



# Talent Acceleration Program

## Introduction



[tap.kiitos.tech](http://tap.kiitos.tech)

We are living in amazing times. Technological advances have allowed us to get instant access to entertainment, track our physical activities and communicate with people on the other side of the planet, in real time!

What has made all of this possible? Software. As a developer-to-be you will be leading the way and building the future that will make all of this and more available for generations to come. Welcome to this exciting new world!

In this introductory Workbook we will give you a primer into what it means to be a software developer, and more specifically as applied to the web, and why it's such a great field to go into.

We will do this by looking at from 3 different angles:

- [Market demand](#)
- [Career](#)
- [Challenges and Motivation](#)



*Are you ready to get started?*

# Market Demand

You hear it all the time: *every company wants to hire software developers.* It's not hard to guess why: technology is transforming every element of our lives.

Take *ordering food*, for example; we use a food delivery mobile app for creating an order, online banking to pay for it and tracking software to keep tabs on delivery schedules. All this infrastructure is born from tech. The world is moving online.

## COMPANY NEEDS

Companies want software developers, because they solve 2 major problems:

1. Software can automate business processes.

To illustrate this, let's take the example of a supermarket. This is a company specializing in selling products for daily living: fruit & vegetables, kitchenware and many more things. At the end of a week they need to restock, but how will they know what items they need to buy and in what quantity?

Of course, you could count everything manually (and hope that you're right every time!). However, a more cost-efficient way is to have a computer system that automatically keeps track of the supply in the store, as well as what's left in storage in case you need to restock sooner than expected.

This is where software helps by automating this process of keeping track!

2. Software can be sold to make money



What if the supermarket builds an app that allows customers to also buy online? This would be a great way to service more people and so generate more money for the business!

But there are other examples of companies building software products for money: video games (i.e. [Playstation](#)), online education (i.e. [Udemy](#)), music (i.e. [Spotify](#)) and many more.

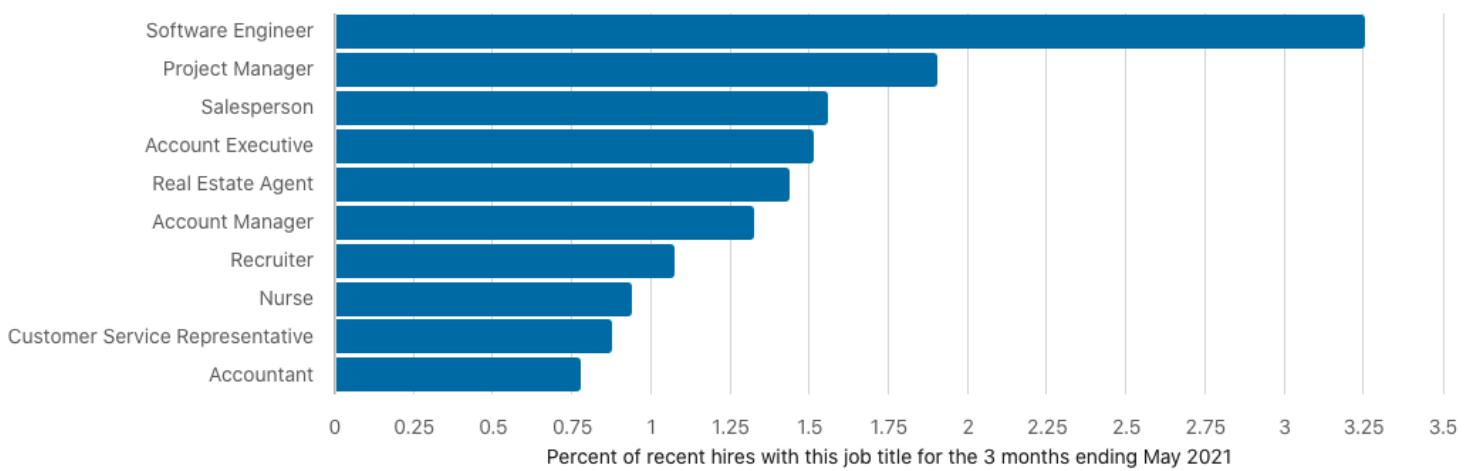
In conclusion, the acceptance of using technology to satisfy our needs is here to stay. Especially given the recent events with the pandemic. Companies can take advantage of this and go digital!

## MARKET DEMAND

As a result of these changes in company needs there's a very high demand for software developers globally. The following figure shows the amount of trending jobs that are searched for on LinkedIn, the world's largest professional networking site:

### Top Trending Jobs

*Most popular job titles among LinkedIn members hired in all industries in all regions, United States.*



(source: [LinkedIn Graph](#))

And in this figure below you'll find the top 20 job roles that will both increase, as well as decrease over the course of the next few years. Look at how many of the roles with increasing demand have to do with software development!

FIGURE 22

Top 20 job roles in increasing and decreasing demand across industries

↗ Increasing demand	↘ Decreasing demand
1 Data Analysts and Scientists	1 Data Entry Clerks
2 AI and Machine Learning Specialists	2 Administrative and Executive Secretaries
3 Big Data Specialists	3 Accounting, Bookkeeping and Payroll Clerks
4 Digital Marketing and Strategy Specialists	4 Accountants and Auditors
5 Process Automation Specialists	5 Assembly and Factory Workers
6 Business Development Professionals	6 Business Services and Administration Managers
7 Digital Transformation Specialists	7 Client Information and Customer Service Workers
8 Information Security Analysts	8 General and Operations Managers
9 Software and Applications Developers	9 Mechanics and Machinery Repairers
10 Internet of Things Specialists	10 Material-Recording and Stock-Keeping Clerks
11 Project Managers	11 Financial Analysts
12 Business Services and Administration Managers	12 Postal Service Clerks
13 Database and Network Professionals	13 Sales Rep., Wholesale and Manuf., Tech. and Sci.Products
14 Robotics Engineers	14 Relationship Managers
15 Strategic Advisors	15 Bank Tellers and Related Clerks
16 Management and Organization Analysts	16 Door-To-Door Sales, News and Street Vendors
17 FinTech Engineers	17 Electronics and Telecoms Installers and Repairers
18 Mechanics and Machinery Repairers	18 Human Resources Specialists
19 Organizational Development Specialists	19 Training and Development Specialists
20 Risk Management Specialists	20 Construction Laborers

(source: [Future of Jobs Survey 2020, World Economic Forum](#))

In short, you've made a great decision to start a career in software development!

## DEMAND IN PALESTINE

Unfortunately, there are not that many statistics available for Palestine just yet. According to the [Palestinian Central Bureau of Statistics \(PCBS\)](#), the CAGR of the local ICT market is around 12,7% and was worth \$593,2 million in 2011.

Although current data do not allow for an accurate assessment of the current economic size of the ICT sector, the sector has seen huge growth in recent years. This is especially applied to the outsourcing sector: companies from all over the world who want to have their software development done in Palestine. This is a great opportunity to work on an international level!

## SALARIES

How is all this demand reflected in the salary? This question depends on several factors:

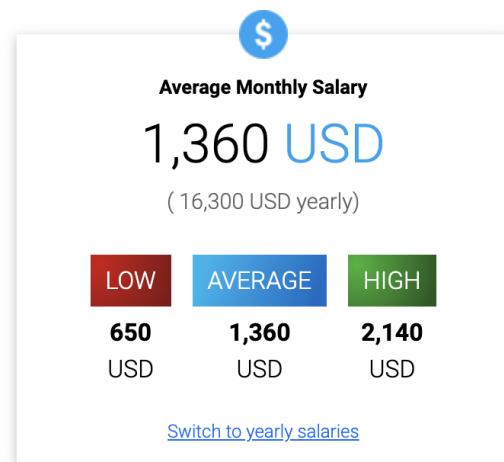
- Geographic location
- Economic status of the country
- Company size
- The employee's experience level
- Negotiation skills
- Skills asked for (i.e. programming language)

That said, there are several statistics we can look at to get an idea.

### Palestine Salary Range

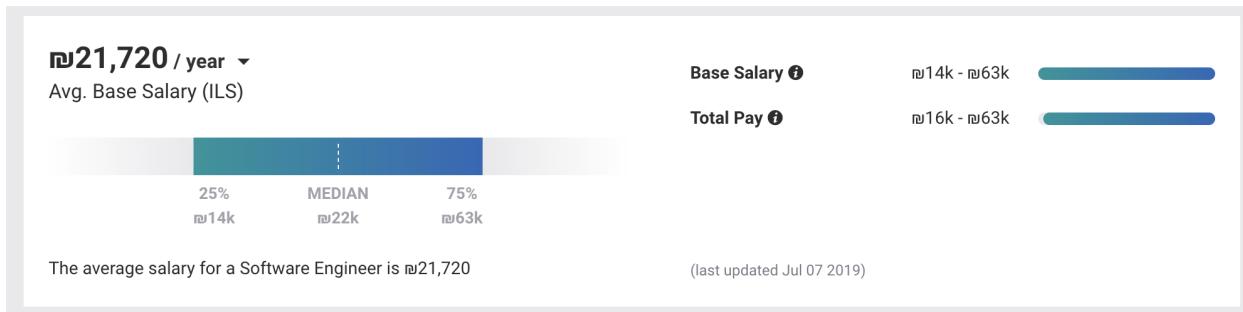
Currently in Palestine, the salary range for a software developer between 0-2 years is around **\$1300**.

The factors mentioned above play a huge role in this, and so one entry level developer might earn a little more or less than another.



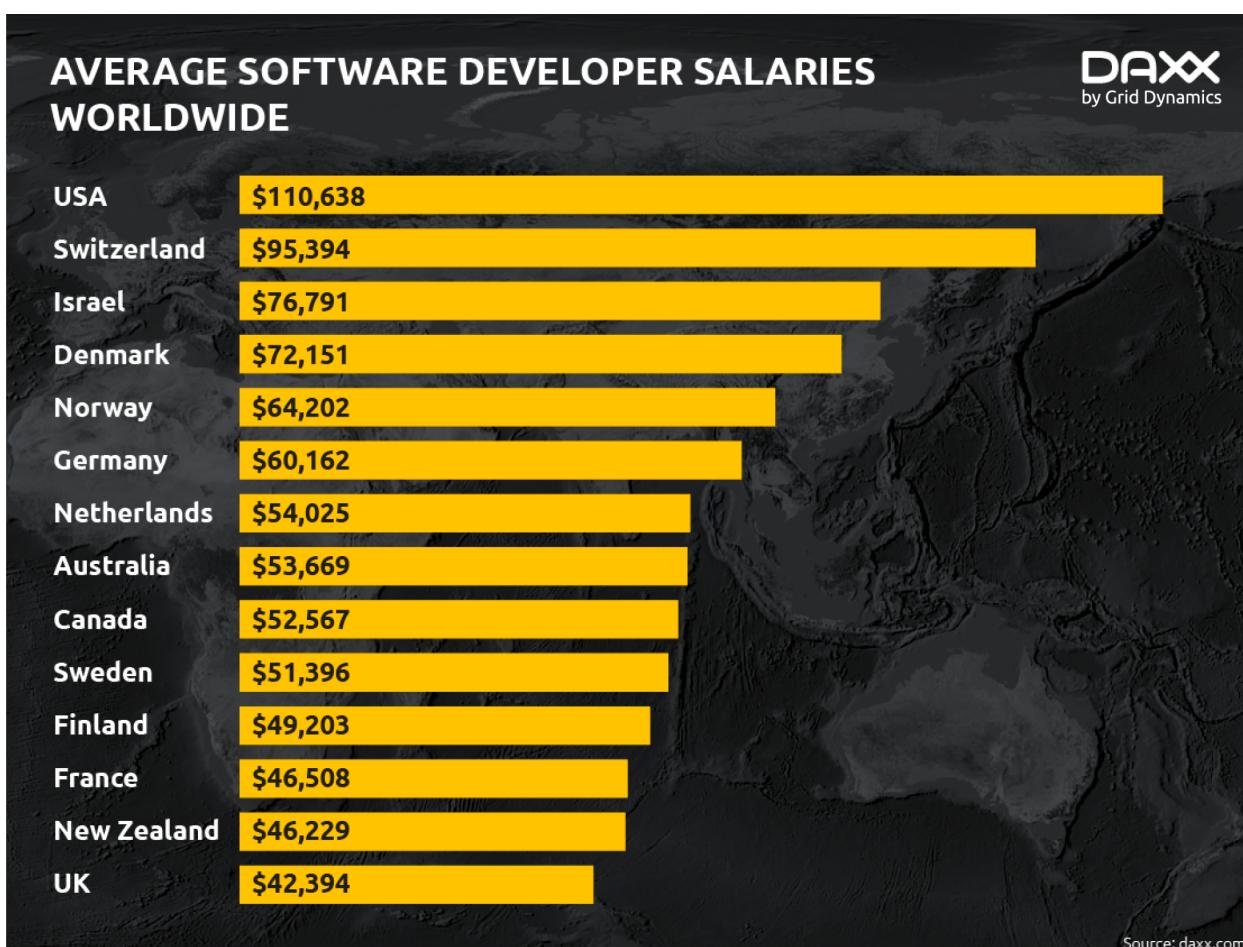
(source: [SalaryExplorer](#))

On a yearly basis this could amount to a range between **\$14000** and **\$22000**. Of course, this doesn't just happen and you have to work hard for it.



The current market in Palestine is still in development, so the salaries aren't that huge in comparison to the rest of the world. It's important to not let this discourage you, because *with software development it's possible to work for a country on the other side of the world (and get paid a different salary too)*.

So if we look worldwide, there's much more potential to earn more!



# Career

## WHAT IS A SOFTWARE DEVELOPER

As the name suggests, a software developer (also known as a software engineer) is a person that “develops software”. But what does this exactly mean?



Any time you interact with a software application, a developer has written code that makes it possible for you to do that. Examples include websites, mobile apps, video games, operating systems, desktop apps, banking software, cryptocurrencies and many more.

This means that you are the one that writes those lines of code in order to

create useful tools that make all of our lives easier!

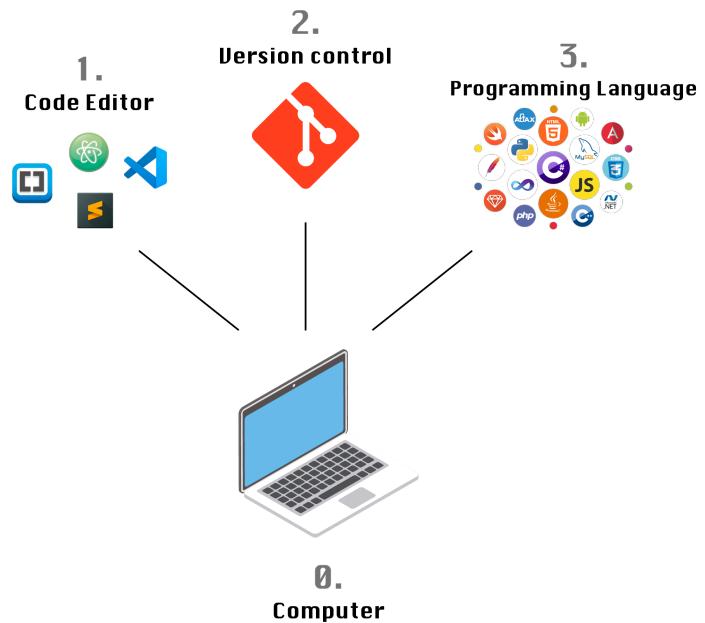
The role of a software developer comes in many shapes and sizes. There are *front-end* developers, *back-end* developers, *full-stack* developers, *Android/iOS* developers, *game* developers and many more roles. All of these roles aren't mutually exclusive and you may well find yourself doing any of these throughout their career.

## REQUIRED TOOLS AND SKILLS

Like any craftsman, also the software developer requires certain tools and skills in order to be able to do their job well. Regarding tools, it boils down to the following 4:

## TOOLS

1. A computer (hardware)
2. Code Editor (software)
3. Version Control (software)
4. Programming Language (software)

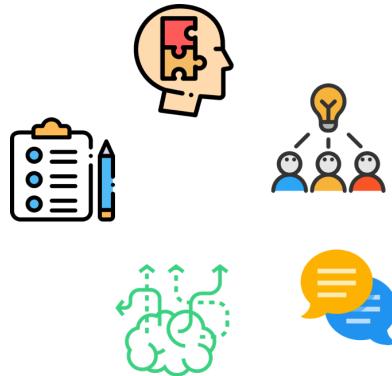


Armed with these 4 tools you have the potential to create any type of application you can imagine!

Of course, tools themselves are not enough. You should also possess the right skills in order to best make use of them.

## SKILLS

1. Algorithmic thinking
2. Self-learning
3. Teamwork
4. Communication
5. Being structured



These skills are always a work in progress. When developing your career you will continuously work on these in order to improve your abilities!

## **DAILY RESPONSIBILITIES**

Now that we are familiar with what a software developer is and what tools/skills they use, let's take a look at their actual activities. A software developer's daily responsibilities, also known as workflow, are defined by the

company they work for. However, there are a couple of universal activities that they all have in common.

## *The Developer's Todo-List*



### **TASK #1: Synchronize**

The first thing to do when starting any working day is to synchronize: with both your team as well as the code base you're working on. This is important, because the work tends to become complex very quickly. By synchronizing with your team members as well as your code you'll know what to do and what not to do when you start working again.

### **TASK #2: Fix bugs**

Bugs, also known as error, flaw or fault in a computer program, are an important part of your daily work. Whenever you're building a new feature, or synchronizing your colleagues' work, it might introduce unknown errors in your codebase. You will most likely spend each working day solving at least a few of these.



### **TASK #3: Build features**

This is what software development is all about: building new things! A feature is an aspect of a product a user can benefit from. As a developer you are building new features continuously in order to create more value for the customer.

### **TASK #4: Demonstrate progress to stakeholders**

Every week or so you'll get together with everyone that has an interest in the product you're building: the stakeholders. These could be a project manager or tech lead, but more



often than not the paying customer. In these meetings, also known as client demos, you (and the rest of the development team) will showcase the progress you've been making and gather feedback.



### **TASK #5:** Research and learn

Maybe the most frequent activity in your work day is to do research, or in other words: using Google to figure out answers to all of your questions! Writing code is complex and more often than not you'll need some support from technical documentation, blog posts of experienced developers or online forums like StackOverflow.

## **FRONTEND VS. BACKEND VS. FULL-STACK**

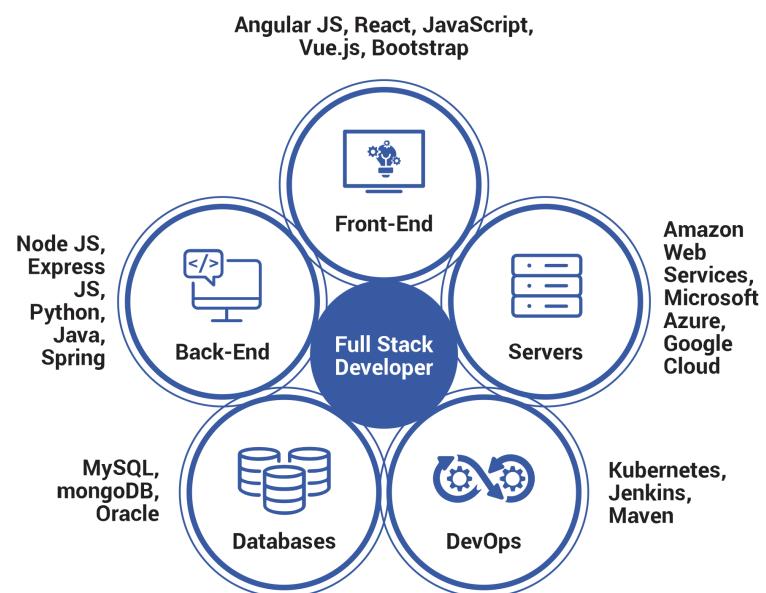
One of the first questions you'll get when you enter the field is: *what kind of developer are you, frontend or backend?* Before we answer the question for which one you should go first, let's define the terms.

	<b>Frontend</b>	<b>Backend</b>
<i>Definition</i>	Front-end web development, also known as <i>client-side development</i> , is the practice of <u>producing HTML, CSS and JavaScript</u> to build user interfaces for web sites and applications.	Backend Development, also known as <i>server-side development</i> , is the practice of creating web servers that interact with APIs and databases.
<i>Job description</i>	<ul style="list-style-type: none"><li>→ Build user friendly, accessible interfaces</li><li>→ Improve Search Engine Optimization (SEO)</li><li>→ Work closely with</li></ul>	<ul style="list-style-type: none"><li>→ Design data models</li><li>→ Configure databases</li><li>→ Develop and integrate backend services</li></ul>

	designers	
Popular technologies	HTML, CSS, JavaScript (React, Angular, Vue)	Java (Spring Boot), PHP (Laravel), Node.js (Express), Python (Django)
Main challenges	<ul style="list-style-type: none"> <li>→ Make UIs cross-browser compatible</li> <li>→ Keep up with rapidly evolving frontend ecosystem</li> <li>→ Requires some understanding of design and user experience</li> </ul>	<ul style="list-style-type: none"> <li>→ Translating business rules into backend algorithms</li> <li>→ Build scalable web servers</li> <li>→ Understanding and working with cloud providers</li> </ul>

But there's a third term you might have heard about: the full-stack. This magical word encompasses *both frontend and backend*. If you look at job descriptions it seems that everyone wants one of these, especially at the entry level.

The honest truth is that it's *very hard* to be a (good) full stack developer in practice. You definitely should understand the basics of both frontend and backend, but to always keep track of the latest is very difficult to do.



*Our recommendation is to specialize in one direction first and then try out the other side to make your skillset more complete!*

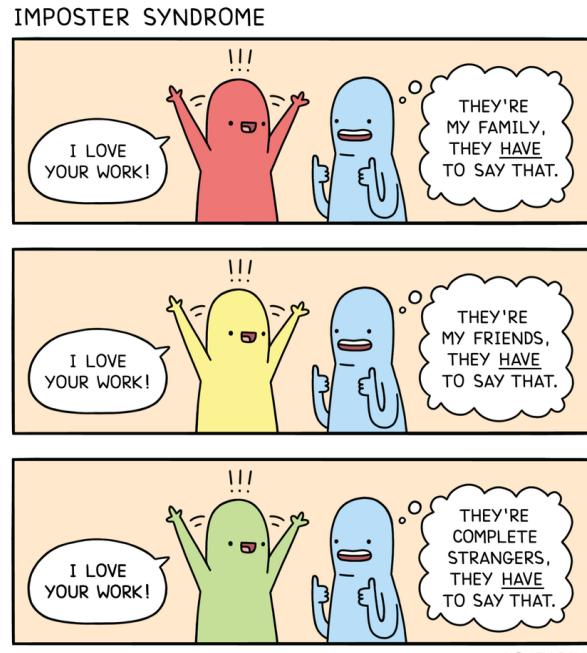
# Challenges and Motivation

Building a career in software development is not easy. There are many factors inherent in the field that make it challenging, no matter your experience level.

## **CHALLENGE #1:** Defeating imposter Syndrome

Most new engineers start off knowing they're incompetent. There's a lot to learn and everyone else seems far ahead. You might worry that you don't belong or that landing your job was luck!

This is called the imposter syndrome: the belief that you're not good enough, despite your demonstrated abilities.



- The honest truth is that this never really leaves you. As our field is always changing you will always feel incomplete. It's best to accept this and just continue on step by step.

## **CHALLENGE #2:** A near infinite amount of new things to learn

This is a major challenge that is especially unsettling for junior software developers. There's so much out there to learn, how will you ever catch up?

If you don't watch out, you might also fall into tutorial hell: the phenomenon that you can't write your own code without looking at tutorials to help you out.

- The way to get out of this is to focus on principles and core concepts. In your journey you will find that technologies will change, but principles and concepts will help you learn so much faster without having to “learn it all”!

### **CHALLENGE #3:** Staying Healthy

Building software is hard. It's mentally exhausting and stressful. On top of that the job requires you to be sedentary, which isn't good for your body either. And lastly, most likely you are haunted by the two previous challenges; this means you'll be coding and trying to improve yourself every waking second of every day.

- While it's good to focus on improving yourself, you are much more than just your job. Instead, try to practice hobbies that have nothing to do with software development. This will bring more joy and variety to your life!

### **COMMON MYTHS**

As with any complex profession, there are many myths surrounding the field of software development. Everybody knows of the stereotype of the anti-social, nerdy guy who can be found only behind his computer all day.

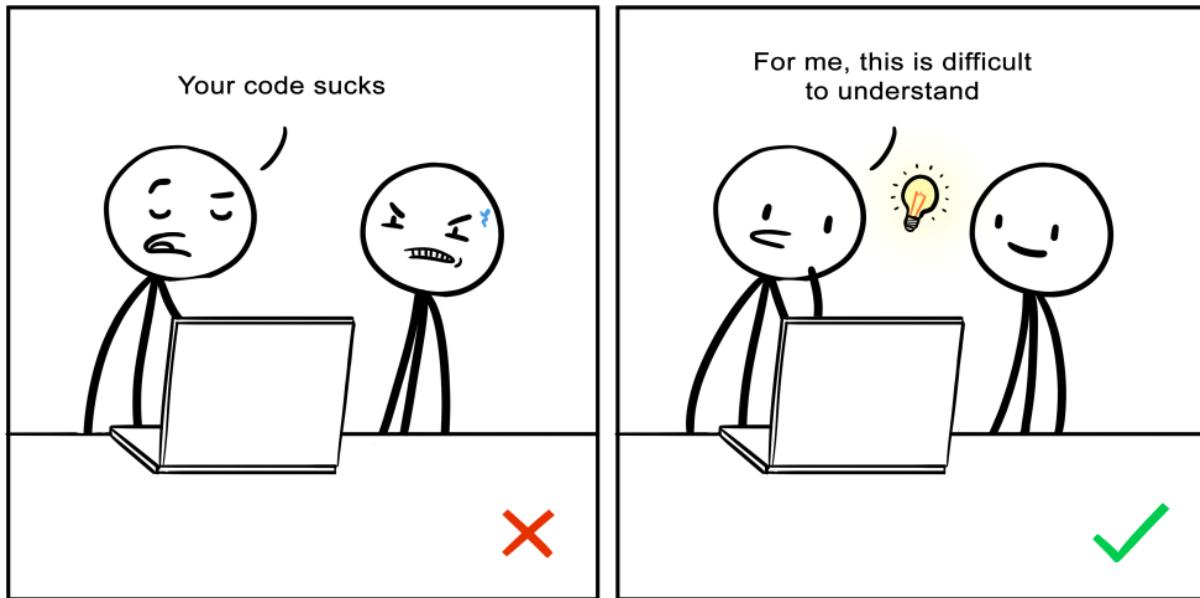


But is this really true? The following are 5 common misconceptions regarding software development that are actually not true!

#### **MYTH #1:** It's only about knowing more technology

Clearly, software developers are hired to write code and have mastery over many technical subjects. However, in the context of business and (especially) teamwork it's just as important to communicate your ideas and understand the bottom line of business in order to do your job well.

This is why it's crucial to develop your "soft" skills (i.e. leadership, time management and communication). You will work with roles other than software developers (*i.e. project managers, marketeers and designers*) on a daily basis, which means that you should be flexible enough to adapt to non-technical situations.



#### **MYTH #2:** Full-stack developers are an expert at everything

This view comes from business people who have little understanding of the complexity software developers are facing. Although a full-stack developer has broad knowledge of both frontend/backend, they still are *specialists* to a degree. This is because there are many different technologies and tech stacks available that it's near impossible to master them all.

As you start building your career, it's much more useful to focus on one direction first so that you develop actual expertise.

This also makes it easier for you to keep up with the fast-moving industry changes that you are required to adapt to!



#### **MYTH #4:** Frontend is easier than backend

This view used to be true 20 years ago, when the frontend was just simple HTML, CSS and JavaScript. Nowadays the frontend, with its fast-evolving JavaScript ecosystem (as well as the newly arrived technology WebAssembly) has evolved to be just as complex as the backend.

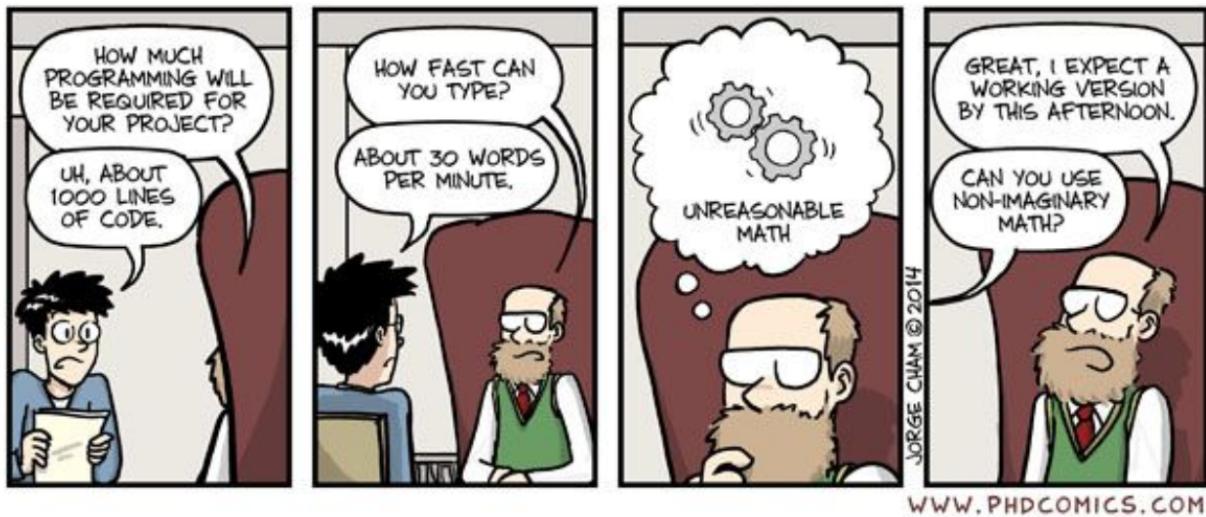
In addition, companies are not looking for either frontend or backend developers but just software developers in general. Are you able to explain what data structures & algorithms are? How about Object-Oriented Programming? Focusing on these topics is a much better use of your time!

#### **MYTH #5:** The more code you write the better

Productivity usually gets measured by amount of output. So it makes sense that the amount of code you produce should also equate to how good of a job you're doing. This is not true in the case of software development.

When building applications, we're always trying to make things as simple as possible. If you can achieve a computational result with fewer lines of code that is always preferred. Your task as a developer is to come up with smart

ways to reduce complexity as much as possible. This is why we have *design patterns, frameworks and best practices*.



JORGE CHAM © 2014

WWW.PHDCOMICS.COM

## STAYING MOTIVATED

In your journey to become a better developer you will face many obstacles. Some of them might even make you feel demotivated and wondering why you keep doing this. Here are 3 tips that can help you stay motivated as you build your career.

### **TIP #1: Keep your eye on the prize**

Whenever you're facing hard times, remind yourself of why you're doing this: is it the money, the skill set, the potential job opportunities? All of these are great motivators and it's important to never forget those.



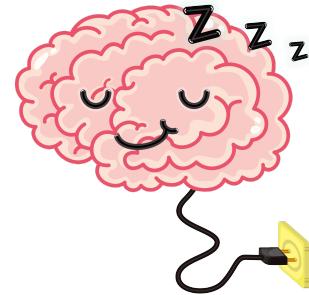
### **TIP #2: Do a small amount every day**

Consistency is the best way to learn something new. If you write code every day, even if it's just a few lines, you keep the topic fresh in your mind. Just set yourself a small achievable goal. It might be half an hour, it might be 15

minutes. You may even find when you start you get on a roll and want to go longer!

**TIP #3: Give yourself time to grow and adapt**

Learning takes time. Your brain is actually *changing* when you're learning something new, making you an improved person as a result. When you fail to rest, or keep forcing yourself to work, you're actually *slowing down the learning process!* This is because stress and learning don't go well together.



Programming isn't easy and learning to program will likely require you to think in ways you've never thought before. If you're feeling stuck or frustrated, understand that it's all part of the learning process. Embrace it! Through getting unstuck, you'll not only internalize the skills you're learning, but you may find personal growth as well.

# Final Words

With this introduction you hopefully have a better understanding of the industry, the rewards and challenges, and how to best approach it. In the next 12 weeks you'll come across many different topics that will challenge how you think about software development, as well as how you think about yourself.

Go in with an open mind and question everything you encounter. This will not only help you become a better developer, but also a better professional.

We wish you the best and hope to see you on the other side!

*The TAP Team*