

Cameron Kennedy

ENGW1111

Dr. Ethan Whittet

3 November 2017

You're Doing it Wrong

The Problem

Whenever I see a job posting or a university bragging about its computer science program, all I see are languages. “Are you a full-stack JavaScript developer?” they ask. “We teach *C++* in *our* school.” they say. Languages are a tool, and nothing more. If I am building a house, I do not relearn how to hammer nails if I change the brand of tool; I just continue where I left off. Likewise, if I need to change languages for a project, I just do it. This notion of *tool-based* (as opposed to *problem-based*) development pervades this discipline, and is increasingly holding back otherwise talented developers.

The Consequences

So what do I care if someone else has a bad idea? I care when it affects my employment. Looking on Indeed, I see all these postings from companies looking for individuals with *experience*, but none willing to give it. Thus, this dichotomy has given rise to this ideology that any successful applicant *must* have a vast repertoire of Open-Source contributions which he or she submits with his or her resume. The industry thinks that there is nothing more to a programmer than the discipline itself, that we *like* to spend every waking minute programming, and it punishes us if we have any contradicting ideas. Do you like rock-climbing on the weekends? “I’m sorry, but I don’t think you are *qualified* for

this position. Your Open-Source contributions seem to be *lacking* compared to some of our other candidates.” Let me be clear: I support FOSS, and have made several contributions to project. What I disagree with is how the Industry perverts the system into a way of screening for applicants. I support projects because I like or use them, not because I fear I won’t get a job if I don’t.

Besides the threat of being unemployable, this tool-based approach directly impedes even learning how to program. Every day I hear something along the lines of, “I’m a *python* developer, not a *Java* developer!” Many programmers have this idea that they *cannot* (in the sense of being unable *and* unwilling) work in a different language than the one which they currently favor, that if they were to broaden their skills, they endanger themselves by not specializing. The danger is actually the opposite. Frameworks and languages are like fads; they come and go, and exist only during a finite time. By specializing, and refusing learning new frameworks, programmers endanger themselves by existing in a state of impending obsolescence.

The Solution

Since far too often I see individuals with a laser-like focus on the particular, I propose we train new developers not with a single language but many. I understand there is concern that such a training program would bring about developers OK at most things, but lacking specialty; this concern is unwarranted. If developers learn to learn new languages, they ought to be able to quickly pick up new frameworks and languages. Additionally, companies ought to begin looking at the whole of the applicant during the hiring process. Yes, I know this costs more money, but it is ultimately a more efficient solution to their predicament as well. Hiring tests exist today to disqualify *bad* developers, where instead under my proposition, these processes will qualify *good* developers. The difference in

ideology between these two methods is that mine eliminates the middle-ground of those who just scrape by, while also raising the lowest common denominator of programmers in the workforce. Better education leads to better hires, and better hires lead to more stringent tests.

Cameron Kennedy

ENGW1111

Dr. Ethan Whittet

15 November 2017

Letter to *WIRED* on “You’re Doing it Wrong”

FROM: kennedy.c@husky.neu.edu

TO: opinion@wired.com

SUBJECT: Op-Ed Pitch

Dear *WIRED* Editor,

Recently I have found myself considering the type of education I received prior to and the experience I received during my time as a programmer. I’ve worked as a developer for several years now, and I’ve noticed a trend in (especially) new developers: they are wholly unwilling to go outside their comfort zone. Just the other day, I was working with a trainee who I know is particularly talented in development, and they said something which caught my attention. I asked them to assist on fixing a bug in part of our in-house company infrastructure written in C#, and they refused citing that they were a Java developer. This statement left me quite perplexed due to both C# similarity to Java, and (even if one was not based off the other) the completeness of C#’s documentation in the MSDN. I asked this trainee what he meant by the statement, and he told me that he did not feel comfortable working on a production system with his limited knowledge. Normally, I would share this sentiment; the potential for damage in a production system is far greater than in a purely development environment. However, this

system was not mission-critical, a fact known to this trainee. Thus, this exchange led me to take a closer look at we (the software industry) educate and hire individuals. During this search, I identified several deficiencies which I have found in the industry, and wrote about them in the attached document. I petition you to consider this for your next edition, but regardless of your decision I thank you for your time.

Sincerely,

Cameron Kennedy

Senior Developer

H804 Blockchain Inc.

Cameron Kennedy

ENGW1111

Dr. Ethan Whittet

8 November 2017

On “You’re Doing it Wrong”

“You’re Doing it Wrong” is a manifesto by Cameron Kennedy demeaning both the current system of education of and the hiring process for computer programmers. The essay points out the dichotomy of how companies love individuals with experience, but are hesitant to offer it to those with none, as well as the phenomenon of individuals learning programming via a single language. The text manages to meet the qualifications of entry into the genre of manifesto set forth by Boyareddygari, Kennedy, and Narwani’s paper “Manifestos, a Manifesto”. Mainly, these criteria are being short in length, segmented, and persuasive in tone. I paired the text with a fictitious letter submitting it to *WIRED* for consideration as an op-ed. The essay is roughly based off of Loyd Blankenship’s “Hacker Manifesto”.

“You’re Doing it Wrong” manages to meet the structural criteria set forth in “Manifestos, a Manifesto” by being short in length (under 1000 words), and heavily segmented (Boyareddygari, Karthilk, et al. 3). The document consists of three sections: “The Problem”, “The Consequences”, and “The Solution”. These sections flow logically from one another by presenting their argument in a sequential form. By doing so, “You’re Doing it Wrong” manages to be very focused on convincing the reader of its contents, and does not spend time with extraneous syntax, a requirement for earning the designation “manifesto” (Boyareddygari, Karthilk, et al. 11). Thus, because of the document’s brief length, heavily segmented structure, and concise syntax, it follows the guidelines set forth in “Manifestos, a Manifesto”.

The second qualifier in “Manifestos, a Manifesto” a document must reach to be considered a member of that particular genre is to be heavily persuasive in tone. “You’re Doing it Wrong” manages to meet these requirements by proposing scenarios in form of brief quotations, and then responding to them in a didactic, almost mocking format. The seeming dichotomy in the level of authority between the counter-argument and the manifesto lends credit to the document by placing any naysayers in a position of weakness. Additionally, the document appeals to baser aspects of human nature by offering financial incentive to all parties involved if they follow its ideals. Primarily, the document argues that if institutions were to adopt its policies at large, programmers would get better pay, and employers would have more productive programmers.

I based the style of the arguments presented in “You’re Doing it Wrong” off of those presented in “Hacker Manifesto” by Loyd Blankenship. I found his use of voice to be quite persuasive when I read his text, and wanted to try and emulate it in mine (Blankenship). I feel as if it allows for a deeper relationship between the reader and author, which, according to Yanoshevsky, is necessary for an effective manifesto (Yanoshevsky 263). Blankenship’s text is very digestible due to it taking the form of a short essay, a trait which I mirrored due to its ability to concisely deliver an idea. Manifestos are successful so long as they deliver their messages (Boyareddygari, Karthilk, et al. 2), and more so if they do to a large number of people. Thus, I wished to write a text which could be distributed as easily as possible, and concluded that in order to do so, it must be short.

I make a nod to the “GNU Manifesto” with my reference to FOSS, and affirm my support for the movement (Stallman). I strategically placed this line so as to grant myself a moral base on which to stand. “You’re Doing it Wrong” argues that the current state of software education and employment is flawed in an economic and developmental sense, and this statement supports those arguments by appealing to the ethical views of the reader. Most developers support FOSS (as evidenced by the prevalence of things like GNU/Linux), and this statement appeals to that mentality. Likewise, it serves

as a hook to further draw the reader into my argument by demonstrating a shared interest. The goal of a manifesto is to not give explicit instructions for action, but rather guide the reader to a specific conclusion which comes from within, and by priming them with this statement, I provide myself the grounds for further persuasion (Boyareddygar, Karthilk, et al. 2).

Following a set of arbitrary guidelines a manifesto does not make. *Per se*, the format prescribed by critics such as Abastado and Berg mentioned in “Manifestos: a Manifesto” is not the whole picture. Manifestos must be made in such a way that they impact as many people as possible; thus, manifestos *must* have an obvious mode of distribution. *WIRED* magazine was kind enough to publish their guidelines for opinion pieces on their website, and the part of my performance mirroring a potential letter to one of their editors demonstrates adherence to these rules (Pearlstein). I am an individual who has a personal connection to the subject matter; *check*. I deal with potential counter arguments in the manifesto, thus following the guideline which states that I must earnestly consider detractors to my argument; *check*. I am not a company executive plugging my own product; *check*. I boiled my ideas down into one sentence (namely the introduction); *check*. In regards to the email to the *WIRED* editorial staff, the guidelines on *WIRED*’s website prescribe I talk mainly about myself and my connection to and experience with the subject matter. Additionally, I follow *WIRED*’s guidelines for op-ed submissions, namely having the subject line be “Op-Ed Pitch”, and primarily discussing my credibility as a writer.

In conclusion, “You’re Doing it Wrong” meets the requirements set forth in “Manifestos, a Manifesto” by being short in length, concise in syntax, and persuasive in tone. The document condemns the current state of both education and employment in the field of computer science, and offers alternative philosophies to the ones it perceives as being detrimental to those involved in said field. I based its arguments (in a structural sense) off of those present in “Hacker Manifesto” due to that document’s popularity and my ideas as to the causation of its widespread distribution. It is paired with a

fictitious letter to the *WIRED* editorial staff which fits the criteria laid out on their website, namely being more biographical so as to establish ethos, and succinctly explains my desire to publish “You’re Doing it Wrong”.

Works Cited

Blankenship, Loyd. "The Conscience of a Hacker." *Phrack*, 8 Jan. 1986.

Boyareddygar, Karthik, et al. *Manifestos: a Manifesto*. Northeastern University, 29 Oct. 2017.

Pearlstein, Joanna. "Here's How to Submit to WIRED Opinion." *WIRED*, WIRED, 3 Aug. 2016.

Stallman, Richard. "The GNU Manifesto." *GNU.org*, Free Software Foundation, 2 June 2015.

Yanoshevsky, Galia. "Three Decades of Writing on Manifesto: The Making of a Genre." *Poetics Today*, vol. 30, no. 2, 2009, pp. 257-86, poeticstoday.dukejournals.org/content/30/2/257.full.pdf+html.

Accessed 27 Oct. 2017.