CASE STUDY

A Disciplined Approach to User Interface Development Using Qt (QML) for Medical Devices

LAYERED SOFTWARE PLATFORM WITH ROOM TO GROW

Project Snapshot

SITUATION

Two leading global MedTech companies needed an ISO-certified software firm to build GUIs for medical devices using QML within the Qt framework. Program managers at a diagnostic company ("Client A") needed a proof-of-concept GUI in order to move through a "funding gate" and maintain project momentum on the company's first new blood gas analyzer in a decade. Global healthcare leader ("Client B") required full application development of a GUI for a wireless heart monitor.

CHALLENGES

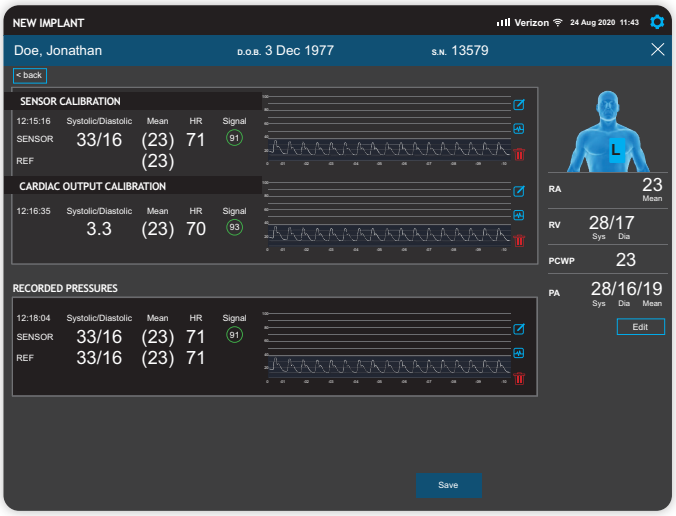
- Given the complexity of these medical devices, they required customization above-and-beyond Qt's standard widget library
- In both cases, internal client teams lacked the expertise to develop the Qt-based GUI using QML with enough precision and speed to ensure high-quality code

SOLUTION

- A layered approach to compartmentalize the code
- C++ code to talk to the back end, Qt as the middle control layer, QML for the GUI layer
- Meticulous code tailored to clients' needs with reusable components made of 100% custom widgets written in QML

RESULTS

- Well-constructed code can easily be updated as requirements change in the future
- Stakeholders for Client A successfully demonstrated the proof-of-concept to senior management, obtaining a "green light" to move forward with the project
- GUI for Client B's heart rate monitor exceeded expectations with custom features and style



Two leading manufacturers needed custom QML code to develop tailor-made user interfaces for new medical devices.

INITIAL ISSUES THAT NEEDED CORRECTION

Both clients needed ISO-certified engineers with high proficiency working in QML within the Qt framework to create GUIs for medical devices on very tight timelines. The devices required custom elements that exceed those in a standard Qt widget library.

Client A urgently needed a proof-of-concept of their GUI built using QML. It had to function well enough and impress senior executives to gain their approval to move forward with the new development effort. In order to protect the investment and schedule, the client mandated that this could not be "throw away code." Further, in order to identify the resources that the client would use for ongoing GUI development, both MedAcuity and an internal client team were given the same task so the client could assess the quality and timeliness of each team's deliverables.

Client B was developing a next-generation GUI for an implantable device with a controller outside the body. They approached a Qt representative who suggested MedAcuity as a "go-to partner" with best-in-class ISO 13485-certified software development expertise and a track record of speed and quality.

medacuitysoftware.com
866.376.1931

MEDACUITY
MedTech Software. We live for this.®

MedAcuity used Qt to deliver a GUI that appears and operates like the UX design and can communicate seamlessly with the client's existing back-end code. The solution has custom components that can be reused as requirements change in the future.

HOW MEDACUITY TURNED THE SITUATION AROUND

MedAcuity's engineers leaned on decades of experience developing GUIs with numerous widget toolkits to bypass common mistakes that inexperienced developers can make—mistakes that can lead to unstructured and difficult-to-maintain code. They used a disciplined approach to write layered code and developed a model-view-controller paradigm to separate each purpose. By keeping C++, Qt, and QML isolated, they ensured the robustness and flexibility of the application. Qt served as the middle layer to get patient data through the client's API to the GUI on the screen.

MedAcuity's specialists also created reusable base components for buttons, text fields, and sliders. Then they built an application from those components without overloading them with excessive capabilities that could cause problems in the future—a common mistake if you're not experienced working with QML. MedAcuity also included an automated test suite to complete the end-to-end software development lifecycle.

RESULTS

The medical devices now have GUIs with full capabilities built on structured code that is built for scalability.

Client A scratched their internal team's unsuccessful attempt and brought MedAcuity's custom-tailored GUI to the executive team instead. The project was approved. They are now ahead of schedule for developing the device because the initial code is solid enough to take further into development.

> ***"The work was done at a critical moment, the value was high, and the project was managed well toward meeting our goals. They provided solid solutions and added 'bells and whistles' appropriately, which were appreciated."***
>
> *- Program Manager, Client A*

Client B received a GUI with all the features and style they wanted; even bespoke elements such as a moving waveform that can zoom and scroll on a grid.

> ***"Impressive, since this was based on requirements that weren't fully developed."***
>
> *- Sr. Manager of Program Management, Client B*

Accelerating the pace of innovation while reducing development time and risk.

It's possible. *Ask us how.*

866.376.1931

medacuitysoftware.com

MEDACUITY
MedTech Software. We live for this.®