Open Letter to Developer Bullshit

###Prologue

So, the other week I made a series of tweets about my thoughts about a tone of general dissatisfaction at developer tooling.

> Real talk, cocoapods is great, and absolutely fabulous thing. Major problems exist with our tooling. Issues nobody really wants to address.
>
> — Sam Marshall (@Dirk_Gently) [September 8, 2014](#)

> The more I see these git submodules vs cocoapods debates the more I lose faith in our understanding of tooling.
>
> — Sam Marshall (@Dirk_Gently) [September 8, 2014](#)

> So maybe instead of complaining about dep management, you learn how hard the tooling problems are and why solutions aren't perfect.
>
> — Sam Marshall (@Dirk_Gently) [September 8, 2014](#)

Yesterday morning I woke up to see some tweets by a friend of mine, Tony Arnold, and the following conversation:

> Gosh CocoaPods brings out the nasty in some people. It didn't kill your Mum, dude, simmer the fuck down and use whatever tool you want.
>
> — Tony Arnold (@tonyarnold) [September 18, 2014](#)

> [@tonyarnold](#) I've been thinking of proposing it at work, but worried about some people's seemingly religious objections to it
>
> — Nick Randall (@n_randall) [September 18, 2014](#)

> [@n_randall](#) yeah, one of the devs I really look up to pissed all over a younger developer on Twitter for bringing it up today. Disappointing.
>
> — Tony Arnold (@tonyarnold) [September 18, 2014](#)

That same morning I have someone make a comment to me about Tony's original tweet in the form of a joke that centered on explicit violence against women. This morning I wake up to have this website sitting in my IRC scrollback

history: shouldiusecocoapods.com. Which has a *delightful* selection of
responses, which I shall quote a few here incase the site goes down:

- Not if you want to ship something that matters.
- Sure, why not? More work for real developers.
- Sure, I'm too cheap to fly to Vegas.
- Of course! The best bad code is bad code you don't have to write. ...badly
- // TODO: Add intelligent handling of this question later.
- Sure, I wanted to download all of GitHub anyway.
- #import "BuggyCode"
- Yes. Wait, were you actually serious?
- How much bourbon do you have and can you drink it?
- Do you work for LEGO? Then... FUCK NO!
- Only if you're a Brony!
- Yes. Just kidding... WTF?
- Not if you don't hate future you.
- Does your client know your home address?
- diff: /../Podfile.lock: No such file or directory
- Sheesh.
- 1-star: This site sucks.
- Not if you have to maintain it after it ships.

I was under the impression that we were professionals here, not children.
However, the more I look around the more I am completely unsure of this. This
feels like a kid's clubhouse that I've been given membership to because I fit the
"acceptable person" criteria. When I originally wrote this post a week ago, I
didn't want this being about one specific incident. I wanted to start a
conversation about how we can have better conversations about the
development process. What I got instead was lousy bullshit.

###Leave your ego at the door

The tech industry is a club house. How we categorize developers, software,
tools, designs, methodologies, even people; is built around exclusion and an
abusive evaluation system. Maybe this had merit at one point when software
wasn't such a part of our culture and society, but this system has absolutely no
merit to us anymore.

####Who are you impressing?

SuperAmazingStartup Co. invites you to interview with them, you ace it, you get
the job, the end.

That is it. You got the job, there is no additional requirement for you to lord your
ability over your coworkers. There is no reason that you need to ignore or

belittle comments made from a Jr dev if you are a Sr engineer. Yet this is something we see time and time again, where we must be ready to prove ourselves at all times and make sure our peers are up to snuff. This atmosphere has created a cult-like environment, which promotes adherence to a set of ideals over the validity of them.

Cargo Cult Science

The phrase "[Cargo cult science](#)" is used to describe what appears to be science, but doesn't follow the protocol of the scientific method. This is usually used in the attempt to achieve an end-goal without the necessary or practical understanding of the requirements to attain that goal.

There is another phrase "[Cargo cult programming](#)", which follows along the same lines as cargo cult science. Cargo cult programming describes the methodology of attempting to solve a problem without full understanding of what that problem truly is. This manifests under the guise of "community standards" or mannerisms of "that is how we do it". In practice this manifests as the application of a pattern or design to a problem without the understanding of the principle behind the application of the pattern. Some refer to this as "copy paste coding", but I think this goes far beyond the "novice" implementation of code to a problem. This stretches far into the realms of developer dogma.

Knowledge

Last night I was attending the Boston NSCoder event, and there was someone that attended that has only been writing code for iOS in the last few months. Having recently finished their undergraduate degree in management, and was looking for a job. In the process of looking for a job they discovered a high demand for project management positions in the tech field. They figured that the best way to get considered for such a position would be to have some background in programming and IT topics. Almost right away they fell in love with programming, and was now actively pursuing that over management. This tale ended with them explaining that when they were first starting to learn Objective-C Apple was announcing Swift at WWDC, and was looking for some guidance in how to proceed into becoming a developer.

I remember exactly where I was when Swift was announced. I was San Francisco, standing in the living room of a friend of mine, surrounded by a few friends and many acquintances that I only knew by twitter handle. All are truly amazing engineers, and I highly respect and admire them. I bring this up because a fair number of them do not have backgrounds in CS, or even engineering degrees. Some of my friends don't have college degrees, others don't have high school diplomas. It wouldn't be possible to pick out who they are based on their code, if I told you then you would probably brush it off as a

joke. They aren't exceptions to the rule, nobody is magically imparted with the knowledge of what code is right and what code is wrong. Knowledge is only gained through collaboration and analysis of ideas, it isn't hoarded or owned. Developer dogma is no substitute for knowledge. The developers that inspire you were not gifted their knowledge, pursuing your own understanding is more valuable and rewarding than just living off what others tell you.

####Ego in Engineering

Engineering is a science, you might call yourself a ninja/rockstar/wizard/whatever, you are a scientist. You have a protocol that you follow to accomplish a desired goal. As an engineer, the only thing I care about is if you are going to impede the goal we are working towards. I don't care what methodologies you have to use to make things happen, so long as that they can and do work. If you feel like you are struggling with it, it is perfectly ok to ask for help. I honestly cannot stress this enough, your peers, your friends have more to gain by helping you succeed than they do by laughing at your failures.

As I mentioned earlier, the pattern of behavior we see is that we cannot ask for help from others, we must be ready to prove ourselves at any given moment. This creates a breeding ground for something called "Impostor Syndrome". Impostor or Fraud Syndrome is a condition where we do not feel deserving of any accomplishment or achievement we have earned. Someone with impostor syndrome dismisses their own successes by writing them off as trivial or insignificant. This is all too common to developers, from the beginning as Jr developers we are never celebrated for our accomplishments. We get told "good job, that is what you are being paid to do", the only example we can follow is when we see the more senior developers be congratulated for a job well done. What separates the work and pushing of understanding required for their achievement as so different from that of a Jr developer learning the ropes for the first time? Maybe we take criticism so personally because our work is not as tangible as that of other sciences. As software engineers we don't build bridges, construct houses, or plan cities. Our toolbox is based on our own thought patterns, logical proofs and how we rationalize and comprehend problems in the world around us. For some of us, hearing "this code sucks" is very difficult. We take it as an personal affront to our how we think and understand. It isn't easy to hear that from anyone, much less to someone that wants to treat it is your problem that they personally insulted you. I think this is a large problem that no one person can come up with an answer for. Personally, I consider the best approach to start the conversation with "have you considered doing" and propose your take on the problem. This goes both ways, you are not going to be directly hostile, and you are leaving yourself open to learning something of the problem you aren't currently aware of. Ego isn't required to exchange ideas, and often results in preventing the open exchange of ideas due to personal biases.

Ending Lousy Bullshit

I am writing this post because I'm fucking tired of experiencing this. Through my own experiences and those retold through stories from my friends, I feel like I am done with iOS development. I don't want to be a part of a community that has such ingrained toxicity for each other based on such trivial bullshit as what development tools we use. I'm tired of hearing stores about how "company culture" basically boils down to your ability to performance under pressure. I don't think I can stress this enough, I don't want to write posts like this given how fucking childish these disputes are.

Tony is 100% correct, this is disappointing. How can any of us, with good conscience, advocate this field to others with this being the environment we are in. No amount of catered meals, access to new hardware, or revolutionary projects can convince me that having to experience any of this bullshit is worth it. Personally, I am extremely tired of seeing nothing but vitriol be thrown no matter what stance you take as a developer. Some joke that, as engineers it is our job to disagree with everything. Our work is produced through a process of logical reasoning, this doesn't correlate with results we might agree with or even understand. However behind whatever we do there was some understanding set the process into motion. To that I want to issue a challenge, next time you are in the roll of either answering or asking a question phrase it around the words "in my understanding". Rather than issue answers as absolutes, explain why they are absolutes. To the person asking the question there is sometimes no distinction between technical absolutes and limitations and the personal tastes of the person attempting to answer. Issuing both as absolutes is the birth of development dogma, and knowledge without query. This results in the creation of more lousy bullshit with the foundations being built from "they said" reasoning.

In conclusion I'd like to link to a project ([AnyoneCanLearn](#)) that was started by my friend Ash Furrow. The aim of this project is to discuss ways to foster and grow communities based on positivity and learning. This is something I think we need more than ever, and to make this more than just discussion but also about action to change the current culture. At the end of the day I see only two sides to this: You can spend your days spewing exclusionary vitriol like "shouldiusecocoapods.com" or you can spend it working towards making a more inclusive and welcoming environment for everyone.