Для разработки будущей программы будет использоваться язык QT версии 5.11. Выбор обоснован тем, что данный язык:

Здесь я перечислю основные пласты Qt и почему он, а не другие языки.

**№1 Qt очень удобное API**, а конкретнее говоря, модуль qtbase содержит достаточное количество классов для большинства повседневных задач (Qt — это больше, чем GUI фреймворк,). Я уже говорил об обертках STL-ных контейнеров в своей статье трехгодичной давности — [тыцк](http://habrahabr.ru/post/127870/). Классы для работы со строками, отладочный вывод, и много-много чего, так же included.

**Стоит сказать, что у Qt** также есть модули для удобной работы с XML, базами данных (**с интеграцией системой MVC**), OpenGL, аудио/видео-работы (Phonon), сетевого программирования, WebKit2. Для большинства задач, которые стоят перед среднестатистическим проектом — этой кухни хватает в 90% случаев, а с модулями редко случаются проблемы.

**С учетом моей любви к C++,** я очень и очень сильно доволен тем, какую поддержку различных нетривиальных вещей Qt представляет на кросс-платформенном уровне.

**№2 Qt Quick** — это новый подход к созданию графического пользовательского интерфейса. Используя декларативный язык QML (угадайте, где его придумали), похожий на JavaScript, можно добиться высокой производительности при прототипировании интерфейса в приложениях любой сложности. А самое забавное, что при таком ходе дел, **с прототипированием интерфейса может справиться даже дизайнер, который знает синтаксис JavaScript**.



Рис 1.

**№3 Сообщество.**

Qt проводит ежегодные мероприятия — Qt Developer Days и Qt Contributors' Summit. Я был на каждом из них по одному разу, в прошлом году, и мне очень понравилось — уровень подготовки высокий, а впечателения доставляют. Мне также приходилось общаться с «ветеранами» Qt — людьми, которые посещали саммит на протяжении 10 лет. К новичкам эти люди очень терпимы и относятся хорошо. На Qt Project есть форумы, где каждый желающий может получить ответ на волнующий его вопрос. Забавно, но девнет действительно **очень живой** и там **реально** отвечают на вопросы, возникающие, в процессе познания Qt.

#### №4. Открытый исходный код и code review

**№5 Qt в сравнении с тем же самым сильным Visual Studio**

Теперь о самом программировании. Писали на C++/CX, где вы можете комбинировать C++ и широкий выбор библиотек Windows Runtime. Тут Microsoft активно продвигает стандарт C++11 и более новые. Допустим, все асинхронные функции построены на основе [лямбда-выражений](https://vk.com/away.php?to=http%3A%2F%2Fhabrahabr.ru%2Fpost%2F66021%2F), активно используется квалификатор auto, часто встречается необходимость в [умных указателях](https://vk.com/away.php?to=http%3A%2F%2Farchive.kalnitsky.org%2F2011%2F11%2F02%2Fsmart-pointers-in-cpp11%2F) (таких как shared\_ptr). И это круто. Волей-неволей, но придется вникать в современные стандарты языка, который почему-то даже в новой Qt Creator по умолчанию отключен.

В принципе, как в Qt, так и в Windows Runtime все построено на классах. Но как только начинаешь писать код и использовать какие-то библиотеки, то понимаешь, что есть одно огромное отличие – большая часть работы построена на создании/вызове асинхронных функций. Если вы прежде не сталкивались с этим, то поверьте — вам придется почитать документацию (в этом случае это даже хорошо, вникните в асинхронное программирование). Вот материал по теме PPL (библиотеки параллельных шаблонов):

* [Создание асинхронных операций в C++ для приложений для Магазина Windows](https://msdn.microsoft.com/ru-ru/library/windows/apps/hh750082.aspx);
* [Параллелизм задач (среда выполнения с параллелизмом)](https://msdn.microsoft.com/ru-ru/library/windows/apps/dd492427.aspx);
* [Асинхронное программирование на языке C++ (приложения среды выполнения Windows)](https://msdn.microsoft.com/library/windows/apps/Hh780559.aspx).

Думаю, что через пару дней прочтения и разбора пары примеров, у вас сложиться представление о том, как писать приложения на C++/CX. В сравнении с Qt: использование уже готовых асинхронных функций, а также довольно-таки простое создание своих собственных намного проще, чем параллельные вычисления с помощью потоков в Qt.