

Краткое описание приложения

Qt Creator — это интегрированная среда разработки (IDE), предназначенная для создания приложений с использованием фреймворка Qt. Qt Creator поддерживает разработку как для настольных, так и для мобильных платформ. Он включает функции для редактирования кода, компиляции, отладки и управления проектами.

Основные функции Qt Creator включают текстовый редактор с подсветкой синтаксиса, поддержку отладки на нескольких языках, управление проектами, конфигурацию сборки и визуальный редактор интерфейсов. Однако у Qt Creator имеются недостатки в интерфейсе и некоторых функциях, что делает его хорошим объектом для CWT-анализа.

Задачи для CWT-анализа

Мы выберем две основные задачи в Qt Creator, которые требуют выполнения нескольких действий и могут быть улучшены:

1. **Добавление и настройка нового проекта** — процесс, требующий выбора шаблона, задания конфигураций и настройки параметров компиляции.
2. **Отладка программы** — включает запуск отладчика, настройку точек останова, наблюдение за переменными и выполнение шагов кода поэтапно.

Задача 1: Добавление и настройка нового проекта

Действие	История	Проблемы	Пути решения
Выбор типа проекта	<p>Пользователь на начальном этапе создания проекта выбирает тип из предложенных категорий (например, консольное приложение, графическое приложение и т. д.). Qt Creator предлагает различные шаблоны, но названия и категории могут быть неочевидными, особенно для начинающих пользователей. Некоторые шаблоны предназначены для специфических случаев использования, что может сбивать с толку. Обычно этот этап требует внимания к деталям, чтобы не выбрать неправильный шаблон, что может привести к сложностям на дальнейших этапах.</p>	<p>Интерфейс выбора шаблонов бывает сложен для восприятия новичками, так как некоторые шаблоны непонятны на первый взгляд. Отсутствие ясных, кратких описаний создает барьер для быстрого понимания, какой шаблон выбрать. В результате пользователи могут тратить время на повторное создание проектов или сталкиваться с несоответствиями в шаблонах.</p>	<p>Упростить интерфейс выбора типов проектов, введя более понятные, интуитивно ясные названия для шаблонов. Добавить краткие описания под названием каждого шаблона, поясняющие, для каких случаев он подходит. Это поможет новичкам быстрее разобраться в выборе и избежать ненужных ошибок.</p>

Настройка конфигураций сборки	<p>На этом этапе пользователь задает параметры конфигурации сборки (например, выбирает режимы «Отладка» или «Релиз»). Qt Creator предоставляет несколько стандартных конфигураций, но пользователи могут добавлять и настраивать свои. Однако, если пользователь неправильно понимает конфигурацию, это может привести к ошибкам при компиляции или работе программы.</p>	<p>Не все пользователи, особенно новички, понимают разницу между конфигурациями. В интерфейсе бывает сложно разобраться, особенно когда нужно переключиться между конфигурациями. В некоторых случаях система сохраняет неверные конфигурации, что вынуждает пользователя возвращаться и менять настройки повторно.</p>	<p>Добавить всплывающие подсказки для каждой опции конфигурации, с объяснением их назначения и случаями, когда они наиболее уместны. Это поможет пользователям быстрее понять, как выбрать подходящую конфигурацию и предотвратить возникновение ошибок в процессе сборки.</p>
-------------------------------	---	---	--

<p>Настройка параметров компиляции</p>	<p>Пользователь может вручную добавить дополнительные параметры компиляции, такие как флаги оптимизации или указания на использование определенных библиотек. Эти параметры настраиваются через текстовое поле, что удобно для опытных пользователей, но может вызывать трудности у новичков. Встроенные подсказки и объяснения отсутствуют, поэтому поиск правильных параметров может занимать много времени.</p>	<p>Отсутствие подробной документированности в интерфейсе усложняет настройку параметров для неопытных пользователей. Некоторые параметры компиляции сложно найти и понять, из-за чего новичкам трудно определить, какие настройки подходят для их задачи. Кроме того, предустановленные параметры не всегда находятся в доступном месте.</p>	<p>Включить кнопку «Документация» рядом с полем параметров компиляции. Кнопка должна открывать справочную информацию, где перечислены наиболее часто используемые параметры компиляции с пояснениями и рекомендациями по их применению. Это сделает процесс настройки более интуитивным и понятным.</p>
--	--	--	---

<p>Сохранение и запуск проекта</p>	<p>Завершающий этап создания проекта предполагает выбор папки для его сохранения, ввод имени проекта и начало процесса компиляции. Однако интерфейс Qt Creator на этом этапе выводит множество настроек, таких как подключение к системе контроля версий, конфигурация путей и переменных окружения, что может показаться сложным для восприятия, особенно новичкам. В результате многие пользователи оставляют настройки по умолчанию, не понимая их назначения, что может привести к проблемам при дальнейшей работе с проектом.</p>	<p>Для первого запуска проекта интерфейс сохранения и настройки бывает излишне перегруженным. Новичкам часто трудно понять, какие параметры важны, а какие можно пропустить. Это создает путаницу, и многие оставляют настройки по умолчанию, не всегда подходящие для их проекта.</p>	<p>Разделить процесс создания и настройки проекта на несколько окон или вкладок, чтобы упростить структуру интерфейса. Каждый этап настройки можно разбить на логические блоки, так пользователи смогут последовательно заполнить поля и настройки, не чувствуя перегрузки информацией.</p>
------------------------------------	--	--	---

Задача 2: Отладка программы

Действие	История	Проблемы	Пути решения
Запуск отладчика	На этапе отладки пользователь запускает встроенный в Qt Creator отладчик для текущего проекта. Отладчик предоставляет доступ к функциям установки точек останова, просмотра значений переменных, пошагового выполнения и др. Однако на больших проектах запуск отладчика может затянуться, а производительность IDE снизиться. Особенно это заметно, когда проект содержит много файлов и зависимостей.	Запуск отладчика на крупных проектах может замедляться, что вызывает неудобства и влияет на продуктивность. В ряде случаев IDE становится нестабильной, и пользователи теряют время на ожидание. Это особенно раздражает при частых перезапусках для тестирования небольших изменений.	Оптимизировать производительность отладчика на крупных проектах, снизив задержки. Можно добавить предупреждения о возможном увеличении времени на запуск отладчика для больших проектов, что поможет пользователям планировать работу с учетом задержек.
Установка точек останова	Пользователь устанавливает точки останова на строках кода, где он хочет приостановить выполнение программы для проверки состояния. Qt Creator позволяет добавлять точки останова, но интерфейс не всегда удобен для их управления. Особенно это проявляется, если точек несколько и пользователь теряет в их расположении.	Интерфейс точек останова иногда сложен для понимания, особенно при множестве точек останова, что затрудняет управление ими. Нет отдельного окна, где можно было бы увидеть все установленные точки останова и их состояния, что делает процесс их поиска и управления сложным.	Создать отдельное окно для управления точками останова. В этом окне пользователь мог бы видеть список всех установленных точек останова, их состояние (активны или нет) и легко включать или отключать их при необходимости. Это сделает работу с точками останова более удобной и быстрой.

Просмотр значений переменных	Во время отладки пользователю часто нужно просматривать значения переменных. В Qt Creator значения переменных выводятся в отдельном окне, однако если переменных много, это окно становится перегруженным, и пользователю трудно быстро находить нужные данные.	При работе с большими проектами окно переменных становится перегруженным, и пользователю сложно отслеживать изменения нужных переменных. Отсутствие категорий для переменных также усложняет навигацию, особенно при множестве локальных и глобальных переменных.	Разделить просмотр переменных на категории, такие как «Локальные», «Глобальные» и «Аргументы функции». Это позволит пользователям быстрее ориентироваться в списке переменных, находя нужные данные, что повысит удобство отладки.
Шаговая отладка	Пошаговая отладка используется для пошагового выполнения программы, что позволяет пользователю детально просматривать выполнение кода. На больших проектах при пошаговой отладке производительность IDE иногда падает, а окно со значениями переменных может переставать реагировать на действия пользователя.	На больших проектах пошаговая отладка может сильно замедляться, и работа с переменными становится менее отзывчивой, что затрудняет анализ кода. Пользователь теряет время, ожидая обновления интерфейса и синхронизации данных переменных с выполнением кода.	Оптимизировать работу отладчика при пошаговом выполнении для больших проектов. Можно уменьшить объем данных, обновляемых в окне переменных, что снизит нагрузку на систему и сделает шаговую отладку более отзывчивой.

Заключение

CWT-анализ выявил несколько важных моментов, на которые стоит обратить внимание при разработке и улучшении Qt Creator. Оптимизация этих моментов могла бы существенно улучшить пользовательский опыт, особенно для новичков, которые могут столкнуться с затруднениями при выполнении стандартных задач, таких как настройка проекта и отладка кода.