Finding your finish line

You know that iconic scene from the 1995 film “Hackers” in which a man dubbed The Plague coolly schools a room full of suits over weak password protocols? “Our recent unknown intruder penetrated using the superuser account, giving him access to our whole system.” He continues, “Someone didn’t bother reading my carefully prepared memo on commonly-used passwords.” He commanded the room’s attention and made it look cool to know your stuff in tech.

Today, software development is a career that isn’t slowing down anytime soon; the projected change in employment from 2018 to 2028 is a substantial 21 percent for developers (against a 5 percent average growth rate for all other occupations). If you’re considering carving out a path for yourself in programming, here are five steps straight from in-house Veracode developers that will help you further your career and become a successful programmer (nickname like The Plague optional).
1. **Hobbyists and Students** – Hobbyists and students tinker with languages and penetration testing for fun and practice, leveraging those skills during school or in a novice role at a tech company.

2. **Junior Developer** – Junior Developers are sometimes also given the title Engineer I or Associate Software Engineer, best suited for roles that require the fundamentals of programming.

3. **Intermediate Developer** – Intermediate Developers, also sometimes called Software Engineers, typically have a few years of good experience in programming.

4. **Senior Developers** – Also sometimes called Engineer III or Senior Software Engineers, Senior Developers have been in the game for a while and have worked their way up in their organization.

5. **Principal Engineer** – Principal Engineers, also sometimes called Senior Principal Software Engineers, are the cream of the crop with several years of experience under their belts, and will often oversee projects for the team.

6. **Developer Manager** – When developers decide to take a managerial track and branch off from engineering, they become a Developer Manager and supervise software engineers of all levels.

While you may find different labels and career trajectories for each of these as you join new organizations, the standard remains: building real-world experience with coding is key to moving up as a software developer.

Know the stages of a software development career

Career paths are structured differently at every company, in every department. For the most part, you can expect the stages of a career in software development to look as follows:
Never stop learning, practicing, and growing

“I’m not a great programmer; I’m just a good programmer with great habits.” – Kent Beck

It’s true for every craft. The more you practice and learn, the better you’ll become. We asked a few of our in-house developers and team members what they think is the best plan of attack for becoming a sharp software developer. A common response: be diverse and learn to write code in multiple languages. That sundry of experience will carry through your career and open new doors.

Imagine you’re sitting in a room full of Junior or Intermediate-level developers and your manager asks if anyone can tackle a project in C++, one of the most difficult programming languages for new developers to learn. You raise your pen with poise, because you’ve practiced that. As Kent Beck knows well, great habits can lead to good success.

Ask yourself, “How does that work?”

Have you ever felt completely bewildered watching software developers chat about a coding project or application? You’re not alone. One of the key characteristics of a developer with grit and determination is asking, “How does that work?” at every turn. Try looking behind the curtain of everyday tools (Chromium, for example, is the open source project for Google Chrome) and learning how they work. Even Albert Einstein was on board when he said, “The important thing is not to stop questioning. Curiosity has its own reason for existing.”

Think of it this way: as a software developer, one of your top priorities is building a sturdy application that won’t fail, but priority number one for a malicious actor is to break your build and undo that hard work to access valuable information. If you’re not curious about how an application works inside and out – and how an attacker might crack it – then you’re probably missing one of the greatest thrills of programming; looking at code from all angles so that you’re one step ahead of the bad guys and always expanding your skills.
There’s a perpetual security problem in software. Based on data from our 10th Annual State of Software Security Report (SOSS X) in which we looked at 85,000 application tests (including 1.4 million individual scans), we know that 83 percent of apps have at least one flaw when they’re initially scanned. That’s a big problem, but one that software developers can help fix. Working with security teams, or possibly even becoming a security champion on your team, is like using the Force to bind your DevSecOps galaxy together.

Upping your security game will save your organization a lot of extra work (and money—security flaws are much cheaper to fix up front before they become a problem) with speedier time-to-market and less cleanup down the road. That’s a recipe for success that will help you become a more marketable developer through your career, too. You don’t have to call yourself a Security Jedi. But if the shoe fits and you find that you care deeply about the fortitude of your code, by all means, use the Force.
Harness the power of existing tools to prepare yourself

As a software developer, you're going to have a lot of tools at your disposal throughout your career, some of which can be used to build foundational good security hygiene. Getting familiar with common open source tools and frameworks will lessen your learning curve when you join a new organization. For personal projects, use revision control like Git or hosted options like GitLab and GitHub. Security flaws can be caught early through good code review, so try to read through merge requests on open source projects, particularly if you can find patches for flaws. Dan Murphy, Principal Senior Software Engineer at Veracode, also suggests getting comfortable with source code scanners, debuggers, and tools that hackers use themselves.

Training tools are critical, too. While there are many demo or presentation-style learning solutions out there, nothing works quite like a hands-on coding tool that gives you real real-world experience. For training on secure coding, Veracode Security Labs packs a punch in that respect; the program allows developers to access a real application in a contained environment, where they can then exploit the application and learn how to fix it. Instead of sitting back and observing, you become an active participant in the fight against cyberattacks.

Conclusion

While life doesn't always imitate art from the mid-90s, modern software developers need to be able to keep pace with tight development timelines through continuous learning and practice with the right tools. As you head into your career in programming, seize the chance to become a capable developer with the skills and drive to impress your way into the role of your dreams. As the principal developer of the Linux kernel (Linus Torvalds) once said, “Talk is cheap. Show me the code.” Time to get to work.