Olivia Ryan: Could you just state your name for us?

Chris Hofmann: I'm Chris Hofmann

OR: Tell us what you do here at Mozilla and how you came to work here?

Hofmann: I was the first employee hired by the Mozilla Foundation and built out the small engineering support staff that we had in the first 18 months of the Foundation existence. We had a staff of about 10 people. Prior to that, I was at Netscape. I worked at Netscape and have worked on browsers for about 10 years now and was part of the effort to take the Netscape source code and put it into open source and start to create a developer community and kind of get the Mozilla project off the ground.

That was kind of an interesting experiment. That was the first time a commercial company took its intellectual property and source code and put it out to the public and Netscape saw that as the only way that they could compete with Microsoft in the long term. This was in the middle of the browser wars in '97 where Microsoft had thousands of people working on the browser code and Netscape could afford to support a couple of hundred people and so there was no way that they were going to— they thought that they could compete over the long run to make a competitive product. And so this was a strategy to engage the open source community to help build browser software and the idea was that Netscape would harvest some of that and turn it into products and still be able to continue to ship the Netscape browser.

So it was an interesting challenge. The first kind of impression that a lot of developers had when they saw all of the code that Netscape put out into open source was a lot of confusion. It was a code base of several millions of lines of code and it was developed under very competitive and time pressure and so trying to understand the code was often very difficult. Netscape ramped up from the same kind of 10 people working and the company grew to over 4,000 in just a year or two. And so integrating all those people, making them productive, helping in building products led to rapid growth of the code and kind of a spaghetti development process where there were lots of ideas about how browsers should work and trying to inject that and then competing with Microsoft almost a day-to-day release comparisons, you know, who's doing what, who's adding new tags, what features are going in, and there was lots of visibility in the press and it was— Each company was under this constant, so the code was actually not in very good shape and the open source developers that tried to get engaged with the project kind of immediately recognized this.

Browser code is going to be a hard— It's a difficult technical challenge to build a browser when you think about it. Currently you've got over 10 billion web pages out there and that's all the input for a browser and so a browser being able to take all of these very diverse inputs and then through the Internet, through all this technology of routers and hubs and networking configurations, being able to take that and then trying to create a user interface that makes— That harnesses all that and makes it very simple for a novice user to be able to get to that content, to be able to show it on their screen and manipulate it is a pretty tough technical challenge. And so browser software is going to be complex no matter what you do, but all of these competitive pressures kind of multiplied all of those problems. And so once Netscape had put their code into open source and then the feedback from developers was this is too complicated, this is too complex, we kind of saw that if the project was ever going to be sustainable over the long term, we had to essentially like start from scratch, start with a new slate and build the browser engine kind from engine up all over again and so that project was started in about '99 I think.

That was the long and slow and tedious process because then we had hundreds of developers inside Netscape and a growing open source community that kind of reviewed all the work that had been done and tried to synthesize it and create a body of technology kind of starting over from scratch and it was a pretty slow process and very painful process trying to get all of the web compatibility that had existed in previous versions was a difficult process. Trying to build in support for new web standards that were emerging and then trying to solve this user interface problem where people were using the browser for a wide variety of different purposes and trying to kind of sort out this. Should be the browser all encompassing to try and suit all those needs or should we target specific kinds of users or uses of the browser? So the UI challenges were pretty difficult as well. And Netscape was still competing with Microsoft so you had all those competitive pressures injecting themselves into the open source process, so there was tension around that as well.

So, the project was — And most people kind of deemed it as a failure for the first several years because it wasn't producing anything and the first thing tangible that the project produced was the Netscape 6 release and from all accounts, that might've been the worst browser that ever shipped. It was very slow, very unstable, but Netscape decided that it needed to get out into the marketplace. That was around the time that IE 6 was also shipping and so they were under this competitive pressure so Netscape decided that we'll take whatever Mozilla has at that point and try and turn it into a product and try and ship it and that turned out to be both detrimental to Netscape and their reputation for producing browsers and to the Mozilla project it had side effects there as well.

But the engineers that worked on the project at that time knew things could get a lot better. We were starting to turn the corner on a lot of the major problems, engineering problems. We were starting to turn the corner on footprints and performance and the invisible things that make a browser comfortable to use. And so after Netscape 6 shipped, there was kind of a renewed emphasis— kind of charter within the engineers to say let's go out and let's create a really— let's start working towards creating something that we then can be proud of. And a kind of invigorated effort to doing that and so there was a lot of focus on footprint and performance and continued focus on security and web compatibility.

When Netscape 6 shipped, shortly after that CompuServe was also interested in using the Mozilla browser engine and the CompuServe browser, but they were concerned about this web compatibility. So AOL actually funded a bunch of research into trying to define this compatibility so we started testing 1,700 of the top sites and drilling down into the content, finding places where there were compatibility problems. And there was about a $6 million effort into assessing this problem and trying to figure out solutions for it and the solutions came with engineering fixes into the code base. But also evangelism to the sites to try and get more awareness of web standards and how to make your browser compatible across or how to make your content compatible across all browsers. So that was the key factor for our improving the technology.

And then off on the side, a lot of engineers said— the technology at that point was the combination of a web browser, e-mail clients. It had hooks into the AOL chat Instant Messenger and a web page composer. It was kind of this kitchen sink of all kinds of client side Internet technology. And it made a product that was difficult for novice users to use because you kind of look at the way the people used software and if they don't understand it, they start searching through all the menus and when you've got the combination of all these products, the menu trees get very deep and when you're trying to do something, you navigate through these menu trees trying to find out what you're doing and it just slows down the operation. So there was an effort started to kind of split out each of these applications into their own independent application and that's kind of the early generations of Firefox where it was the browser effort towards that and the early versions of Thunderbird also started to be constructed at that time, so we started— We took this kitchen sink application and started to divide it up and that was another kind of key factor that led to the success that we've had in the last couple of years.

Tom Scheinfeldt: When was that?

Hofmann: That was— Netscape 6 shipped in I think it was the end of 2000 and so this effort was kind of kick-started in early 2001 and so there's this focus on compatibility and footprint and performance. We've always put lots of effort into trying to create an architecture that is secure and stay on top of any security problems. That's been an underlying thing. And that focus on security over all these years led to the opportunity for Firefox to grow dramatically in 2003, 2004. That's when Internet Explorer had a number of very severe security problems that were imported. The most severe was an attack where hackers in Russia were able to exploit the IE web server and inject content onto bank sites in the U.S. When a user when to visit their bank site, key stroke loggers were automatically downloaded invisibly onto those systems and then that information was relayed— password and bank account information, for some people, was relayed back to sites in Russia. And at that point, CERT, the Computer Emergency Security Response Team, a government-funded agency that monitors software security, made the recommendation for people to at least put another browser on their system and that was kind of a major turning point. When they made this recommendation, our downloads just for Firefox started to skyrocket. We went from about 30,000 downloads a day of people using our technology to about 200,000 within a couple of weeks of this announcement.

And what people found— So these were all people that were using IE. They had an inkling that there was something wrong with their system. The reports that we would get is: I have been using Internet Explorer; my system's gotten really sluggish; I don't know what's going on with it; I'm getting all these pop-up ads. And what had happened is their systems had been infected with viruses and spyware and things that made use of the browser almost impossible. And so when they heard these reports about these public viruses and all the press coverage around those, they said, well, it's time for me to go look for another solution and that's where Firefox started to bubble to the top of good solutions for people to use.

And what they found was technology that— What they were using was Internet Explorer version 6 which stops— Microsoft, after they had won the browser wars and in 2002, 2001, they had closed down browser development except for just a few people that were doing maintenance and security and it stopped innovating because there was really no business case. They had captured 95% of the market and their business models were around building client software and operating systems. Their use of the Internet wasn't really lined up and user browser technology wasn't lined up with their business skills and so people were using this version of IE 6 that was locked in 2000, 2001 technology and when they upgraded to Firefox for these security problems, they noticed that there was like four or five years of innovation that they were exposed to. They didn't get pop-ups anymore because we have pop-up blocking technology built in. They were able to navigate to content that they wanted to use.

One of the things in the development of Firefox, we looked at how do people use browsers. We kind of like just opened up very basic questions and so one of the scenarios was, well you get up in the morning, get a cup of coffee, sit down at your computer and you start up your browser. And we'd watch people go sequentially from one news site to another trying to get— just like you're reading the newspaper and they would type in the location for five or six news sites and so we figured, we said how can we streamline that or make that easier, so that's where like the application for tab browsing came in and the ability to set multiple tabs and have them be your home page, so it was not a matter of starting your computer and typing in and visiting six or seven sites, it was a matter of starting the browser and those pages that you visited most would automatically load and then you could just click on individual tabs and you could go down the list.

Things like RSS feeds where you get a very short synopsis of the content that's on those sites allows you to get to the stores that you want to quickly, so people that read news, people that do blogging, Firefox was very in tune with trying to make use of the Internet a lot easier and faster. And these new features, this new innovation that went on between 2002 and 2004 when Firefox started to take off, it became visible to people immediately along with these performance improvements. It went from a browser that was old technology, very slow and difficult to use because of this infection with all this spyware and viruses to a new browser that had all this new technology and made their experience much easier and that's where we were able to build out this community of hundreds of thousands of people on Spread Firefox that kind of had this experience with their web browser and were pretty excited about that and wanted to tell other people about it. And so that's kind of how this viral grassroots effort kind of took off as more and more people becoming exposed to the benefits of Firefox.

TS: When did you first start using computers?

Hofmann: Back in college I guess, around 1980 where you had to sneak into the economics lab and type up punch cards, so it was long long ago. I worked at Borland for a while. I worked for some defense contractors working on tracking systems, but then got into commercial software development when I started working at Borland in the early '80s. Worked on Tbase and Borland had a software tools division at one point and I worked on software tools and programming tools and then there were quite a few people that left Borland to go to work at Netscape. Netscape was an interesting combination of a bunch of different— recruiting from a bunch of Silicon Valley companies, SGI and Borland, some people from Adobe and Apple, so it was kind of an interesting culture where we took a bunch of different cultures from around Silicon Valley and they all kind of merged together.

TS: What's the first project you can remember working on, software computer project?

Hofmann: It was these tracking systems doing software testing and analysis of data, stuff like that.

OR: Can you make any comparisons between the way a business model for a commercial software company works as opposed to an open source business model?

Hofmann: It was very interesting being part of this transition from taking a commercial software product like the Netscape browser and putting it into open source and it was a eye-opening experience for just about everybody that was involved in that. The kinds of things open source software and open disclosure and open development principles have is the very first thing that happens is that there's a whole new perspective on security and quality that gets injected like the immediate— From the moment there was an announcement that Netscape was going to do an open source to the engineering organization, it happened immediately.

The engineers, the first thing they said, you can't take my code and put it into open source and have peers and potential employers of mine in the future. I don't want them to see this code that I've been working because it's in such bad shape. It may not have been in bad shape but that's their perspective. They wanted to be able to clean it up and write better comments and make sure that it worked correctly. And so that was the impact that it had immediately. And then the other thing is, well, we know about some security problems that we've got that we haven't gotten around to solve yet; we've got to fix those before we put this or we'll expose all of our users to these security problems that no one else has discovered because they haven't had access to the code yet. But as soon as we put it out there, it'll become obvious, you know, they'll read these lines of code and they'll go develop an exploit, so that— just the announcement, you know, this was kind of an eye-opening experience for everybody that worked on it and people often have these concerns.

Well, all of the code is open. Don't you think that there's the potential for people doing this reading through, discovering a place where security vulnerabilities. We've crossed that gap. The code has been out there for 10 years. People have had the opportunity to develop exploits but people have also had the opportunity to do research projects to go find those. We've had Ph.D.s and professors and graduate students and junior high school hackers all looking at the code for 10 years and we've ground out everything that people can conceive of to protect our users to build in strong security. So for a company like Microsoft that's got all of their code behind a firewall and protected, it would be a disaster for them if they put their code into open source at this point because they're at this stage where disclosure of all those things would generate a whole new round of problems for them. But we crossed this gap and we did it 10 years ago and so it makes our code stronger and it raises the quality bar.

The other kind of impact that it has is that any feature— There's ability to kind of incubate some ideas and thinking in open source still— but as soon as you check in any code to implement any kind of new feature or do anything new, that code shows up in the source code repository and it shows up in the nightly releases that we create for testers and so there's immediate feedback on anything that you do. Commercial companies can go off and they can incubate things and they can develop it and have it evolve and then they usually don't get feedback until they do a beta test or a public beta test or maybe even until they ship the product— They won't even disclose or encourage feedback, But the thing that this creates is it makes it very hard for us to get very far off track with what people will accept in the software and tradeoffs that we'll make against security versus new feature development that improve usability. And so because we have this testing community of 10,000 testers that are downloading our releases every night and we get instant feedback and if we do go off track, if we do go in a direction that we've got a wide variety of people that will start yelling and screaming and that will translate into like press reports that Mozilla is developing this strange feature that could put user privacy at risk and so we have to immediately kind of factor in all that. A lot of it can be noise, but we can summarize and take that feedback to heart and kind of go in and make adjustments if we need to and that happens a lot faster. There's this constant feedback that we're in that commercial companies— It's on a longer time period where they get feedback, so there's a lot more that can be invested before you find out how far off track you've gone.

OR: Do you think in an open source community coders feel a sense of ownership of their code?

Hofmann: Yes.

OR: How does that work, though, if you're—

Hofmann: This idea that I mentioned about you had a bunch of people inside a commercial company doing software development and then the idea that you were going to take the code that they wrote and the comments that they wrote and publicly disclose that was kind of— The reaction was, no, you can't do that and then the management at Netscape said, no, we are doing it. Tell us what you need to clean it up so that you can get it in shape because we're doing this. And so it took a lot of management leadership to kind of make sure that once the decision had been made to kind of make sure that there was— And it was expensive. It took from the time that the announcement was made in January of '98 until the end of March, it took three full months of about 60 people going through the code to fix these security problems, to clean out expletives from the code and to get it into shape where we thought it could be successful in an open source development environment.

TS: How's it sort of decided who works on which parts of the code and sort of who makes decisions on what makes it into— How are those sort of structures arranged?

Hofmann: That part of it, too, is much different than a commercial. If you've got a commercial company you'll have the lead engineer or the chief technology officer or someone saying I've got this idea, we're going to build out this team and they're going to go implement this stuff. With Mozilla, the way it's gone is all of that was in place for development of the commercial Netscape browser. When the code went into open source, you had lots of expertise across all these areas of the code and then those engineers kind of just moved over into this open source environment. But over time, people from the community would just come in. They'd start looking at code. They were looking for the motivations for people to get in involved in the project are along the lines of they're looking for some interesting intellectual challenge. That's one thing that motivates people. They've got an idea and they want to see it implemented in some browser product like Netscape or Firefox or any of the products that have shipped off the Mozilla code base. And so that's their motivation to get involved and then the way that people kind of take ownership of different areas of the code is just based on peer review, based on the quality of their contributions.

We have a system where no code gets checked in to the source code base unless it's gone through a couple of levels of peer review. So that our process is someone comes up with an idea, a bug fix or a new feature and they'll develop that code and they'll attach it as a patch to the bug system so it's not really part of the code base yet and then we'll have someone who has worked in that area of the code previously will take a look at the changes and see that they're going to work right or use their historical background of potential problems and try and ward off— Do kind of a quality of control level. And then another person will do a review of the code based on how it fits into the overall architecture and sometimes that architectural review will be pretty simple, but sometimes it's a lot more complex depending on the kind of changes being made and the kind of impact it will have across the whole system. So it'll go through this review and super review process and if those people agree that this looks okay, it's something that we want, it's something that's going to improve the software, then the okay is given to check that change into the code base and it'll become part of the product, so a lot of— 98, 99%

of the decisions are made between people who have had experience in developing code in that area and people who have kind of experience in understanding the whole architecture.

There are going to be cases where there are conflicts in the assessment of the patch or what it's trying to do or what it actually does and so we have kind of a bubbling up of decision making and then we've got another group that we call the drivers and these are kind of like overseers of the technical aspects of the project and they'll kind of step in to resolve disputes, see if they can get all the parties to agree on modifications that need to be made or decisions that— if decisions can't be made, we plug along until— So there's a lot of consensus building about what's the right thing to do.

TS: How would you move from being a kind of just a contributor to being a reviewer or a driver?

Hofmann: A lot of it is based on just experience with the code base and kind of a desire to kind of move into that position in the project. The reviewers that we had— Like I mentioned, we had kind of all this in place when we moved Netscape engineers over from doing commercial software to developing the open source code base. Over time, people that have come into the community either as— There's grad students that were looking for interesting projects and they started to download the code and play around and develop patches and people recognize that they were doing very good work and so those people have kind of migrated from— They might've started out as just as tester of the code and then they started doing some development work and providing lots of patches and then they became the module owner for a particular—

So they kind of worked themselves up through— Hierarchy is really not a good word for it, but they've worked themselves up through this peer review process and kind of risen based on the merits of their contribution. And so we've gone from this system where it was pretty much a hierarchy of Netscape management and engineers to a place where we're very far away from that now where lots of— The decision-making is very distributed and the idea generation is very distributed as well.

TS: Are there sometimes any tensions between, let's say, people who work on like the rendering engine and people who work in the user interface for Firefox?

Hofmann: Oh, yes. So you look at these— I mean, even the high level things that kind of people evaluate of their browser and footprints and performance and security and stability and web compatibility and then all of the user interface and features, a lot of the project is trying to figure out how to balance those things correctly because you can make things that will improve performance but they also might have impact on footprint and memory usage. You can do things to make the system more secure but they might have impact on making the browser harder to use and usability and vice versa. You can make it very easy to use and that might open up huge security holes so trying to balance all of these things is very important. And so one of the things that happens in the peer reviews and the architectural reviews is trying to factor in all those things and saying of all these things that we think are important to build a good browser how are we going to balance those with this change and sometimes the answer is very easy and you can get through it. Sometimes it’s very complex and it takes months to kind of sort all those problems out and get good discussion and have good communication about exactly what the thing is doing and what impact that it might have.

TS: Are there ever people who end up kind of just getting angry and not getting their way and leaving the project? Have there been kind of—

Hofmann: There’ve been isolated instances of that, but I think the people that continue to work on the project are pretty intensely motivated because they know that they’re having a huge impact, even when Mozilla’s market share wasn’t 1 or 2%, they knew that we had good technology and that it had impact on some people’s lives. One or 2% of Internet market share translates into millions of people and so they know that they can work on a project that had impact for a lot of people. And they knew that there was the potential that just hadn’t been harvested yet for something like Firefox to happen and have like a dramatically increased impact in the way that people use the Internet and access information.

OR: To what extent does Mozilla rely upon the work of volunteers?

Hofmann: Huge.

OR: Why do you think people volunteer?

Hofmann: A lot of these, the motivations are having impact. Interesting intellectual challenge motivates some people. Having your contributions have impact to a wide variety of people. I mean, when we went from having lots of funding through Netscape and AOL and having a hundred-person development organization and then being able— When the code went open source, there was kind of immediate attraction from other companies, Sun and IBM, people that worked with Netscape now could also work more directly and take the Mozilla technology and they were using it either internally or building products based on it. The Sun browser, the IBM browser or pieces of the technology and they could have direct access, they didn’t have to go through Netscape anymore. So that was kind of the first layer of community people, so these would be full-time employees of these companies that just were assigned to go work on Mozilla and most of them had some knowledge of the code already.

And then as I mentioned, we went through the stage of the Netscape technology wasn’t very well received and then we kind of started over and the first generations of starting over weren’t very well received because there was so much work to do. But around Mozilla 1.0 which I think around 2003 was kind of a transition point where the technology had gelled well enough that people could see that, yeah, this is pretty good and it is starting to have impact, wide impact and it could take off. And so that’s where the people that are motivated by the impact that they can have started to grow dramatically.

And the other interesting thing was Netscape had a— They were investing 200 people in helping to build this browser and then we had these volunteers that were coming into the community and just doing interesting extensions or ideas or helping to improve the core code. And the first kind of evolution in the kinds of people that were working is people left Netscape because they thought it wasn't an interesting company to work for and they wanted a new start-up challenge and they were replaced by people out of the community that went to work at Netscape to work full-time that had risen up through this process of working on new challenges and kind of getting involved in the community. So you really— Netscape became dominated by people working through the open source starting as volunteers and then becoming full-time employees and working on Mozilla full time. And the community— The outside community or the volunteer community continued to kind of grow and so by the time that we got to the point where Netscape decided it wasn't going to do browser development anymore and AOL provided $2 million for the Foundation, that was enough funding to support a staff of 10 people and we had huge reliance— We went from 200 people down to 10 people. We still had people at Sun and IBM and a few other companies that were dedicated full time but that probably left us like about 40 or 50 people that were full-time, paid employees working on the technology, so at that point we were heavily reliant on volunteer contributions.

The place where it's probably had the biggest impact is on the international translations. So the Mozilla suite, the full set has been translated into over 105 different languages and Firefox is slowly getting there. It's over 35 now at this point. And no commercial company can sustain that kind of translation and simultaneously shipping in all those different languages like we have with Mozilla and Firefox. Microsoft Office ships in 30 languages and so that's— that's a huge expense and those are all volunteers that do the translation up to this point and it's pretty interesting. Some of these people that do these translations in languages that few of us have ever heard of, the people that are doing that work are like national heroes in some cases because they're the people that are bringing the Internet to these small remote— There is no browser in that language because Microsoft has decided there's not a business case for it. These people can download the source code, go through this translation process that's pretty well established now and they can create a browser for that country and open up access to the Internet. So there's some pretty amazing stories in the international volunteer community and their ability to bring the Internet to places where there is no business case for it.

TS: Why do you think, particularly Firefox, has been able to attract so many users? What sets it apart from other open source projects or even other Mozilla products that allows it to attract so many people?

Hofmann: I mentioned kind of these five areas or six areas of kind of building out the technology and doing the refinements on footprint and performance and stability and compatibility with the 10 billion web pages that are out there. Grinding off all of the rough edges was very important to make a good browser. And then the focused effort on Firefox to kind of look at these user scenarios of how people are using browsers and kind of making the browser very easy to use for those scenarios, those were all important to kind of building out the technology and getting it into shape where a lot of people would be able to use the product and they'd be happy and satisfied with it. And that was an eight-year journey going though and doing that.

And then there's this external market factor where you had this CERT recommendation that said people ought to have a different browser because IE has this track record for continually exposing people to these very severe problems and then lots of people having those problems. There's some AOL and Earthlink and government studies showing that like 80% of PCs had some form of spyware, viruses or ad ware on their system and some of the problems were not so severe, you know, they were just minor, but when you consider 80% of the Internet surfing population has some kind of problem on their system. That created a huge opportunity and trying— And so Firefox had kind of positioned itself to take advantage of that opportunity and then has kind of positioned itself along the way to kind of grow our market share through grassroots and word of mouth. So all of those factors are important, doing the investment, taking the time to build a community and build a technology and then being able to kind of step in when there was a marketplace opportunity have all been critical.

TS: Have you ever contributed either as a volunteer or an employee to any non-Mozilla open source project?

Hofmann: No. I've been pretty much focused on browsers.

TS: How would you define a successful open source project? What constitutes success in the open source realm?

Hofmann: I think it dramatically varies. I mean, I think you really have to look at the software that you're trying to build and what constitutes success for it rather than trying to figure out that problem for all of open source.

TS: So what would you say constitutes success for Mozilla?

Hofmann: Market share, impact. Starting with impact, we want to have impact on people's use of the web. We want to make sure that the web stays available to people that want to use all kinds of technology and stays open and competitive and innovative. And to do that you start to look at, well, what do we need to do to make that happen. Well, you've got to innovate. You've got to have significant enough market share so that website designers and people that are creating content, make sure that they're compatible with your technology. And to get market share, you've got to have a product that people see a need to go get and it's good enough that they continue to use. So that's kind of how things have fallen out for our success.

OR: So if market share is a goal, do you see marketing as something that coders can do or would you suggest or do you think that marketing experts are needed?

Hofmann: With Firefox, we really didn't have any marketing experts. We had a very small team. We had no budget for marketing. We kind of sat around and we said— We asked this exact same question and how are we going to solve this problem with no budget and no marketing experience within the team and we said, well, how about we try an open source approach to this and that was Spreadfirefox.com. And the idea was let's build a community that will help to build a story and their experience and that will help to spread the word and spread the software and let's just throw that out on a web server somewhere and see if it sticks, and it was pretty successful. And, you know, we had to have really good technology to make that happen. We had to have this kind of life-changing experience. It's probably over-dramatic but when people got Firefox, their life got better and they were willing to tell people about that as well. And so that's what happened with Spread Firefox and it fed on itself, you know, we started Spread Firefox, people started to get engaged with it.

That community came up with this idea, well, we need to take this beyond; we need to get into the mainstream press. Well, then the idea kind of gelled into we're going to take out a two-page ad in the New York *Times*. Well, how much is it going to cost to do that? Well, this is $100,000 that we don't have. Well, let's ask the community and let's see if they want this to happen. And so this grassroots growing community and they raised $250,000 in 10 days, which kind of just blew us out of the water. So we needed $100,000 for the two-page New York *Times* ad. Then it grew beyond that. Then people started writing about the story about how we were taking a new approach to marketing and how we raised this money so quickly and how the ideas got kind of bubbled up and through peer review and open source and these same kind of principles that we were doing with development kind of found their way to this marketing effort as well. And so there were stories written about that as well, so you have to get into the mainstream press as well as this grassroots effort. And we were able to kind of talk to people about what was happening in the grassroots effort and that translated into the mainstream press and so we got the benefits of both out of that.

TS: What, if anything, do you think the popularity of Firefox will do for the sort of open source as a whole and the open source movement as a whole and where do you see this sort of future of open source software going?

Hofmann: There really haven't been any examples that people could point to as a kind of a success path in taking huge amounts of market share, on the consumer side. On the server, on the back end side, open source dominates, right, and that's kind of invisible to lots of people. Having Apache have, what is, 70% market share, is a huge success story, but no one's heard about that. Probably no one's likely to hear about that because it's not technology that's in front of them every day that they have to make decisions about, that has direct impact; it has indirect impact, but people just don't see it. So on the consumer side or the user facing side, Firefox has shown that there is a model that you could follow. There's a model that you could start to experiment and play with, different variations that could drive you to success. And I think each project is going to have to figure out, well, what is— We've kind of figured out over time that there's these six areas that are pretty important for us to stay on top of and resolve conflicts and create the right balance, but each project is going to have to kind of figure that out on their own. What's important to build the best software or the software that's going to solve some problem for someone down the road. And then what are the set of market conditions that could come into play that can kind of make the growth take off.

And so one of the projects I'm working on now is Minimo, this Mozilla browser technology for mobile devices. And it's kind of at the very early— You know, I look back and kind of see where Firefox was a couple of years ago, or where Mozilla technology was a couple of years ago, and Minimo has the opportunity to follow this model. There's lots of things that have to come into place, on the market opportunity side and on the technology side. But one of the advantages that we have is, we've got that figured out. We've seen what things are important to focus on, how to balance those and kind of be able to kind of watch how the mobile market in general, the devices that are out there and how the telecoms work and how all the infrastructure works to solve— Mobile browsing has huge problems now. Some statistics that say a person who uses mobile browsing a lot usually— they have one mobile browsing session per month. Which kind of shows you— And it's mostly because trying to use a browser on these small devices is so hard at this point. And there's a bunch of factors that kind of play into that: slow bandwidth connection, the screen size is small, it's hard to type or now to navigate on these things. And the browsing engines in them are very old technology or very simple technology that can't get at the eight billion— They either crash or they don't render those eight billion webpages correctly.

And so we can go back and look at the things that were important for Firefox and a lot of those, these web compatibility problems, these stability problems, footprint, performance, all those things kind of play in. And if we're able to create the right balance on these mobile devices and we're able to have kind of the right set of market factors come into place and the right amount of openness on the part of the telecoms and the distributors of the devices and the interest of them in our technology, there's a potential that we could do similar things in the mobile browsing market. But it's a long ways to go on that so we're just kind of getting started.

TS: Great. Well, that's all we have. The only other thing I think we'd ask is if there's anybody who's not here today or that we wouldn't already know about that would have a really good story to tell that we should get in touch with.

Hofmann: I known Brendan is a good one. Have you talked to Brendan yet?

OR: Not yet.