Sophia

Role of a Software Engineer

by Devmountain Tutorials

WHAT'S COVERED

In this lesson, you will learn how to explore the responsibilities and job characteristics of a software engineer. Specifically, this lesson will cover:

Hi, I'm Monique. I'm the Software Engineer at Poodle Jumper and I've been working here for nine months. Before I joined tech, I taught French and Spanish to high school students. I loved teaching but I needed a career with more flexibility. I enrolled at Hackbright Academy, a Software Engineering bootcamp for women based in San Francisco. Making a career pivot later in life was intimidating, but I love the community of talented engineers I joined. The schedule flexibility I have as an engineer makes it possible to raise a young family and have a career that challenges me. I've been able to leverage my teaching skills to mentor other engineers who are just starting out.

1. A Broad View

There are many niches of software engineers; one you've already learned about iOS Engineering and Web Development. The term 'Software Engineer' is broader than those roles. The low-level problems solved are very similar, but the high-level problems are different. Software Engineers have additional responsibilities to understand how computer memory works, creating logic or algorithms, and managing how software interacts across the different layers of technology. You may hear these systems referred to as the backend. Backend technology is the servers, applications, and databases that work behind the scenes to power the interface users interact with.

Software engineering is a unique discipline that combines features typically exclusive to mathematic, scientific, and creative fields. Like mathematicians, they use specialized languages to denote ideas. Like scientists, they must understand the rules that govern how code runs on computers. Like designers and engineers, they create solutions by assembling individual components into systems and evaluating tradeoffs among alternatives.

You don't have to love math, be a trained scientist, or a skilled designer to excel as a software engineer. The most important skill for a software engineer is problem-solving—the ability to formulate problems, devise creative solutions for them, and communicate those solutions effectively. The types of problems that Software Engineers encounter can vary, but the way they approach building solutions to those problems is similar.

STEP BY STEP

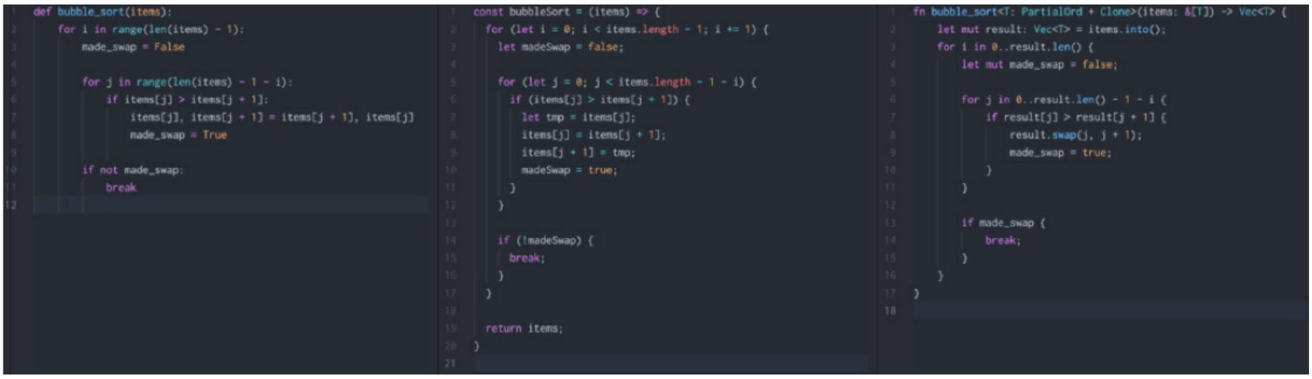
1. Develop a general solution.
2. Communicate the solution to a computer, in a way that the computer understands.
3. Use the computer to automate the execution of that solution.

At their essence, a Software Engineer writes code to solve problems and create computer systems or applications that do something useful.

2. Different Languages, Algorithms, and Servers

While Ruben focuses on the web application and Camilla focuses on the iOS apps, my role is to make the code that ties it all together, keeps it in sync, and enforces the business logic.

My experience with multiple languages has been really helpful in learning the syntax or structure of different programming languages. In my role, I work with multiple different programming languages depending on what I'm trying to do. This can be a tricky part of starting in software engineering because [brackets], {curly braces}, and 'single quotes' do different things in different languages, but like most things–it gets easier the more you work with it.



```python
def bubble_sort(items):
    for i in range(len(items) - 1):
        made_swap = False

        for j in range(len(items) - 1 - i):
            if items[j] > items[j + 1]:
                items[j], items[j + 1] = items[j + 1], items[j]
                made_swap = True

        if not made_swap:
            break
```

```javascript
const bubbleSort = (items) => {
  for (let i = 0; i < items.length - 1; i += 1) {
    let madeSwap = false;

    for (let j = 0; j < items.length - 1 - i) {
      if (items[j] > items[j + 1]) {
        let tmp = items[j];
        items[j] = items[j + 1];
        items[j + 1] = tmp;
        madeSwap = true;
      }
    }

    if (!madeSwap) {
      break;
    }
  }

  return items;
}
```

```rust
fn bubble_sort<T: PartialOrd + Clone>(items: &[T]) -> Vec<T> {
    let mut result: Vec<T> = items.into();
    for i in 0..result.len() {
        let mut made_swap = false;

        for j in 0..result.len() - 1 - i {
            if result[j] > result[j + 1] {
                result.swap(j, j + 1);
                made_swap = true;
            }
        }

        if made_swap {
            break;
        }
    }
}
```

This image shows a bubble sort algorithm in three different languages.

Bubble sort is the first sorting algorithm most developers learn because it is simple and easy for small data sets, but it is notoriously inefficient in real-world applications. It's so bad that it's become a joke in the tech community. The left section of the code is in Python, the middle is in JavaScript, and the right is in Rust. Don't worry about being able to understand the code, but you can see differences in the structure, formatting, and syntax.

The backend systems I manage include powerful computers—also known as servers—that send information (or data) to code that display for users. Ten years ago, most companies had to buy and manage expensive servers on-site, but now, most companies leverage servers from a cloud technology provider such as Microsoft, Google, or Amazon.

When someone says "It's in the cloud," what they mean is that it is on a server somewhere else. The "cloud" is a really large network of computers that talk to each other, just like the internet. As the engineer, I know where those servers and data are physically located. I deploy, or release, the updates to the code on the servers and make sure our code and data are backed up regularly in case of a service outage.

I’m responsible for managing the server’s performance and memory. I set up monitoring with alerts to tell me when something is out of the ordinary. This can happen at any time, day or night, so the team and I have an on-call rotation to make sure someone is always available to solve an issue that might come up. This is a lot of responsibility. When I’m on call, I keep my computer close and my notifications on.

☆ BIG IDEA

Are you someone who hates math, and you are afraid that it may get in the way of your ability to learn how to code? A recent study published in Scientific Reports found that language aptitude accounted for a higher variance in learning outcomes than numeracy. Don’t let stereotypes get in the way of your career options. We need more engineers with diverse backgrounds and perspectives in building technology.

SUMMARY

In this lesson, Monique, a software engineer at Poodle Jumper, explained her role and responsibilities as a software engineer. The lesson covered a **broad view** of software engineering, which involves creating logic or algorithms, managing how software interacts across different layers of technology, and understanding how computer memory works. The most crucial skill for a software engineer is problem-solving, and they use specialized languages to denote ideas, understand the rules that govern how code runs on computers, and assemble individual components into systems. Additionally, Monique discussed **different languages, algorithms, and servers** and explains how she manages server performance and memory, sets up monitoring with alerts, and deploys updates to the code on servers. The lesson emphasized the importance of diversity in the field of software engineering and encourages individuals from diverse backgrounds to consider pursuing a career in this field.

Source: This tutorial was authored by DEVMOUNTAIN and Sophia Learning. Please see ourTerms of Use.