Some instances where I advised Software Developers to run away from their employer on Quora!

### Q. I work for a startup, our CTO is a big fan of Golang, he insists to use Golang for every projects, including machine learning projects where Python is the de facto standard, what should I do?

Originally Answered Here: https://qr.ae/pNnZby

Run away and don’t look back. There is going to be no use working under someone that fanatic. I am a fan of Go as a language myself, but this is a bit too much. There is no doubt a need of uniformity in the codebase and there needs to be a fixed stack, but I don’t think it is common sense to hamper productivity so much so as to start using esoteric Machine Learning frameworks in Go.

Instead of working on Machine Learning, most of the time will go to try out and fight/enhance Go’s nascent Machine Learning ecosystem. While some people will love enhancing Go’s Machine Learning capabilities (I think there is a decent community), its not something every Data Scientist would want. Also as a startup there will be tight deadlines and working on places where the libraries aren’t optimized will be a nightmare.

No go according to me.

### Q. How do I survive in a startup with undocumented code base?

Stories like this are a norm. I'm a fresher. We've a large code base running for over two years. I was told to implement an interface which was committed over a year - and there was no documentation at all. I asked about the functioning of the individual methods to which I got a reply that I ought to look at examples where it is implemented.   
Is it normal? Does everyone have to read un-commented code for hours while the person who's originally written it could explain it in five minutes?

Originally Answered Here: https://qr.ae/TfxZ43

A lot of startups dont have formal documentation, but that doesnt mean young employees are left on their own.

3 subquestions you asked :

Is it normal?  
No. This shows poor engineering standards of the company. Your PM and Lead are not doing their job well. Most companies either have documentation or KT (knowledge Transfer sessions) before a new Engineer takes over the job

Does everyone have to read un-commented code for hours while the person who's originally written it could explain it in five minutes?   
After they are given an overview of the project, people sometimes do need to read undocumented code, but there needs to be someone whom they can approach for guidance. Maybe their lead or a senior developer in the project.

What should my approach be?  
Run. All I can understand is that the company you work in wants you to be a code monkey. Do you think such mess of code can ever scale ? If you want to be a good engineer, join a place which has *some* engineering practices.

### Q. In my first job in a new industry, I am underpaid with no sign of a raise coming from my current employer. However, I am learning quite a bit/improving consistently. How does one remain focused on continuous learning for now and more money later?

Originally Answered Here: https://qr.ae/pNnZOV

Talk to your employer and understand what timelines they have in terms of promoting you/giving you a raise. You need to understand how your company is doing. There are many possible scenarios:

1. There is no reason your employer cannot give you (justified) raise pending from a long time if your company is doing good. They are just bad people to work for if they reject a genuine raise in this case.
2. If the company is not doing well and you are learning a lot at your work, you need to understand what is the marginal gain in learning (I understand that workplaces with uncertain futures are great learning experiences) as compared to a stable salary you are being offered somewhere else. Are they allowing you to cash out on your learning by generating your own client interface/networks or publishing papers/blogs/patents ? Remember, just learning isn’t enough, putting it out there also is. You need to understand till what level you want to continue learning (Specialization is a field only pays off for a certain level).
3. If the company is not doing too well and you feel it will get better, you are sharing the risk of the company. In that case, you should ask for share in gains as well (equity, promised bonuses etc.). Set a timeline till when you will wait for the company to do better.
4. The fourth scenario where the company is not doing well and you dont expect it to rebound, quit. They would have fired you if they did not see you get better, they might give you a pink slip to cut losses in future.

### My answer to “How can we approach a product-based startup even if we’re ready to work without salary for experience and a full-time job in Bangalore?”

Originally answered here: https://qr.ae/TqMatO

(My startup is not based out of Bangalore, so you can consider this as a third party opinion).

How startups work and how people outside might visualize them is maybe very different. Most people who don’t know a startup inside out, relate them more to the large companies like Infosys, Google, Amazon etc. (and now probably Flipkart, Ola ?) , which they are more familiar with .

Any startup’s ideal employees are:

1. Self taught (or slightly experienced) beginners (not novices) who will be able to take low salaries an deliver tasks assigned to them. The work culture is flexible, so fitting in doesn’t matter, people have to deliver that’s all. (Entry Level)
2. Smart people who are willing to compensated by equity instead of salary and can lead employees like 1 (above) into building something that scales and can raise money. (Senior)

A large company’s ideal employees are:

1. Non-Experienced people/novices who can join and learn the company’s work culture and methodology and learn skills to deliver accordingly. Company can afford to spend good money here and sometime novices in big companies are better paid than beginners in startups. (Entry Level)
2. Then hierarchy wise the company has various openings with high salaries where they acquire senior people, including even the C-suite. (Senior Positions)

Apart from salaries (which are typically low cash + high equity in startups and high cash + low equity in large companies), there is a second difference at the entry level hirings : Startups need people who have somewhat a non-zero level (beginner > novice, a beginner for example will be able to write a simple web service in Python, while a novice will just know Python syntax) of knowledge and can learn on the job, large companies prefer novices whom they can train in their own efficient way. Remember startups are optimized for speed, large companies for efficiency.

Don’t be willing to work for free, try to get to a level where you can get an entry level job is what I will recommend if you want to work in a startup. It is not that hard; do a couple of online courses, don’t pay, just audit or just read good books and get yourself slightly above the curve.

The other way can be , take a novice level job (given you are a novice and not a beginner) in a large company where you are getting the profile you want and get some experience by their training. You can join a startup later if you don’t find their culture good.

If you are beginner/advanced beginner who is applying to work in startups for free and not getting calls, improve your medium (switch to Hasgeek/AngelList etc) , because this is not really possible if you are doing it well.