	The Qt Company	**Comments**	**RightWare**
	Qt		**Kanzi**
HMI Evaluation			
Target Ent. Unit Level (Base, Mid, High)	most OS and any SoCs which support OpenGl ES 2.0		most OS and any SoCs which support OpenGl ES 2.0
Source code availability	YES		No
Technical support	Direct vendor support from Shanghai office or supported by partners in China.		Support by partner
Performance optimization / Quality	Provide toolings + Open source. You can modify and optimize as you need. Also Qt can provide support.		Kanzi LIBs are blackbox. Need Kanzi HQ engineer's support to optimize
Expandability	Able to setup a unified HMI Dev environment for car (IVI, Cluster, HUD, RSE, HVAC control and more)	Qt is simple and easy to use and develop re-usable, extensible and configurable solutions that are essential for IVIs and Clusters.	No
Ecosystem / Engineer availability	Abundent. Many 3rd parties and engineers available in lcal	Qt framework is currently widely used in various spheres, the automotive industry being among them. The Qt community is called home by many developers, be it a beginner or an experienced C++ specialist. Qt Project offers great community support while Qt Partners offers commercial support at any level.	Very Poor. Hard to find experienced engineers
General			
Learning Curve	Short (QML is easy to learn)	QML, a JavaScript-based declarative language. It is a part of Qt Quick, which provides a way of building dynamic UIs with fluid transitions and effects. It is a highly readable, declarative, and easy to learn the language.	Long (Takes much more time than Qt)
Rapid Prototyping	Yes	Tool chain supports for rapid prototying from design to target deployment With Qt/QML, build a structured prototype, constantly refine it, and evolutionary prototype can become the product. Qt/QML provides faster round trip time, simplified user interface, easy export of assets, easy access to data, verifiable on hardware, and openness to evolve to meet raid trends in Automotive.	Yes
Developer Community Support	Yes	More than 2M Qt deployments a year in China	No
Concept to Deployment tool chain	Yes	Tool chain supports for rapid prototying from design to target deployment	Yes
2D Graphics editor	Yes	Qt Quick Designer	Yes
3D Graphics editor	Yes	Qt 3D Studio Scene editor	Yes
Animation editor	Yes	Qt Quick Designer	Yes
Windows Toolchain	Yes	Supports Windows/MAC/Linux	Yes
Open Source Toolchain	Yes	Qt LGPL v3.0 available	No
Simulation mode	Yes	Qt Creator has emulator/simulator. Qt for Device Creation Emulator enables you to test and debug applications in an environment practically identical to that of the device, without having to own a device and connect it to the development host. In addition, you can use the emulator controls to change the information that the device has about its configuration and environment. This allows you to see how applications look and function in different situations. For example, you can view the application layout in both landscape and portrait orientation. Or you can check how your application behaves when device battery power decreases to low or critical level.	Yes
Debugger	Yes	Qt Creator has embedded debugger. Qt Creator provides a debugger plugin that acts as an interface between the Qt Creator core and external native debuggers such as the GNU Symbolic Debugger (GDB), the Microsoft Console Debugger (CDB), a QML/JavaScript debugger, and the debugger of the low level virtual machine (LLVM) project, LLDB.	No
Code Generation	Yes	Qt Creator and Qt quick designer	No
Widget toolkit	Yes	Qt Creator has widget tooling. Qt Designer is a powerful tool for interactively creating and arranging widgets in layouts.	Yes
Widget development	Yes	Qt Creator. Qt Designer's plugin-based architecture allows user-defined and third party custom widgets to be edited just like you do with standard Qt widgets. All of the custom widget's features are made available to Qt Designer, including widget properties, signals, and slots. Since Qt Designer uses real widgets during the form design process, custom widgets will appear the same as they do when previewed.	Yes
Automotive specific support	Yes	Qt Automotive Suite (app manager, Qt IVI and more). The Qt IVI module provides C++ and QML interfaces for accessing vehicle features, and also enables implementing new IVI features. Already defined interfaces from the GENIVI alliance are exposed in the Qt GENIVI Extras module.	Yes

Model View architecture	Yes	Qt contains a set of item view classes that use a model/view architecture to manage the relationship between data and the way it is presented to the user. The separation of functionality introduced by this architecture gives developers greater flexibility to customize the presentation of items, and provides a standard model interface to allow a wide range of data sources to be used with existing item views.	No
Formal Specification	Yes	Qt Quick Designer + QML coding + SCXML (State Chart Machine)	Yes
Multi user support	Yes	?	Yes
Version Management	Yes	Qt Creator allows you to create repositories for version control systems that support local repository creation, such as Git, Mercurial, or Bazaar.	Yes
Test Support (Auto & Manual)	Yes	Qt has a test framework. Qt provides various functionality to help you develop high quality code. Qt itself has a significant number of unit tests written with the Qt Test and Qt Quick Test frameworks (autotests). Qt Test is a framework for unit testing Qt based applications and libraries. Qt Test provides all the functionality commonly found in unit testing frameworks as well as extensions for testing graphical user interfaces. Qt Test is designed to ease the writing of unit tests for Qt based applications and libraries:	No
Touch screen support	Yes	Multi-touch is also supported in Qt Quick Controls 2	Yes
Speech input support	Yes	Qt Speech - Text to speech and speech recognition functionality. Currently only the text to speech features are released. Includes backends for several speech synthesizers on Windows, Linux, macOS, and Android. Also, it can be easily integrated with 3rd party solutions through plug-in.	Yes
Multilingual input support	Yes	Qt Virtual Keyboard include customizable keyboard layouts and styles with dynamic switching, predictive text input with word selection, character preview and alternative character view automatic capitalization and space insertion, scalability to different resolutions, support for 19 different character sets (Latin, Simplified/Traditional Chinese, Hindi, Japanese, Arabic, Korean, and others), left-to-right and right-to-left input.	Yes
Hard Key Knob Support	Yes	Qt provides hardware key support for 2-way and 5-way navigation.	Yes
Gesture support	Yes	Qt provides Handwriting support, with gestures for fullscreen input.	Yes
pinch zoom support	Yes	Supported	Yes
drag and drop support	Yes	Supported. Qt drag and drop provides a simple visual mechanism which users can use to transfer information between and within applications. Drag and drop is similar in function to the clipboard's cut and paste mechanism.	Yes
HMI usability testing	Yes	Supported	No
HMI flow chart support	Yes	SCXML (State chart editor)	No
Industry standard for Linux or QNX	Yes	All linux, QNX 6.5, 6.6 and 7.0. http://doc.qt.io/qt-5/supported-platforms.html	Yes
Memory footprint	Yes (configurable)	Configurable from 20M to 5M. Qt Lite configuration options, we are able to create a more than 60% smaller deployable binary with Qt 5.9.	Yes
Min. Performance/Throughput requirement	ARM Cortex A5 (400Hz)	Without GPU	
Font Support(ttf)	Yes		Yes
OEM Tool Usage (e.g. Would OEM use the tool?)	Yes	For building OEM specification, prototyping and OEM SDK	Yes
Port availability for the Platform	Yes	Qt is OS & HW agnostic	Yes
Graphics Import	Yes (PNG Express)	For 2D PSD import supported. For 3D Autodesk (.fbx) Collada (.dae) glTF (.gltf, .glb) Blender 3D (.blend) 3ds Max 3DS (.3ds) 3ds Max ASE (.ase) Wavefront Object (.obj) Industry Foundation Classes (IFC/Step) (.ifc) XGL (.xgl,.zgl) Stanford Polygon Library (.ply) *AutoCAD DXF (.dxf) LightWave (.lwo) LightWave Scene (.lws) Modo (.lxo) Stereolithography (.stl) DirectX X (.x) AC3D (.ac) Milkshape 3D (.ms3d) * TrueSpace (.cob,.scn)	Yes
Transition to Product	Yes	?	Yes
Code Editor	Yes	Qt Creator	Yes
Code Coverage	No		No

Performance analysis	Yes	QML profiler, C++ Analyzer included	Yes
Auto Consistency Checks	Yes	?	No
Handicapped input support	Yes	Supported	Yes
Linux Toolchain	Yes	Supported	Yes
Architecture			
Cross Platform	Yes	Qt framework works on multiple computer platforms, which makes it attractive for those who adhere to "write once, compile anywhere" philosophy. With Qt, one can write application once, and deploy it across desktop, mobile and embedded operating systems without rewriting the source code, saving time and development cost. In the automotive world, a variety of operating systems is used - Linux, WinCE, QNX, VxWorks; and Qt runs on all of these. http://doc.qt.io/qt-5/supported-platforms.html	Yes
Cross OS	Yes	Qt is best known as the best cross platform. Qt framework works on multiple computer platforms, which makes it attractive for those who adhere to "write once, compile anywhere" philosophy. With Qt, one can write application once, and deploy it across desktop, mobile and embedded operating systems without rewriting the source code, saving time and development cost. In the automotive world, a variety of operating systems is used - Linux, WinCE, QNX, VxWorks, Integrity; and Qt runs on all of	Yes
OEM independence	Yes	?	No
Separation of code and UI data	Yes	Logic is/can be separated from UI and also customizable.	Yes
Is a separate renderer required?	No		No
Mulitiple frame buffer support	Yes	Supported	Yes
Multiple application frame buffer access	Yes	Supported	
State Machine support	Yes	Qt SCXML and State Machine tooling in Qt Creator	Yes
Portable device integration support	Yes	Possible	Yes
Open Standards / License / etc.	Yes	Qt is C++ and Java script based. Qt has been on a dual licensing policy which are LGPL and Commercial.	Yes
3rd Party App support	Yes	3rd parties can easily port their app on Qt platform	Yes
MOST Interface support	No		No
Graphics		Supported. PNG, JPEG, Tiff, BMP, Vector graphic and more	
2D Graphics Support	Yes		Yes
3D Graphics Support	Yes	Supported. See above line 46	Yes
Open GL software support	Yes	Supported	Yes
Open GL hardware support	Yes	OpenGL ES 2.0 and 3.0 supported	Yes
Animation support	Yes	Qt Quick supports animations	Yes
3D file format support	Yes	See above line 46	Yes
Common graphics file format support	Yes	PNG, JPEG, Tiff, BMP, Vector graphics and more are supported	Yes
Easy Theme change support	Yes	Supported by Qt quick controls by standard. (Need to be designed and implemented)	Yes
Multiple HMI on single screen support	Yes	Supported by Qt Wayland. We have released the Qt Wayland Compositor API, which allows you to create your own Wayland compositor. We give you all the tools you need to create a compositor in a simple way. You can easily define the UI and UX of your compositor with the power and simplicity of QML. If you still need or want to use C++, that's also possible.	Yes
Vector graphics support	Yes	Supported. Qt SVG, Scalable Vector Graphics (SVG) is an XML-based language for describing two-dimensional vector graphics. Qt provides classes for rendering and displaying SVG drawings in widgets and on other paint devices.	No
Custom Widget support	Yes	Supported. Qt Designer can display custom widgets through its extensible plugin mechanism, allowing the range of designable widgets to be extended by the user and third parties.	Yes
Mix mode graphics (2D & 3D on same screen)	Yes	Supported (multiple options)	
Performance/Throughput requirement (?)	Yes	Performance optimization can be done by customer or support from Qt	Yes
Anti-allias support	Yes	Supported	Yes