goto; conference

When and how to explore: an engineer's guide

Julie Pitt (@yakticus)

goto; conference

Click ‘Rate Session’ to rate session and ask questions.

Chapter 1

The Founding

Julie
Pitt

Greg
Orzell

Christian
Kaiser

Order of Magnitude Labs founding team

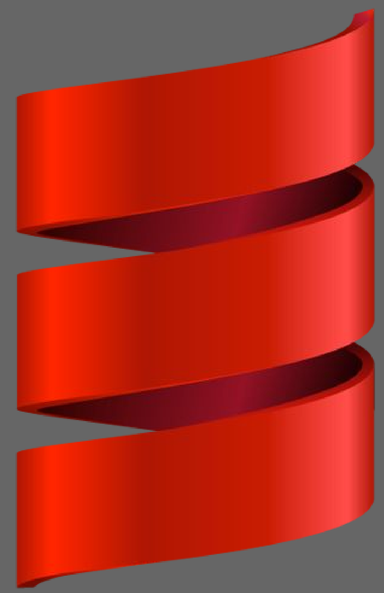
the
frontier

intelligent
machines

the business hypothesis

why prototype in Scala?

- functional
- immutable
- type system
- industrial scale libraries



prototype #1

- distributed, async neural net
- actor = “neuron”
- message passing between neurons

Just because you **can** do it doesn't mean you **should**.

prototype #2

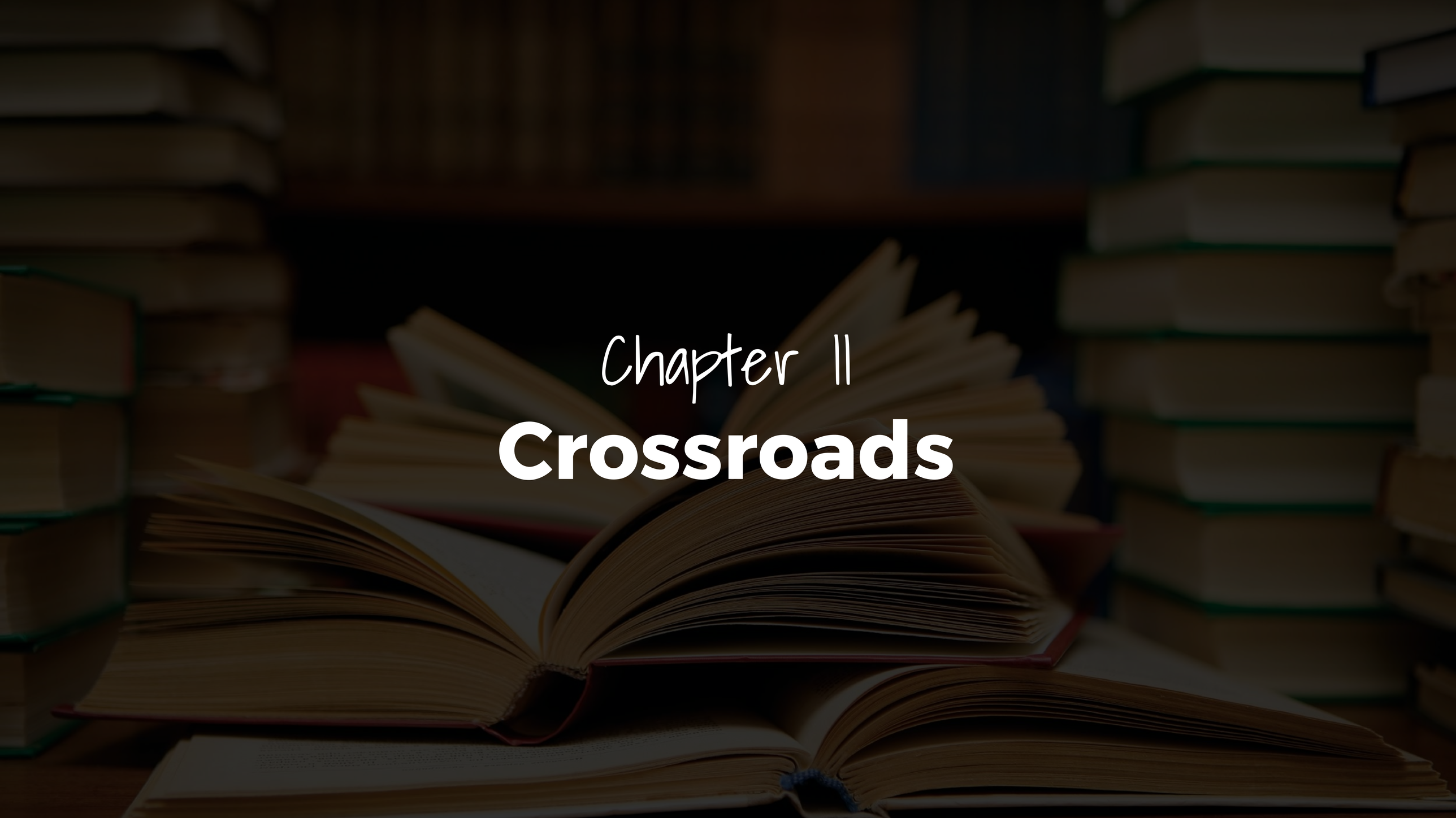
- embarrassingly simple
- immutable classes
- pure functions
- single threadded

https://www.youtube.com/watch?v=ddCgUOJqE64

https://flickr.com/photo/13547802@N05/8907414305/

perception

reality



Chapter 11

Crossroads

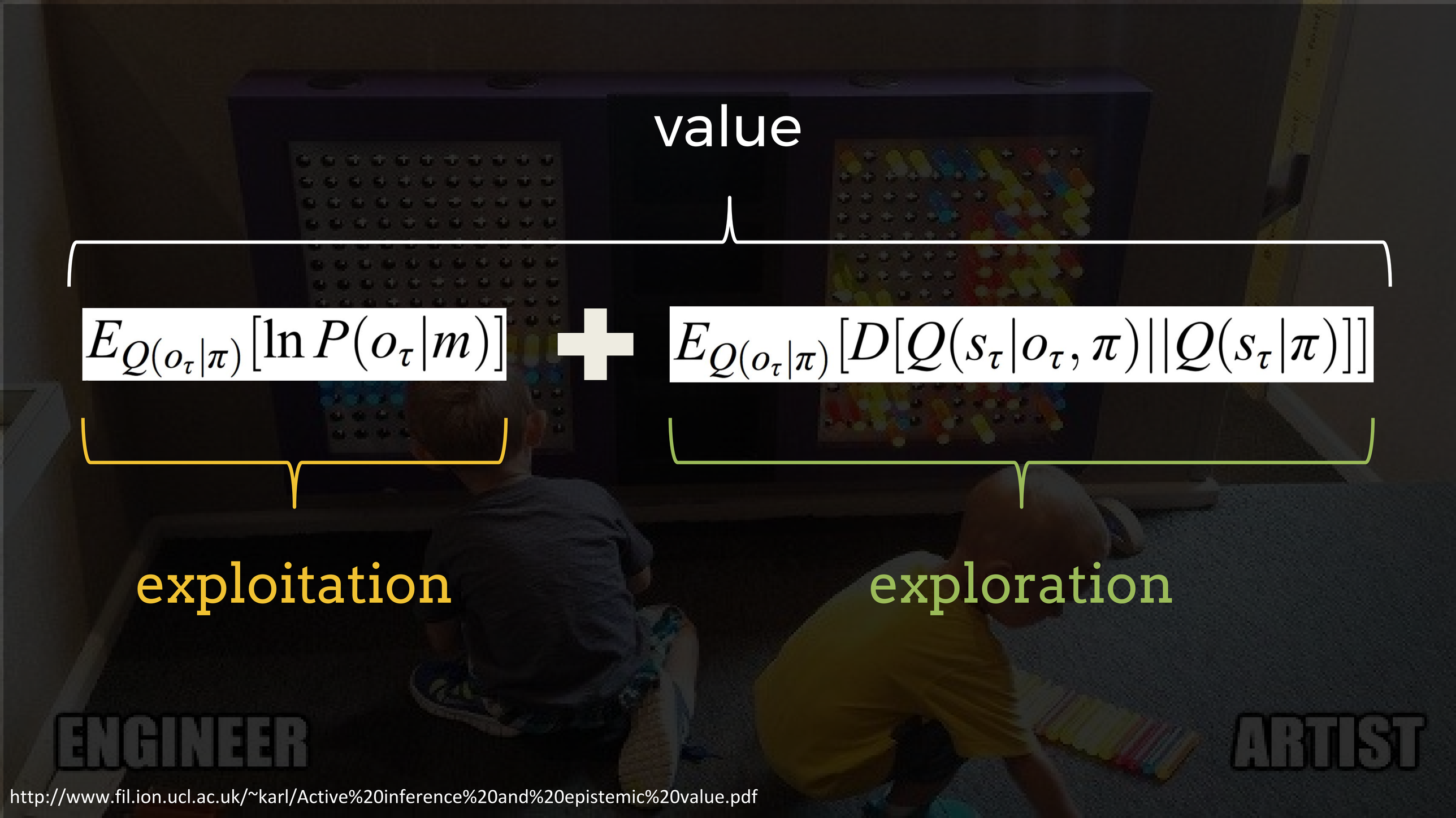

ENGINEER
ARTIST

value

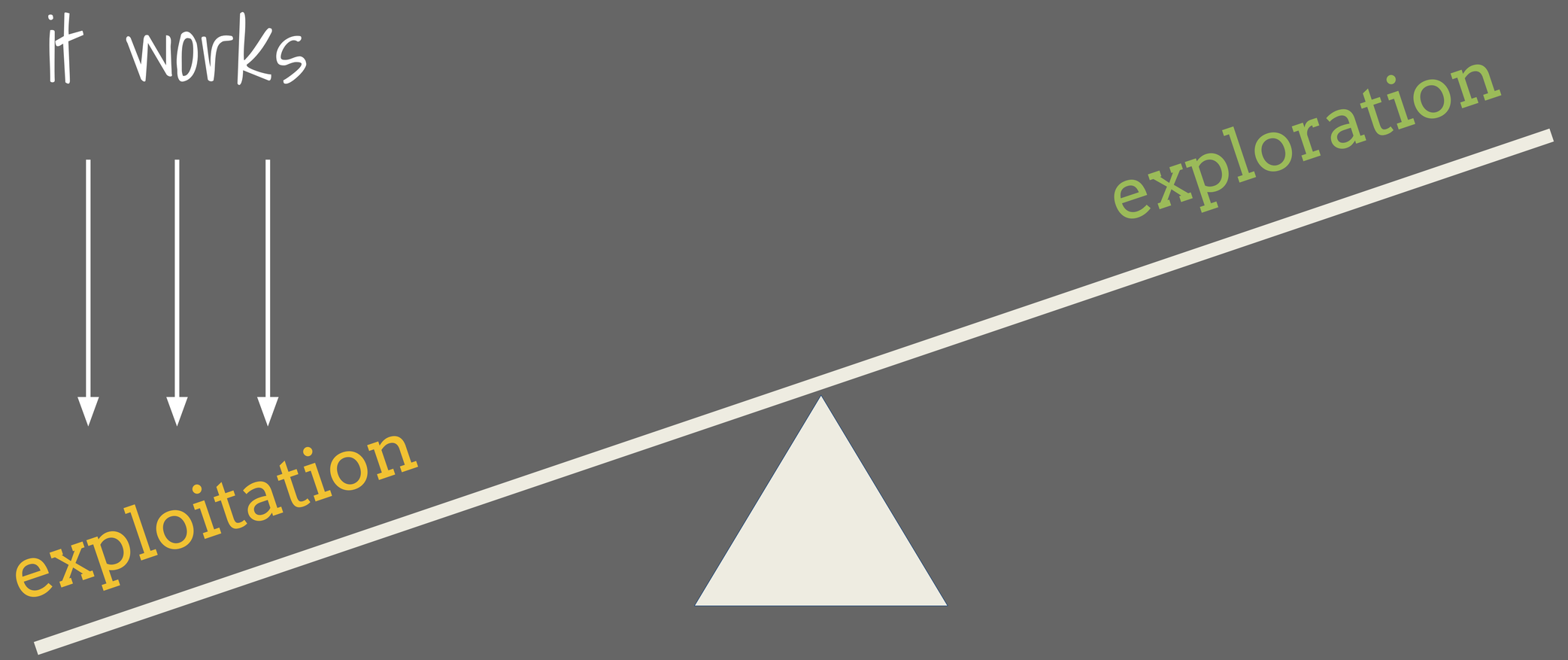
$$E_{Q(o_\tau|\pi)}[\ln P(o_\tau|m)] + E_{Q(o_\tau|\pi)}[D[Q(s_\tau|o_\tau,\pi)||Q(s_\tau|\pi)]]$$

exploitation

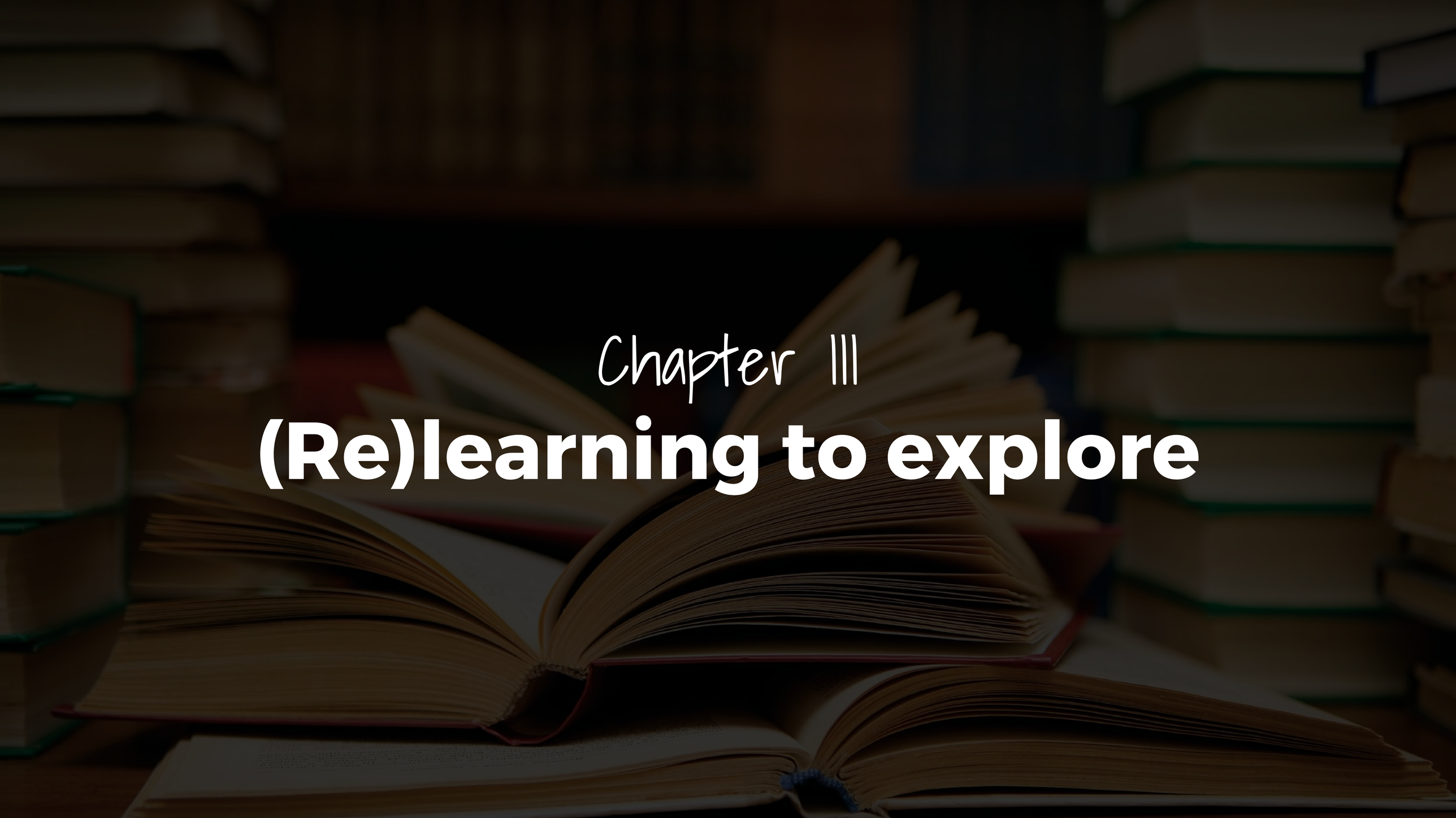
exploration



http://www.fil.ion.ucl.ac.uk/~karl/Active%20inference%20and%20epistemic%20value.pdf



why exploration is hard



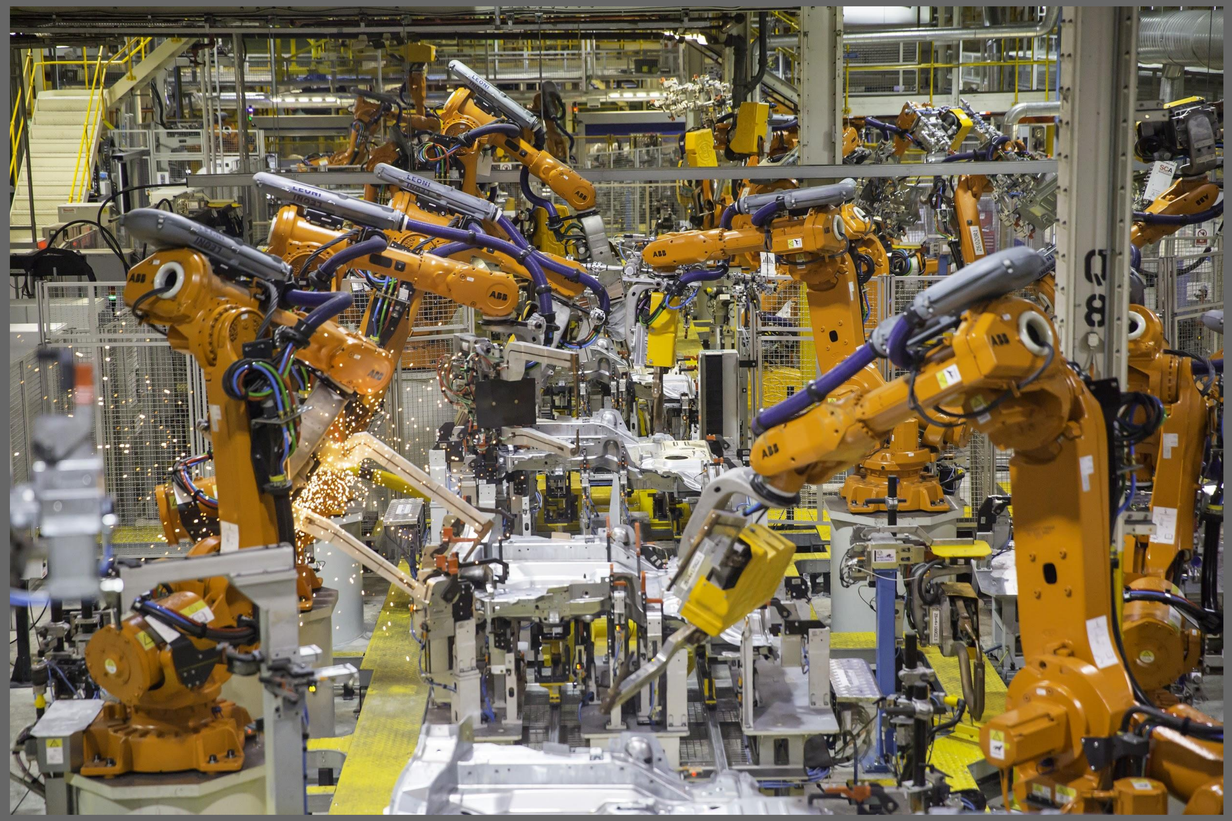
Chapter III

(Re)learning to explore


engineering best practices

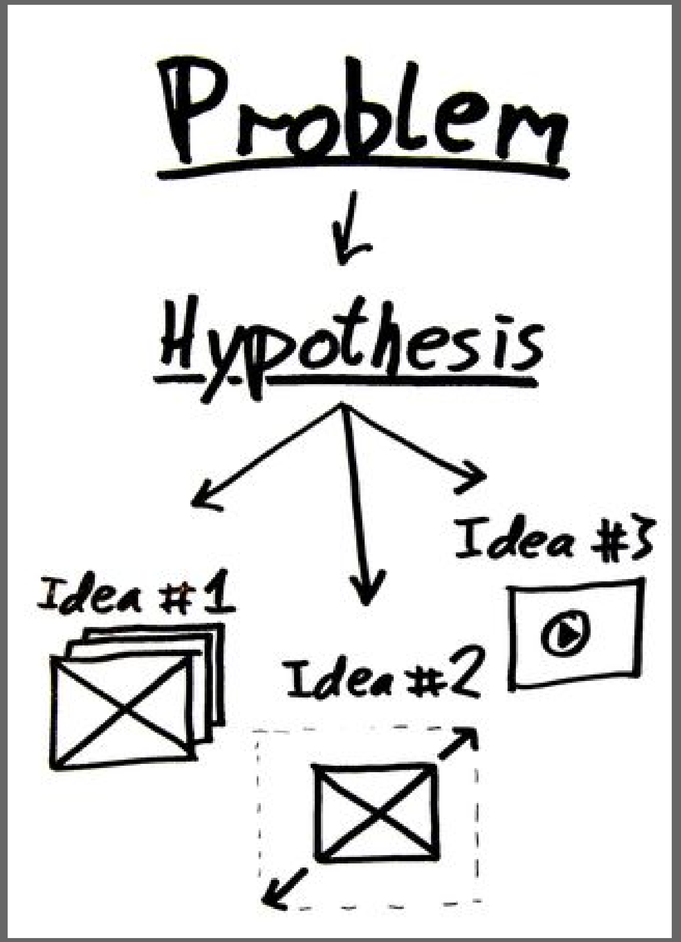
I was optimizing for...

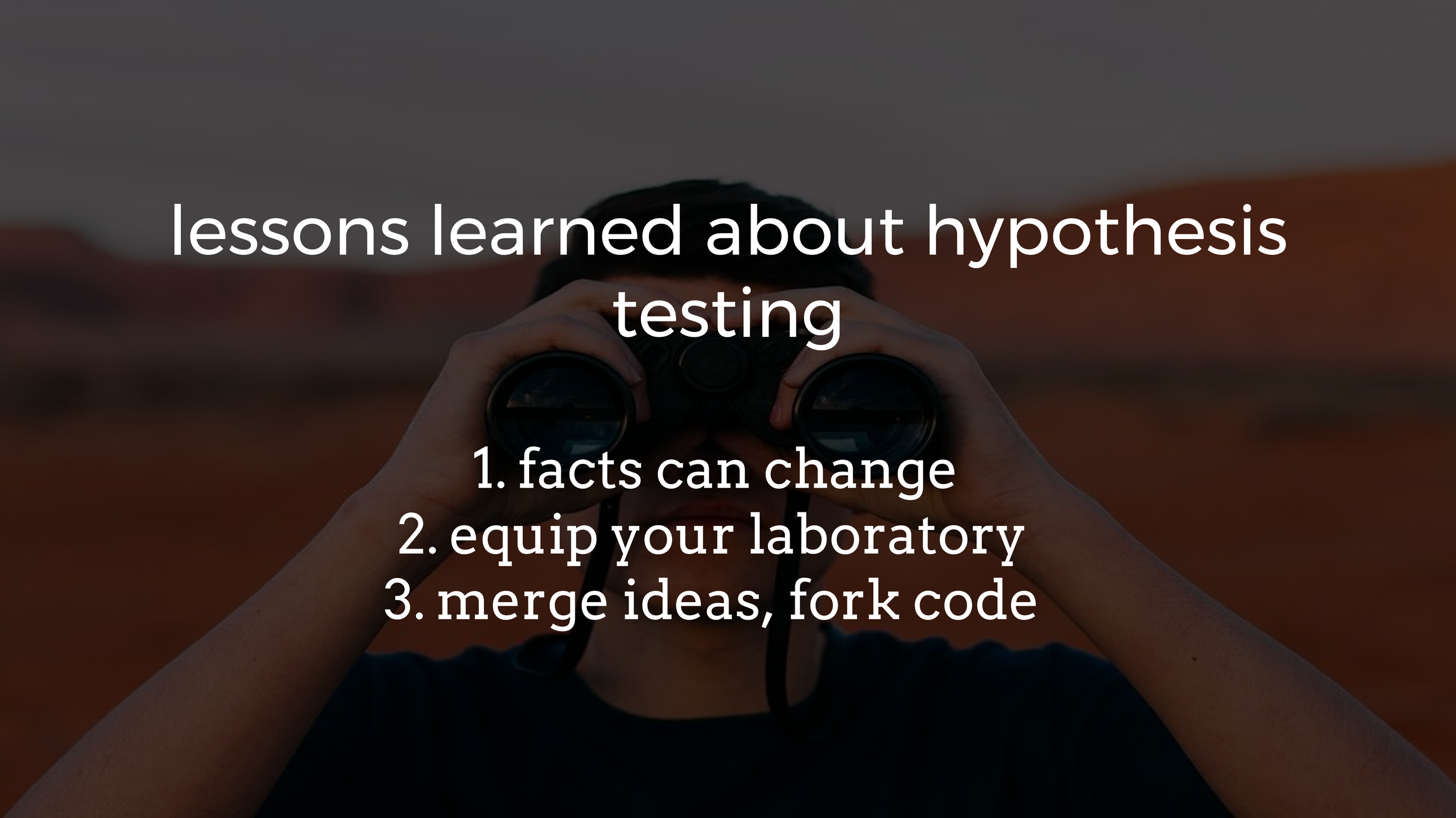
- building to spec
- shipping code
- operations
- minimizing tech debt



I should have optimized for...

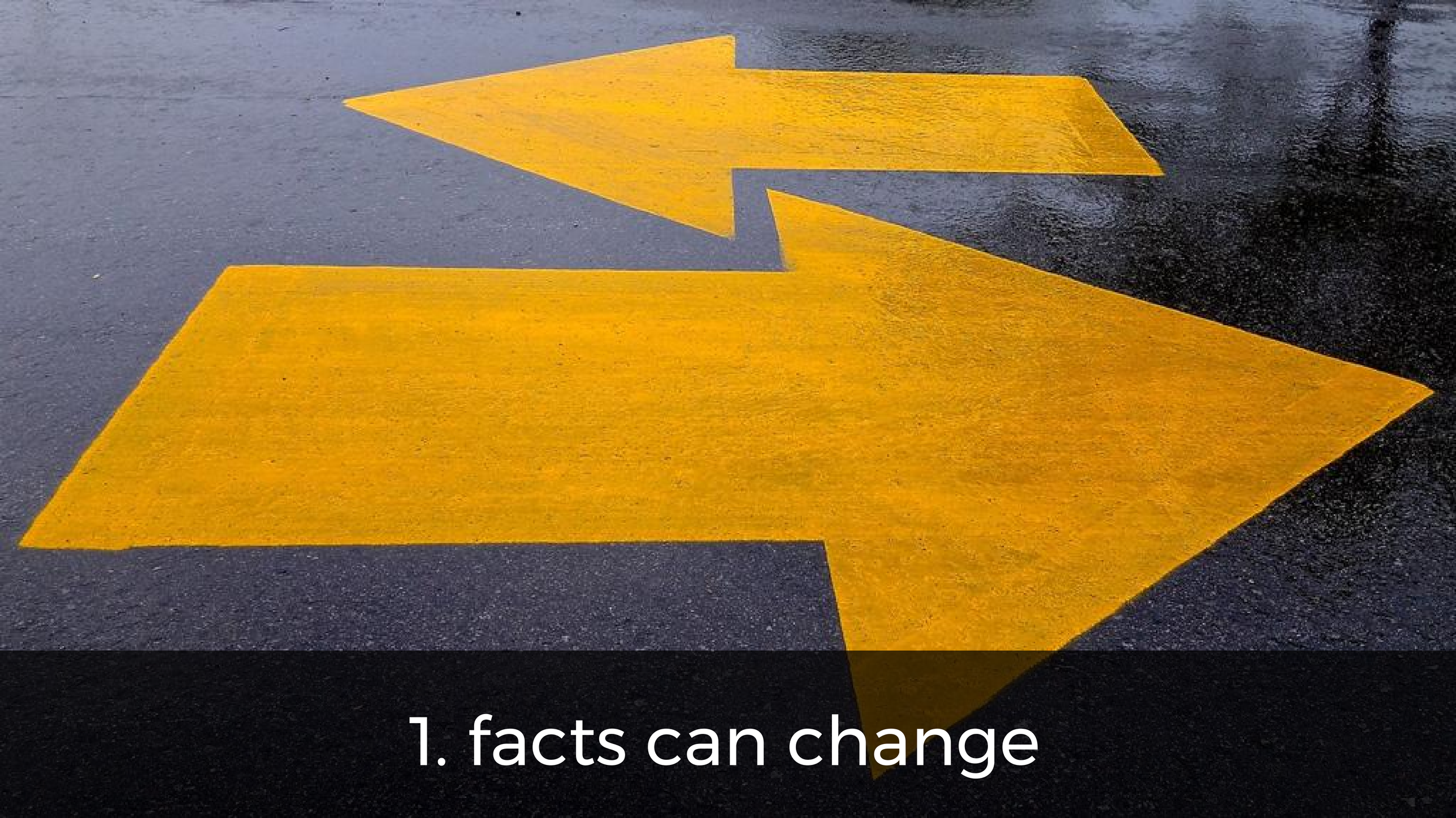
- formulating hypotheses
- ...testing them
- ...rejecting bad ones
- repeat!



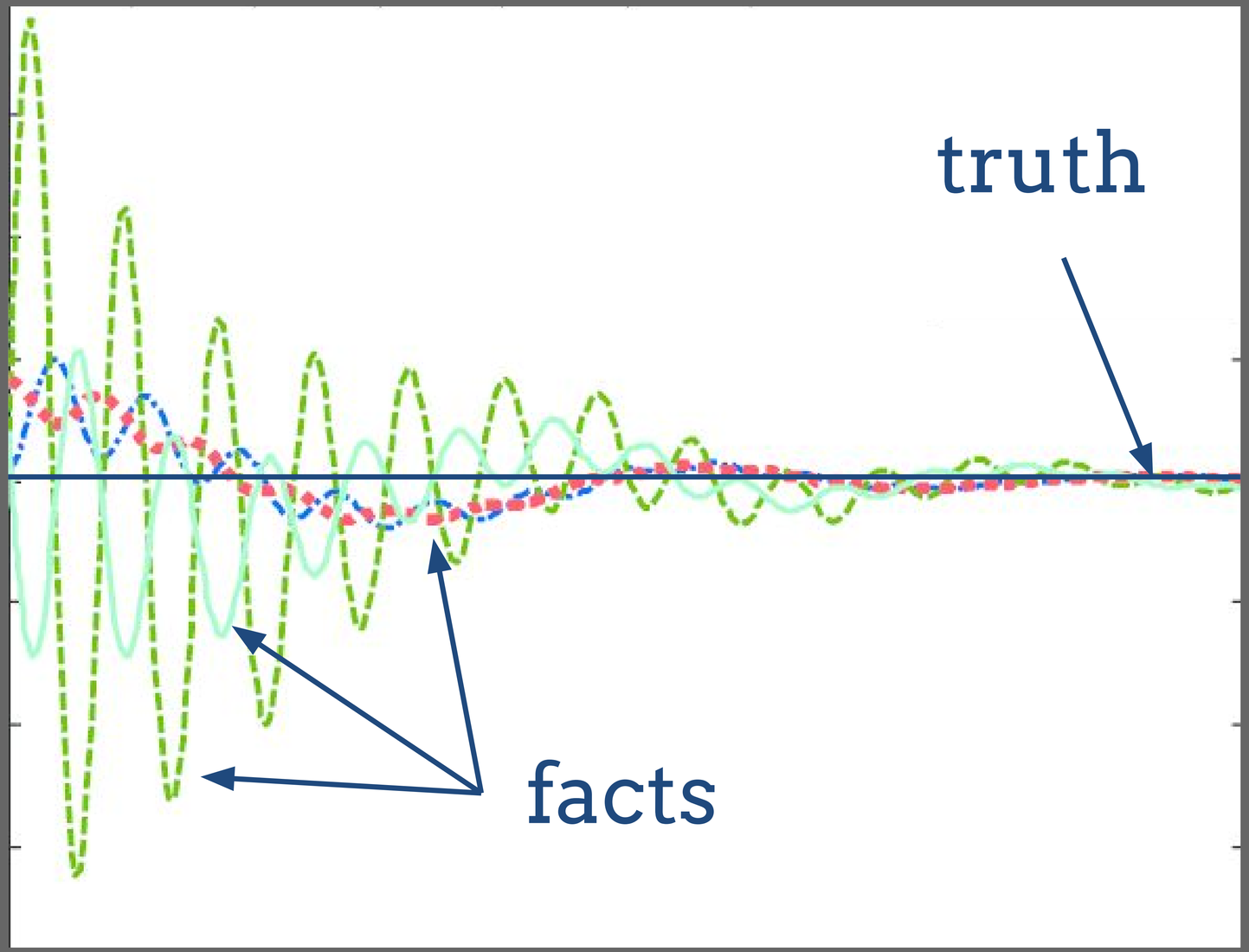


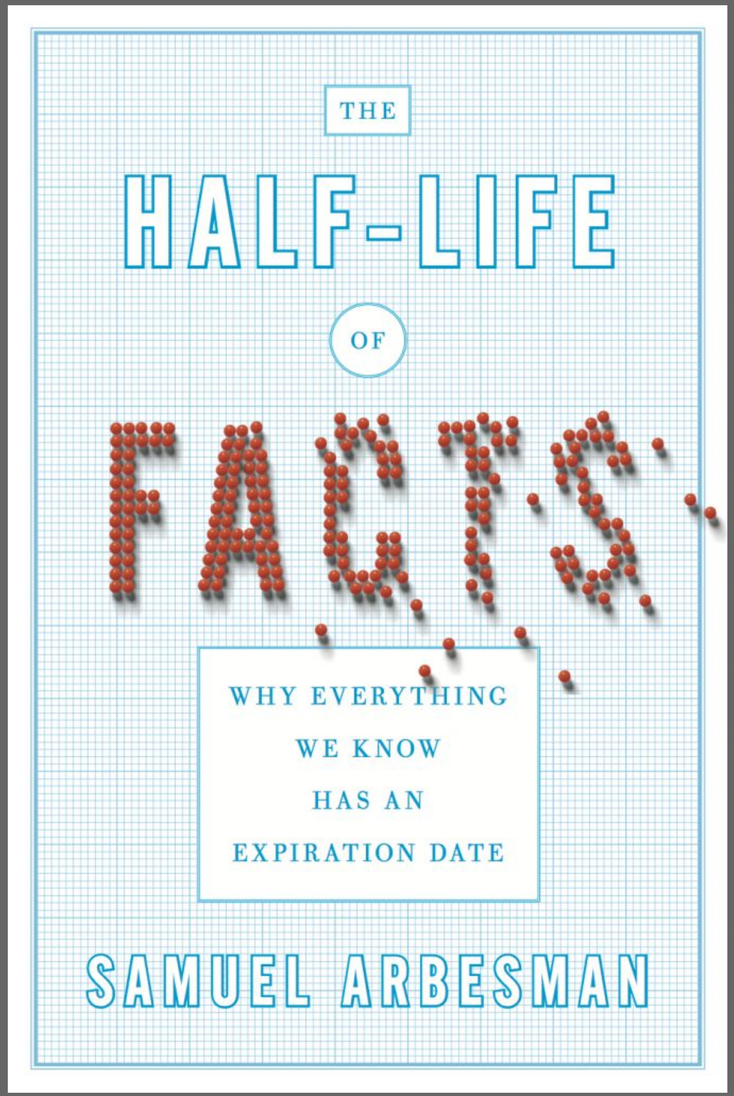
lessons learned about hypothesis testing

1. facts can change
2. equip your laboratory
3. merge ideas, fork code



1. facts can change

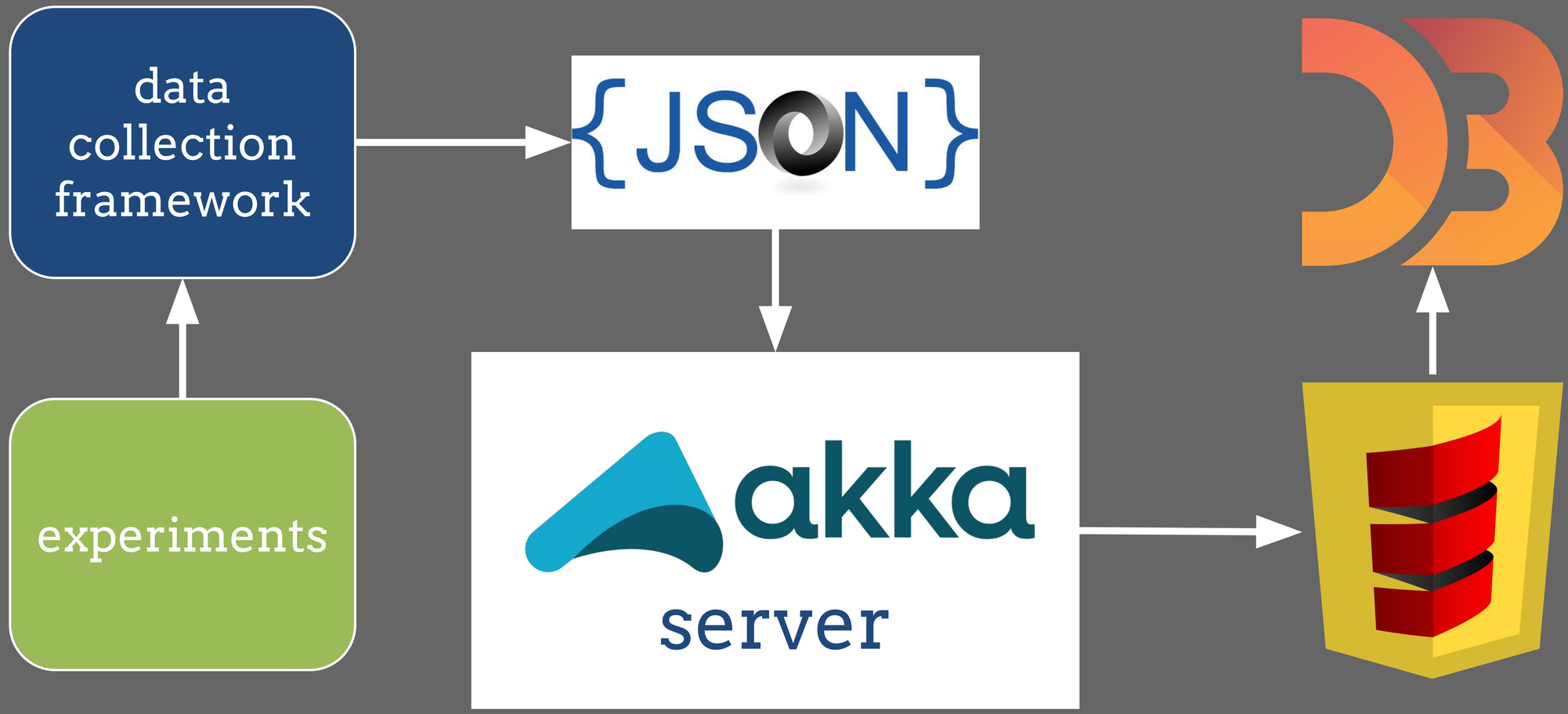




http://www.arbesman.net/the-half-life-of-facts/

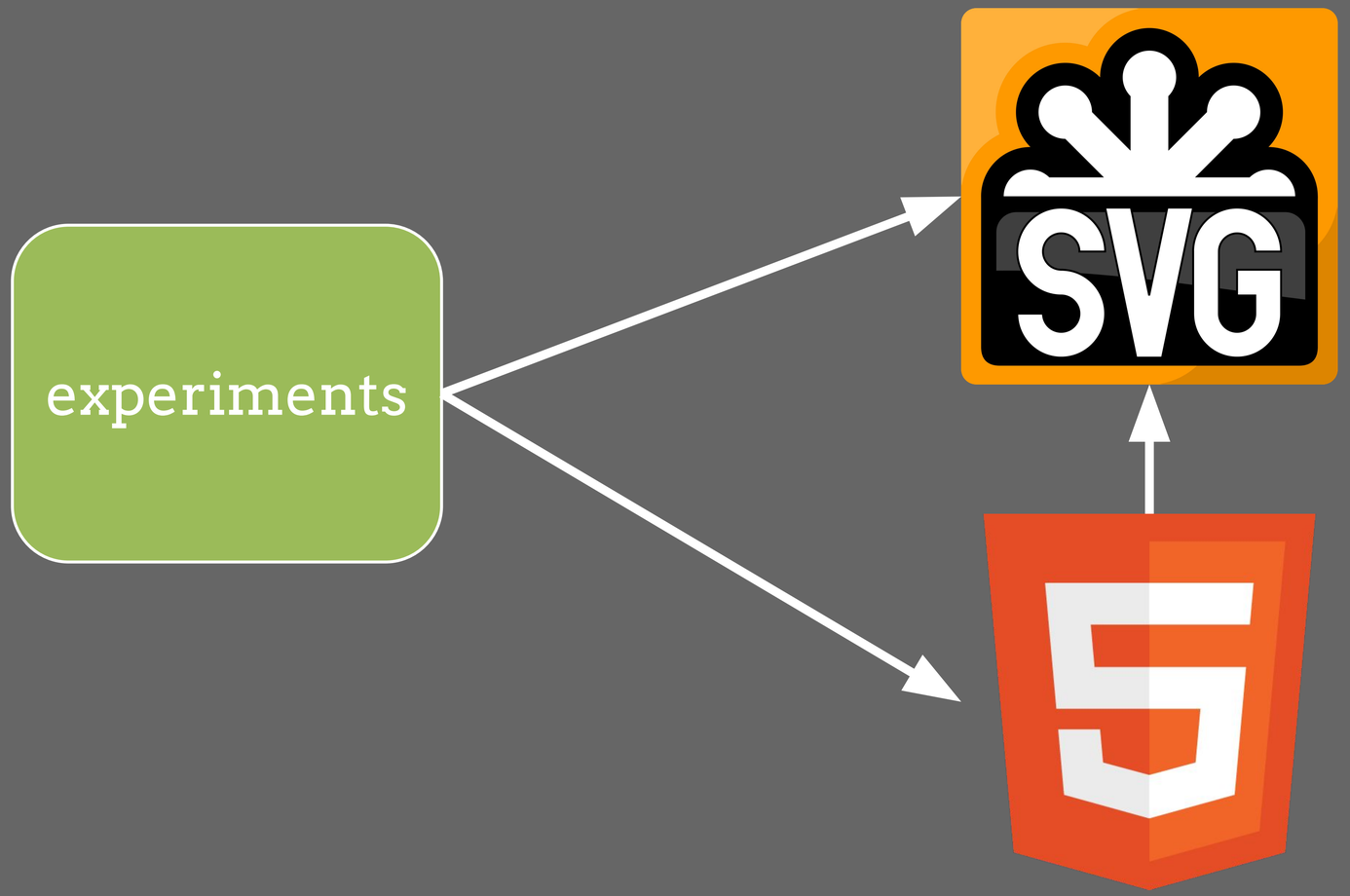
In order to grasp the truth, one must let go of many failed hypotheses

experimental insight app



https://www.youtube.com/watch?v=PQuDD_EHM9I

low-tech compromise

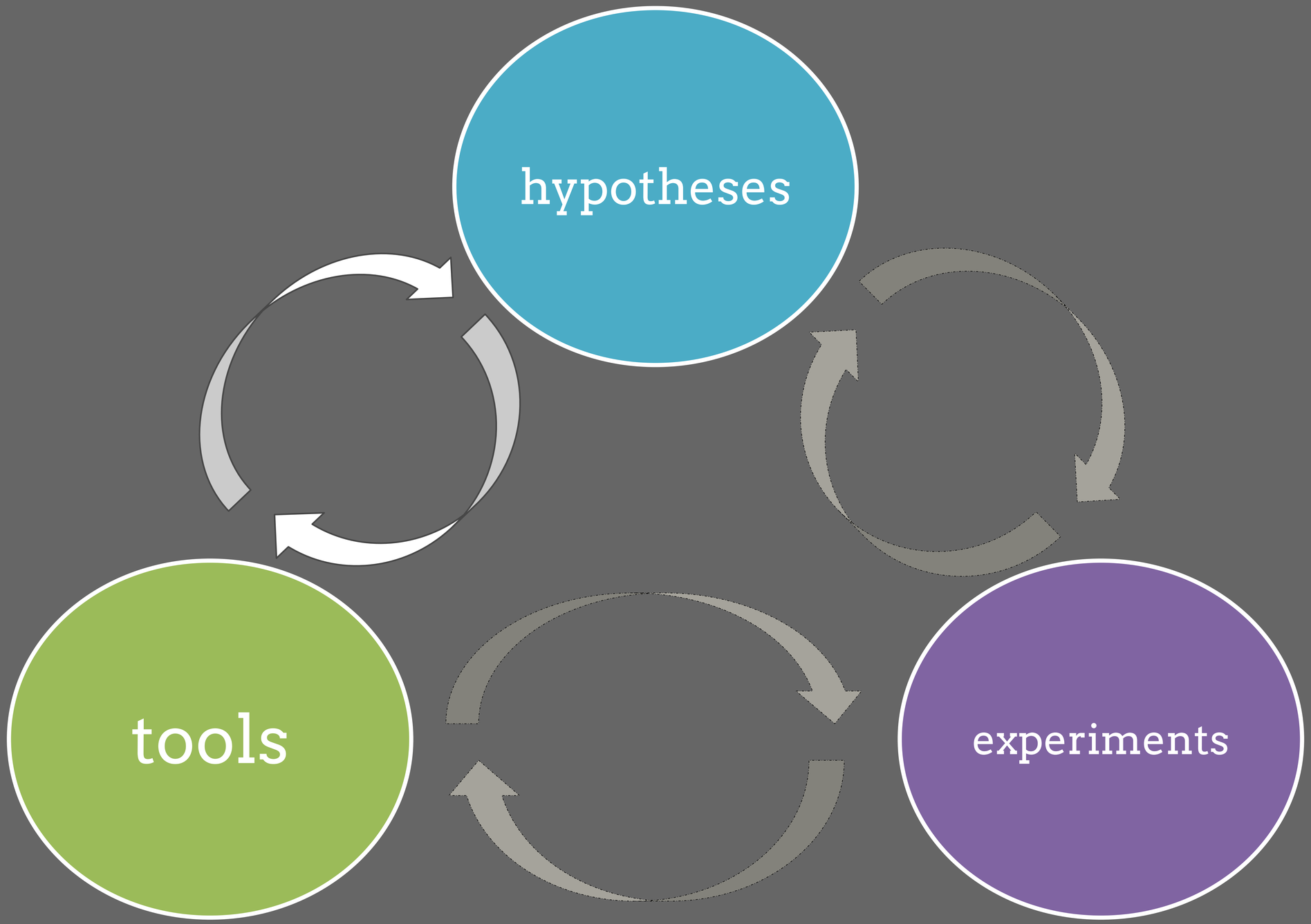


The more you've built, the harder it is to admit you're wrong.

When being right means adding complexity, apply Occam’s razor.

2. equip your laboratory

When all you have is a hammer, everything looks like a nail.


hypotheses
tools
experiments

building your library of tools

- tools = knowledge index
  - unused tools != dead code
- make tools versatile
  - reuse across experiments
- don’t over-engineer
  - when in doubt, print to console

types as instruction manual

- primitives = pure functions
- types define what’s possible
- types ensure tools used correctly

workspace requirements

- make tools accessible
- minimal setup
- distraction-free

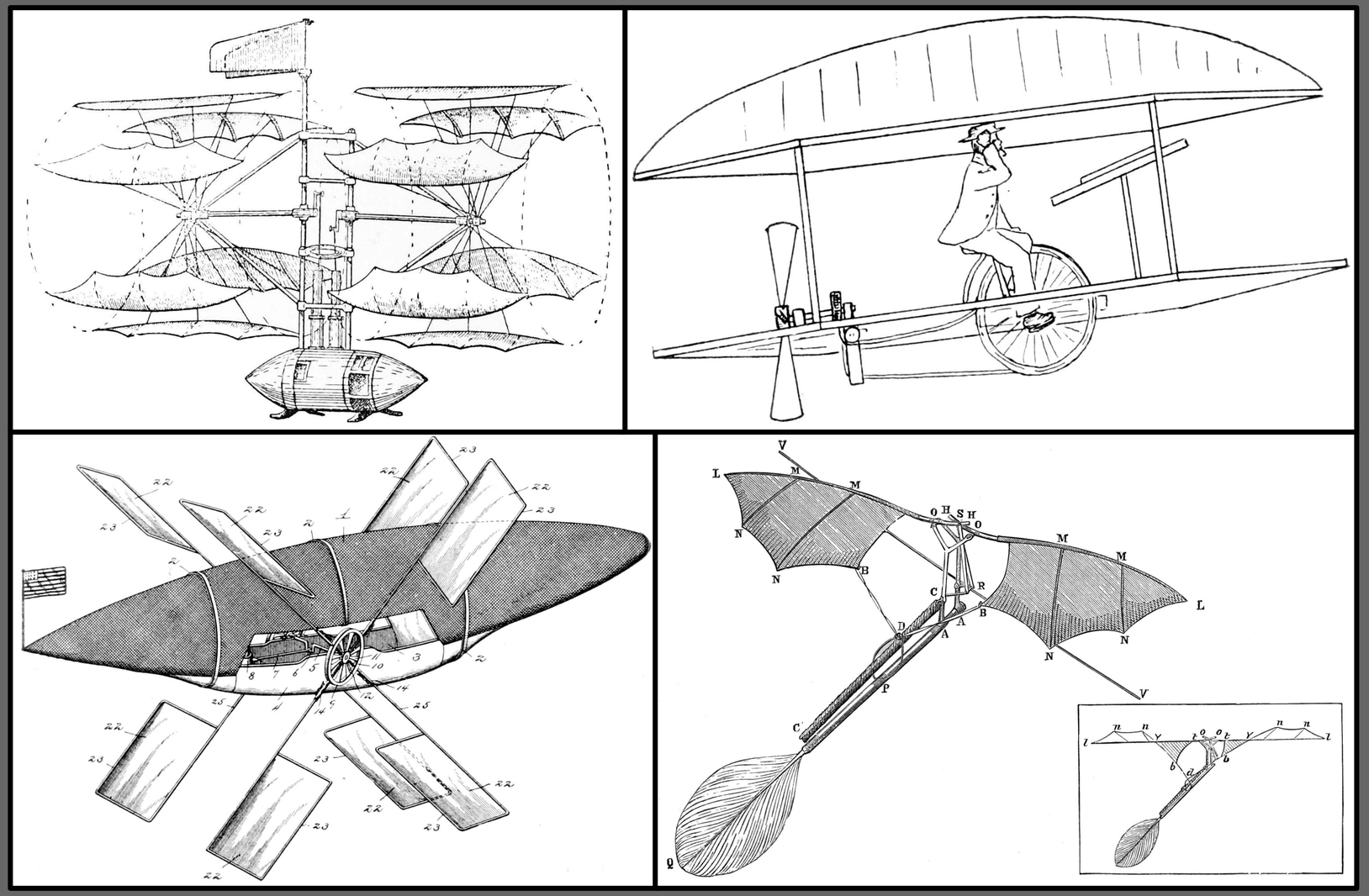
hierarchy of environments

type of test	import tools?	type of environment	our choice
quick & dirty (<90 seconds)	✓	REPL	Ammonite REPL (Scala)
lightweight, reusable (<60 minutes)	✓	script	Scala scripts (Ammonite)
ongoing, collaborative	✓	full-blown project	SBT, Gradle build system

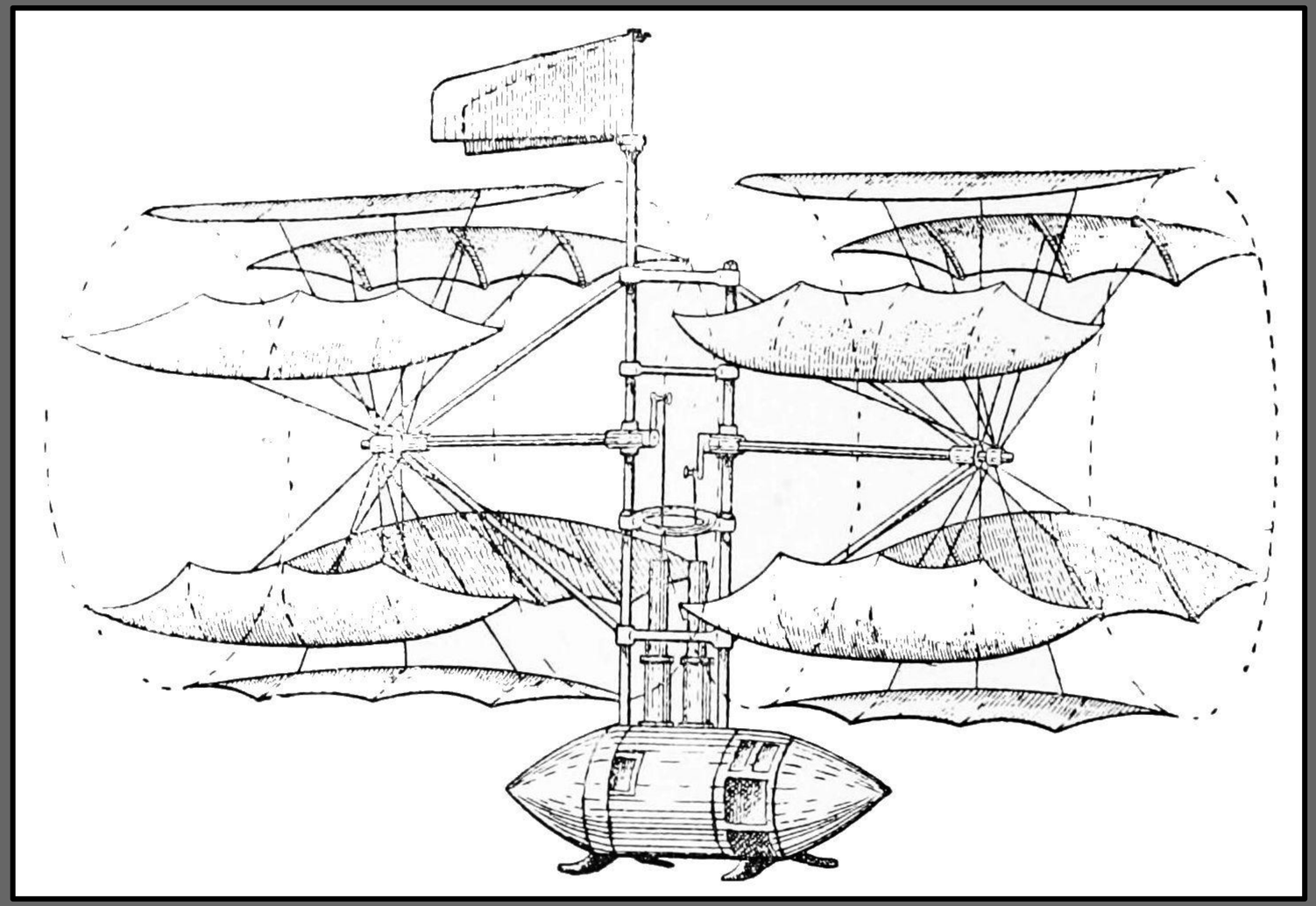
http://www.lihaoyi.com/Ammonite/

3. merge ideas, fork code

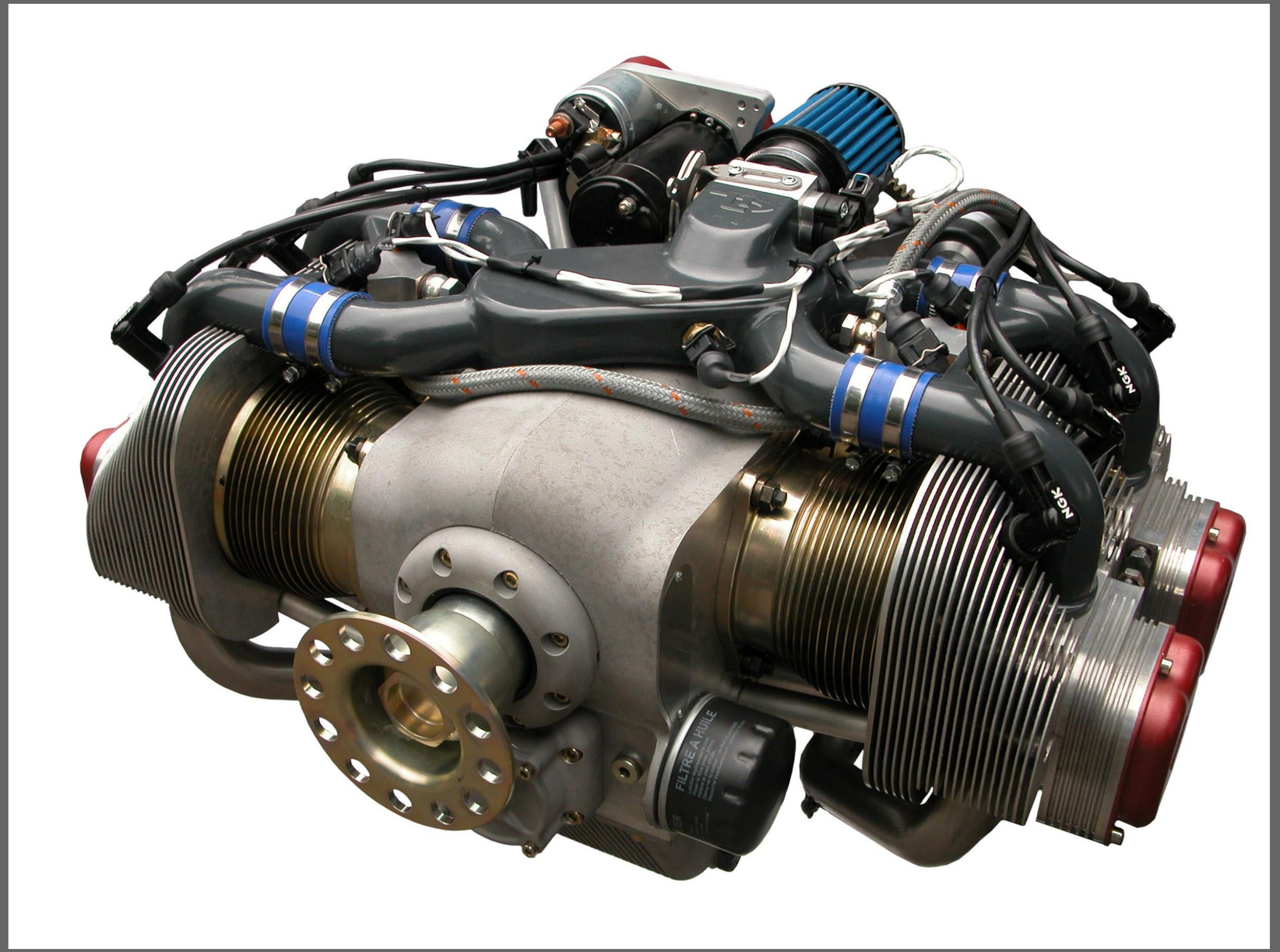
ambiguity means several hypotheses seem equally plausible



requiring a single codebase downplays ambiguity

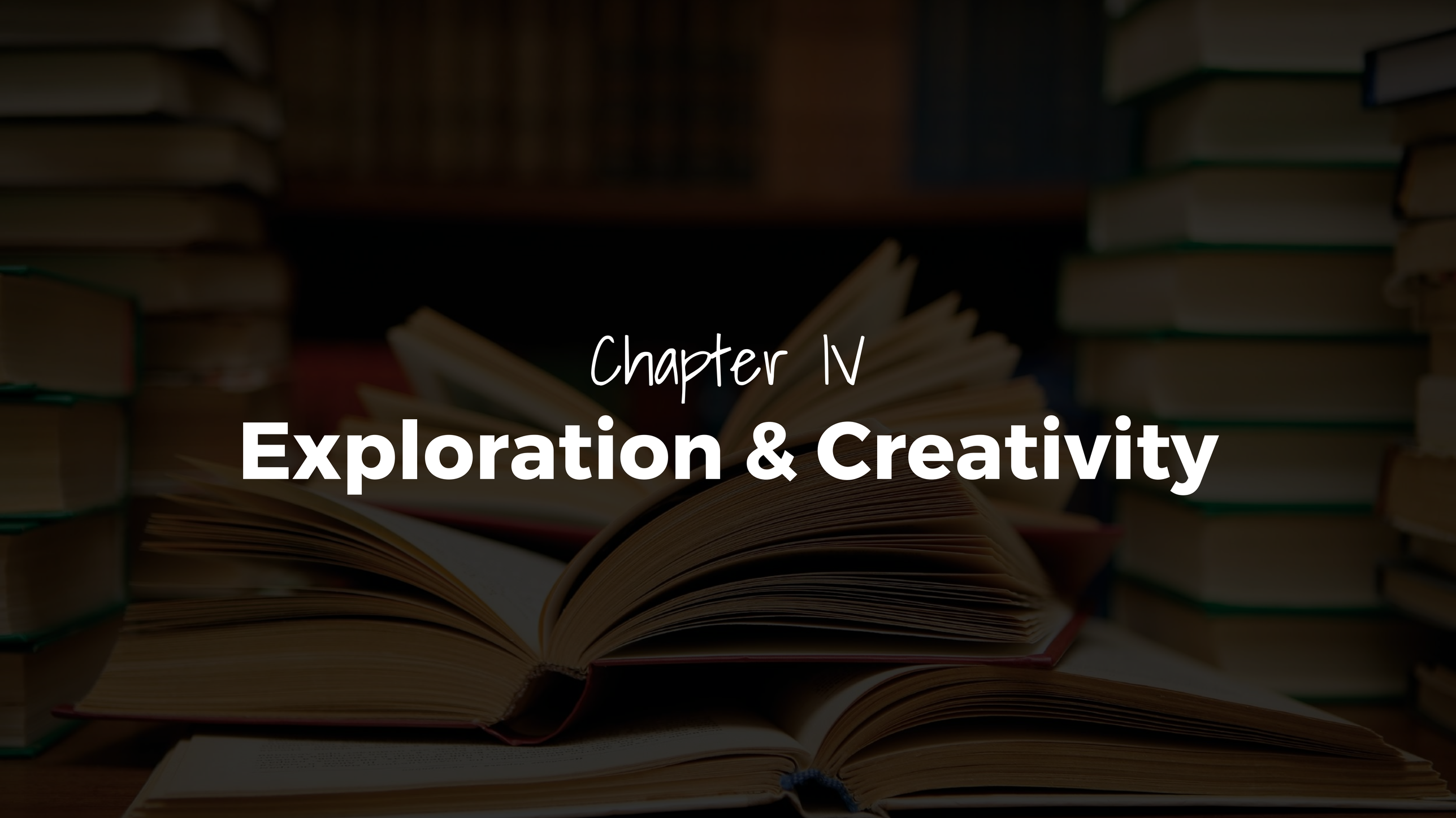


with a single codebase, discussions center on “how”



what to share

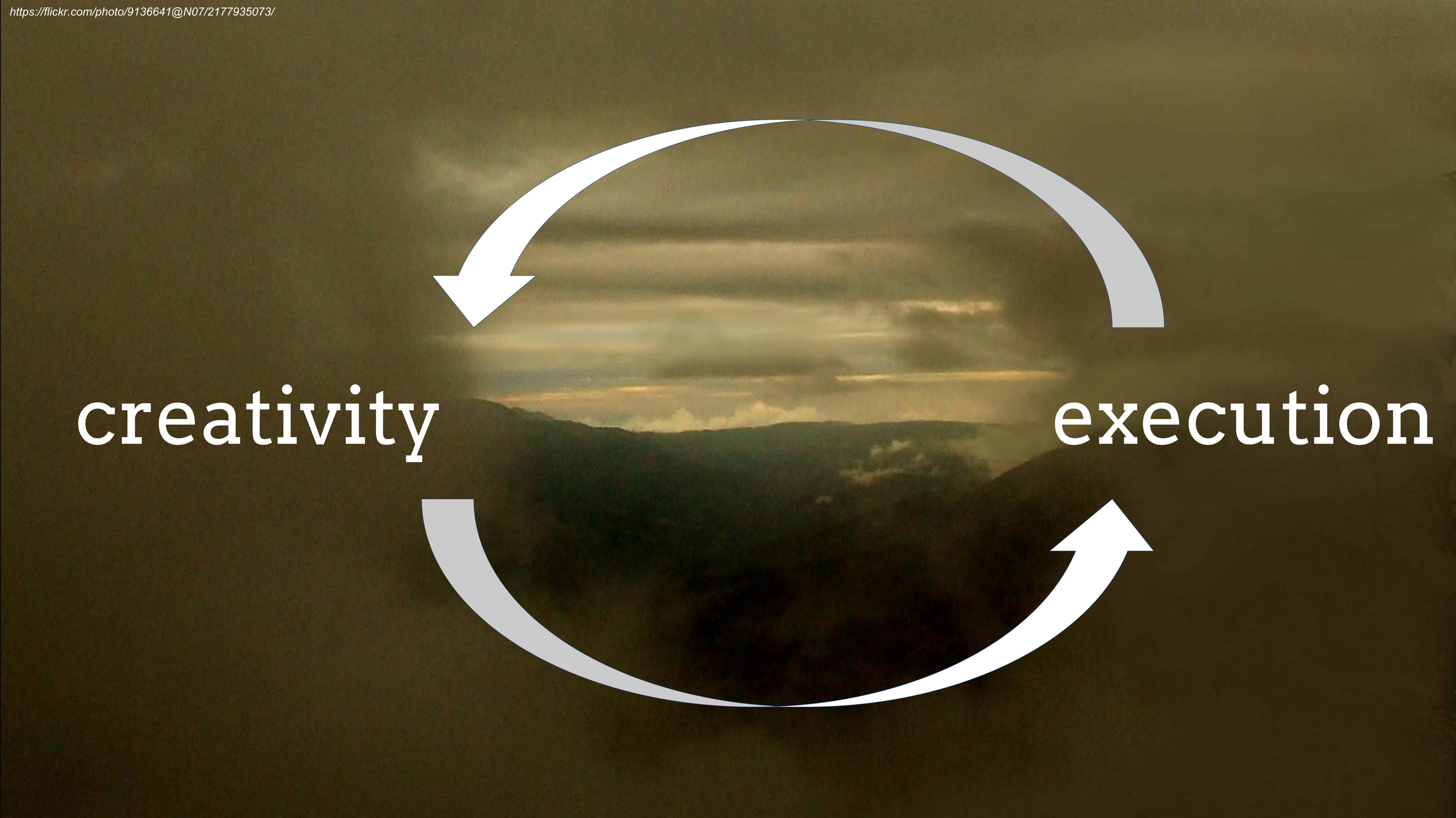
- hypotheses
  - divide & conquer
- tools
- requirements*
  - *as they become clear

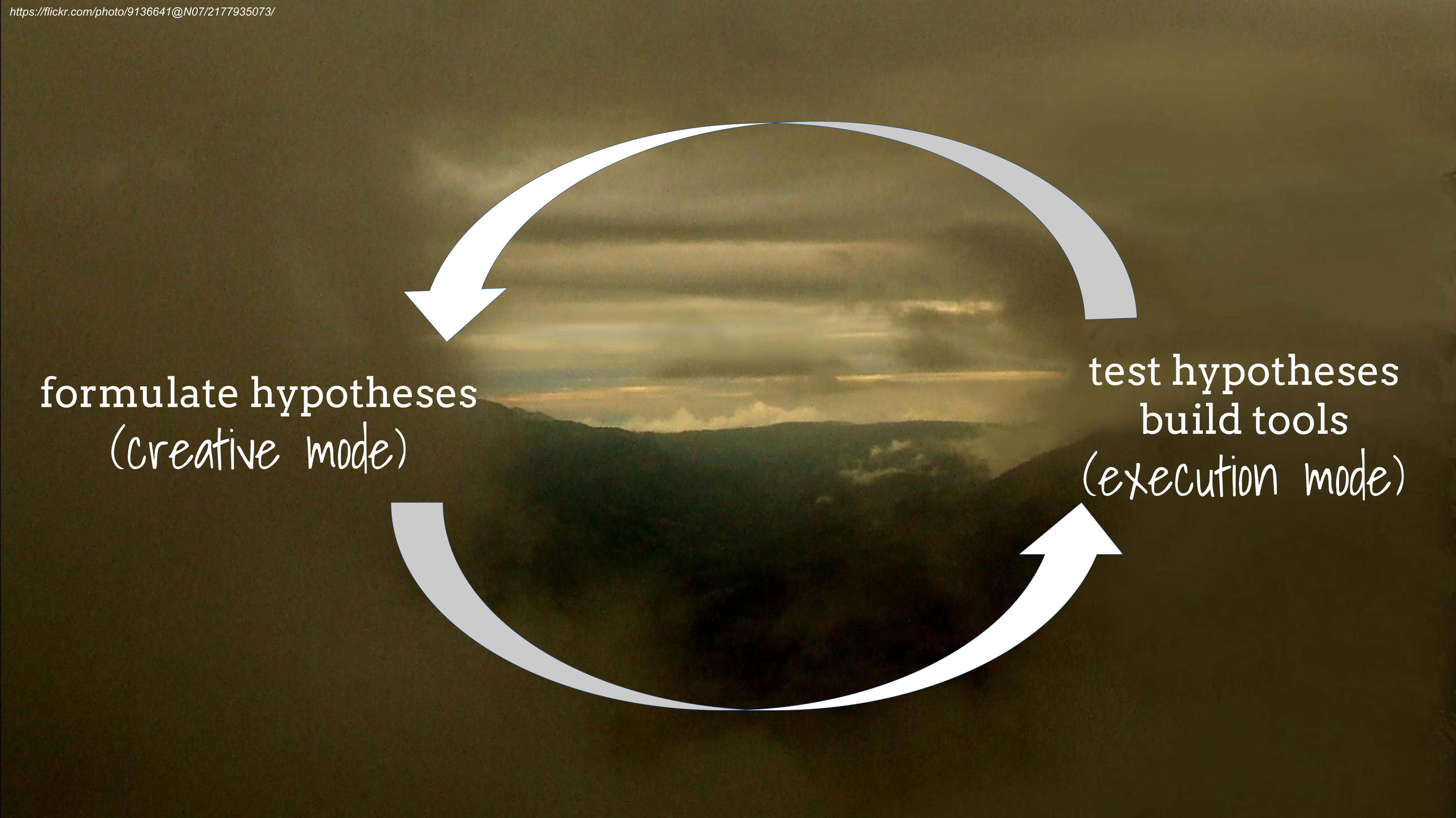


Chapter IV

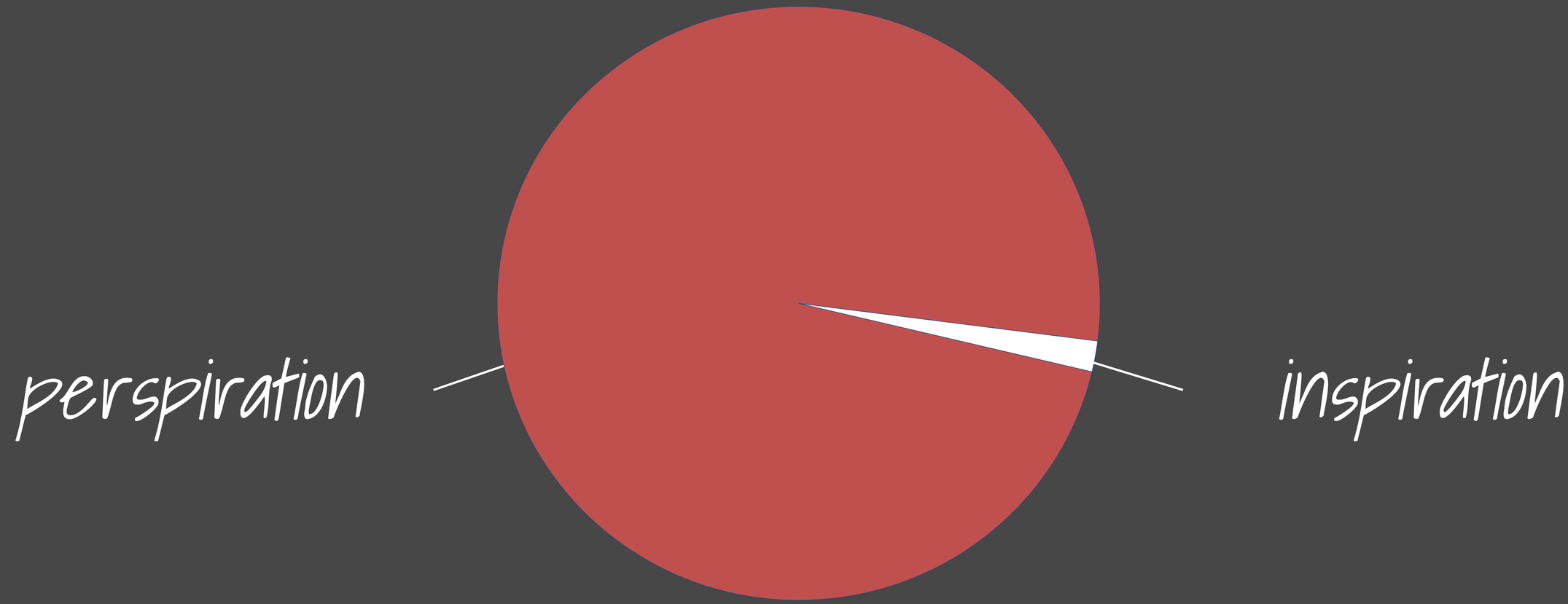
Exploration & Creativity

https://flickr.com/photo/9136641@N07/2177935073/

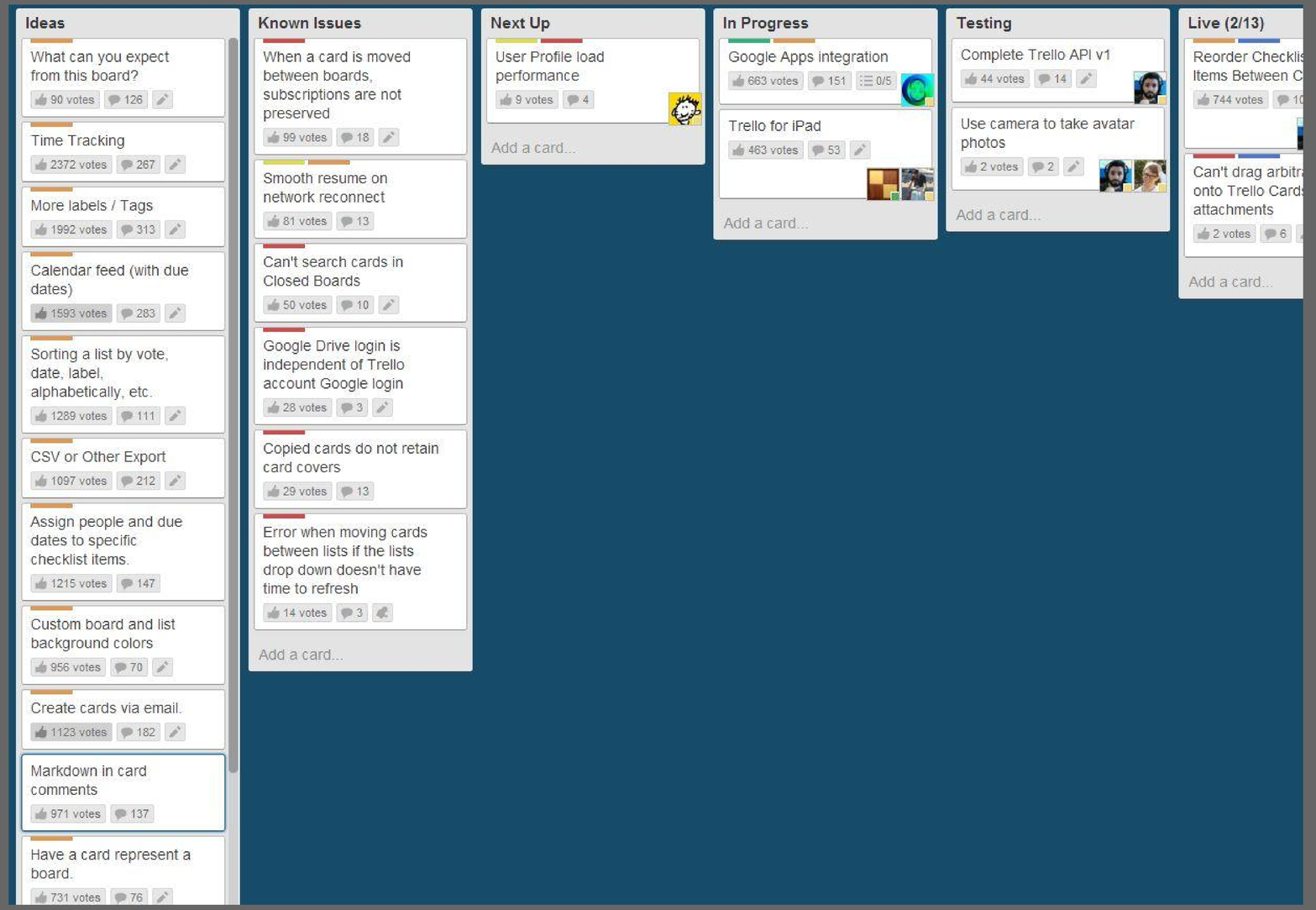



https://flickr.com/photo/9136641@N07/2177935073/
formulate hypotheses
(creative mode)
test hypotheses
build tools
(execution mode)

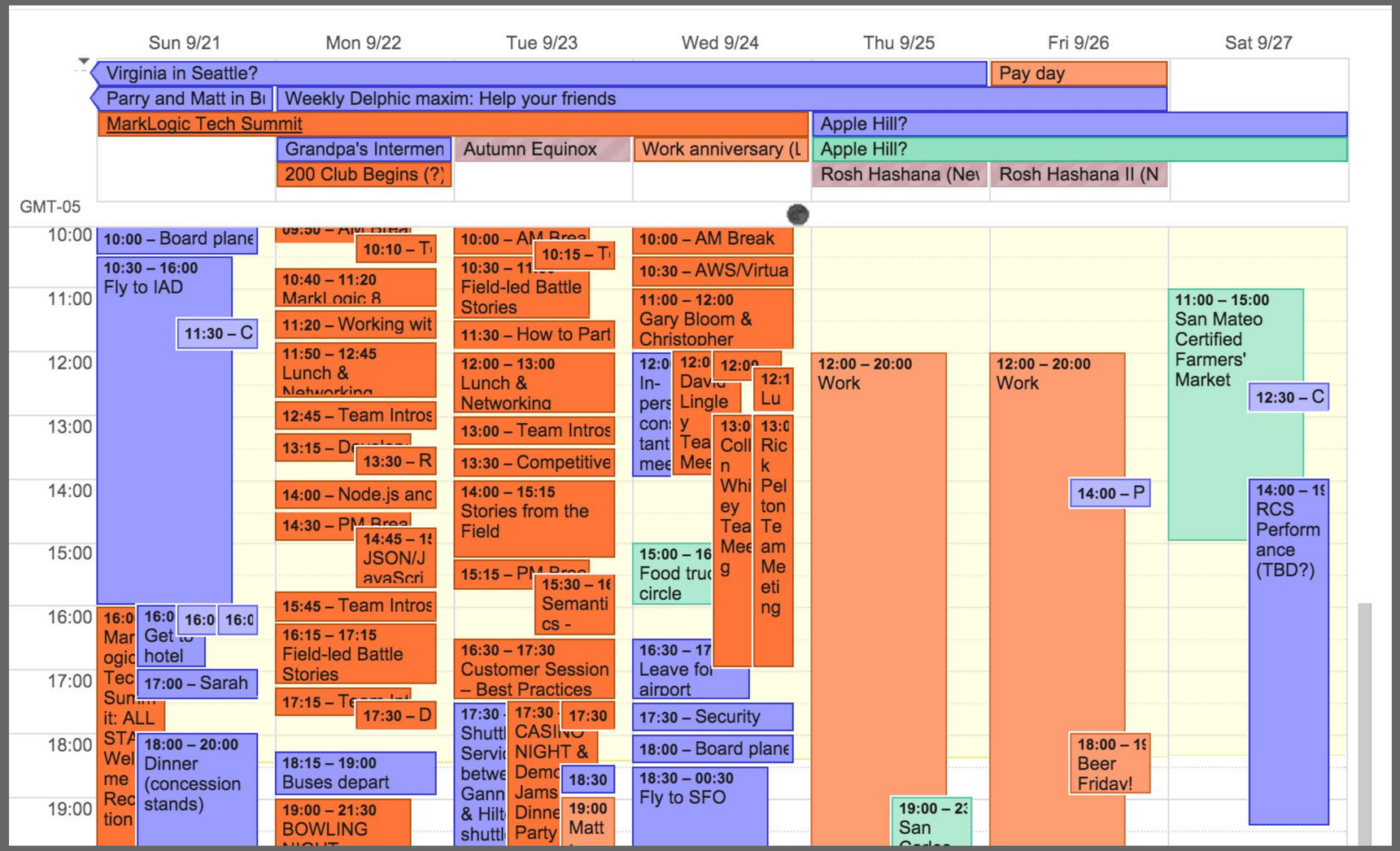
creativity

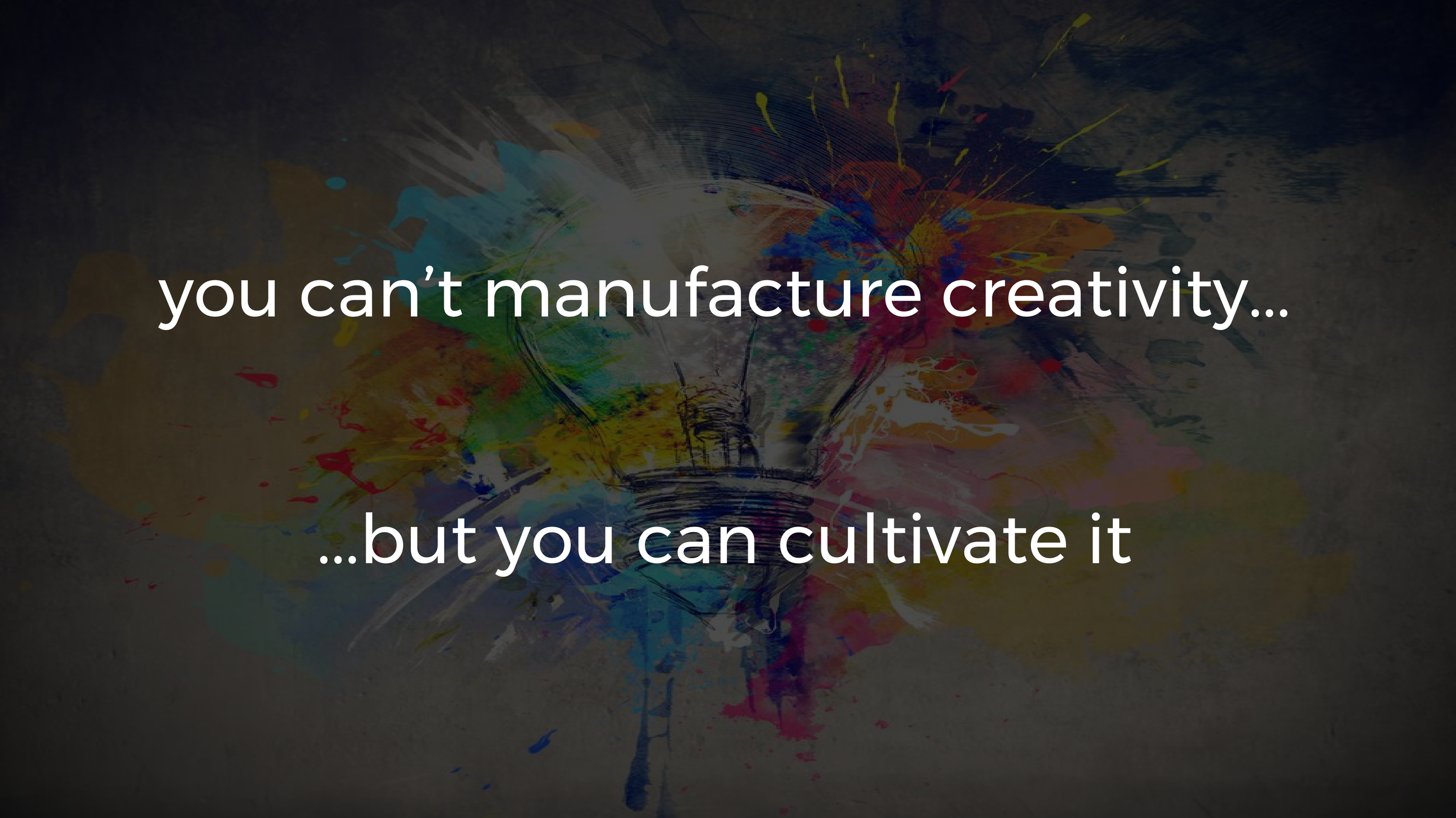


creativity can’t be managed



creativity doesn’t stick to a schedule

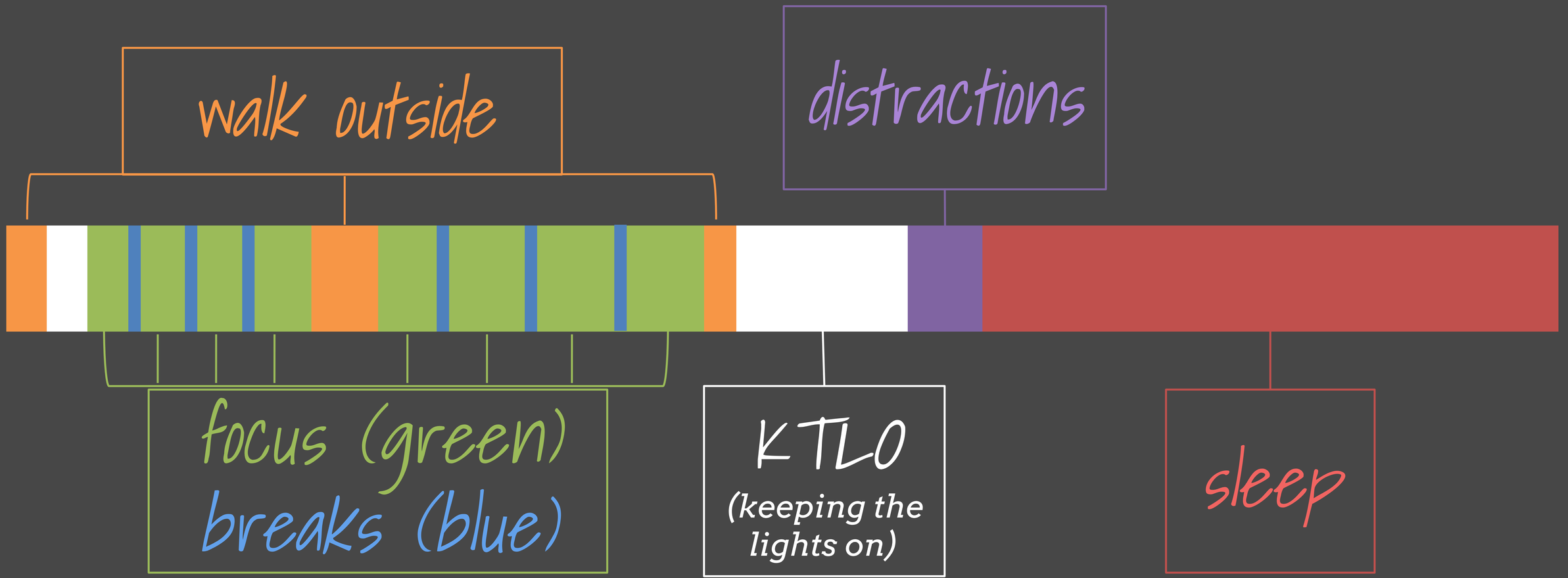



you can’t manufacture creativity...
...but you can cultivate it

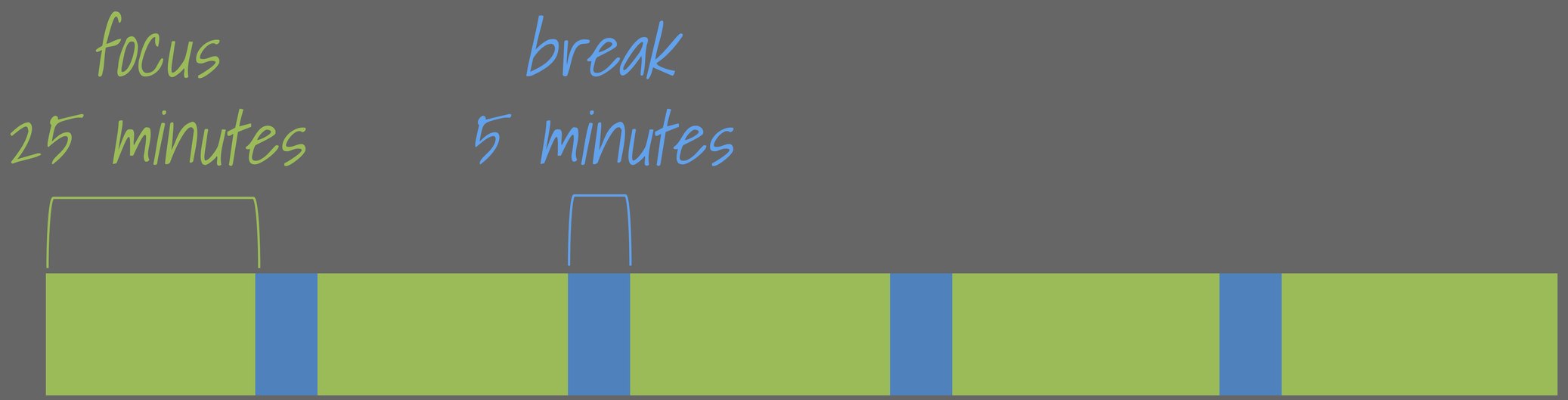
cultivating creativity

nourishment
exercise
rest

Julie's *ideal* creative mode day



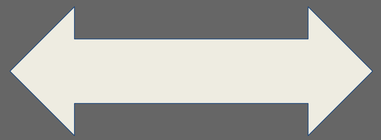
exercise (pomodoro technique)



- time-based, not outcome-based
- permission to rest
- avoid distraction
- daily quitting time

https://www.coursera.org/learn/learning-how-to-learn

https://www.coursera.org/learn/learning-how-to-learn



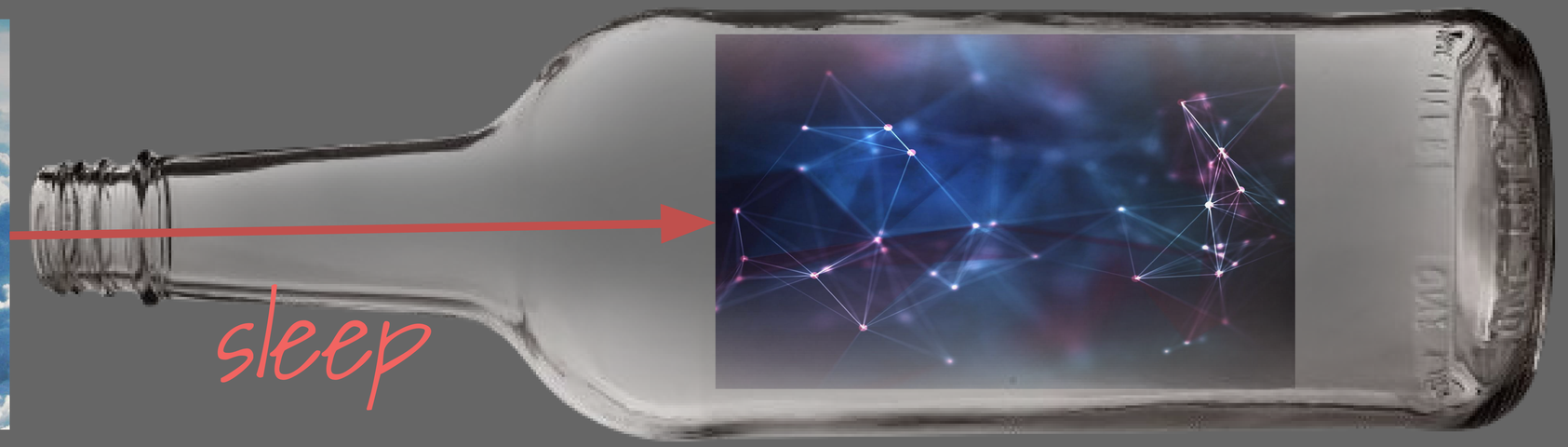
focused mode

diffuse mode

outdoor exercise pops you into **diffuse mode**, which can get you unstuck.

rest

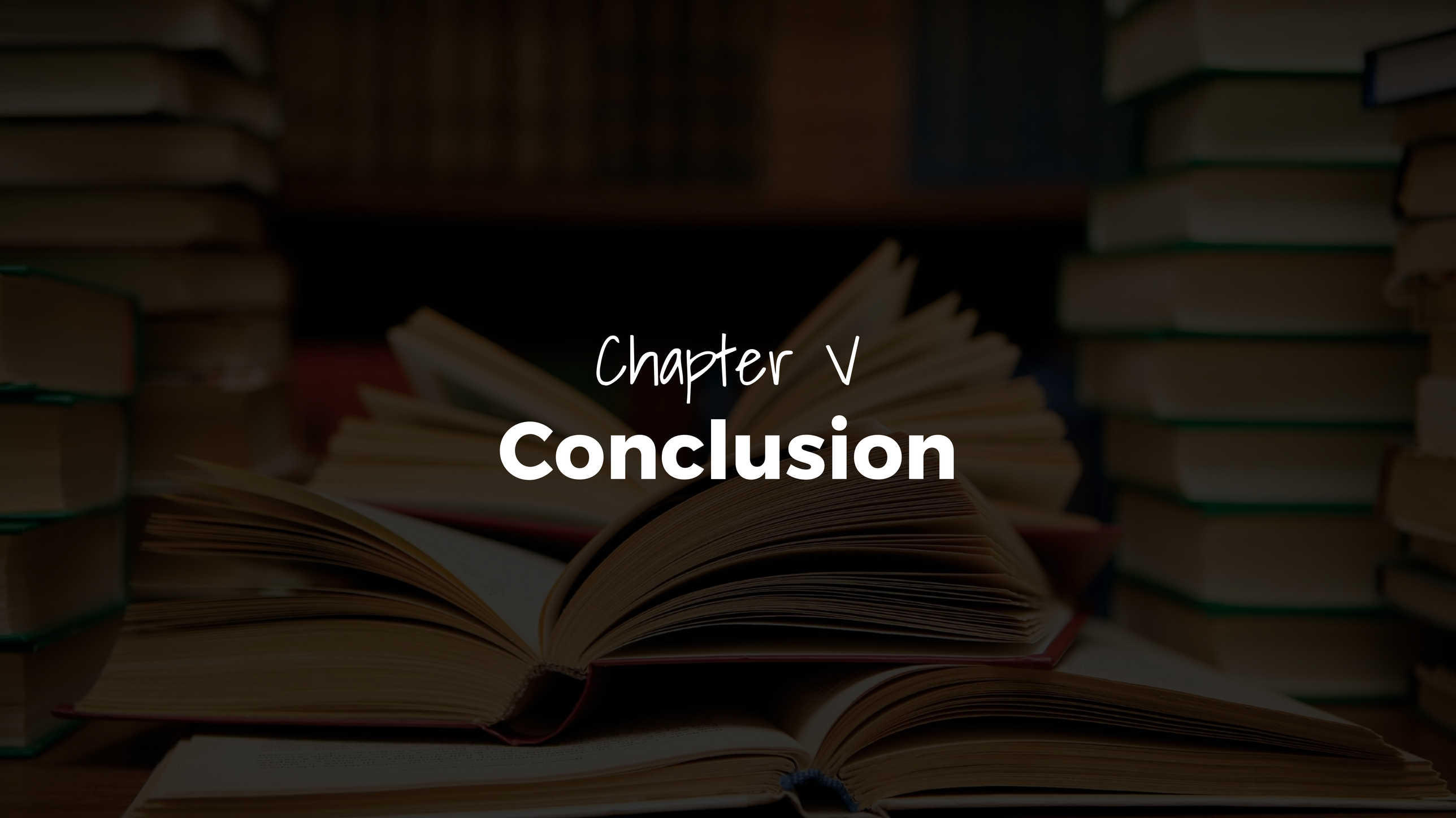
short term memory
(high complexity)



insight
(low complexity)

nutrition

too much distraction creates a complexity reduction bottleneck during sleep



Chapter V

Conclusion

at the beginning, act like a beginner

- build the embarrassingly simple version first
- just because you can do it doesn’t mean you should

optimize for hypothesis testing

- facts can change
- equip your laboratory
- merge ideas, fork code

cultivate creativity

- creativity and execution are modal
- creativity requires nutrition, exercise and rest

https://flickr.com/photo/69793885@N00/7874464932/

goto; conference

Please

Remember to rate this session

Thank you!

follow us @gotochgo