




Costly mistakes IoT developers make and how to avoid them

Tino Pyssysalo, Senior Manager
The Qt Company

Embedded Conference Finland

The Qt Company – Solutions for Embedded SW

 **>300**
Professionals
in ten countries


 **>1M**
Developers

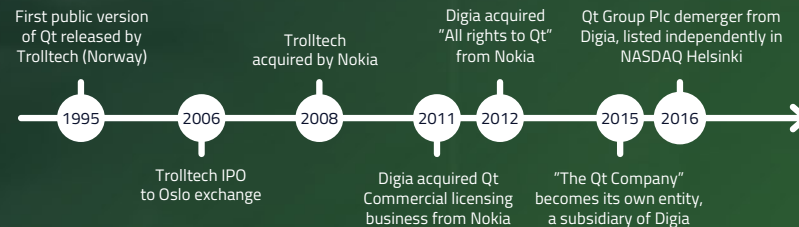
 **22** years
company history




 **>5000**
Customers

 **>70**
Industries

 **>30**
Cross-platform SW
libraries for desktop,
mobile, and embedded
UI + engine



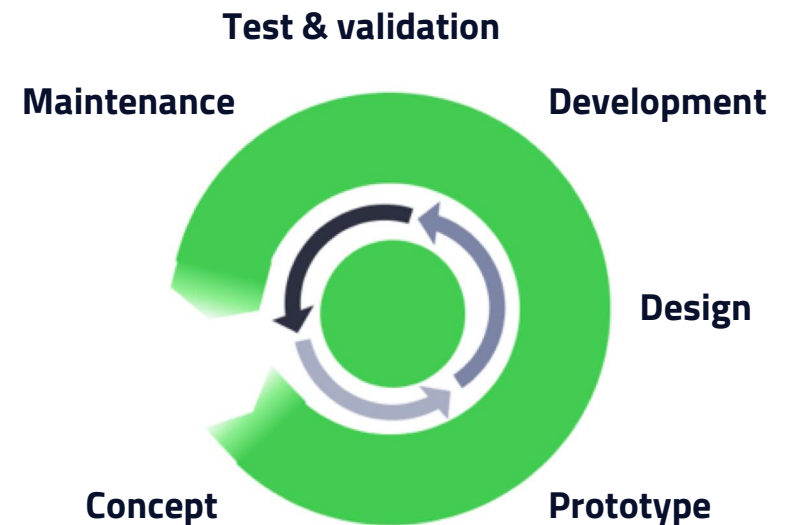
Mistakes Increase Time to Market and Total Cost of Ownership



TTM
Last 10% of features take 80% of the dev time



TCO
The later a mistake is found, the more costly it is to fix it



The Qt Company findings about common mistakes in customer projects over the years



Insufficient IoT Technology Evaluation

- > Evaluation based on incomplete requirements
 - > Memory footprint compared, but run-time memory consumption ignored
- > Chosen technologies are challenging to mix
 - > SW requires accelerated graphics
- > Lack of community
 - > Missing support
 - > Dependency on one partner

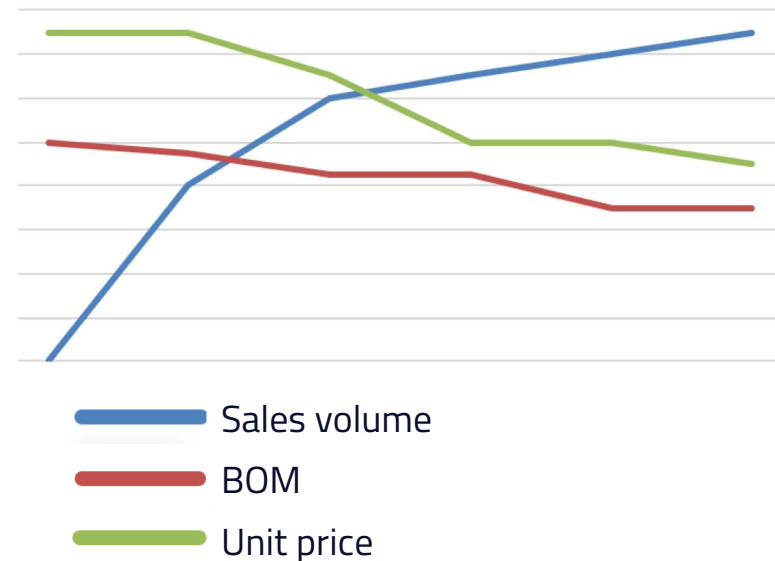
- > Choosing OSS solutions without understanding the obligations
 - > IPR and patent protection
 - > OSS license incompatibilities
 - > Static builds
- > Immature or fast evolving technology
 - > Technology life-time
 - > Fixing problems affect TTM and TCO

Technology evaluation based on a defined and known requirements

Too Optimistic Rightsizing

Lower BOM => bigger margin

- › SW designed independently of a chosen HW
 - › HW often chosen before SW exists
 - › HW requirements based on assumptions: IoT SW is simple, small flash memory sufficient
- › Upgrading HW increases BOM costs
 - › Larger flash or RAM
 - › i.MX7 => i.MX6
- › SW developed in the host and validated in target HW late in the dev project



Early prototyping and early fails – even in the emulator with adjustable parameters

Lack of Maintenance Plan

- › After deployment, critical bugs may need to be fixed
- › Secure updates
- › Wired vs. OTA solution
- › Firmware and application level updates
- › On-the-fly updates



Maintenance requires a clear plan – sell and forget won't work

Underestimating Development Effort to Multiple Targets

Several code bases

- > Several development teams
- > Larger tooling investments
- > Multiplied expertise needed
- > Multiplied effort to manage device BSPs and toolchains
- > Increased maintenance costs

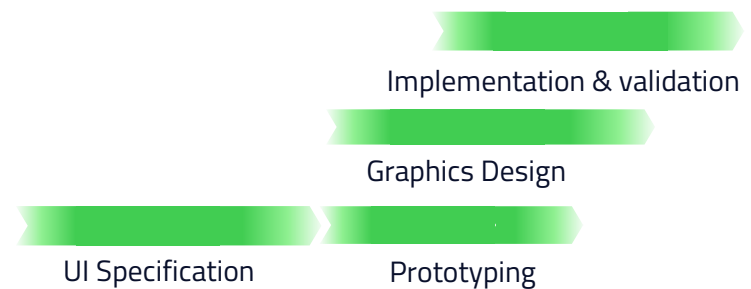
Scalability

- > Low-end and high-end devices
- > Memory footprint
- > Run-time memory usage
- > Multiple screens with different sizes, resolutions, pixel densities

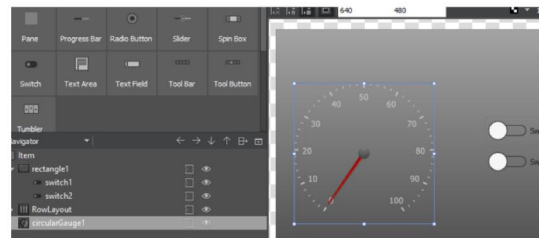
Cross-platform tools and libraries
Know the real HW requirements early in the project
On-target debugging
Scalable technology

IoT Gateways – Hick-ups in UI Design

- › Lack of workflow and tools between UI designers and developers
 - › Incomprehensible UI requirements
 - › Long feedback cycles in both directions
 - › UI design ignores SW and HW capabilities
- › UI prototyping is time-consuming
 - › Prototypes cannot be used in the real product
- › User experience ignored
 - › Design looks beautiful, but UI transitions do not work
 - › Too late UI validation in the target HW



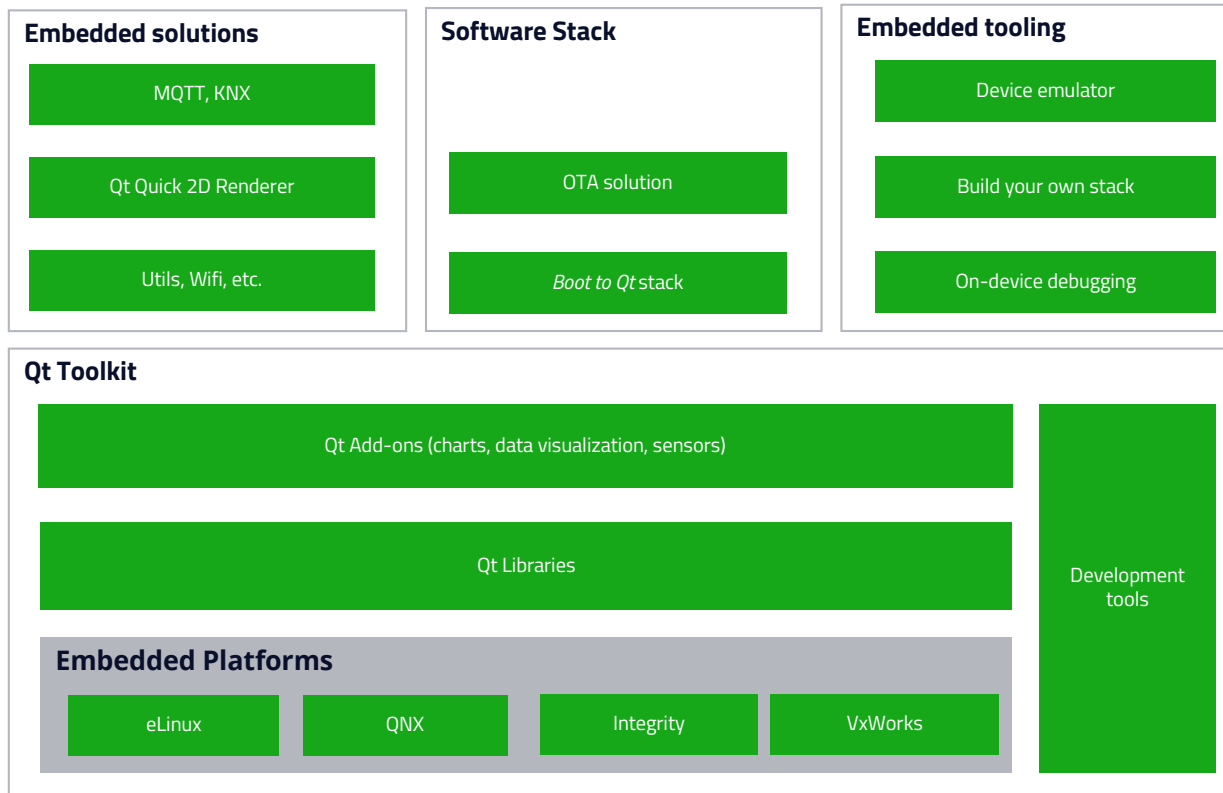
WYSIWYG editor



Target validation



Qt Platform for IoT Devices



IoT Sensor Tag demo



Home Automation demo





Thank you!

Visit our booth in the exhibition hall!

tino.pyssysalo@qt.io

