

用Qt开发的 十大理由



KEY:



视频链接



博文/网页



ROI 计算器

#1 完美的用户体验

卓越的用户体验不仅大幅提升品牌，还能吸引众多忠实用户。

Qt 跨平台框架和工具大幅提升开发团队的开发效率，专注于让产品脱颖而出，具体表现在以下几个方面：

- 在任何平台上都能以最佳方式稳定、高性能地运行软件
- 现代、响应迅速的2D和3D用户界面，超出用户期望
- 在不同目标设备上保持一贯的品牌或原生外观



{ [查阅更多](#)
[中文资料](#) }

“就最终体验、性能和特性而言，Qt 绝对是开发Radeon Software Crimson Edition的正确选择。”



“MBUS 是梅赛德斯-奔驰汽车内的全新用户体验。我们用 Qt 开发了绝大部分的UI体验和软件，包括屏幕动画，屏幕间的过渡和小组件。Qt 使我们能够快速开发出原型系统，实现软件的跨平台运行。”



“凭借Qt的快速迭代，我们能够为客户提供最佳的用户体验。”



#2 降低开发成本

- Qt支持多个平台，能在节省您宝贵开发、维护资源的同时，降低您的硬件成本。
- 您只需要一个开发团队用Qt开发、维护一个代码库，就可以支持所有的目标平台。
- 得益于C++的优势，基于Qt的应用在低成本硬件上也可以高效运行，从而大幅降低您的硬件成本。

研发成本

“客户通过使用Qt平均节省了30%的软件开发成本。此外，单一的开发环境整体上简化了设备创建过程。”



Qt

{ [查阅更多中文资料](#) }

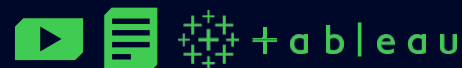
硬件成本

“相较于基于Qt的解决方案，用基于web解决方案实现类似iPhone的用户体验，需要更强大、更昂贵的片上系统。Qt可为全球最大家电制造商之一节省约53%的硬件成本。”



维护成本

“当我们实现开发一份代码，而程序可以运行在众多平台上的时候，我们就能把过去维护多个平台的成本降低到维护一个平台的水平。”



#3 加快产品上市

Qt简化了项目各个阶段的工作流程。有了Qt，设计师可以在真实的硬件环境中创建原型并迭代，无需开发者的协助。

Qt库、预先制作的UI元素以及从设计原型自动生成的代码大幅减少了开发者的工作。得益于Qt跨平台的特性，当您的应用需要运行的设备类型越多，您节省的时间就越多！

如果您的项目需要改进，Qt专业服务团队随时能够为您提供帮助。



{ 查阅更多
中文资料 }

代码重用

“我们开发各种C++组件，并需要在多个项目中使用这些组件。我们创建了各种模块，当需要在其他项目中使用时，只需使用Qt QML创建一个新的UI，而不必每次都从零开始开发整个系统。”



高效的工作流

“我们能够非常快地创建原型。唯一的问题是客户以为这已经不是一个原型了，因为它看起来太棒了！Qt让我们能在多种平台上快速重用之前的项目，第一时间满足客户的需求。”



咨询、培训和支持

“我们只与最好、最有信誉的公司合作。Qt提供的技术支持在整个开发周期中对我们来说是不可或缺的，显著缩短了产品的上市时间。”



GLPSYSTEMS

#4 提升应用速度

快速的交互响应

“Qt帮助我们实现了良好的外观图形界面，同时实现了高性能和高帧频，这对提升用户体验非常重要。”



“一位开发人员花了160个小时在嵌入式系统上分别用Qt/QML和HTML5创建了相同的应用程序，结果发现基于Qt的应用程序总体而言运行更快、响应更迅速，而且更易于实现。”



{ 查阅更多
中文资料 }

今天的消费者对自己的设备有更多的期望，因为有更多的选择，用户体验中的微小细节就能够使您的产品脱颖而出。

得益于C++，基于Qt的软件能以更少的内存开销实现硬件集成，这使其无论运行在高端多屏系统还是微控制器上，都能实现快速的交互响应、更短的启动时间、更流畅的动画，以及更好的性能。

更好的性能

“当需要在单一屏幕上同时显示2D和3D内容时，使用Qt会大有裨益。在Mini上，我们在2D屏幕上渲染了AR，同时我们还将Qt运行在一个资源非常有限的SoC上，能做到这些实在是太重要了。”



#5 提升开发体验

享受开发过程的开发者才能富有成效。

如C++开发者所愿，Qt提供了**最快、最简单和最有趣**的开发体验。

Qt拥有**简单直观的开发环境**，其代码编辑器具有语法高亮显示、自动补全、拖拽式的UI设计创建、可视化调试和分析以及许多其他功能。

Qt还因其高质量的文档和示例享誉业界，您可从中获得灵感，或直接节省开发时间。

快速上手

“我们没花多长时间就能用Qt开发了。事实上，学习并开始使用Qt都很容易。”

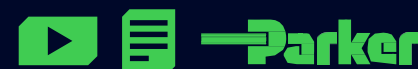
“Qt非常适合这个项目，从一开始就能帮助加速开发，我们刚安装完几天就开始使用了。”



{ [查阅更多](#)
{ [中文资料](#) }

简单易用

“用QML编程的效果真是令人惊叹。它的易用性和效率让我们很惊讶。QML为我们的软件提供了坚实的基础。”



丰富的文档和示例

“我们认为用Qt开发我们的UI非常有趣，其中主要原因是它的使用手册内容齐全。”



#6 丰富的类库和工具集

基于不同框架的库和工具集开发应用程序，可能面临着不稳定和复杂性的问题——问问Frankenstein博士就知道了。

Qt是唯一最完整的C++软件开发套件，拥有开发软件所需的所有工具和构建模块：从开发到部署，从前端到后端。没有相关依赖需要担心，也无需浪费时间去寻找最佳的开发工具——因为这些它都有了。

完整的工具套件

“Qt拥有开发者所需要的工具套件，同时它拥有大量类库，并且不断在增加。”

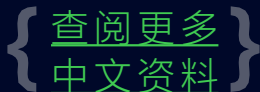


易于理解的各种库

“我想并不是每个人都知道Qt库有多强大。使用Qt以后，我们的开发过程比以前容易多了。以前开发大型自动化系统可能需要1000到1300小时。使用Qt后，整个开发过程减少到了50个小时。”



“Qt 提供了大量工具和支持类，能够帮助您以最有效的方式构建结构良好的软件。”



#7 丰富的社区

一个开发框架的生态系统可以长期、有效地支撑基于它开发的项目。Qt拥有超过一百万专注的用户群体。植根于开源社区，Qt的不断发展得益于世界各地乐于奉献的开发人员。Qt公司的发展已经超过了25年，并将继续成为您值得信赖的合作伙伴。

开源社区的贡献

“令人惊喜的是，Qt社区已经完成了大量的工作。我们能够利用社区的资源、代码和合作伙伴关系，为我们的产品打下了良好基础。没有开源社区的帮助，我们就需要更长的时间来完成交付。”



{ [查阅更多](#)
[中文资料](#) }

>100万开发者



合作伙伴网络



技术

服务



培训

分发



社区

#8 跨平台能力

“更少代码、更多创意、任意部署！”是Qt的宗旨。跨平台开发不仅节省时间，更是战略推动者。

基于Qt开发的软件在所有设备上的外观、体验都能保持一致，并为潜在市场提供面向未来的支持。随时为下一个伟大创新做好准备：一旦完成了基本代码开发，后续扩展到其他目标平台，无论是高端多屏系统还是MCU，都轻而易举。

一个代码库



{ 查阅更多
中文资料 }

海量支持和可扩展

“我们有手持设备、监控设备、平板设备，未来我们还会有很多其他的设备架构。我们想要一次开发，在任何设备上都能表现出色。Qt的跨平台支持使我们能够做到这一点。”



一致的外观体验

“我们正在开发中的项目是让iOS代码也适用于安卓。我希望你认为你使用的是Android应用，而不是在Android上运行的iOS应用。我们选择Qt的原因之一是为了让一套代码在不同系统上都有原生UI的外观体验。”



开发语言

“我们选择Qt就是因为它是最好的C++工具包，具备出色的用户界面功能——在跨操作系统开发应用方面，Qt没有什么能真正竞争对手。”



“大公司通常会有庞大的开发团队开发应用。用QML为新平台创建版本就像变魔术一样容易。”



Qt

{ 查阅更多
中文资料 }

#9 多语言支持

对您的开发人员和客户来说，Qt已让语言障碍成为过去。

Qt 支持不同的开发语言。C++ 高效、功能强大且用途广泛。QML是Qt中创建UI的声明性标记语言，它使UI的创建过程变得快速而容易。几乎所有可以用Qt做的事情，现在都可以用Python来完成！

如果您希望产品国际化，Qt 的本地化、国际化工具可以为您省去无数与语言翻译、货币转换、日历格式等相关的烦恼。

本地化和国际化

“如果没有 Qt Linguist工具，我们就必须从外部聘请专业人士。在整个项目期间，它为我们节省了一到两个人的工作。”



#10 灵活易扩展的架构

可以从第三方图形设计工具导入UI资源、应用Qt的预制小部件和控件，或者使用QML来设计您的UI资产。

在Qt自己的 IDE 中编写软件、添加插件，或者在其他IDE中使用Qt库。准确选择目标平台所需的模块，通过板卡级支持包部署软件到设备，通过WebAssembly运行在浏览器，或通过WebGL运行于无屏设备。

“我们想做很多事，我们想做各种尝试和试验。有了Qt，对于我们能做什么、不能做什么再没有了技术限制。”



“过去，UI开发者需要先为嵌入式开发者创建模型。现在，Qt让前者可以直接在产品上工作，消除了设计师和开发者之间日常复杂、耗时的来回过程。”



“车内的用户体验将变得日益重要，甚至远比现在的重要性。我们很高兴能与Qt公司合作，Qt公司大力支持了我们设计过程。Qt让设计师变得无限自由，创造出来的东西令人不可思议。”



——汽车设计师
Murat Günak



{ 查阅更多
中文资料 }



**THE FUTURE IS
WRITTEN WITH**

