

AIM IM with brantgurganus200.

8/12/06, 11:56 AM

This transcript has been (slightly) edited for spelling and punctuation.

Brant Gurganus: You ready for that interview?

Olivia Ryan: Hi Brant. Yes. you?

BG: Sure.

OR: Great. I have a series of questions here, but just cut me off if you need to. When did you begin using computers?

BG: My grandma purchased a computer for my family as a Christmas gift when I was in second grade. That is when I started using computers.

OR: How did you get interested in working with computers? Do you remember the first programming project you worked on?

BG: My mom had some knowledge about computers and I started learning from her. After that I started reading books about computers and I read the computer manuals. I got into programming when I found the Q-Basic interpreter included in MS-DOS 5.0 and inherited some old 3-2-1 Contact magazines from my step uncle. At the time, there were two computer-related sections in the magazine: The Slipped-Disc Jockey and Basic Training. The Slipped-Disc Jockey featured questions and answers about computers. Basic Training featured programs written in variants of BASIC that I would type into Q-Basic and run. I started getting books from my school library on Basic and tweaking the programs I would type in to figure out how they worked. That is how I got into programming.

OR: Have you also had formal computer training?

BG: I have had formal training in the sense that I am currently pursuing a Bachelor of Science degree in Computer Science, but I would say that the vast majority of my computer-related knowledge is self-taught.

For example, I took a course in databases, but I already had experience with them from an internship out of high school. I didn't learn anything from that class.

OR: Okay. When did you begin contributing to open source project and how did you first connect to open source (e.g. magazines/newsletters, bulletin board systems, conventions, clubs, etc.).

BG: In seventh grade, there was an opportunity to take some weak or two long electives on various topics. One of the topics I picked was HTML.

As a consequence, I started looking into the World Wide Web Consortium and their

specifications.

OR: I see.

BG: In looking up HTML-related information, I started hearing about something called Mozilla. I finally found mozilla.org and tried it. At some point I filed a bug and found that you could actually contribute to the project.

OR: What year was that?

BG: I am thinking around 2000 or 2001, but I can probably look in Bugzilla to find out exactly. I think Mozilla was around 0.9.5 or 0.9.6 at the time.

OR: Did you find it easy or difficult to break into the community initially?

BG: I think it was pretty easy to get into the community. It was also fairly easy to get access to contributing the Mozilla Website. However, I still have not obtained access to the actual code tree.

OR: How long would you say you observed things on mozilla.org before you filed a bug?

BG: I really don't remember.

OR: Okay. What Mozilla projects have you worked on? In what capacities have you worked?

BG: I did tech evangelism work for a while. For that, I wrote a history of the tech evangelism project and Tristan Nitot sponsored a contest for doing work on the most Europe-related bugs, which I won. I also ported the help browser from Seamonkey to Firefox. That has probably been the biggest contribution code-wise. I also did QA work on the Mozilla Calendar project for a while.

OR: How did you get interested in tech evangelism? What audience were you hoping to reach by writing a history of tech evangelism?

BG: I was interested in tech evangelism because it was something I felt I had the skills to do and it definitely had a need since there were sites that were still rather IE-specific. For the history of tech evangelism, I don't think I had a particular target, I just had found bits and pieces of tech evangelism history in different places and figured I'd put them into a single document.

OR: When working on Mozilla projects, do you generally work alone or within groups? How is the division of labor generally determined? Who is in charge?

BG: As far as physical people I worked alone, but for tech evangelism and the Mozilla Calendar, I worked virtually with other people through IRC. The Tech Evangelism

project was headed by Bob Clary at the time. The Calendar project was headed by an Oeone employee. For Firefox Help, I worked with R.J. Keller primarily over instant messenger. Since he received commit access to the Mozilla Tree, he became the driving force.

OR: Did you generally decide what you would work on or did someone assign work to you?

BG: It was all me. There were always pages here or there that asked for help in certain areas, but I always worked on things I cared about. That is why I felt I have done better work for open source than when I worked for commercial companies. I don't use Powerway or SQL Server significantly, so my work at Microsoft and at Powerway seemed boring and meaningless for me. I have learned from those experiences though. There is still one thing I'd like and that is working commercially for an open source company to see how that experience compares.

OR: From your experiences working on Mozilla, how do you think working for an open source project differs from working on commercial products?

BG: When you work on an open source project, it is very likely that you use that product, so you are driven to fix issues. When working on a commercial project, you may be working on software that you don't use so you don't have the passion to work on it.

OR: Does the management of the two differ? Either your relationship with "managers" or the way in which you manage your own time, work flow etc.?

BG: I think the management of open source listens more to its community or at least has a closer connection to it. The reasons for decisions are also more likely to be explained. I think general product direction is similar and driven by the same desires, but the enforcement of that is different. I find the culture of developers to be different as well. It seems that open source developers are more receptive to bug reports and such while closed source developers don't care once the project is over. I have also noticed (though done no actual research) and found that open source developers are more likely to be more (trusting?) of the world. For example, most developers for Mozilla use screen names that are derived from their real name. My AIM screen name is such an example.

OR: Interesting. You said above: "I think general product direction is similar and driven by the same desires, but the enforcement of that is different." Can you expand upon that a bit? Do you think both are user-focused or driven by market share or did you mean something else?

BG: I think that statement applies to larger projects, particularly ones that are commercially related (Mozilla spawned from Netscape and OpenOffice spawned from Sun Office). Smaller projects just care about solving a problem of the developer and don't care about issues like market share because it is a niche application. Regarding enforcement, open source management doesn't control you through salary or

advancement, but commercial management does.

OR: What are the benefits then of working on open source software? Why do you volunteer?

BG: I work on open source for several reasons. It gives me experience with bug tracking technologies and source control technologies. It provides connections to other people in the industry. It is fun. It also helps solve problems I encounter without waiting for a company to fix the problem.

OR: Do you think these are the reasons why most people volunteer?

BG: I think solving a problem encountered is a major driving force. I think communicating with other people is also a driving force. I don't think my other reasons are as significant for other people.

OR: To what extent has Mozilla relied on the work of volunteers?

BG: I think Mozilla relies more on volunteers now that it is not affiliated with AOL or Netscape, but many of the core developers are employed by Google or Mozilla Corporation or other companies with interest in the success of the Mozilla products.

OR: Why do you think Mozilla Firefox has been able to attract a large number of users? What sets it apart from other open source projects? What sets it apart from other Mozilla projects?

BG: Netscape had a cult of followers that transferred to Mozilla as a start. That same argument probably applies for OpenOffice as well. When Firefox came along, it was clearly doing things better than its competition. Namely, it was actually being maintained. Then, it received exposure when Firefox 1.0 was released in the New York Times among other places. Now that Microsoft is back in the game with IE7 and as long as they stay in the game, Firefox may start to flounder again though.

OR: What do you think Mozilla will have to do to keep Firefox from floundering?

BG: They have to innovate. The product that is innovating is the one that gains. the product that plays catch-up flounders.

OR: The Spread Firefox website states that Spread ffx was “founded on the same principles of community involvement that drive the development and testing of Firefox.” How do open source principles influence marketing techniques? Do you see marketing as something coders can do or does this require marketing experts?

BG: It is driven by open source principles in that ideas are proposed and commented upon and evolved to something that is better than the original. Marketing, though, requires skills that a person who can develop may not have such as artistic skills and

being able to communicate in a non-technical fashion.

OR: How would you define a successful open source project? What elements or practices do you see as necessary for developing a successful open-source project?

BG: Success is defined differently for different projects. I would say Firefox is successful because most people recognize the name now.

Competitors can both be successful because they define their successes differently. Epiphany is successful because it follows the Gnome UI design guidelines better than Firefox, but Firefox is still successful.

OR: Do you consider open source software projects as a public service?

BG: It depends on the origin of the project. Personal projects are open source as a public service in many cases. Commercial projects that turn open source want to take advantage of the fixes contributed by the public, not necessarily provide a public service.

OR: I have about 3 more questions, if you have time.

BG: Sure.

OR: Okay, thanks. Switching gears for a moment. How important are comments in the code to the smooth development of Mozilla software?

BG: I can't speak about Mozills specifically since I haven't done development on the actual compiled code. However good comments and documentation tremendously increase understandability of algorithms used and interfaces between components.

OR: If you know -- Is strict ownership of specific areas of code enforced?

BG: What do you mean, technical ownership or copyright ownership or something else?

OR: Not copyright. But are there certain areas of the code designated to one individual? Like a module owner?

BG: There is definitely technical ownership of areas of the code. That is what the review/superreview system for patch review is about. A review ensures that a patch to the JavaScript console, for example, doesn't break something in the JavaScript console and is reviewed by somebody familiar with the JavaScript console. A superreview makes sure the patch doesn't break overall application architecture and is reviewed by somebody from another area of the application that is familiar with global application architecture.

OR: Do you think this system works well?

BG: Yes, it works well. It helps ensure that a patch isn't checked in that is a so-called

hack where it should be fixed a better way.

OR: Last question(s): What (if anything) do you think the popularity of Firefox will do for the open source software movement as a whole? What's the future of open source?

BG: Open source will never overcome closed source development, but the community models and development transparency will become more prevalent. Over the past couple of years Microsoft has started blogging and having public, visible feedback systems that more closely try to follow the community and transparency of open source without actually making the code open source.

OR: Thanks for taking the time to talk with me, Brant.
The interview will be up on our site: mozillamemory.org soon.

BG: Okay. Cool.

OR: Also, you can upload other material there. If you have saved materials (emails, documents, memoirs etc.) that you'd like to add, please do so. Do you still have that history of tech evangelism that you wrote for the contest? If you would like to add that paper or anything else, you can either add it yourself or email it to me.

BG: Okay.

OR: Great. Well, thanks again. Have a great day.