

VE482 — Introduction to Operating Systems

The story of Firefox OS

Manuel — UM-JI (Fall 2021)

The following text was written by Ben Francis, a Software Engineer who worked full time for five years on the “Boot to Gecko” project. As such he gained much insight at the engineering, design, and product levels.

The original post can be found [here](#).

General context

I’d like to tell you my version of the story of Firefox OS, from the birth of the Boot to Gecko open source software project as a mailing list post and an empty GitHub repository in 2011, through its commercial launch as the Firefox OS mobile operating system, right up until the “transition” of millions of lines of code to the community in 2016.

During this five year journey hundreds of members of the wider Mozilla community came together with a shared vision to disrupt the app ecosystem with the power of the open web. I’d like to reflect on our successes, our failures and the lessons we can learn from the experience of taking an open source browser based mobile operating system to market.

This is not a short version of the story, in fact it’s over 7,500 words long, so below is a summary of the key points in case you don’t make it to the end! Hopefully it goes without saying that these are just my personal opinions.

Summary

Biggest Achievements

Design Designed a simple and fresh looking mobile OS with some innovative features like edge gestures for switching windows. Came up with some great designs for how a browser based OS could work, even if the vision was never fully realised.

Engineering Built a mobile operating system from scratch, entirely in the open, and got it to market in less than two years. As far as I know no other team has ever done this. Got web content rendering very close to bare metal on mobile devices with a very lean technology stack, in many cases beating the performance of native code on equivalent hardware. Pushed the envelope of the web with 30 experimental new web APIs and helped create new web app trends.

Product Successfully launched 20 devices in over 30 countries, mobilised the Mozilla community to help run launch parties and created slick branding and marketing campaigns. Sold the message that “the web is the platform”, even if we failed to live up to it.

Partnerships Won unprecedented support from the mobile industry. Repeatedly stole the show at MWC, the world’s biggest mobile event, and got big telecommunications corporations on board with Mozilla’s mission, with competing companies working together towards common goals.

Biggest Mistakes

Design Started out by imitating what already existed, not something simpler, unique and focused on the web. Assumed we could change direction later, which turned out to be harder than we thought.

Engineering In a rush to get to market we imitated the app store model with packaged apps, adding to the problem we set out to solve. Lost focus on what makes the web the web (URLs). Too much focus on the client side rather than the server side of the web stack.

Product After realising “open” on its own doesn’t sell, ultimately chose a strategy to compete mainly on price, which is just a race to the bottom. Suffered from a lack of product leadership and direction after the initial launch.

Partnerships Treated carriers and OEMs as our customers with a tick box approach to feature prioritisation, lost sight of the needs of end users and the original mission. Failed to secure a license to distribute updates directly to users.

What I Would Do Differently Today

Design Start out with a very simple browser based design with a focus on web content rather than try to recreate every feature that other operating systems have. Be ruthless about leaving features out and not even attempt to play catch up.

Engineering Have a clear separation between “chrome” and web content rather than try to force the web to do things it isn’t suited to. Create device APIs using REST & WebSockets on the server side of the web stack rather than privileged JavaScript DOM APIs on the client side. Create a community curated directory of web apps on the web rather than an app store of submitted packaged apps.

Product Not try to create the cheapest smartphone. Create a mid-range tablet (and later possibly a smart TV stick) with a focus on web content, gaming and entertainment. Carve out a niche in developed markets first, rather than try to tackle emerging markets at scale.

Partnerships Be a lot more forceful with partners about Mozilla leading the direction of the product roadmap and be willing to turn down opportunities if they don’t fit with our strategy. Provide software updates directly to end users and walk away from any chipset vendor who blocked us from distributing binaries of our own OS (this would have been very hard back then, but easier now).

Booting to the Web

The *Boot to Gecko* (B2G) open source project was announced in July 2011 with the goal of building “a complete, standalone operating system for the open web.”

There was no mention of commercial smartphones or mobile market share at this stage, merely the desire to “push the envelope of the web” to prove what it was capable of.

The Co-founders

The four co-founders of the project were Andreas Gal, Chris Jones, Brendan Eich and Mike Shaver.

Andreas is a super smart engineer who wrote his PhD thesis on just-in-time compilation of high level programming languages and joined Mozilla in 2008 to build TraceMonkey, the first JavaScript just-in-time compiler. He'd recently finished writing PDF.js, a PDF renderer written in JavaScript, and wanted to show what else JavaScript was capable of.

Chris got his PhD in Computer Science from Berkeley, joined Mozilla in 2009 and had worked on PDF.js with Andreas.

Brendan invented JavaScript and co-founded Mozilla in 1998. When B2G was announced he was Mozilla's CTO, and the executive sponsor of the project.

Mike was a founding member of Mozilla in 1998 and had served as the VP of Engineering and VP of Technical Strategy, but left Mozilla shortly after the B2G project was announced.

I applied to work on the B2G project at Mozilla the day after it was announced so I wasn't employed there at the time of the announcement, but my understanding is that it ruffled some feathers. Partly because it was announced on a public mailing list before any significant internal discussion, and partly because most people (including senior executives) didn't know much about the project and weren't expecting the announcement.

It had the feel of a skunkworks project and was run a lot like a startup, partially outside the normal operation of the Mozilla Corporation.

The Building Blocks

Starting from an empty GitHub repository the co-founders built a small initial team to work on "Gonk" (a "low-level substrate" based on the Linux kernel and Android Hardware Abstraction Layer), "Gecko" (an app runtime and device APIs built on the web engine of Firefox) and "Gaia" (a smartphone user interface built with web technologies HTML, CSS and JavaScript) (Fig. 1).

Some code was re-used from the Android Open Source Project but what was unique about this architecture was that it cut out the entire Java layer of Android, with Gecko rendering web content as close to bare metal as the team could get it. This would give B2G a real edge when it came to performance and would help put web content on a level footing with native code.

This distributed and small but growing team had their first in-person work week in Taipei in December 2011



Figure 1: B2G architecture

where Mozilla's Taiwan office was in the process of being fitted out, in a building overlooking the iconic Taipei 101.

It was at this work week at the Grand Hyatt Hotel that the first ever phone call was made from a web page without the use of a browser plugin. The basics were starting to come together.

The Prototype

At a follow-up work week in Paris in February 2012, the team (now including Michael Vines from Qualcomm and our one-man UX team Josh Carpenter) raced up until the last minute to put together a prototype to showcase at Mobile World Congress in Barcelona that same month.

We succeeded in delivering the initial components described in the original B2G announcement far enough to have a proof of concept demo running on existing Android hardware. We went to MWC with a prototype running on a Samsung Galaxy II smartphone. This was revealed in a joint announcement with our first partner, Telefónica, with additional support pledged from Adobe, Qualcomm and Deutsche Telekom. At the press conference Telefónica demoed B2G software running on Qualcomm reference hardware (Fig.2).

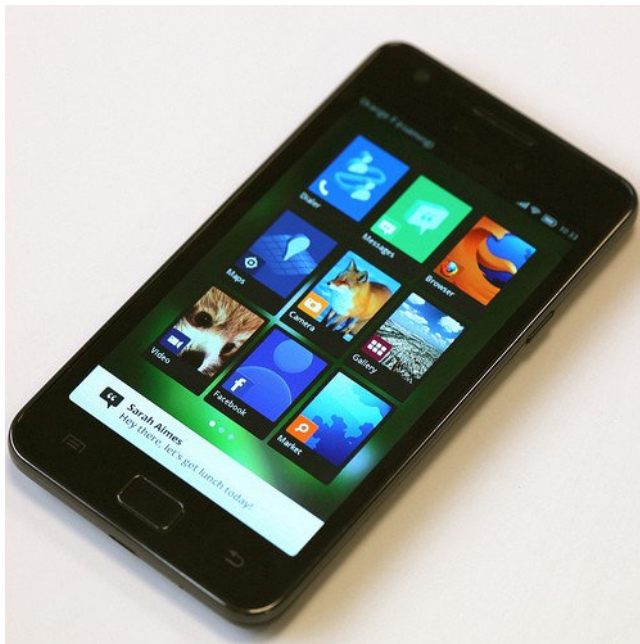


Figure 2: B2G prototype, MWC 2012, Barcelona

This caught the eye of other mobile carriers who were on the lookout for a platform with the potential to challenge Android, to create more choice in the mobile operating system space and wrangle some control back from the dominant Google and Apple.

As a non-profit with a mission to “promote openness, innovation & opportunity on the web” Mozilla was selling a unique vision – not that our own app platform would somehow become the “third platform” on mobile, but that the open web could fulfil that role. Like on desktop, the ubiquity and scale of the web could make it the only viable contender to the incumbent app platforms, with Mozilla leading the way.

Partnerships

With the support of new partners, Mozilla promised to return the following year with commercial devices ready to go to market.

One of the first partners was Telefónica, who had been working on their own “Open Web Device” project.

At first the idea was that Mozilla could provide the underlying Gecko-based platform and Telefónica would add their own Open Web Device front end and contribute their expertise in certifying and taking devices to market.

Working with a huge telecommunications corporation like Telefónica was new ground for open source shop Mozilla and I'd say initially there was a bit of a culture clash. There were a few points of tension and for a while Mozilla and Telefónica worked on their own competing front ends.

It quickly became clear that it made more sense to pool our resources and work on a single mobile front end, and soon Mozilla and Telefónica engineers were working side by side. Both teams of engineers really embraced this approach and it soon felt like we were all one team.

Deutsche Telekom was another partner who made big early contributions and with mobile carriers on board it became easier to get OEMs interested. The first two OEMs were Chinese manufacturing competitors ZTE and TCL.

One of the reasons these partnerships were so important was that Qualcomm, the first chipset manufacturer we were working with, would not license their firmware and device drivers to us directly, it had to be through a carrier or OEM. This meant that we legally couldn't ship complete builds or updates of the OS directly to users, which due to the economics of mobile OS updates would turn out to be a serious problem for us.

Firefox OS 1.0

For the next year Mozilla worked intensively with our partners to build a full production quality mobile operating system.

Design

The brief given to the UX team was basically not to do anything too surprising for the first iteration. Stick to a grid of app icons like Android and iOS and keep it simple, we'd figure out how to differentiate the user experience later on. First we would prove it was possible to build the kind of UI that already existed using web technologies.

The designs the team came up with were pretty slick, but in retrospect starting out by imitating what already existed on the assumption we could change direction later turned out to be a mistake (Fig. 3).

Implementation

On the Engineering side we would prove that it was possible to build a smartphone using open web technologies, with the whole user interface written in HTML, CSS and JavaScript.

It was a requirement for all of the built-in Gaia apps to function offline when an Internet connection was patchy or unavailable, so that the user could continue to make phone calls, take photos and listen to music etc. At first we started to use AppCache for this purpose, which was the web's first attempt at making web apps work offline. Unfortunately we soon ran into many of the common problems and limitations of that technology and found it didn't fulfil all of our requirements.

We also needed to invent lots of new APIs to talk to the underlying hardware of a smartphone from

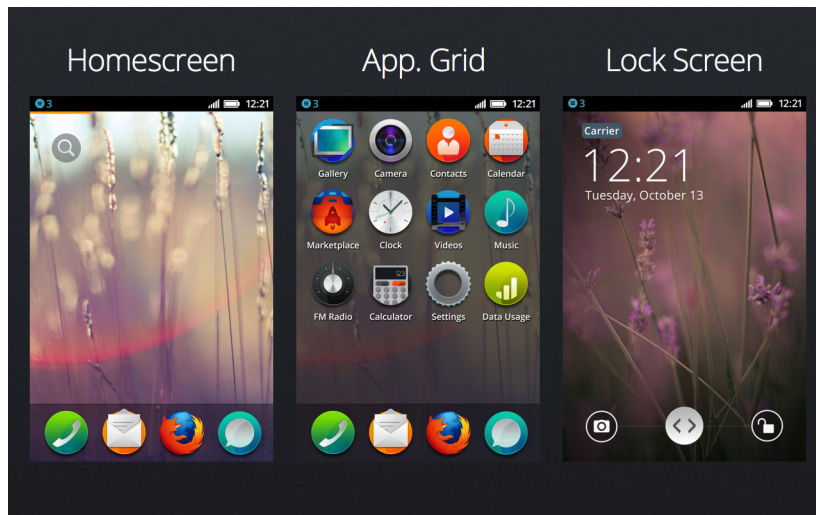


Figure 3: Design imitated was already existed, and was expected to change later

JavaScript, and a privilege model which made that possible. The existing security model of the web allows browsers to safely execute untrusted code by restricting it to a very limited sandbox. In order to provide web content access to much more privileged functions like making phone calls and sending text messages it was thought a new security system was needed.

In order to ship version 1.0 of B2G on time, it was decided that the simplest way to achieve both the offline and security requirements was to introduce the concept of packaged apps to Gecko. A package of resources could be “installed” onto the device to work offline and cryptographically signed to verify that its code is trusted.

Packaged apps solved our immediate problems but they are not truly web apps because they don’t have real URLs on the web and they ultimately have to be signed by a central authority (like Mozilla) to say that they’re safe. I argued against the packaged app approach at the time on the basis that it wasn’t really the web, but nobody could come up with a more webby solution we thought we could implement and ship on time.

At a work week in Telefónica’s offices in Barcelona in July 2012 it was decided to go ahead with packaged apps as a stop-gap solution until we came up with something better. This was another decision which I think turned out to be a mistake because, as well as creating significant technical debt, it set us on a path which would be difficult to turn back from.

Race to the Finish

Anyway, all of this effort culminated with a buzzing work week in Berlin in January 2013 where Mozilla worked side by side with partners, including chipset maker Qualcomm, carrier competitors Telefónica and Deutsche Telekom and manufacturing competitors ZTE and TCL, with the common goal of shipping version 1.0.

This event, hosted by Deutsche Telekom, turned out to be one of the most memorable moments in B2G’s development. Everyone worked together in the same room with a super fast Internet connection, and a friendly competition had been set up between the front end and back end teams to see who could fix all of their bugs first. The results were displayed in real time on a projector so everyone could monitor our progress. There was incredible energy in the room for the whole week while over 200 bugs were fixed.

By the end of the week Andreas Gal declared “zarro boogs” and Firefox OS version 1.0 was born. (The front end team won, by the way!)

Mobile World Congress 2013

The first commercial version of Firefox OS to ship was actually 1.0.1, once all of the necessary testing and certification had been completed. The first commercial devices were announced at Mobile World Congress in Barcelona (the biggest mobile conference in the world) in February 2013. They were the ZTE Open and the Alcatel One Touch and they would launch via Telefónica and Deutsche Telekom in Brazil, Colombia, Hungary, Mexico, Montenegro, Poland, Serbia, Spain and Venezuela. LG and Huawei had also committed to shipping devices (Fig. 4).

The announcements were made at a press conference the day before MWC. We were expecting around 250 people, but on the day over 700 press, media and analysts showed up.



Figure 4: ZTE Open

We managed to get an unprecedented 18 CEOs of telco companies up on stage to express their support for the project, which I understand took even the GSMA by surprise.

I think it's fair to say Firefox OS was really the star of the show that year. We made the front page of the conference magazine every day of the event, we got around 80,000 press results on Google News, and Firefox OS was trending on Twitter with the Oscars! It was a PR dream.

We had a very cool custom designed booth at the conference (in addition to a presence on the Telefónica and Deutsche Telekom stands), with a reception desk, a giant fox tail overhead and demo stations dotted around (Fig. 5).

I was helping to give demos and press interviews at the event and I can tell you that the footfall was non-stop. There was a huge amount of interest from carriers, OEMs and press in

what we were doing.

At an event which is infamous for its “booth babes” (companies hiring professional models to show off their wares), something which I think really made the Firefox OS booth stand out was that most of the people manning our stand had actually worked on the project, and many of them were even volunteers. This was something that really surprised some of the conference attendees, that people would be passionate enough about the project to give up their time freely to help promote it.

This volunteer community involvement continued long after MWC when the devices actually started rolling out in the first launch countries. Launch events in Spain, Poland, Venezuela, Colombia and others were manned by local members of the Mozilla community.

Firefox OS 1.x

Firefox OS had made it to market. But before the 1.0 launch parties had even got going, we were already back at the office hard at work figuring out what came next.

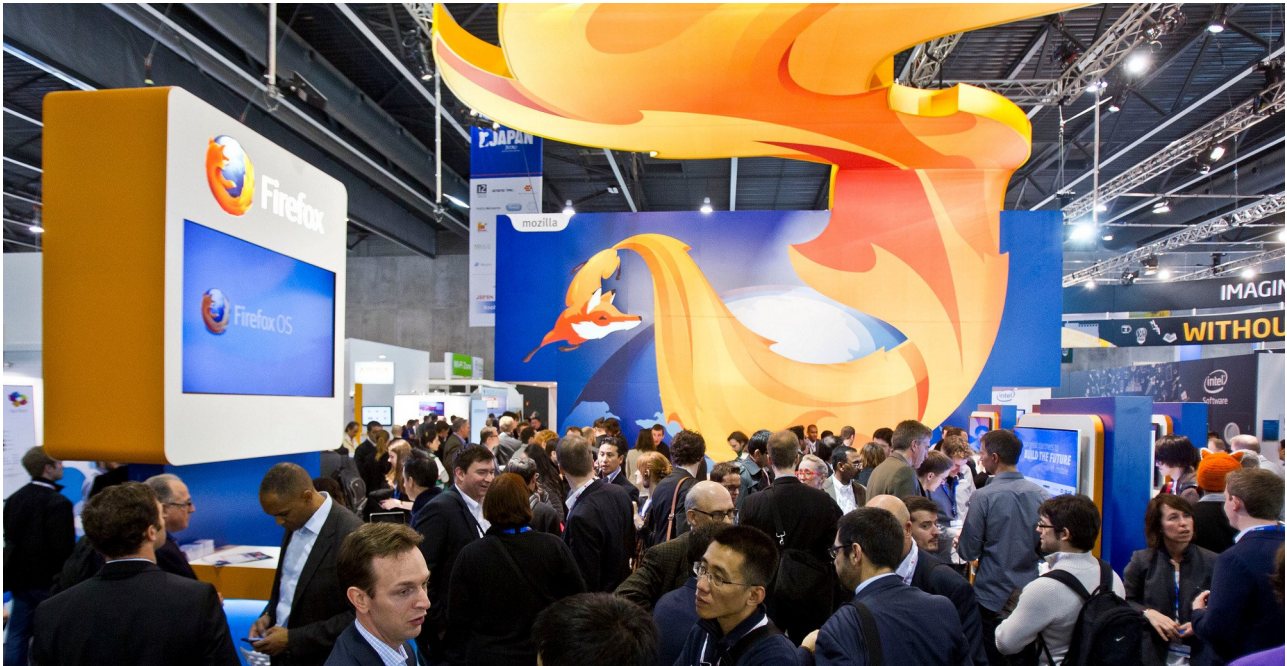


Figure 5: Firefox OS booth, MWC 2013, Barcelona

At first the focus was mainly on adding basic smartphone features like MMS which had been missing from the 1.0 release, then the focus shifted to better development tools for app developers and an awful lot of requirements from partners.

We quite quickly slipped into a pattern where we were treating the mobile carriers and OEMs as our customers. They had a never-ending list of requirements which were essentially copy-pasted from their Android device portfolios. They wanted Firefox OS to reach feature parity with Android.

I remember spending an entire week shut in a room in Qualcomm's offices in San Diego in 2013 with a group of very hard-working South Koreans from LG, making sure that all the built-in Firefox OS apps started quicker than equivalent Android apps on the same hardware. For confidentiality reasons they wouldn't tell us how many milliseconds the Android apps actually took to launch on their device, only whether or not we were quicker. We'd just have to keep hacking away until we had the faster launch time. The LG Fireweb finally launched in October 2013 running Firefox OS 1.1.

Over the course of 2013 and 2014 versions 1.1, 1.2, 1.3 and 1.4 were released. Eventually it became clear that we were just chasing the coat tails of Android, and with Android having a five year head start on us we had no chance of catching them. If we wanted Firefox OS to actually compete in the market and gain any significant market share, we had to differentiate.

Differentiation

Everyone basically agreed that we couldn't compete with the likes of Android and iOS on their own terms. We couldn't catch up with Google on Android features and we could never out-Apple Apple on design. Mozilla was used to punching above its weight and had taken on titans before and won, but we wouldn't win if we played by their rules - we had to play by our own rules.

The way I remember it is that there were basically two schools of thought about how to differentiate Firefox OS.

The Web is the Platform

One school of thought was that we should not try to reach feature parity with our competitors but instead look for a different take on the user interface. We should come up with a much simpler UI which worked to the strengths of the web.

A big challenge we had was that the press would often compare Firefox OS with its competitors based on how many apps were in the app store. Firefox OS was never meant to have a single central app store, the idea was that the web was the platform. Unfortunately some of the early technical decisions we made around packaged apps had backed us into a corner where we were essentially dependent on the few thousand apps submitted to the Mozilla-run Firefox Marketplace.

We tried to patch over the problem by partnering with a startup called Everything.me who maintained a directory of web apps, in order to boost our content catalogue. But ultimately by relying on a central directory we'd ended up just adding to the centralised app store problem we'd set out to solve.

We talked about refocusing on real web content (by far the biggest collection of digital content on the planet) and coming up with an innovative user interface optimised for the interlinked nature of the web and the “flow” of navigating from page to page via hyperlinks. Working to the strengths of the open web would allow us to come up with something genuinely different to the competition.

Connecting the Next Billion

The other school of thought was based on the assumption that what we really needed was scale, and to get scale our best bet was to target emerging markets. They were the real growth areas in the mobile industry.

We did some market research and user studies in developing countries and it quickly became clear that much of the world's population was simply priced out of the smartphone market. What was needed was an affordable smartphone that people in places like India and Bangladesh could actually buy.

If we could crack emerging markets then we could very quickly scale Firefox OS. Then at least we could escape the “other” category on those market share pie charts and start to gain some traction.

The \$25 Smartphone

In mid-2013 Li Gong took the helm of Firefox OS. He was responsible for starting and running Mozilla's China and Taiwan operations, but took on the additional role of “Senior VP of Mobile Devices”.

Li was firmly in the “affordable smartphone” camp and championed the concept of a \$25 smartphone, which was felt was the price point we needed to hit if we were to get scale in emerging markets.

The team pulled off an incredible feat of engineering by managing to get Firefox OS to run on some very low end hardware based on a Spreadtrum chipset with only 128MB RAM – way below the minimum requirements for Android at the time.

We went to MWC 2014 with the “\$25 smartphone”, new partnerships with low-cost handset makers Intex and Spice and a new message around “connecting the next billion”, which seemed a good fit with Mozilla's mission. This message resonated well at MWC, with the mobile industry wondering where to grow next (Fig. 6).

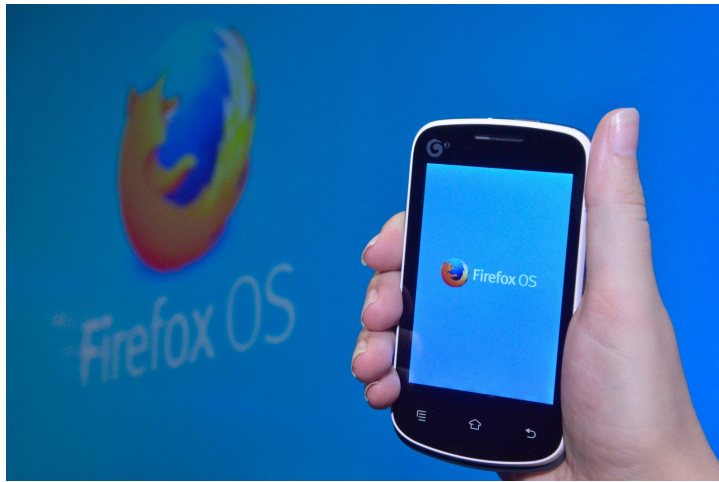


Figure 6: The \$25 smartphone, MWC 2014, Barcelona

With our new affordable smartphones Firefox OS started to expand into Asia (India, Bangladesh), Latin America (El Salvador, Panama, Nicaragua, Guatemala) and Eastern Europe (Czech Republic).

Unfortunately the execution was not so great. Firstly, the \$25 smartphone eventually ended up costing more like \$33 by the time it got to market. Also whilst Firefox OS “ran” on 128MB RAM you couldn’t really multi-task in the way you’d usually expect on a smartphone.

Another serious problem was the lack of

a key app, Whatsapp, which was essential for many of these markets. We failed to convince WhatsApp to make a web version, or even let us write one for them. These shortcomings, coupled with some hardware quality issues (particularly around the camera), meant that these affordable smartphones didn’t do very well in the market.

The problem with competing on price is that you soon get into a race to the bottom and whoever has the biggest economy of scale ends up winning. Google were quick to notice this chink in their armour and responded with Android One, affordable devices targeted at emerging markets. Android One arguably didn’t really do very well either, but it was enough to expose the fact that simply competing on price was not going to be a winning strategy.

Just because people in emerging markets may not have much disposable income, doesn’t mean they’ll settle for poor quality. They don’t want a cheap knock off of an iPhone, they want an iPhone. If they can’t have an iPhone they’d rather stick with their feature phone. It’s affordable, it works and it has a far superior battery life.

Firefox OS 2.0

Most of the work to create the \$25 smartphone had been done by engineers in the Taipei office with a special release called 1.3t. Meanwhile, other teams had been thinking about what a version 2.0 might look like.

In April 2013 the UX team held a summit in London where they got together to discuss future directions for the user experience of Firefox OS. I was privileged to be invited to participate. We brainstormed about what was unique about the experience of the web and how we might create a unique user experience which played to those strengths. UX Summit, London, April 2013

A big focus was on “flow”, the way that we can meander through the web by following hyperlinks. The web isn’t a world of monolithic apps with clear boundaries between them, it is an experience of surfing from one web page to another, flowing through content.

Haida

In the coming weeks the UX team would create some early designs for a concept (eventually code-named “Haida”) which would blur the lines between web apps and websites and create a unique user experience which would flow like the web does.

This would eventually include not only the universal “Rocketbar”, which would be accessible across the whole OS and seamlessly adapt to different types of web content, but also “sheets” which would split single page web apps into multiple pages which you could swipe through with intuitive edge gestures. It would also eventually include a content model based around live apps which you could surf to, use, and then bookmark if you chose to, rather than monolithic apps which you have to download and install from a central app store before you can use them.

In June 2013 a small group of designers and engineers met in Paris to develop a throwaway prototype of Haida, to rapidly iterate on some of the more radical concepts and put them through user testing.

This drastic change was too big to get out in one release, so Haida was split into different phases. Version 2.0 brought a visual refresh and the implementation of edge gestures for moving between windows. Version 2.1 included the integration of the browser into the main system UI to start to create the unique browser-based experience.

But by version 2.2 the focus on the Haida initiative started to slip. Competing feature requests started coming in from partners to create greater feature parity with Android, improve performance and add support for right-to-left languages so that partners could ship devices in more countries.

The full Haida vision was never fully realised and people started talking about another change of direction for a version 3.0.

Firefox OS 3.0

After better than expected sales results reported by partners for the first device launches, the \$25 smartphone initiative had been a disappointment. With no clear point of differentiation having yet fully materialised to continue the momentum of the initial launches, partners’ sales numbers were starting to look disappointing. A loss of confidence started to emerge and disagreements about which direction to take the product ultimately resulted in the departure of several key executives.

Li Gong left Mozilla to create a fork of Firefox OS through his startup Acadine Technologies, with \$100m of funding from several Chinese government-backed organisations. When Li left he took a lot of engineers from Mozilla’s Taipei office with him, which caused a set back for Mozilla and some bad blood. This turned out to be a real waste because the \$100m eventually failed to materialise and Acadine was wound down.

Andreas Gal also left to create Silk Labs, taking Chris Jones and other key members of the original B2G team with him. With Chris Lee, the head of Product, having also made a departure there was a real vacuum of leadership at the top of the Firefox OS team.

This was around the time that Chris Beard took over as CEO of Mozilla and it fell to him to temporarily take charge of the Firefox OS organisation while they tried to hire someone. Chris is a great CEO and as a long-time Mozillian has a deep intuitive sense of what Mozilla is, but as CEO he didn’t really have the time or capacity to actively lead Firefox OS from a product point of view.

3.0 Ideation Process

With a lack of any real product direction, the “v3 Ideation Process” was kicked off at a last minute whole company meetup in Portland in December 2014. The idea was to crowd-source ideas for a new direction for Firefox OS, from individual members of the Firefox OS team and the wider community.

On the surface, for people like me who had well developed ideas about what direction I thought we should take, this seemed like a great opportunity. But as around 36 ad-hoc groups were set up all brainstorming different ideas, good ideas started to get lost in the noise. With no real constraints put on the ideation process and an insufficient process for evaluating them, people were coming up with all sorts of suggestions from smart watches to reinventing the concept of currency!

In the end no clear direction emerged and the 3.0 release was downgraded to a “2.5” release with some hurriedly put together features.

The Pivot

With an underwhelming 2.5 release and still suffering from a complete lack of product leadership, during 2015 Firefox OS started to stall.

Eventually, Ari Jaaksi was hired as the new head of “Connected Devices”. The Firefox OS team was renamed the Connected Devices team and at a whole company meetup in Orlando the “pivot” was announced.

Mozilla would stop our work to build and ship smartphones through carrier partners and we would instead start to prototype user experiences around connected devices and the Internet of Things. The technology created for Firefox OS would be used as a starting point and Firefox OS could live on with devices like Smart TVs (which had been showing some promise through a partnership with Panasonic), but we would no longer focus on smartphones.

Nobody noticed. They were too busy chatting among themselves and trying to absorb this news and what it would mean.

The premise of Ari’s talk was that Firefox OS had set out to compete with Android and iOS and it had failed. Firefox OS was too late to market, the app store hadn’t taken off and the smartphone war had been won. It was time to move onto the next big thing – the Internet of Things.

This analysis was a little frustrating to me because I’d never felt that what we’d set out to do was to make Firefox OS the third mobile app platform, it was about pushing the envelope of the open web to make it a competitive platform for app development. It was true that the project was stalling, but we’d had some really good ideas with Haida, what we’d been lacking was focus.

However, I tried to embrace this new direction as I’d long suspected that tablets would have been a better form factor for Firefox OS than smartphones in the first place, and the Smart TV work was showing promise. These were both “connected devices” after all (Fig. 7).

The “Product Innovation Process” was kicked off, another bottom-up ideation process to generate product ideas, but this time with much more structure (arguably too much). Connected Devices operated much like an internal startup incubator, teams were sent on “lean startup” training and we all had to become experts in the emerging field of the Internet of Things as we pitched product ideas to a “Product Innovation Board”.

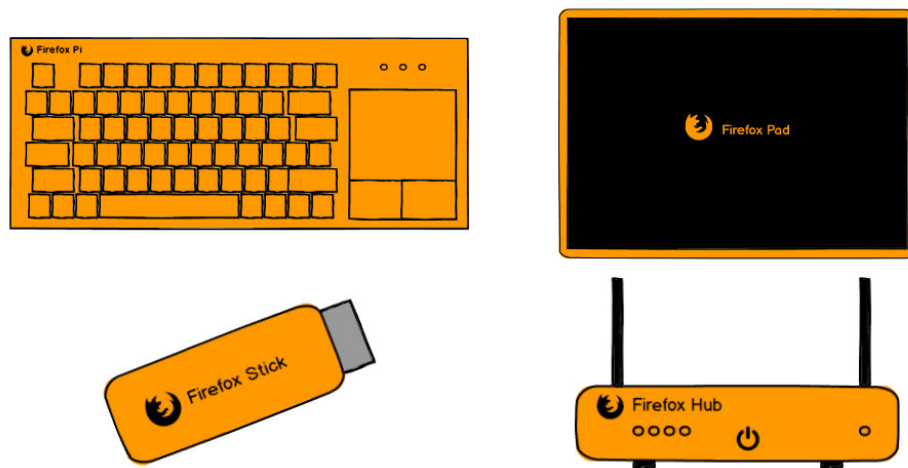


Figure 7: Some of the proposed Connected Devices products

The Transition

The announcement in Orlando was quickly jumped upon by some inside Mozilla as a signal that “Firefox OS is dead”. With all the resources that had been put into Firefox OS over the years, other teams at Mozilla had been feeling the pinch. The flagship Firefox team and supporting platform team had been complaining about a lack of resources for a while, and with Firefox market share slipping the finger of blame was pointed at Firefox OS.

I’d say some of this resentment was maybe justified but it started to spiral out of control and Firefox OS soon became a scapegoat for everything that wasn’t going well at Mozilla. There was a general feeling that Mozilla had “bet the farm” on Firefox OS and it hadn’t paid off. Significant political pressure started to grow inside Mozilla to remove all traces of B2G from the codebase and re-assign resources to our flagship product, Firefox.

The Connected Devices team was downsized, with around 40 people being made redundant. Meetings were held to evaluate what value the existing B2G codebase still had with our new direction, and how much of it could be removed. The argument was the B2G code was slowing down Firefox development and the cost of maintaining it couldn’t be justified.

With some Connected Devices projects (like the tablet and smart TV projects) still planning to use parts of the B2G code, we agreed on a plan to drastically slim down the B2G architecture. We’d completely remove the mozApps packaged app runtime in favour of new web standards, and remove around 30 device APIs we’d added to Gecko which were clearly never going to become web standards, in favour of a new web services based approach (Fig. 8).

The “B2G Transition Project” was started and the Mozilla community were involved in the process so that any smartphone-specific code could be transferred to community ownership if anyone wanted to continue to maintain it. The community really rose to this challenge and got involved in the process, ironically generating the greatest level of community involvement the B2G project had ever seen.

Although we successfully managed to remove the mozApps runtime and 10 of the 30 device APIs from Gecko very quickly, internal political pressure continued to grow to completely remove all traces of B2G from the Mozilla codebase. The tablet and smart TV projects were stopped and the Connected Devices team started to focus their efforts elsewhere.

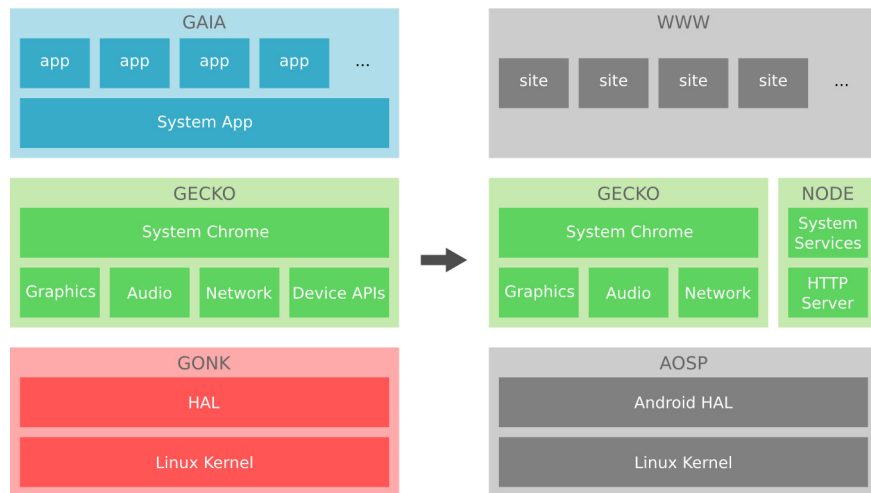


Figure 8: B2G OS Architecture Transition

The module ownership system of the Mozilla project is technically entirely independent of the Mozilla Corporation's reporting structure. As a last ditch effort the module owner of B2G (then Fabrice Desré) disputed the removal of the B2G module from the Mozilla project and the decision was escalated all the way to the top of the tree, but ultimately the decision was to remove it.

Technically the new community-owned "B2G OS" could still continue as a fork, but with infrastructure being shut down and no Mozilla staff being paid to maintain the code, a fork was never really going to be practical. At its peak around 300 full time Mozilla staff were working on the Firefox OS project and the community didn't have anywhere near that number of skilled full time resources.

Boot to Gecko was effectively dead.

Conclusions

Measuring Success

I'm incredibly proud of what we achieved with Firefox OS. If you measure its success against the original stated goals I think the project went way beyond its initial objectives. We went beyond a technical prototype to successfully ship 20 different commercial devices in over 30 countries, including smartphones and smart TVs (Fig. 9). Apart from the incredible feats of engineering, prototyping 30 new APIs for the web platform and building possibly the most complex JavaScript codebase ever created, we built a mobile operating system from scratch and got it to market in less than two years. As far as I know no other team has ever done this.

I don't actually know how many individual Firefox OS devices were sold in the end because our partners didn't share their sales numbers with us and we didn't ask them. As an educated guess I'd say it was probably around 5 million (give or take a million or so). To me that's an incredible achievement, it's a lot more than other "alternative" mobile operating systems have achieved and certainly the most successful project I've ever been involved in. But in an industry which ships over a billion units a year that's not even half a percent of market share!

For me it was never about Firefox OS being the third mobile platform. It was always about pushing the limits of web technologies to make the web a more competitive platform for app development. I think

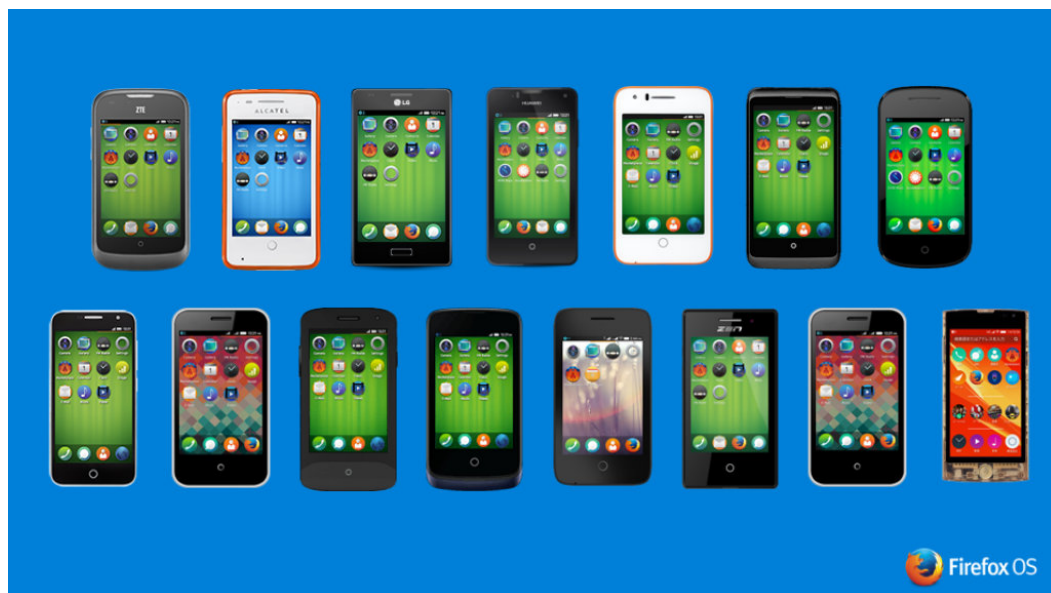


Figure 9: Some of the 20 Firefox OS products that went to market

we certainly achieved that, and I would argue our work contributed considerably to the trends we now see around Progressive Web Apps. I still believe the web will win in the end.

At some point along the way, the team lost sight of our original goals and somehow defaulted to trying to become the third smartphone platform alongside Android and iOS, with market share as the only metric of success. The B2G team grew from four co-founders to a hundred people in a very short space of time and it's not uncommon for a startup which grows very fast to lose its mission and focus along the way. (That's what Firefox OS felt like to me, it was a startup inside Mozilla).

A Matter of Timing

A common narrative is that Firefox OS was simply too late to market. Talking with experts from Shenzhen in China I'm actually no longer sure that's true. They've suggested to me that if Firefox OS started today, the licensing issues which caused us so many problems and forced us to go through mobile carriers rather than ship software updates directly to users would no longer be a problem. As mobile hardware becomes more commoditised it would be trivial today to find white label hardware on which to run our software. There's also a real desire from OEMs to loosen their dependence on Google and differentiate themselves. It's possible that rather than being five years too late, Firefox OS was actually five years too early!

Those of us who work on the web like to think that what we do is really cutting edge. But the reality of web standardisation is that it takes a long time. New features will always come to native code first before they make it through the standardisation process to a cross-platform solution. I'd argue that the web is really more suited to more mature hardware platforms which have become more of a commodity, and that eventually it will win out on mobile in the same way it has on desktop. It could be that the time is quickly approaching when a web-oriented mobile OS could be successful.

The Right Decision?

It's been over a year now since the announcement was made in Orlando and I've had plenty of time to process the various stages of grief associated with the end of Firefox OS: denial, anger, bargaining, depression and acceptance!

Now that the dust has settled I can honestly say that on balance I still think cancelling Firefox OS completely was the wrong decision. The project had lost momentum due to a lack of product leadership over an extended period of time, but I believe the decision to kill it was ultimately based more on internal political reasons than sound business or technical ones.

As the mobile market has matured and become more commoditised, web technologies have also matured. The opportunity is still there, in fact the timing may now be better than ever. Although I understand the reasons the project was stopped, I feel Firefox OS was ended prematurely, before it had the chance to reach its full potential. What it needed was a reboot and some ruthless and clear direction, which unfortunately it didn't get.

Many will argue that I'm delusional and that it's easy for me to say this, after all I wasn't the one who had to sign the eye-watering cheques that needed signing to keep a project of that scale running. They'll say that we had our chance and we blew it. Mozilla should now be focusing all of its efforts on its flagship product, Firefox. For me, that's a little too inward looking. In many ways the web has now moved beyond the browser, it's time to look to the future and all of the new challenges which face the Internet.

Lessons Learned

Sadly, it would be politically impossible to reboot B2G inside Mozilla today, but if it was possible there are a number of things I would do differently a second time around.

Design I'd start with a very simple browser-based user interface with a focus on web content, rather than try to replicate all the features of other mobile operating systems.

Engineering Have a clear split between "chrome" and web content rather than try to force the web to do things it isn't suited to. Create device APIs using REST & WebSockets on the server side of the web stack rather than privileged JavaScript DOM APIs on the client side (JavaScript is not what makes the web the web, URLs are). Create a community curated directory of web apps on the web rather than an app store of developer-submitted packaged apps.

Product Not try to create the cheapest smartphone. Create a mid-range tablet (and later possibly a TV stick) with a focus on web content, gaming and entertainment. Carve out a niche in developed markets first, rather than try to tackle emerging markets at scale.

Partnerships Be a lot more forceful with partners about Mozilla leading the direction of the product roadmap and be willing to turn down deals if they don't fit with our strategy. Provide software updates directly to end users and walk away from any chipset vendor who blocked us from distributing binaries of our own OS (this would have been very hard back then, but easier now).