Problems Faced at Programmers Force 2017

By Sadan Mallhi

Disclaimer

After yesterdays talk in the afternoon I quickly realized that talking in person about my opinions on what I think where the biggest failures of this company over the past year is impossible without creating a hostile environment.

Therefore I have decided to write my thoughts down in a presentable format with the goal to perhaps help steer the company on to the path of more success. Please don't think that I am writing this document in an angry tone, rather I am trying to be as critical and analytical as possible with the sole purpose to help provide a better understanding of what it felt like working at Programmers Force for a Year.

Introduction

There is a difference between criticism and constructive criticism, my goal is to avoid the former and provide as much constructive feedback as humanly possible. However I by no means claim to have answers to all the questions in the world. I may have cooked up some solutions to problems I have noticed in the past year but there are some situations in which I am clueless.

I have divided this document into two main parts, the first section contains problems/mistakes/concerns followed by what I would consider a viable practical solution. The second section is just a list of some problems I have noticed with no comment on how to fix those problems. The reason I have added the second section is because I believe the first step to recovery is correct identification of the problem. Only when we have correctly detected the problem can we start to fix it.

A fitting analogy; when a program has unpredictable behavior, the first course of action is to try to pin point 'why' is there unpredictability in our program. Why is it not behaving the way we expect? We try to 'debug' the problem by pin pointing the exact line of code causing the issue. Only after we correctly identify the exact routine causing the problem can we even being to fix it. After we found out 'what' is causing the problem we think of a way to workaround it, we thing of solutions to the problem after identification of the problem. Therefore I think accurately finding the root cause of the problem is the first step to finding a possible solution to it.

Another fitting analogy is how doctors treat their patients. The first course of action is to meticulously study and observe the symptoms, then using the observations to try to identify or diagnose the problem. If the diagnoses is correct, all the doctor has to do is prescribe the cure to the patient; which could be as simple as a prescription drug or in serious cases a medical procedure. However, if the diagnoses is incorrect, almost certainly the prescription provided will be of little to no benefited to the patient. If someone has Lung cancer but it diagnosed with the FLU, even a room full of antibiotics won't cure his illness.

Section I: Problems with Solutions

Professionalism

Around the time Ali bhai was still working at the office, I remember a group discussion being held regarding employee punctuality. The crux of the issue was that most employees were arriving at the office later than 9:00 AM and the company wanted to find a way to encourage their employees to be more punctual. I discretely remember that at one point it was said that flexible timings isn't really an issue from the companies perspective, rather it looks bad or "unprofessional" to bystanders such as Richard to look at the employee check in timing being 30 minutes late everyday.

This term has been thrown around in many different contexts over the past months, another recent example I can remember is when a huge fuss was created about the time Hassan, Sami and Zaid went to get their passports made. I believe I remember words such as irresponsible, unprofessional, etc. Although I am of the opinion that for certain, every individual in this company must be able to look at himself in the mirror everyday and think "thats what a professional looks like". However, this really is only possible if everyone feels that they are also being treated as a professional. For a moment allow me to play devils advocate and pin point instances of unprofessionalism by the company towards its employees.

Its been almost a year since I have been at the company, I believe 11 months to be exact. With maybe a single exception, almost every month we have been receiving our salaries 10 days into the next month. If I would grade the companies punctuality at paying their employees monthly salaries, Programmers Force would get an F. What justification does the company have to withhold an employees salary for almost half of the next month? Most of the employees at the company have rent to pay, and I can testify that on more than one occasion, Ali bhai mentioned that he had to borrow money from his roommates to send back home at the end of the month only because we didn't get our salaries till 10 days into the next month.

I remember strict rules were set in place for employee punctuality. I believe every 3rd late of the month would be considered as a day off. And salaries are deducted based on the days off. If all these strict rules are set for the Employees, why shouldn't the Employer get charged for every day of the next month salaries are not transfered into our accounts?

Here's another example of company unprofessionalism. I still remember to this day, the first day I came to this office, Khurram bhai interviewed me. After the interview Khurram bhai asked me if I had a laptop I could bring to the office. I answered yes but I would prefer a PC. Khurram bhai replied, "I have ordered new PC's which should be arriving in a couple of weeks. Till then please bring your laptop.". I agreed. I'm still waiting.

Hassan used to bring this crappy laptop for about 6 months. His productivity was massively affected due to his laptop, I am a witness to this. Despite the fact that everyone can see that his laptop is slowing him down, he doesn't have a better laptop to bring, the company really never thought about getting a PC for him exclusively. All the new employees being hired and even old ones like Zahid and Me still bring our own laptops.

Whats alarming and concerning to me is that I am getting the feeling the company is actually turning a blind eye on the matter. "If it aint broke, dont fix it". What was most disturbing for me personally was that everyday Rafey was forced to hand over his personal phone to Adil so he could test the iOS app. Why isn't the company providing the devices to developer and test the products that are being made?

There are probably more things I could rant about but I'll just stop here for this section.

TLDR - Problem: A lot of professionalism is expected from the employees, while the company continues to display unprofessionalism. - Solution: If the company wants its employees to act like professionals, it has to lead by example. Do first ask later.

Project Management

Personally the lack of proper resource management has been the largest source of frustration for me over the past year. I believe I talked about this before but I'll mention this again just because I think its so critical to get this right.

At this company projects are actually not managed at all. Projects are just thrown at the developer and its expected that the developer will wave his magic wand and create a perfect product in the least amount of time. If I can be frank I've seen 16 year olds manage teams of 60 better than Programmers Force has been managed this past year. Allow my to explain.

> "Insanity: Doing the same thing over and over again and expecting different results"
> Albert Einstein

What has been going on this past year is no short of 'Insane'. This past year out all the delivery dates we planned, how many did we actually manage to deliver on time? Every day you ask Zahid, how long will task X take. How often is the task actually done in that time? How many times does this company have to repeat the same mistakes till everyone realizes that the current system of managing tasks and projects isn't working the way it should?

The most recent practical example I can think of is the whole staff timer deadline fiasco. I distinctly remember Khurram bhai set a strict deadline of a certain Friday some time in May I believe. Everyday Khurram bhai would tell Zahid, "I need this done by Friday". All Zahid could say back is okay, I'm trying. Giving deadlines and providing zero technical assistance or guidance is just asking for trouble. Is it expected that by pressuring a developer he'll somehow plop out a perfect project?

A fitting analogy, lets say you want a monkey to build a space ship and go to the moon. You yourself don't know how he's going to do it, you have no clue, your not a rocket scientist right? But you decide to scold, scream at the monkey in frustration that he's not making a spaceship. No matter how much you try to pressure the monkey into making a spaceship, we won't be able to. He's a monkey.

> "You avoid the problem of false delivery by creating an independent definition of “done.” The best way to do this is to have your business analysts and testers create automated acceptance tests that must pass before you can say that you are done. These tests should be written in a testing language such as FITNESSE, Selenium, RobotFX, Cucumber, and so on. The tests should be understandable by the stakeholders and business people, and should be run frequently." Bob Martin

Here is a video of Bob Martin talking about Time Estimates. Its amusing to watch and very educational. Might help bring some insight.

Another point I want to make is the lack of planning before a project starts. Architecture level planning. And time and time again I have seen our projects code bases go to crap just because of what has been explained flawlessly by Bob Martin. Video Link here and Quote below;

> "There is no way to rush. You can’t make yourself code faster. You can’t make yourself solve problems faster. If you try, you’ll just slow yourself down and make a mess that slows everyone else down, too." Bob Martin

TLDR - Problem: In my opinion neither the tasks, nor the projects are being managed correctly in the office. - Solution: Create a stream lined, disciplined system/pipline to help improve the flow of deployable code.

Section II: Problems without Solutions

Team Composition

Ever since Ali bhai left, there has not been a senior developer on the team. To this day I feel like there is still need for someone to be in the office who acts as a trainer of new developers. Someone who is technically sound and can help out developers when they are stuck. Not only that but who's sole purpose should be to help guide the higher level direction of all of the projects.

This video hits the nail in the head and helps explain what I'm trying to say.

> "The training of less experienced programmers is the responsibility of those who have more experience. Nothing can bring a young software developer to high performance quicker than his own drive, and effective mentoring by his seniors. Therefore, it is a matter of professional ethics for senior programmers to spend time taking younger programmers under their wing and mentoring them. By the same token, those younger programmers have a professional duty to seek out such mentoring from their seniors."

Bob Martin