



DEVELOPER  
DAYS **2014**  
EUROPE

## Why Qt Matters in the Big Picture

**Till Adam**  
**KDAB**

**[till.adam@kdab.com](mailto:till.adam@kdab.com)**



# The Big Picture



- Interesting Times
- Desktop
- Embedded
- Mobile
- Risks
- Opportunities





Major changes are in motion in technology:

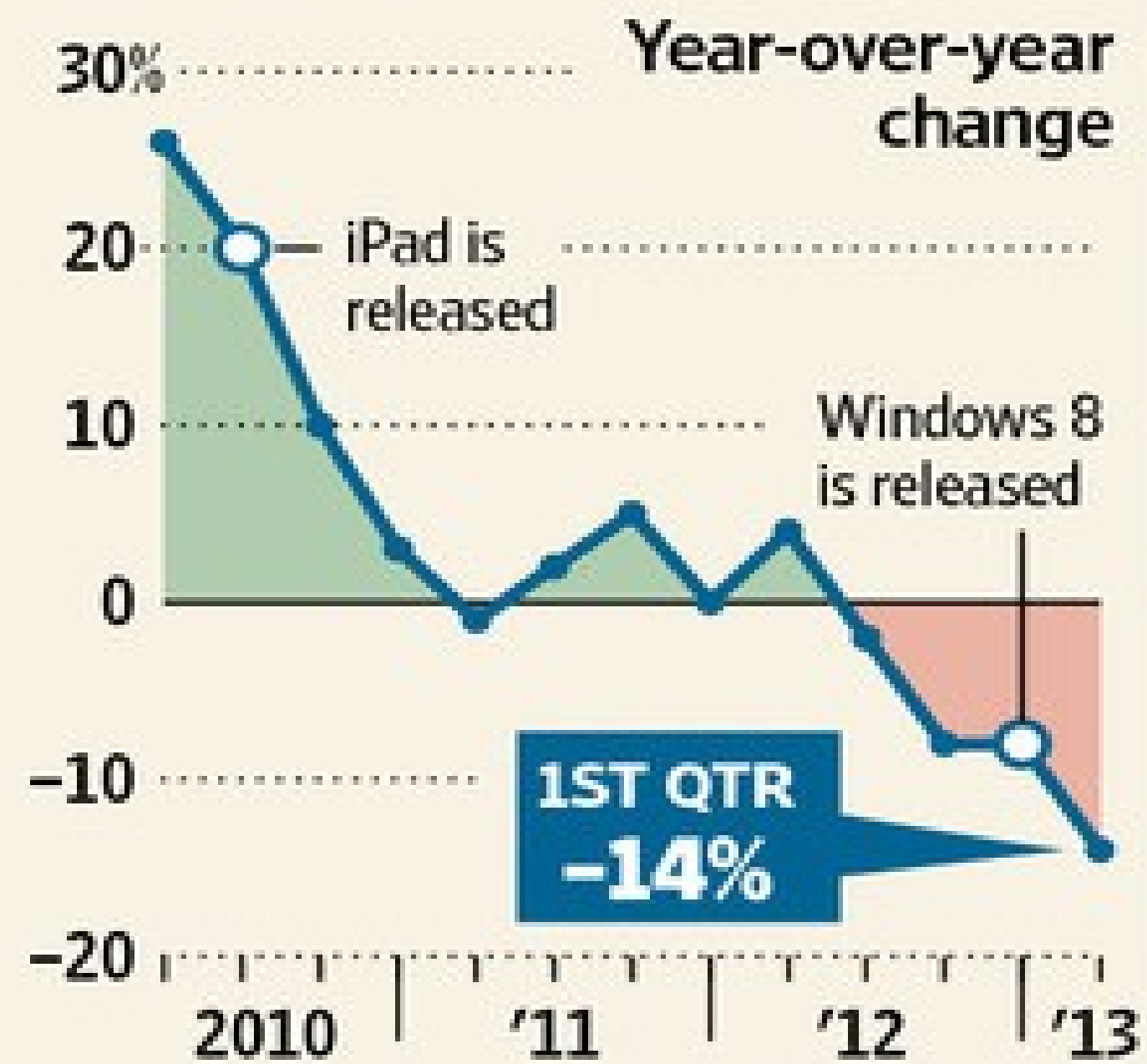
- Mobile Revolution
  - Economies of scale
  - Mobile first
  - Consumer expectations
- Cloud (ubiquitous connectivity)
- BRIC (2013: 41.3% of total global population and 20.2% of total global GDP)
- Internet of Things / M2M





## PC Meltdown

The slide in global shipments of desktops and laptops accelerated last quarter.

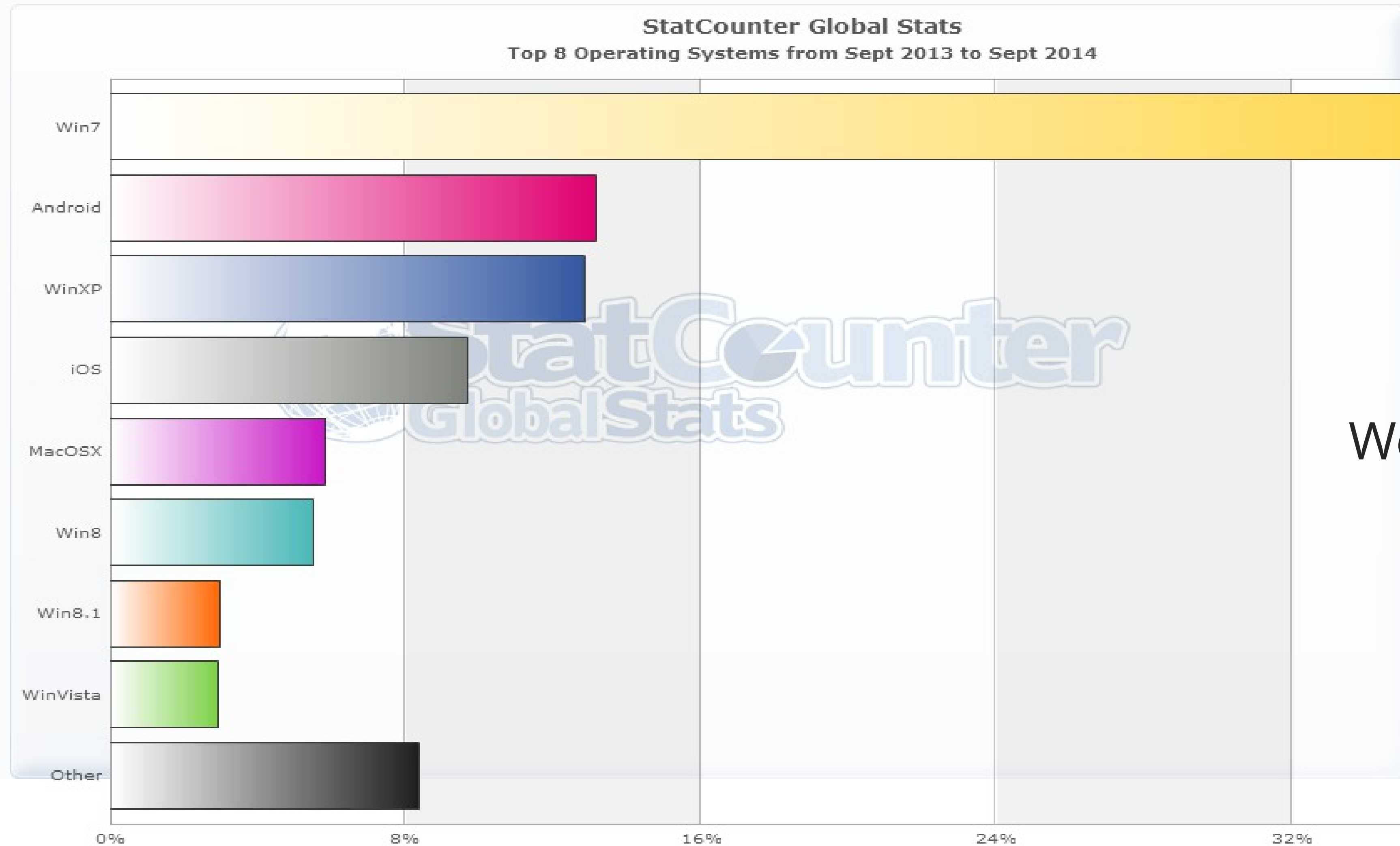


Source: IDC The Wall Street Journal



It's dead, dead, dead!





Well, not quite.







Most Qt Developer Days Attendees develop for the desktop (and also for other platforms).





.NET/Silverlight/XAML/WindowsTechDuJour:

- Windows only (more or less, there is mono)
- Desktop not a focus
- Big but powerful





## Mac OS X Cocoa and Carbon:

- Mac only
- Desktop less of a focus
- Big but powerful







Misc. obsolete toolkits:

- MFC
- Motif
- Delphi
- GTK+
- FLTK
- Swing, AWT
- WxWidgets
- Flash/Air/Flex

=> none of them even close to feature parity with Qt





Raw OpenGL (GLUI/GLUT, VTK, Clutter)

Home grown special purpose toolkits

=> primitive tooling, maintenance nightmare





Web technologies:

Jquery, Sencha, Apache Cordova, ...

=> unfit for the demands of "real" desktop use cases





Qt is the only feature rich, modern, native, cross platform solution. For advanced UI needs, efficiency with large data volumes, high performance demands or single-source approaches there are no viable alternatives.

This is unlikely to change as no big player is investing in the desktop.





All popular embedded operating systems provide Qt as their primary UI option (Linux, QNX, Wind River VxWorks, Green Hills Integrity, Microsoft Windows Compact) usually combined with OpenGL and HTML5. Hardware accelerated, GPU based rendering is key (OpenGL ES2), as are reasonable disk, RAM and CPU footprint and thus power consumption.

Yes, even Microsoft encourages Qt and its partners are Qt partners (Adeneo, Toradex).







It's a two horse race, why not just write native iOS and Android apps?





Code sharing between iOS and Android is nice, but not the point.

Code sharing between desktop, embedded and mobile is worth it.

And then there is Blackberry, Windows Phone, Jolla, Ubuntu Phone, Tizen, ....





- Risks for Qt
- Risks of Using Qt
- Non-Risks of Using Qt





- Lack of Standardization
- Lack of Coherent and Comprehensive Tooling
- Is the funding of Qt development sustainable?
- Non-technical platform barriers (app store rules, patents, managed-code-only)
- The next Microsoft or Apple platform might not be possible to support





- Talent availability
- Limited 3<sup>rd</sup> party component ecosystem
- QML is dangerously and deceptively easy
- Blind trust in the blackbox
- Technology mismatch with web side







- Vendor lock-in
- Platform lock-in
- Getting stuck on a dead toolkit (Delphi, Motif, ...)
- Monoculture
- Strategic dead ends (MFC, .NET, ...)





Data and functionality moves to a central place, away from a single machine or device and becomes accessible from many endpoints.

- => cross platform becomes more important
- => strong networking capabilities are key
- => local state keeping and native approaches enable good usability





Internationalization, localization, high quality font rendering, input method integration, RTL support and UI flexibility make it possible to target emerging growth markets with great diversity.

A global community helps avoid euro-centric or america-centric biases.

Community can support languages or scripts that are too small to be commercially viable to support.





Qt can uniquely reach onto almost any device and platform and allows to build complex connected solutions quickly and securely. It can reach all regional markets. It thus has the potential to become the standard middle-ware and UI solution for the next few billion devices coming online.





- Qt is strong on the desktop and very strong on embedded.
- As those category silos fall, mobile gives Qt additional reach.
- Qt is well positioned to take advantage of large scale developments around cloud, the internet of things, global growth and the rise of consumer mobile devices.







Questions?

