



Open Source Skills Report 2026

Case Study: Red Badger



4.2 Fireside Chat: Pedro Martin Valera, Red Badger

Senior Software Engineer, Red Badger
Instructor, General Assembly



About Pedro

A senior software engineer with two jobs, I work at Red Badger as a consultant and I teach software development to non-technical people at General Assembly, which is a boot camp academy. I teach immersive courses that are designed to take non-technical people and enable them to become coders. Based in London, I'm originally from Venezuela and back home I was an environmental science teacher for almost 15 years. Thirteen years ago, due to the political situation in Venezuela, I left the country for England. I spoke no English and my degree did not translate so I was unable to teach and instead worked in construction as a labourer, where I learnt English. A couple years later my wife had the idea to create a website, something that would now take me two days, but at the time took me three months! I did it and thought it was great so I went to the General Assembly, trained as a software engineer and they were crazy enough to hire me as a teaching assistant immediately after. I've been teaching there for almost 11 years.

Teaching versus doing, why do you need both?

The truth is, I think I would teach for free. That's just me... engaged and high spirited. I love to be a part of someone else's learning journey, seeing where they're at several years later and feeling grateful to have been a part of that journey. Since the moment I started teaching I just could not stop.

What role does open source play for you?

One of the things that attracted me to software development is that things evolve so rapidly and the community knows that the only way to keep up is to be proactive in your learning and training. Most of the people have the spirit and ethos to share knowledge. Engineers love to do things and then talk about it. Having something you built be consumed or utilised by other people makes you feel that you are a very small part of something that is great. The only place I have really seen this flourish and evolve is within the open source community.

One of the great things about the company I work for, Red Badger, is that I can contribute to open source projects for up to 20 hours a month as employees and be paid to do this. It's a pat on the back, a way of acknowledging that we benefit from the open source community so we want to give back and help others.

What approach do you take when teaching?

What I think is needed, for many people, is an instructor or mentor then and to help them go to a corpus of curriculum and to go through it, and hopefully inspire them. In my classroom what I encourage is not that you necessarily understand the content, it's that you feel safe to ask questions. Nothing we are doing is going to break the internet, no one is being harmed, so just experiment and if something goes wrong, that's okay! Ask questions, ask your peers, learn from each other. As well as this, I think contributing to open source projects is a good way to gain practical experience, as it allows you to apply your knowledge to real world challenges, collaborate with the global community, and see the real impact of their work.

What are your feelings about the futures of those you teach?

Unfortunately the prospects are not the same as when I entered the industry 10 years ago and this is because of the pandemic. After COVID-19, big corporations made massive layoffs in software departments and a lot of other companies followed. I think my students will have more challenges in learning the ropes in a placement as a junior or as an apprentice than when I was a junior 10 years ago. Nowadays, companies want people that are mid-seniors, but how can you get into that if you don't have the experience? It's hard to keep your spirits high when dealing with so much rejection during the hiring process. I've now seen many people who, after five months of trying to find a job, they just return back to where they were before. That's a horrible feeling. All the toil and sweat they put in, all the effort, and the money is just not paying back. I think that's where getting involved in the open source community can be invaluable, as a way to gain the practical experience these companies want.

To what extent do you think open source aligns with fostering innovation in the broader tech industry? I think it goes hand in hand. We are very blessed that the engineering department is spearheaded by Stu, he's a crazy head and I say that with all the love and good intentions. So for example, what Stu is doing in terms of open source with something called Crux, which is a cross-platform tool that is fully testable. It's all the business logic in one place and we try user interfaces as a side effect. It's something that no one has tried before and we are proving it is a good way to do things. We have been paid by clients to implement that and they are receiving millions of savings on that, which is proving the value of this product we built as open source.

Have you considered international jobs where you'd be working remotely?

Personally, I don't see myself doing that because I have two children and the opportunities here in terms of education and access to whatever is happening in London. For them I think that is invaluable. But I have seen many UK graduate developers going abroad because it's cheaper to work completely remotely, earning a GBP salary but living in Portugal, Spain, Cyprus, Malta, etc.

What would your advice be for developing skills and how to stay connected to that despite seeing a difficult pathway?

I think that most engineers by nature are curious. But what I have found, especially with British educated young people, is that they are afraid to ask questions because they are worried about how teachers, instructors or facilitators will look at them if they are wrong - a very British approach to being scared of being foolish or failing.

So my advice is to not be afraid to ask questions. Maybe no one will answer it but what if they do? And don't think your questions are stupid because you are not too different from the rest of us and that question has probably popped into other peoples heads and perhaps another person has seen that problem with a very different perspective and your question can help them to rephrase it.

Once you have enough knowledge, return the favour, write a paper, write a full-blown comment on an issue on GitHub. Don't be afraid, if you are wrong, the community will help you figure out the correct answer through a collaborative conversation. Also, we are not breaking the internet. Most likely, if you are starting out, you won't be in a very sensitive place where if you do something it will break the system. If something goes wrong we have the system to roll back, make everything green again and solve the issue. So don't be afraid, just experiment and ask questions!

How do you balance technical and non-technical skills?

What I normally do, even just internally within the team, is ask them often, perhaps once a week, to do a demo about what they're doing, explain to us what they're doing, what challenges they faced, how they got around it. Through that exposition and being in front of your peers with them asking you questions, not with a spirit of inquisition but just wanting to pick your brain to find out the thought process, they will learn to order their ideas and think about the process. The process for me to code is not about typing keys on a keyboard. It is about translating into plain English what we want to solve and how we might solve it, then talking about that with peers, being open to critique and then applying that and demoing it. By doing that once, twice, three times a week, everyone benefits - yourself, the client, our mini community. I hope that inspires people so that when they move on to a different project they carry on with that same spirit.

Pedro Martín Valera, Senior Software Engineer, Red Badger

Pedro Martin Valera is a Senior Software Engineer at Red Badger and a Distinguished Faculty member at General Assembly, where he has taught software development to non-technical people for nearly a decade. Originally from Caracas, Venezuela, he arrived in London in 2013 and retrained as a web developer after a career as an environmental science teacher. Pedro is passionate about open source, accessibility, and inclusive education — including the design of General Assembly's first Spanish-language immersive course for people with disabilities. A regular open source contributor, he believes curiosity, asking questions, and giving back to the community are essential to every engineer's growth.

First published by OpenUK in 2026 as part of Open Source Skills Report 2026 ©OpenUK2026





©OpenUK 2026 - Registered Office: 8 Coldbath Square, London EC1R
5HL Company Number: 11209475 - VAT Registration: GB379697512

