# KDE Developer Documentation Report Update

09 December 2019

It has been roughly six months since the final report regarding KDE's Developer Documentation was submitted. A lot has happened during that time that has affected the community's targets for the next few years but the overall goal remains the same for this particular aspect remains the same. The KDE Community needs to create and maintain a sustainable documentation system that will help introduce developers to our libraries and to the community.

KDE is beginning a journey of change, from the Frameworks to the new Goals to a growing interest in mobile and cross-platform applications. This represents a prime opportunity for the community to dust off our documentation and take stock of current systems that need to be updated or, if necessary, changed. This document serves as an addendum to the comprehensive report submitted in June to provide updated recommendations for next actions given recent circumstances.

## What's New

There are three large events that transpired since June, specifically in the last four months. While these do not change the need to do significant work on our developer documentation, they better define the things the community wants to and must focus on in that regard.

### 1. Akademy 2019: KDE is All About Apps

Last September, the KDE community gathered to, among other things, vote on what the project's goals for the next few years would be. One of those, perhaps even the most voted one, is a focus on applications. As much as it is about helping users discover these applications, it can also be about helping developers not only maintain existing applications but perhaps even grow that number. This ties into the need to have topnotch developer documentation for our APIs, tutorials, and entry points for interested contributors to different projects.

## 2. KF6 Sprint: Getting the House in Order

More recently, KDE developers huddled together in Berlin to review the status of the foundations of all KDE software: the Frameworks. This was done due to the development of Qt 6, which, in turn, gives an opportunity to start the development of KF6. Part of that development is clearing up dependencies between frameworks. That cleanup also touches on the need to review and update the API documentation as well as the system that we use to generate it for online viewing.

## 3. Plasma Mobile and Kirigami: New opportunities

There is a growing interest among both users and developers to see our software on mobile devices, primarily in the form of a "KDE Phone", even if installed on an Android phone. While it can't really be attributed to a single event, the availability of devices like the PinePhone, the ARM-based Pinebook Pro, and Purism Librem 5 phone has revived the idea of running "true" Linux and open source software on such devices. This is a chance to introduce new developers to our software and libraries, specifically Plasma (Mobile) and Kirigami. These two would benefit from having documentation ready to greet newcomers and answer their burning questions.

# KDE API Documentation

The KDE Frameworks are about to go a period of big change but that isn't going to happen instantly. And even when KF6 finally goes gold, the community as well as external developers will still have to maintain existing software that use KF5. In short, there is no better time than now to brush up our API documentation.

The overall goal and strategy for updating apidocs remain the same. Given the limited resources (time and manpower, specifically), priority should be given to the most used and most important Frameworks as suggested in the June 2019 report. These mostly involve fleshing out top-level Framework and Class documentation and providing examples.

# TechBase Wiki

TechBase still remains the primary location for information that is aimed at external developers but are also pertinent to new contributors. This is particularly true for tutorials for API, libraries, and tools that are used for writing software both for outside use as well as those under the KDE umbrella.

The biggest focus for this area would be not just to update existing tutorials but to create new ones that are more relevant for developers. These tutorials have to go beyond "Hello World"-level information and provide information and code that is both interesting and useful. Tutorials found in Qt documentation are good examples of such topics.

# Community Wiki

The Community Wiki is the home "inward-facing" documentation. It should be a hub for welcoming new contributors and directing them to areas of (their) interest. This is also perhaps the area most affected by the changes in direction and focus.

Whereas the previous report focused on cleanup and organization, the more important action now is one of clarification. There needs to be a heavy focus on onboarding interested developers and newcomers and pointing them to the apps and projects they may want to be a part of. This includes ensuring a clear starting point and path with regards to the tools and processes, especially when building KDE software from source.

# Plasma, Kirigami, and Applications

The ubiquity of mobile devices and the launch of Linux-based phones have made both users and developers interested in KDE experiences on non-traditional computing devices. These experiences don't just code themselves and it could be an opportunity to pull in new developers coming from a different entry point.

These developers will need guidance and documentation needs to be written to conserve the energy and time of those already involved in the projects. Plasma Mobile already has some documentation in place but needs to be updated. Kirigami needs to have its docs started

completely. These two are just the starting point as new KDE developers will also grow to be interested in other parts of the software, including Plasma, KWin, and other applications.

# Suggestions and Concerns

Given the points above, here are some more concrete steps that can be taken towards updating and improving KDE's developer documentation.

## 1. Pre-crisis mode

While there is no need to panic, the window of opportunity for updating documentation might be closing in fast. When work on KF6 starts in earnest, there will be fewer resources to spend on writing documentation and too many moving parts to focus on. By doing the work now, we can lay the foundations that will make updating them easier when KF6 launches. Furthermore, it will also help in guiding developers on software that will be in maintenance mode.

## 2. Content first, format later

KDE currently has a number of different systems and software for documentation. API docs use KApidox, the wikis use MediaWiki, and a few projects make use of Sphinx. Trying to immediately mind the formats and idiosyncrasies used by each would only hamper efforts to write documentation. It might be better, at least at this starting point, to first focus on writing the content of the tutorials before properly adding them to their respective locations.

## 3. Developer Story, Developer Handbook

There is a saying that goes "everybody loves a good story". The KDE developer journey might need one, too. The old KDE Developer Guide and Frameworks Cookbook were good starting points but they need to be less static and more encompassing. As previously mentioned, what form it takes can come after the content has been created. Content that will give developers, no matter where they're coming from, a comprehensive overview of the software, tools, projects, and people that make up KDE.

There are also a few concerns that may need to be raised during the process of updating our developer docs.

1. The MediaWiki translation system, while useful, is unfortunately rather complex and brittle, potentially causing problems for documentation writers and translators as they write or move things around.

2. We are in need of new tools for documentation, such as proper tools for QML and Javascript API, or systems for sharing pieces of tutorial code around while having them stored in a way that they can be checked like any other code.

3. The Community Wiki needs to include information or decisions that have been reached in sprints or important meetings. Not everyone will be able to read blog posts written in the past or open collaboration notes. This can help avoid misunderstanding and miscommunication especially when such decisions were made years in the past.

= Conclusion =

The need and the opportunity for updating KDE's developer documentation have probably never been greater now that the community and its software are on the cusp of another period of transition. The community has spoken on what goals it wants to focus on but that hasn't drastically changed the direction of developer documentation. And while maintenance is an important part of having documentation we can be proud of, it is becoming more urgent to focus on getting content in quality form before we can take steps towards keeping it that way.