FAST TRACK TO CODING MASTERY

5 Powerful Steps to Become an Expert Developer and Build a Thriving Business (Step #2 is Pure Gold!)

DANIEL GAKWAYA

You've done it hundreds of times! Spent countless hours sifting through outdated or overly **complex** tutorials, books and courses, only to end up more confused than when you started. This scenario is all too familiar for many aspiring (and experienced) developers. A survey conducted by Developer Media found that over 60% of developers feel frustrated and overwhelmed by the sheer volume of ineffective learning resources available out there.

As a (aspiring) professional developer, **your time is incredibly valuable**. Wasting it on ineffective tutorials not only hampers your progress but also diminishes your motivation to learn and grow. You need a **clear**, **efficient** path to mastering your technology of interest—one that cuts through the noise and provides you with the practical skills necessary to build robust products and healthy businesses from your skillset.

In this guide, I will walk you through 5 definitive steps to becoming a **leading expert** in your field without the frustration and inefficiency of traditional learning methods. These strategies have **transformed thousands of aspiring developers into industry leaders**. By following these steps, you'll gain a solid foundation in software development, build practical skills through hands-on projects, and **build an audience that flocks to you like butterflies to a flame**. Let's get you started on this journey to streamline your learning process and **fast-track your path to expertise.**

SPECIAL NOTE:

I am a professional C++, Qt and QML developer and I own a few businesses revolving around these skills. My advice may use these areas of my expertise as examples, but **the strategies in this guide apply to any programming language or technology**. Don't be deceived by the simplicity of the strategies. **Many fall into that trap and never get their career or business off ground!**

STEP 1: UNDERSTAND THE BASICS THOROUGHLY.

We're all tempted to do great things, make them perfect and all that in as little time as possible. There are many resources online claiming to make you an expert little time, but deep down we all know that's not how it works. Valuable things things take time and effort to learn and master. But it doesn't have to be forever: **you only need to be strategic about it**. Here's what has worked for thousands of students we've worked with over the last 5 years:

Pick one or two good resources and stick with them up to completion.

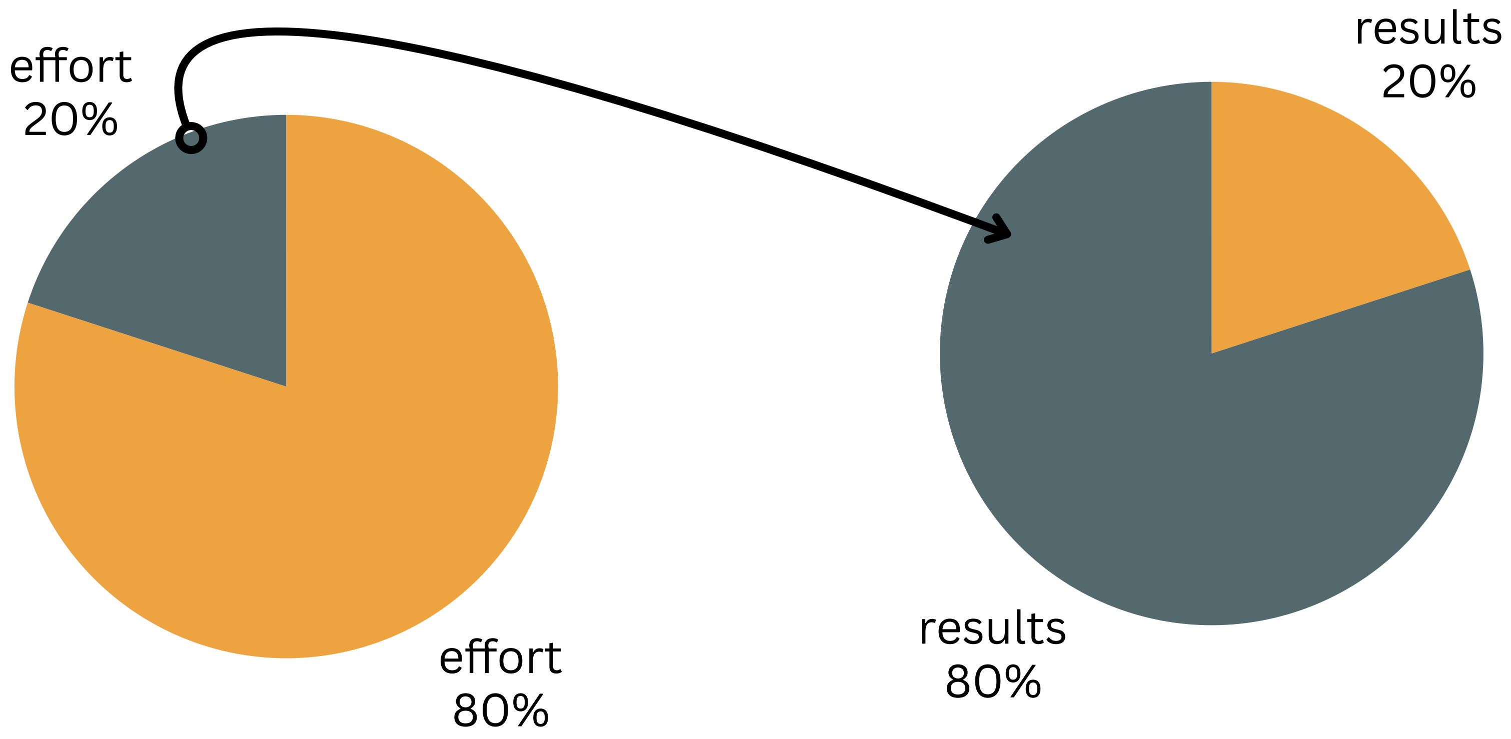
This sounds basic and straightforward, but you would be **shocked to know how many people jump from one resource to another** without completing any. This is a recipe for disaster. You end up with a lot of half-baked knowledge that doesn't help you in any way. In our experience, students who work on this at least **3 hours a day, 5 days a week, complete this step in 1-2 months.** That's a reasonable time to have the basics under your belt. You don’t have to dedicate that exact same amount. These stats are purely for your use as a reference.

Action points:

- Pick one or two resources around your technology
- Stick to them until the end. Don’t jump around
- Give yourself a reasonable time frame to complete them.

STEP 2: BE VERY PICKY. FOCUS ON WHAT MOVES THE NEEDLE.

Assuming you completed step #1, you should have a good general idea about your technology, and you're probably excited about thousands of things you can do with it. But here's the thing: **about 20 percent of what you've learned will be responsible for 80 percent of your results.** That's right! Put another way, about 80 percent of the knowledge you've acquired will be responsible for only 20 percent of your results. Let's visualize this:



Twenty percent of your effort is responsible for eighty percent of the results that you achieve! I didn't come up with this fact. It's a well known principle in business and productivity circles called the Pareto Principle. Seeing this, any reasonable person would want to focus on the 20% that moves the needle. But **that's not what most people do.** They get excited about the possibilities and end up spending 80% of their time on things that don't matter that much. **This is a recipe for mediocrity.**

This is where you have to sit and think hard about what you really want to do with your technology. What are the most important things you want to achieve with it? What are the most important features you need to learn to achieve those goals? What are the most important projects you want to work on? What are the most important problems you want to solve? May be you're looking to be employed and your physical location has lots of job opportunities for car infotainment systems. These are embedded systems at the core and they have a fancy display touch screen. This already gives you a hint that you may need to focus on Qt Quick and QML, and how to integrate them with C++. That's just one example, but you get the point.

The key is being very specific about what you want to achieve up to a point where **everything else becomes almost irrelevant**. I call this the **"power 20%"**. Once you have your power 20%, the answer to anything else that tries to grab your attention should be a resounding **"NO".**

Action points:

- Be very specific about what you want to achieve with your technology.
- Work real hard on identifying your power 20%
- You don't have to get it right the first time. You can always adjust as you go
- Once you have your power 20% in sight, laser focus on it
- Say **"NO"** to anything else that doesn't align with it.

STEP 3: FOCUS ON HANDS-ON PROJECTS. JUST DO IT!

With **step #2** out of the way, you should have identified your power 20%. Remember those days in high school when your crush was with a bunch of other people and for some reason all other people were blurred out and you could only see him/her? **That's how you should feel about your power 20%.** Let's recap what we've done so far:

- You've completed a good resource or two and have a good general idea about your technology of interest.
- You've identified your power 20% and are laser focused on it.

Now it's time to **get started** working on your power 20% by building hands-on projects. Don't be shy or sell yourself short. **What you'll achieve will greatly depend on how high you aim.** Pick a real projects that aligns with your power 20% and get started on it. **Let me give you a personal example:**

Back in 2015, it became clear that I would be working on a project that would record the screen and process the video in real time. The rest is confidential. I spent a few days pondering on the task and it occurred to me that I had been using a piece of software named **OBS Studio** for a while, and it already did a lot of what I had to achieve. But given my skillset at the time, I was **terrified** by how professional the project looked and felt like it was a mountain I would never climb up to the top.

One morning, I decided to **try** and build OBS Studio from source. The project had **a lot going on** for the **junior developer** I was at the time, and I was faced with hundreds of compiler errors and warnings. I attempted to fix them one by one and after a few days, **I had a working version of OBS Studio that I had built from source, running on my Windows box at the time. I suddenly felt like I had a magic wand that could turn OBS Studio into anything I wanted to.** I spent the next few weeks identifying parts that were responsible for screen capturing and some FFMPEG processing that it did, and extracted those into the project I was working on. **I was able to achieve in a few weeks what would have taken me months to do from scratch.** Not to mention what I picked up by attempting to pick up a project done by the professionals, and well above my skill level at the time.

The point I am trying to make is that **once you have your power 20% in sight, you shouldn't dance around in directions that deviate from it. Be laser focused and aim high.** You'll be surprised how much you can achieve in a short amount of time. You don't have to build a project from scratch, you can also pick a project interests you on GitHub and try to add features to it or fix bugs. The key is to get your hands dirty and learn by doing. **One last piece of advice here:** don't work on multiple projects at the same time. May be I am not much of a multi-tasker myself, but I have seen better results when I focus on one project within a given time frame, and only moving to others when I have it completed. You can experiment and see what works best for you.

Action points:

- Pick a project that aligns with your power 20%.
- Set reasonable goals and deadlines.
- Learn from the pros by working on their projects when possible.

STEP 4: BUILD YOUR BRAND. MAKE THE RIGHT FRIENDS.

I know what you're thinking: "What do friends have to do with me becoming an export in my field?" Well, let me tell you this: The connections you make with other developers can make or break your career. I have seen it happen over and over. You can be the best developer in the world, but if no one knows about you, you'll be stuck in the same place for a long time. On the other hand, you can be an average developer, but if you know the right people, you'll be amazed at how far you can go.

Another way I like to look at this is to think of myself as if I were selling something. If you're looking to be employed, you're looking at selling your skills, and if you're aiming to build a business, you're looking to sell your products or services. **In both cases, you need to build a brand that people can trust.** This is something you have to think about from the get go. If you're not doing it already, **start now!** Here are a few things you can do to build your brand:

- Start a blog or a YouTube channel where you share your projects and what you've learned.
- Contribute to open source projects.
- Attend meetups and conferences around your technology.
- Connect with other developers on LinkedIn and Twitter.
- Share your projects on GitHub and other platforms.

You don't have to engage others just for the sake of it though. You should provide value. **People love value, and they flock to it like moth to a flame.** Trust me, if you're providing value, the right people will find you and over time, you'll build a network of like-minded developers, and business people who can help you grow and achieve your goals. **This is not optional. It's a must!**

Let me end this with my favorite quote from James Clear: ***"You do not rise to the level of your goals. You fall to the level of your systems. Your goal is your desired outcome. Your system is the collection of daily habits that will get you there."*** You should incorporate building your brand into your daily habits. It's not something you do once in a while, it's something you do every day. Over time, **all this will turn into a powerful engine that propels you towards your goals.**

Action points:

- As you build projects that align with your power 20%, build a strong network of people that will help propel you through the roof.
- You don't have to do one big thing in a day, you don’t have to be extreme. Small consistent effort and actions over time will get you there.

STEP 5: SCALE. SPREAD YOUR WINGS WIDE AND FLY.

I'll start this with one of my favorite quotes: ***"When you stop growing you start dying"*** - William S. Burroughs. I'm assuming you carried out step #1 through #4 over a period of time, may be a year or two. You've completed a good resource or two about your technology, identified your power 20%, built hands-on projects, and built a strong network of people that will help you grow. **You're now at a point where you can see the fruits of your labor.** You're getting job offers, you're getting clients, you're getting recognition. **You're on top of the world!**

But the technology world is brutal. If you stay in one place and stop going forward, what you built in 5 or 10 years could crumble in one month. You need to future proof yourself. step #1 through #4 gave you a recipe to build a strong foundation for your profession or business and all you have to do is to constantly ask yourself how to make it better. Here are a few things you can do to scale:

- Start a new business around your skills.
- Teach others what you've learned.
- Write a book or create a course.
- Speak at conferences and meetups.
- Collaborate with other developers on bigger projects.

Action points:

- Keep asking yourself how to make what you've built (product) better
- Have a strategy that helps you keep learning and growing.

Conclusion

If you’ve known me for a while, you might have expected a lot of knowledge about modern C++, Qt and QML in this guide, but knowledge is pretty cheap these days. The value is in how to capitalize on that knowledge to create value for yourself and that's what tried to focus on in the 5 steps I shared with you.

Becoming a leading expert in your field is not an overnight process. It takes time, effort, and dedication. By following the steps outlined in this guide, you’ll be well on your way to mastering your technology and building a successful career as a professional developer. Remember, the key to success is not just in acquiring knowledge, but in applying that knowledge to real-world projects and building a strong network of like-minded people around yourself.

If you consciously work on your career with these steps in mind, I am 100% confident that you'll quickly rise to the top of your field and become a well respected expert in your technology community.

Here is to your success!

Daniel.