

The Career Path of a Software Engineer

By Xiaomao Wu



A software engineer like me aged 34 with advanced degrees will have been or being asked the following question: "How would you like to be in 3, 5 or 10 years?". That's a tough question and no easy answers can be found.

I was trying to be a top researcher before, but eventually I gave up this idea, because I wanted to work not only on research but also products. Industry R&D is the best choice for me.

My general idea was to do something great, whatever it is, to change industry or people's life. I then started to look for R&D opportunities in the graphics/animation industry, trying to find some place that can not only generate novel technologies, but also encourage research. By research I mean no following-up of other people's paper and then fine-tune it, it should be original influential research and can be adopted by the industry in 1 or 2 years. Crytek was my final choice, and has been proven to be the right choice for me.

I started as a R&D Animation Programmer, then Senior Application Engineer, and after that, Lead Software Engineer of a central R&D team called **Cinebox** (<http://mycryengine.com/index.php?conid=59>).

The big differences between a software engineer and a lead engineer are: 1) I need to build the team and motivated in a right way; 2) I have to spend time on communication between teams and waste some time on meetings, although I tried to optimize the meeting time; 3) I have to give responsibility to my team members, trust them, and start to accept that my success is not only defined as the amount of code I write or the feature I developed, it's the overall team performance; 4) My coding skills are not grown fast as before. My team members are younger than me, learn faster and have more time than me cause they don't have kids to take care of:).

I was a bit uncomfortable of those changes at the beginning. But I then realize that this has to happen, and what I should do is to accept them, try my best to help my team members to dig out their potential, and define my success to be the success of my team members, as well as the product we are making.

After that, I started to think about what I'd like to do next in future? I was thinking about becoming a manager, a director, or creating my own company. Becoming a pure manager without requirement on technical background isn't that interesting for me. I'll quickly lose my competition and start to compete with the guys being trained in a MBA/EMBA school and with many years' of experience. So I'll need to always keep updating my technology skills, as well as analyzing, communication and decision-making skills. That being said, whatever I'm gonna to be, I'm interested in a position that require both strong R&D skill as well as great leadership experience.

I read tons of posts before. And until today I found the most useful and practical ones:

1. **Career Calculus** (http://www.ericssink.com/Career_Calculus.html). The most important thing for us, is to learn every day from your work, to keep your competency. We should learn from mistakes, new knowledge and from our peer workers. Every week we should ask ourselves, what did I learn this week? Did I gain more experience? If not, how to improve next week? Learning should be a life-time thing. Never stop reading and experimenting new things. That won't burn your brain, instead, it will make your brain works like a butter when you are 70 years old. I observed couple of retired professors and I found out that the one with the most clear mind is the one who continuously learn. That professor was learning C++ from me when he was 65 years old.

2. **Manager Secret Sauce** (<http://steve-yegge.blogspot.de/2006/05/not-managing-software-developers.html>) by **STEVE YEGGE** (<http://www.blogger.com/profile/14812997485690838920>). If the reasons why you want to become a manager are that you don't want to program anymore, or you want more power to control people, then you are already hurt your career. This way you have less chance to become a good manager. That's exactly what I was thinking about. Things become more clear after I read that post. My personal interest is to lead a high-tech team, develop top-class IT products that change the industry's pipeline, or even change people's daily life. There are many valuable points in this post. I strongly recommend you to read it.

My general recommendation for you would be:

1. Think about what do you really want to be. Wake up your child dreams. Break some limits if you still have passions for your child dream. That's your original power in your body that will bring you to

Lead Editor: Xiaomao Wu, Crytek

The *C/E* Blog brings together expert bloggers whose primary focus is entertainment technology. Like the editorial content in *ACM Computers in Entertainment* magazine, the blog will focus on art, music, games, television, movies, society, education, interviews, conferences, news and events related to entertainment technology.

Blog Submission Guidelines

success.

2. List your strengths and weakness, by discussing with your good friends, your family and your lead. Which ones can be overcome by learning? This will assist you to decide which way you want to go.

3. Communication skill is something you always have to learn, even if you just want to be a principle programmer working in a dark room alone. Leadership skill is important even if you just want to be a software architect or senior software engineer.

Whatever you're gonna do, the most important thing is try to find something you like, if you cannot, find a way to start to like your job, cause half of your awake time will be spent on it and obviously you don't want an unhappy life isn't it? Try your best to do your job in a creative and efficient way. Even you are just a cleaning lady in a company, you can think about how to clean the kitchens faster to save you time, and how to keep yourself happy. You might have a chance to run a cleaning company successfully if you find out the current issues of the cleaning service and how to solve them.

Any comment is welcome.