

Teaching

Adrian Kosmaczewski

2021-03-19T18:23:12+01:00

In November 2015 I attended the DO iOS conference in Amsterdam. After the event was over, I had dinner in a pub with my friend Daniel Steinberg, whom I had not met in more than three years, and who had just held a terrific Swift workshop in the event.

I remember during that evening, Daniel mentioned one of the most prominent laws of learning ever muttered, one that probably was first pronounced by Plato, Archimedes or some other wise Greek person 3 thousand years ago:

Teaching is the best way to learn.

- Speaking
- Teaching
- Publishing
- Coaching

Many programmers are usually very shy of talking in public, let alone capable of explaining technical concepts in intelligible ways to non-technical audiences.

Achieving both is a worthy goal, and one that can paradoxically teach you a lot.

Speaking

The most important soft skill you should master is presenting in front of an audience. It is surprising, but I still come across many MBA and university curricula without any course about presentation skills as a core requirement.

The thing is, you are going to be showing your work to people all the time, whether you want it or not; hence, you should start working out your presentation skills as soon as you can.

How? Well, very simple; go to the local user group of your technology of choice, and give a ten minute talk about any subject you like. Just that. Did I say ten? Heck no, just five minutes is enough. And make sure that there are no more than ten people in that session. Small audience, small subject, short time. That is all.

Start small. Smile and breathe.

Sounds easy? Well, there is a catch: you should write down your speech. In Markdown or in AsciiDoc (seriously, avoid Word and company). This is the most important part; to prepare your talk. It is fundamental.

It does not matter if there are ten or a thousand people in the room, you have to know your subject inside out, and for that, the best way is to write down your speech. It will help you remember it, it will reduce your anxiety (because, yes, you will be anxious) and it will increase your confidence.

By the way, written speeches can also make for terrific blog posts. Just saying.

Where to start? Very simply; write down what you want to say. Make sure that you have an introduction, a body, and a conclusion: remember the simple formula for any good text:

1. Say what you are going to talk about.
2. Talk.
3. Summarise and repeat what you just talked about.

Unlike programming, where we strive to DRY, repetition is very important in speeches. Getting your point across means repeating yourself.

At the end, thank everyone, and ask politely for questions. Smile and breathe. It is done.

Once you have your speech on paper, rehearse. Three or four times, at least. With a timer. You can use your phone for that, or a kitchen timer, it does not matter. Preferably in front of a non-technical audience, like your partner or some best friend. Ask them how you did and remember to smile and breathe.

Take time to breathe. It is super important.

Then, after you have written your speech, and only then, you should create your slides. Make them eye-catching, not a lot of text, and with big fonts. Use images. Lots of them. Images are better than text.

And then, give your presentation. Remember, smile and breathe.

Thank everyone and ask politely for questions. Smile and breathe one last time. It is done.

If you liked presenting to ten people a five minute subject, you should gradually expand both variables. Pitch your talk to technical conferences – they crave new content, and many find it very difficult to find good speakers.

After ten or twelve times of doing that, one day you will be talking for a whole hour in front of a thousand people. It happened to me and it is frankly exhilarating.

Teaching

The next logical step will be to use the material you wrote down (in Markdown or AsciiDoc, remember), and expand on it. Think about a small group of twelve to twenty people, to which you could teach that during a whole day.

It is a business! If you charge 200 USD to each person for such a workshop (a relatively reasonable price), and you have twenty people signing up, you can make four grand a day. Renting a classroom might cost you from 300 to 1000 USD per day, depending on location and whether you offer food or not. In any case, repeat that operation one week a month and you have a very decent salary. Use the rest of the month to create new trainings, rinse and repeat.

Publishing

This is the final step. Take those classroom notes you wrote down, and publish a book. Yes, do it.

Because you have written your notes in Markdown or AsciiDoc (you have, right?), you can use pandoc to export them to a beautiful PDF, EPUB, HTML and a few more formats in one shot.

Next thing you know you can start pitching a full manuscript to O'Reilly or Pragmatic Programmers, and maybe you will get yourself published! Publishers love when you pitch an almost-ready book, because it makes them more confident of your capacity to deliver stuff in time. Do not worry about being copied; the ones I mentioned above are trustworthy.

And if you do not get a publishing contract with one of those publishers, then you can go to Leanpub.com and (again, thanks to Markdown) compile your notes and publish your book anyway. Just do it.

Coaching

Finally, at some point you are going to find a younger developer hungry for knowledge and anecdotes. Coach them. Use your knowledge to shape a better generation of developers in the future. Show them how you used to code before, and let them show you new tricks.

Coaching a younger developer should be mandatory for all of us; it makes coachees wiser, and it keeps coaches younger at heart.