How the

KDE Free Qt Foundation

strengthens Qt

by Olaf Schmidt-Wischhöfer (board member of the foundation)[1], December 2019

Executive summary

The development framework Qt is available both as Open Source and under paid license terms. Two decades ago, when Qt 2.0 was first released as Open Source, this was exceptional. Today, most popular developing frameworks are Free/Open Source Software[2]. **Without the dual licensing approach, Qt would not exist today as a popular high-quality framework.**

There is another aspect of Qt licensing which is still very exceptional today, and which is not as well-known as it ought to be. The Open Source availability of Qt is legally protected through the by-laws and contracts of a foundation.

[1] I thank Eike Hein, board member of KDE e.V., for contributing.

[2] I use the terms "Open Source" and "Free Software" interchangeably here. Both have a long history, and the exact differences between them do not matter for the purposes of this text.

The KDE Free Qt Foundation was created in 1998 and guarantees the continued availability of Qt as Free/Open Source Software[3]. When it was set up, Qt was developed by Trolltech, its original company. The foundation supported Qt through the transitions first to Nokia and then to Digia and to The Qt Company.

In case The Qt Company would ever attempt to close down Open Source Qt, the foundation is entitled to publish Qt under the BSD license. **This notable legal guarantee strengthens Qt. It creates trust among developers, contributors and customers.**

The KDE Free Qt Foundation is a cooperation between The Qt Company on the one hand and KDE on the other hand. KDE is one of the largest Free Software communities for general purpose end-user software, founded in 1996. In case of ties, KDE has an extra vote, ensuring that The Qt Company does not have a veto on decisions.

My in-depth presentation below provides an overview of the history of the Foundation and describes its importance for Qt today. It explains in detail why the **existence of the Foundation has a positive influence on the long-term market success of Qt**.

Note 1: *This document was written by Olaf Schmidt-Wischhöfer and published in December 2019 after receiving comments from the other board members of the foundation. In early January 2020, the board members representing The Qt Company informed me that the text does not fully represent their views. I received no specific suggestions for textual changes.*

Note 2: *This document has* ***not*** *been updated to respond to the announcement of The Qt Company from 27 January 2020* [4]*. I have not yet decided on publishing an updated version or not.*

[3] From the statues of the KDE Free Qt Foundation: "The purpose of the Foundation is to secure the availability and practicability of the Qt toolkit for developing free software."

[4] https://www.qt.io/blog/qt-offering-changes-2020

In-depth presentation

1. Qt: A commercial success. An Open Source standard.

Since its first release in 1995, Qt has become an amazing success story. In the market of frameworks for development and user interfaces, it has established itself as a clear industry leader with its high quality, with its consistency and ease of use, and with its wide cross-platform support.

A **key enabler** for this achievement is the **license strategy** of Qt. The framework is available for Open Source usage. It is also available under paid licenses for proprietary software development.

What Open Source Qt means in practice

Qt is developed as a true Open Source project[5]. People from different backgrounds join with The Qt Company to contribute to the framework. Many contributors come from companies that work for Qt. Many come from Qt-based Free/Open Source Software projects, for example the KDE community. All of them need to sign a license grant for their contributions[6]. Thanks to the KDE Free Qt Foundation, they **know that their contributions** will continue to be **available as Free Software**.

Some other companies have dual-licensing strategies where the Free Software releases are limited to small core part of the product. In these cases, the Free releases are incomplete and function as advertisement for the full product. Developers need to be careful building their work upon such products, since they never know whether the Free version

[5] https://wiki.qt.io/The_Qt_Governance_Model

[6] https://www.qt.io/legal-contribution-agreement-qt

might be closed down at short notice. **The negative perception of "Open Core" products is countered by the guarantees of the KDE Free Qt Foundation.**

Licenses of Qt according to the contracts of the foundation

Most parts of Qt are available under the Open Source license LGPL, which allows usage in closed source software under certain constraints. This ensures a large market share of Qt – in a world where the number of permissively licensed Open Source libraries is growing.

Platform:	essential modules			most add-ons			some add-ons		design & develop-ment tools		support (Service Level Agreement)
X11 (Linux)	GPL	LGPL	paid	GPL	LGPL	paid	GPL	paid	GPL	paid	paid
Android	GPL	LGPL	paid	GPL	LGPL	paid	GPL	paid	GPL	paid	paid
Windows	GPL	LGPL	paid	GPL	LGPL	paid	GPL	paid	GPL	paid	paid
Windows Phone	GPL	LGPL	paid	GPL	LGPL	paid	GPL	paid	GPL	paid	paid
iOS	GPL	LGPL	paid	GPL	LGPL	paid	GPL	paid	GPL	paid	paid
MacOS	GPL	LGPL	paid	GPL	LGPL	paid	GPL	paid	GPL	paid	paid
other platforms			paid			paid		paid		paid	paid

	GPL version 2, 3 or later	LGPL version 3	paid license
This license can be used for products that are themselves Free Software under the GPL:	✔	✔	✕
This license can be used for closed source products:	✕	✔	✔
There are additional provisions for the benefit of product recipients (e.g. soure code, patents, reverse engineering):	✔	✔	✕

Releases of Open Source Qt are made on the same day as the paid versions of Qt. The contract with the KDE Free Qt Foundation allows a delay of up to 12 months. In some cases, this has been used by The Qt Company to release new modules first to paying customers and only a bit later as Open Source.

2. Benefits of the availability as Open Source

Market relevance

Increasing the market penetration of Qt

The Qt framework is used by many Free Software applications, such as those developed by the KDE community. Because of this, **Qt is included in almost every Linux** distribution. Many developers already have it on their computers and start to use it. This enables a strong growth of its usage – and of license sales. This was a key point during the early growth years of Qt, and this mechanism is still important today.

Reducing competition from other frameworks

The wide availability gives Qt an important advantage. Developers can easily evaluate it for their own software projects. **Other frameworks** that are not available under Free/Open Source terms have **difficulty competing** against this. For this reason, the majority user interface libraries are available as Open Source.[7] Since Qt was one of the first to go Open Source, it could establish itself in the strong leadership position it has today.

[7] https://en.wikipedia.org/wiki/List_of_platform-independent_GUI_libraries

Developer uptake drives contributions and license revenue

Growing the Qt talent pool as a major benefit for Qt license customers

Free/Open Source Software organizations touch the lives of many of the leading industry engineers of tomorrow through significant education programs and student-age participation. **Qt's license customers benefit** from the increased talent pool to hire from, often making a significant difference in the decision to build on the Qt technology.

Growing Qt licensing revenue

The young developers taught with Open Source Software go on to choose Qt again and again in later stages of their careers, promoting growth in Qt licensing revenue.

Contributions to Qt development

Free/Open Source projects have written, directly or indirectly, much of the Qt technology. They contribute code to Qt regularly, while many of The Qt Company's own engineers were originally exposed to Qt through them as well. The Qt Company's engineers use Free Software projects as test cases for new Qt developments, and their developers as willing and receptive audience to early-adopt and provide feedback on changes. Free/Open Source organizations regularly host The Qt Company engineers at their development sprints and events to enable this to happen.

Promotion of Qt

Free advertisement for the Qt technology

Free Software organizations and projects using Qt do much to promote the Qt technology, frequently and publicly touting its benefits and justifying their choice of the technology. The organizations, non-commercial, appear as less biased than their commercial counterparts, creating a strong narrative of Qt's technical strengths and superiority.

Showcase of the Qt technology

Free/Open Source projects choosing Qt act as freely and highly available demonstrations of the technology's prowess and suitability for the wide range of use cases. Some of these projects have reached hundreds of millions of users. Unlike many of Qt's other success stories, they are not subject to confidentiality terms.

3. A historical perspective on Open Source Qt

The business history of Qt includes several transitions to different companies: Trolltech, then Nokia, then Digia and now The Qt Company. During all this time, Qt continued to be available both as Open Source and under paid licensing.

The legal contracts of the foundation were written to survive such transitions. The aim of the KDE Free Qt Foundation stayed the same. The exact nature of its support for Qt developed and grew when there were changes in Qt and around Qt.

Establishing Qt as a standard solution

Since 1999: Open Sourcing made Qt successful

Today, **Free/Open Source** offerings are the **dominant players** in the market of developing frameworks. When Trolltech first released Qt as Open Source, the software world was still a very different place.

Qt was among the first to drive the trend of Free/Open Source frameworks. And Trolltech greatly benefited from it. Free Software projects such as KDE were able to use Qt, so Linux distributions shipped Qt – in many cases in the default installs. The great dissemination of the framework led to a steadily increasing usage among developers.

2005: Qt as a formal industry standard

The Open Source nature of Qt made it possible for the Linux Standards Base (LSB) to include it as a **mandatory part** in its "desktop" specification. The KDE community enabled this by acting as an independent advocate for its inclusion.[8]

Originally, the LSB had been planning to standardise on a competing library named Gtk as the native UI framework for Linux. And the LSB was not alone in trying to establish Gtk as the standard solution. Many Linux distributions contributed to the Qt competitor, as did many volunteers. One motivator for this were fears that the company behind Qt might close it down in a future version, thereby harming the Linux desktop. The existence of the KDE Free Qt Foundation was a major force for addressing this fear.

[8] https://marc.info/?l=osdl-lsb-discuss&m=117555449706767&w=2

Without KDE and the KDE Free Qt Foundation, Qt would not have been standardised as one of two UI frameworks for the Linux desktop. It would not have grown to the popularity it has today.

Supporting evolvements of the business model for Qt

When software development companies decide upon a framework to base their work on, they need to consider a number of different factors: quality of the framework, supported platforms and programming languages, availability of consulting and services, licensing terms and costs, but also the business model for developing the framework.

Different models have different advantages and challenges:

Business model of Qt:	**Model 1**: Qt is the **main product** of its company	**Model 2**: Qt is developed by a large **multinational company**
Examples from Qt's history:	Trolltech, The Qt Company	Nokia
Advantages:	The company can focus exclusively on improving Qt. There is no interference from other company departments into the development strategy of Qt.	There can be major synergies between Qt and other company work. Qt development can be cross financed. Availability of Qt as Open Source can be more important to the business strategy than licensing income.
Challenges for building trust into the future of Qt:	Qt contributors and users may fear an acquirement of the company by investors that do not act to the best of Qt's interests. They may fear the company restricting Open Source usage of Qt to generate short-term license revenue, thereby harming	Qt contributors and users may fear that the company ties Qt to other company products and abandons Qt parts that are not well aligned with the company's other products. They may fear the company abandoning Qt altogether if its larger business strategy changes.

	the long-term prospect of Qt.	
How the foundation addresses the challenges of trust:	The foundation guarantees the **unrestricted continuation** of Open Source Qt. The legal setup of the foundation **survives acquisitions** and even bankruptcies.	The foundation guarantees that **Open Source Qt is a complete** release of Qt. Abandoned parts of Qt must be released under the BSD license, so others can continue developing it. The **legal setup stays valid** in case Qt is sold.

The legal setup of the KDE Free Qt Foundation is independent of the business strategy for Qt. Nonetheless, the foundation actively strengthened the business of companies developing Qt – by supporting evolvements of the business model. I will give two examples.

Growing developer uptake by going LGPL

After Nokia bought Trolltech, they analysed the market position of Qt and wanted to trigger an increased developer uptake of Qt.[9] They changed the license to LGPL, which allows usage in closed source software. This made Qt more competitive in comparison to other LGPL licensed frameworks. As a result, **Qt grew in popularity as planned**. It also triggered many contributions to Qt (see the next section "Opening Qt for contributions").

The contracts with the KDE Free Qt Foundation were updated to reflect the new license. This enhanced the credibility of Qt for companies considering to base their investments on it. It also ensured that **contributors to Qt can trust** that their **code will continue to be available** as LGPL. With both aspects, the KDE Free Qt Foundation helped Nokia reach its goal.

[9] http://web.archive.org/web/20110808141917/http://qt.nokia.com/about/news/lgpl-license-option-added-to-qt

Pragmatically enforcing compliance, and introducing GPL version 3

After Nokia sold Qt to Digia, the business model changed again. Cross-financing Qt development was no longer possible, so **licensing revenue** became central to the future of Qt (in addition to **continuing contributions** from third parties).

To make it more attractive to buy licenses, Digia introduced a number of closed source modules for Qt. They were available to paying customers, but missing from Open Source Qt. This was not allowed in the legal agreements with the KDE Free Qt Foundation.

The foundation could have insisted on the release of these modules under the LGPL license. It chose to use another, more **pragmatic approach to enforcing** compliance. It negotiated an updated version of the contract which still required a complete Open Source release, but allowed using a **stricter license for new code** (GPL version 3, which cannot be used in closed source software). Existing LGPL code in Qt kept the license (but also updated to version 3).

In exchange for allowing this, the contract now protects more platforms (Android, iOS, MacOS, Windows, Windows Phone, Linux/X11).

The pragmatic approach enabled many benefits:

- The **trust of contributors** was protected.
- Qt received **better funding**.
- Companies could continue using the LGPL Qt code in closed source software and did not migrate away from Qt in masses. As a result, we **avoided a negative narrative** about Qt usage being on the decline.
- **Free Software** projects continued to **use, support and contribute** to all parts of Qt.

- More platforms are covered by the contracts, thus **strengthening trust in the market**.

Opening Qt for contributions

Open contribution workflow

Contributors to Qt are major stakeholders that benefit from the existence of the KDE Free Qt Foundation. When the foundation was created, third party contributions were not yet possible for Qt. This changed after Nokia had acquired Trolltech.

Nokia also set up **open governance**, allowing third party contributions to be involved in the decision making processes[10].

The KDE Free Qt Foundation was involved in setting up open governance. For example, Nokia asked the foundation for cooperation on their drafts for the contribution agreement.

Moving LGPL licensed code from KDE to Qt

The KDE community offers a large number of Qt-based products as Free Software. One of them are the KDE frameworks, which have an LGPL license to allow usage also in closed source applications. Historically, some of the KDE frameworks were a duplication of Qt frameworks, because the KDE frameworks were older, or KDE needed additional features, or bugs in Qt forced KDE to write its own alternative code. With the opening of Qt for contributions and the **acceptance of LGPL licensed code into Qt**, **KDE contributed** many of those solutions to Qt directly.

[10] http://web.archive.org/web/20100924180854/http://labs.qt.nokia.com/2010/06/03/qt-and-open-governance/

The KDE Free Qt Foundation was instrumental here, because the KDE contributors to Qt knew that the license to their contributions could not suddenly be closed down.

Porting Qt to Android and protecting it

One special case where the KDE Free Qt Foundation **enabled a major contribution** to Qt was the Android port of Qt. This was done originally done separate from Qt by a third party developer. It was named Necessitas and available as Free Software from KDE. When Digia bought Qt from Nokia, they wanted to integrate this work into Qt itself. Before contributing his code to Qt, the developer asked the foundation to **include his Android port** in its protection of Qt.

4. Conclusion: How the KDE Free Qt Foundation helps Qt today

In chapter 2, I have described the benefits of Open Source Qt. Through the KDE Free Qt Foundation, these benefits are reinforced and accompanied by additional advantages.

Unlocking contributions to Qt

License guarantees for Qt contributors

For many other third party contributors to Qt, the foundation also plays an important role. It guarantees that their contribution will stay available. For Qt consultancies, the foundation offers the safeguard that there will be **no arbitrary license changes**[11]. This makes it **easier for them to contribute** their code to Qt itself, rather than keeping it in-house.

Discouraging forks

Qt has a dual-licensing approach of both Open Source licensing and paid licensing. It is therefore necessary that contributors to Qt sign a license grant to The Qt Company. This introduces an asymmetry. The Qt Company can sell all contributions, but the contributors cannot sell Qt.

Fortunately, the **asymmetry is counterbalanced** by the responsibilities that The Qt Company has towards the KDE Free Qt Foundations. The contributors can know that their code will stay available as Open Source.

The foundation therefore makes it more likely that third-party changes or additions to Qt end up in the official Qt repository, and not only in a forked repository separate from The Qt Company.

[11] There have been a number of recent license changes away from Open Source – in markets different from the one Qt is serving (e.g. MongoDB, Redis, CockroachDB and Sentry). See also: geekwire.com/2019/mongodbs-licensing-changes-led-red-hat-drop-database-latest-version-server-os

Open Source Qt is the real thing

Third party contributors are far more likely to contribute to a real Open Source project than to semi-closed product. The foundation ensures that Open Source Qt cannot be turned into an incomplete promotional version. Features available solely to paying customers might drive up license sales of Qt. But they would also decrease the perception of Open Source **Qt as a real product**, thereby reducing contributions to Qt (and the other benefits mentioned in chapter 2).

Mutual benefits of cooperation with Free Software projects

Many developers from the KDE community are contributing to Qt. For them the KDE Free Qt Foundation continues to be important for **safeguarding the existence of KDE**.

Enabling Qt business by instilling trust that Qt is viable long-term

Protecting The Qt Company

The legal **setup of the foundation stays valid** even in cases of acquisitions, mergers, bankruptcies or sales of Qt. The Qt framework cannot suddenly be taken off the market through financial transactions unrelated to Qt development (such as a takeover by a competitor). This fact acts as a strong deterrent against hostile takeover attempts.

The foundation ensures that Open Source Qt cannot be closed down (which might trigger new license sales but would seriously harm the market positioning of Qt). This fact protects The Qt Company in the (hypothetical) case that investors try pressuring it into prioritising short-term licensing income over the long-term health of Qt development.

Protecting customers

Because of the foundation, customers of The Qt Company know that **all features of Qt will continue** to be available. The contracts ensure that Qt features can only be dropped if they have been released under the BSD license, allowing other developers to continue development.

Since The Qt Company is protected by the agreement, customers also have a much easier contingency planning about the continued availability of paid Qt licensing.

Keeping the license of Qt competitive

Today, the most relevant UI development frameworks are all available under Open Source licenses. These **competitors to Qt typically use permissive licenses** that also allow usage in closed source software.[12] In medium and long-term perspective, Qt can only stay credible and competitive in this market with a similar license.

The current Qt licensing situation is well-placed here, which Qt Essentials and many add-ons available under the LGPL. At the same time, having the GPL license for newer add-ons allows The Qt Company to still generate license sales.

The KDE Free Qt Foundation protects these licenses, and thus the continued relevance of Qt in the market of Open Source UI frameworks.

[12] Examples: .Net, Xamarin, WinUI etc. (Microsoft); PhoneGap (Adobe); Flutter (Google); React Native (Facebook)

Strengthening the business of The Qt Company by instilling trust in the market

As a result of all the separate points mentioned above, companies are more likely to choose the Qt framework if they know about the protections of the foundation, which **makes the business of The Qt Company stronger**.

Résumé: The role of the KDE Free Qt Foundation for securing trust

During the past two decades, the benefits laid out above have strengthened and grown the Qt ecosystem. As a result, Qt today is perceived as a strong technology to build on and as a technology here to stay. This **perception of legitimacy and permanence**, a key factor in the adoption of Qt, is principally conferred by its availability under Free licensing terms and the guarantees provided by the KDE Free Qt Foundation Agreement.