

ZULF ON ISSUES OF CODING

ZULFIKAR MOINUDDIN AHMED

I know what I am good at and what I am not good at. I am a world class Scientific Genius who is greater than Einstein and Schroedinger for my successes. I am a pretty good coder but not world's top by any means. But I've been coding since 1995 in professional environment at Lehman.

There are 19 million coders, of whom 13 million are pro [1]. I do have the skills and experience to be a coding pro but that is not who I am. I am a Scientist, a Finance pro, and I do strong code for other purposes. I am an expert on Design Patterns and so on from experience. I have worked on coding projects at Lehman people used a decade after I was gone. When I do something, it's for serious purpose. I don't do retail generally.

1. CODING ENVIRONMENT

With 19 million coders, coding environment is a massive arena. I don't control where they will all go, and I am not into trying to be a prima donna in this space. So I have my rational manner of dealing with *anarchic chaos*. That is what it is. With 19 million coders, this is a sprawling anarchic chaos. No one controls coding. Microsoft has 120,000 employees, a drop in the bucket compared to 19 million. Who will lead a herd of cats? Not me. So I take a rational approach to this chaos. First, I never ever work on re-inventing the wheel. More lives have been wasted by purists trying to produce a new paradigm in coding than in any other field. You see, in Science, I have confidence to produce new paradigm because I know my genius and insights will make progress. In coding, I am amazed at the efforts of some people and how they did not manage to produce any consensus. I do not know the byways of coding world. So I take a pragmatic approach. First, I never ever solve a problem that is likely to have been solved a thousand times. I look for what other people have done first. It's worth spending weeks to find what others have done on any problem that is roughly standard. Only when it is unpopular do I even think about solving a coding problem. This is because I do not want to be the purist who had a tragic life in coding. I can do that sort of thing in Science because I am an Immortal Genius already in Science, and I can die of starvation doing a Science problem and it's worthwhile but no such luck in coding world. I can die doing all sorts of esoteric problem and be laughed at – for good reason – in coding because coding moves evolutionarily and new tools and technologies totally eclipse others and purists are essentially Darwinian fodder here.

Second, I use technologies that people have been impressed by. I am not interested in contrarian strategies here. I go with Meteor and MongoDB the moment I see relatively smart people doing that. Why? Because I want *them* to be doing the cutting edge research. I have other things to work on where I am the pioneer. I let others lead in coding world. If the 19 million coders are doing NodeJS code

with specific libraries, I'll do that too because it's not worth my while to go against the tide here. I will try to find solutions of other people and only when there is no hope will I even lift a finger and do something original. Why? Because I am not a child. I am an adult and I don't need to prove anything. I need things to work and be compatible with where the crowd is.

It is totally irrational to go against 19 million coders when they are motivated to produce all sorts of open source things that I can use. I use other people's work 90% of the time and only specialise to my cases as needed. I don't want to go against the crowd here at all. I don't even want to have strong opinions because *I don't want to do a lot of work*. I want things to work nicely from other people's work.

2. BILL GATES IS INCOMPETENT

Bill Gates talks about wanting code from scratch in C. No professional in coding would even give him a job if they heard it. I would not. I see coding anarchy. I cannot even imagine doing anything nontrivial that works in finite time if I did not use 20-30 sophisticated components someone else made. React components or whatever. Why would I go against Meteor when I can do some doo-hickey and have a MongoDB backend and a nice frontend that gives me account-pass and all manner of ways to make my front end polished, make most of the db-sync go away and I can focus on real substance such as why anyone would like to pay for my services? That is huge amount of work in psychology and social science etc. Why do I want to increase my burden by a puerile egotistical project of proving how much I can do C code for things that 2000 coders have solved 50 times each for their curiosity or homework? That's just absurd. I am 48 and I have good sense. Never re-invent the wheel to prove yourself to tenth graders. Spend your limited time in life wisely.

3. CODING IS A JUNGLE

Many moons ago, Herbert Simon was interested in Artificial Intelligence, during the 1950s and the Cognitive Revolution in Psychology. I know something about Deep Learning and Convolution networks and Statistical Learning Theory. These things are not Artificial Intelligence! I just saw a baby rabbit in the garage. No one will ever convince me that what goes on in coding industry will ever reach the sophistication of a baby rabbit trying to understand its environment, finding where it can find warmth, staying away from humans with a coffee cup and a cigarette pacing in a garage, etc. It's a *joke* Artificial Intelligence. Maybe next century, when we understand organic intelligence then I will be interested. Coding today is just getting to know how to do some interesting things so that I have some web front end and some database backend and I/O with convenience of the customer. That's all. The substance is still what I am offering that they need in their lives. It's just a distributed way of passing on my solutions to the life problems of people I don't know who are willing to part with their hard-earned money because what I have to offer is valuable to them, even though they don't know me from a can of tomato soup. That's technology component for me. The vast 19 million coders are doing their anarchic set of things, and I do respect their talents but I am not really interested in all manner of technologies that does not solve the problems I

have. I am not going to go through all sorts of things they are doing. It's a goddam carnival.

4. WHAT IS IMPORTANT IS UNDERSTANDING OF HUMAN NATURE AND HUMAN PSYCHE AND FACTORS OF LIFE SATISFACTION

No one in their right mind would reduce Biology and Psychology and Philosophy to Technology. Technology is neutral. For me, technology now makes some things possible. It makes communication and certain computations possible. But the substance is Science, it's knowledge about Human Psyche, of how we want to live and what give us, human beings, higher Life Satisfaction. Then I use technology to implement some solutions that people can benefit from. Technology is a useful tool here, and not the center of attraction. It's enabling, but it does not tell us the substance. It is, in my view, a catastrophic philosophy that says *technology is our Saviour*. No. Technology is neutral. Its anarchic. Some of it can be harnessed to solve important problems. But the solution comes from Scientific knowledge about substantial things like Human Psyche. Every single thing I have seen from pure technology thought about solving fundamental problems of Human Condition and Human Psyche have been total crap. This is because the solution is not technology. Technology is necessary but secondary to Science here.

It's possible Technology will come closer to the core of the solutions one day, but that will be only after the Science is crisp first. For example, what will increase Life Satisfaction given Unique Genome is a matter of pure Science. Technology will not answer this question until Science is clear. Then after there is an implementation that does a reasonable job for 7.8 billion further enhancements might be contemplated that are Technology-centric. But pure Technology attempts to solve this will lead to a large amount of pure nonsense.

REFERENCES

- [1] Nate Swanner, There Are 19 Million Developers Worldwide, Only 13 Million Pros: Study, <https://insights.dice.com/2019/11/25/worldwide-developers-study/>