Debugging with the Scientific Method

@stuarthalloway

Why Debugging?

because bugs!

look smart to you friends

understand Clojure

Why Me?

Clojure screening

Datomic support

gray hair

lazy

Why Clojure?

I often find myself "guessing" what's wrong with my Clojure code. Especially when interacting with dependencies and I handle some of the data incorrectly ("type" errors), it's not uncommon to get only some cryptic "Cannot cast Java.lang.xyz to abc". Then I go through the code adding tests or print statements, stare at different parts of the code, and finally realize the problem was that I forgot to retrieve the first element of a list, and returned a unitary list instead, or something silly like that.

https://www.reddit.com/r/Clojure/comments/3ry3vp/do_you_feel_in_the_dark_when_debugging_clojure/

Why Not Tools?

Cursive

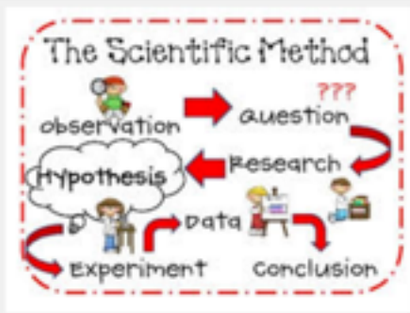
(cider[ ])

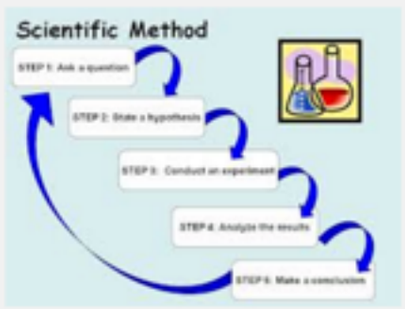
If you don't know
where you are going,
you might wind up
someplace else.

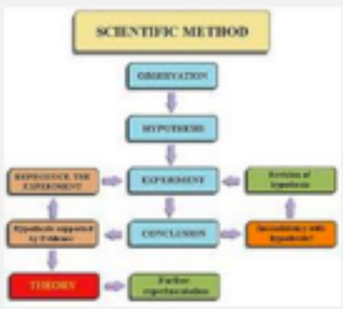
https://upload.wikimedia.org/wikipedia/commons/f/f1/1953_Bowman_Yogi_Berra.jpg

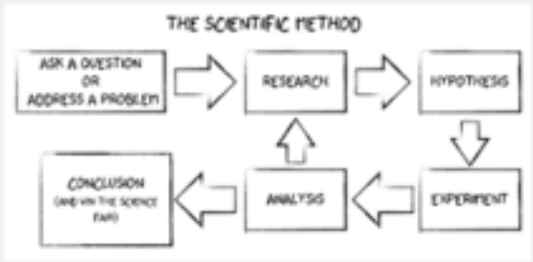
Why Scientific Method?

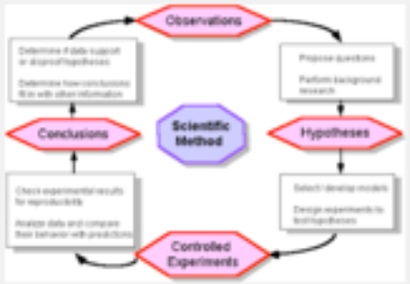


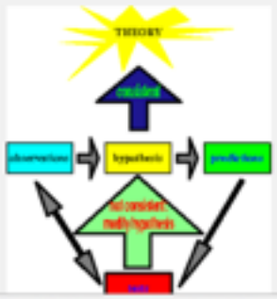


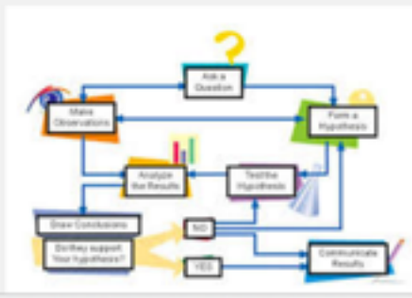


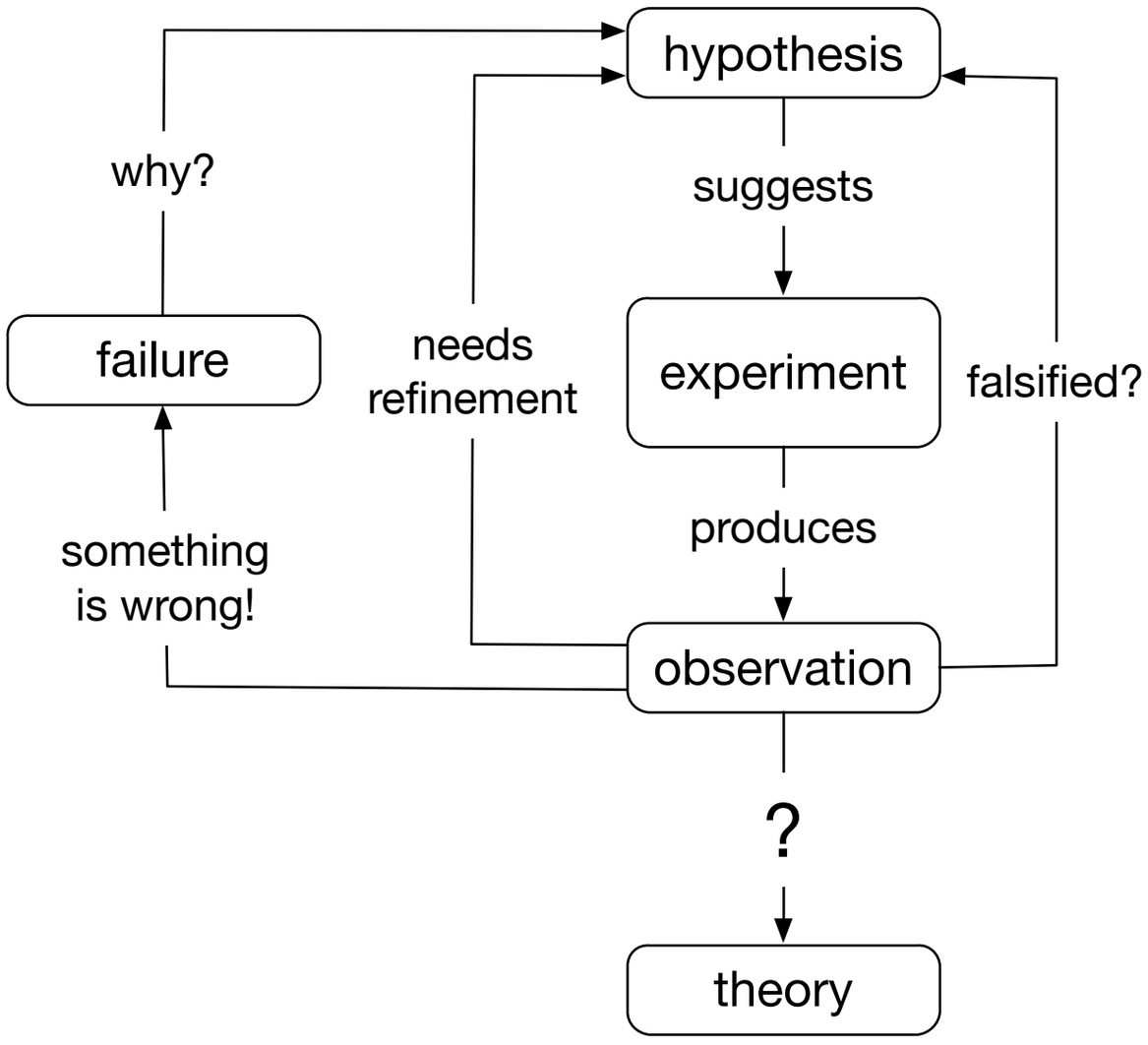




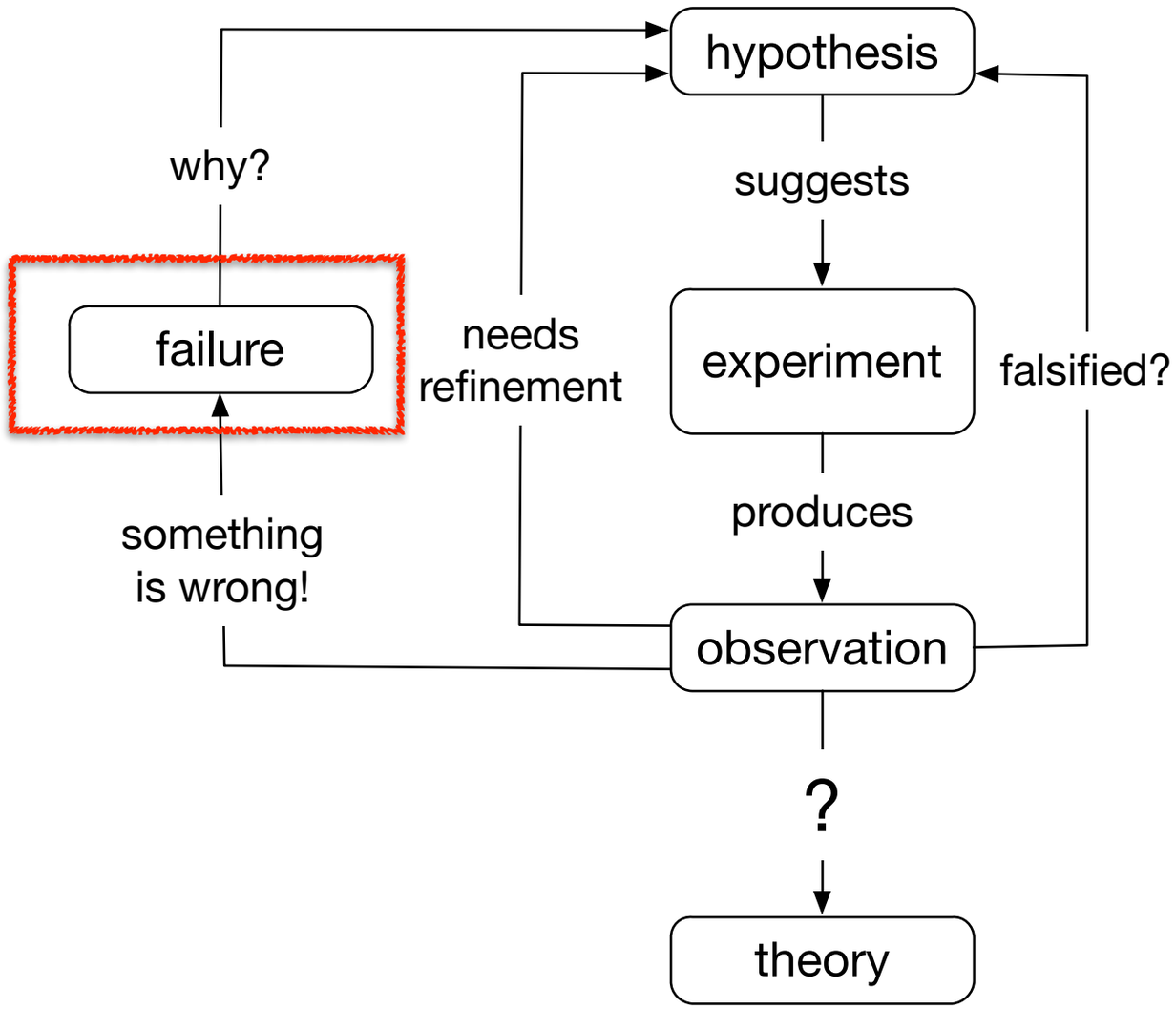




SM for Debugging



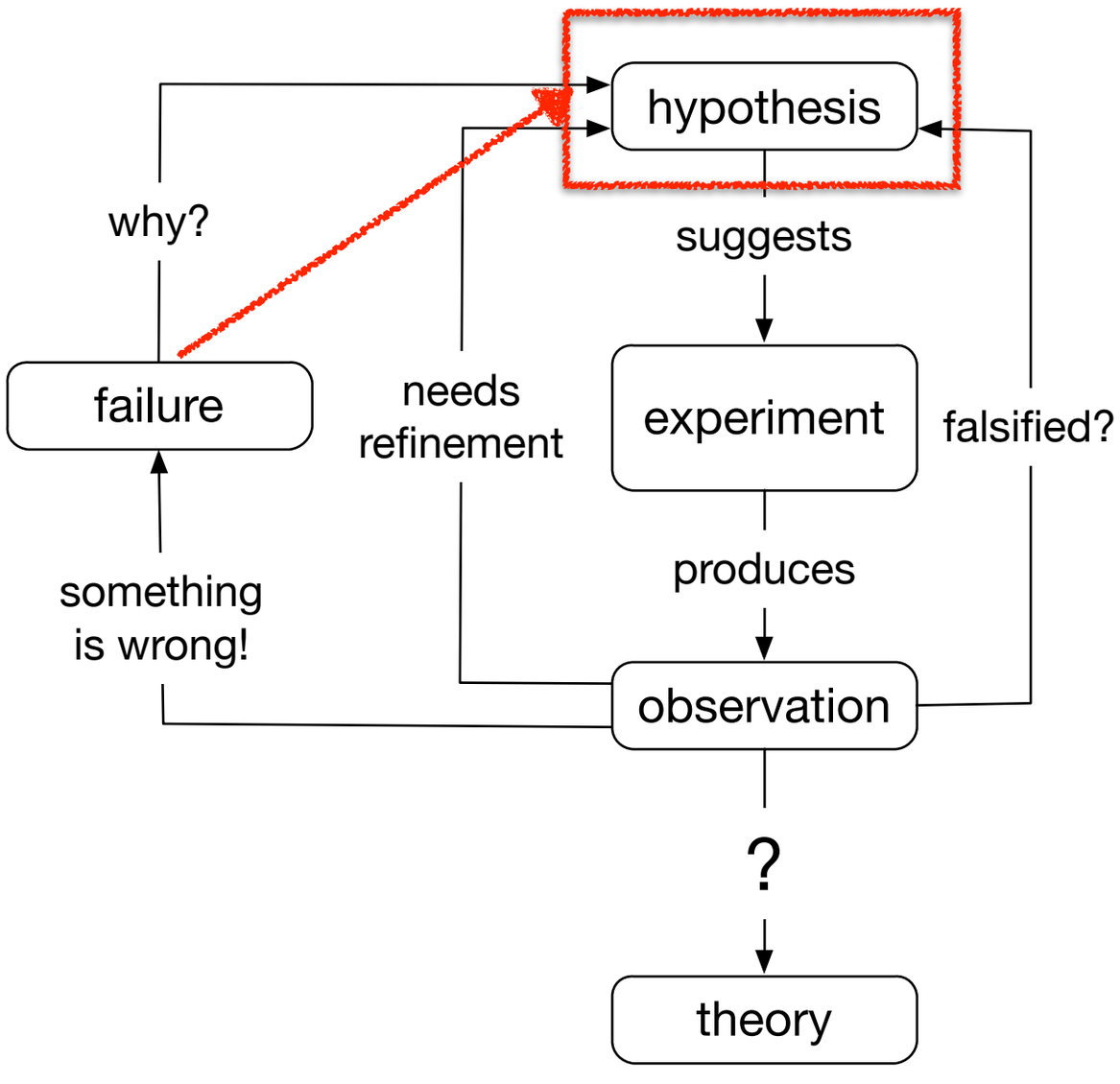
Failure



lack of success

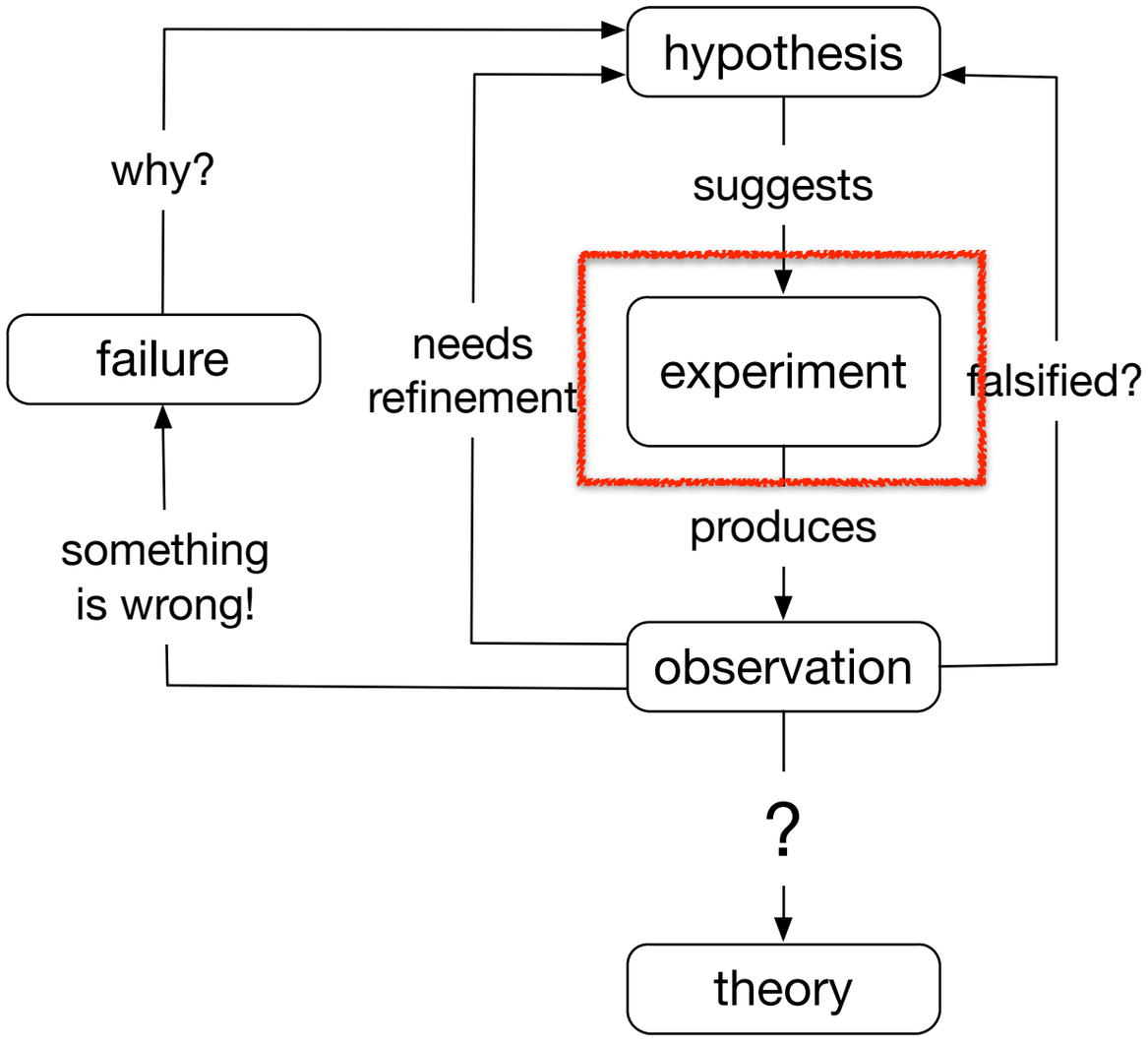
omission of
expected action

Hypothesis



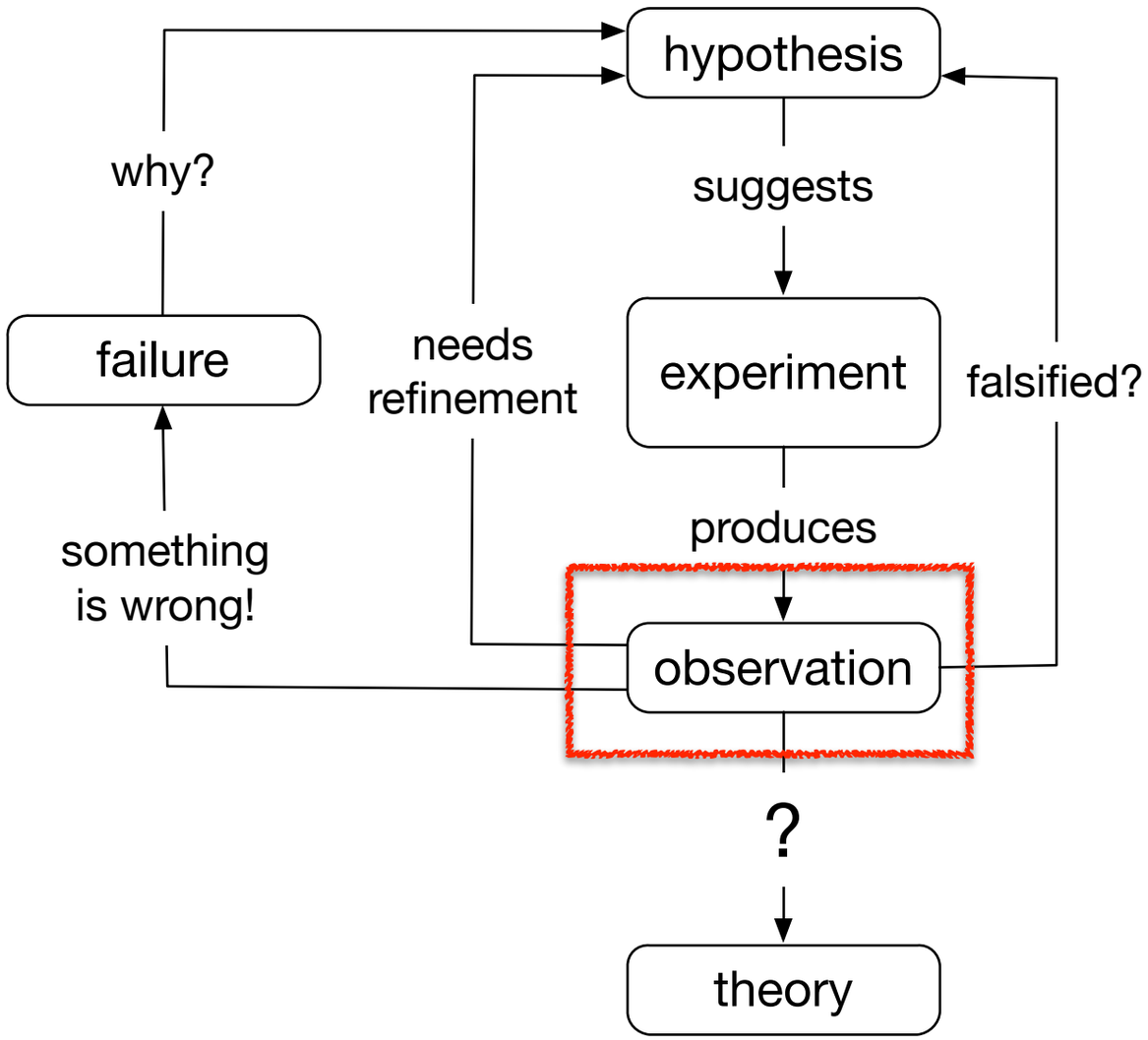
a proposed explanation made on the basis of limited evidence as a starting point for further investigation

Experiment



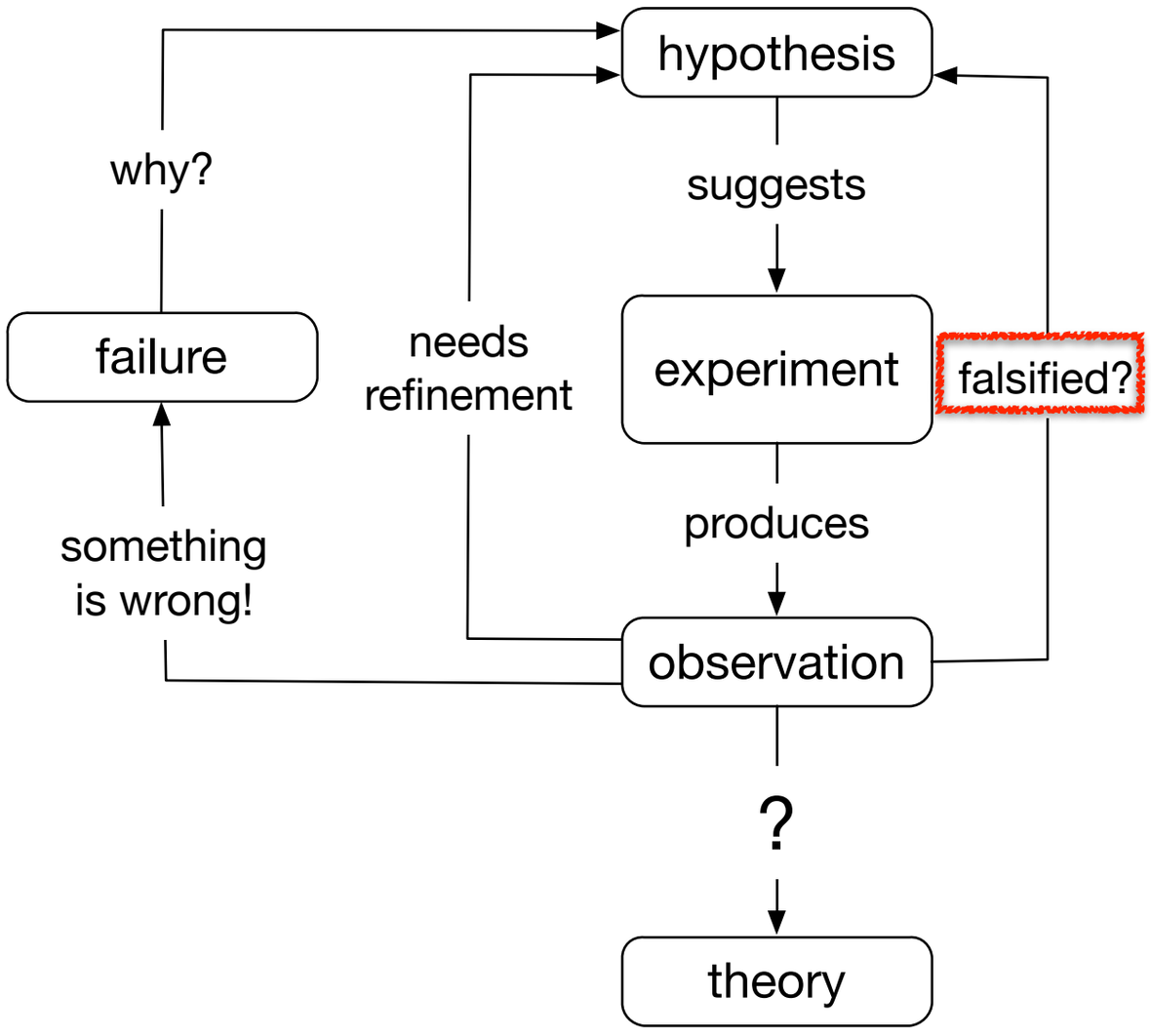
a test, trial, or tentative procedure

Observation



active acquisition of information from a primary source

Falsification



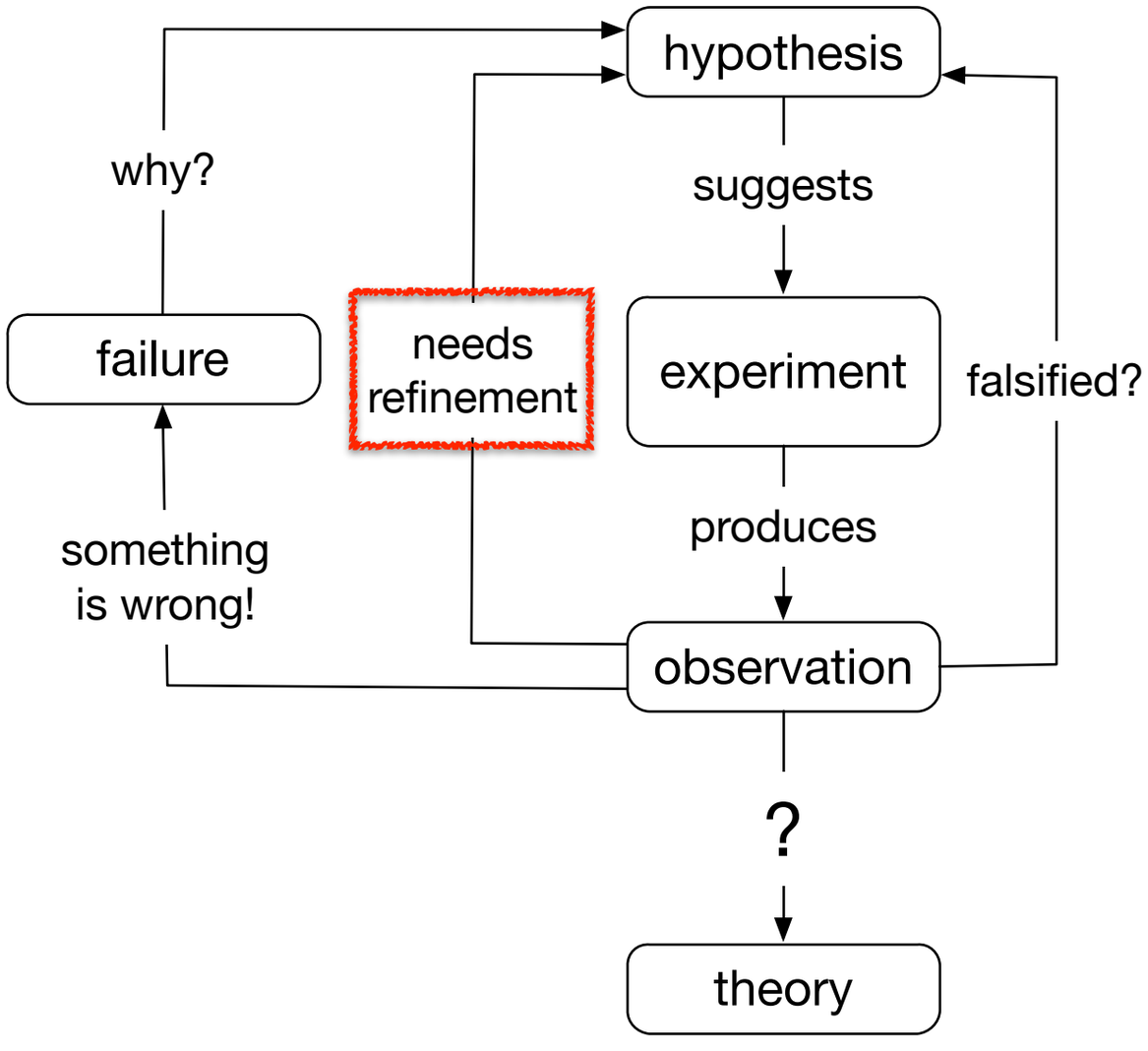
deductive process
using modus tollens:

$$H \rightarrow \neg O$$

$$O$$

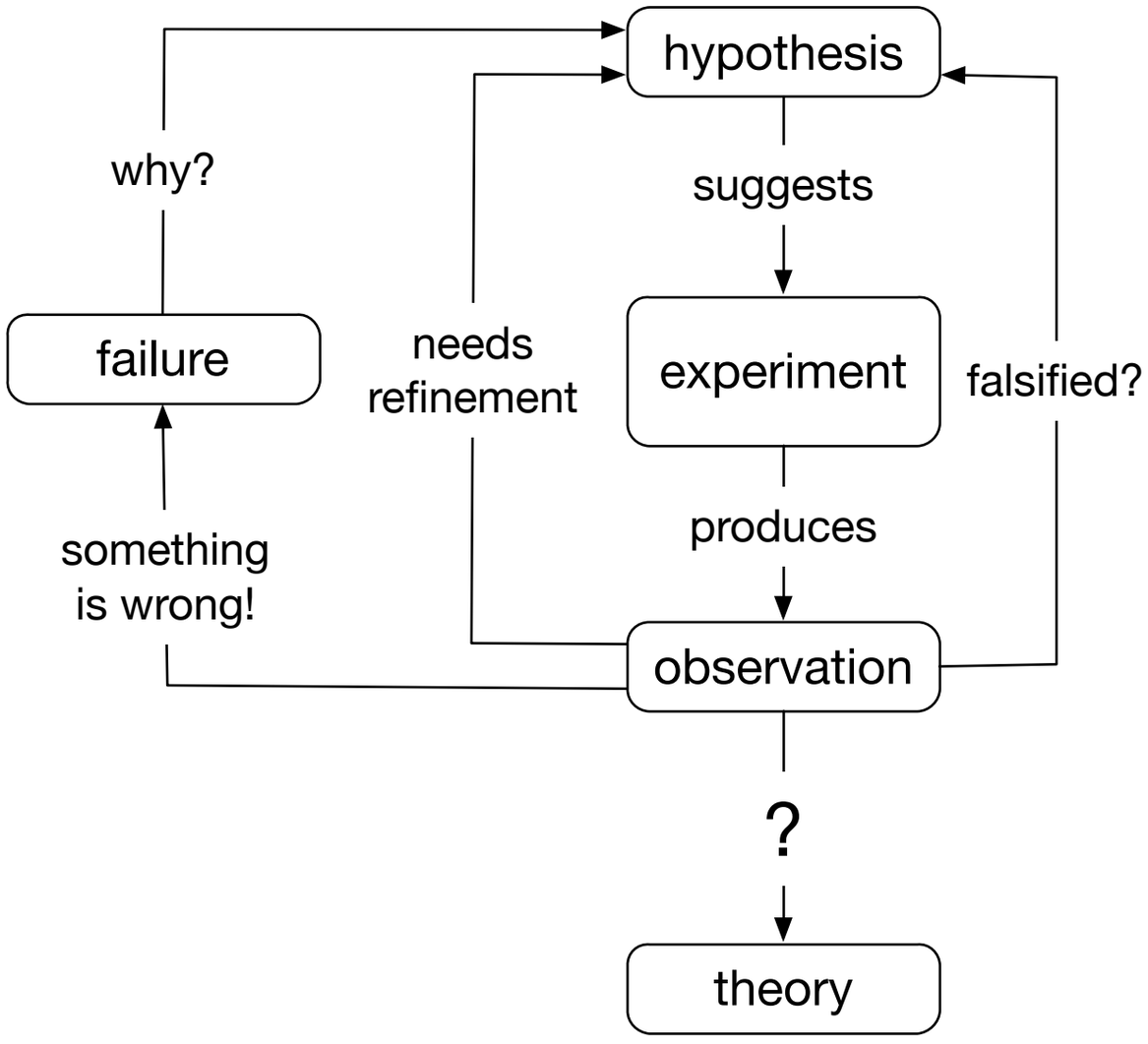
$$\neg H$$

Refinement



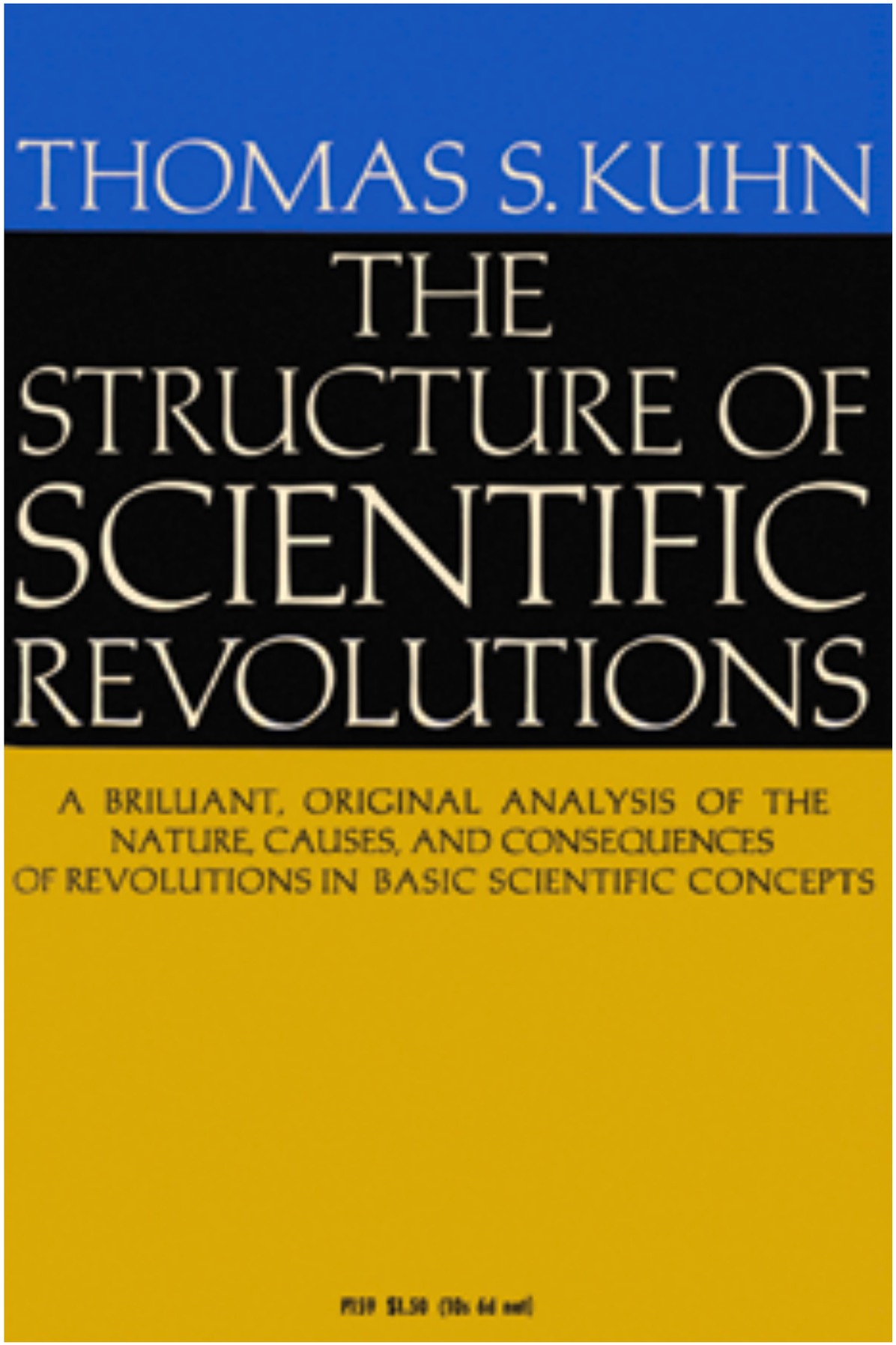
the process of
removing impurities
or unwanted elements
from a substance

Theory



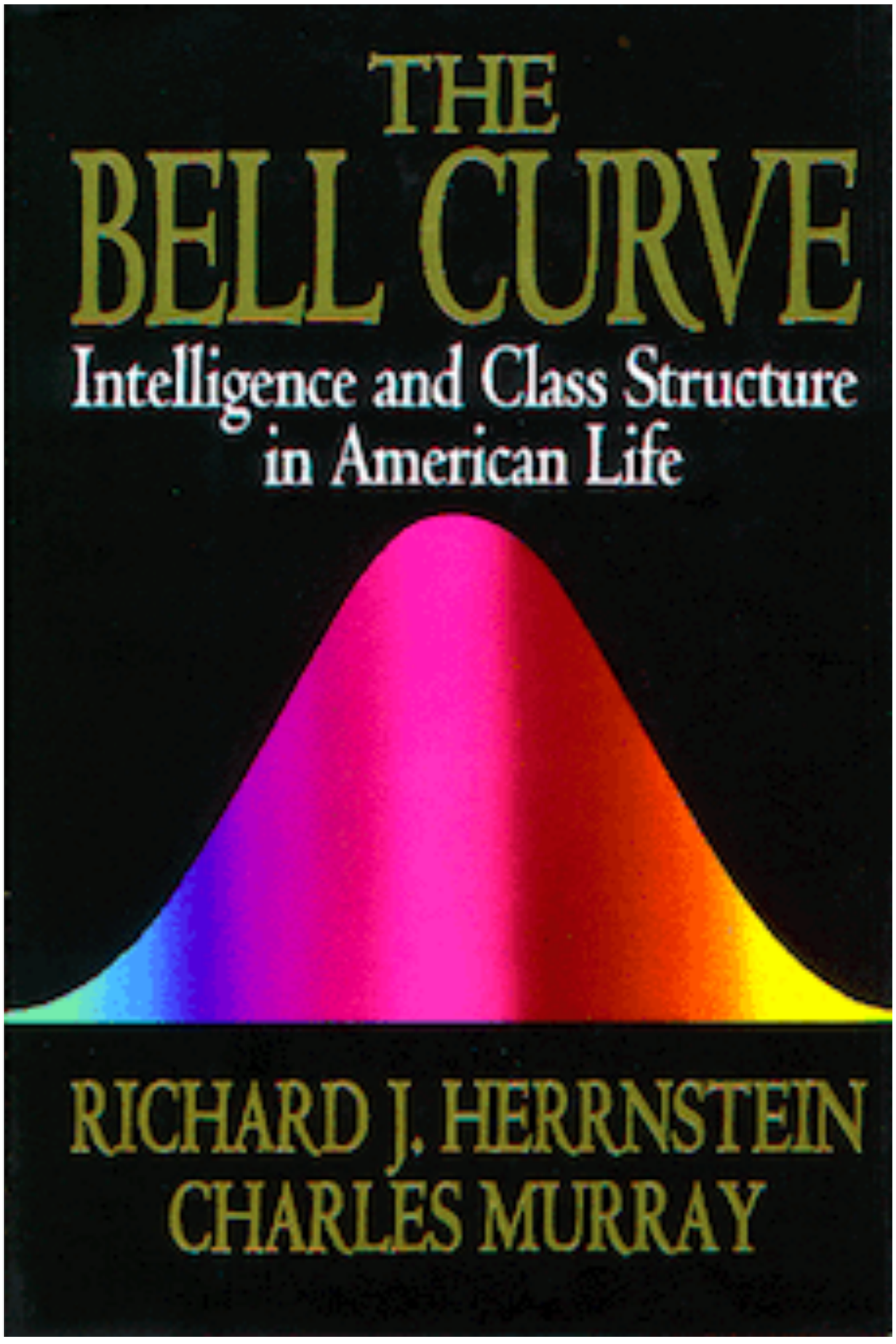
a hypothesis offering valid predictions that can be observed

http://a2.files.biography.com/image/upload/c_fit,cs_srgb,dpr_1.0,q_80,w_620/MTIxNDI3MjkzNDE1MTc5Nzg5.jpg



epistemological challenges

moral challenges



Debugging:
Most “Scientific” Thing Ever!

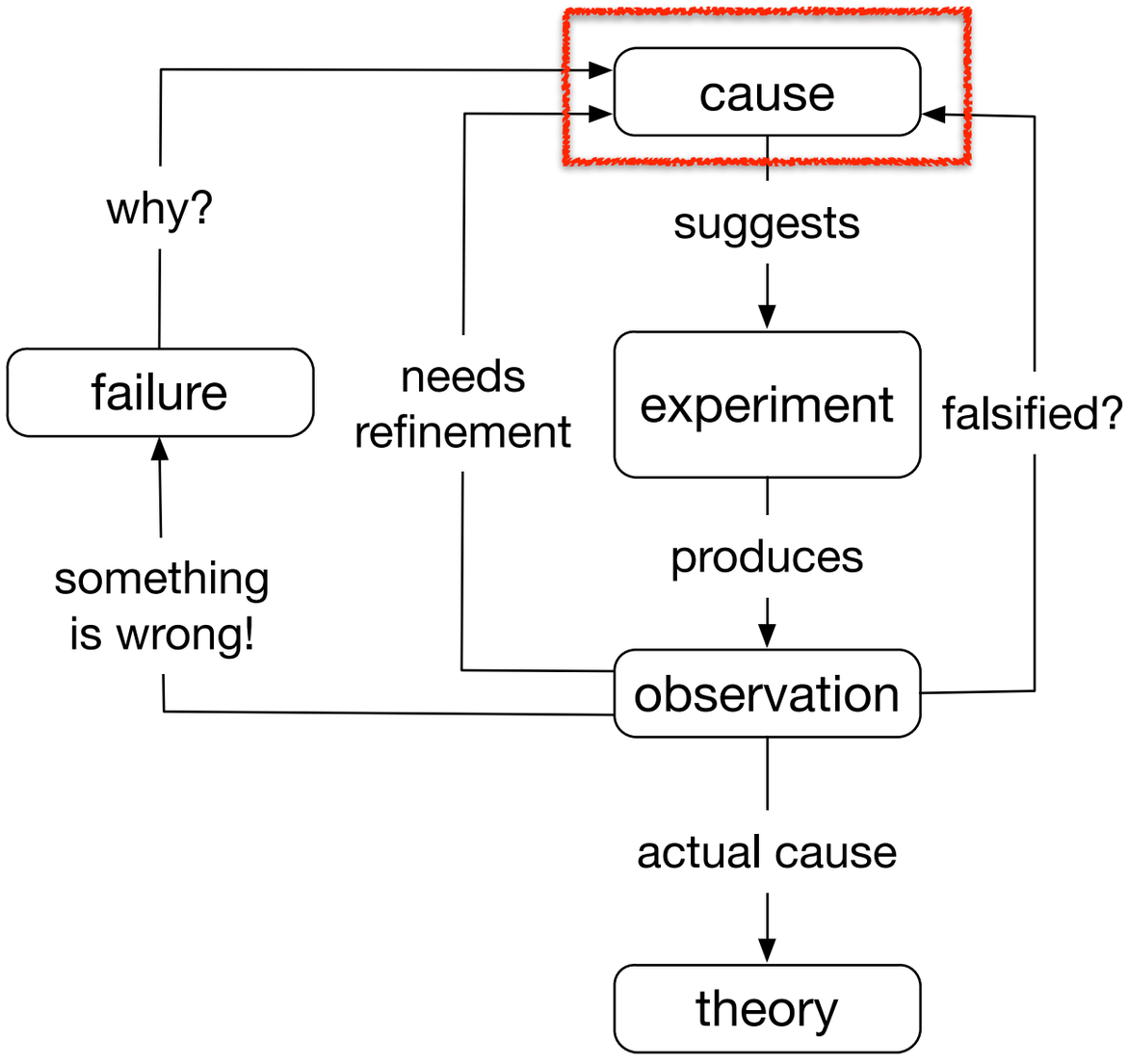
more constrained than science

deductive, not inductive

stakeholders typically identified

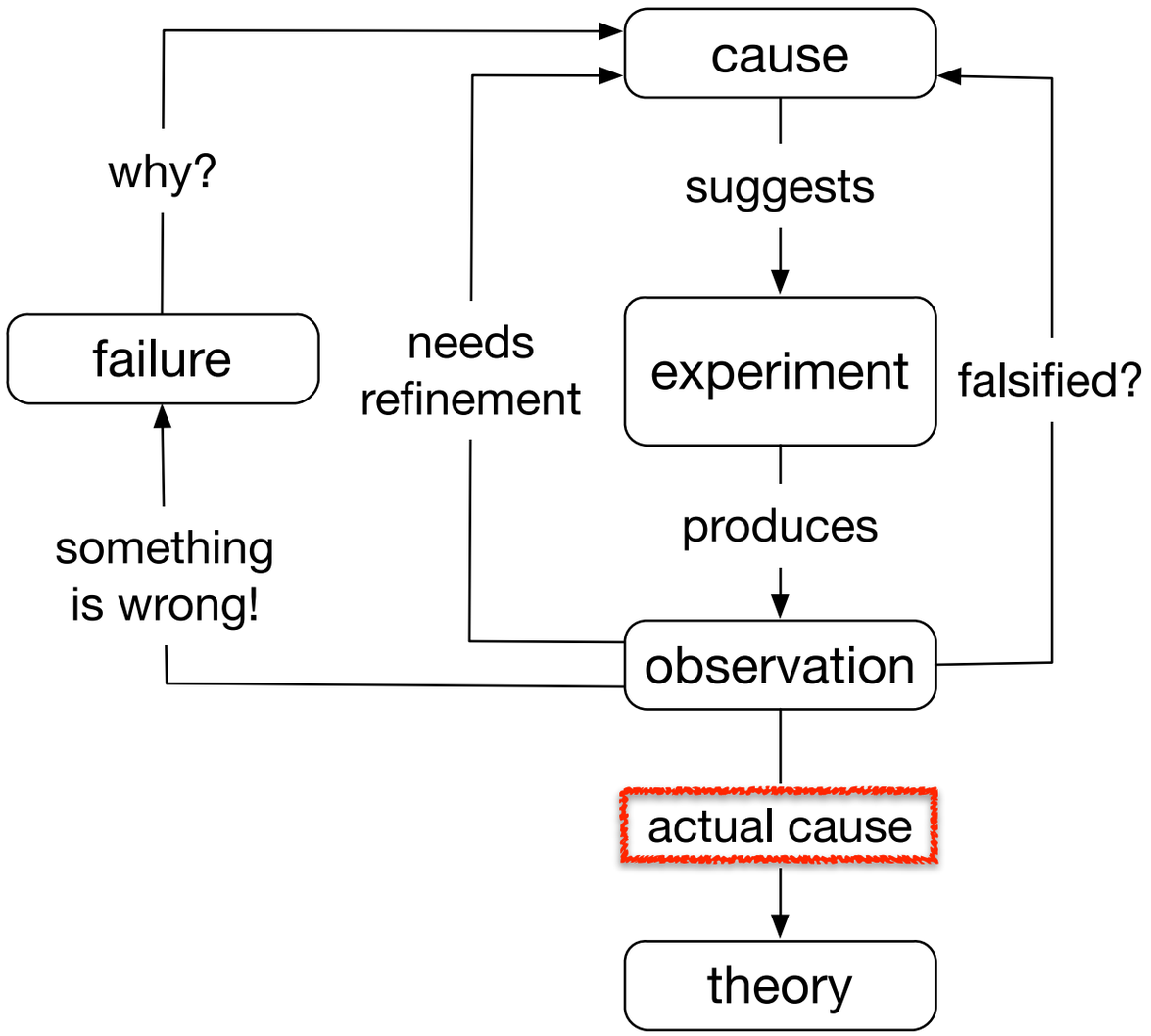
moral outrage pleasantly rare

Cause



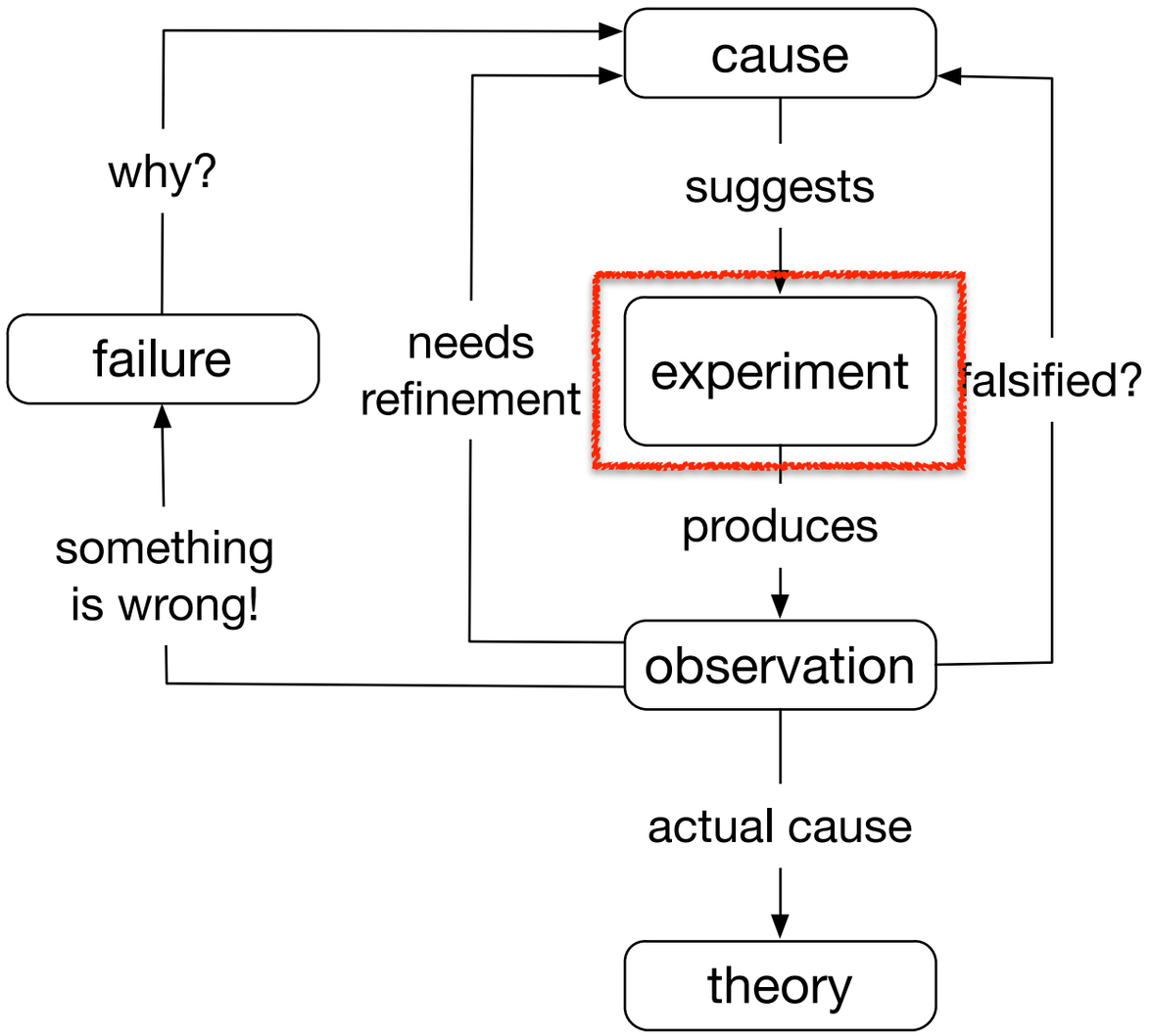
an event preceding an effect without which the effect would not have occurred

Actual Cause



difference between the actual world and the closest possible world in which the effect does not occur

Fix



an experiment that establishes an actual cause

Why Is This Partial Not Working?

```
(def partial-join
  (partial (clojure.string/join ",")))
```

```
=> (partial-join ["foo" "bar"])
```

```
ClassCastException
java.lang.String cannot be cast to clojure.lang.IFn .
repl/eval12557 (form-init2162333644921704923.clj:1)
```

http://stackoverflow.com/questions/32929149/why-this-partial-not-working

What to Do?

Better error messages?

Better docs?

Debugger? Syntax highlighter?

Static typing? Schema?

Stare at It?

Science

Hypotheses

"why is this ~~partial~~ not working"

join doesn't (do what I expected)

partial doesn't (do what I expected)

def doesn't (do what I expected)

```
(def partial-join
  (partial (clojure.string/join ",")))
```

Bottom Up REPL Check

```
(clojure.string/join ",")
=> ","
```

```
(def partial-join (partial ",")))
```

Weak Science is Stronger than Strong Tools

Consider the previous example

Poor problem statement

Incomplete hypotheses

Exploratory experiments

Minimal domain knowledge

Doing SM Well

clear problem statement

efficient hypotheses

good experiments

useful observations

writing it all down

Problem Statements

steps you took

what you expected

what actually happened

http://oceans11.wikia.com/wiki/File:Vlcsnap-2013-08-28-00h56m04s3.png

Don’t Sweat Naming

Divide and conquer

Decrease and conquer

Bisection

Interval halving

Proportional reduction

Quick! Where's the Bug?

Your App	
Clojure Wrappers	
Clojure Lang	Popular Java Lib
JVM	
OS	
hardware	
physics	

Good Experiments

reproducible

driven by hypothesis

small

change only one thing

Which of these should *not* be in your repro case?

clojure.test

Cursive

Prismatic schema

Midje

Potemkin

nREPL

Leiningen

Lein plugins

core.typed

test.generative

Making Observations

understand all the outputs

suspect correlations

use good tools

Unrelated?

```
10:40:20,444 |-WARN
Resource [logback.xml] occurs multiple
times on the classpath.
```

Write It Down

problem statement

hypotheses

what experiment should show

why experiment even makes sense

observations

https://en.wikipedia.org/wiki/Mastermind_(board_game)

https://en.wikipedia.org/wiki/Mastermind_(board_game)

Software Specifics

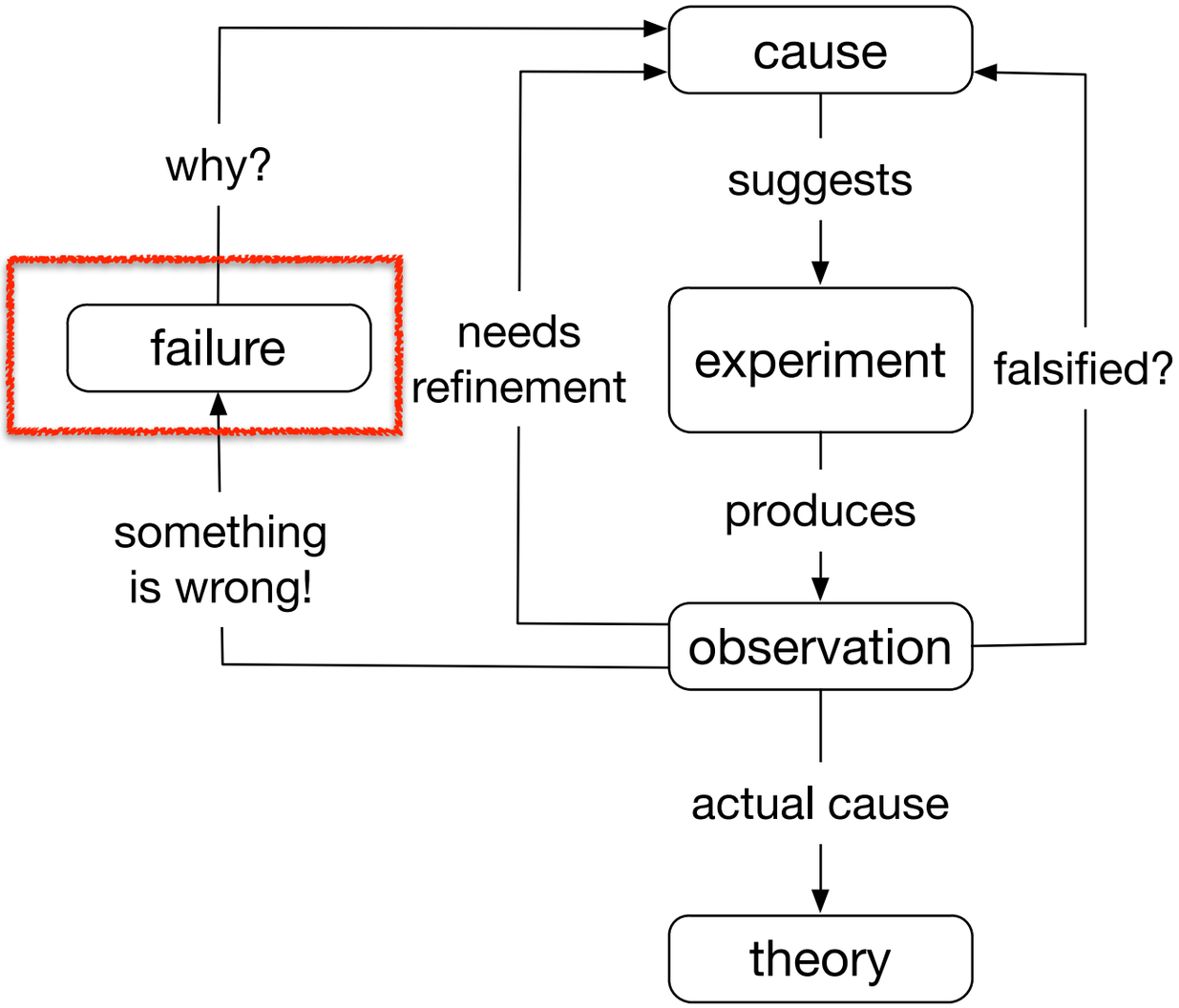
SECOND EDITION

THE C PROGRAMMING LANGUAGE

BRIAN W. KERNIGHAN
DENNIS M. RITCHIE

PRENTICE HALL SOFTWARE SERIES

The Failure is not the Defect

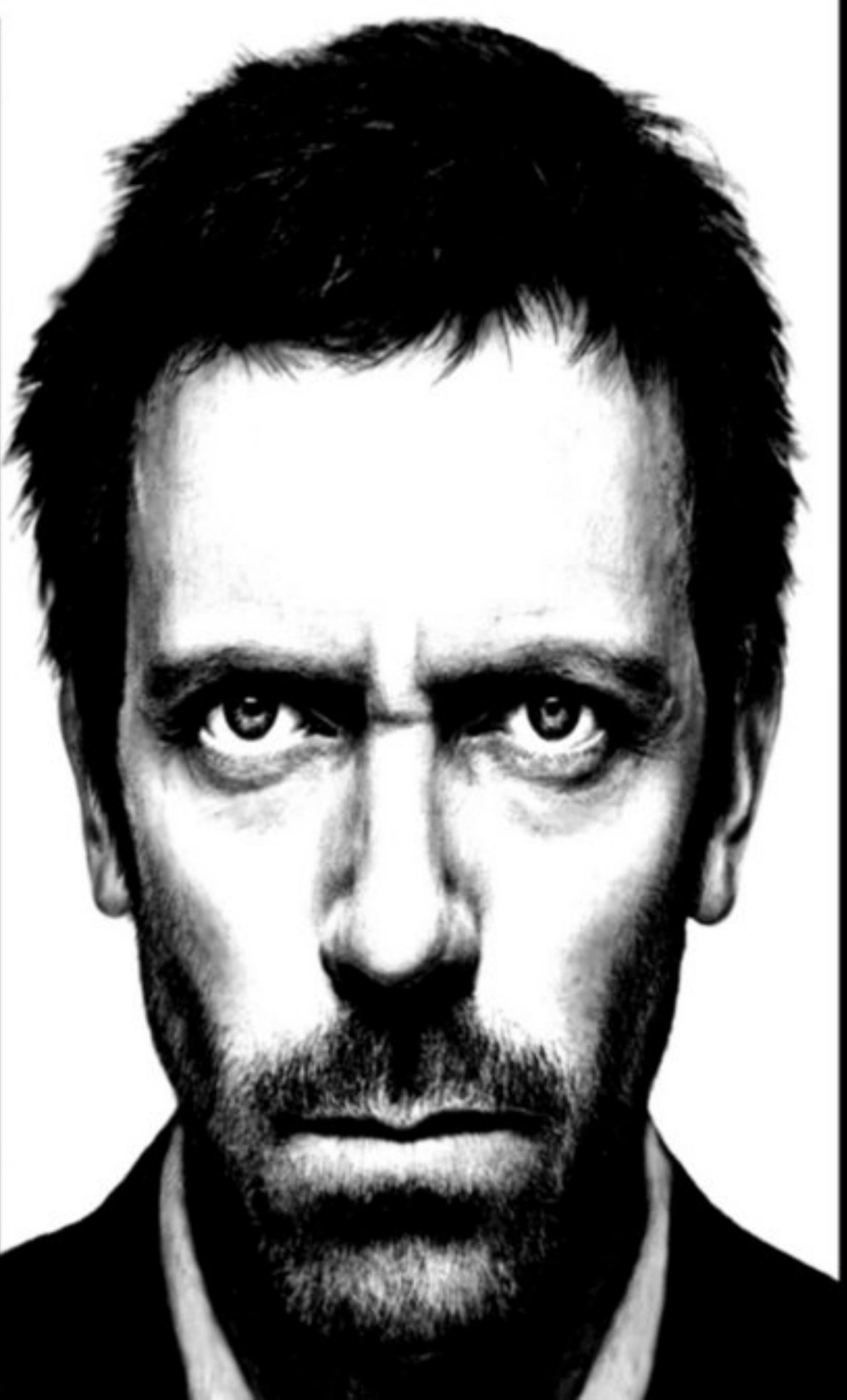


Example: Is HornetQ Broken?

large Datomic query

high CPU utilization

IllegalStateException in HornetQ



IT'S NEVER LUPUS

http://ambrohexal.deviantart.com/art/Its-never-Lupus-186073495

It's *Always* GC

OOM is typically unexpected

OOM can happen anywhere

OOM can appear as almost any other exception

near-OOM dramatically impacts scheduling

OOM related problems cascade

RTEFM

a bug starts as an “unknown unknown”

so that means read the ***entire*** manual

good docs (for debugging) are

short

specifications

“Partial”, Revisited

```
(def partial-join
  (partial (clojure.string/join ",")))
```

---

join
function

```
Usage: (join coll)
       (join separator coll)
```

Returns a string of all elements in coll, as returned by (seq coll), separated by an optional separator.

Added in Clojure version 1.2

Source

http://clojure.github.io/clojure/clojure.string-api.html

What Would You Do Next?

large Datomic query

high CPU utilization

IllegalStateException in HornetQ

happens on Cassandra (prod) but not on H2 (dev)

Test Cassandra vs. H2

create a test env with same data

run Test.java + Datomic Peer

Test.java starts & performs problem query in a loop

What We Learned

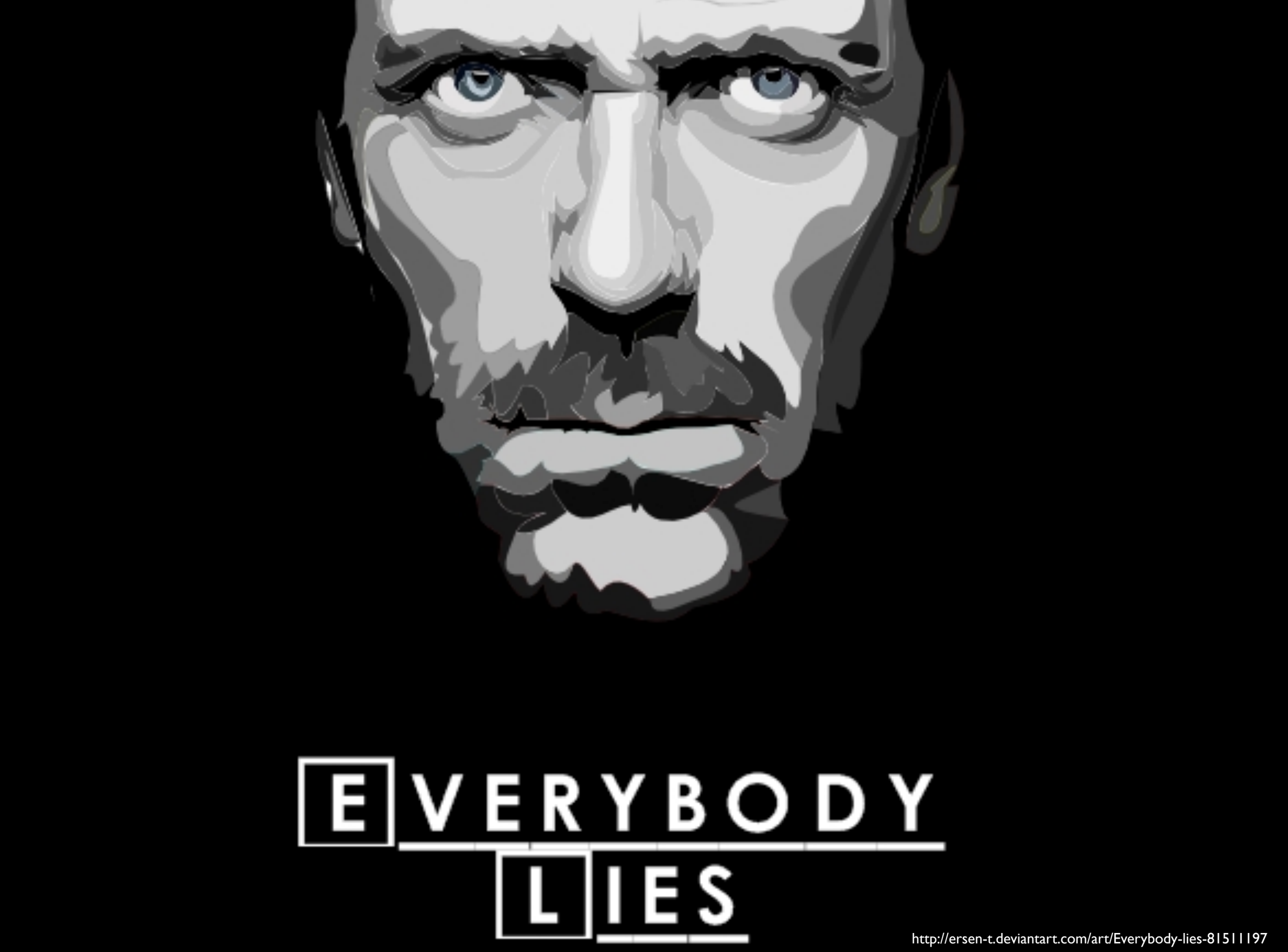
happens on Cassandra (prod) but not on H2 (dev)

trivial repro happens with Cassandra

surprise, it is lupus! (GC)

trivial repro also happens with H2

hmm…



http://ersen-t.deviantart.com/art/Everybody-lies-81511197

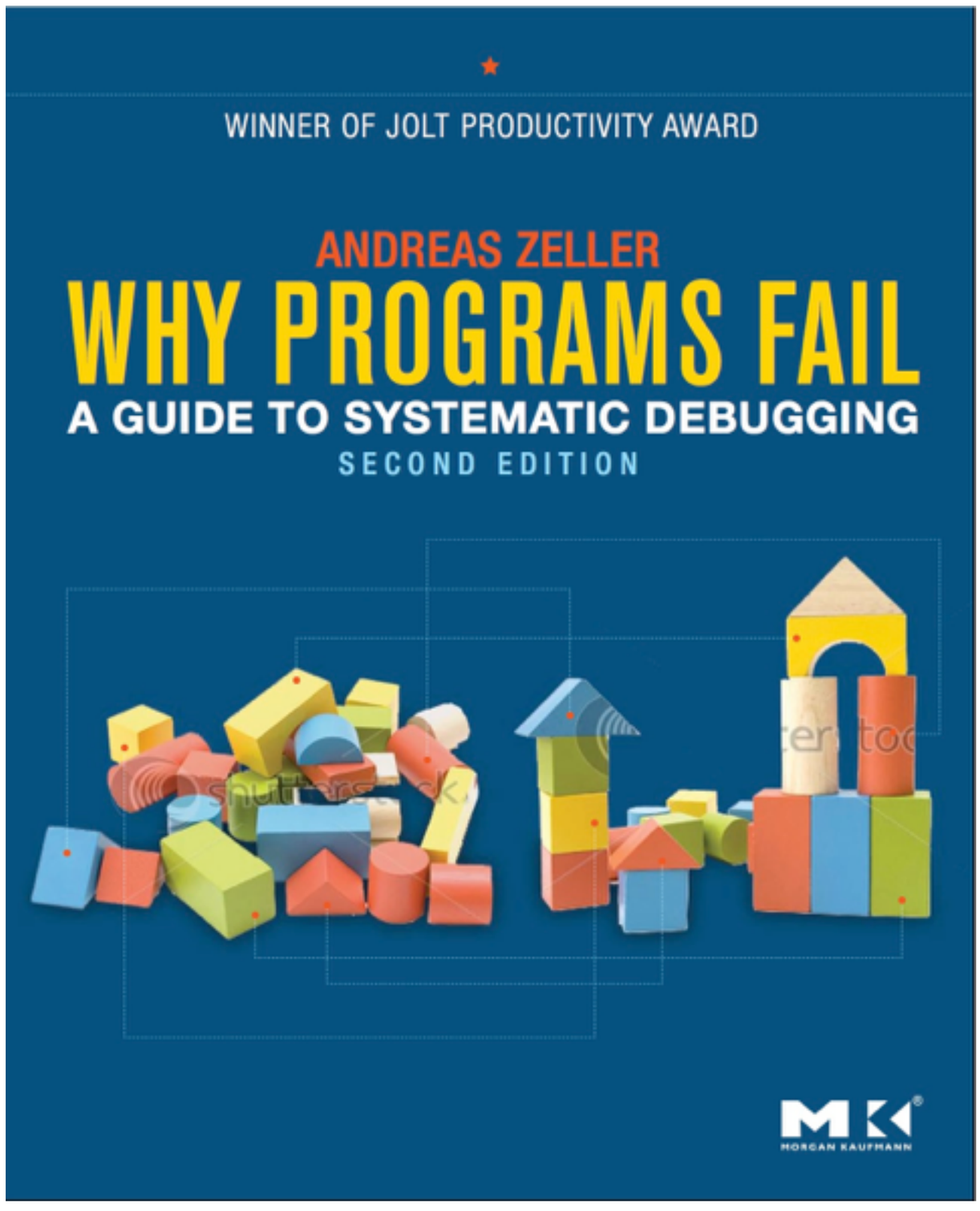
Search Automation

http://boundlessgeo.com/2013/07/finding-a-performance-regression-with-git-bisect/

Call to Action

read chapters 5-7, 11-14

delta debugging for Clojure



http://www.whyprogramsfail.com/

Software Specifics

work at a high level of abstraction

the fault is not the defect

lupus (*GC*) did it

RTFM * 2

don’t trust, reproduce

automate

Science Made Easy

know where you are going

make well-founded choices

write stuff down

Debugging with the Scientific Method

@stuarthalloway