Joining Your First Collaborative Software Project

George Dill

Engineer / Founder Avondalien Software LLC

Avondalien
Software

Summary

- Goals change when you transition from student to collaborator
- Teams can decide what works best for them, but an agile framework can be a good way to collaborate
- Many popular tools exist to facilitate collaboration - we'll examine git and GitHub
- Software composition patterns that may seem onerous in solo dev actually facilitate group work

About Me

George Dill | Age 40

- Software Engineer since 2020
- Worked in enterprise shops like John Deere and Echo Global Logistics
- Earned a Masters Degree in CS from University of Illinois at Chicago
- Previously worked in Construction Mega-Project Management both in the US and South Korea
- Also ask me about AWS, React, and Pizza

Transitioning to Collaborative Projects

A Student's Journey

George's Perspective (USA)

Graduate

Work at an Internship

Study for Finals

Do a Group Project

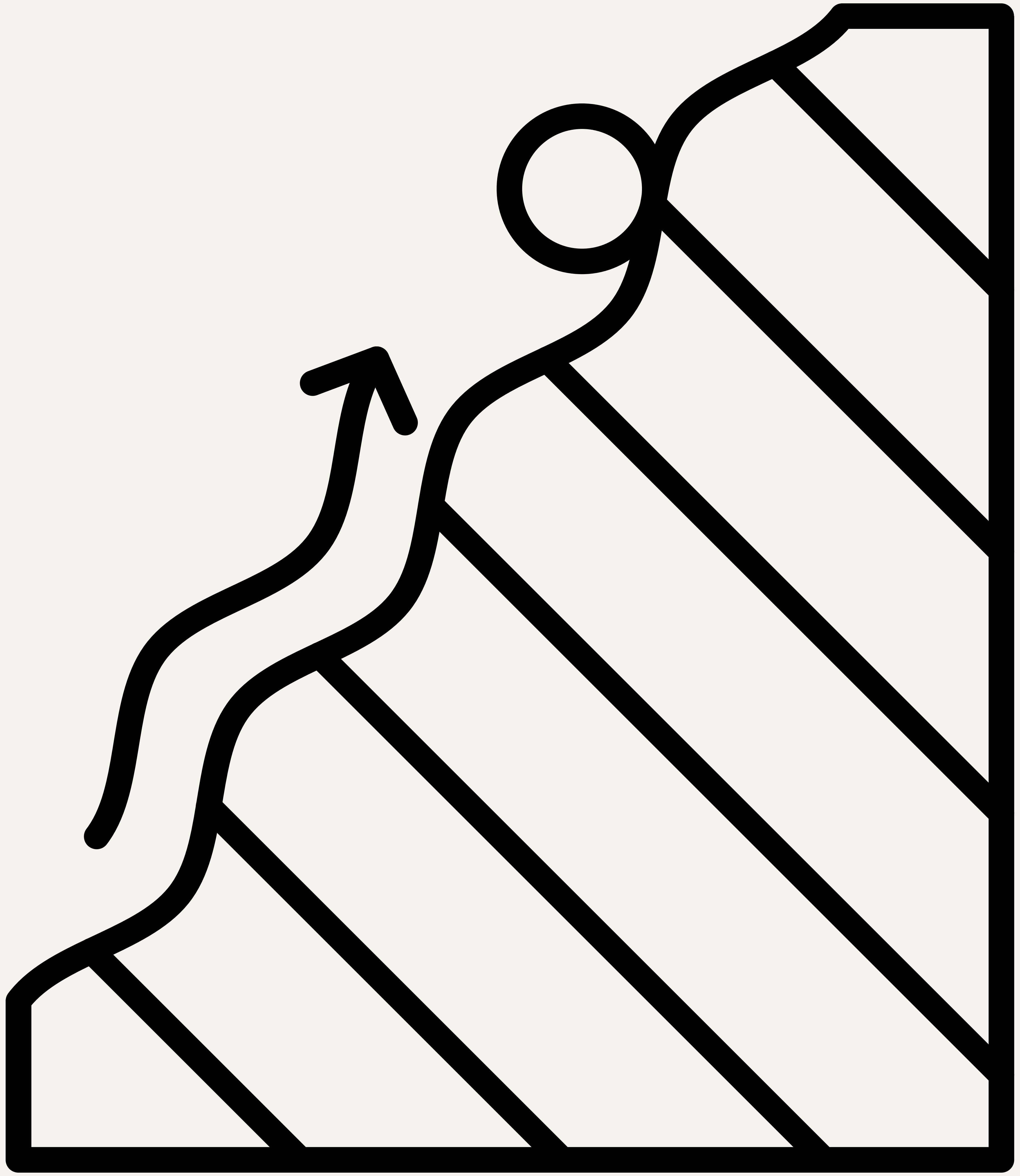
Study for Midterms

Get Admitted to a University

Take college admissions exams

Study flash cards

Learn to read



A Student’s Journey

George’s Perspective (USA)

Graduate

Work at an Internship

Study for Finals

Do a Group Project

Study for Midterms

Get Admitted to a University

Take college admissions exams

Study flash cards

Learn to read

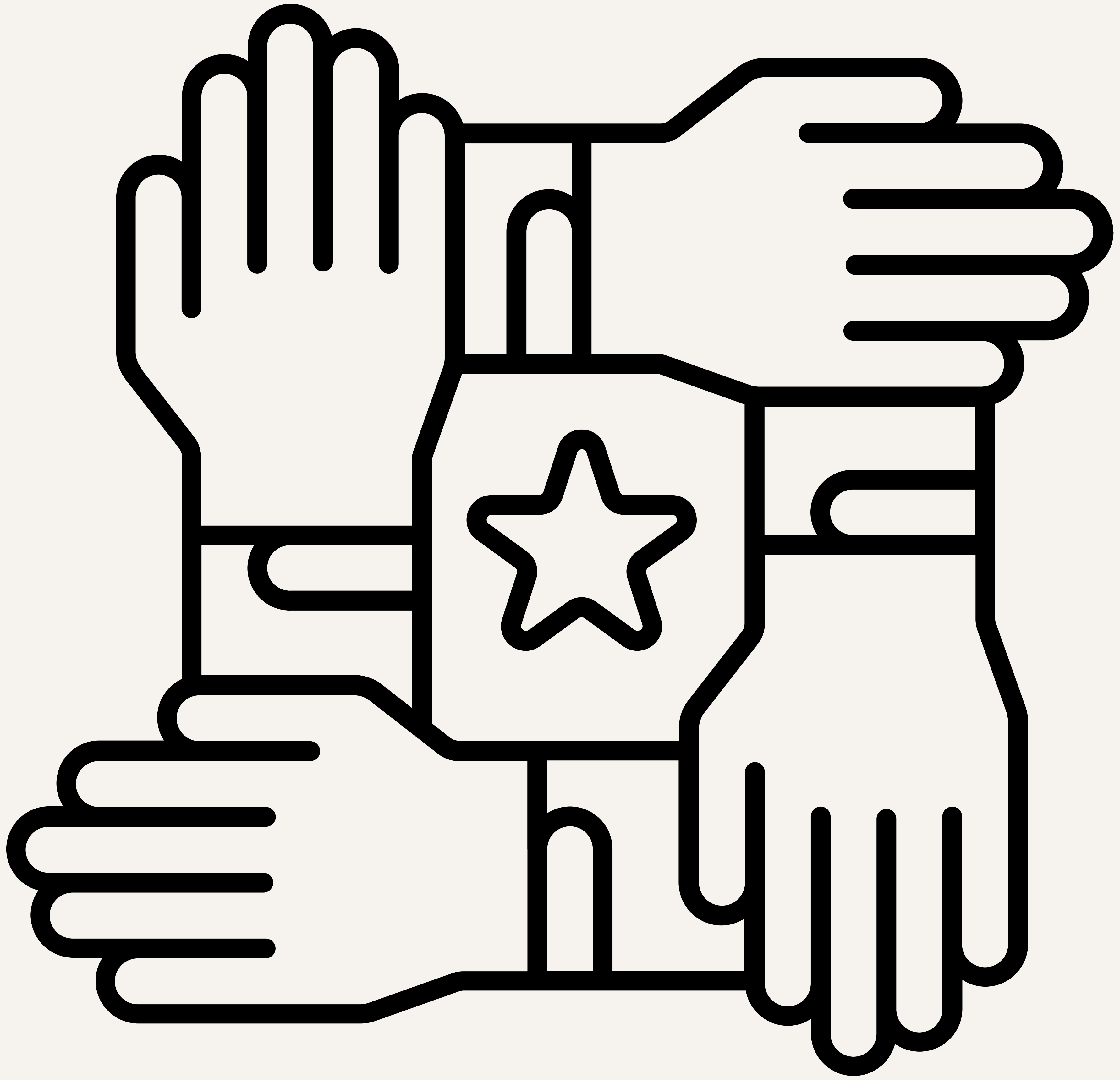
The student’s journey is focused on individual mastery of course material

What is the Objective of Collaborative Work?

Use everyone's unique abilities to accomplish a goal together!

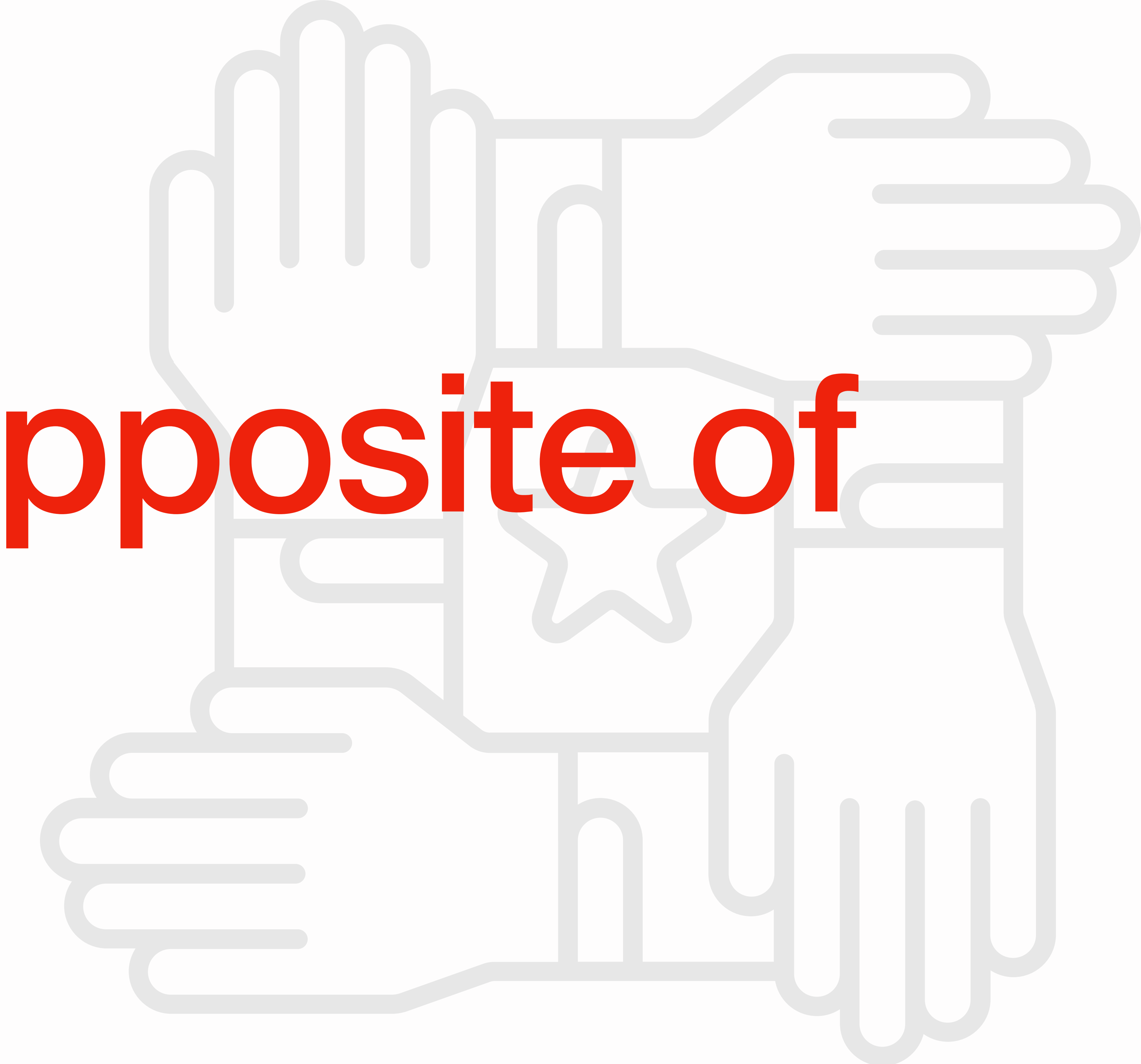
Adopting a Collaborative Mindset

- Be vulnerable - Your teammates want to help you accomplish your goals
- Over communicate your progress and updates - You might think you’re on the right track, but maybe you aren’t.
- Help your teammates - when your teammate is on the group chat looking for help be there for them
- Always be learning - You’ll pick up more knowledge by taking on a challenging task.



Adopting a Collaborative Mindset

- Be vulnerable - Your teammates want to help you accomplish your goals
- Over communicate your progress and updates. You might think you're on the right track, but maybe you aren't.
- Help your teammates - when your teammate is on the group chat looking for help be there for them
- Always be learning - You'll pick up more knowledge by taking on a challenging task.



Collaboration is the opposite of homework!

How Software Project Teams Work Together

Introduction to Agile

agilemanifesto.org

- **Individuals and Interactions** over processes and tools
- **Working Software** over comprehensive documentation
- **Customer Collaboration** over contract negotiation
- **Responding to Change** over following a plan

In Practice - A Typical Software Team

Product Owner

- Voice of the customer
- Makes decisions on software requirements
- Provides feedback on in-progress software
- Accepts the “finished” product

Engineers

- Architect a solution that meets the PO’s requirements
- Test their own work
- Provide value recommendations
- Implement the Project

Engineering Manager

- Provides guidance on software architecture
- Communicates project status to stakeholders
- Coaches engineers on opportunities for growth

An Engineer’s Day

Task	Hours	Frequency
Coordinate with teammates	0.5	Daily
Clarify business requirements	1-2	Weekly
Architect a Feature - Get consensus on scope	2-4	Weekly
Review Code	2	Daily
Write Tests	2	Daily
Write Code	2	Daily
Demonstrate work in Progress	1	Weekly

Collaboration Tools

What Should I Be Working On?

Kanban Board

How Do I Know My Code Is Current?

Version Control Systems

How Do I Get Approval or Provide Feedback?

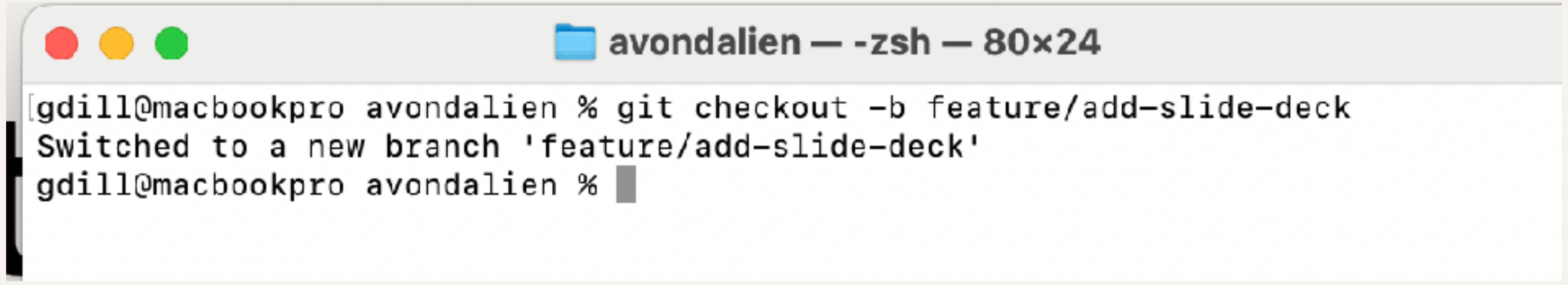
Git Hosting and Collaboration Platforms

How Do I Communicate?

Messaging Software

Demo

1. Create a new branch with a descriptive name



```
[gdill@macbookpro avondalien % git checkout -b feature/add-slide-deck
Switched to a new branch 'feature/add-slide-deck'
gdill@macbookpro avondalien %
```

Software Patterns for Collaborative Projects

SOLID Principles

- Single Responsibility
- Open-Closed
- Liskov Substitution
- Interface Segregation
- Dependency Inversion

Test Driven Development

Try Writing Your Tests First

Recommended Reading

- Clean Code - Bob Martin
- The Agile Manifesto
- The Phoenix Project - Kim, Behr, et al.
- The O’Reilly’s manual for your favorite programming language