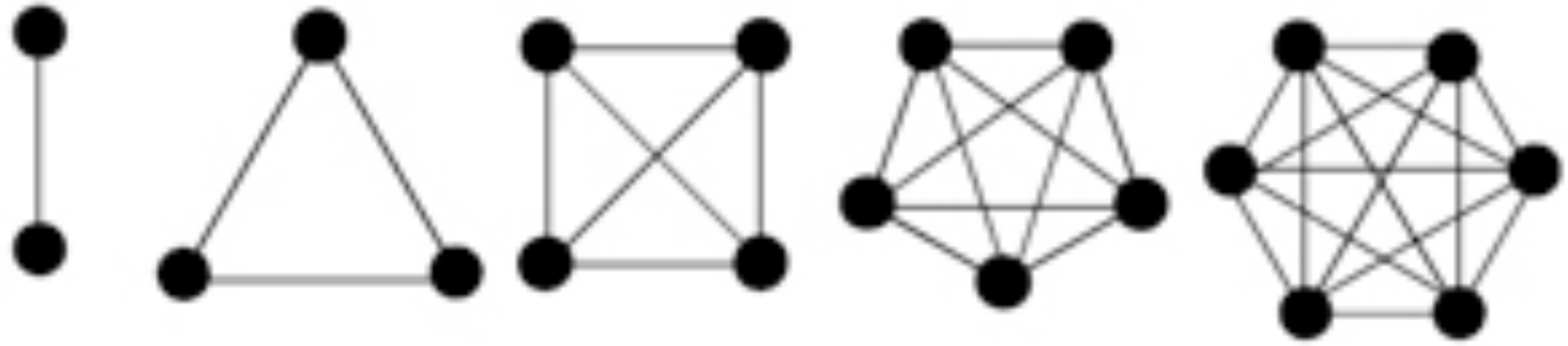
“A basic principle of data processing teaches the folly of trying to maintain independent files in synchronism. It is far better to combine them into one file with each record containing all the information both files held concerning a given key.”

– Fred Brooks (in 1975)

Stage Gates

- Sign-off required by another team
- Work required by another team
- Any process required for each task (or group of tasks) to proceed

Communication



People	2	3	4	6	6	(n)
Interfaces	1	3	6	10	15	$\frac{n^2-n}{2}$

Fuzzy Version Matching

- Package A: 2 point versions released
- Package B: 3 point versions released
- Package C: 2 point versions released
- Total: 12 configurations

Language Design

```
(defvar foo 42)

(defun foo ()
  79)

(defmacro foo ()
  ‘(* 7 9))
```

```
(symbol-value ‘foo)
  => 42

(symbol-function ‘foo)
  => #<FUNCTION FOO>
; equivalent to #‘foo

(macro-function ‘foo)
  => #<FUNCTION {...}>
```

```
(setf ‘foo 24)

(setf
  (symbol-function ‘foo)
  #’(lambda () 26)))

(setf
  (macro-function ‘foo)
  #’(lambda (x) x))
```

```
(let ((bar 5))
  ...)

(flet ((bar ()
         6))
  ...)

(symbol-macrolet ((foo ()
                    ‘(* 6 4)))
  ...)
```

```
(let* ((bar 5)
       (baz (* bar 2)))
  ...)

(labels ((bar ()
           5)
         (baz ()
           (* (bar) 2)))
  ...)

; No equiv. for symbol-macrolet
```

defun

defvar

defmacro

macro-function

symbol-value

symbol-function

let

let*

flet

labels

symbol-macrolet

macrolet

compiler-let

makunbound

fmakunbound

```
define
let
let*
define-syntax
let-syntax
let-syntax*
```

=

```
(let ((x 5))
  (labels ((somefn ()
             ...))
    (let ((y 7))
      (labels ((otherfn ()
                 ...))
        ...))))
```

```
(let* ((x 5)
       (somefun (lambda () ...))
       (y 7)
       (otherfun (lambda () ...)))
  ...)
```

```
x = 5
somefun = ...
y = 7
otherfun = ...
```

Java

- int versus Integer
- int[] versus List<Integer>
- (autoboxing only works for simple types)

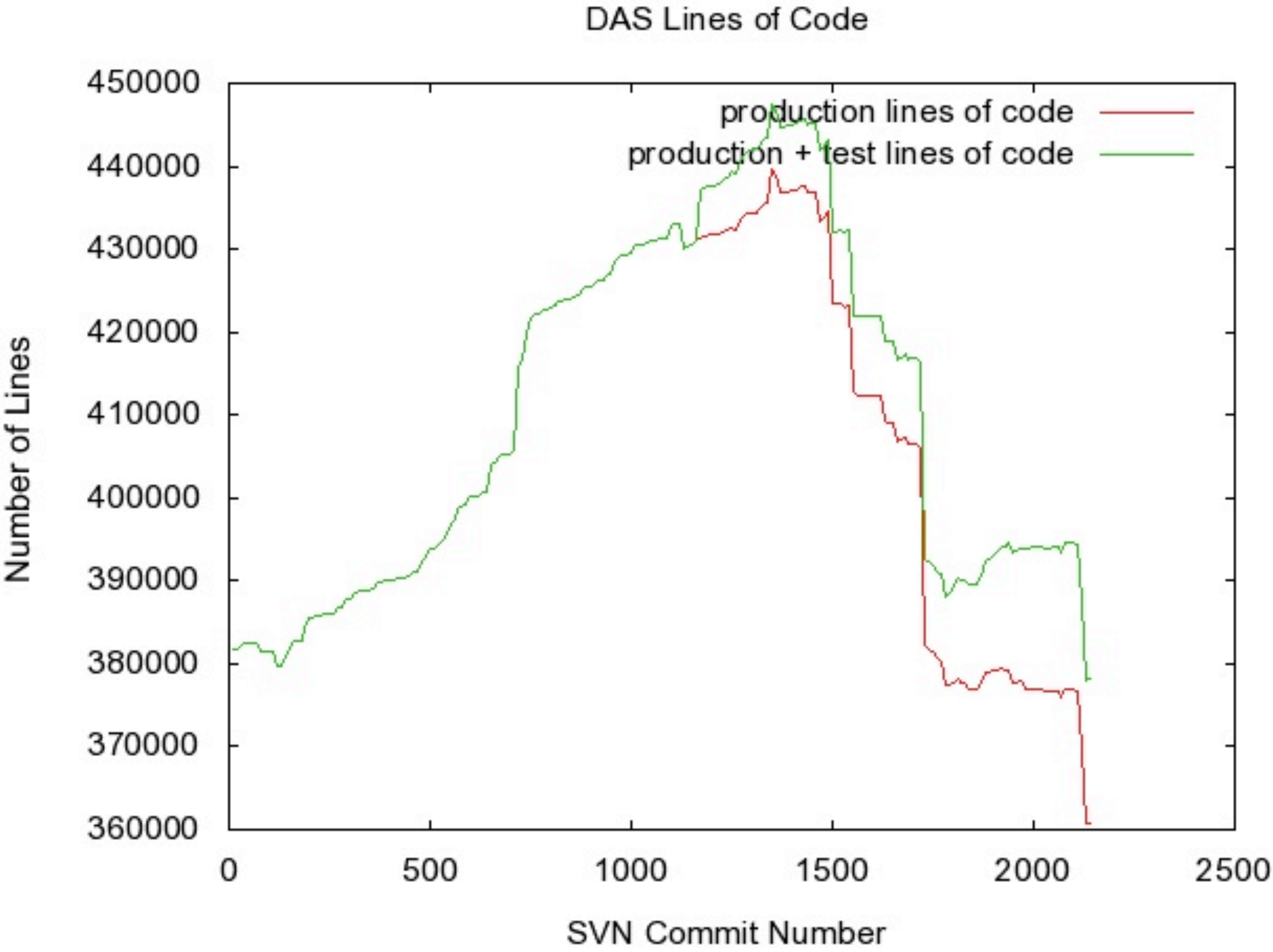
In Node.js

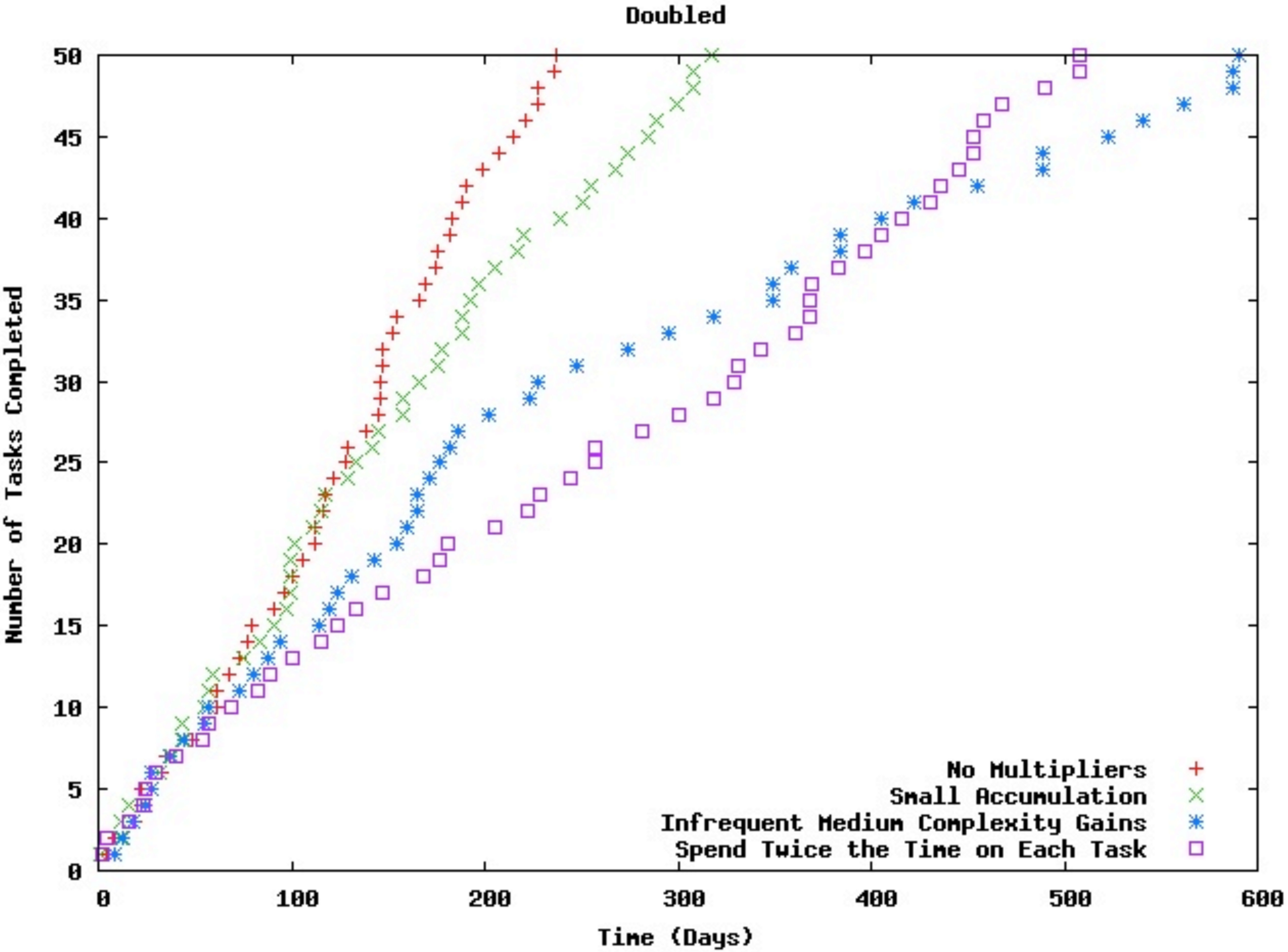
- CPS-style functions versus regular functions

Act III: Resolution

Complexity Dividers

(or "Complexity Multipliers < 1.0")


DAS Lines of Code
production lines of code
production + test lines of code
Number of Lines
450000
440000
430000
420000
410000
400000
390000
380000
370000
360000
0
500
1000
1500
2000
2500
SVN Commit Number


Doubled
Number of Tasks Completed
0
5
10
15
20
25
30
35
40
45
50
0
100
200
300
400
500
600
Time (Days)
No Multipliers +
Small Accumulation ×
Infrequent Medium Complexity Gains ✱
Spend Twice the Time on Each Task □