SOFTWARE ENGINEERS DO WHAT NOW?

By Shaun Michael Stone

First Edition UK - 2020

Table of Contents

Creative Commons Notice

Every precaution was taken in preparation for this book. However, the author assumes no responsibility for errors or omissions, or for damages that may result from the use of information.

Please note: All facts, claims of the use of languages and their industry relevance are based on the time of writing and circumstances can always change.

GitHub - https://github.com/smks

Twitter - https://twitter.com/shaunmstone

YouTube - https://www.youtube.com/c/OpenCanvas

LinkedIn - https://www.linkedin.com/in/shaunmstone

Introduction

> "You can't be in the tech community... without realizing there's a big shortage of talent."
>
> – *Mitch Kapor*

The tech industry in the UK is thriving. Start-ups, large corporate entities and everything in between are all screaming out for tech specialists in areas of Software, Artificial Intelligence and Financial Tech (FinTech). The UK–especially London–is inhabited with skilful tech-savvy individuals who can bring a lot of innovation to the table. Companies yearn for these individuals, but there's a problem.

The demand doesn't match the supply. There's a lot of jobs out there, therefore, it's a battle for companies to find people–the right skilled people anyway. It's easy to recruit someone, but it's hard to recruit someone who matches the role's requirements. That's why I decided to write this book, to encourage anyone out there who fancies a career change, or is fresh out of University and needs a bit of guidance or direction.

With this book, we will introduce you to the variety of technical roles out there, the positions that exist on the career ladder and make our way through an abundance of sought after technical languages, tools, libraries and frameworks that companies seek. If you are interested in a career in software or just looking to understand the tech side of things, then this is the book for you. Feel free to cherry-pick the bits that interest you more.

End Goal

You will get a feel and basic understanding of the tech that is out there. It may give you a kickstart and the motivation to pursue a career or hobby in software engineering yourself.

This book summarises each of the technologies at a high-level and does not go into real depth. There are *entire* books likely dedicated to each one of them!

Book Coding Style

In this book, you will see code examples in one of the programming languages. What you need to know at this point, is whenever someone creates a source file, it's simply a file on your computer. The file gets opened in a text editor and the developer types letters, symbols and numerical values into this file in such a way that it can be interpreted by the language they are writing in. When the file is saved, it's given an extension to understand what type of source file it is.

```
my-file.js
```

I am giving the file an identity, essentially a passport so anyone who sees it, knows what that file's purpose is. In the example above, it's a JavaScript file.

This book uses examples when working on a Mac and sometimes Windows.

Some of the code examples may wrap onto the next line due to spacing limitations.

Code snippets

When I want to demonstrate to you what is inside a file, I present it in the following way.

my-file.py

This is where I introduce you to what on earth is going on.

```
# start of script
print("hello, hola, ciao");
```

Above is the first bit of code. This is where I bore you with the details of what's going on, or what will happen next.

```
print("goodbye, adios, addio");
# end of script
```

I may or may not show you the output of running the script. Depends if I'm feeling cheeky.

The output will show like so:

```
OUTPUT:
hello, hola, ciao
goodbye, adios, addio
```

All of the code examples can be found in the repository below. Feel free to contribute to this project.

https://github.com/smks/sedwn-code-examples

Prerequisites

1. A Laptop or Desktop

2. Internet access for further research on the technologies

3. Motivation to finish the book. You can do it!

Assumptions

The only thing this book assumes is that you know how to read and put up with my lame jokes. Sorry in advance…

Please note: The book has been designed *not* to expect you to carry out the code examples, but to give you a feel for the subject in question. It may help you understand what area of software you may want to follow or allow you to understand things a bit better when recruiting someone with those specific skills.

Suggestions

You can contact me via any of the social networks if you'd like to give me feedback good or bad (fingers crossed). I'm open to adding or deprecating topics as time goes on for new editions of this book.

- GitHub - https://github.com/smks
- Twitter - https://twitter.com/shaunmstone
- YouTube - https://www.youtube.com/c/OpenCanvas

Or connect with me on LinkedIn for business-related requests.

LinkedIn - https://www.linkedin.com/in/shaunmstone

Also, *please* post a picture with you holding the book. On social media or by email. Regardless, it will *really* make my day! That way, I know it isn't just robots reading it, but actual human beings. Wow, imagine that!

Structure

The book is broken into four parts:

Part 1 - Software Careers

The first part focuses on the software industry ranging from the types of roles out there, recruitment, and what a typical day as a software engineer looks like.

Part 2 - Terminology

The second part is centred around programming and testing terminology used in the industry.

Part 3 - Programming Languages

The third part is a collection of programming languages used by software engineers. This isn't an exhaustive list, but a majority of the most common languages used commercially today.

Part 4 - Web Libraries & Frameworks

The fourth part is focused on web-related libraries and frameworks.

Software careers

Here we will observe 'my' career to date, the career paths of individuals in general and take a look at the types of technological roles that are in demand from companies.

- My career experience to date
- Career paths of a software engineer
- The software engineering roles out there
- Agile software development process
- Reasons why software engineers remain at their jobs
- In-depth hiring process of a software engineer
- CV screening tips
- Face to face interview pitfalls
- How tech recruiters stand out
- Learning how to code

My career

> "Choose a job you love, and you will never have to work a day in your life."
>
> – *Confucius, Chinese philosopher*

I love the quote above! Before we delve into the book, I wanted to discuss my experience in the tech industry for anyone interested in following the same path. Feel free to skip if it's of no interest to you. I won't be offended. Okay maybe a little bit, but passively.

Hello

My name is Shaun Stone. I'm a Front-end Tech lead from London, UK. I work in the FinTech (Financial Tech) industry. To anyone out there who wants to pursue a career in web development, you may find this useful. I really enjoy what I do, it presents so many challenges I have to tackle. I get to do fulfilling work, mentor and also learn from others, as well as see my work get used by thousands of customers.

Before I applied for a permanent role, I went to University to do a Computing course. When the course finished, I had to do the dreaded job hunt. Why was I so concerned about this? Because I had a degree… but no commercial experience.

To solve this, I started looking for freelance work off my own back, where I would work for next-to-nothing just to get the experience I yearned for. I found some work via Reddit and PeoplePerHour and spent hours working on each project, which I was ecstatic about because I was earning money. On reflection, I would say University helps, but it's not essential.

First role

My first role was at a software house that catered for automotive dealership clients. I started out as a PHP developer and worked on big sites such as Mercedes retail group (was never given a Mercedes unfortunately), Nissan, Mazda and Lookers. Being my first role, I was exposed to a commercial environment, where what I did was vital.

I had to be very articulate and careful with the development work because there was a lot at stake. It essentially threw me into the deep end. The most important thing about this role was that I learned so much from my peers. I was surrounded by smart people I could learn from. A rule I follow is I should always be learning, and if I feel like the smartest person in the room, I'll probably stop learning.

Covent Garden

I decided I wanted to work in London. That was always the plan, so I worked for a finance company based in Covent Garden where I would learn a lot about the financial sector, and I would grow from a Junior into a mid-level PHP developer. Like a Pokémon evolving from a Charmander into a Charmeleon.

I managed high-risk tasks such as sending dynamic emails out to our whole customer base and eventually working on my first User Interface (UI) focused project, which involved rebuilding the customers' web dashboard. It was because of that big project, and it being a great success that I decided I preferred working on the front-end – the visual elements of a web page.

Piccadilly Circus

I moved to a gaming studio company in Piccadilly circus, which was my first front-end development role. It was heavily focused on

working hands-on with designers who wanted their designs converted into programmed mini-games. I liked this design-to-development collaboration because things got very creative.

They asked me to implement complex animations a lot which was great as the animation was something I wanted to do as a kid. Because I had a lot of repetitive tasks in general, it inspired me to write a book called 'Automating with Node.js.'

Vauxhall

I now work as a Front-end Tech lead for an investment company in Vauxhall, London. I am involved with recruitment, leading projects, planning meetings and enforcing coding standards/conventions for development across teams. I am also collaborating with the UX/UI/Design team, something I very much enjoy. Woohoo! (Homer impression).

Personal projects

Throughout my career, I've always been busy working on my own hobby projects. This involves writing books and making games or web apps. All of these experiences have helped me grow and understand what's involved so I can foresee technical challenges for other projects I need to tackle. I think it's good to work on things for yourself, for your own self-gratification.

Career path

> "Every great developer you know got there by solving problems they were unqualified to solve until they actually did it."
>
> – *Patrick McKenzie*

Companies have their own roadmap for growth. Similar to my path, a very common pattern of progression as a software engineer is the following.

Please note: *The term developer and engineer are used interchangeably and tend to refer to the same position.*

1 - Junior developer

Junior developers have little-to-no experience. They need to be guided by more senior members of the team to do their work. They usually ask many questions but can learn a lot and are very motivated to do so. Their salary is entry-level. On a side note, companies usually rate your competency level down to the number of years of experience you have. To me, this is a fallacy. I've worked with new starters who were labelled more 'junior' but were knowledgeable and highly competent. Candidates shouldn't always be taken at face value.

2 - Mid-level developer

Mid-level developers have a reasonable amount of experience. They can achieve certain tasks on their own, but still need to look to senior members for guidance from time to time. They can assist juniors and