**Daniel Di Lillo, O’Neil and Associates**

My name is Daniel Di Lillo and I am a Software Developer at O’Neil and Associates. What we do at O’Neil is really help our customers make the things they make better. And we do that by optimizing their product support information, and then optimizing the technology they use to deliver that documentation to their customers. A software developer, in general, we write code. We also interact with a lot of other team members - software analysts, testers - and we build software. One of the great parts about my job is being able to design a user interface and then write the code to bring it to life.

We collaborate with our customers to get a sense of how they think the finished product should look and feel. And then once we’ve got a really good sense of that, we’ll turn around and start to flesh out the back end and find out: do you need a database? Do you need some kind of integration with some other systems? Each customer’s story’s a little bit different, but we always build the software to meet with their expectations. Our end product is typically a website or a system that allows our customers to produce PDF documentation - the actual manual - for their end users. We will write some code, do some testing, incorporate the content that they have - or potentially set them up so they can start producing content within our system. And then ultimately that’s delivered to the customer by way of the web site or desktop application we built to their specifications.

Some of them are commercial players. We also do a lot with the department of defense, Really, it’s a wide variety of customers who are all interested in making their products better in much the same way.

B-roll audio featured :

I think that’s something where we’d have to work directly with the customer and talk about the impacts to their data…

I grew up in a small town and my dad was a photographer. My mom worked in a social services organization. We weren’t necessarily computer people, but like so many kids in my generation, you got exposed to video games and I thought: hey, that’s way cool. I’d like to do something in that field. Academically, I was always much more inclined towards art than science. I really enjoyed the creative process. But I also found other classes that really melded math and science and the creative side of me together. So drafting was a good example of a class where I got to use a program to draw up building plans. And then the software would show me what it was gonna look like in 3D, and I thought that was so cool. And so in many ways that parallels what I do now. It’s kind of, you write something very flat. But then you bring it to life and you see it animated on the screen and that’s really cool to see.

I have a Bachelor’s degree in Management Information Systems. One of the things that’s really cool about my industry is you don’t necessarily need a four-year engineering degree to become a great software developer. In a lot of ways, it’s about having just the aptitude, the passion, the drive to become really skilled at writing code and analyzing problems. Software developers have to spend a lot of time keeping our skills fresh. Some of that happens during our nine-to-five, just kind of interacting with other folks, prototyping things. But a lot of it happens at night or on the weekends. You pick outside projects. You may engage in some online training, and so it’s a lot of learn by doing and, luckily we can rely on the internet to do that. There’s a wealth of information out there about new techniques, new tools, and that becomes a great source of training.

On a day-to-day basis it can really vary, but typically, I may attend a few design meetings where we start to look at individual details of the software, kind of put together the hard requirements that ultimately are gonna dictate what we’re gonna build. And then, usually in the latter part of the day, I’ll actually sit down and write some true code, which later on will be incorporated into the product.

To be successful as a software developer, you really need a lot of hard technical skill. When you build the software that’s integral to people’s workdays, when something’s not working right, it’s important to get in there and solve that problem as quickly as possible, because time is money and you want to get that person back being productive. But, it’s also really important to have soft skills. It’s really a people job. You’re talking to customers, you’re talking to colleagues, administrators, you’re talking to sales. So being able to have those conversations, being able to articulate opinions, helping folks who are not technical understand technical problems is really vital.

I think if you’re someone who’s interested in computers, if you’re someone who is inherently curious about technology, this could be a really interesting path for you. It’s probably one of the best combinations of creative and technical that I’ve ever come across. It’s really fun to be able to sketch up what a user interface might look like. Or imagine what a product should be, and then try and identify a technical solution that can bring that to life. It feels great when everything’s working right, when our users are telling us: Hey, this software made my life a little bit easier today. That’s a really great feeling.