Ken Albers: When did you begin using computers? How did you get interested in computers?

Stephen Donner: I began using computers probably back in 1983 or '84 (sometimes around then). As my father and his coworkers were all engineers, they all had home computers, nearly all of which were some variant of Commodore. Although I was around the PET and the VIC-20s, our own first computer was the much-beloved Commodore 64, on which I mostly played learning-related games, at first.

KA: What is your education background? Have you had formal computer training?

SD:I am currently finishing up a Bachelor's Degree in English from Indiana University South Bend.

I have had formal computer training, but have taken no programming courses other than whatever Basic program we were taught back in High School ('92-'96).

From January to March of 1999, I was trained to administer Microsoft Windows NT 4.0 at Clark University's Computer Career Institute in Braintree, MA. Although I finished, and took a smattering of the certification tests from Microsoft itself, there were one or two tests I never took (one of which was MS Proxy Server 2.0, which required rote memorization of many ODBC error codes). Also, at the time I was finishing up the testing in order to become a Microsoft Certified Systems Engineer (I currently am a Microsoft Certified Professional, having passed 5 out of the 7 required tests), Microsoft began its training and testing for Windows 2000, which incorporated yet another slew of very difficult services, such as Active Directory, for which I wanted no part.

KA: What's the first programming project you remember working on?

SD: That would be Mozilla; I actually became involved in the Mozilla project back in March of 1999; I downloaded and played around with M3 for quite a while, but merely as an end-user. I'm pretty sure I had run precompiled binaries from Chris and Jason over at MozillaZine of Mozilla's Gecko rendering engine, which at that time was called NGLayout.

KA: You wrote that Mozilla was your first programming project and that you connected to it first as a user. How did you first discover Mozilla, and how were you able to engage and become part of the community?

SD: As a long-time fan of Netscape--the first version I used was Netscape Navigator 2.0, I think, in 1996 as an undergraduate--it was natural that I would want to check out the company's efforts to regain its foothold in the newly competitive browser market. I'll save the arduous recap of the so-called "browser wars," but suffice it to say I had always admired the spirit of Netscape--not to mention its great branding/marketing. Anyway, in

1998, while I was attending Clark Computer Career Institute, I became aware of Netscape's foray into open-source territory--a bold, but necessary move--and was excited that I could perhaps download/compile the code, and actually help test and refine their 5.0 product (I never liked 4.0). Jason Kersey's MozBin offered pre-compiled binaries of the earliest pre-alpha builds of Gecko/NGLayout, and I ran those just for curiosity. As the code matured, so did my interest and involvement, and I hung out often on IRC channels, where I would frequently talk to other notables Chris Nelson, Jason Kersey, Asa Dotzler, et al. (Dotzler would soon be hired, then Kersey, and I followed suit on Kersey's inside recommendation to the Netscape Mail team). While still a volunteer, I helped to triage bugs, which involves sorting, confirming, updating their information, and verifying them, among other things. Although I had submitted my resume a few times, it wasn't until Jason Kersey got hired that I began to think I might be a candidate. Even still, Jason had been running MozBin, which had later merged with Chris Nelson's MozillaZine, both of whose skills outweighed mine. Still, in November of 2000, after submitting my resume for a third time at Jason's behest, I got a brief phone screening interview, which for the large part consisted of the logistics of getting interviewed. I had an interview with the Mail/News Quality Assurance team within three days, I think, and the rest, as they say, is history.

KA: What Mozilla projects have you worked on and in what capacities have you worked?

SD: Netscape Mail/News Quality Assurance team, as a Software Quality Engineer, filing, triaging, and verifying bugs, as well as writing test cases for the NNTP (news) component, and performance (where I would end up spending the large majority of my time). I have also fixed a few front-end XUL/DTD/JavaScript bugs along the way, but QA is really my forte. During one phase of the project, I ran Rational (now IBM) Purify, and filed and verified 60 leak bugs.

I haven't done too much with Firefox these days, having spent the last three years buried under a mountain of undergraduate schoolwork, but I've filed the occasional core Gecko bug--its rendering engine--as well as a few SeaMonkey UI bugs here and there.

KA: Do you generally work alone or within groups? How is the division of labor generally determined? Who is in charge?

SD: Even though I was a Netscape employee for nearly three years, for the most part we worked individually, although when we ran Basic Functional Tests and Certification Tests, we of course worked in teams. Before my paid role there, testing Mozilla was mostly group work, consisting of hanging out in IRC channels and getting a few people on different platforms to confirm that they saw the same bug you did. Regarding division of labor, I remember that Asa was the driver--even before his employment--of "Bug Days," in which predefined lists of "UNCONFIRMED" bugs would need to have a few eyeballs look at them. It was very metric-driven; we knew all along the way how we were doing in the lists for each module (section of the code), so we would load-balance our efforts to help out the swamped modules. I feel it worked surprisingly well, for such

a rag-tag and disparate group of people, from literally all over the world, coming together solely over an electronic medium for one common goal: improve their future (or current) browser.

KA: You talked about a "disparate group of people, from literally all over the world" working together >on Mozilla. How did you generally communicate with these people? If you used different means, what >did you generally use each of these methods of communication for?

SD: I should've elaborated on this earlier. The majority of my job was going through the NEW/UNCONFIRMED pile of bugs contributed from outside sources (i.e. non-Netscape-paid employees). Without the sheer amount and, frankly, the variety of testing platforms/scenarios, the Netscape, and later Mozilla, browsers, wouldn't be nearly as good; here, I'd like reiterate that, for applications as large and complex as Firefox and Thunderbird, outside contributions (whether that be code contributions or good bug reports), are absolutely essential to its continued growth and survival. I'd say the majority of the communication took place directly in the bugs, as comments. A few times, for sensitive information, the reporters and I would email each other directly, and for the more technically minded, some would also appear on #mozilla on irc.mozilla.org. Still, as I've mentioned, Mozilla's bug database, Bugzilla, was the primary venue for ongoing communication.

KA: How important are comments in the code to the smooth development of Mozilla software? Can you give an example? Are comments ever misused?

SD: Sorry, as a non-developer (the XUL/JS/CSS fixes I've mentioned I've done are indeed so minor as to induce an embarrassed hue), I can't rightly answer this question. Some of the more vocal developers--at least those on IRC--complain about the lack of good code comments; I'd hope that as the product has matured, so too have its API/code comments, and/or documentation (on the Wiki or on Mozilla Developer Central). As a QA engineer, though, I really don't get down to this level, sorry.

KA: How would you describe your programming style?

As I truly have none, I just say the following: a lot of my code changes were mere string changes (literally search/replace operations within the code), but for the few JavaScript changes that I made, I just followed the pre-existing API and passed in the right arguments. Simple stuff; I can't do anything complex.

KA: You wrote a little about methods of communication. Do you think that one mode of communicating should be used for managing projects? What method do you think works best?

SD: For me personally, I think that email works best, as you have a fast, very efficient, and handy record of communication, especially when/if coupled with an archive, as in a mailing list.

That said, I still heavily use IRC, because it's even faster, but not everyone chooses to use IRC; you can't (or shouldn't, rather) avoid responding to emails. There's a lot of movement towards Wikis, and I think that's great, since they are so flexible and make it easy for anyone to quickly correct and add items. I used them in a college web-writing course briefly, but not yet in the real world. I think they hold a lot of promise too, and, as with mailing lists and newsgroups, they leave a nice revision history to follow.

KA: Have you noticed any tension between those who work on the front end and those who work on the back end?

SD: Not personally, no.

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

SD: Because I'm just coming out of finishing a three-year stint at college, I've been less involved with Mozilla than I'd like, so I'm not sure exactly what percentage of patches, proposals, bug reports, and testing coverage emanates from the community at large. However, I'm positive that it is and will continue to be a vital part of Mozilla's success; it certainly was crucial for the earlier Mozilla-derived Netscape-based browser suite's testing coverage, new-feature submissions, and the like. There are use-cases and system/application configurations that exist in the wild that simply can't be accommodated for in most labs (this is true with nearly all software companies). As part of this realization, I think Mozilla does a good job of remaining transparent about its needs and goals, however much others might disagree. It's also willing to adapt its focuses to the market--Firefox and Thunderbird were started by volunteers--and both have undoubtedly helped shape Mozilla's impressive and continued success in the browser and email client markets.

KA: Why did you volunteer? Why do you think other people volunteer?

SD: I volunteered because it was exciting to think--and as time progressed, know--that my involvement would have a direct effect on the quality and feature set of Gecko/Mozilla-derived products (when I started I was chiefly interested in how that would transpire in the Netscape-branded builds).

KA: 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?

SD: I think it's well-known that many of Microsoft's competitors--many of which might now be seen as "underdogs," but which in their day enjoyed a large market share--are championed largely because they don't come from a behemoth (which Microsoft is seen as). However, I think the case is different with Firefox; by this, I mean that the product itself is worthy of its admirable market share (15%, last I checked), because it's organic software, truly driven out of the needs and input of its users. Its Add-ons/Extensions and

Software Update systems are especially noteworthy (Netscape's 6.x-7.x product line relied on complicated XPIs which were nearly impossible at times to uninstall/configure, for instance.) The Add-ons/Extensions systems are flexible/extensible enough to empower users to refine their browsing experience not only for themselves, but also for others, while maintaining a minimal yet sufficient experience for the typical user, out-of-the-box, as it were.

Over the years, I've seen a number of very talented and motivated hackers literally step on the scene and rewrite some code, implement some feature, make some great suggestions, and so forth. Disclaimer: I'm not privy to other open-source projects, so I can't comment on the dynamics of community involvement there. However, having been involved in the Mozilla community since its inception in 1998, I can attest that if you put forth the effort in the right places and are willing to learn how to navigate it as a project (this was an early concern, but it's getting better all the time, from what I hear), it is a rewarding one with which to be involved. Firefox and Thunderbird are two products any given end-user is likely to use daily (especially the former, which brings me to my next point).

Certainly, the Firefox project is the most visible--users with webmail-only accounts will likely only have need for it--but Thunderbird and Sunbird are both rapidly establishing themselves in their respective spheres as well; Thunderbird now has built-in hooks to allow users to quickly set up their Gmail and other email accounts, as well as a nice RSS reader. Yet, what also makes Firefox unique is its role in monetizing search traffic, something which is very important to the continued funding of the Mozilla Corporation (hence the setting up of the Mozilla Foundation; I'll leave those more-complicated details to the actual press releases and the respective FAQs/blog posts). I'm positive that as long as individuals and corporations care about having an open-source browser that can be organically developed to keep up with the changing web, Mozilla will be around in some capacity (funded/incorporated or not).

KA: How would you list Mozilla's priorities today? How does this compare with its priorities in 1998?

SD: I see Mozilla really listening to the community at large, soliciting ideas and feedback on some prototypes; "The Coop," which makes an attempt at bringing friends together via the browser in various ways, is one such example. The priorities are really the same as they've always been, on the larger scale: to deliver choice and innovation on the web. Having known many of the key Mozilla folks for a while, I know that to be a genuine concern, not just some nice-sounding mission statement. Every year, I continue to be amazed at the amount and quality of work that goes into Mozilla products. From 1998 to around early 2000, there was a slew of changes to nearly every component, and so there was a lot of focus on performance, stability, and implementing features (not necessarily in that order, but I digress). What I see now is all of the above, but since Firefox's 1.0 release in November of 2004, there has been a continued focus on refinement, and also

on not regressing any of its features, nor introducing performance hits or instability (hangs, crashes).

I think there will always be challenges, specifically in balancing users' requests/feature implementations, and such; like any project or product, it can always improve, but I feel that Mozilla has done remarkably well in this area. I hope always to contribute to it in at least small, yet meaningful ways, and I strongly encourage others to do likewise.

KA: You mentioned you became involved with Netscape/Mozilla because you realized your work would have a "direct effect on the quality and feature set of Gecko/Mozilla-derived products." Did you view this as more of a personal concern, i.e. being able to scratch an itch, or as part of an effort to reach a wider audience?

SD: For me, it was just a more productive past-time than what I could've been doing, and it came at an especially exciting time in the Mozilla project itself. To be perfectly honest, it was a great way to get close--as close as one can get via the internet--to the well-respected (and yet beleaguered at this point) group of Netscape developers, and, soon thereafter, a flood of likewise-minded, talented individuals who all wanted to contribute in some way. So, to answer your question more directly, it was mostly personal, though I certainly had the goal of helping out in whatever fashion I could.

KA: You wrote a little about how you perceive Mozilla's priorities. What is your vision for Mozilla moving forward?

SD: Simply put, I would hope that it continues much as it does now: always focused on what its users need and want, balanced with enough insight to help charge the forward-progress of web technologies in general. I really can't put it more succinctly than that.

KA: 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?

SD: Is this a question for me? (It's below your signature, so I'm not sure.) If so, that's a little tricky for me to answer. At the genesis of Spread Firefox, I was heavily entrenched in the academic world, and didn't focus much of my limited free time on Mozilla, other than filing bugs that directly affected my own end-user experience. I see open-source development and marketing models as working in concert. The marketing model, as in Spread Firefox, invites its users to try the software, recommend it to others, but more importantly, empowers the users to contribute back and improve the software in whatever capacity they are able. I know that word "empower" seems trite and overused on the web, but as that is what got me and others directly involved with the project, I think it's fitting.

KA: Have you contributed (as a volunteer or employee) to any non-Mozilla open source projects? If so, to which ones and in what capacity did you participate? How did those experiences compare to your work experiences at Mozilla?

SD: Sorry, I haven't.

KA: Similarly, do you have experiences working on commercial products? How did those experiences differ from working on open source?

SD: I've worked on both the Mozilla-derived Netscape browser, as well as the AOL client and AOL webservices proper. In the former, community input was vital to the quality of the product; in the latter, although we solicited feedback, it wasn't done in real-time, along with our development, and therefore there were a few instances in which our testing didn't match the end-user's experience. Having the community being involved in an open-source project is great, but the sheer amount of information to sift through can be daunting; it still remains the best way to get the best testing coverage and user feedback, due to that real-time factor (daily builds).

KA: 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?

SD: For me, open-source projects are more than simply opening the source code of an already successful product or idea and then sitting back and "letting things take shape." It requires drive, coordination, constant awareness of shifting focus among the community, and above all, being transparent and over-communicating (if such a notion exists at all). The ways in which community members can become and stay involved should be clear; so too should the ever-evolving needs, both long and short, of the project, as well as clear milestones. Additionally, the project needs a way to recognize its contributors for all of their contributions, no matter how seemingly small. And finally, above all it should be organic, responding directly to the desires and needs of the community which helps to shape it; at the same time it needs to represent the majority of its users' needs, it needs to not forget, where appropriate, to be accommodating of smaller subsets, too.

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

SD: I think the co-op model (or credit union) is perhaps a bit closer than straight public service; in order to provide services (in this case, software) someone has to do the work. Although a high volume of feedback flows into some of the higher profile projects (such as Mozilla), in order for that to translate into the product, there must be a lot of planning, coordination, coding, and testing. Bottom line: if it's a vibrant community, its needs will be met at some point / measure by someone. If it's not a paid employee with a reason to implement feature "x" or fix "z"--or test "y," the burden--though it's not really a burden in the strictest sense of the word--lies with whomever steps up to do that.

KA: What (if anything) do you think the popularity of Firefox will do for the open source software movement as a whole? Do you think open source techniques can be applied to other areas of production in today's society?

SD: My hope is that Firefox--and open-source projects in general--continue to gain the respect and visibility that they deserve. I think Firefox's success--quite a bit more

widespread in its adoption than were Netscape and "vanilla" Mozilla builds at the time--will give the software world a model by which open-source software can and does work (OpenOffice, likewise). To answer the second question, I think it's fair to point to MIT and Harvard's open lectures as perhaps an open-source inspired idea: for the greater good of the community, sharing technologies and materials and soliciting feedback is a good way of helping to ensure that what you put out there is valuable.

KA: What's the future of open source?

SD: I have a rather limited purview, but it's rather obvious that the trend is to open-source more projects/products--where it makes sense to. Not all open-source projects are successful, but it's a bit hard to gauge that up front. I think SourceForge and Google Code (and others) are great venues to experiment with putting something out there and seeing where it goes. Open source can be contentious--as can all projects with multiple vested interests--but if done well and adopted by its community--can be driven to create a product/service of real value to a multitude of users.

For the complete version of this email exchange please download the file StephenDonnerFull.doc above