

Qt and Open Governance



Posted by Thiago Macieira on June 3, 2010[[edit](#)]

It has been a while since my last blog. Last time, I talked about how the [Qt repository in Git was one year old](#). It was a humorous way of showing the growth and success of our contribution model. Or, at least, that's what I intended...

For the past two months I have been working with a group of people inside Applications and Service Frameworks (the unit of Nokia that is responsible for Qt) on opening up our way of working even more.

The change to open the governance model more is the natural continuation of what we've been so far: Open Source is the foundation of the future work in Nokia, as can be seen by the Symbian Foundation, the MeeGo work we're doing with Intel and the Linux Foundation, and especially Qt. And this is not news for Qt either: for over a decade we've been opening up our model: the OPL, the GPL, the LGPL, the daily snapshots, the open repository, the contribution model, etc.

Last year we created the contribution model, whereby anyone could contribute code to Qt. To enable that, the open repository was clearly necessary, as contributors need access to the latest changes and to the individual commits. It has been a success, with hundreds of merge requests integrated in Qt and Qt Creator.

The next step is even more open governance. The model we have selected is that of a typical Open Source Project. That is to say, beyond contributing code to Qt, we want to allow people to know even more where the project is going, and get involved in its actual development process and decision-making structure, like a many Open Source projects do. The rationale is very simple: we want people to be informed and involved – and involved deeper than just adding one-off patches to the projects. It's the best thing we can do to show our community how much we want them to use, support and grow Qt with us.

A few things are clear to us that we will need to open up in order for this to happen. It's clear that we will need to move our technical discussions to the public, as well as the decisions that affect the product, like roadmap and schedules. We will also need to somehow give the community access to the QA process, like test results and reports, coverage reports, integration reports.

And we'll also need to open up the decision-making structure. That is to say, contributors who have shown themselves to be trustworthy and good at what they do deserve the right of having a say in the decisions. Take, for example some of the contributors of the past year: there are a couple of cases where they know the code better than people working in the Qt offices. We have come quickly to the point where we have to say "I trust you that this contribution is good". This is part of the [meritocratic process](#) that we want to have in place.

While those basic things are clear to us, we can't claim to have thought of everything. This is where you come in: we need input and help in making this happen. We need to come up with a way of working that works for both us and for the rest of the community. We have an idea what might work for everyone involved, but those ideas need refining.

So, what we'll do soon is start these discussions; in public and with the public (details to be announced later). Nokians will be participating in these discussions to ensure that we can come up with a model

that works for everyone. For example, there are a few legal aspects that will need to be taken care of and we have some ideas on how they could work. All of these details that we have thought of and our initial suggestions on a model will be shared, so as to get the discussions going.

We expect to reach some level of consensus soon. After we (together) come up with a model that works for everyone, please understand that it will take some extra time to adapt to it. There may be tooling work that is necessary, and there may also be changes to the/our way of working. The changes won't happen overnight, but we have reserved some time for them to happen.

In the meantime, we can and probably will start to address some low-hanging fruits.

Anyway, these have been interesting weeks since we decided to go down this route. I'm hoping that the next weeks of discussion are equally productive.

Possibly related posts:

1. [Qt/Windows Open Source Edition to support VS Express \(13.112\)](#) So here is more dramatic news following the Qtopia Phone...
2. [Trolltech Open Source Development Award: Finalists \(10.029\)](#) We use a lot of tools and applications when we...
3. [Qt 3 and 4 licensed under GPLv3 \(9.932\)](#) I know this information is in Trolltech's frontpage right now,...
4. [Open Source embedded development: My Gumstix Story \(9.626\)](#) I recently got a new embedded Linux device to decorate...
5. [Qt Public Repository Launched! \(9.381\)](#) With the announcement back in January of Qt going LGPL...

{ 18 comments... read them below or [add one](#) }



[Robin Burchell](#) [June 3, 2010 at 3:09 pm](#) [\[edit\]](#)

Excellent news! It's great to hear that the ball is now properly rolling on this, hopefully the results will start to become more visible. It may well be that this largely renders useless the work I've been doing on community integration – to be honest, I hope it does.

While I don't want to inject pessimism into this, it should however be noted that it is very different to say something, it's another to *do* it, but at the end of the day, I assume good faith, and I will do my best to help push this along where I can.

Can you elaborate on the legal issues that have to be considered during this process? It might be nice to know the constraints and gotchas that need to be kept in mind while mulling this news over, before the discussions and heavy lifting have to start properly.



[Thiago Macieira](#) [June 3, 2010 at 3:20 pm](#) [\[edit\]](#)

Hi Robin

The **doing** will happen soon. Like I said, there are low-hanging fruits, things we know that need changing. We are doing them as we speak, and results will be announced when they happen.

I don't think your efforts in community integration of patches will be wasted. It will be some time until we can move on to the model that we (together) agree on using, including the necessary changes to the tooling. Your work will both serve as input to the changes we need to implement as well as serving

a particular need until those changes are made.

My point in this blog is to make one thing clear: we are committed to seeing this through.



Thiago Macieira [June 3, 2010 at 3:53 pm](#) [edit]

See also <http://blog.qt.nokia.com/2010/06/03/open-governance/>



scorp1us [June 3, 2010 at 4:13 pm](#) [edit]

This I think would be welcome. While we're all fans of the Trolls and Qt here, many of us were disappointed at what is in (or rather not in) the 4.7 release. The problem I see with open governance is though is I think many people will push for more features to be included (right now bug fixes are not a problem) which would break the usual 6-month release cycle.

How do Nokia/Trolls feel about more or less releases per year? (Immediately, I recognize if we have more releases, that means more Q/A and packaging churn. Is that a big deal?)

What is on the road map? Correction: Thiago's link provides that. But Some of those research things I would rather have today than as research projects.

Specifically, things I see Qt needing sooner than later:

- An Open Source Flash replacement, based on QML+QtScript. (HTML5 just won't cut it) (Browser plug-in, standalone EXE and IDE (QtCreator))
- real SOAP servers and clients via the Services API
- HTTP REST
- JSON serialization/AJAX in Qt itself, not WebKit

The last 3 are so Qt apps can communicate with web apps without needing Webkit, for light-weight apps that talk to web services. They are vital to the mobile app space.

The first one is well, freedom for all, so Adobe /Apple can stop their bickering.



[digitalsurgeon](#) [June 3, 2010 at 5:15 pm](#) [edit]

@ scorp1us QJson is an excellent JSON library for Qt, check it out.



[Robin Burchell](#) [June 3, 2010 at 5:21 pm](#) [edit]

@scorp1us:

I think it's better to leave road-mapping of future features to ..the future, simply because it's difficult to paint the fine details on a picture without first having brushed out the background in rough detail. Let's wait until some of the path has been laid down and we have more of a concept of how things are going to fit together.

Besides that, moving to a meritocracy also means that the very point you mention (people pushing for more features) will – hopefully – become less of a point of contention, because it will no longer be (for the most part) left to Nokia alone to implement such requests. So instead of requesting a feature and moving on, you can (in theory) happily write it yourself and push it into production-ready state and have it integrated into Qt, and maintain it yourself.

Regardless, any project faces similar issues with scheduling features, and at the end of the day, provided you keep a pragmatic approach to it (and are prepared to drop something if it impinges on quality), it isn't too hard. The only difference is that such decisions will – hopefully – eventually be in the open, and discussed by all of the people contributing to Qt, not just those employed by Nokia.

Moving aside and discussing a more important point you raise.. that of more frequent releases. With many hands, all work becomes light.

I touch on this above, but simply put, if something is of high quality, is well tested, and integrates well, then there should be no problems pushing out more frequent releases containing updates like this. The only real problem that this poses is for Qt on devices (such as MeeGo ones), if the vendor decides to stop updating Qt, as developers targeting that device (or maximum portability) would have to sacrifice newer features.

This, however, is no different from the situation of Qt on any other platform, and at the end of the day, I think that consumers will vote with their money, and avoid vendors that drop support too quickly, making this less of a problem.



Thiago Macieira [June 3, 2010 at 5:47 pm](#) [edit]

@scorp1us: you're raising points that are very important and relevant... but not for right now.

We do want to have those discussions with the community and share the decisions: where should Qt be going, how often it releases, etc. But please remember that we want to implement a meritocracy, so, while we will listen to you and others, in the end it's about who does the code.

But that's not a discussion for right now: first we need to figure out **how** the discussions and decisions will take place. Then we start discussing and deciding.



JoeMerchant [June 3, 2010 at 7:26 pm](#) [edit]

@ scorp1us , I've seen you asking for SOAP 1.2 support before, and I couldn't agree more when I tried to get SOAP info off of weather.noaa.gov . SOAP and similar interfaces will be extremely helpful to mobile app developers.



scorp1us [June 3, 2010 at 11:11 pm](#) [edit]

@Joe, Yes, and I might have to take matters into my own hands after QtMobility's Service API gets stabilized. 😊 The challenge is writing it using only QtCore (as QtXML's DOM is getting deprecated... grumble grumble.)



quinn [June 4, 2010 at 2:06 am](#) [edit]

So is this a selfish or a selfless move?

I think that's gotta be clear before you proceed. What would happen if one of the trustworthy contributors, which are not employed by Nokia goes rogue and e.g. sneaks malicious code into the codebase, hijacking millions of phones, tracing their messages – or something.

I think you really have to envision the absolute worst-case, just to have done it, make yourself conscious about it – and then, finally, proceed.

That said, I think it's the best move ever made 😊



James Mansion [June 4, 2010 at 9:30 am](#) [edit]

Its all very well talking about meritocracy from a developer-of-Qt community perspective, but I think you should make clear how you're planning to deal with different inputs from:

- developers of Qt
- users of Qt (developers of apps and systems built on Qt)
- users of apps and systems of Qt

Its easiest to hook up to the first of these, but ultimately its the feelings of the third group that matter the most in terms of reaching beyond the incestuous tech community.



espenr [June 4, 2010 at 11:57 am](#) [edit]

@quinn: Well, this is already a problem for other open source projects like Linux etc. Commits are in the open, anyone can see and review. I don't think this will become an actual problem.



damien [June 4, 2010 at 1:40 pm](#) [edit]

Wy I love Qt ? It's because it's user friendly. I had never seen a better documentation a better opensource project, so user friendly. Qt act as a refecrence for other project evaluation. When a have to evaluate a technologie I ask myself how best as Qt is it ?

Many times I have ask wy Qt is a so good open source project, why many other open source project failed ?

For me the answer is Qt is User centric, user experience centric.Qt user's are developpers. Not developer who develop Qt but users wich use Qt framework.

Build your gouvernance upon the meritocrati of Qt'developpeur and you will get a big bazaar, a bazaar. Build your gouvernance connected with the market, that to say, the user need, and you will continue to have something great to use.

Trolltech know that, and the aim of trollech was to satisfy user because it was the trolltech market. I was haapy to use and pay for that. Trolltech is gone, Nokia aim is to sell phones, so they bought the best for theyre phone, and they open it to save money on developpement. Users are happy to that. But if you change the "paradigm" build to satisfy personns who build, and not to satisfy person who live in the building (users), Nokia will break the last toy they bought.

It's the reason for, for me it's a bad news to open gouvernance in such way. And finally what'is so crazy ? It's a lot easier to open a form to vote on what user want to be develop than your solution.



Drizzt [June 4, 2010 at 1:41 pm](#) [edit]

Would this step improve the bug fixing rate for (almost) trivial bugs, where big users have already a workaround in place too? I hope so, because there are really some bugs which could be fixed "overnight" so to speak. From the top off my head #QTBUG-10219 comes to mind. The solution is already (partially) there in QTopia and fully in KDE.

So thinking this to the end, the above announcement would mean, that e.g. some KDE devs could get

access and therefore move the “custom solution” to the Qt source where it’s available for all. Right?

Cheers,
Drizzt



[Robin Burchell June 4, 2010 at 2:44 pm \[edit\]](#)

@Drizzt,

Such things are hard to tell, but I’d say: yes, more or less.

Qt becoming more open means (in the longer term, not now) that more people, not necessarily employed by Qt, can get the access they need to fully scratch their itches.

That means (amongst many other things) – once a contributor has trust – they will not have to submit merge requests and wait (potentially months) for the right person to come along and take care of it.

This will have two effects in the long term, IMO: it will hopefully speed up patch integration (more hands to do that integration), and this will in turn hopefully encourage more people to submit patches in the first place, as their work will be better taken better care of.

With regards to the bug you mention specifically: it’s one thing to point to a workaround, it’s another to actually work to have it integrated in Qt (both at current). Stuff like that won’t just magically do itself, now or in the future. Someone *will* have to put in the legwork to get it integrated, if it is to happen. This just hopefully empowers everyone to do that a little easier.



[Thiago Macieira June 4, 2010 at 3:58 pm \[edit\]](#)

@Drizzt,

What Robin said, plus: it’s not about magically accepting every single contribution. Contributions must still be good for the project, create good and coherent APIs, be cross-platform, take Qt in the correct direction, etc.



[anoncoward June 5, 2010 at 12:07 am \[edit\]](#)

It was my understanding that Qt’s superior quality came from the fact that it was NOT based on an open model, that it’s put together by a few well integrated group of individuals with good communication and payroll motivation, instead of a big sprawling open source project with egos and factions and mailing list communication. I hope Qt stays quality.



[Thiago Macieira June 5, 2010 at 9:49 am \[edit\]](#)

Qt’s quality definitely comes from the group of talented people working on it, the motivation behind it and the processes and culture around it.

That doesn’t have to change.

Payroll has never been a strong motivator. And we do use mailing lists internally, and IRC channels, besides walking to other people’s offices. Again: that doesn’t have to change, we’re just making the lists public and the IRC channels. The walking to other people’s offices will be a challenge, but no greater

than it is to walking to Australia from Norway.

Previous post: [Qt Simulator is going public](#)