



The Career Hacker's Guide To

# Software Engineering Careers

# “Software is eating the world.”

Marc Andreessen's quote from nearly a decade ago seems prophetic today as the number of software tools and companies continues to grow rapidly.

Software touches many aspects of our lives. Today, computer software allows us to connect with anyone, anywhere, anytime. And its impact is continuing to offer people the world over ways to improve their standard of living, solve complex problems, obtain information, and connect with others.



But none of this would be possible without the people behind the scenes who turn ideas into applications: software developers.

So, could software development be for you? Well, that's what we created this resource guide for—to help you gain an overview of what a career in software development could be like, especially just starting out.

# Quick navigation

<b>“Software is eating the world.”</b> .....	<b>2</b>
<b>So...what is software engineering, anyway?</b> .....	<b>4</b>
<b>What career paths are there in software?</b> .....	<b>6</b>
<b>How much money do developers make?</b> .....	<b>10</b>
<b>How to get started in software</b> .....	<b>11</b>
<b>The skillset for a career in software</b> .....	<b>13</b>
<b>Resources for learning more about a career in software engineering</b> ...	<b>15</b>
<b>The best companies to work full-time in software development</b> .....	<b>19</b>
<b>Software career success stories</b> .....	<b>20</b>

# So ...what is software engineering, anyway?



If you check Wikipedia, you'll read that software development "is the process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components."

But what does that mean in non-jargon?

Simplified, software development is the full process of transforming an idea into a digital application you can interact with. Sure, there's more nuance to it than that. But the basics of software development involve creating. Sometimes that means creating something from scratch. Other times it means iterating on something someone else has already created.

Software development has dramatically changed since its origins. The cost of developing software has also dramatically declined. Where people once used punch card machines to write computer programs, today, anybody with a laptop can get started learning.

The benefits of software development are all around us. It's what allows me to write this sentence on my Macbook and serve it up to you in the cloud so you can read on your smartphone. Software development is what's enabled us to connect with people from across the globe as if distance no longer mattered.

As Isaac Newton once said, “If I have seen further than others, it is by standing upon the shoulders of giants.” In a way, software enables us all to stand on the shoulders of giants—by giving us access to information and tools that previous generations could only dream of.

But, software is very much still in its youth. As technology continues to advance, the impact of software on our lives continues to grow. Keeping in step, the number of software developers in the world continues to grow, too—with [millions of people](#) entering the field each year.

This trend is to the advantages of new, aspiring developers—it’s never been easier to break into software development than it is today. There are countless bootcamps, training programs, and apprenticeship opportunities for anyone to break into software development.

It’s an exciting time to take part in. And if it interests you, well, then read on, because this page is just for you.

# What career paths are there in software?

There are a number of different career paths in software development. The path you choose can largely determine your options, but it's also okay to choose more than one. With software development, you can choose to specialize or be more of a generalist—but both come with tradeoffs. Below, we've listed some common types of software development paths. The path toward upward mobility will slightly vary for each of these, but in the table below, we've highlighted common career progressions across software development.

But before we get started, a few words of advice...

A good place to start before jumping down a specific path is to think about where you want to go and what type of developer you want to be. Entry-level programming jobs could mean any type of programming—they vary so much.

So instead of trying to learn all the languages, it's good to find the paths most interesting to you, then look for overlaps in languages used by programmers on those tracks.

For example, if being full-stack developer sounds interesting, where you design apps and visuals and UX but also do back-end work and use databases, learning JavaScript is a good option because it is used on both front-end and back-end development. Similarly, if you know you like data science and want to be more appealing to an employer, learning Python is a good choice because it's used in both data science and back-end development.

Especially at a startup, learning continually and building your skills in different areas and languages will help you ship in other areas inside the company—thus making you incredibly valuable to any company.

Here are some common types of web development—and they're great to launch your career in.

## Front-End Web Development

Front-end programmers build the visual part of any site. Instead of the web being full of white pages with text running from one edge of your screen to another, front-end web developers build and style websites so the user gets a stunning experience.

*Languages: HTML, CSS, JavaScript & its frameworks*

A note on frameworks, like React.js: if you can learn any JavaScript framework and show that, it can help you win a job—even if the company you're interviewing with doesn't use that specific framework—because they see you can learn.

## Mobile Development

Mobile development is a great path for anyone who wants to build the operating systems running on any mobile device. If working on the software running daily on your phone—and the phones all around the world—sounds exciting, it's good to focus on choosing one primary OS to focus on (iOS, Android) as their languages vary.

*Languages: Swift, Objective-C, Java; iOS SDK and Android SDK; Dart/Flutter*

## Back-End Development

Back-end developers connect the work front-end developers do (the part of a site you see) with servers so websites and web applications run and function properly. This is the core part of any website or web application—the web wouldn't run properly without this type of development! Back-end developers make sure processes run logically and often write documentation so others can use the systems they've built.

*Languages: JavaScript, Ruby, PHP, Django & more*

## Data Science

Data scientists extract meaning from and explain data. But they aren't your typical white-coat scientists—they are the people who strive to make sense of any amount of information so other parts of an organization can use that information in the best ways possible. At their core, they work to help businesses determine goals and make wise decisions based off of what they discover.

It's helpful to have a solid understanding of mathematical concepts if you want to go into data science—a background in linear algebra and matrix math is especially valuable.

*Languages: Python, R, SQL, Java, Julia, TensorFlow, & more*

To highlight a few of the common ranges, here's a breakdown by experience level, courtesy of Payscale.



## Positions

---

**Entry-Level  
Positions**

---

**Senior  
Positions**

---

**Management  
Positions**

---

**Senior  
Management  
Positions**

---

**C-Suite  
Positions**

## Common Roles

---

Junior Developer

---

Software Developer  
Sr. Software Developer  
Technical Architect

---

Development Team  
Lead,  
Software Development  
Manager,  
Lead Engineer,  
Lead Developer

---

Director of Software  
Engineering,  
Vice President of  
Engineering

---

Chief Technology Officer  
Chief Product Officer

## Average Years of Experience

---

0-3

---

3-7+

---

5-10+

---

10+

---

10-20+

## Average Income

---

\$43k-\$79k  
Mean: \$60k

---

\$84k-\$153k  
Mean: \$115k

---

\$81k-\$154k  
Mean: \$115k

---

\$110k-\$233k  
Mean: \$167k

---

\$89k-\$240k  
Mean: \$159k

# How much money do developers make?

A career in software development offers substantial income trajectory. That said, how much money you can make will largely depend on the path you choose, the type and stage of company you work at, and whether or not you earn equity in addition to your compensation.

While base earnings can still amount to large sums, the lavish get-rich stories of Facebook or Google software engineers who became multi-millionaires are because of their equity earnings. Equity is essentially a percentage of company ownership offered in addition to salary, in many cases offered as stock options.

Beyond equity earnings, though, software development offers strong earning potential on a pure income basis. While we're on the topic, here's a breakdown of some of the most common income ranges by experience level, courtesy of Payscale.

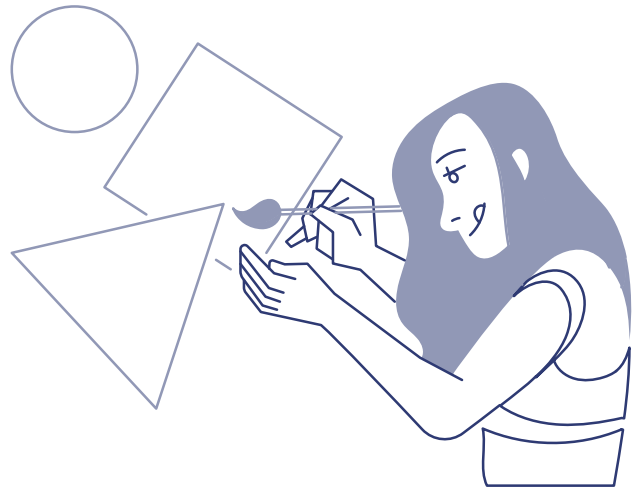
## Average Income

Entry	Senior	Management	Sr. Management	C-Suite
\$43k-\$79k Mean: \$60k	\$84k-\$153k Mean: \$115k	\$81k-\$154k Mean: \$115k	\$110k-\$233k Mean: \$167k	\$89k-\$240k Mean: \$159k

See more details on the Payscale reports for [entry-level](#), [senior](#), [management](#), [senior management](#), and [C-suite](#) engineering salaries.

# How to get started in software

If you're itching to get your hands dirty on some code, all you need is an internet connection. You'll find countless resources online that teach you to code. (You've also got the whole world of no-code app development at your fingertips.) Coding bootcamps are also getting more student-friendly as they grow in number and compete with each other.



## Teach Yourself Code (or No-Code)

The quickest way to learn anything, whether you're in high school or already have your master's degree, is to start dabbling in it right now. This philosophy will get you far in life, and it's just as applicable to software development as anything else. Thanks to the amazing internet on which you're reading this article, you can start learning software development languages that matter to potential employers and clients.

Websites like [Codecademy](#), [freeCodeCamp](#), [Codewars](#), and [GA Dash](#) are great places to start writing code from day one. Get familiar with [Stack Overflow](#) as soon as possible. You can also take free and paid courses on sites like [Udacity](#), [Treehouse](#), and [Coursera](#). Those platforms also have tons of resources for learning no-code, as well

as sites like [MakerPad](#), [Zeroqode](#), [Nucode](#), and many more.

If you're trying to teach yourself to build apps, remember that the quickest way to learn and improve is to just build. Start with baby steps, like this awesome [React.js tutorial](#) that walks you through building a tic-tac-toe game. Walk before you run, practice every day, and before you know it, you'll be running full speed ahead!

## Get Into a Coding Bootcamp

Coding bootcamps are all the rage these days. You might have heard of [Lambda School](#), [Galvanize](#), [General Assembly](#), [App Academy](#), and many others. You might even be one of the [23,043 students estimated to have graduated in 2019](#) or one of the tens of thousands more who will graduate from a part-time or full-time program.

Coding bootcamps are great because they fully immerse students into the work, focusing specifically on applied learning rather than pure theory like many computer science classes in universities. They teach computer engineering, information technology, successful software habits, software design, computer science, and many more related fields.

The best coding bootcamps are [free until you get a job](#) making enough money to start paying them back. This new model uses [income share agreements](#) (ISAs) to de-risk the experience for students and align the bootcamp's incentives—meaning, teaching computer programming languages that are in high demand in the real world. Here's [a helpful piece](#) on how to launch your career through a coding bootcamp by showing your work.

# The skillset for a career in software

Coders have so much more to them than only coding languages—they are people who drill down into problems and figure out what the solution actually is, then take action on discovering or creating that solution.

Prospective software engineers should focus on building this skillset:

- Excellent communication skills
- Ability to take issues and break them down into smaller issues
- Logical approach to problem-solving
- Research skills (Google-fu)
- Ability to ask good questions
- Precision of language to provide clarity for complex, technical topics
- Patience

# Most common engineering tools

These are some of the top tools software engineers use.

## Version Control



## Code Editors



## Deployment



## Communication



## No-Code



## Prototyping



# Resources for learning more about a career in software engineering

The internet is full of resources on software development, perhaps second only to productivity or personal development. Here's a compilation of the best ones we've found so far.

Enjoy!

[Podcasts](#) • [Books](#) • [Leaders](#)

# The best software podcasts to start your career

Like to take your learning on the go? Us, too.

And if you're looking for some great places to get an extra-heaping dose of engineering insight, then we highly recommend you check out some of our favorite dev podcasts:





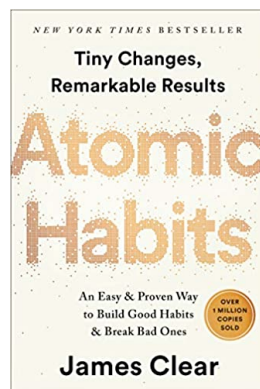
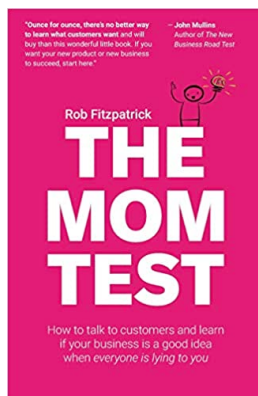
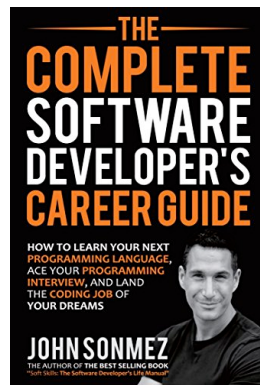
# The best software books to start your career

As United States President Harry S. Truman once notably said, “Not all readers are leaders, but all leaders are readers.”

The same goes for anyone who wants to launch a legendary career in software engineering. But before you slip and fall running as fast as you can to add every top Amazon book result for “engineering” to your cart, take a deep breath.

Then, check out our essential reading list for every aspiring young software engineer:

Click any book to open a preview online.



# The best software leaders to follow as you start your career

Okay, okay, so this is a list that could easily go on forever.

Which is an awesome thing when you think about it—because it means there are so many phenomenal engineers out there you can start learning from right now.

So, while it's not a comprehensive list of every great software leader you should follow, we think it's a pretty good starting point.



Yehuda Katz



Cassidy  
Williams



Sara  
Soueidan



Sahil Lavingia



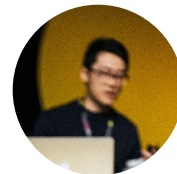
Courtland  
Allen



Sarah  
Drasner



David  
Khourshid



Evan You

Know someone who should be on this list? Tweet their name and why you like them to us @HackingCareers, and we just might add them.

# The best companies to work full-time in software development

Of course, there's Apple, Google, Amazon, Facebook, and the rest of the holy grail companies everyone thinks of. But we're also featuring some lesser-known companies making waves in the software space, as well as some of the top-rated ones on Glassdoor.



Stripe — Very hot company, famous for being developer-friendly



Microsoft — Open source, famous leader in software industry



Webflow — Rising star in no-code industry, remote-friendly



Figma — Constantly releasing new features, hot design tool



Intercom — Great design and awesome customer support



Vercel — Small, scrappy, built for developers, simplifying their industry



Netflix — Pioneer of video streaming subscription, innovating quickly



Lululemon — Top-rated place to work, pioneer of “athleisure wear”



Adobe — World-famous, top leadership, top-rated on Glassdoor

# Software career success stories

## How Austin Tackaberry Landed a Six-Figure Software Engineering Job in Nine Months

*“In no other engineering discipline can you just have an idea, start building it, show it to users, and iterate with little capital and low barrier to entry. In chemical engineering, you essentially need a running plant or a lot of money to design a plant if you had an idea for a new product.”*

[Read more of Austin's story here.](#)

## How Madison Kanna went from Complete Beginner to Software Developer

*“Two years ago, I was right where you are today. I wanted to become a professional programmer. But I had no idea how to make it happen. I had no college degree, no previous coding experience, and I sucked at math. And there was the nagging doubt: can someone like me become a developer? Well, I made it happen. I have my dream job. I'm a software developer.”*

[Read Madison's story here.](#)

# How Jonah Grimm Pursued His Dream of Becoming a Developer and Launched His Career

Jonah started his career working at Chick-Fil-A—but he always wanted to be a coder. So he started coding—and now he's doing it full-time for a startup while working on building a gaming company on the side. [Here's an interview](#) we did with him to hear his story.

# Final resources

Congrats on making it this far!

*Subscribe to the Daily Job Hunt newsletter. Join 200,000+ readers and get a kick in the inbox every morning.*

*Read the Career Hackers Manifesto: The core principles of discovering and doing what makes you come alive.*

# THE DAILY JOB HUNT

 **careerhackers**  
manifesto

*And check out the other career-hacking guides in this series:*

