



# Технологии QNX и КПДА в России

Санкт-Петербург, 30 октября 2018

«Использование фреймворков модульного тестирования в среде разработки QNX Momentics IDE»

Глеб Осипов, ООО «СВД Встраиваемые Системы»

# О чем будем говорить

- Что такое модульное тестирование и зачем оно нужно:
  - Понятие;
  - Роль в общей системе тестирования;
  - Известные решения;
- *QNX Momentics IDE 7.0* и фреймворки для модульного тестирования;
- *Google C++ Testing Framework* и *QNX Momentics IDE 7.0*:
  - Особенности получения и подключения фреймворка;
  - Совместное использование;
  - Пример использования.

# Что такое модульное тестирование?

- Это процесс, позволяющий проверить на корректность отдельные модули (*юниты*) исходного кода программы:
  - Для процедурных и функциональных языков - юнитом является функция;
  - Для объектно-ориентированных языков модуль может быть как отдельным классом, так и методом.
- Реализовываться модульные тесты могут:
  - До начала процесса разработки (*TDD*);
  - Параллельно с процессом разработки;
  - После окончания разработки.
- Реализация – сфера ответственности разработчика.

# Какова роль модульного тестирования в общей схеме?



*Пирамида автоматизации тестирования*

# Какие существуют фреймворки для модульного тестирования?

- Для языка программирования C:
  - *CUnit, Check, CuTest, MinUnit* etc;
- Для языка программирования C++:
  - *CppUnit, Google C++ Testing Framework, Qt Test Framework* etc;
- В качестве примера будет использован фреймворк *Google C++ Testing Framework*.

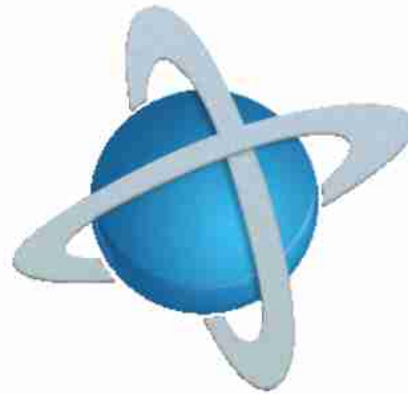
# Что такое Google C++ Testing Framework?

## *Google C++ Testing Framework:*

- Проект с открытыми исходными текстами;
- В собранном виде представляется в виде библиотеки и набора заголовочных файлов;
- Исходные тексты доступны по ссылке:

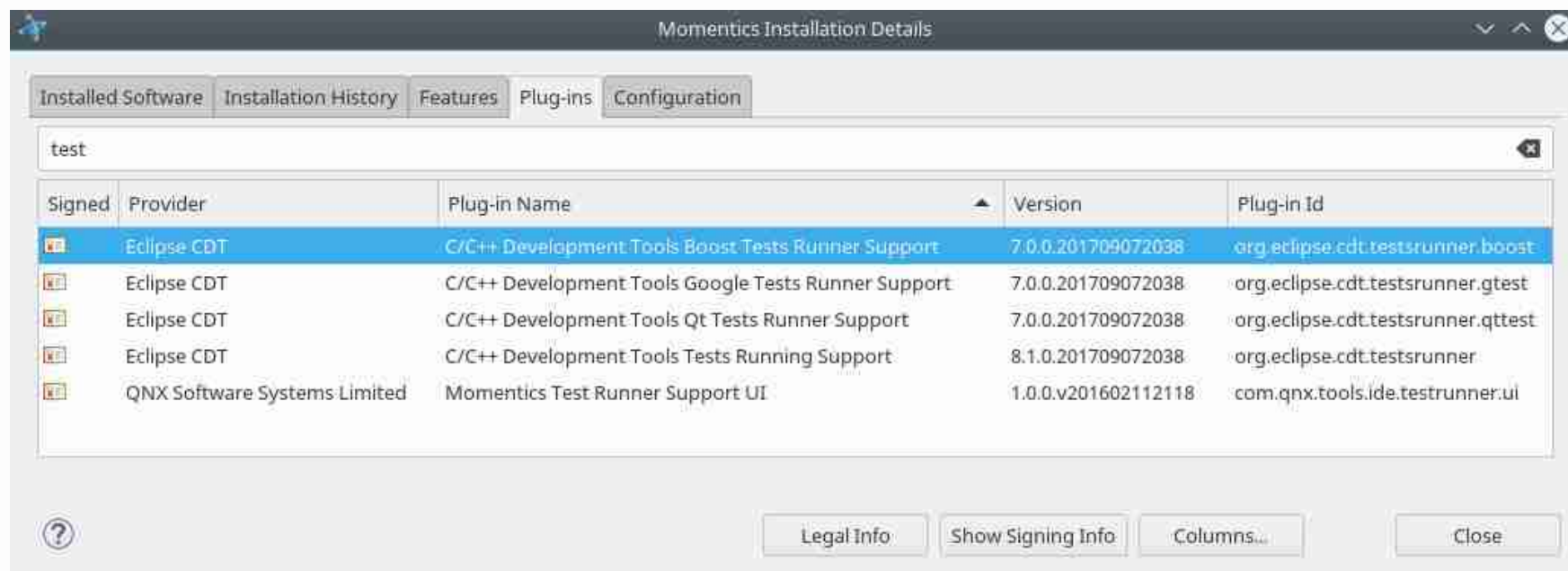
<https://github.com/google/googletest>






# *QNX Momentics IDE 7.0 и фреймворки для модульного тестирования*



*QNX Momentics IDE 7.0 предоставляет новые возможности для работы с фреймворками для модульного тестирования относительно **IDE 4.7**.*

# Что качественно изменилось при переходе на QNX Momentics IDE 7.0?



| Signed  | Provider                     | Plug-in Name  | Version             | Plug-in Id                        |
|---|------------------------------|---|---------------------|-----------------------------------|
|  | Eclipse CDT                  | C/C++ Development Tools Boost Tests Runner Support  | 7.0.0.201709072038  | org.eclipse.cdt.testrunner.boost  |
|  | Eclipse CDT                  | C/C++ Development Tools Google Tests Runner Support | 7.0.0.201709072038  | org.eclipse.cdt.testrunner.gtest  |
|  | Eclipse CDT                  | C/C++ Development Tools Qt Tests Runner Support     | 7.0.0.201709072038  | org.eclipse.cdt.testrunner.qttest |
|  | Eclipse CDT                  | C/C++ Development Tools Tests Running Support       | 8.1.0.201709072038  | org.eclipse.cdt.testrunner        |
|  | QNX Software Systems Limited | Momentics Test Runner Support UI                    | 1.0.0.v201602112118 | com.qnx.tools.ide.testrunner.ui   |

В состав *QNX Momentics IDE 7.0* включен набор плагинов, обеспечивающих взаимодействие с некоторыми тестовыми фреймворками.



# Что качественно изменилось при переходе на QNX Momentics IDE 7.0?

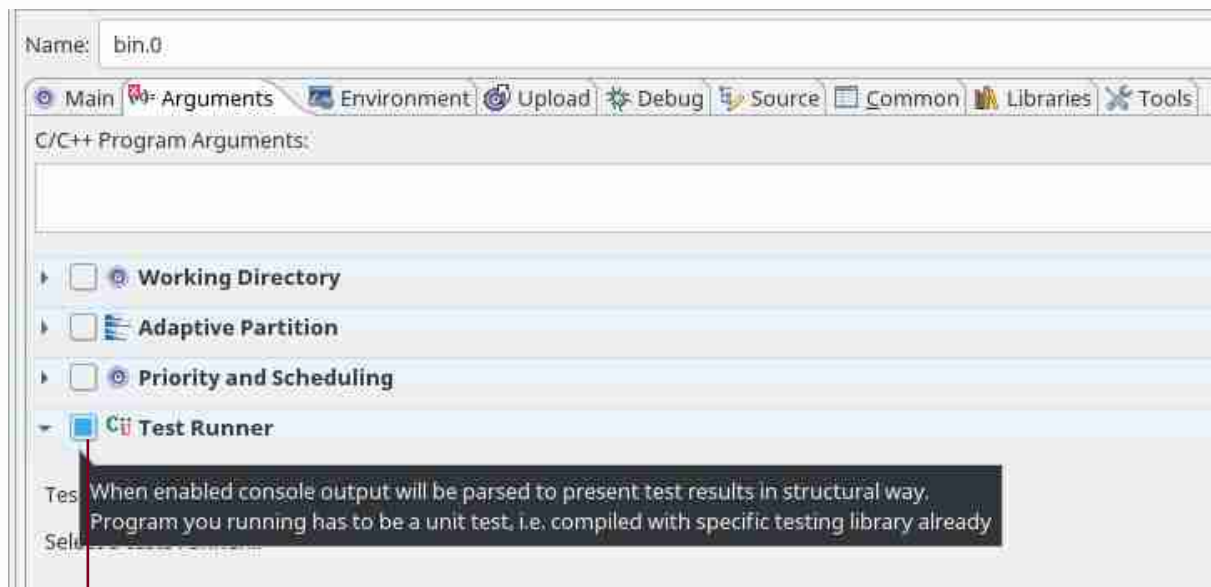
Добавлен плагин, обеспечивающий визуализацию результатов выполнения ТЕСТОВ

QNX Software Systems Limited    Momentics Test Runner Support UI    1.0.0.v201602112118 com.qnx.tools.ide.testrunner.ui

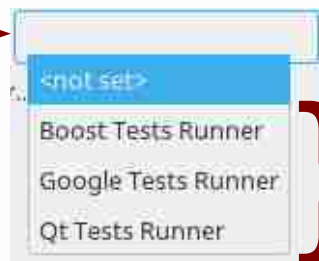
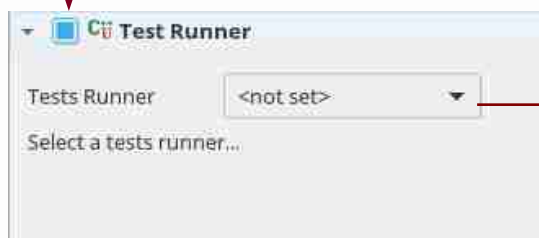
- QNX Momentics 7.0.0 IDE User's Guide
  - Working with QNX Momentics IDE
  - Developing Projects with the IDE
  - Debugging Applications
  - Unit Testing**
    - Writing and building test programs
    - Running test programs
    - Measuring code coverage
    - Importing Code Coverage results
    - Exporting Code Coverage results

В документации появилась секция «*Unit Testing*»

# Как включить визуализацию результатов выполнения тестов в QNX Momentics IDE 7.0?



*Создание новой конфигурации запуска*



*Поддерживаемые тестовые фреймворки*

# Google C++ Testing Framework и версия QNX SDP

В соответствующем разделе документации встречается следующая фраза:

*«This release of QNX SDP includes the Google C++ Testing Framework and the IDE is configured to work with it.»\**

- Речь идет о **QNX SDP 7.0**. В **QNX SDP 6.5.0** *Google C++ Testing Framework* изначально отсутствует, и вопрос его привнесения будет освещен ниже;
- Данный инструмент планируется к включению в **комплект разработчика для ЗОСРВ «Нейтрино»**.

---

\* «Эта версия QNX SDP включает в себя *Google C++ Testing Framework* и *QNX Momentics IDE 7.0* сконфигурирована для работы с этим фреймворком»

# Как получить Google C++ Testing Framework?

Два способа получения *Google C++ Testing Framework*:

- Оставить запрос по адресу - [support@kpda.ru](mailto:support@kpda.ru):
  - Предоставляется в виде архива, который необходимо распаковать по пути, указанному в переменной **QNX\_TARGET**;
  - Собирается отдельно для каждой целевой архитектуры;
- Получить в составе **комплекта разработчика для ЗОСРВ «Нейтрино»**.

# Практическое использование Google C++ Testing Framework

- Файлы тестовых программ целесообразно хранить в отдельной папке (в рамках проекта);
- В файле заголовка тестовой программы указывается заголовочный файл *Google C++ Testing Framework*:

```
#include <gtest/gtest.h>
```

- Для выполнения тестов необходимо вызвать функцию *RUN\_ALL\_TESTS()* в функции *main()*.

# Полезные ссылки

- Исходные коды *Google C++ Testing Framework*:

[\*https://github.com/google/googletest\*](https://github.com/google/googletest)

- Начальная документация по использованию *Google C++ Testing Framework*:

[\*https://github.com/google/googletest/blob/master/googletest/docs/primer.md\*](https://github.com/google/googletest/blob/master/googletest/docs/primer.md)

- Документация по использованию *Google C++ Mocking Framework*:

[\*https://github.com/google/googletest/blob/master/googlemock/docs/ForDummies.md\*](https://github.com/google/googletest/blob/master/googlemock/docs/ForDummies.md)

# Заключение

- Модульные тесты – самый “дешевый” тип тестирования, при этом, имеющий наиболее высокую скорость выполнения тестов;
- Реализация модульных тестов повышает качество и скорость разработки крупных проектов;
- Наличие модульных тестов в проекте позволяет осуществлять контроль регрессии;
- *Google C++ Testing Framework* обеспечивает широкие возможности по реализации модульных тестов;
- *QNX Momentics IDE 7.0* предоставляет возможность удобного использования *Google C++ Testing Framework*.

# Спасибо за внимание

**Глеб Осипов**

Инженер-программист

+7 (812) 346-89-56

[g.osipov@kpda.ru](mailto:g.osipov@kpda.ru)

[www.kpda.ru](http://www.kpda.ru)

[www.swd.ru](http://www.swd.ru)

