

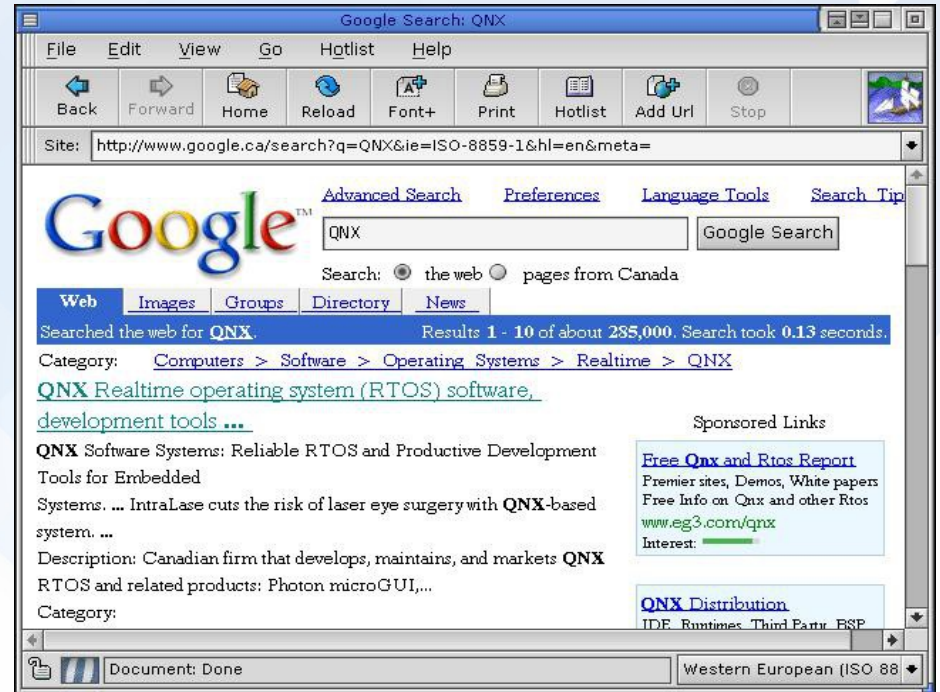
# Porting applications to Qt



Kevin Funk, Software Engineer  
KDAB

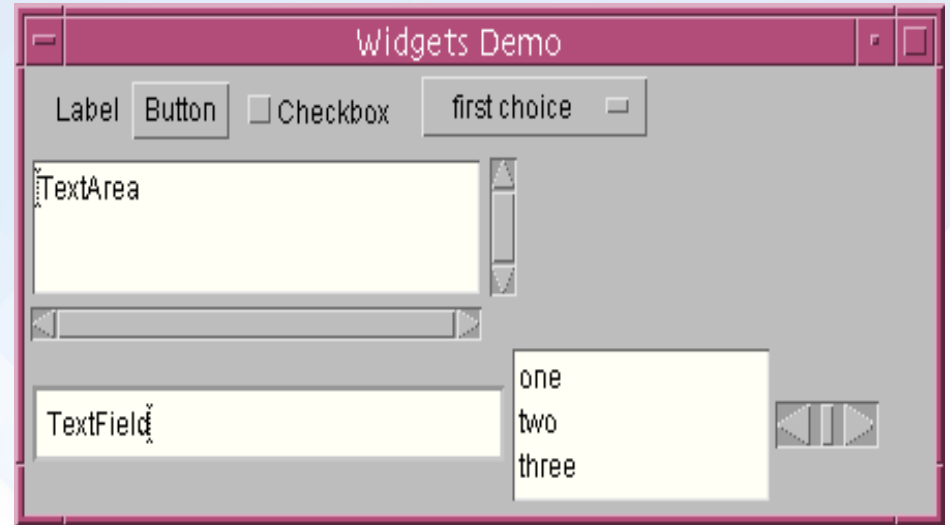
# What is a migration?

- Some other toolkit → Qt
  - QNX Photon
  - Motif
  - MFC
  - Java AWT
  - ...
- Older Qt version → Qt5



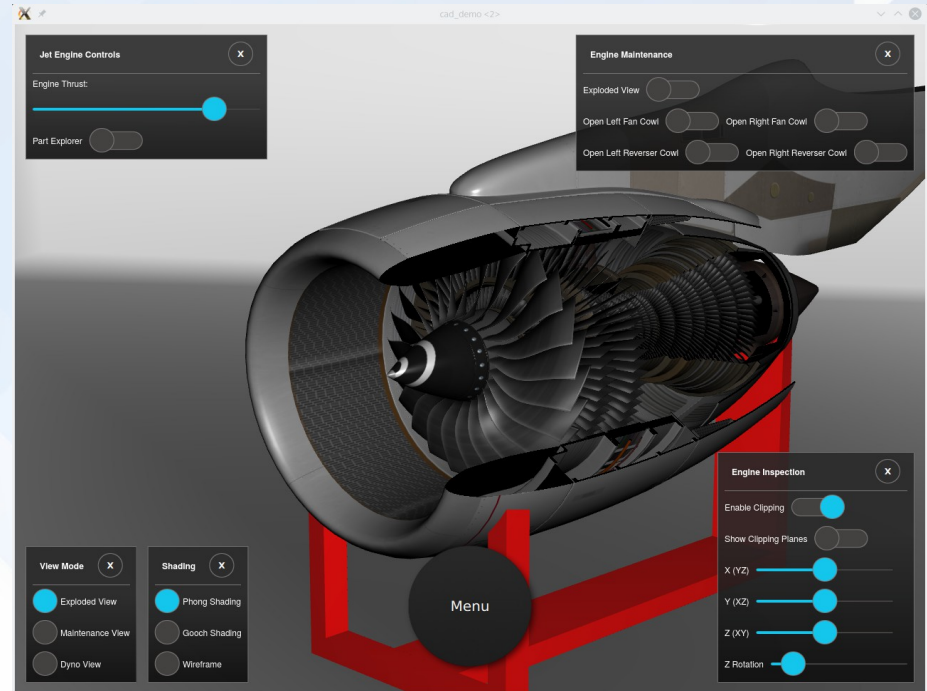
# Why migrate at all?

- Hard to find developers who know the legacy toolkit
- Legacy toolkit is not actively developed any more
- Applications have an outdated look-and-feel
- Support for more platforms
- Job satisfaction - it is easier to attract good developers to work on something cool than something outdated



# Qt Framework to the rescue!

- Actively developed C++ framework
- Under Open-Source Governance model
- Cross-platform
- Deployable to embedded devices
- Used in over 70 types of industries world-wide



# What is a migration?

- Migration techniques
  - Throw out all code, just copy the basics over.
  - Port code gradually using framework integrations
    - your app will be part Qt & part legacy toolkit (think of MFC)
    - Note: Not suitable for production code!
  - Comment out all code that doesn't compile on the target platform, and port it bit by bit.

# What is a migration?

- Potential tasks
  - Changing a legacy toolkit push button to a QPushButton.
  - Reworking mechanism like callbacks to signals/slots.
  - Converting code for painting and color allocation.
  - Rewriting usage of third party libraries, like
    - Custom JSON parser library
    - ...



# What is a migration?

- More potential tasks :(
  - Replace your own implementation of what Qt now supports natively (XML/Multi threading/File support)
  - Replacing unsupported third party tools like charting engines with new or custom made components.
  - Take advantage of the target frameworks capabilities like model/view or Qt designer to improve overall design quality.
  - Now think *we have a few million lines of code*, and understand the complexity!

# What is a migration?

- Some of the tricky parts :((
  - Should you use *char \**, *QString*, or your own string implementation – including Unicode support?
  - Legacy toolkit may have been plain C, but you likely had a C++ wrapper layer, how do you port that?
  - You'd like to keep your custom style of the application – how to bring to the Qt world?
  - User interface description files of legacy toolkits – e.g. how to port QNX Photon's *.wgt* files?



# What is a migration?

- Issues to consider
  - Which build system to use - *QMake* and *CMake* are both good candidates
  - It might be a good time to update your naming convention
  - It might be a good idea to update to a newer C++ standard
  - Still using *CVS*? Consider upgrading to *SVN* or *Git*
  - Peer reviews may help you towards a consistent port – needs experts in the target framework

# What is a migration?

- Automatic Conversions
  - A one-to-one mapping is likely only possible for the simplest part of a port.
  - Improve your editors to support you while porting
    - Code snippet plugins
    - Stored RegEx-based replacement routines
  - Use scripting!
  - Popular these days: **Clang tooling!**

# What is a migration?

- Pitfalls
  - **Feature Frenzy** - It is more fun to add a new feature than doing porting, so you risk you will never get done
  - **Refactoring/Clean-up Frenzy** - when porting - it isn't the right time to clean up(!)
  - **Bug fixing Frenzy** - when you have a new look at the code you may find new bugs, or new ways to fix old bugs. Be careful not to drift away in a bug fixing frenzy where you fix a bug that everyone has accepted is there.

# What is a migration?

- General recommendations
  - Remind **management** why you want to port - don't push for new features.
  - Remind **developers** why you want to port - stay with what is the default in Qt (colors, fonts, ...) – *for now*
  - Remind **customers** why you want to port - this is the hardest one!

# KDAB Migration

- What we offer as part of a migration
  - free in-depth analysis report if you decide to use us for the project
    - Using our own set of code analysis tools
    - Reviews by our Senior engineers
  - porting of your code base from another toolkit or older Qt version to an up-to-date Qt version
    - Using both proprietary and our open-sourced tools
    - In collaboration with the customer via code reviews, meetings
  - delivering a ready to ship version of the complete source code

# Thank you!



- KDAB Services
  - Consulting
  - **Customer projects**
  - ...
- KDAB Trainings
  - QML for Embedded Linux
  - Debugging & Profiling
  - C++11 and C++14
  - Modern OpenGL
  - ...

Visit our booth for demos!

Contact: [kevin.funk@kdab.com](mailto:kevin.funk@kdab.com)