

AIM IM with Johnny Stenback.

8/22/06, 6:00 PM

This interview has been edited for spelling and punctuation.

Olivia Ryan: Hi Johnny. Is now still a good time for the interview?

Johnny Stenback: Hello Olivia! Yes, now works for me.

OR: Oh great. I have a series of questions here. If you need to wrap up, just let me know. We can also pick up later if need be.

JS: Ok. I don't have anything particular going on this pm so I should be able to chat for a while.

OR: Thanks. When did you begin using computers and how did you get interested in computers?

JS: I got interested in computers at the age of maybe 12 or so... Mostly through friends that at that time used them primarily for gaming (Commodore 64 etc). I grew up in Finland, and there PC's weren't really available to your average teenager at that time, so I didn't get a computer myself until I was 16. My parents refused to buy one for gaming only, but once when I was 16 PCs were available and I guess my parents thought I was old enough to have one and actually do more than play games with it. Once I got it, I actually ended up doing more non-game related things with it. I never was a big gamer.

OR: Have you also had formal computer training?

JS: Later on in life I did, yes. Some in junior high school (or the equivalent, the school system is a bit different in Finland so the grades and "schools" don't quite line up with what's done in the US). After junior high I went on to vocational school for a computer science degree, which at the time was brand new, it had formally been an electronics engineer degree, and still was, mostly. The amount of computer classes etc. was very low.

OR: I see.

JS: By reading some books, experimenting on my own, etc more the experimenting part than the books part.

OR: When did you begin contributing to open source projects and how did you first connect to open source?

JS: I forget exactly when, but it wasn't too long after I got my first computer that I started *using* open source software and learning what it was all about. It all started with Linux for me, mostly watching from the side how the Linux kernel was developed, and chiming in occasionally where I thought I had something to add. Never really wrote any

code for the kernel, but participated to some degree in the mailing lists etc. My first real contribution to an open source project was Mozilla, when I started working on a Mozilla based project for the company I worked for in Finland, even. That was back in 98, in the summer, same year the source was released to the public by Netscape.

OR: How long did you work on Mozilla projects on a volunteer basis and in what capacities have you worked?

JS: I started that summer, partially as a volunteer on my own, and also partially through the company I worked for (CiTEC). At CiTEC I started what was to become doczilla (<http://www.doczilla.com>), which was going to be a SGML/XML browser based on the Mozilla source. Since then I contributed source code and help in fixing bugs etc, until I was hired by Netscape in the summer of 2000. Mostly in code related to what I was working on for CiTEC at the time, which was dealing with document parsing and building up the internal representation of those documents in Mozilla aka the DOM.

OR: Did anyone else at CiTEC work on Mozilla projects?

JS: Yes, there were a couple of people, most importantly Heikki Toivonen and Michael Leventhal. Heikki also joined Netscape (at the same time as me, even). And others later on as well, but I don't have names for them.

OR: How, if you know, was the relationship between CiTEC and Mozilla established?

JS: Primarily through me and Heikki. We more or less just started working on the code, and got to know people through our work, and eventually others in the project got to know us. But CiTEC never really did have any official relationships with Mozilla or Netscape, they were just another user of the source.

OR: I see.

JS: Heikki and I ended up going to the XTech conference in San Jose in 99 to present what we've been working on and also to give a tutorial on extending Mozilla (available online still IIRC). While we were in San Jose we ended up getting in touch with Eric Krock at Netscape and he organized for us to stop by to show them what we've been up to and to meet some of the people at Netscape.

So we met for lunch at the Moz cafe' at Netscape, got to talk to a handfull of the developers we'd only ever interacted with on IRC, and got to see the Netscape campus etc.

OR: What projects did you work on when you first moved to Netscape?

JS: My first position at Netscape was basically to take over the ownership of the DOM module, which at the time was owned by Vidur Apparao. Vidur was going on his

sabbatical and needed someone to stand in while he was away, and since I had already worked on that code it was a natural transition. Once Vidur got back he moved on to other projects and I remained the owner of the DOM module.

OR: Can you explain how module ownership works?

JS: Module ownership is basically a way to have either one or a couple of people who have the final say in what happens to a particular module (part of the source code). Generally it's whoever has done the most work in any given module, and that person is in on making decisions about changes to the code in question.

OR: How is the division of labor generally determined? Do people generally work on the projects they are most interested in? Is work ever *assigned*?

JS: It depends. For individual contributors they generally work on what's most interesting to them, whether that's because they themselves really want that work done or other developers that are working on related code convince the person in question to do some of the work. But on the individual contributor level work is rarely assigned per se.

For employees of companies the company can of course assign any of their employees who are working on Mozilla code as their day job to do any given task. That's then of course all up to the company etc. And code always needs to be accepted by the project as a whole too, of course.

OR: I see. How do you generally communicate with people you work with?

JS: Mostly through email and irc. And of course in person with the people that happen to be close by.

OR: Do you think any one way of communicating is better than others?

JS: It depends on what's being discussed. For very targeted discussion of a fairly narrow subject there's no faster way than to get all people face to face, or at least in on a telephone conference), but for most work that's not time critical or doesn't involve a lot of people irc or email works just fine. Our bug database is also something we use a fair bit for discussing issues related to a specific bug.

OR: If a company wants to pay an employee to work on a Mozilla project, who might they contact at Mozilla to establish a relationship? Or would they just start volunteering?

JS: Generally people start by volunteering. There's really no one person at Mozilla who's aware of every single person who's working on the source code. And in some cases people work on their own Mozilla based projects, but the project is not public in which case there's not necessarily any knowledge outside the company that someone there is doing Mozilla related work.

Volunteering is how people show their commitment. That's how people get recognized, and that's what's needed before we'll let someone get write access to our source repository as well. And of course volunteers can do lots of things that don't involve coding as well, like QA work, marketing related work etc.

OR: Why do you think people volunteer? (coders, QA, marketing, etc.)

JS: Mostly because they realize that they can make a difference. I think the biggest driving force is to be able to make a difference in lots of other peoples lives. If you help fix a crash, or help organize the bugs reported for a new feature to get developers to focus their time where it matters, etc, *millions* of Firefox users will benefit from that little bit of work. I think that's what makes it exciting for people to help out.

OR: How do developers generally resolve disagreements over code?

JS: Generally by talking over the issues in bugzilla or on irc. If that doesn't solve it then the issue might be escalated and other people, like Brendan Eich and others, can come help out and find the right answer.

OR: How much communication and coordination is there between different aspects of the project? (e.g. between those who work on the front-end and those who work on the back-end)

JS: Generally quite a lot. Most of the time people discuss what they're doing with people who know the parts of the code that their work depends on if there's any question about whether all the necessary support already exists in the code they depend on, and that way know what they're dealing with before they've done tons of work expecting something to work that hasn't been developed yet. There's not even always a 100% clear separation between all aspects of the project, like a number of the people writing front-end UI code sometimes also dive in and work on back-end problems.

OR: Getting back to the topic of volunteering for a moment...to what extent has Mozilla relied on the work of volunteers? Has that reliance changed over time? If so, how?

JS: It has most certainly changed over time. In the very beginning, it was all done by Netscape. But that's been changing ever since the code was released.

Nowadays the work done by volunteers is *huge*. I don't have numbers, but w/o the volunteer work we get from others, be it individuals or people working for other companies Mozilla wouldn't be moving anywhere near as fast as it is.

OR: How do your experiences working on closed source software products differ from working on open source?

JS: Working on open source projects involves a lot more online communication. And it involves communicating a lot more with people who aren't employed by the same

company. Over time you'll also end up working with a much broader spectrum of people, both in nationalities and abilities. We've got contributors from all over the world, some are already veterans in the software business, some are newcomers and need more help getting started. Whereas working on a closed source project you're generally working with a relatively small team, under the same management etc, so the decision making process is entirely different and the driving force behind those decisions is entirely different.

OR: Why do you think Mozilla – in particular 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?

JS: I think the primary reason Firefox has attracted so many users is that it's been designed, from the very start, to be an end-user product. Unlike the Mozilla suite (Seamonkey nowadays), Firefox was targeted from the very beginning at average internet users. One of it's beauties is that it does **not** have every feature imaginable in its UI, it's simple, clean, and elegant. That's how the whole extension mechanism came to be, Firefox was supposed to be simple and do only what it's designed to do. Everything else should be an extension that any given user can install as they find a need for different features.

OR: Do you think open source (generally speaking) is moving in the direction of focusing a bit more on average users?

JS: I think so, yes. The two projects that come to mind are Mozilla and Open Office. There's also a lot of open source projects that are not necessarily targeted at end users, and never intended to be (i.e. the Linux kernel, Apache httpd, etc, etc). And the linux desktops (Gnome, KDE), also of course target the end users and they're getting better and better as well.

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

JS: It entirely depends on the project. If the project goals are met, the project was successful. If the goals are to write a simple tool with a very targeted audience and two people end up working on it and get it done in a reasonable time frame, then that's a success, but if the goal is a large complex application intended to be used by a lot of people you'll obviously need more people involved which means you need to build a community and get contributors and so fourth. The latter part is obviously **much** harder, which is why there's not a **huge** number of successful large open source projects today.

OR: How do open source principles influence marketing techniques? Do you see marketing as something coders can do or does this require marketing experts? (This question is just in respect to Mozilla.)

JS: I'm no marketing person, but I do think good marketing requires its own set of special skills, and those skills are generally not available in your average set of coders. Now, I do think it's been shown several times over at Mozilla that marketing can be done by volunteer communities too (spreadfirefox, the NY Times ad, etc), but I do think marketing expertise makes the marketing much more successful in general.

OR: Why only Spread Firefox and not Spread Thunderbird or Spread Seamonkey, Minimo etc? Has there been any discussion (either by Mozilla employees or volunteers or users) to start spreading other projects in that way?

JS: I think there's been discussion, but there hasn't been enough momentum to get anything off the ground yet. The enormous interest we got in Firefox was what drew all those volunteers to spreadfirefox etc, and while there's significant interest in other Mozilla projects as well, the numbers are quite different in the end.

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

JS: It certainly adds to the credibility of open source. If Firefox hadn't gotten the attention that it got, far fewer people would have any knowledge or belief in open source as a good way of developing good software.

OR: I have two more questions.
Do you consider open source software projects as a public service?

JS: Yes.

OR: Why? I ask because we get a number of different responses to this question.
7:40 PM

JS: I see it as a public service since generally open source software is doing the public a service by writing software that's doing the right thing rather than generating the most money. It saves people from being locked into decisions made by companies whose primary target is to make as much money as possible, rather than to provide the best software possible.

OR: Last question: What's the future of open source?

JS: "Bright". I see more and more companies investing in it, and I see more and more contributors to the Mozilla project. I'm also hearing more and more that governments in various countries, states, or cities, are moving away from proprietary software to more open software. The more interest there is in open source, the more companies and individuals will contribute to it, I think.

OR: Thanks so much for your time!

JS: Thank you! It's been my pleasure

OR: We will be putting the interview in the archive soon <http://mozillamemory.org/>
And I'm glad we finally got to do this after so many attempts to meet in person.
Also, you can upload any of your materials directly to this site. So, any old emails or IRC chats that you think are relevant. Or photographs, documents...whatever.

JS: Yes , me too! Ok, good to know!

OR: We have completely about 27 interviews now, although only a few have been transcribed and uploaded (most were in person). We should have many more up very soon. Thanks again for the fabulous interview. Have a great night.

JS: You too!

OR: Thanks.