

# **SHIFT LEFT WITH REVIEW CULTURE**

**Questions?**

Contact me at:

W: <http://redelastic.com>

E: [jason.goodwin@redelastic.com](mailto:jason.goodwin@redelastic.com)

# **KEVIN WEBBER. CEO, REDELASTIC**

# JASON GOODWIN. CTO, REDELASTIC

# REDELASTIC



- » Modernization
- » Enablers
- » Accelerators

# **AGENDA: SHIFT LEFT WITH REVIEW CULTURE**

- » Part 1: 45 min: Why Review
- » Part 2: 15 min: Provide for Review
- » Lunch break
- » Part 3: 45 min lab: Give Reviews

# **FROM STARTUP TO GOOGLE**

# **COMPROMISES IN A START-UP**

- » Intentional (market pressure)
- » Unintentional (bad outcomes)

# ACQUISITION

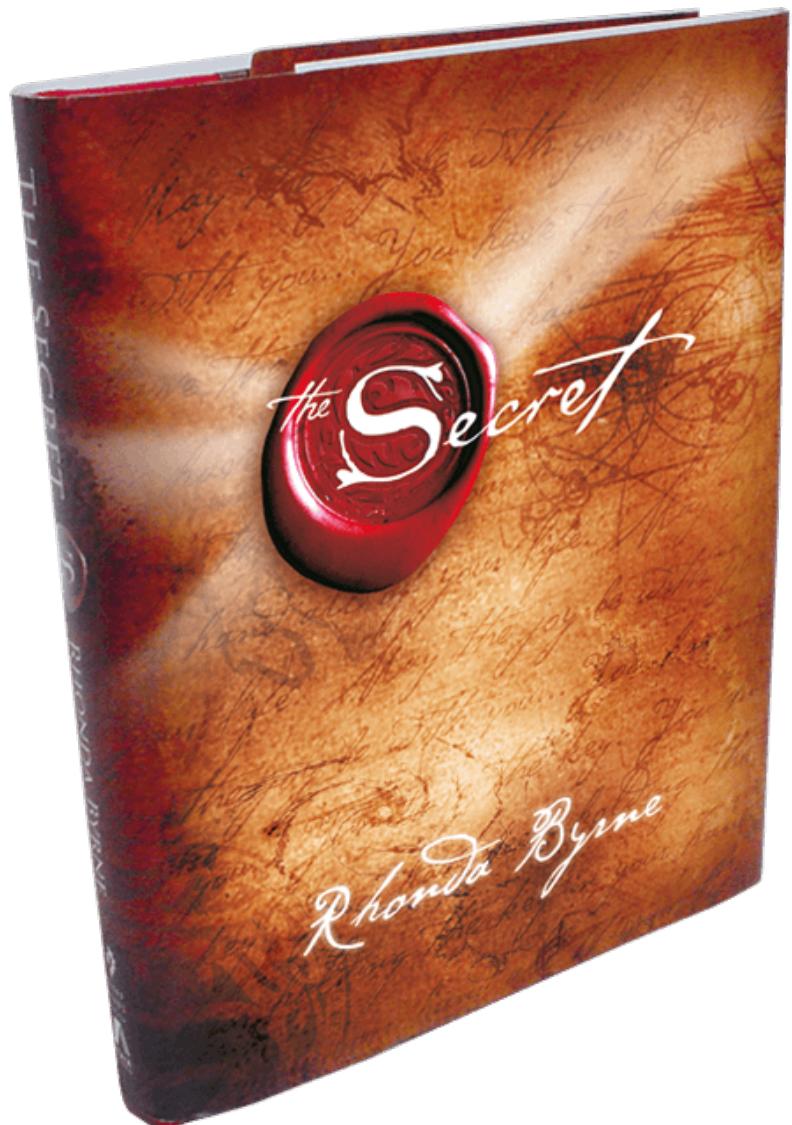
- » Operate inside a tech org with high standards
- » Leadership replaced by Googlers
- » QA removed

mDialog + Google

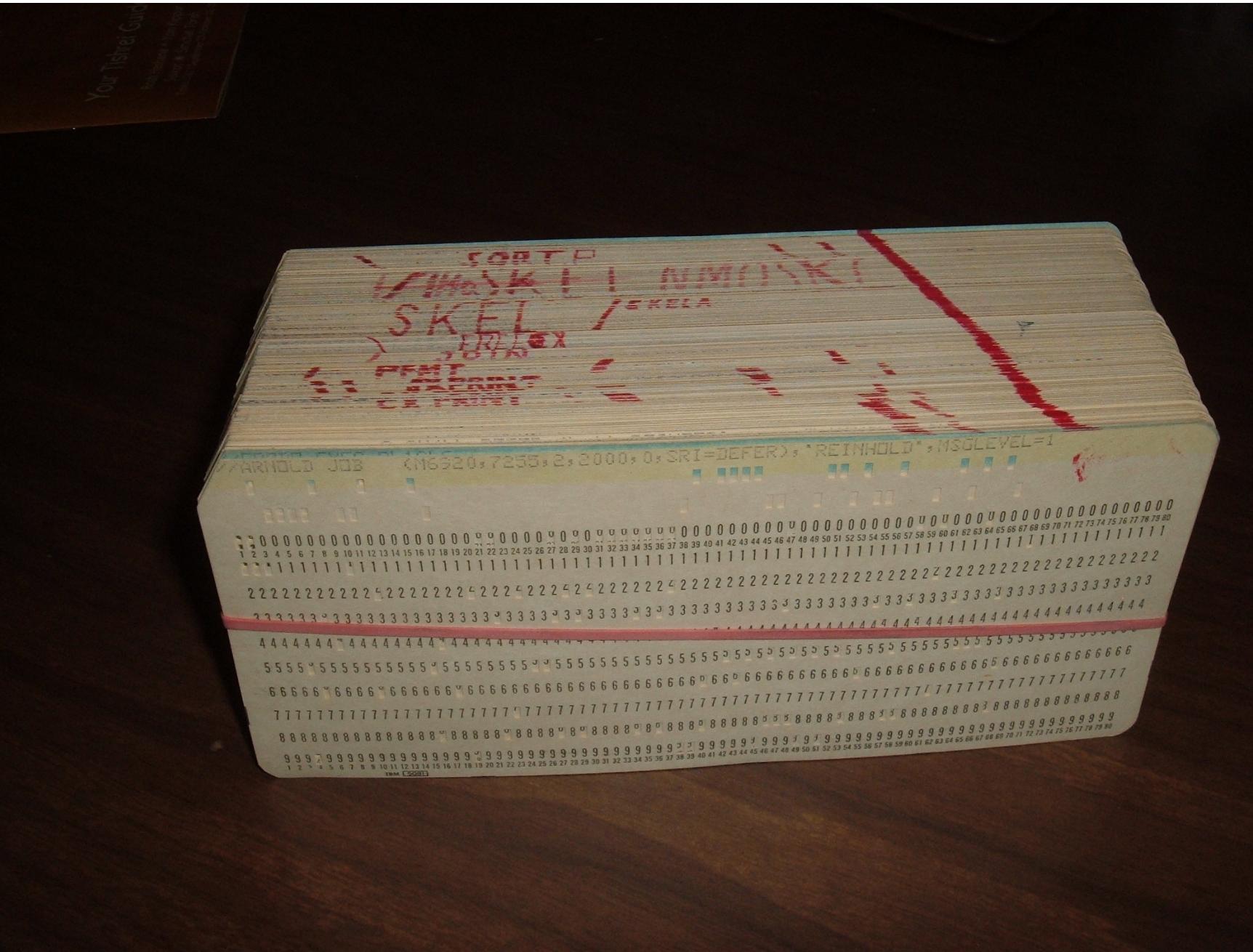
# **CODE PURPLE**

- » "Code Health"
- » Changes to code caused customer impact frequently

# WHAT I LEARNED WHILE MAKING SOFTWARE "GOOGLE STABLE"



# COMPLEXITY THEN <~1975



# COMPLEXITY NOW



Prius  
Hybrid



Lexus LS



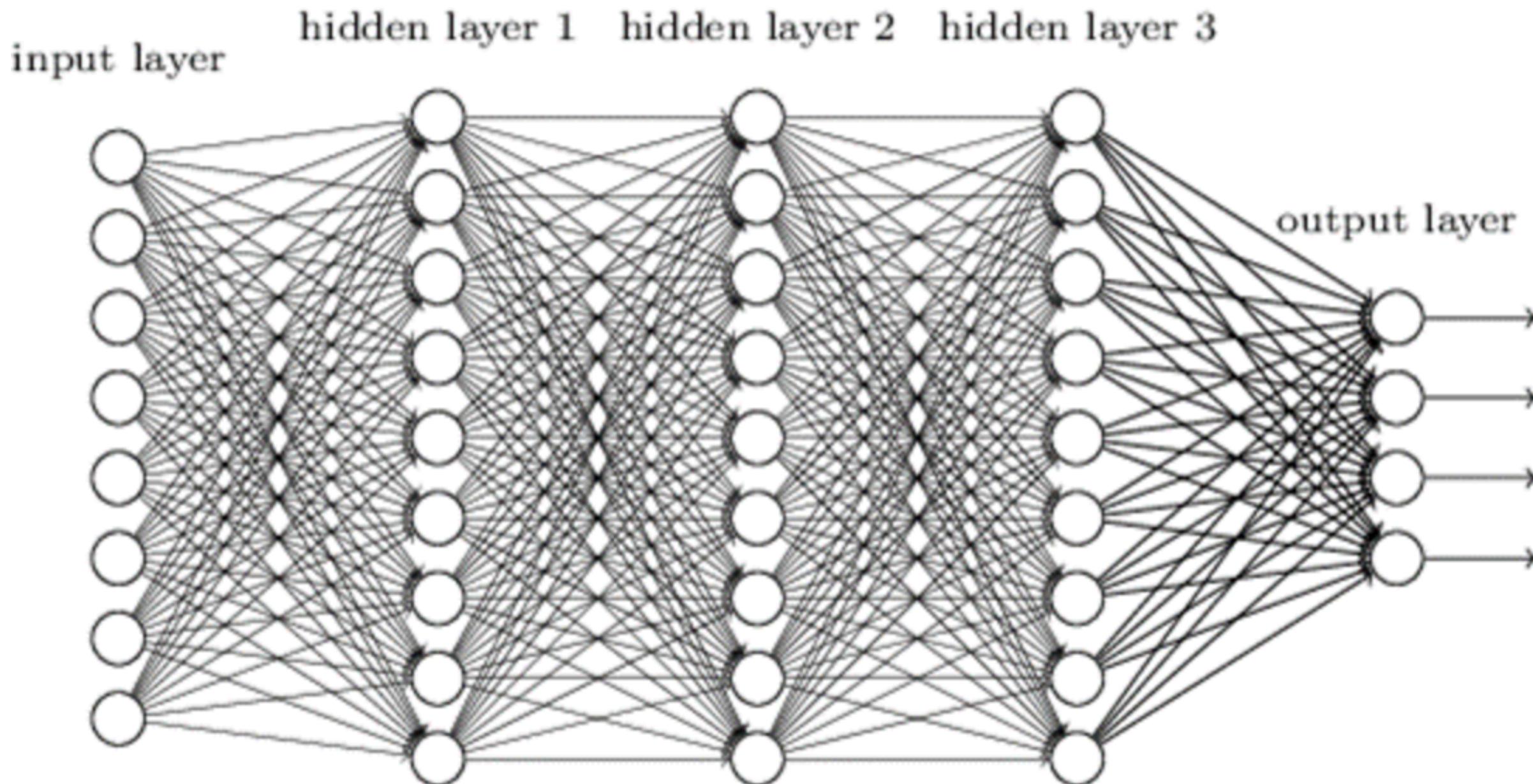
787



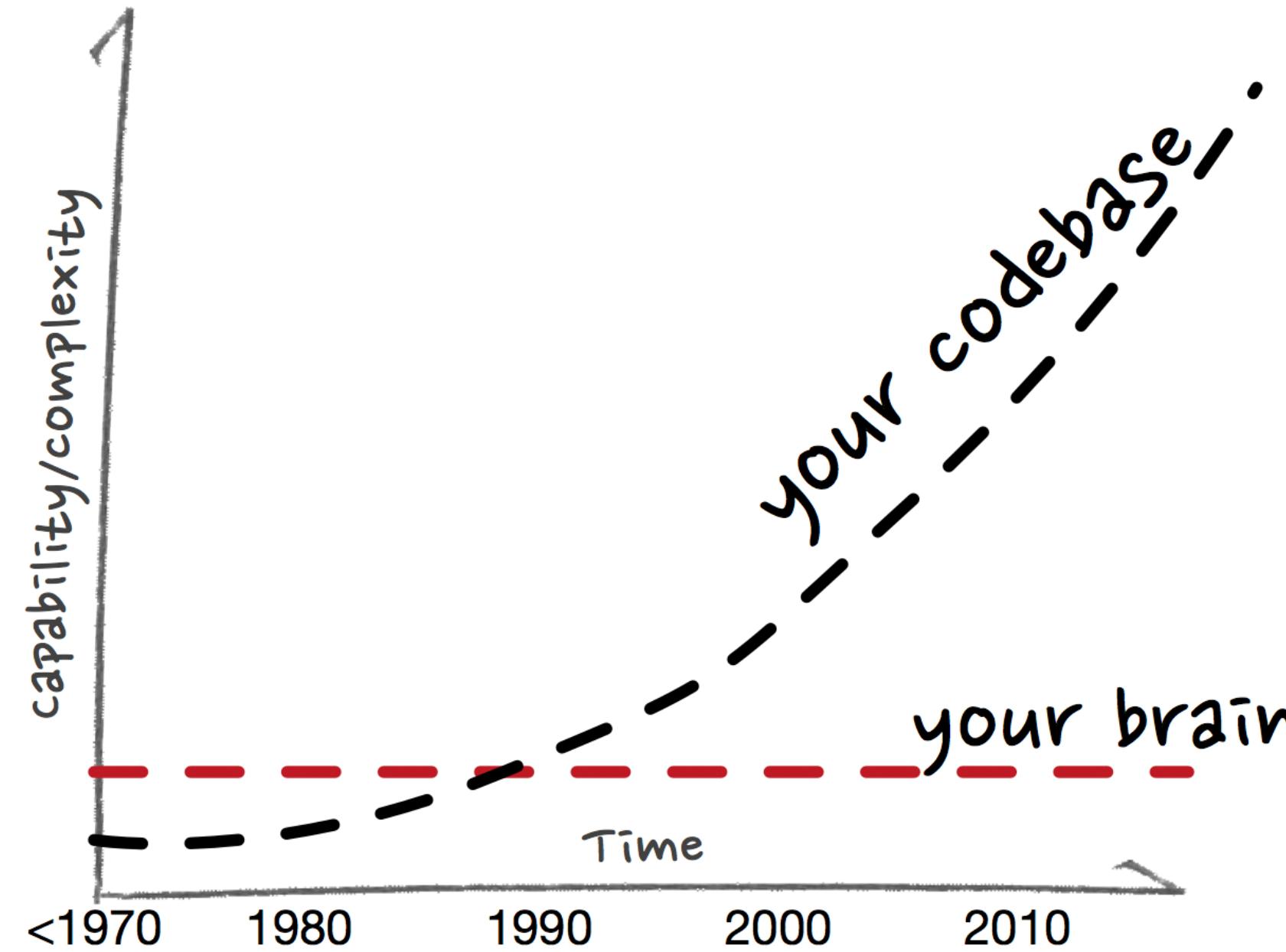
F-35



# SPECIALIZATION AND COMPLEXITY



# BRAIN COMPLEXITY OVER TIME

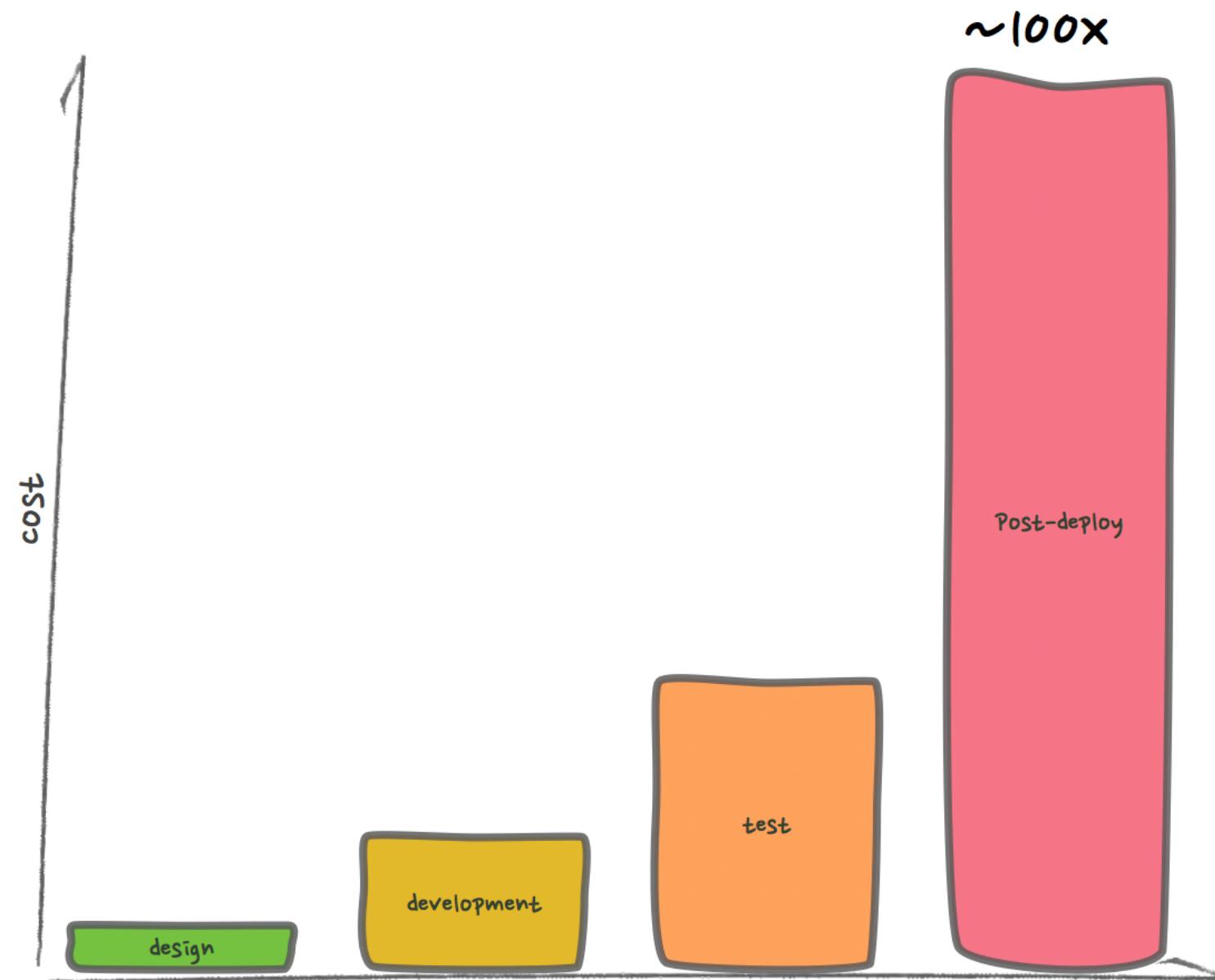


# THE SECRET

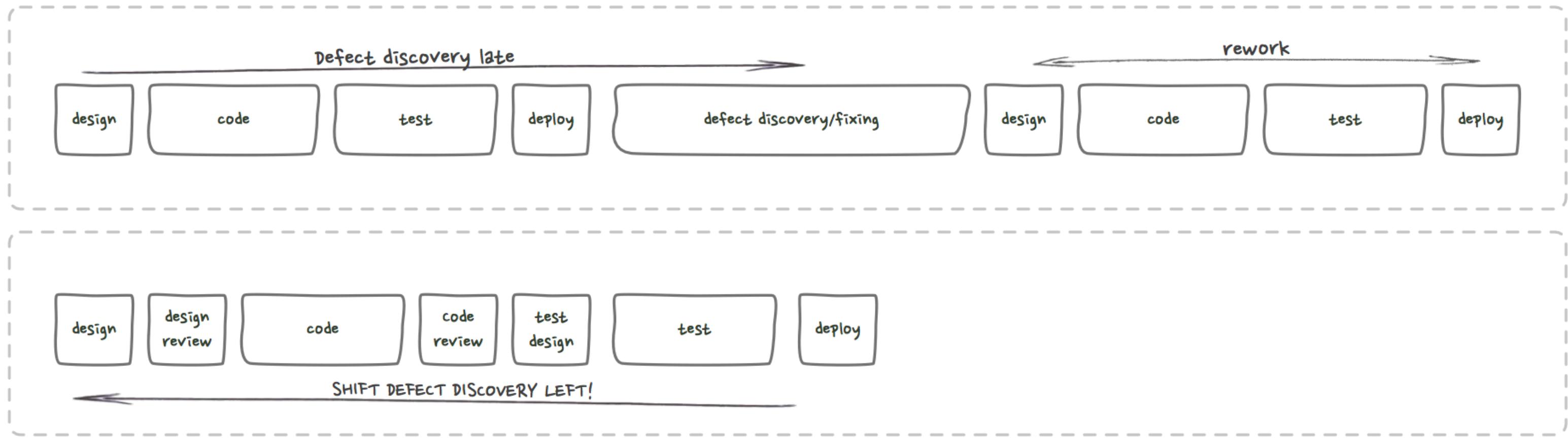
# **FALSE ECONOMIES OF SPEED**

“In economics, a false economy is an action that saves money at the beginning but which, over a longer period of time, results in more money being spent or wasted than being saved.”

# COST OF LATE DEFECTS (IBM SYSTEMS)



# "SHIFT LEFT"

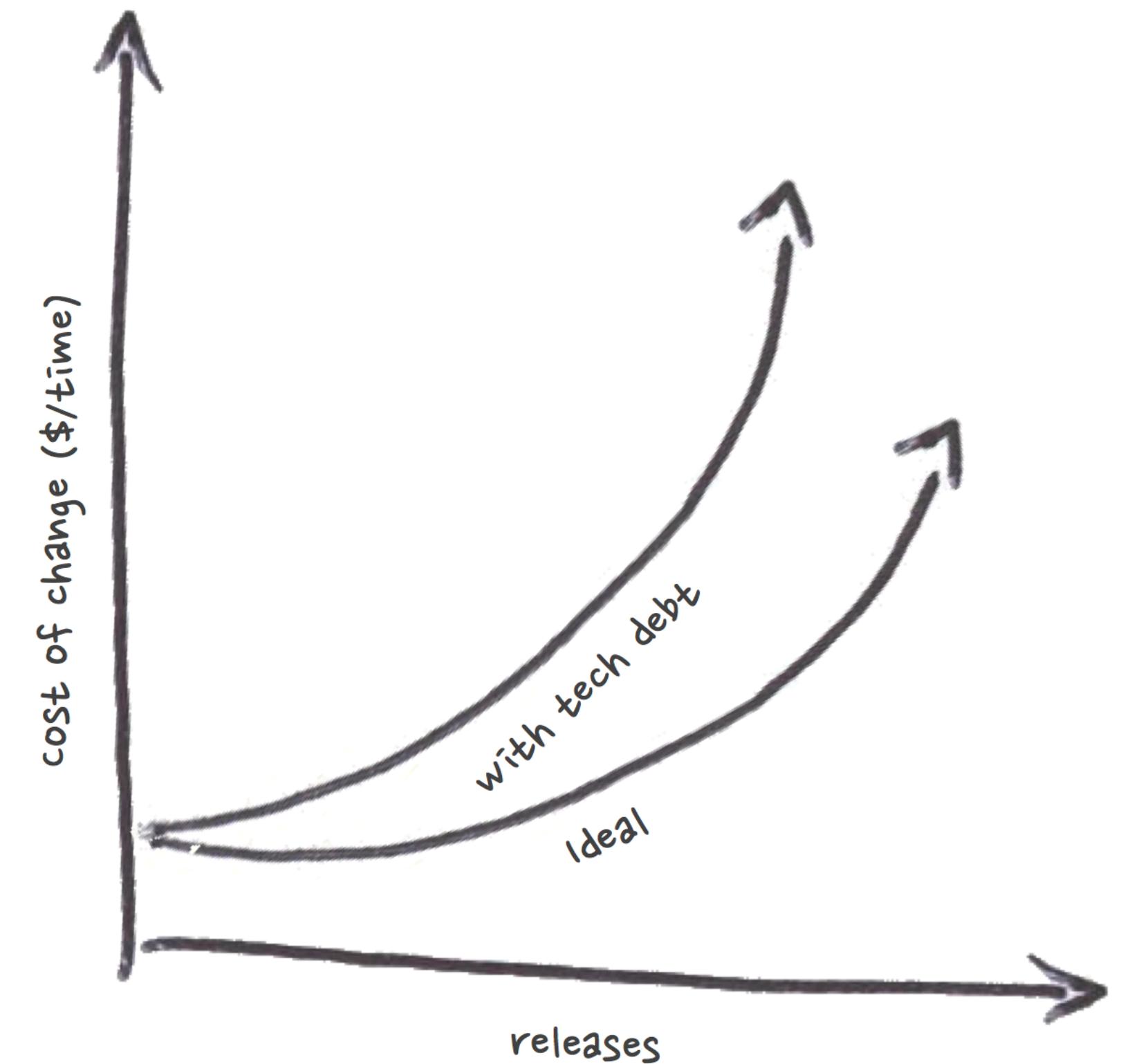


# **REVIEW ISN'T JUST CODE**

- » Each Step of a Feature's Life
  - » Design Documents
  - » Code
  - » Test Plan
  - » Release Plan
  - » Failures ("Blame-free PostMortem")

# CARRYING COST OF POOR CODE HEALTH

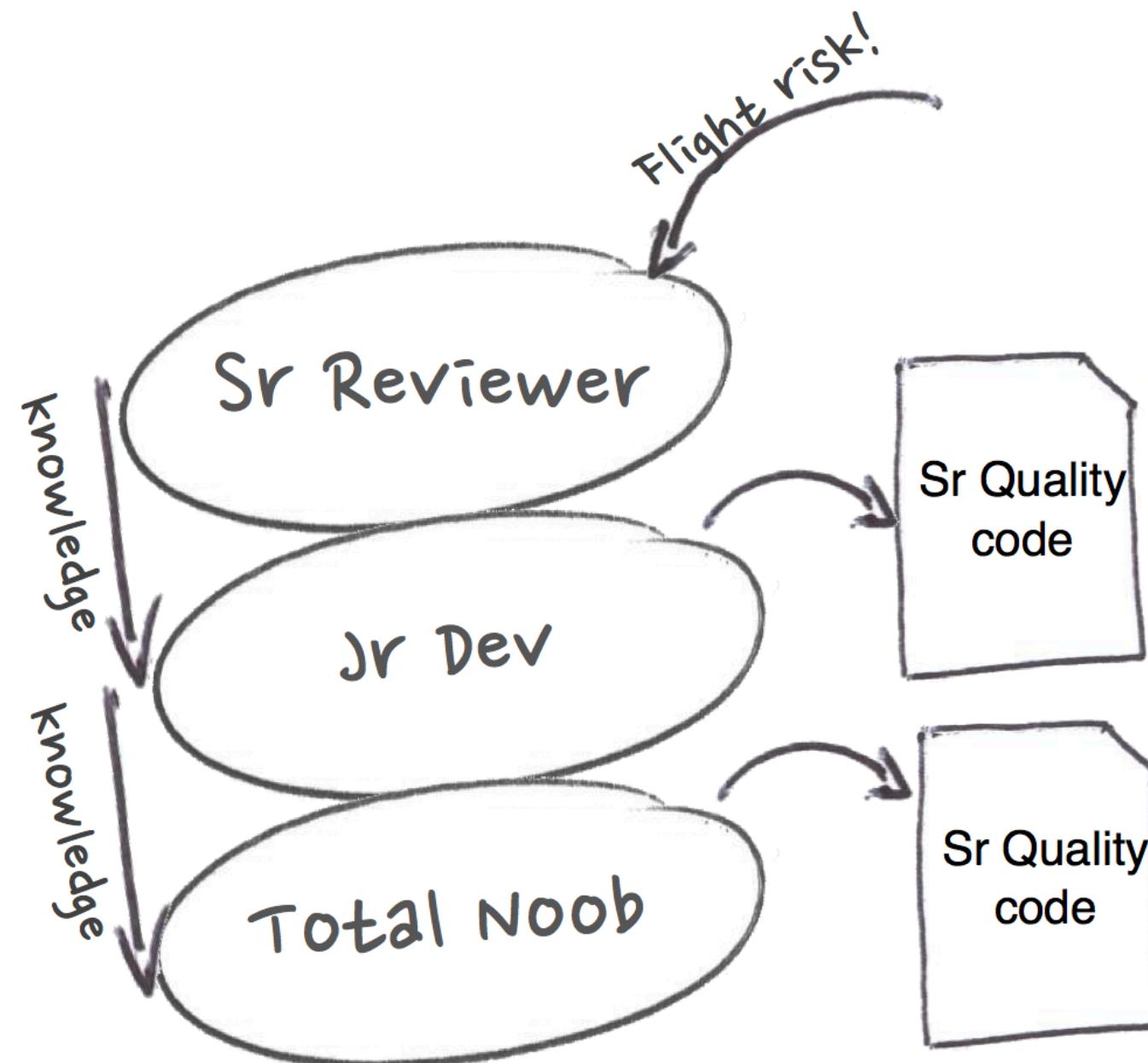
“In retrospect, I would have done it differently. Just release it - we'll refactor it later.”



# **OTHER POSITIVE EFFECTS**

**“It's not just for breakfast anymore!”**

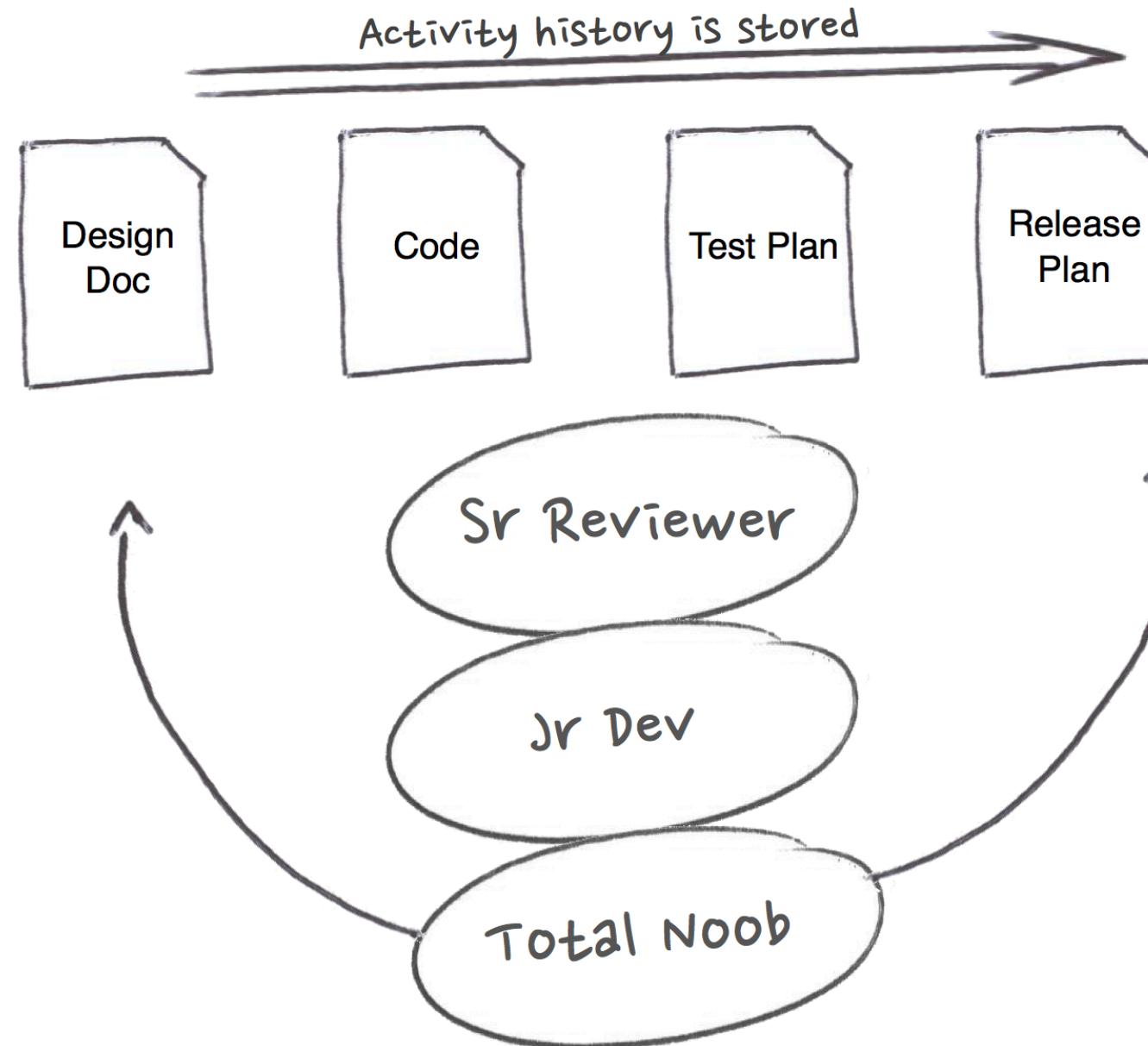
# NORMALIZING KNOWLEDGE



# **EGO FACTOR**

“I Know someone will review my docs, code...  
I better check it myself.”

# PERFORMANCE EVIDENCE



# **PART 2: HOW TO PROVIDE FOR REVIEW**

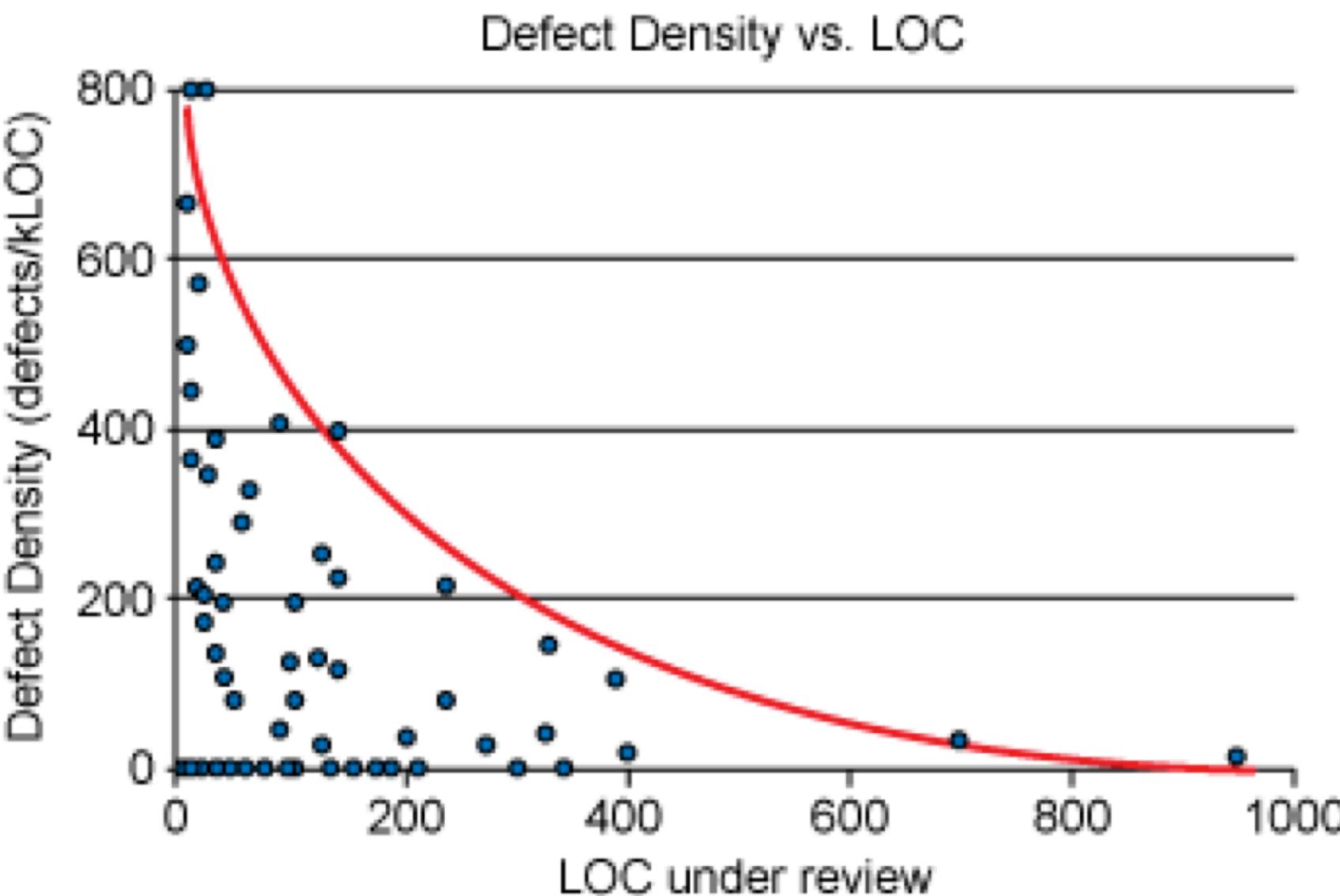
# EFFICIENCY (INPUTS VS OUTPUTS)

- » Input Quality
  - » design docs
  - » change sets
  - » hours in review
- » Affects Outputs Quality
  - » defects found
  - » quality improvement

# INTENTION

- » Code's intention should be clear
- » Describe in commit msg & ticket
- » Problem, solution

# SIZE



- » Most important metric
- » Beyond 400 LOC, defect detection drops off

# DESIGN DOCS

- » Design docs should undergo their own review
- » Small docs for small changes
- » Change Set -> Ticket -> Design Doc

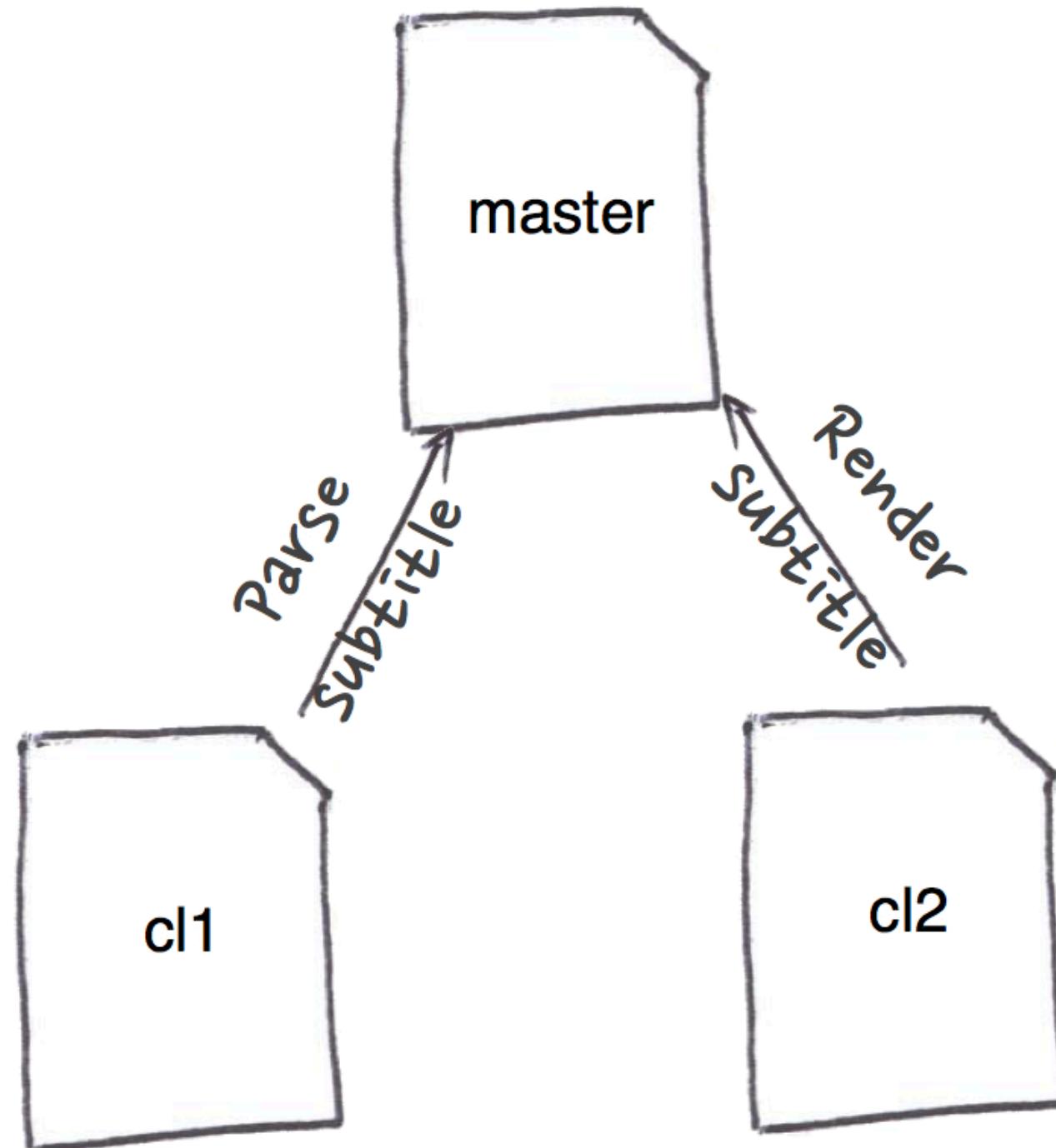
# TESTS

- » Tests are documentation
- » Behavioral changes described in tests

# **SELF REVIEW, ANNOTATE**

- » Review your code before providing
- » Leave guiding comments

# ATOMIC CHANGESETS



- » Make small changes against the root
- » Use config/feature toggles
- » Review and merge before stacking

# **PART 3: LAB: HOW TO REVIEW**

# SPEED

» Less than 500 LOC/hour

# TIME

» Less than 60 minutes at a time

# ATTITUDE

- » Foster Positivity
- » Make suggestions, not commands
- » Ask clarifying questions

# **ASSUME I'M A NOOB**

# HOW WE WILL WORK

- » Create a google doc
- » Copy the PR (if you like)
- » Leave comments in google docs
- » Share with jason.goodwin@redelastic.com
- » All code in Java w/ Play Framework

# MY EXPECTATIONS

- » Jr point out bugs
- » Intermediate point out code design
- » Sr point out system design

# STRUCTURE

- » Clear intention
- » Code smells
  - » <https://sourcemaking.com/refactoring/smells>
- » Too large

# **PART 1 - BEFORE REVIEW WARMUP**

**goo.gl/eDt0dY**

- » Don't try to compile this one
- » Focus on general review feedback

# PART 1 - DEBRIEF

- » Speculative generality
- » Too large, multiple intentions
- » Lack of unit testing
- » Insane packaging
- » Doesn't compile

# CLARITY

- » Should be clear history of the change
- » Things should be named logically
- » General design

# TACTFUL FEEDBACK

- » Make suggestions, not commands
- » Ask clarifying questions, expect better code in response

# **PART 2 - CLARIFY**

**goo.gl/Yj19wB**

- » Design doc provided (see readme)
- » Share your doc with [jason.goodwin@redelastic.com](mailto:jason.goodwin@redelastic.com)

# PART 2 - DEBRIEF

- » unclear class names
- » bad test names
- » no error handling
- » missing log level TRACE
- » API calls GET to set the level.
  - » why is it /level?

# **BE CRITICAL TOO**

- » Ask questions of the design
- » Be critical, think through edge cases

# **PART 3 - JUNIOR'S SUBMISSION**

**goo.gl/uLH1bs**

- » Design doc provided (see readme)
- » Share your doc with [jason.goodwin@redelastic.com](mailto:jason.goodwin@redelastic.com)

# **PART 3 - DEBRIEF DESIGN**

- » in memory - should be in store for HA
- » assumes sticky sessions possible

# PART 3 - DEBRIEF API

- » Not RESTful API
- » Incorrect response codes
- » Incorrect Response Type

# PART 3 - DEBRIEF CODE

- » Commit message not clear
- » multiple threadsafety issues
- » global state - should have dependency injection of the behavior
- » ttl is hardcoded - should use DI or read from config

# **PART 3 - DEBRIEF DESIGN**

» Why would we do it in memory?