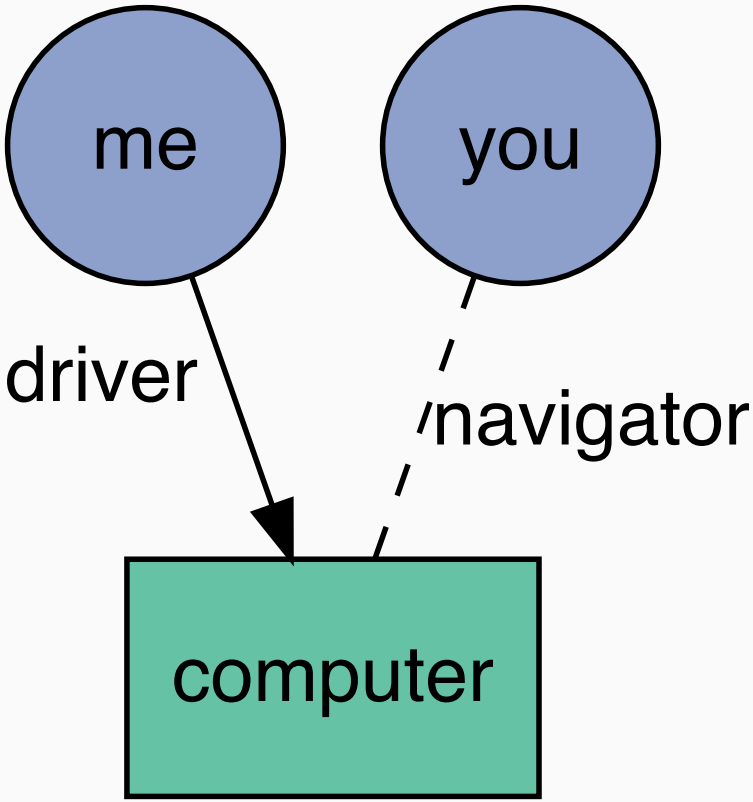
Introduction to pair programming

What is pair programming?



Why pair programming is terrible

- Coding in front of someone else is terrifying.
- Typing on someone else's computer is frustrating.
- Explaining everything takes too much time.
- Watching and understanding nothing is a waste.

We think pair programming is terrible because we think it is inefficient.

Why pair programming is worth it

- Customized learning environment.
- Interactive learning is better than passive.
- Pair programming solves “unknown unknowns”.
- People over processes (Agile Manifesto).

Agile principles relevant to pair programming

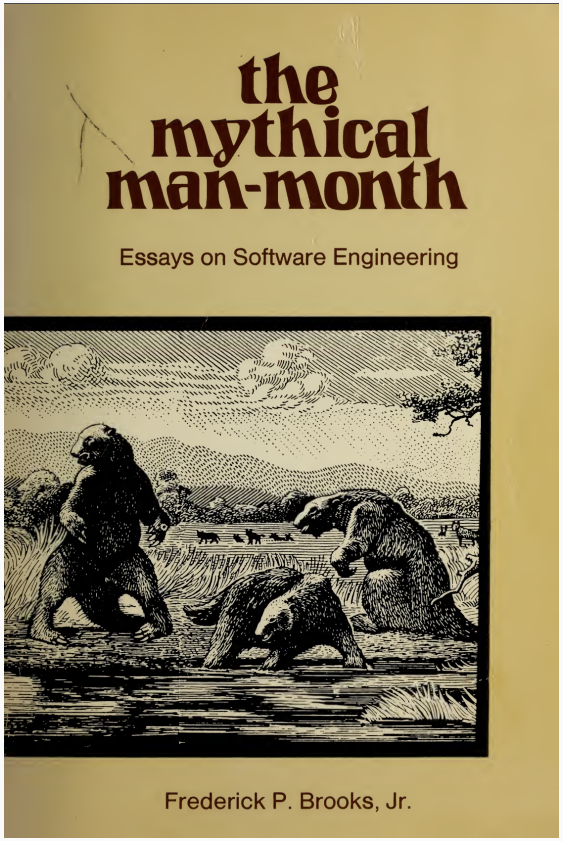
- "Welcome changing requirements, even late in development."
- "Build projects around motivated individuals."
- "Working software is the primary measure of progress."
- "The best architectures, requirements, and designs emerge from self-organizing teams."

Productivity in programming

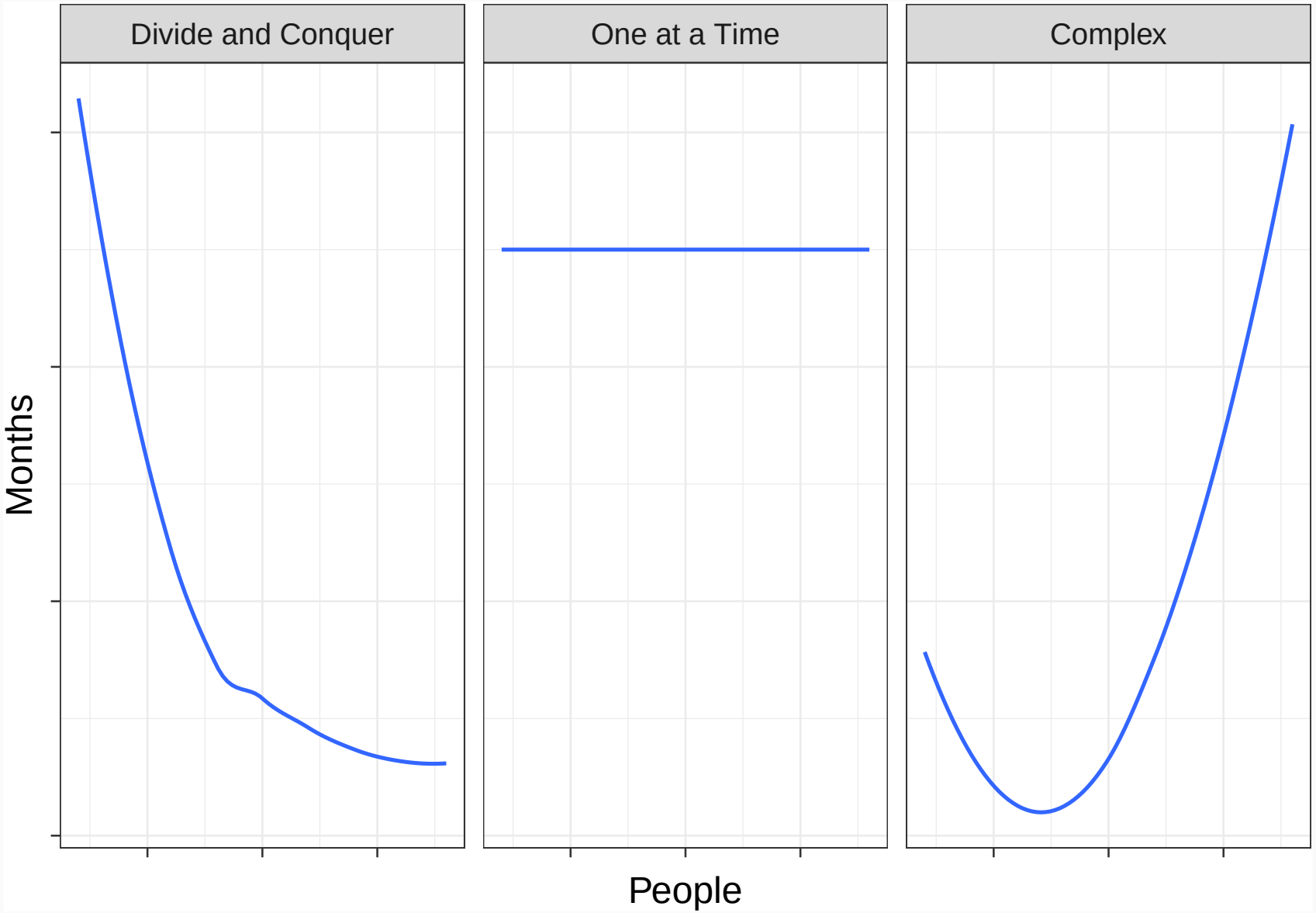
It's not what you think it is!

1. The mythical man-month
2. The cathedral and the bazaar

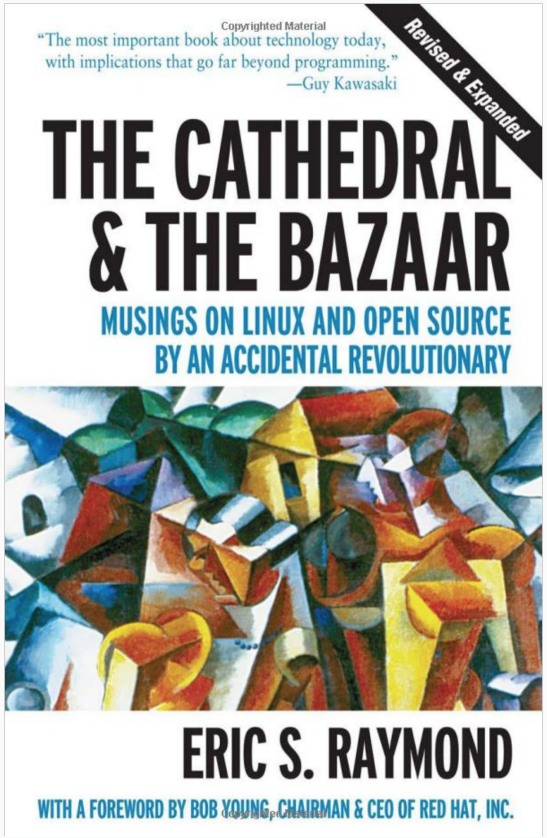
The mythical man-month (Brooks, 1974)



The mythical man-month (Brooks, 1974)



The cathedral and the bazaar (Raymond, 2001)



Is pair programming actually inefficient?

Strengthening the case for pair programming

Williams, Kessler, Cunningham, & Jeffries (2000). *IEEE Software*.

> *Teams solved problems faster than individuals with better algorithms and with higher satisfaction.*

Evaluating pair programming with respect to system complexity and programmer expertise.

Arisholm, Gallis, Dyba, & Sjoberg (2007). *IEEE Transactions on Software Engineering*.

> *No overall gains in productivity, but junior programmers were better able to solve complex problems, and senior programmers were faster at solving simpler problems.*

Madpy Pair Night

Where: Industrious.

When: Weeknight (survey results).

Why come to Pair Night?

- **Student-Teacher.** To learn something you don't already know, or practice a skill you need to improve.
- **Teamwork.** To do something together you could not have done on your own.

Goals for pairs

- **Student-Teacher.** Student learns from Teacher, Teacher gets better at teaching (rubber duck debugging).
- **Teamwork.** Teammates play to each other's strengths and accomplish something together.

What are we going to work on?

Pairs should be working to solve some problem.

"Working software is the primary measure of progress."

- **Practice problem.** Take a problem off the shelf together.
- **A real problem.** Work on a problem that will be used beyond the Pair Night.

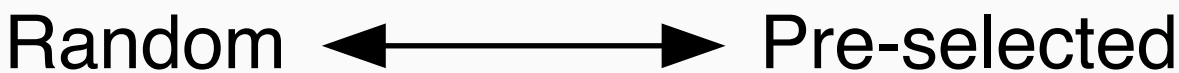
Practice problems

- Advent of Code
- Nifty assignments
- Code golf
- Kaggle competition
- Example project (e.g., from a book)

Real problems

- Find the expert!
- GitHub issues
- Make a thing you both want to exist

Who are we going to work with?



House rules on pairings

Randomly assign groups and select pairs from the group.

- Everyone completes a simple survey.
- I generate groups of like-minded people.
- At the Pair Night, people meet their group, and form pairs.

Note: If you want to come pre-paired, that is fine!

Timeline: 2 hours and 15 minutes

- 45: Intro, pairings, and pizza
- 45: Pair #1
- 45: Pair #2