What is Software Carpentry?

Ben Morris

(thanks to Steve Crouch, Greg Wilson, Ethan White)

What is Software Carpentry?

“Software Carpentry helps researchers be more productive”

In the Seven Years' War, 1754-1763...

Britain lost 1,512 sailors to enemy attacks.

In the Seven Years' War, 1754-1763...

Britain lost 1,512 sailors to enemy attacks.

*...and nearly **100,000** to scurvy!*

The first (?) controlled medical experiment

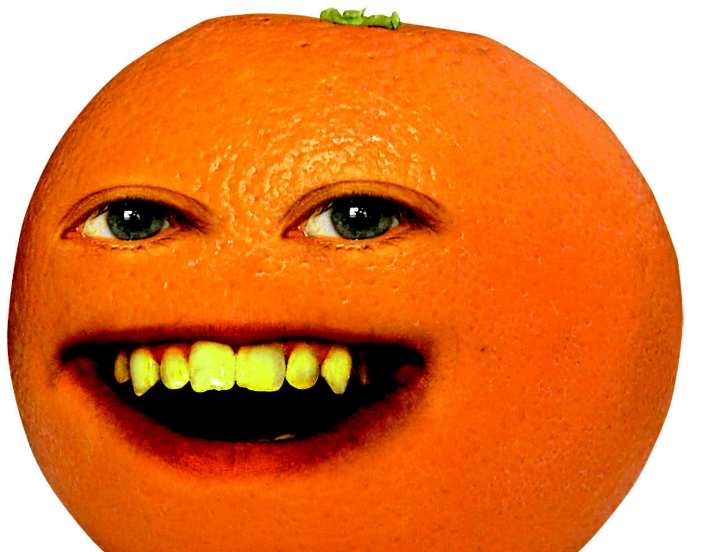
- James Lind, British scientist, in 1747
- Tested the efficacy of many substances thought to prevent scurvy:
  - Cider
  - Sea water
  - Sulphuric acid
  - Oranges
  - Vinegar
  - Barley water

The first (?) controlled medical experiment

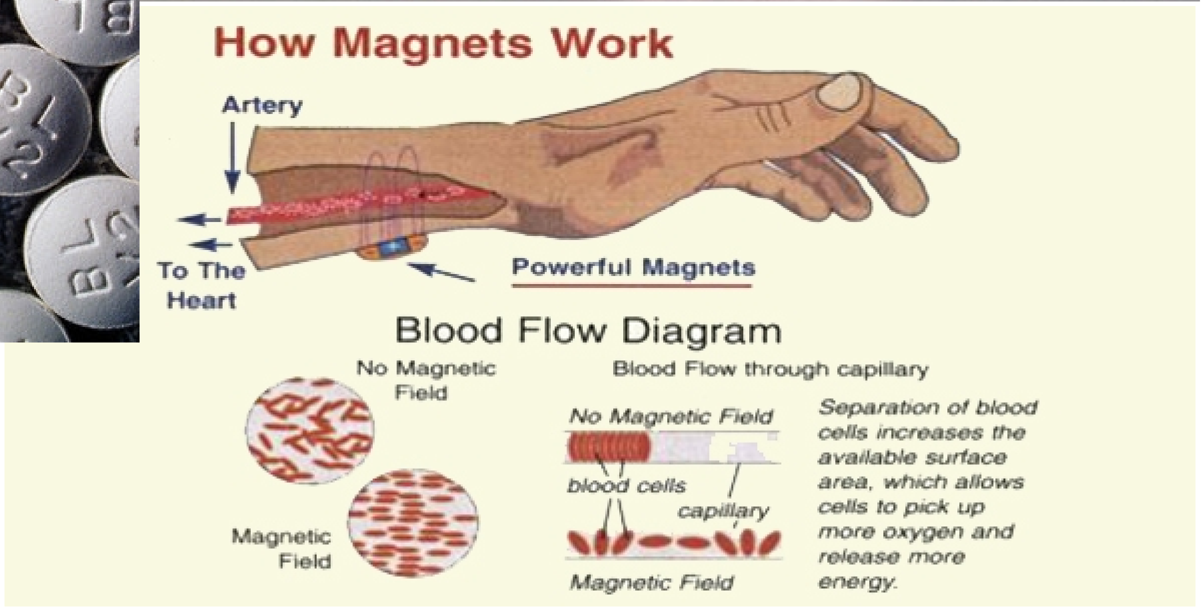
- James Lind, British scientist, in 1747
- Tested the efficacy of many substances thought to prevent scurvy:
  - Cider
  - Sea water
  - Sulphuric acid
  - **Oranges < == we have a winner!**
  - Vinegar
  - Barley water

The first (?) controlled medical experiment

- Yet the British Admiralty didn't listen (Lind wasn't an English gentleman) until 1794
- After 1794, dramatic worldwide decrease in deaths due to scurvy
- The scientific method worked!



Modern medicine


How Magnets Work
Artery
To The Heart
Powerful Magnets
Blood Flow Diagram
No Magnetic Field
Magnetic Field
Blood Flow through capillary
No Magnetic Field
blood cells
capillary
Magnetic Field
Separation of blood cells increases the available surface area, which allows cells to pick up more oxygen and release more energy.

Modern medicine

- How do we distinguish what works and what doesn't?
- **Evidence-**based medicine
  - Randomization
  - Double-blind studies
  - Transparency, data accessibility

Software is no different!

- Should be based on **evidence** of what works, not superstition or anecdotes
- What do we know about how to effectively develop software, and how do we know it?
- A certain amount of skepticism towards common software engineering anecdotes is healthy!
- “This works because many people believe it does”

A bold claim

- “The best programmers are up to **28** times more productive than the worst”
  - Sackman, Erikson, and Grant, “Exploratory experimental studies comparing online and offline programming performance” (1968)

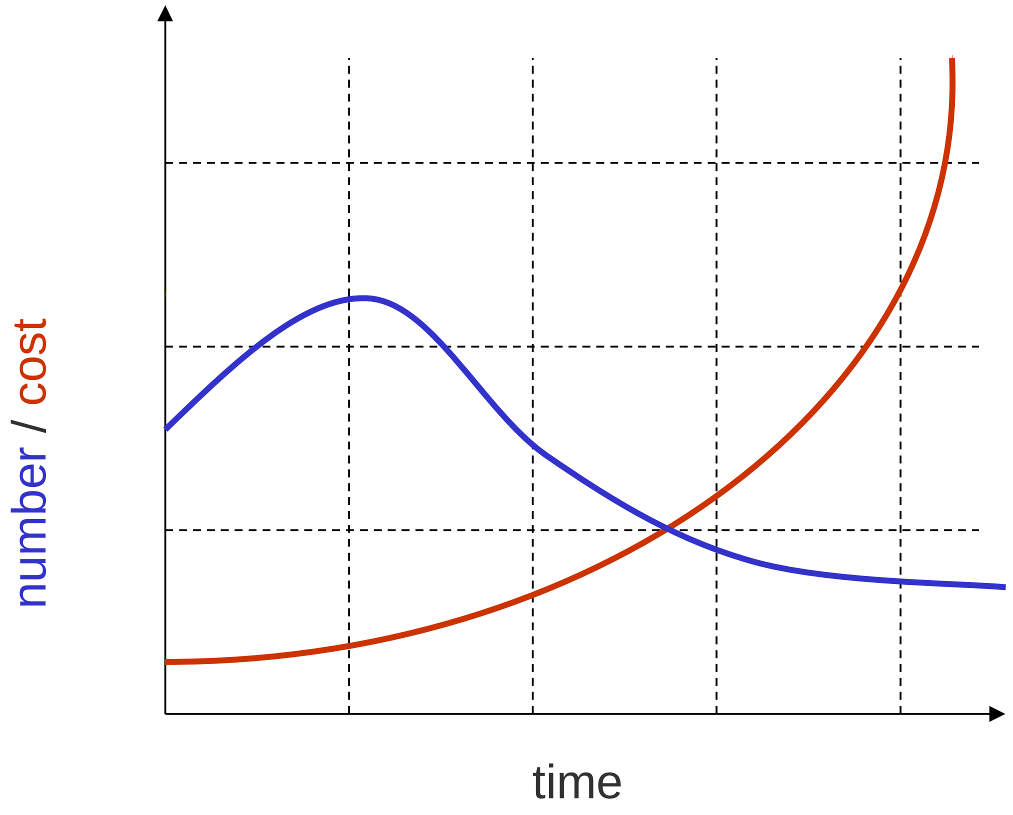
A bold claim

- “The best programmers are up to **28** times more productive than the worst”
  - Sackman, Erikson, and Grant, “Exploratory experimental studies comparing online and offline programming performance” (1968)
- Hold up...
  - **1968**
  - Study involved 12 programmers for an afternoon
  - Designed to compare batch vs. interactive

So what do we know?

- Most errors are introduced during the early stages of development (design and requirements analysis)
- The later an error is detected, the more costly it is to address

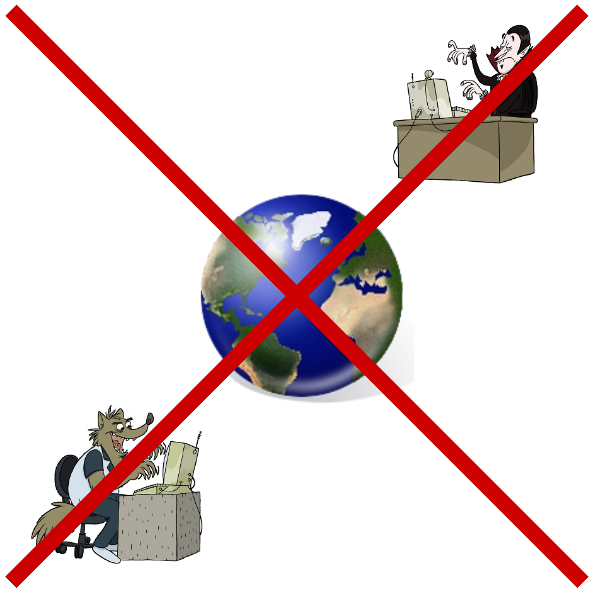
Boehm et al (1975)

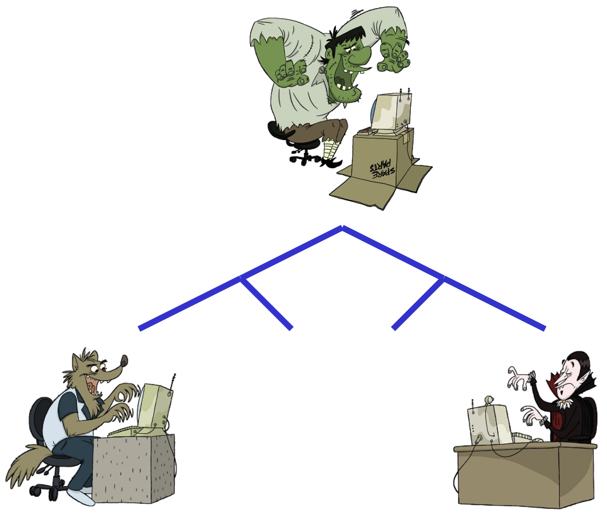


So what do we know?

- Physical distance doesn't matter
- Organizational distance does

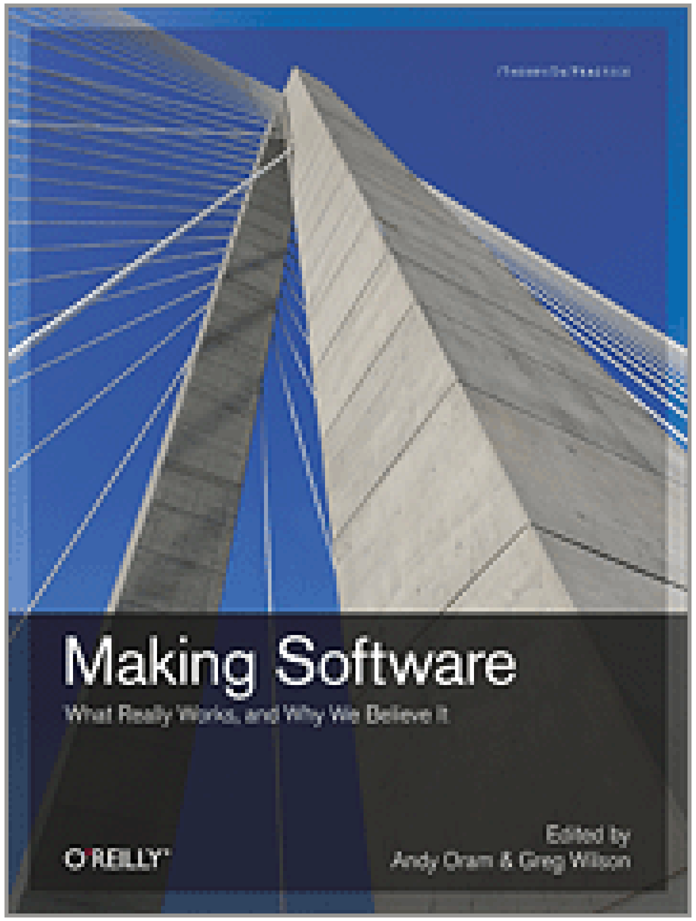
Nagappan et al. (2007), Bird et al. (2009)





So what do we know?

Facts and Fallacies of Software Engineering
Robert L. Glass
Foreword by Alan M. Davis



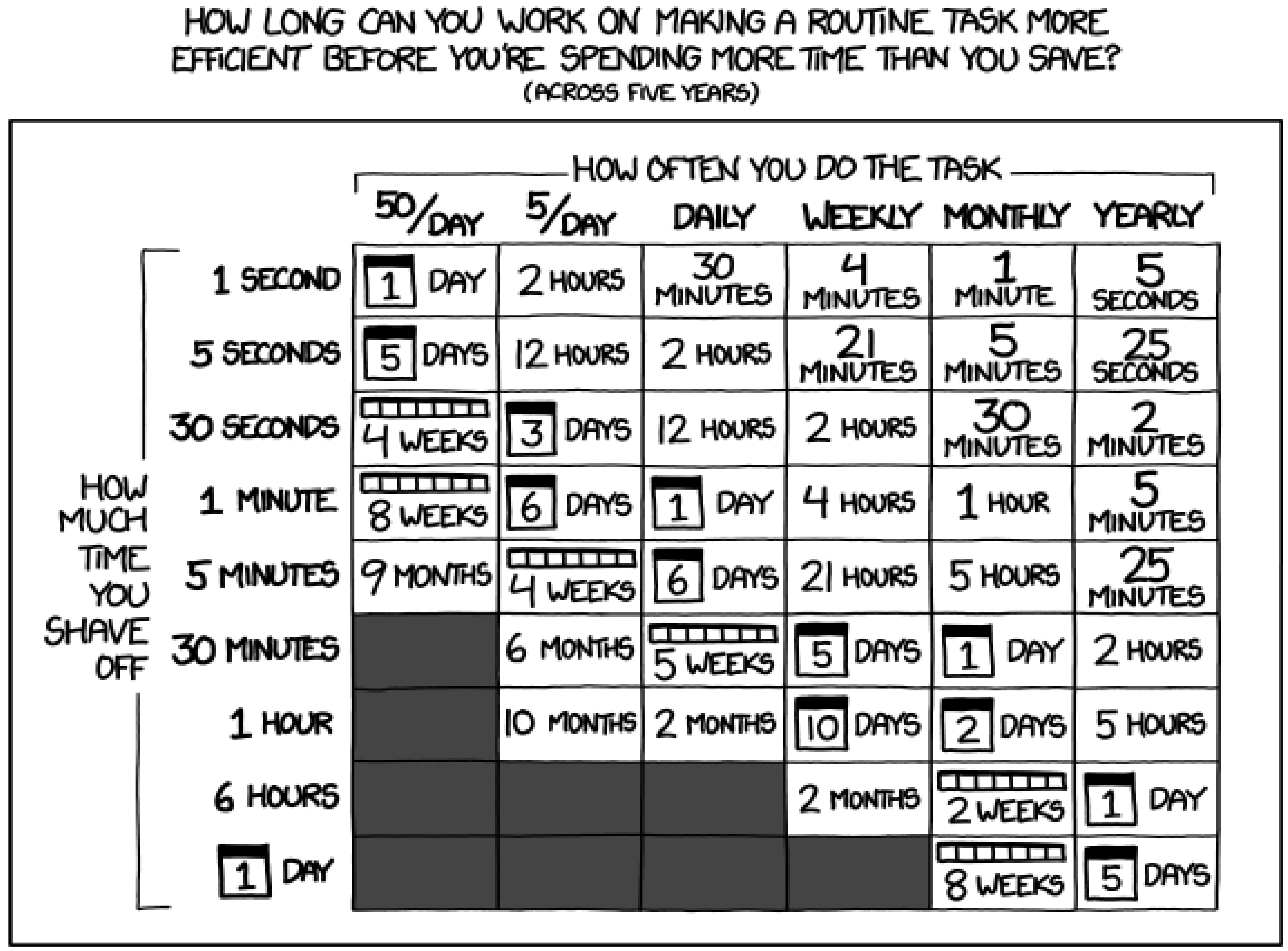
http://software-carpentry.org/about/biblio.html

Optimization

- What are some things we can optimize in software development?
  - Computing time (often fairly cheap)
  - Programmer time (expensive)
  - Cognitive load
    - Your brain can juggle about 7 ± 2 chunks of information at once in its short term memory
- Whatever we choose to optimize, there should be a reason

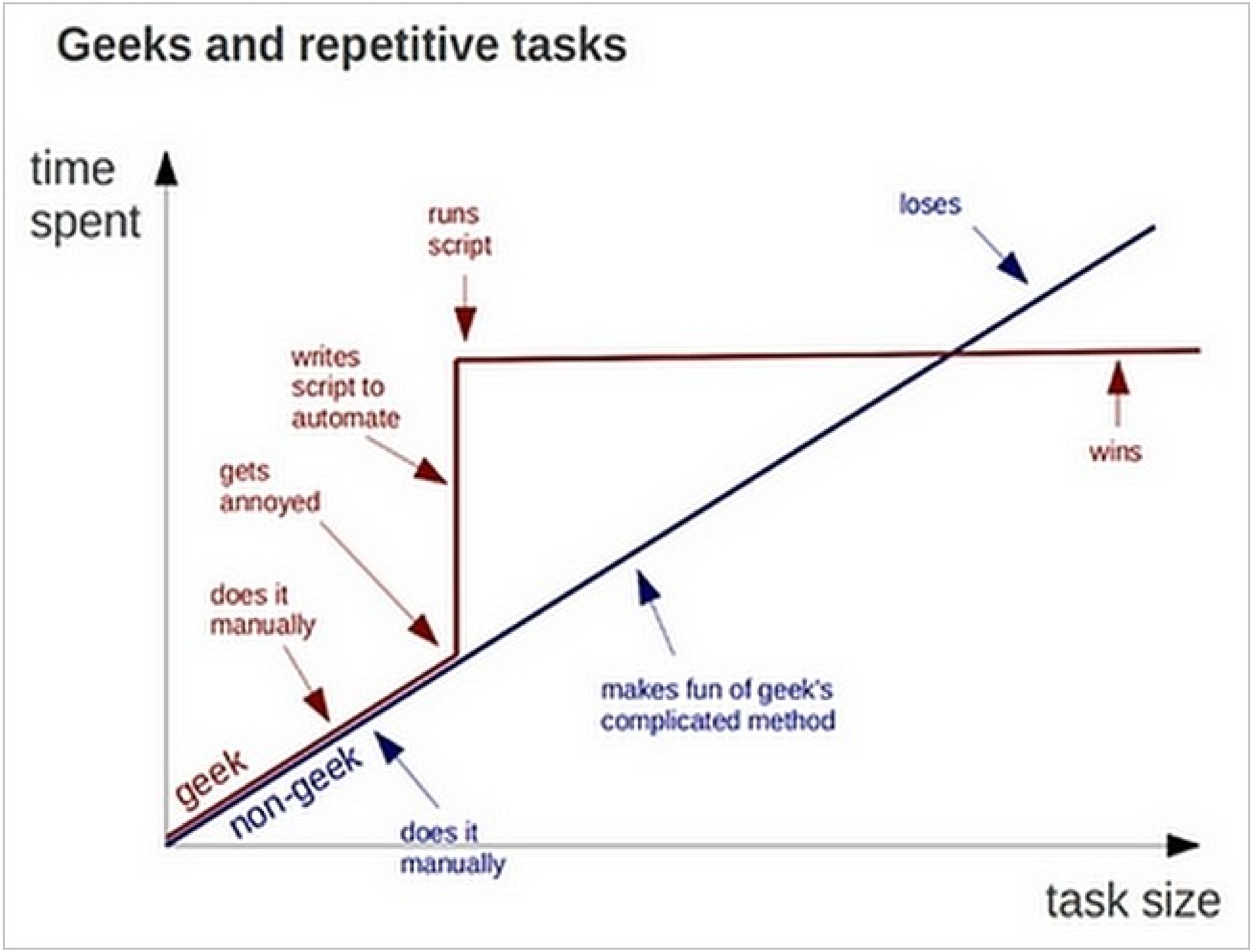
Why automate?

- Optimize programmer time: let the machine handle things without your supervision
  - e.g. on a computing cluster
- Optimize cognitive load: record those pesky command line options that you can never seem to remember, and forget them!
- For yourself – *repeatability*
- For others – *reproducibility*



HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

HOW MUCH TIME YOU SHAVE OFF	HOW OFTEN YOU DO THE TASK: 50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
30 MINUTES	■	6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
1 HOUR	■	10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
6 HOURS	■	■	■	2 MONTHS	2 WEEKS	1 DAY
1 DAY	■	■	■	■	8 WEEKS	5 DAYS


Geeks and repetitive tasks
time spent
task size
runs script
loses
writes script to automate
wins
gets annoyed
does it manually
makes fun of geek's complicated method
geek
non-geek
does it manually

Reproducibility

“Commonly research involving scientific computations are reproducible in principle, but not in practice.”

“In our laboratory, we noticed that after a few months or years, researchers were usually unable to reproduce their own work without **considerable agony**.”

Schwab, Matthias, et al. "Making scientific computations reproducible." Computing in Science & Engineering 2.6 (2000): 61-67.