



# KDE's Journey to Qt6 and Beyond

2023/11/29, Qt World Summit  
Nicolas Fella - [nicolas.fella@kde.org](mailto:nicolas.fella@kde.org)





Qt





Qt6






# About KDE

About KDE — Kate

 KDE - Be Free!

About Report Bugs or Wishes Join KDE Support KDE



**KDE** is a world-wide community of software engineers, artists, writers, translators and creators who are committed to [Free Software](#) development. KDE produces the Plasma desktop environment, hundreds of applications, and the many software libraries that support them.

KDE is a cooperative enterprise: no single entity controls its direction or products. Instead, we work together to achieve the common goal of building the world's finest Free Software. Everyone is welcome to [join and contribute](#) to KDE, including you.

Visit <https://www.kde.org/> for more information about the KDE community and the software we produce.

Close



## About me

- KDE Contributor since 2017
- KDE Software Platform Engineer since 2023
- Leading the Qt6 transition in KDE
- Software Engineer at KDAB
- Qt contributor





## I've done this before

- betterCode Qt 2021: KDEs Weg zu Qt6
- Qt DevCon 2022: KDE's journey to Qt 6
- QtCon Brasil 2022: KDE's journey to Qt 6 and beyond
- Various talks at KDE's Akademy conference



## KDE's Codebase

- KDE has *a lot* of code: ~15 Mio. LoC
- Virtually all C++, Qt5, QML
- Many independent subprojects
- Multi-generational code
- Both Widgets and QML
- Graphics-intensive applications
- CMake build system



# KDE Frameworks

- 80+ modular libraries, solving recurring problems
- Available for external users
- Stable API + ABI
- Qt6 means new major versions for KDE Frameworks





## Kickoff

- First planning session at Akademy 2019
- Setting initial design goals and principles
  - Easy porting without surprises
  - Improve APIs and drop old ballast
  - Separate Core/Widgets/QML
  - Improve cross-platform support



## Timeline

- Preparation work in 2019 and 2020
- First proper Qt6 builds in 2021
- First Qt6-only builds in 2023
- Qt6-based alpha releases in November 2023
- Release planned for February 2024



What took you so long?



## Problems While Porting

- Some things just take time to port
  - QRegExp
  - QStringRef
  - QDesktopWidget
  - Shader
  - Third-party dependencies



## Missing Functionality

- Some functionality was/is missing:
  - QTextCodec
  - QtGraphicalEffects
  - QtX11Extras
  - QML TreeView



## QML Porting

- Porting QML code was easy and hard at the same time
- Easy because of few breaking changes
- Hard because of no good way to keep Qt5 support
- Wish: `#ifdef` version check in QML



## Bugs

- Naturally we found some Qt bugs
- Many of them reported and/or fixed by us
- Contributing upstream is vital for KDE
- Lack of 5.15 LTS is a challenge



## Good Things

- An openly maintained Qt
- Much improved Wayland support
- Many small but useful API additions
- Improved QML tooling
- Performance improvements





## KDE For All

Improve  
accessibility  
across our  
products

## Sustainable Software

Reduce energy  
consumption and  
hardware waste

## Automation

Automate and  
systematize  
internal  
processes



## Automation

- Formalize and automate internal processes: code review, release management, etc
- Tools like qmlformat and qmlint can help
- Qt and KDE can learn from each other



## Sustainable Software

- Reduce energy consumption of software
- Avoid hardware obsolescence
- Performance improvements in Qt help



## KDE For All

- Improve accessibility of our products
- Lots of activity in Qt
- Still some open issues, esp. in QtQuick



# Strategic Challenges



## QML Evolution

- QML is evolving rapidly
- We have not fully caught up
  - `qt_add_qml_module` not usable for libraries
  - Lots of “legacy” QML code
  - Bespoke QML deployment
  - Issues with third-party QML modules and tooling



## Wayland

- Future of the Linux desktop
- Important target platform for Qt
- Substantial contributions from KDE
- More work needed in Qt and upstream



## QtQuick Desktop Experience

- Widgets vs QtQuick is hotly debated in KDE
- QtQuick is lacking on desktop platforms:
  - No native dialogs, popups, menus
  - Keyboard navigation is hard
  - Lack of high-level controls
  - Different theming system than widgets
  - No unified action class
  - Developer Experience





## Summary

- KDE's Qt6 transition is drawing to a close
- Quite smooth, but not without issues
- Lots of modernization work ahead
- Some strategic issues still exist
- Qt6 helps KDE achieve its goals

Qt6 ~~will~~ has fixed it!



Questions?