Rafael Castro Gonçalves Silva

Professional Cover Letter

I'm a working computer scientist and a developer.

Things that I like and care, but also things that I don't

I like challenges in programming! But not *fake challenges* like understanding frameworks, though this is perhaps one of the most valued things in the industry nowadays. I like to reason about the semantics of the programming language or stuff like "what is the time complexity of this function?" Real challenges in programming usually require you to be creative and *educate yourself* about the problem. There is no silver bullet, you need to learn about the thing that you are solving, you need to read papers about it. It doesn't matter which framework you are using if the problem is NP-Hard or if the data size is so huge that SQL databases can't handle it.

People should notice that what makes someone a good programmer is to understand Computing Science foundations. Yes, tools are important to be more efficient and so on. And engineering is about using those tools and the resources that you have to solve some problem. But a tool just *stays* important for some time, eventually some other new tool appears to take its places. So in this section, I will only talk about ideas and values that I like and car about, but not the tools that I use. See my CV to know about the tools that I use.

I care about code quality, test coverage, static analysis, and well-written code. Tests are not the ultimate tool in software quality, but it is the thing that the industry has widely adopted. I think we should not forget other styles of quality insurance - I mean, there is a huge and diverse academic research field about that. If you are dealing with a distributed system why not model it in a model checker? And, please, chose static and strongly typed languages, and go beyond unit tests.

I think that software development should follow some basic "practical software engineering", like CI/CD, code review, git flow and so on. Communication in programming is also very important, so we should always make clear "why some piece of code exists", other questions usually are easy to answer by just reading the thing. In my opinion, understanding the goal of some function is more important than understanding how it works. A function should have a single and clear goal. A single function shouldn't solve everything.

It looks like that the Agile Manifesto is misundestood or people are forgetting its real meaning. Somehow people are selling "agile" as a coach service. That is bullshit! The Agile Manifesto is about values: "we value individuals and interactions over processes and tools". How can you buy professional values for yourself? Or you understand and have those values or you don't. Okay, you could pay someone to teach you about those values, but is your decision to follow it. The decision of having some values is an individual one. You can't expect a team to behave "agile" because the team leader now says that they will start to use "agile", and they usually do that by enforcing some ritual (just like a cult). The decision of being "agile" is individual, not a team decision. My decision

is to be “agile”, but don’t expect me to like your rituals.
Since the early 2000 the comporate world is going crazy with the “values and culture” thing. It is the organizational silver bullet of the moment. “The employees will be more motivated and they will produce more”. Really? How indoctrination[1] makes people more motivated? At least with me this **obsession** is annoying and it unmotivates me. So I may work in a company with values, but not in a place where this is systematically repeated to the employees.

How I’m like

I’m a quick learner and learning is the most important thing for me, so I wouldn’t like to work in a place where decisions are made based only on using the technologies that the employees already know. Developers should be eager to learn the best technology for this or that specific problem. I rather work in a place where I’m continuously learning.
I’m usually very comunicative and I don’t fill ashamed of asking people to teach me about something. I don’t pretend to know something that I don’t. I enjoy working with teams that do *real team work* and members suport each other.
As you probably already noticed, I’m very critical/argumentative, so I don’t want to work in place that I should just obey

That is the correct word for this, but companies use the euphemism “culture”.