

Goodger: I work at Google contributing to the Mozilla project, primarily to Firefox and I continue to do the sort of the work that I've done for the past few years which is to play an active role in the development of new versions of Firefox from the early stages of each release cycle where we figure out what we want out of the next release through the planning, documentation and development, bug fixing and release, so a whole bunch of different things.

OR: How did you come to work at Google for Mozilla?

Goodger: Towards the end of 2004, we were shipping Firefox 1.0 and it was around that time that I had come in contact with several of the engineers here at Google and just was generally impressed by the enthusiasm of the people here, even over things that I thought superficially were boring tasks. They seem to make everything fun and it wasn't long before in the process of working with them on these things that they sort of became fun for me, too, and I thought that was sort of infectious and I thought maybe after working with— working closely with the same set of people for the previous number of years which is about four years, maybe slightly more at the time, that if I were to come and work at Google I could potentially still work with those people but then also be exposed to a whole bunch of new people with different ideas and enthusiasm, so that was one of the reasons why I chose to come here.

OR: So how does that work? They pay you to work on Mozilla products. How did that come about?

Goodger: Towards the end of 2004 I sent a note to somebody I knew here and saying that I was interested in anything that they might have and it turned out that Google was interested in Firefox. They liked the product and they thought it would be good to support its development, so eventually they hired myself and several other people from the Mozilla community to continue development on it.

OR: When did you first start working with computers?

Goodger: Not until I was older than most people around here. I started— I think I got my first computer when I was about 12 or 13 years old. It was an Apple 2. It was sort of a hand-me-down from my relatives. I used that for a while until it broke and then I had a— I think when I was about 14 I got a 486 and that basically didn't do anything interesting at all. It played a couple of games and ran Windows 3.1 and it was not anything like stuff that we have these days or what even necessarily what my friends had. So I just made the best of what I had with that and played around teaching myself to program with Excel Scripts and stuff like which was all that was on the machine, so that was about 13 to 14 was about the age that I got into this.

OR: What's the first programming project you worked on?

Goodger: I got into programming from an interesting angle although one that's probably becoming more common these days in that I was interested in having my own—

I wanted to make my own website, so I began to write HTML because it seemed easy enough to figure out and then just reading some computer magazines I found tips about JavaScript which is more of a programming language than HTML. Began to make my website more elaborate and interesting, whizzy effects and that sort of stuff. Then over time that sort of grew and the website became bigger and bigger until I sort of switched over to contributing to Mozilla.

OR: Which Mozilla projects have you worked on?

Goodger: I got involved with Mozilla because I was so curious as to what had happened with Netscape. Netscape had announced that it had released its source code to the public in 1998 and I'd sort of looked at it at the time, but didn't think there was any way I could contribute to it because it was too difficult to work on given my current set of skills. So I came back to the project to see what was going on about year later in 1999 and saw that they had switched gears a little bit and were now working on the new version of the application that was written in languages much more familiar to what developers in sort of XML and JavaScript type things for their user interface so I began contributing to that, trying to— because they basically rewrote the entire application from the ground up and so most of the user interface was either buggy or sort of in poor shape or not there and so I figured I could probably help with this given my experience with web development. So I sort of got involved with that and figuring out what the— helping to figure out what some of the syntactic elements of the XUL language were and so on. And it was from that work that I got a job offer from Netscape and I went to work for Netscape in the browser group and worked on a number of different application features there for a long while, then a bunch of different internal projects after Mozilla sort of began to lose favor within that company and then after that onto Firefox.

OR: So it sounds like you've sort of had a variety of roles within different projects. Is that right?

Goodger: It's being mostly worked on the browser for the past six or so years but I have worked on different things within the browser, different browser functionality pieces and on ultimately different browsers. Back in 1999 through 2002 or so, I was working what's now called Sea Monkey or what shipped as Netscape 6 and 7 before eventually switching gears to work on Firefox and then various other things along the way.

OR: Do you generally work alone or within small groups or within larger groups?

Goodger: It depends. If I'm working on a sort of a fairly self-contained feature, then I can probably do it myself, although there is a testing community that as you make changes or report bugs back, all that sort of stuff, but for more ambitious features, it's usually a small group of people, maybe three or five at most for certain things. Right now, I'm working on features with about I guess four or five other people since it's significant enough in complexity.

TS: Are those people usually located locally or remotely?

Goodger: It depends. The one feature I mentioned where I'm working with a large number of people, those people are all here and people paid by Google to work on Firefox and so that's one thing. But for some of the other features it's a matter of, for instance, if I'm working on a feature that's part of the user interface but the user interface requires some feature from some other part of the code, then I will go and find that person and try and figure out a way to get that thing done. And that person who maintains that other section of the code may work— may be a student in some other country or something like that, so it's a matter of using the project's communication infrastructure to locate the right person.

OR: Who's usually in charge when there's so many different people working on one thing?

Goodger: With respect to Firefox and the Firefox sort of release process, there are a set of people that sort of work together to try and make decisions and organize things. Like at the start of each release we sort of sit down and figure out what we want out of it and then people sign up for tasks and take on the role of making sure that what they signed up for gets done and if that means that they need to work with three different people to get that done, then they need to drive that. Aside from that, just sort the practice, the operational side, Mozilla works on sort of what they call a benevolent dictator model in open source which means that each little subsection of code is called a module and has an owner and peers and basically the owner and peers are the set of people that do— coordinate the work and get it done with the owner basically making decisions where consensus cannot be reached and all that sort of stuff. So within the context of the Firefox browser and user interface, I am the module owner.

OR: How do you generally communicate with people you work with?

Goodger: Right now it depends on the project. Like, for instance, if someone is far away, it would be through IRC or e-mail. People are nearby, it'll just be a hallway chat, but what I've been trying to do lately and my most recent communications have sort of been of this sort is to try and get more discussion into the news groups and mailing lists so that we have sort of an archival record of all of the decisions that have been made because it can be difficult if a year from now we're trying to figure out why you did something and we might forget the discussion that we had, so that's sort of where I'd like to see things head. We've actually been poking things along in the past month or so.

OR: Is it generally working, you think? Or do you think it will?

Goodger: I think it'll take some time because one of the things about open source projects, especially when you have an open source project that's started by a company like Netscape, there're different worlds and when you are a commercial entity, you get a bunch of people who're used to working in a closed environment on a particular campus in the same building and on the same floor. The convenience of being able to walk over

and talk to somebody about something is very— It should not be understated how powerful that is and, so, as a result, from the very start of the project when Netscape was significantly involved, it was difficult to get people to sort of more fully engage with the community. In fact, that culture has sort of followed the project and I know that myself there've been times when I have done that sort of thing and it just makes it very difficult later on to figure out what's going on.

OR: Where do the code contributors work on the code?

Goodger: How do you mean where do they—

OR: Where— I don't know...

TS: Well, I think what we're getting at is sort of what you were saying about sort of maybe expand a little bit upon the ownership and how it's determined who's going to be on which teams, on which modules. Is it strictly voluntary? How is it decided who's going to be the owner of the module? How is it decided who's going to be up here and what is the working relationship like?

Goodger: I think people— as far as people joining projects, people join projects that they're interested in and they'll help out where they think they have skill or have something to contribute. Sometimes if people work for a company and that company's interested in contributing to Mozilla, that company may have interest in a particular product or a feature and thus will pay its employees to work on that particular task. Someone may pay someone to work on Thunderbird but not Firefox, for instance, but as far as the actual sort of the leadership, what generally happens is when there is a case where there are sections of code that are not clearly owned or not being well owned, the sort of organizational structure that sits above all of the Mozilla projects, like Firefox, Thunderbird, etc., the Mozilla Organization/Foundation I guess at this point, will take nominations for who the new owner is to be and there's sometimes discussion and people will agree or disagree, etc. But usually it's the case that there was a section of code and it's been unowned for some time or somebody else— It's been unowned and somebody else has stepped up and began acting like the owner and basically been reviewing patches and producing new code and keeping that code alive and so someone will say, well, why don't we just make this person the owner and usually the case is, well, yeah, looks like that the person's the owner anyway, so we'll just make it so and so that usually happens.

TS: Maybe you can think of a specific example of a module where that's kind of played out?

Goodger: Well, most recently I think an example has been the Java to XPCOM Bridge which is a thing that allows code written in Java to talk through XPCOM to other Mozilla components and services. And there was some old code in the Mozilla source tree called Blackwood which I don't believe was being— or at least wasn't as up to date as it could be and I think somebody from IBM, Javier Pedemonte, had written a new one and he was actively developing that and so people thought that he should be owner of this

thing and so there was a brief discussion and it was made so. They have a tool for tracking who— what all of the modules are, what code those modules contain and who the owner appears and members are and that system is called DeskBot, so somebody and went and made that change in that system and now he's the owner.

TS: Who manages the DeskBot System?

Goodger: I'm not sure. I think a couple of people at Mozilla. Brendan, Mike Shaver, those are people that know how to add and edit people on that system. I mean, individual owners can— If you're the owner of a module, you can manage your own module. You can add peers, people from the community who've been volunteering a lot and making significant contributions. You can make them peers and stuff like, but as far as sort of the meta, these are people at Mozilla.

OR: How important are the comments that are within the code? Do you use those as a way of communicating instead of, I don't know, any other means or—

Goodger: For a long time— One just sort of personal editorial, I don't think Mozilla code is commented enough and I've sort of contributed to that as well over the years in trying to improve it, just from having to go back in later years and re— figure out what's going on, but it was never a tool for documentation on the website that was easy enough to use that developers would use it as they wrote code. So typically the code is sort of, you know, where there are comments, those comments are documentation for the code. If someone went the extra mile and wrote actually web page documentation for how things worked, that's sort of a— within Mozilla at this stage, at least, that's sort of an unusual thing. So a lot of the comments are to either themselves, people to themselves, or to other contributors as to how this stuff works.

OR: Do people ever go back and make comments later?

Goodger: Yeah, I mean, sometimes you'll find people who— you'll look through a file and you'll see some code that's very old that might date from when the project, when this version of the application was still getting started back in '99 for instance and very very old code that's maybe a little messy or not well documented and there'll be a comment later on from somebody saying what does this do or what's this stuff, maybe in the hopes someone will come along – someone else who knows – might come along and either document it better or improve it. So, yeah, sometimes there're notes like that. Or just sort of often editorial comments like this is a really bad way of doing this, this sucks or something like that, as a mental note to the person that maintains that file that next time they come through they should consider optimizing or something like that.

OR: Can you think of a time where that ever led to a significant change if a comment like that—

Goodger: I'm not sure. I mean, usually they're over relatively small things because a lot of the major changes or re-architectures and that sort of stuff, they're being tracked

outside of the code. But back in— after Netscape 6, Netscape 6 shipped and was very slow and buggy and there was a significant effort after that within Netscape developer team to improve the performance, for instance. So a lot of those optimization things or at least the really massive ones with huge impact were identified and filed through the bug system and then tackled that way.

TS: Are comments ever misused in any way, however you want to define misuse?

Goodger: People will sometimes write bad comments. Say you have some code that `I + 1`. There might be a comment that says `increment I` which is not really that helpful because the code clearly says that and generally saying it's— not all comments say what the code is supposed to do. Sometimes they say what it does do, which is in that particular example, that's what it does do. It doesn't really sort of give you an overview of what the intent is and the intent is usually what's more important when you're trying to find bugs because you're trying to figure out what it should be doing and comparing that with the actual code and sometimes you get so deep into the set of code that it's difficult to figure out what the intent was. The idea is to some extent that code be self-documenting and that people choose good names for variables and functions so that you can read through the code and assuming you have a basic knowledge of the vocabulary of the system, that you could figure it out but that's not always the case and there isn't always sort of that overview documentation that establishes that vocabulary.

OR: Just generally, how would you describe your programming style?

Goodger: What do you mean by style?

OR: I guess— I don't know. However you think you would—

TS: Maybe a better way to put it is have you ever clashed with other programmers over a way of working or not so much a particular aspect of the code, but a kind of work flow or a work style?

Goodger: Over a period of time, certainly there've been incidences. People have— everybody has different expectations and different ways of working and sometimes it's easy, especially when a lot of communication is done electronically, for subtlety or humor or anything like that, senses of humor to be lost in transmission. So if you say something in a way that's incompatible with the way something else works or with the way that they themselves (inaudible), they might take offense or there's that and there's just legitimate things where either I've been wrong or I've learned something from an experience or the other person's been wrong and is learning something from the experience, so there're probably countless examples over the years. This is pretty much the only project I've worked on while being a software engineer.

TS: How are conflicts between different people, how are they resolved? Between peers, let's say.

Goodger: Typically, people try to— There can be argument, but ultimately at the end of the day, it either sort of fizzles and/or people need to— people sometimes step back and take a look at it from another angle or somebody else who— If people are arguing on a public list or an IRC channel with other people, someone else might come along and offer a different perspective which either causes them to join the argument or for the other two people to sort of stop and think about it for a moment and either continue arguing or stop. And then people either form a difference of opinion or they resolve or come to a conclusion, so that's— I think those are some of the things that I've noticed.

OR: Do you generally work on assembly language portions of the code or in JavaScript and XML?

Goodger: Mostly JavaScript and XML although some level of C++ because there's a very small amount of code implemented in assembly language which just handles connecting functions in JavaScript to C++, but most of the back end rendering and all that sort of stuff is done in C++. Most user interface is in JavaScript, so sometimes working on a user interface feature I might need a feature in the back end, so I may have done it myself or somebody else may do it, but since most of my work is in user interface, it's mostly JavaScript XML

TS: Are those boundaries between sort of the back end and the front end, the C++ and the JavaScript XML, are those hard and fast? You mentioned that you kind of cross into the C++ things if you need to do something. Are there sort of boundary conflicts, territorial—

Goodger: I don't think so much territorial. I think there are people— This is what I've learned over the years through at times being very isolationist or working on specific features and then needing something and going and producing a patch. I don't think people are territorial. I think people like to know— People who work in other parts of the code. Actually, this is true for everybody. People like to know when you would like to make a change to their code even before you start writing it, so that they can think about it and then, well, (1) is this a good idea or not, (2) what might some good ways of doing this be, so that they can help you come up with the best idea, the best way of pursuing it, proceeding and so that they're ready for review. When your patch comes along, they can review it and sort of having the sense of knowledge of how this works rather than being surprised by (1) a completely new feature that they've never considered before with an unknown design that they then have to not only figure out just from your set of changes, but also give technical commentary on. So people like to be told first. Otherwise, people can be sort of irritated and I appreciate that because people have produced stuff like that and thrown it over the wall to me to review so I've been sort of irritated, so I think that's sort of the level of conflict you get across different module boundaries.

TS: And do you see that there's any kind of cultural differences between people who work on the rendering engine and people who work on the user interface? Is there a

differenced in the kinds of people who get involved in one or the other or in other aspects of the project?

Goodger: I think that there are definitely cultural difference and variances across Mozilla contributors. I don't know if it necessarily maps to front or back end. Certainly with respect to a number of developers on Firefox over the years, the sort of ideology of those folks including myself, has been very practical results oriented in terms of getting things done really quickly and whereas other folks may take a more relaxed, laid back attitude to it. Like I'm working on the exchanges and I'm going to get them done on whatever time— however long it takes and if they don't make this next release then that doesn't really bother me, so I'll just wait for the next one. Whereas the case for Firefox, it's always been— we've been trying to figure out exactly how much we can cram into each release.

That sort of stuff may also break down based on how much time we spend on a project. If you're just volunteering a little of your spare time, then you're not going to necessarily be able to get something into a certain release so you may sort of accept that up front and not stress out too much about it. Whereas if you're working full time on something you may be more inclined to do so. That's one thing and just to add more detail to that, I think some people are more product, or—that's one word—or another word is result-oriented than other folk who are just doing what they do and what they get out of it is sort of the intellectual challenge and stimulation from actually writing code or figuring out the answer to an interesting problem. And other people are trying to say how does this affect the bottom line, to use a corporate term, but how does improve usability or how does this make— how does this improve our standards compliance or any number of different things.

TS: Which camp would you put yourself into?

Goodger: I tend to just by necessity have to be more focused on things I can try and get done right now versus what things I would like to do. And sometimes I feel a little sad about that because a number of years ago when I was more of a free agent before I started working for Netscape I could go and pick any project that I wanted to do even if it wasn't necessarily that important and I could pay some attention to that and explore or research. And sort of the trade-off when you have people who are all focused on what's going happen next and what the highest priority thing is, is that some of the small details are lost or the interesting projects, little side things that don't seem important, but maybe they might seem important later on. You know, so those things are lost. So I think it's good that there is this variance because there're people looking at all angles.

TS: Just to sort of finish up on this line, what kind of most interests you about your user interface work? Why that? Why not something else?

Goodger: Well, I mean I certainly enjoy working in other parts of the code because a lot of sections of code are easier to test than user interface. If you're working on something that just takes a set of data and produces another set of data, you can write unit

tests for it and be very confident that it works well. User interface is harder in that respect, but also I tend to be very— I like building things that I can see and making things that look nice. And by nice— I think I'm very fussy over little details, even down to like a pixel on the screen where a line is— if it's slightly to the left or slightly to the right, where does it look right and creating balance and that sort of stuff. So I like to be able to— getting sort of the instant gratification, too, that we get when you put something together that you can see is what I think interests me about doing user interface. And then on top of that, it's like I said, the fussiness and attention to detail in making it work better and more reliably and seem more complete or high quality than previous versions or other products.

TS: So to sort of change gears a little bit, to what extent do you think Mozilla and Firefox, in particular maybe, really relies on the work of volunteers?

Goodger: I think ultimately completely because, I mean, certainly companies supply developers. There're several people here that work on Firefox and the Mozilla Corporation itself as a corporation that pays a bunch of people to contribute to a bunch of different Mozilla projects but I think a lot of the spirit or— I guess that's a good word for it—comes from the fact that Mozilla projects are community developed and I think that that will be— As soon as Mozilla software just becomes another corporate thing, I think some of the advantage that some people perceive from it might be lost. I think people see Mozilla as sort of an independent organization without as many of its— without sinister motivations or anything like that, someone who's sort of looking out for you with nothing to really gain from it but your trust, so I think the fact— the community development plays into that.

Also the community— A lot of developers from different parts of the world contribute. I started as a volunteer. A lot of people who work on Mozilla— who are paid to work on Mozilla started as volunteers. You talked to Brian. He started as a volunteer implementing Mouse Wheel support for the browser. So engineering contributions and then there are hundreds and thousands of people who download every really crazy buggy nightly build in between releases and tests and subject their bookmarks to the abuse that we're putting them through right now and all that sort of stuff and report bugs. And when bugs are filed, there're people that have the entire bug system it seems, all 300,000 bugs in their heads. Someone files a bug, they can say, oh, that's a duplicate of that one and just close it out and keep the system in check so that developers can get work done in terms of— and be able to work effectively and managers can see useful bug lists and not ones that are thousands long so that QA effort is mostly community. As is localization. Firefox has many localizations and those are— The only one that's sort of developed by the developers who work at various companies is the English one and then all of the other languages are (inaudible) by independent volunteer teams that at a certain point in the release cycle give them the go-ahead to start translating all of the strings and then they produce the localization and that becomes the official one for the language.

So, there's lots of different places and I think without a lot of those different things, I don't think Firefox would necessarily have the reach or quality or any of that sort of stuff.

I think we'd probably— Without the people who help in testing bug management side, the engineers would probably have all pulled their hair out and gone on to something simpler, so—

OR: Why do you think people volunteer?

Goodger: Well, it's just interesting. The reach is such that it's an interesting project to work on because you have the ability to help shape the future of a product that lots of people use. And Mozilla software has lots of users, it's not just Firefox. Thunderbird is probably a couple of million users and the same with Camino on Mac and these products are well received and reviewed in the press and that sort of stuff, so it's fun and cool to work on that sort of thing. It's also software that we all use every day so sometimes I thought, you know, maybe it would be more interesting to work on something else. I mean, certainly Google has lots of interesting projects. Google has products that are fascinating like maps and stuff like that but I think about it and I think this is software that we all use every day. It doesn't really— It feels like one of the most important things that you could contribute to.

The other thing is I think it's just web browsers are really complex problems and some people like to work on really complex problems and there're so many different aspects to them, not just the user interface, I mean, all of the rendering. Making sure that you are both standards compliant and acceptably fast. There's lots of tough problems there and interesting research angles and lots of stuff. So I think people contribute for different reasons, but those are some of them I can think of.

OR: Do you ever find that people who contribute have different reasons for contributing, for volunteering— kind of have different goals and perhaps clash over the reasons why— If they have a different vision because of why they're doing it.

Goodger: Certainly. I mean, if someone is— If someone's goal is to create the most standards compliant system, then implementing something that is not a standard might seem to be outside their goals or might even seem bad to them. One example to look at for this sort of thing is XForms. XForms is a specification that defines how web forms should work in sort of an XML like way rather than as they exist in HTML right now. And it's supposedly a— it's supposed to be a companion specification to XHTML which is an XML version of HTML but XHTML has no built-in forms—I don't think—it was designed to rely on XForms. And so there're people who think that that should be implemented and that should be the way to go and then there're other people who say, well, standards compliance is important but we may not necessarily need to implement every standard or we may choose to do something that's more practical so that existing web page developers don't need to rewrite their entire site using this new technology, but can make a few small changes to get the same effect. And you know there can be heated discussions between the two people, the standards, the XForm side might say, well, by doing this, you're effectively undermining our technology and people say, well, no one's going to adopt your technology and these arguments form and eventually someone— Either they'll work it out themselves or if it's too polarized, there're people— like I said,

the decision-making structure within Mozilla will come out and say, okay, well, considering all of this, we say this one.

OR: So I understand that during the early development of Firefox, CVS access was limited to a small team of workers. Why was the access restricted in that way?

Goodger: I think that the reason for that was that— And I think just, my personal feeling is that probably wasn't the best thing to do in retrospect, but it was sort of a response to the way the SeaMonkey product had developed over time and that was that SeaMonkey itself is a collection of different pieces. There's the browser and mail, etc., but also within the single piece the browser, there were different parts like sidebar and bookmarks and all that sort of stuff. And I think each of these things were separate modules and after Netscape sort of disengaged from the development process, it was never clear who managed any of that and how it would be managed. And as a result, it just became subject to sort of incremental change but without really any combined direction and so lots of small changes went in and the browser bloated a little over time and it wasn't particularly good with Netscape because my personal opinion is that the Netscape UI was pretty bad for 6 and 7. It was based very much on Netscape 4 which was already kind of bloated.

It had a bunch more stuff thrown in to appease the requirements of Netscape.com and the rest of it was kind of overwrought because within Netscape, with a hundred or so engineers, an engineer might be assigned to work on a small part of the world and that would be the only thing that they would do and so they would over-engineer some— All these things sort of grew as massively over-engineered pieces. And so then that was sort of left and rather than considering this sort of mass of not very efficient user interface, people have sort of set about continuing to work on it and add to it and all that sort of stuff, and it was very chaotic because the only— because there was no clear either sort of product requirements or product plan or any sort of vision for what that should look like.

The standard rules for checking in code applied which mean that anyone who had review and a second review could check in, so things were happening all over the place and it was growing and it was crazy. And so the idea was to create a new place in this CVS repository that prevented this from happening by restricting the check-in lists, although in retrospect, the very active forking as it was, the user interface, was probably enough to prevent people from checking in and establishing rules for who should review code and all that sort of stuff, so I think what the— putting that restriction on there probably did— was go a little bit too far and effectively say we don't want you which is not really the case, but—

OR: So how was it decided early on who would participate?

Goodger: Basically, it was just a bunch of people on IRC got together and were talking about it. I know that a number of us had expressed dissatisfaction with the Mozilla product and Netscape products and were thinking about what could be done better and Dave Hyatt had started what is now called Camino which was a separate browser and

designed this to be a nice, streamlined user interface. And so he thought it would be a good idea to do the same thing in XUL which is the XML language—for Windows and Linux, later it turned out it would work on Mac too but— so there was a sort of a discussion and just the set of people that that participated in that discussion joined the initial project and I think Brendan was in that IRC channel and so he just went about setting up a module and DeskBot and all that sort of stuff and then we were able to check in code.

OR: So how did access restrictions change over time to become more open?

Goodger: Later on, I guess around— Maybe it was late 2003 or early 2004, it was clear that the system— Well, one, it was causing to people to be sort of— It caused— The very fact there was this access restriction was causing there to be division within the community. It was like, what makes these people think that they're better than anybody else? And then also the fact that changes would happen in the back end that would lead to changes being required by the front end, like an API would change for a user interface or something like that and so the people that made the change in the back end were very diligent about making sure the SeaMonkey front end was updated because they could, because they could have access to the code, whereas with Firefox they couldn't, so changes would get made and Firefox would break, so it just wasn't an efficient way of working. When people make API changes, people need to be able to patch across the entire code base and so with these things in mind, the restrictions were lifted and just some—

What should've been done in the first place which is basic review rules were put into effect which basically stated if you want to change this code, you need to get review from either a module owner up here and if you're just making small changes like if you're fixing a spelling mistake or if you're changing an API or something like that in hundred places across the entire source string, just to review for the entire change we don't need to necessarily look at it then because you're not doing anything that specific to Firefox, you're just renaming a function foo, a function bar, so as long as someone reviews the change, it doesn't matter. So that sort of covered all of the bases really and made it seem less exclusive, exclusionary, but I think still to an extent some of that followed the project until today.

OR: Oh, really. In what ways?

Goodger: I just think people have a negative opinion of the way Firefox has developed and people who work on SeaMonkey, for instance— SeaMonkey's a very inclusive process in terms of allowing people to contribute changes and that sort of stuff and you can read some of their documentation. They had a Reasons page that talks about some of the problems that they see with Firefox development and I think that some of the things that they said are correct and I think we should all try and fix them and some of the other ones we disagree, so there you go.

TS: How about other Mozilla projects? What's your impression of those in those same terms? Where in the spectrum do those lie in between the sort of SeaMonkey and Firefox models?

Goodger: I think, for instance, Camino is a very collaborative community process for that one, but there is I think enough intelligent and opinionated people at the core of that project that it doesn't go off the rails. I think they have a mailing list and there's lots of discussion on that and people talk about what they'd like to see and that sort of stuff. Then there are community leaders like Mike Pinkerton who actually works here at Google and is here this week and other people within that community that are sort of leaders that steer the discussion and make sure that bad things aren't accepted, so I think that that works quite well for them. It's like— I think that's sort of the right mix of having an inclusive process but also being careful not to let things become unstuck.

Then there're other projects— I mean, Thunderbird, I'm not sure what the developer— I think Scott Macgregor uses— I'm not sure what communications infrastructure he uses for that, but I know he works full time on it as does David Bienvenu, both who work for the Mozilla Corporation and then there are volunteers and I'm not sure what the primary communication model for them is, if it's a mailing list or if it's a forum on a website or something like that, so I don't know if you're speaking to him, but he would know more.

TS: This is not on our list, but it falls from some way you said it and I was interested. How does a project like Firefox deal with a situation if a leader drops out, if someone, or a Camino or something else, like it's someone who does provide that sort of central vision and that kind of direction, that strong leadership. How is that kind of— How is that person replaced? What is the fallout or have you had that experience?

Goodger: I think it's interesting because I've been reading a book lately which I think is— which I've been enjoying which is called *Producing Open Source Software* by Karl Fogel and it talks about these two models. At least it talks about the benevolent dictator model and it talks about another model that's used elsewhere by other projects and it says that over time projects transition from benevolent dictator into this more democratic system and I think Firefox may have actually been the reverse of that. If you look at SeaMonkey and the way SeaMonkey was developed back in '02 or so, that was sort of a very— If you wanted to change something, if you got the required approvals, you could and sort of transition to the system of being more sort of benevolent dictator style, but I think that if a leader— if a single leader leaves, I think the other people in the peers group will probably take over those responsibilities, either one person will step up or as a group, that they will handle it. Those are the two options I guess. I don't think anyone, completely aside from or has never been involved would step in, or someone who's not known to the peers group would step in and I think that the leadership has to come from within that.

TS: And do people drop out? I mean, do people disappear?

Goodger: Sometimes. Not typically for any of the application front ends, the products themselves, but sometimes for individual— some modules, people either due to time commitment or something else, they will go away sometimes, they'll come back six months later or something like that. Sometimes they'll disappear for good. In that case, whatever code they work on is either taken over by somebody else. There may have been several people working on it, or the code will begin to sort of stagnate a little bit until somebody else comes along that can work on it.

OR: So kind of switching gears here, why do you think Mozilla in particular, Mozilla Firefox has been able to attract such a large number of users?

Goodger: I don't know the exact answer. I think it's probably a combination of different things. I mean, from an engineering point of view I'd like to say that it's because it offers better features or is in some way more compelling as a product and I think there's an amount of that. I think it filled a need from the market at a certain time. People became dissatisfied by Internet Explorer and its security issues and pop-ads and all that sort of stuff. Also I can't help but think that the community marketing— I think that that's had a positive effect and all of the buzz that it's had in the press as a result of that and a few other initiatives like the *New York Times* ads and all that sort of stuff. And then also there's this sort of— Well, I mentioned before about Mozilla and it sort of seeming like an independent entity. There's that I think which might help. It seems like it's not produced by a company that wants to find— this is really part of some secret agenda to figure out some way to sell you something. And then also there's sort of this aura around Firefox. I think it is a really well-crafted combination of name and I think the Firefox name is probably the— one of the best names I've heard for any of the Mozilla software and the sort of imagery icon, the logo, and I think those things work well together and people sort of anthropomorphize sometimes and talk about Firefox as a person or something that they relate to. I was talking to Bret Wilson who I work with and he said that his girlfriend talked about Firefox versus Internet Explorer and she said Firefox crashed too much but she said that— The way she characterized it was that Firefox was very moody and Internet Explorer was a robot and I think some people— The sort of draw for some people was that, that it seems more— I don't know, more something that they can relate to. I don't know how or why.

It's maybe something intangible, but the name itself is interesting, the story behind that. We changed— The name was originally Phoenix to signify sort of rising from the ashes of Netscape and all that sort of stuff and it turned out that Phoenix Technologies, a company that makes BIOS and stuff like that was actually— didn't want us to use that name so it had to change and so we choose something similar sounding which Firebird which is similar to Phoenix and that was also used by another open source project so we had to change it again. These are unfortunate problems you have to solve, especially when you're picking out users, so we had to figure out a new name and we spent about two weeks and these two weeks were— There wasn't a lot of engineering done in that time. It was sort of beating heads against walls and I read the dictionary trying to find words and we came with about three of them and they all were terrible and so Jason Kersey who's the maintainer the Mozillazine, the community news website, was just

running through a bunch of words that started with fire. He was going firecat, firefox and then I said firefox and then I went and looked it up in the U.S. Patent Trademark Office website to see if anyone had trademarks on in software and I don't think there were any and I said that sounds really good and he wasn't even serious. Like he said I wasn't serious. That's a dumb name and I said, no, I think it's great and so we told it to the people, Mitchell and so on and she went and did a formal investigation.

As a group we sort of sat down and looked at a bunch of different— We looked at about a three different names. I can't remember what they all were, but I think firecat was one of them, Firefox, firecat, firefly and tried to pick the best one and we as a group decided Firefox was the best one and so she went and did lots more investigation and found out where there were potential issues and solved them all and then I think the Foundation applied for the trademark so that we'd never have this problem again, so that's sort of the— That's how that name came about.

OR: What do you think sets Firefox apart from other Mozilla projects? Maybe in terms of being able to attract large numbers of users. I know there're other projects that have attracted a large number but not quite in the same way that Firefox had been able to.

Goodger: I think Firefox solves a problem that a lot of people— solves sort of pain points that a lot of people were experiencing, like some of the things I mentioned with Internet Explorer, security and pop-up ads and all that sort of stuff and I think Firefox provides a relatively convenient solution for those. One of the things at Firefox that I tried to really stress was making the transition easy so we made sure that our import wasn't just whatever previous versions of Netscape had, importing IE favorites and all that sort of stuff, but actually going and trying to understand what stumbling blocks would be to people's adoption, like if people had saved all their passwords and I use the password manager. If they came over to Firefox and they tried to log into their sites we wouldn't remember it and if they'd forgotten their passwords, they'd have to go back to IE to get into the site. So we figured out a way to import everyone's passwords and all that sort of stuff, to make sure that that transition was seamless so that since people experience this pain, we can offer them a solution and say that the cost or difficulty for you as you this change is so close to zero that it doesn't really matter and so that's sort of been our focus.

In other places like Camino has more of trouble I guess attracting users than Firefox even though I think Camino has a higher market share on Mac than Firefox has on Windows. I'm not sure if that's true, I can't remember. Just because of the fact that the default browser on MacIntosh isn't— doesn't subject users to as much pain. I mean, Safari is a descent browser and people use Camino only if Safari doesn't meet their needs and Safari meets a lot more needs than IE does. The same problem that Thunderbird has is that a lot of people use web mail these days or e-mail and it's often mandated and rolled out by your IS department and stuff like that. It's not something that you can necessarily configure easily because you need to know all of your settings and put them all in there, that sort of stuff. So those are some reasons I think why they may not have attracted as high user counts, but I think over time either those projects will come up with more

compelling features or migration will be easier or fundamental user behavioral patterns will change or something.

TS: I just want to interrupt ourselves here for a minute. I don't want to keep you here too long. I want to keep an eye on the time.

Goodger: How long is your list of questions?

TS: We've got a bunch more but we can shorten things. We'll be done by 11:00.

Goodger: Okay. That's good. Then we can go and find I think Darin you're talking to next.

OR: How would you compare Mozilla's vision in 1998 with the present reality?

Goodger: I'm not even sure what Mozilla's vision— Let me think about what their vision was then. I think when the source code was released at the very beginning, the intent was to just provide source. And then a little bit later on, so what happened was since they provided source code— I mentioned Jason Kersey before. He actually started the site before he became the owner of Mozillazine. He owned a site called MozBin and their purpose was to take source code and compile it and actually give people something they can use because most people don't have the tools to do that work, at least on Windows or the time because it takes about 45 minutes or so to compile and so he did that work and posted builds. And so then eventually Mozilla sort of expanded its goals a little bit to actually distribute software to people but just for testing and towards the end of that, I guess around the start of '99 or so, expanded to include that, so not just developers but also for testers. And that turned out to be very successful because community testers were interested in running the software, running bleeding edge people love to find out what's going on and be ahead of the game and all that sort of stuff and that turned out to be very successful for reporting bugs. And then as Netscape began to sort of exert more of a stranglehold than even the engineers at Netscape would've liked, Mozilla began to take on the notion that it could perhaps be the organization that ships the pure or more usable version of the software that Netscape was shipping and then, as time went on, Firefox.

I think when the Mozilla Foundation was formed, the mission was to preserve choice and innovation on the Internet and I think that that is a really good mission. That's where they are today. It's a nice sort of— It's fairly broad, but it accommodates all of the individual projects that are occurring within the Mozilla project as a whole and Internet encompassing not just web Firefox but also e-mail, chats, things like Chatzilla and things stuff like that, just various other tools surrounding those projects. So I think that that's— that's where the mission is today.

OR: The Spread Firefox website states that Spread Firefox was founded on the same principles of community involvement that drive the development and testing of Firefox.

How do open source principles sort of influence those marketing techniques, do you think?

Goodger: I think it's just a matter of having a— The thing that makes open source projects function healthily is good communication and having projects available that people with spare time will want to contribute to. I think it's difficult for engineering projects to do that because people who are at the center of a project may not necessarily know what other folk are interested in working on and communications can be difficult. But with something like Spread Firefox, a lot of people have enough time to help with some of the marketing projects that they do and a lot of them don't require that much commitment and all of them are very high payoff and are exciting. So people are interested and not everyone has programming skill and it's a way that they can help contribute to supporting a project that they love in a way that suits them.

OR: So have you ever contributed to something on Spread Firefox or—

Goodger: I haven't contributed to marketing for a while. I did a bunch of marketing work before Spread Firefox was set up just in terms of making the product pages on Mozilla.org, writing the copy for them and all that sort of stuff. But since then mostly engineering and project management but not so much marketing, aside from sending in my 50 bucks for the *New York Times* ad.

OR: Do you think that's because there are marketers who are specifically now working on Spread Firefox?

Goodger: I think it's because that system seems to be functioning pretty well and I myself, my time is always fully booked. Well, it's probably booked in excess so I just sort of find some sort of things to do and I try to look at the highest priority things and marketing's— I don't mean to say marketing isn't the highest priority, but marketing seems like it's being well handled. So it doesn't need folks to jump in and I also think that there are many more people that are interested in that than engineering, so—

TS: Could you just very sort of briefly describe sort of the kind of relationship between people working kind of inside for the Mozilla Corporation and people who are working almost entirely on Mozilla and Firefox projects at other companies like Google or Red Hat or wherever? How is that managed and how does it work?

Goodger: This is just my personal opinion, but I don't think it works as well as it could and this is nothing new. This is nothing new to the Mozilla Corporation; this is something that goes back years. When I worked at the Mozilla Foundation, people would often say what are you doing, what's going on, we don't know, tell us. And oftentimes a lot of these complaints would have a very sort of whiny tone to them and so people would either not get back to folk or there was no sort of infrastructure set up for communicating well in an open environment. And I think that that's improving, especially with the new mailing list and news groups that we have, but I think it's—

As someone who's worked both inside and outside, the realization I have now is that there is a night and day different between being inside and being outside because inside it's an open source project. But in some ways it still functions like a company in that if there's somebody sitting over there you might choose to just talk to that person and reach a design decision rather than going through a public mailing list and that means that when you do that, everybody else on the outside has no idea what's happened and then it seems a surprise or there's a shock or why was this change made this way and all that sort of stuff. So that's sort of a source of difficulty and has been. So basically we're trying to encourage people to seek to use the open communication systems first now versus whatever other means, so there's that.

TS: How about any tensions between you now working on— still working on Mozilla projects but at Google with Google, with the Google side of it? Is there any kind of clash of philosophies or competing motivations?

Goodger: Within Google, I don't think so. I think Google is very— Google likes Firefox and they're very much— The attitude that I've seen at least has been very hands-off, like this has been successful so far, why would we mess with it? So I don't think Google has any sort of interest in causing change. I don't think it necessarily— I don't think it could. I mean, the way open source works. People would be upset if something like that would happen and it's not Google's intent to be evil, so that's my thought there.

OR: Just sort of generally, do you consider open source software projects as a public service? Do you think of it as a public service work?

Goodger: I guess that's— Yeah, I just haven't ever used those words to describe it, but I guess so. I mean, that's what they are. Lots of people make software that's free for people to use. I mean, Microsoft made its beta of IE 7 available for download for free for XP users and I'm sure that when they're done with IE 7 they'll make the final version free as an update for their users. But more so— I don't necessarily think that them doing that is a public service because they have very clear goals in addition to it. Companies might make statements that they want to do this or that, but if their motivation is something else it's not really a public service. The mission of Mozilla is to preserve choice and innovation on the Internet, so with that as its primary goal—yes.

OR: What made you, or even did you choose public service over maybe a potentially more lucrative job?

Goodger: I don't think it was necessarily— I don't think that's the way that maybe some people think about it. I didn't think about it that way at first. I thought it was this is something— My angle or reason for doing this was that this is something really useful and interesting. So now after some years, there's also the additional— I feel additional responsibility in terms of making sure that this continues to work properly, but my initial point of view was— I didn't really understand open source or anything that well when I started. It was just something that was accessible.

OR: So you mostly got into it for interest.

Goodger: Interest, yeah.

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

Goodger: I think Firefox is one of those open source projects that sort of says that this stuff is not just for geeks. This is for everybody and this is credible. This can be deployed not just across your servers but to your users. You can roll Firefox out to your entire accounting department and have them use it as their browser and not just have your system administrators maintain your service on Linux. I hope to see more open source projects that focus on things that people can use. Google has been very successful at creating products that people like to use and Google's source code for that isn't open source but clearly there is a demand for usable interesting software so I think open source can benefit from providing that and not just sort of either sitting around and waiting for Microsoft or Apple or whoever else to provide it.

It's also sort of distributed. I mean, people are free— Much like the web, so this web 2.0 thing that's going on at the moment, people sort of feel like they can very easily play a part in shaping the future of how things go. Not every project succeeds. Some of them don't turn out to be that useful to a lot of people but the fact that it's sort of liberating people, the open source sort of liberates people to go and pursue their own destinies. I think it's positive.

TS: That's just about all we've got. Just a kind of question for our sake. Outside of sort of current employees and key developers and people's names that we would probably know, is there anything that you think we should— Anybody we should talk to who maybe we don't know about already, people who you think would be interesting to talk to that kind of maybe haven't been written up in an article or that's been written somewhere.

Goodger: There're a lot of people at Mozilla this week in town so you'll probably find a number of people there that would be interesting to talk to and who contribute to various parts of the code from the lowest level to the highest. One person who is— Just from a different product to Firefox perspective, there is Mike Pinkerton who works on Camino. He's actually in town this week. Whereabouts is Fairfax with respect to D.C.?

TS: A suburb.

Goodger: He actually lives in Ashburn.

TS: It's actually probably about half way between Ashburn and D.C.

Goodger: He lives there and actually works remotely for Google but he's actually presenting this week, so he's another person. He's the leader of the Camino project so

there's another project and that one is running pretty well. He'd be a good person to talk to, but certainly at Mozilla this week, there're some people who contribute _____ some people. Scott Macgregor who works on Thunderbird and there's all kinds of different perspectives.