

DEPARTMENT OF SOFTWARE ENGINEERING

FRESHMAN SEMINAR 4010-101

WEEK 10

Interview Paper

Introduction

In this report I summarize the information gained through a formal interview with [REDACTED] on his profession as a Software Engineer.

1. Residence and Career

[REDACTED] resides in Los Altos, California and works five miles away in Cupertino, California. In Cupertino, he works for the well known company Apple, Inc. a producer of many technological products such as the iPod, Apple TVs, AirPort, TimeCapsule, the Macintosh Computer and other personal computing systems. The iPods are, as you probably know, a personal mp3 device created by Apple which gained immense popularity for its features. Apple TVs are small devices allowing a person to connect it to a TV and watch their iTunes content. AirPort is Apple's wireless station with TimeCapsule being the more advanced version with a terabyte of data backup.

[REDACTED] worked at NeXT Computer, starting in 1990 when it was still a separate entity from Apple, Inc. In 1997, when Apple bought out NeXT Computer, [REDACTED] continued to work under the new management. The first part of his job is as a software engineer. He defines this by saying that he is responsible for the "specification, testing, and occasionally the implementation of computer language features and associated runtime support functions." In short, he helps come up with the code, occasionally helps to actually code it, and then spends the time to test the completed code until it meets the set standards. Recently, he has helped introduce garbage collection methods to C and Objective C along with closures, and has implemented them at Apple, Inc. He is also active in the ISO standards committees in trying to get garbage collection and closure standardized into new standards for programming languages.

[REDACTED] works in a team of five members at Apple, which he describes to be smaller than the typical size of one of their teams. They use mainly C derived languages (C, C++, and Objective C) but also use Ruby, Python, and Makefiles at times.

His typical day at work is first spent reading and responding to any e-mails he has received. He also uses this time to monitor around ten discussion forums and stay updated on the latest news in business or technology related subjects. Some of the forums he monitors are ones where he is the main responder, these being they ones involving subjects that he is knowledgeable in such as garbage collection. He also normally reviews a list of proposed API's, which are discussed in depth on this forum after being proposed. After this work in the morning, he works on drafting proposals or code if he doesn't have to attend a meeting.

After years of working with NeXT Computer and Apple, he has learned that the most important skill in software engineering is the ability to communicate. He says that even though he has had to do some very difficult coding on the internal mechanics of an operating system, he found it better to convince other people that his idea will work and allow them to code and debug the program. He especially values the ability to transform the ideas and code in his head into something that can be grasped by those with less of an understanding of the technical jargon.

██████████ favorite part about being a software engineer is to look into society and see some unfulfilled need and to develop a solution, then to push the solution until someone accepts it and watch as that child of his mind is given life and accepted. He especially gets excited about the ability to help alter and improve programming languages, and to see those new features used to help build newer and better programs. He specifically mentions the new iPhone developer kits and the upgraded language features that are available for it. The only downside to this, he says, is that he has few peers.

2. Reflection

After my interview with ██████████, I looked over the information that he had given me and realized that some of the things he mentioned I only had the slightest of ideas about what they meant. For example, I originally knew very little about what APIs were. I had heard the word before, but it wasn't until it appeared in his interview that I took the time to look up more about it. I did this on my general principal of not using words whose meanings I am unsure of. So I researched it after his interview sparked an interest and found that APIs are the connection between the applications that run on a system and the operating system itself. In this same sort of fashion, I researched the different Apple products that he mentioned that I wasn't familiar with and began reading about closures to better understand what they are and what they do.

Furthermore, I found that learning more about what the actual work of a software engineer entails to be interesting and helpful to my ideas of my future job. I originally wasn't sure how much of the actual coding a software engineer did or how deeply one was involved in the design process. After my interview with ██████████, I learned that software engineers lean more towards the design and concept of a new program or application than being the ones to grind out line after line of code constantly. Of course, I also realize that different employers will have different ways of doing things, and that any future experiences I have may not be the same as ██████████. I also believe that his valuing of communications as the most important skill of his job to be fairly wise. As a software engineer, one generally works on teams with other people and has a superior who may or may not be versed in the more

technical aspects of the programs that get created. There is also the customer to consider, as they may not know exactly how to express what they want in the clearest of terms. In both of these cases, the ability to communicate between programming language and “normal” language is nearly vital to the success of a program or an idea. It seems that if you can convey a good idea to other people, the support the idea gains from the ability to communicate it with others greatly increases its chances of being made a reality. As he said, he likes it better when he can get another person to like the idea enough that either that person themselves will code and debug it, or assign other people to do so.

I also feel that his view upon why he makes software is similar to mine. He says that he likes to be able to look into the world and see an unfulfilled need, and then invent a solution for that need. In short, he likes to make stuff to help people. This reflects my personal reasons for deciding upon software engineering in that I want to create things or help create things and if they help people, all the better. I always like to build with Legos as a child, and build things with whatever materials I could find. When I began programming, I hit upon something wonderful. I realized that it was greatly intriguing, and that just bringing an idea into creation was a wonderful thing. I realized that it's the ability to look at something and realize that you created it, or parts of it, out of an idea. From what I gleaned of [REDACTED] during the interview, I believe it is nearly the same for him too.

His strides to expand and better the programming languages themselves also fascinated me and increased my respect for him. I realized that by improving upon the programming languages, he would be aiding anyone who wanted to use them. I realized that this could help people with a more limited knowledge of programming achieve a certain result with more simplicity and less stress. I would imagine that new, easier to use functions or functions that perform what have become commonplace actions could greatly help students and programmers alike. It could even help introduce more people into programming and software engineering as the concepts become slightly easier to grasp and the functions become less complex. I applaud the efforts of people such as [REDACTED] in their attempts to either make the programming languages easier to use or give the programming languages the ability to do more.

I enjoyed interviewing [REDACTED] and believe that by hearing the testimony of an actual, on-the-job software engineer that I have gained a better understanding of what it is to be a software engineer and have also gained a greater enthusiasm for the prospect of having a job as one.

4. Additional Information

Transcript:

What is your name, email address, and what city do you live/work in?

I work in Cupertino, CA and live in Los Altos CA, 5 miles away.

What is the name of the company that you work for? What does the company do? And how long have you been employed there?

I work at Apple, Inc., a well known company that builds personal computing devices, including Macintosh computers, iPods, AppleTVs, and related networking and storage backup products (AirPort, TimeCapsule).

I worked at NeXT Computer starting in 1990 which was acquired and its technology adopted by Apple in 1997.

Describe your position and the types of projects that you work on.

I am, foremost, a software engineer responsible for the specification, testing, and occasionally the implementation of computer language features and associated runtime support functions. In recent years I have introduced garbage collection and closures to C and Objective-C, and have managed/contributed to their implementation and adoption within Apple and am active with ISO standards committees on their adoption into forthcoming language standards.

How large is your project team? Is that a typical team size?

I work on a team with five members. That's on the smaller side of typical.

What types of languages, technologies and tools do you use?

We use C, C++, and Objective-C primarily, but also dabble in Ruby, Python, Makefiles.

Describe a typical work day.

In the morning I generally read & respond to email traffic - I monitor about ten discussion forums - and do web surfing on interesting technology or business. There are a couple forums where I am the principal responder (a Garbage Collection discussion list, for example). Another is an

API-reviewers list where all new proposed APIs are presented and discussed. In the afternoons I work on proposals or code and get sucked off to meetings.

What skills are important for your job?

With my seniority and experience my communication skills are paramount - I learned long ago that although I can and did code very tricky things (operating system internals, locking) it was far better to sell ideas to others and convince them (or their managers) to actually do the coding and debugging. I've managed at times and so also know how to describe new work in terms that they can understand.

What do you like most about your job? Like least about your job?

The most fun thing is to see a need, propose/invent a solution, advocate the solution, see it accepted, built it, deploy it, and see it adopted. In language work there is a tremendous multiplier - if I can provide something that makes programming easier-safer-better then programmers can and will build better products. There is a lot of use of newer language features in the iPhone developer kits for example. That's cool.

The downside is that I have few peers.