

Patterns of Software Change

Michael Feathers
Groupon

Architecture |'ärki,tæk ch ər| *noun*,
the part of a system that you can not
replace without having a different
system

Unintentional Coupling is Inevitable

Unintentional Coupling is Inevitable



Class:

- Handles Tax Calculation
- Handles Charges & Discounts

Class:

- Handles Tax Calculation
- Handles Charges & Discounts

*Someone decides taxes can be modeled
as a charge*

Validation in the Model or the Controller?

Organic Patterns



Designer [Christopher Poehlmann](#)

Piecemeal Growth Is the Hallmark of Organic Systems

Piecemeal Growth Biases Toward Existing Structure

Architecture |'ärki,tæk ch ər| *noun*,

the part of a system that you can not
replace without having a different system

The Antidote to Stagnation is Continual Renewal

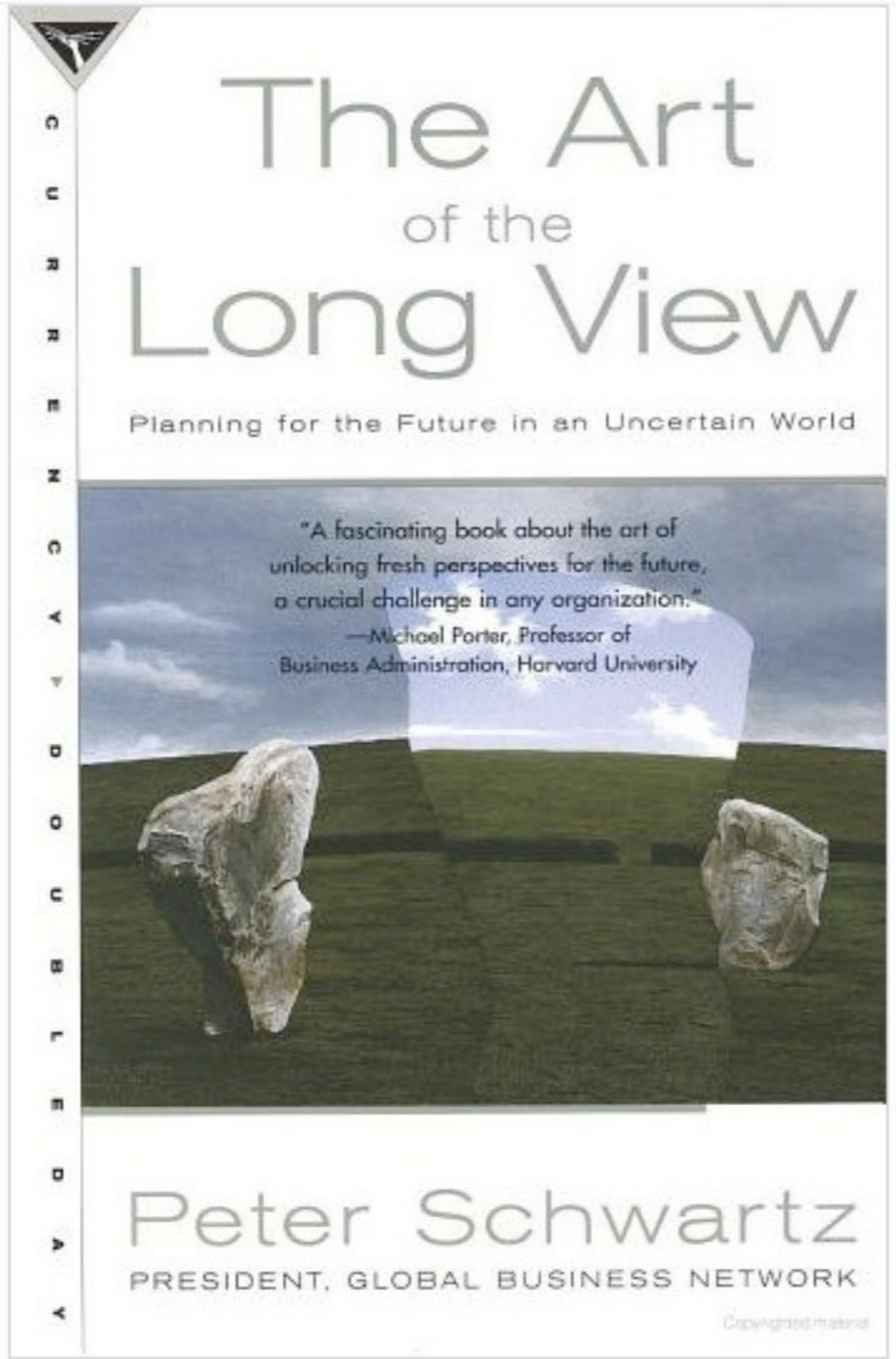
**..and appreciation for things
we can't change**

Self-Induced Disruption

Card Checking a Design

Card Checking a Design

“What is we did this now?”



The Art of the Long View

Planning for the Future in an Uncertain World

"A fascinating book about the art of
unlocking fresh perspectives for the future,
a crucial challenge in any organization."

—Michael Porter, Professor of
Business Administration, Harvard University

Peter Schwartz

PRESIDENT, GLOBAL BUSINESS NETWORK

Copyright © 2004

C
U
R
R
E
N
T
A
F
F
A
I
R
S

Over-Engineering |'ōvər ˌenjəˈni(ə)riŋ| *noun*,

weighting a possible future too highly in
design

The Coupling Sweet Spot

The Coupling Sweet Spot

Single Responsibility Might Be The Closest Thing We Have, But It Is Not Perfect

Accidental De-coupling

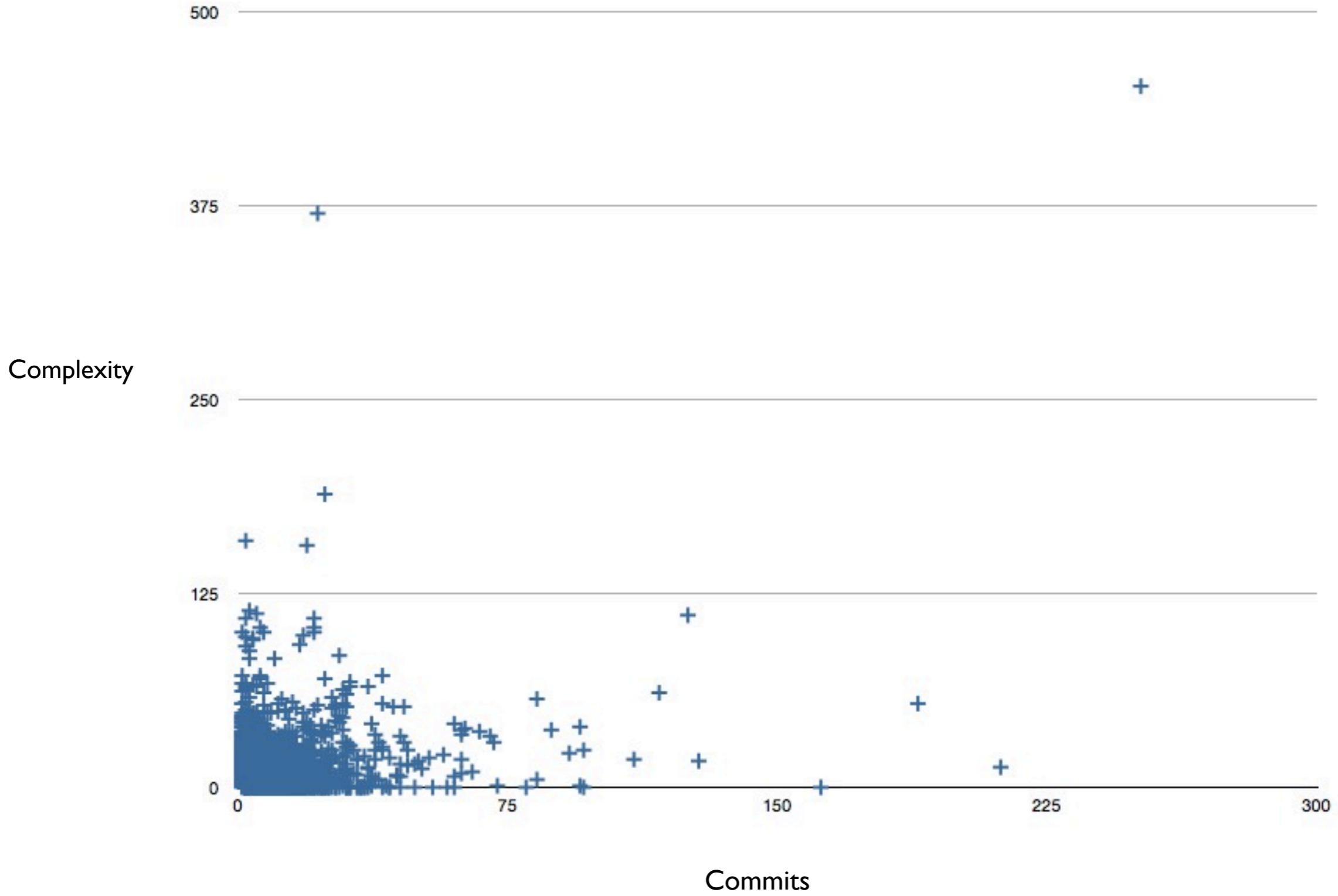
Accidental De-coupling

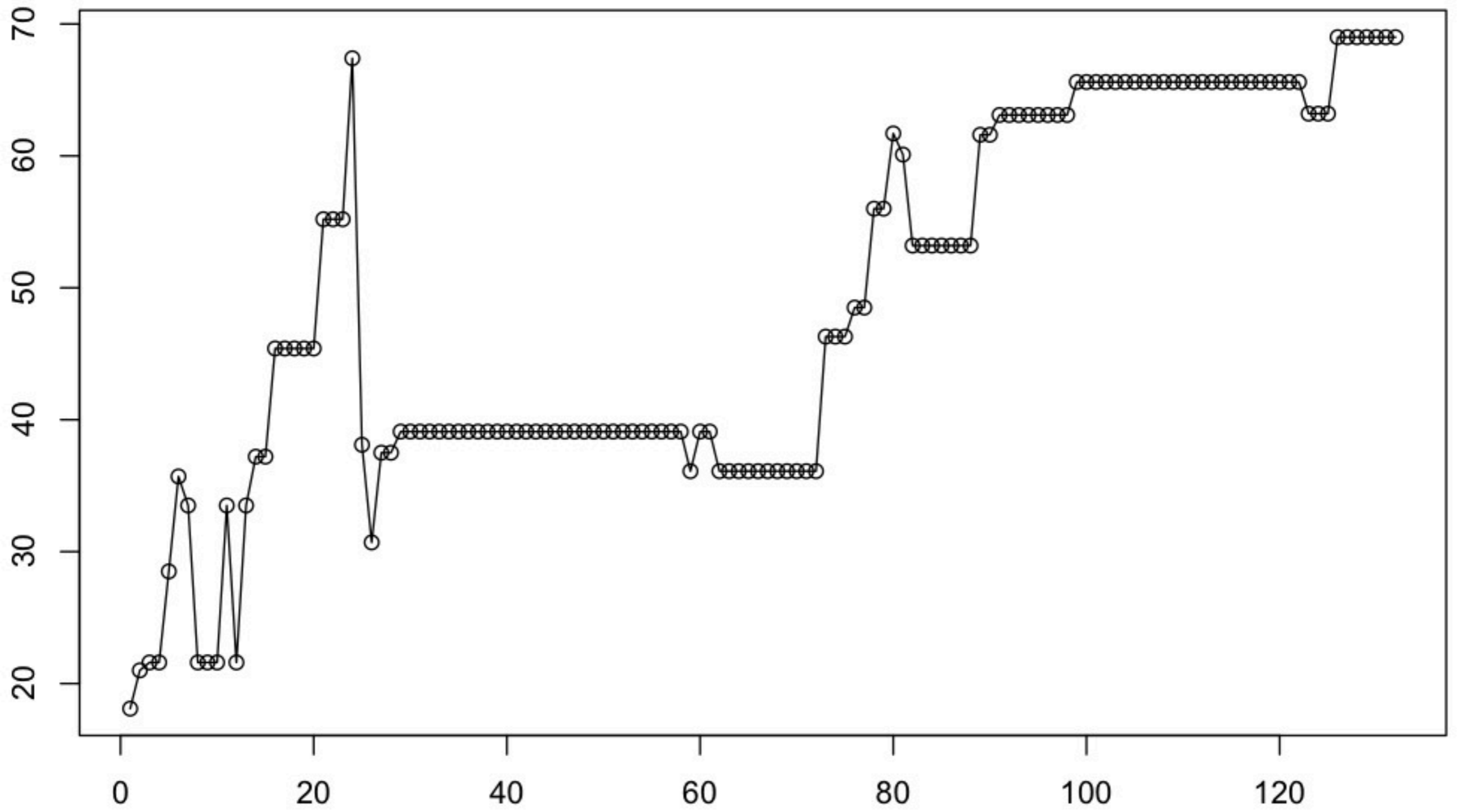
The Fitness Story

When to Jump Course

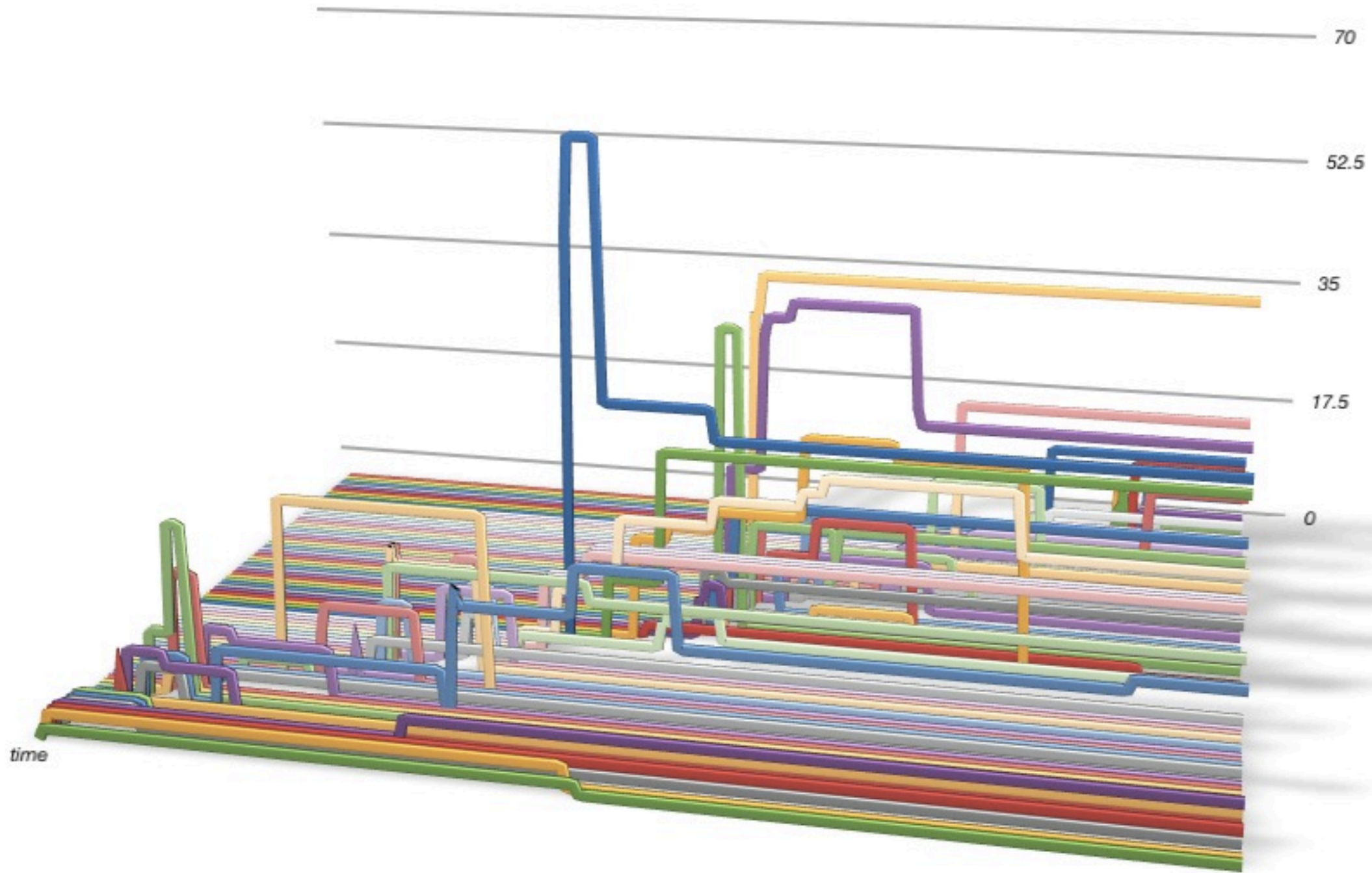
Activity Flurries

Turbulence





Method Complexity Trends in a Class



The Trending View

Methods Ascending

```
def methods_ascending_last_n events, n
  methods_by(method_events(events)) do |es|
    es.count >= n && \
      es.map(&:method_length).last(n).each_cons(2).all? {|l,r| l < r }
  end.keys
end
```

Trending Methods

```
def trending_methods events
  method_events(events).select { |e| e.status == :changed } \
    .group_by { |e| month_from_date(e.date) } \
    .to_a \
    .last[1] \
    .freq_by(&:method_name)
    .sort_by { |_, count| -count } \
    .take(10)
end
```

Static Views

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
```

Temporal Correlation of Class Changes

```
[[["App", "Inventory"], 277],  
[["Inventory", "Object"], 216],  
[["Admin", "Inventory"], 195],  
[["Inventory", "User"], 188],  
[["Inventory", "Users"], 171],  
[["Inventory", "Deals"], 167],  
[["App", "Object"], 159],  
[["App", "InventoryController"], 152],  
[["Inventory", "Order"], 149],  
[["User", "Users"], 149],  
[["App", "User"], 143],  
[["Inventory", "InventoryController"], 143],  
[["Api", "Inventory"], 141],  
[["Admin", "App"], 136],  
[["Campaign", "Orders"], 134]]
```

Temporal Correlation of Class Changes

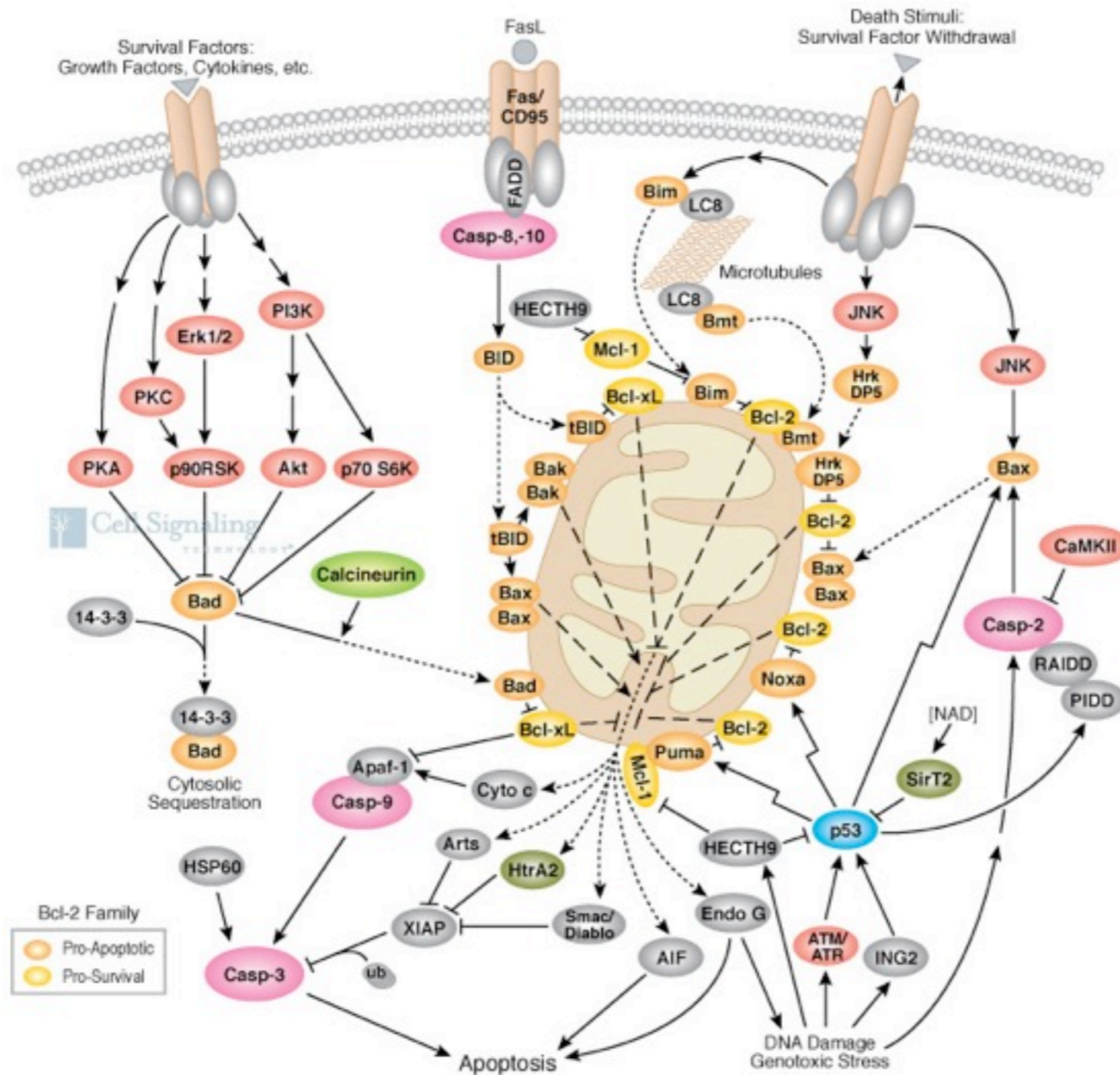
```
def temporal_correlation_of_classes events
  events.group_by { |e| [e.day, e.committer] } \
    .values \
    .map { |e| e.map(&:class_name).uniq.combination(2).to_a } \
    .flatten(1) \
    .pairs \
    .freq_by { |e| e } \
    .sort_by { |p| p[1] }
end
```

Objective vs. Subjective

Card Checking Can Reduce Metrics Bias

Self-Induced Disruption (2)

Apoptosis



Leveraging Conway

Understanding the Limits of Growth