

By Jacob Bernhardt:
Performance Document:

The problem with recruitment by software companies is that it is primarily by programming language or skill instead of by project or problem type. This creates a need for programmers to create published projects that showcase use of specific languages rather than showcasing problem-solving, and guides programmers into believing that they cannot learn or specialize in more than one language or skill. These effects are dangerous and annoying, because languages and skills change and go obsolete much quicker than general problems, and no programmer wants to be forced to work on resume projects all weekend, instead of climbing rocks, to get hired. Furthermore, it's immoral that the FOSS community is used as a corporate showroom for personal resume projects. The solution to these three issues is to invert the initial recruitment ideology problem by training developers in the skill of learning new languages problem solving, and to more fully examine applicants from angles which don't just include FOSS-posted showcase projects.

Self-Assessment Document:

See Minor Points

Major Points

Overall it is a solid manifesto, and effectively sectioned as well.

The first paragraph in Consequences is a little irrelevant, because it gripes about companies not giving out experience, forcing programmers to create it for themselves, and tangentially mentions FOSS abuse. I don't exactly see how this relates to your topic of skills vs. problems, because methods of experience acquisition are independent of the type of experience acquired. The paragraph sort of relates to your concluding statement about more rounded screening processes, but any kind of screening involves experience checking, you are just trying to change which experience is checked. The problem mentioned in this paragraph is a real issue, but it is not related to your topic. I recommend to re-write the paragraph more along the lines of how maintaining a GitHub profile is inherently programming-language focused, if you can argue that.

Maybe get some examples of some fields of work comparable to software which are problem-based and explain why they are so much better. This would be stronger than the analogies you have already.

Minor Points

Recurring error: No Analysis Document, I propose this as an alternative to your dubious proposal.

Almost no mechanical issues (proper comma formatting in quotations, some catalog elements and modifiers of conflicting types, IC-DC fuse).

By Mukul Narwani:
Summary

You're doing it wrong is a manifesto that highlights everything wrong with the hiring and education system where programmers are concerned. It talks about how it's unfair for job companies to hire programmers based on experience but not be willing to offer it to first timers. It also speaks oh how it is incorrect for the community to focus heavily on specializing in specific programming languages and how it is wrong to look at each of these languages as being completely different from one another and not just as tools in a toolkit of a programmer. 'You're doing it wrong' then proceeds to speak of some alternative ideas, different mindsets of thinking that would solve said problems.

Major points

The idea you have when approaching the manifesto is great. The topic nicely fits in with what is expected from a manifesto. It is short and concise and is clearly segmented. What I feel is lacking though is the fact that it doesn't make me care too much. As a normal reader that would've picked up this manifesto, it doesn't make me agree with your ideas; it doesn't make me want to jump up and take a stand for all the programmers. A manifesto is supposed to be a document that implants an idea in the readers mind as if they thought of it, it's meant to be emotionally manipulative to convince the reader to be part of a bigger cause. This document right now feels a lot more like an op-ed. 'You're doing it wrong' talks about a topic most of the reader base can relate to, and this topic works great, but I feel you just need a little bit more work on touching the reader's mind. I feel increasing the length of the main manifesto will do the trick and allow you to elaborate on some points of the document.

Minor Points

There were a few mistakes here and there, which a little bit of proof reading will easily solve. 'You're doing it wrong' is an excellent example of a real world manifesto, that just needs a few tweaks to make it perfect.