



LEARNING TO FAIL FOR FUTURE SUCCESS WITH EXPERT ENGINEER PETE WARDEN

AN INTERVIEW WITH PETE WARDEN

Pete Warden is an engineer, co-founder and CTO of Jetpac Inc, a startup launched in 2012 (and since bought by Google) that analyzes Instagram photos to build what it calls City Guides for travelers. He is an O'Reilly author, blogger, builder of OpenHeatMap and the Data Science Toolkit, and other open source projects.

Pete has been a consultant to the New York Times, was senior engineer at Apple, and is currently the technical lead of the TensorFlow Micro team at Google.

[Arduino Education:](#) Hi Pete, it's really great to meet you. Can we start today by asking you what gets you inspired?

[Pete Warden:](#) What gets me inspired are the projects that I see people building. I always have my mind blown by what people out there can actually conceive with their imagination, and the problems that they're able to creatively solve.

[AE:](#) That's really cool. What about something you wish you'd known at the beginning of your career?

[PW:](#) I wish I'd paid more attention to some of the other lectures in my computer science course, rather than just focusing on the ones I thought I might need for the first career that I had in mind. I'd tell myself to be open to lots of different possibilities.

I started off in computers, because I was really excited about getting into games, and building games and programming games. And that's still great. But after the first few years, I realized that there were so many other things that I could actually be doing with computers that were really interesting, creative, and fun.

[AE:](#) Thanks for letting us in on that. What about a common myth within your industry that you'd like to set straight?

[PW:](#) One thing I am really trying to work hard on is this idea that machine learning is something that you need a PhD in to understand and to use. I think that's just because we



have been really bad at explaining what it is - and really bad at building tools that everybody can use.

My hope is that by building simple examples that people can start playing with, they can see that machine learning is just like any other programming tool. It's something you can play about and experiment with, you don't have to understand everything that's happening under the hood, or be able to design the new cutting edge models. I'm hoping to demystify a lot of the confusion and jargon that surrounds machine learning.

[AE](#): So Pete, we were looking at your blog, and we found that you have on there a quote by Samuel Beckett, that reads “*Ever tried. Ever failed. No matter. Try Again. Fail again. Fail better.*” Could you tell us more about what this quote means to you and what you think about failure?

[PW](#): It's a little tricky to talk about sometimes. Because I'm a Google engineer, it's pretty easy for me to say, “Oh, you know, just don't worry about failure, just keep trying stuff.” It would have been a lot harder when I was working stacking shelves at Quick Save or living with a much smaller margin for error!

But the big thing that I've found is that oftentimes the projects that actually crashed and burned are the ones where I actually learned the most. Actually getting to fail is a lot better than having something that sits in a closet for months or years and never actually gets tested.

There's another quote that sticks with me, which is a little bit cheesy maybe, but I think it was Edison who said something like he wasn't figuring out one way to build a new lightbulb, he was figuring out 1,000 ways **not** to build a lightbulb. And then that actually leads to you coming up with a way that works.

At the heart of it, I want to tell students that we are all figuring it out as we go along, no matter how old or experienced we are. And they shouldn't feel bad if they can't get something working, or if they're struggling with something. Because there's nobody out there who just does it completely naturally the first time.

[AE](#): And how do you think we can fail better?

[PW](#): One thing that I really like doing is just trying to keep notes for myself as I'm going along. My blog is a way of forcing myself to write notes about things I've done, and to leave a trail of breadcrumbs both for other people, but also for myself.

If you're actually able to just explore and try a lot of stuff, but do it in a way where you're keeping track of what you've tried and leaving a record, you'll be learning as you're going along. That I find to be really, really helpful.

[AE](#): Yes, keeping notes is something I think we all try to do here! So you mentioned that you learned the most from some projects that crashed and burned, are you able to share any of those with us?





PW: I once worked on a game called Formula 198. It was the successor to Formula 197, which was a really good, really successful game. We only had something like nine months to get this game out, and we were a new team. We decided to write it from scratch. I think it's still out on Wikipedia as one of the one of the worst reviewed games!

I was a very junior programmer, but I ended up just being handed all of this stuff to work on. And I just had to figure it out as we went along. And I got to make a lot of mistakes!

AE: So to finish on machine learning, are you able to tell us about your current role at Google?

PW: When I joined Google, it was when [TensorFlow](#) (an open-source platform for machine learning) was just being started internally. I was the one working on getting lots of data running on these small handheld devices, which was really new at the time. I became the person who was responsible for getting TensorFlow running first on Android, then on iOS, and eventually on the Raspberry Pi.

It was pretty amazing when we launched. I think the timing was right, because a lot of people really started diving into it. We had a lot of people who wanted to do machine learning models on mobile phones and Raspberry Pi type devices.

Then I managed to get some support from our management to focus on what we call a “moonshot”, a research moonshot within Google, based on how small we could make a machine learning framework. I got a year and the resources to go away and experiment with that - and that experimentation became TensorFlow light micro.

INSPIRED? HERE'S WHAT YOU NEED TO GET STARTED WITH ARDUINO EDUCATION

The [Arduino Student Kit](#) can be used for both home learning and classroom teaching. It will help you get started quickly and easily with robotics, electronics, and coding.

[Find your country's distributor](#), or [buy the kit online](#).

