



# Meet Qt Canada

17.6.2021

MEET QT

# Agenda

- Welcome to Meet Qt
- Qt Roadmap
- Introduction to froglogic
- Qt Design Studio
- Qt Professional Services
- 6** ■ Live Q&As



Qt

# Introduction to froglogic

# froglogic: a history

- › Harri Porten & Reggie Stadlbauer, former Qt devs
- › Founded in 2003
- › Focus on QA tools
- › Side-by-side with Qt since

# froglogic: a (modern) history

- › Moving forward: more tech support added to Squish
- › New tools to support QA: Code coverage and results management
- › Customers worldwide & in diverse industry contexts
- › Acquired by The Qt Company in April 2021

# Who needs GUI test automation, anyway?

- › Testing GUIs and HMIs have been historically time consuming & error-prone
- › Touch-screen devices are ubiquitous
- › Automation:
  - › Significantly reduces time to run tests
  - › Enables more tests to be run
  - › Allows tests to be run more often
  - › Detects regressions sooner

# Dedicated Qt Support

- › Qt6.x to Qt3.x
- › Qt Widgets, QML & Qt Quick
- › Standard & complex controls
- › Qt + Windows, Web
- › Custom Controls



Test Suites

suite\_py

Test Cases

- tst\_general
- tst\_adding**
- tst\_adding\_data

Test Case Resources

Scripts Test Data VPs Gestures

test.py

Test Suite Resources

Scripts Test Data VPs Gestures

- \_pycache\_
- names.py

test (tst\_general) tst\_adding (suite\_py) test (tst\_adding\_data)

```

1 import names
2 import os
3
4 def invokeMenuItem(menu, item):
5     mouseClick(waitForObject({'type': 'MenuItem', 'text': menu}))
6     mouseClick(waitForObject({'type': 'MenuItem', 'text': item}))
7
8 def addNameAndAddress(oneNameAndAddress):
9     invokeMenuItem("Edit", "Add...")
10    type(waitForObject(names.address_Book_Add_Forename_Edit), oneNameAndAddress[0])
11    type(waitForObject(names.address_Book_Add_Surname_Edit), oneNameAndAddress[1])
12    type(waitForObject(names.address_Book_Add_Email_Edit), oneNameAndAddress[2])
13    type(waitForObject(names.address_Book_Add_Phone_Edit), oneNameAndAddress[3])
14    type(waitForObject(names.address_Book_Add_Phone_Edit), "<Return>")
15
16 def closeWithoutSaving():
17     invokeMenuItem("File", "Quit")
18     clickButton(waitForObject(names.save_Table_No_Button))
19
20 def main():
21     startApplication("'" + os.environ["SQUISH_PREFIX"] + "\\examples\\win\\AddressbookWPF\\AddressbookWPF.exe"')
22     table = waitForObject(names.address_Book_Table)
23     invokeMenuItem("File", "New")
24     test.verify(table.rowCount == 0, "On startup, the address list is empty.")
25     data = [
26         ("Andy", "Beach", "andy.beach@nowhere.com", "555 123 6786"),
27         ("Candy", "Deane", "candy.deane@nowhere.com", "555 234 8765"),
28         ("Ed", "Fernleaf", "ed.fernleaf@nowhere.com", "555 876 4654")
29     ]
30     for oneNameAndAddress in data:
31         addNameAndAddress(oneNameAndAddress)
32     test.compare(table.rowCount, len(data),
33                 "After adding {} addresses, the table contains {} rows!".format(len(data), table.rowCount))
34     closeWithoutSaving()
35

```

Test Results Runner/Server Log

Result	Message	Time
TestSuite	suite_py	May 17, 2021 9:23:34 PM +02:00
Test Case	tst_adding	May 17, 2021 9:23:34 PM +02:00
Pass	Verified: On startup, the address list is empty.	May 17, 2021 9:23:38 PM +02:00
Detail	True expression	
Pass	Comparison: After adding 3 addresses, the table contains 3 rows!	May 17, 2021 9:23:45 PM +02:00
Detail	'3' and '3' are equal	



# Coco: a code coverage toolchain

- › Testing blindly or testing smarter?
- › With Coco:
  - › Identify untested code
  - › Integrate into a/typical or generic test frameworks
  - › Eliminate redundant tests
  - › Optimize test execution order
  - › Patch analysis
- › Other features:
  - › CI system integrations
  - › Function Profiling
  - › Blackbox testing
  - › Tool Qualification Kits for safety-critical applications
- › Squish GUI integration

```
validCCN({ 4, 5, 3, 9, 1, 4, 8, 8, -3, 1, 4, 3, 6, 4, 6, 7 });

1 #include <vector>
2
3 // Checks if the given credit card number is valid or not
4 bool validCCN(std::vector<int> number)
5 {
6     if (number.size() < 2) return false;
7
8     // Double every second digit, start from the back
9     for (int i = number.size() - 2; i >= 0; i -= 2) {
10        int digit = number[i];
11        if (digit < 0 || digit > 9)
12            return false;
13        digit *= 2;
14        if (digit > 9)
15            digit -= 9;
16        number[i] = digit;
17    }
18
19    // Calculate sum of all digits
20    int sum = 0;
21    for (int i = 0; i < number.size(); ++i)
22        sum += number[i];
23
24    // The credit card number is valid if the sum is a multiple of 10.
25    return sum % 10 == 0;
26 }
```

# Coverage Metrics

- › Function
- › Line
- › Statement
- › Decision
- › Condition
- › MC/DC (Modified Condition/Decision Coverage)
- › MCC (Multiple Condition Coverage)

# Test Center: A Test Reports Management Platform

- › Lots of tests = lots of test results
- › Results need to be stored and shared centrally
- › Lots of factors involved: multiplatform scenarios; branches, etc
- › Connect results to requirements/test management and issue reporting/tracking

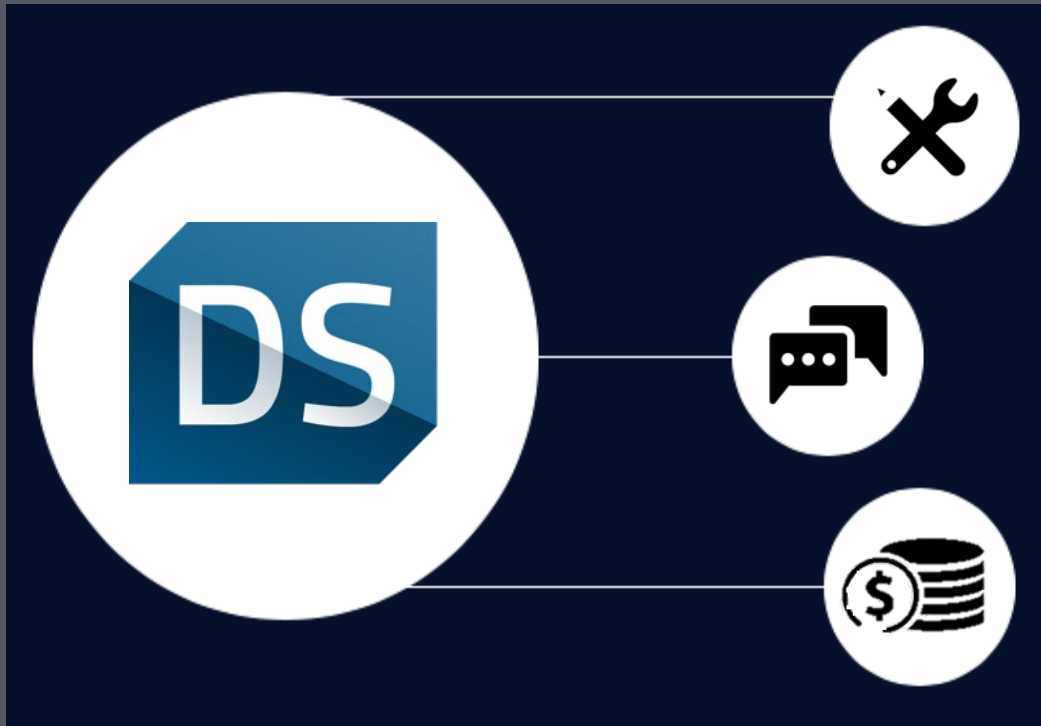




# Qt Design Studio

# What is Qt Design Studio?

*A tool for both **designers & developers** to make collaboration simpler and streamlined*



Design beautiful experiences all the way from early wireframes to final pixel-perfect implementation

Rapid, iterative and incremental prototyping to validate the designs with target device

Bridging the gap between design and development with the unified toolchain

# Qt Design Studio

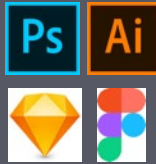


## Qt Design Studio 2.0 Feature Overview

- › **Visual 2D editor**
  - › Import designs from Photoshop, Sketch, and Figma
- › **Visual 3D scene editor**
  - › Import assets in, e.g., such as .blend, .dae, .fbx, .glb, .gltf, and .obj formats
- › **Advanced code editor**
- › **Flow editor with customizable transition effects**
- › **Dynamic behaviors**
  - › Connection editor
  - › State editor
  - › Transition editor
  - › Binding editor
  - › Logic helpers for binding property values, and for mapping numbers and numerical ranges
- › **Keyframe-based timeline animations**
  - › Ready-made and customizable easing curves
- › **Dynamic and scalable layouts**
- › **Ready-made customizable UI components**
- › **Built-in and customizable visual effects**
- › **Live Preview designs on target devices**
- › **Design annotations**
- › **Improved MCU support**
- › **Simulink support**

# Qt Design Studio from wireframes to final implementation

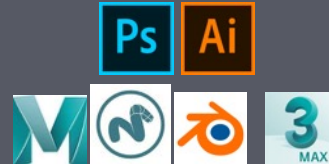
**Wireframe**  
Component level



Bridge

**Prototype**  
Interaction level

**UI mock-up**  
Style level



Import

**Product UI**  
Implementation

DS

Unified 2D/3D design tool for all phases

## Wireframing with QDS

- Create 2D & 3D UIs
- Built-in, ready to use & customizable components
- Scalable layouts

## Prototype with QDS

- Simulate complex experiences
- Full control on dynamic behaviors
- Validate with target HW
- Import designs from prototyping tools

## Motion design with QDS

- Production quality, detailed motion design
- Imports from content creation and prototyping tools
- Optimization with target HW

Asset creation with content creation tools

## Implement with QDS

- Cross-platform
- Reusable prototypes and full UI implementation
- Less need for spec writing & maintenance
- Fast parallel development
- One toolchain with QDS & Creator





# Qt Professional Services

# Qt Professional Services

We're a think tank of subject matter experts and practitioners on a mission to build your world with Qt.

## Workshops

A deep dive into technology considerations, user experience design, software architecture and best practices. We eliminate risks early by advising on best Qt practices.

## Custom Services

We deliver software design and implementation services to enhance the solution and end-to-end project lifecycle management from idea to the product phase-out.

## Training Services

With expert trainers that are experienced practitioners, we move beyond theory, providing valuable high impact training that outlines specific business goals.

## Unique Support

As the owner of Qt, we own the IP. This gives us the unique ability to provide the ultimate in customer support experience as we can provide a direct relationship with our R&D organization.

Minimize  
Total Cost of Ownership

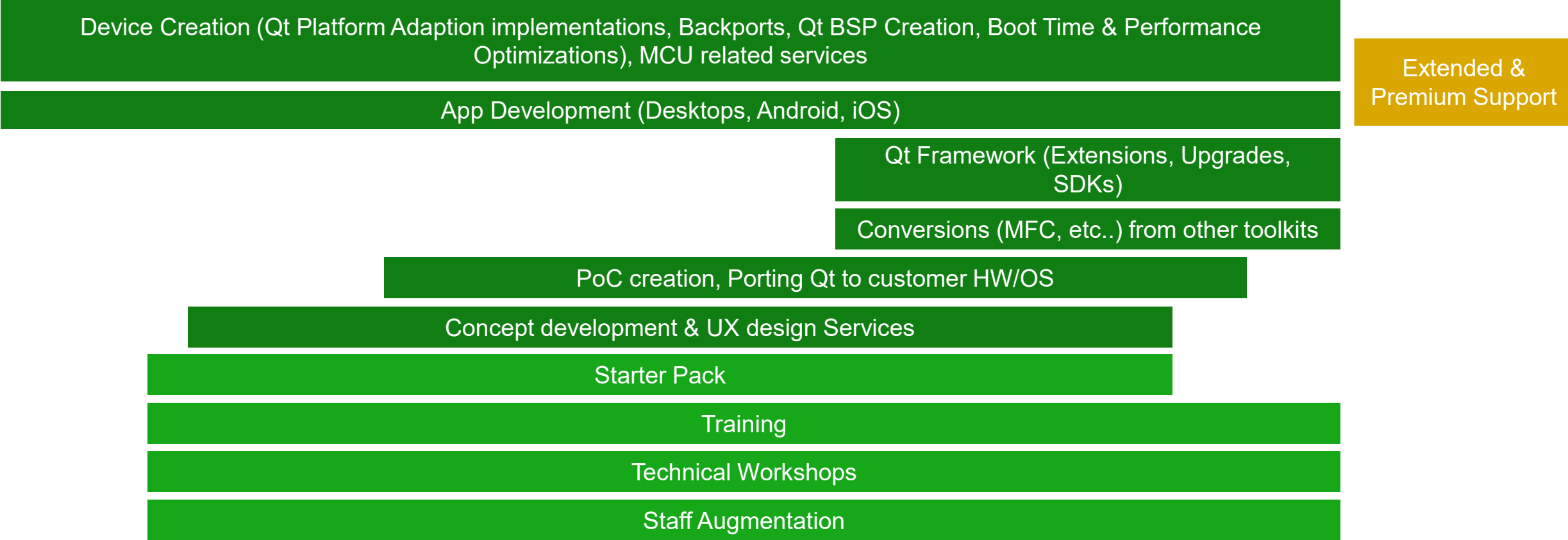
Mitigate Technical and  
Security Risks

Shorten  
Time to Market

Optimize  
Performance

Harness the Power  
of Qt Ecosystem

# The Qt Company Professional Services



# Training Services

## Standard Courses

- › Qt Essentials
- › Qt Quick and QML
- › Qt Application Engine Development
- › Qt Widgets
- › Qt Embedded Development
- › Qt Mobile Development
- › 3D in Qt

## › Open Enrollment

- › Scheduled open seat training
- › Provided by certified partners

## › Custom Onsite Training

- › High impact training that outlines your specific goals
- › Adjusted to all levels of expertise

# Professional Support

Working closely with R&D teams, we help developers address complex technical challenges

## Standard Support\*

- › Support on Qt APIs, functions, methods, and programming techniques
- › 48 hours response time
- › Bug fix prioritization

\*Support granted to Qt license holders

## Premium Support

- › Dedicated support resource
- › 24 hours response time
- › 50 or 100 hours at a fixed price
- › Support for your specific software solution
- › Direct online access

## Extended Support

- › Enable long term maintenance and Qt upgrade opt-outs
- › Longer term support for Qt 5.x releases
- › Qt 4.x releases supported directly by the Services team

# What you can only get through The Qt Company's Professional Services?

- › **Direct access to Qt R&D** -> Qt Professional Services is the only consulting team in the World that has direct access to Qt R&D, to skills/knowledge of the 100+ Experts working in Qt R&D and all the new features that are still under planning/development
- › **Get the best out of the whole Qt ecosystem** -> Qt Professional Services is the only team that has access to the whole ecosystem globally (all the skills, competences, knowledge and resources) and can provide you a project with the best Experts from Qt Professional Services team, service partners, subcontractors, technology partners, freelancers, etc.
- › **Support, maintenance and continuous Integration** -> you can have your Qt based product maintained and supported along with the whole Qt product family and choose to have continuous integration service that your software is always integrated to the latest version of Qt and ready for further development and testing
- › **Global reach** -> we can help you globally with Qt technology and to implement great products by providing international software development projects, consultation, advisors, workshops and trainings

# Q&As

> Ask your questions



# Thank you for joining us!

[info@qt.io](mailto:info@qt.io)

[www.qt.io/contact-us](http://www.qt.io/contact-us)

