TITULO ABREVIADO DO DOCUMENTO

Independent Studies Mozilla Recruiting - Report

Filipe Gonçalves

(Activity report)

Abstract—The hiring process at Mozilla is comprised of several stages. Starting with an online programming challenge, candidates must attend to meetings, are submitted to 3-5 technical interviews, and have a non-technical interview. This report describes my experiences while applying and interviewing with Mozilla for a summer internship position, and discusses how that led to an offer.

Index Terms—Mozilla, interview, recruiting, web development, internship, FirefoxOS

1 Introduction

EVERY summer, hundreds, or perhaps even thousands, of tech companies, especially in the United States, have a considerably reasonable amount of open positions for summer interns to fill. The process for getting a position in big companies is usually quite competitive, and the inner workings of the recruiting process - from the moment a candidate's résumé arrives at the Human Resources' department - is generally an opaque procedure. This black-box bubble that is seen from the outside may interact with a candidate just to say "No thanks, we're not interested", or it can become a whole new world, filled with opportunities, when the right candidates knock its door.

Within the next sections, I will describe the activities that took place under the recruiting process of one of these companies, namely, Mozilla Corporation. Mozilla is most widely known for its browser product, Mozilla Firefox, but a lot of other products are under Mozilla's belt: bugzilla, a bug tracking application; SpiderMonkey, a javascript engine; FirefoxOS, a fresh project aimed at building an open mobile operating system; Thunderbird, a mail client, and others.

Recruiting, both for the employer and the employee, is always a hazardous and costly process. It is also an exhaustive and shaky process for the candidate. Hopefully, this report can shed some light on some of what apparently is the most common practice in the tech industry.

2 Where it all begins

The standard job application, a cover letter and a résumé, is a phenomenally weak way to introduce a candidate. It gives only the faintest clues as to the quality of an applicant, making it pretty hard to tell much about him/her [1]. In fact, it is extremely easy for a candidate to make it look like they are a genius: all it takes is a careful choice of words. This is the first contact a company has with a candidate, and it is already getting a distorted, nirvana view of the reality.

To fight this sad truth, as well as to reduce the amount of noise in the résumés pile, Mozilla seems to be comfortable with a novel approach: candidates must solve a challenge before applying. This is still far from good, because candidates can still cheat, or ask for help, but according to Mozilla, at least it rules out poor candidates who aren't motivated enough

• *Filipe Gonçalves, nr. 70592,*
E-mail: filipe.a.goncalves@tecnico.ulisboa.pt, is a student in Information Systems and Computer Engineering,
Instituto Superior Técnico, Universidade de Lisboa.

Manuscript received May 30, 2014.

(1.0) Excelent	ACTIVITY					DOCUMENT						
(0.8) Very Good	Objectives x2	Options x1	Execution x4	S+C x1	SCORE	Structure x0.25	Ortogr. x0.25	Gramm. x0,.25	Format x0.25	Title x0.5	Filename x0.5	SCORE
(0.6) Good												
(0.4) Fair	2	1	3.6	0.8	7.4	0.2	0.25	0.25	0.23	0.5	0.5	1.93
(0.2) Weak												

to go through a nice programming challenge, thus saving the cost of interviews.

Either way, that's where it all begins: with the challenge. My very first interaction with Mozilla as a developer was to solve their timed progamming challenge (1 hour) in order to gain access to the application submission page. Typically, the challenges don't involve very complex algorithms, in fact, they only test for basic Computer Science knowledge and problem solving capability. The challenge was described as follows: You have a function $h(s)$, which receives a 9-letter string over an alphabet Σ, and returns a 64-bit hash of that string. You are to code a function, $h'(i)$, that reverts the hash, i.e., given a 64-bit hash i, return the 9-letter string s such that $h(s) == i$. There was also a window showing the code for $h()$, and the elements of Σ. The brute-force solution of testing every possible 9-letter string that can be formed from Σ runs in $O(n!)$ time, which is absolutely insane. However, after analyzing the hash source code, a bit of thought and mathematical manipulation shows that the challenge can be reduced to the problem of finding the base-37 representation of a base-10 64-bit integer. This is the property that my solution exploits. 60 lines of Python code, including a long comment explaining my ideas, were all it took to be accepted on the automatic testing system. As usual, the power and beautifulness of Maths yielded a linear algorithm.

2.1 What now?

The next step was to submit a résumé and a cover letter within a 24-hour period. This can be a tough deadline for someone who has never written a résumé before, but fortunately, I reused a résumé template from the previous year, when I applied to a summer internship at GAIPS, and polished it a little bit, using some fantastic tips from *The Google Résumé* [1] (more on this in the learning report). I wrote the cover letter from scratch, where I described my passion for Computer Science, programming, why I like Mozilla, and why I think I would be a good fit for the team during the summer. Before submitting both my résumé and cover letter, though, I had a native English speaker check it for inconsistencies. I know lots of companies where candidates are rejected upfront because of spelling mistakes in their résumé.

3 Is anyone there?

It took a while to hear back from Mozilla: precisely 1 month and a half. Mozilla explained they were receiving lots of applications, and they were having a hard time processing every single one of them. Moreover, at Mozilla, developers, not recruiters, review the résumés: this is because recruiters lack technical knowledge and can easily be impressed by technical jargon that in practice is meaningless. Instead, developers go through the candidates' résumés. This is why it takes so long to hear back from Mozilla.

After the reviewing process, Mozilla determines whether it is worth scheduling an initial phone interview with the candidate.

4 The first real-time contact

Interestingly, the first phone interview I had was with a recruiter. It was more like an informal conversation where she asked me about my interests, both technical and non-technical, what is my summer holidays period, etc. The ultimate goal was to match me with a Mozilla project and team. She explained that once I was matched to a project, I could expect to have 3 to 5 technical interviews with different Mozilla engineers, and one non-technical interview with the team's manager, if I made it in the technical interviews.

To find a project, I had to attend to several meetings from different teams. Mozilla broadcasts meetings in real-time, and anyone wanting to participate is welcome. After the conversation, eventually, the recruiter emailed me some of the meetings she thought would be good for me to attend, given my interests. I'd be attending one meeting per week, and she would reach out to me after all that to see if I found a particularly interesting project. Meetings lasted for approximately 2 hours, and most of them started at 11:00 AM. California, where Mozilla is based, is 8 hours behind Lisbon, making it 7:00 PM for me (that wasn't a

very comfortable hour, but I never failed any of them).

The meetings were pretty interesting. Typically, the team discussed features proposals, defined priorities for the upcoming weeks, and developers would briefly describe the current status of whatever sub-project they were working on. Technical discussions were also common. My participation was insignificant at this point, though I did get the chance to slip inside a technical discussion during the WebAPIs team meeting. One of the programmers was complaining about how hard it could be to implement a cache policy to speed things up in one of Mozilla's subsystems, and out of nowhere, I suggested making use of splay trees (a data structure resembling a balanced binary search tree, but optimized for quick access of recently accessed elements). I feel like I scored some points here, as no one at that point expected an outsider to throw in a possible solution. But I ended up interviewing with another team.

5 The Technical Interviews

The project I found more interesting was FirefoxOS, and that's the team I interviewed with. FirefoxOS, formerly known as Boot2Gecko, is developed by hundreds of programmers at Mozilla; the goal is to create an open mobile operating system based on web technologies. There are numerous teams working on different components of FirefoxOS at Mozilla. I was interviewed by the team responsible for the backend of media apps.

The first 3 interviews were highly technical and very long (the smallest was 90 minutes). The interview layout was different depending on the interviewer: some interviewers went through my résumé, asking me questions about what I listed. One of them was so interested in an entry on my résumé related with compilers that we were speaking about it for 45 minutes. One thing, though, was common to every interview: one or two problems, where I was required to write code in an online collaborative editor. For those that presented me 2 problems, the first would typically be very easy, and the second somewhat harder. I am not allowed to disclose the specific questions, but they were primarily about concurrency and threading, bit manipulation, linked lists, recursive algorithms, trees, and arrays. One of these problems was particularly hard, and the interviewer just throwed it off at me immediately in the beginning of the interview.

6 The Non-Technical Interview

My performance was good enough to be called for a non-technical interview. This is the final stage in the process, where the team manager conducts a non-technical interview with the goal of assessing other skills that are just as important for the job: although Mozilla doesn't expect a lot of experience from students applying for a summer internship, they want to know how much team work we have done at university, how complex and big are the projects, how much time we worked on them, which software development processes we know ("Are you familiar with agile? Pair programming? Scrum?"), and other behavioral skills. During this interview, I was asked things like "Tell me about a time when you had to face a group of people disagreeing with you in a presentation", "What are your weaknesses?", and "Describe when and how you dealt with a problematic teammate".

Fast-forward 2 days. I am sitting at my desk, doing regular work for university, and I get an e-mail from the recruiter, with subject line "Summer with Mozilla". It was an offer for a summer internship with Mozilla for 12 weeks, working with the FirefoxOS media apps team. Later on, the recruiter scheduled a call to discuss the amazing offer conditions. Needless to say, I accepted it.

7 Conclusion

Mozilla's hiring process is a little bit different from other companies. Candidates are required to solve a challenge if they want to apply, and there can be a huge time gap between submitting a résumé and hearing back. The first contact comes from a recruiter, who schedules a phone interview to determine which teams or projects are fit for the candidate. This results in

a list of a dozen meetings that the candidate is adviced to assist to remotely. Hopefully, the candidate finds some of these to be of interest; it is then the recruiter's job to schedule interviews with the team. Most of the interviews are highly technical in nature, with a slot to discuss some of the résumé entries, and a slot for one or two technical problems where the applicant is expected to write some code. Code is always written in an online collaborative editor, so the interviewer actually sees what is typed character by character.

With great success in the technical interviews comes the final step: the non-technical interview. This is mostly a behavioral skills interview, mainly focused on soft skills. There can be unexpected and hard to answer questions. Not hard in the same sense of the technical, algorithmic problems, but hard in the sense that no good answer comes to mind. This is especially tricky with questions like "Describe when and how you dealt with a problematic teammate". Really, when was the last time that happened? It's very easy to lose track here.

Recruiting at Mozilla, and at any other big tech company, can be a painful process, and both the candidate and the company go through a lot of work, but in the end, it all adds up to success, and it is extremely rewarding.

In this type of document (Technical), the CONCLUSION should start with a SUMMARY of the subject adressed and then should highlight the results.

REFERENCES

[1] Gayle Laakman McDowell, *The Google Résumé*. Hoboken, New Jersey: John Wiley & Sons, 2011.

APPENDIX
STATEMENTS OF EXECUTION

The following picture shows a screenshot of the offer e-mail. I decided to include this instead of the scanned offer contract, because the latter is several pages, and is full of legal clauses.