Your Software Engineering Action Plan

The Path to Becoming a Professional Software Engineer

If you're trying to break into software engineering, this is the most direct path. We recommend following these four steps in order.

Step 1: Learn to Read and Write Code

Before anything else, you need to understand the fundamentals of programming. You need to be able to read code, understand what it does, and write your own.

Recommended Resource: Scrimba

Scrimba offers interactive coding tutorials that let you practice as you learn. This is where you'll get comfortable with the basic syntax and concepts of programming.

Step 2: Learn the Basics of Web Systems

Once you can read and write code, it's time to understand how web applications actually work. You need to know how the frontend, backend, and database all connect together.

Recommended Resource: Full Stack Open

Why web? The web is the most popular and growing platform for distributing software ever made. It's not going away anytime soon. This is where the jobs are, and where you can have the most impact.

Step 3: Build Projects (And Ship Them)

This is where the real learning happens. Build as many projects as you possibly can.

The key: Make them as real as possible.

- Deploy them on the internet
- Get them in front of real people

- Demo them to others
- Get actual users
- Iterate based on feedback

There is simply no alternative to the feedback loops you get by deploying your software into the real world.

AI Tools Belong Here

These days, most professional software engineers use AI tools to ship their projects. And so should you.

- Get a **Claude Code Max** subscription at this stage
- Build things with it consistently
- Learn to manage AI agents
- Use AI as a pair programmer, not a replacement for understanding

Important: You need Steps 1 and 2 first. AI tools are incredibly powerful, but only if you understand what the code is doing and how systems fit together.

Step 4: Create Addictive Feedback Loops (The Secret Sauce)

Here's what you'll notice: **It's very difficult to stay motivated doing all of this on your own.**

The secret to success in software engineering isn't just knowing how to code—it's creating systems that keep you learning and improving every single day.

What you need:

- **A community of practice** – Other people learning and building alongside you
- **A mentor** – Someone who's already where you want to be
- **Connection to the software industry** – Practicing professionals who can give you rapid feedback

Why this matters:

- It's easy to lose motivation when working alone
- It's easy to get lost in the details
- It's easy to build things you think are good, but wouldn't actually work
- It's easy to make simple errors that waste weeks of your time

When you have a community, mentor, and professional network, you get direct, actionable feedback every single day. You can keep looping and become better and better as an engineer.

How Long Will This Take?

For most people, learning to code takes about **1,000 hours of deliberate practice**.

That's not passive watching of tutorials. That's active building, debugging, and shipping real projects.

Find Your Motivation

It can't be all about the money. The field is changing so rapidly, and there's so much to learn, that you need some kind of intrinsic motivation.

What excites you about software? What problems do you want to solve? Who do you want to help?

Now is the most exciting time in history to be a software engineer because of all the new tools available. But it's also one of the most confusing times. Having a clear path and strong motivation will make all the difference.

Bonus: What About After Learning the Basics?

Once you've mastered the fundamentals, you might want to go deeper in specific areas:

Computer Architecture

If you want to understand how computers actually work at a low level:

- **Resource:** NAND to Tetris

Computer Science

If you want a deeper understanding of algorithms, data structures, and CS fundamentals:

- **Resource:** CS Primer
- Note: This will take a couple hundred hours, and you'll need the basics of coding to make any progress

Machine Learning & AI

If you want to understand how modern AI actually works:

- **Resource:** Andrej Karpathy's "Zero to Hero" course
- Write all the code yourself
- This alone will give you a solid foundational understanding of modern machine learning

---

About Fractal Tech

At Fractal Tech, we provide everything in this action plan:

- **750 hours of in-person training** plus prep time
- **Professional mentors** who work in the industry
- **A community of practice** with daily feedback loops
- **Open source curriculum** (you can view it anytime)
- **Goal:** Get you from zero to professional software engineer in 3 months

Our Curriculum

All of our curriculum is open source. If you want our opinions on how to do any of this:

- Check out our pre-course repository for people preparing for our AI engineering program
- Check out our main curriculum repository with all our cohort materials

The main repo isn't organized for a general audience yet, but you're welcome to look at it as a resource anytime.

Is Fractal Tech Right for You?

Book a call with us. We'll talk through whether this career transition makes sense for you.

We can help you figure out if now is the right time, whether you have the right foundation, and how to structure your learning for success.

Book a Call

Remember

The path is clear. It's just profoundly difficult to do alone.

You can absolutely teach yourself to code by following these steps. But having mentors, community, and daily feedback loops will dramatically accelerate your progress.

That's what we're here for.