

Введение в тестирование

# Инструментальные средства

Кулаков Кирилл Александрович

# Структура демо проекта

- **qmake-gtest.pro** — основной файл конфигурации
- **app/** — исходный код приложения
  - **app.pro** — конфигурация сборки приложения (не забудьте отредактировать!)
  - **main.c** — точка запуска (функция `main`) (заменяете на свое)
  - **myfunc.h** — декларации функций модуля (заменяете на свое)
  - **myfunc.c** — код функций модуля (заменяете на свое)

# Конфигурация приложения

```
TEMPLATE = app
```

```
CONFIG += console
```

```
CONFIG -= app_bundle
```

```
CONFIG -= qt
```

```
SOURCES += main.c \
```

```
    myfunc.c
```

```
HEADERS += \
```

```
    myfunc.h
```

```
QMAKE_CFLAGS += -Wall -Wextra -Werror
```

```
# gcov
```

```
QMAKE_CFLAGS += -fprofile-arcs -ftest-coverage
```

```
LIBS += -lgcov
```

# Структура демо проекта

- **tests/** — каталог тестов
  - **tests.pro** — конфигурация тестирования
  - **gtest\_dependency.pri** — подключение google test
  - **main.cpp** — драйвер запуска тестов
  - **tst\_test1.h** — тесты
  - **fibonachi\_test.h** — еще тесты

# Конфигурация тестирования

```
include(gtest_dependency.pri)
```

```
TEMPLATE = app
```

```
CONFIG += console
```

```
CONFIG -= app_bundle
```

```
CONFIG += thread
```

```
CONFIG -= qt
```

```
QMAKE_CXXFLAGS += -Wall -Wextra -Werror
```

```
QMAKE_CFLAGS += -Wall -Wextra -Werror
```

```
# gcov
```

```
QMAKE_CXXFLAGS += -fprofile-arcs -ftest-coverage
```

```
QMAKE_CFLAGS += -fprofile-arcs -ftest-coverage
```

```
LIBS += -lgcov
```

```
HEADERS += tst_test1.h \
```

```
  ../app/myfunc.h \
```

```
  fibonacci_test.h
```

```
SOURCES += main.cpp \
```

```
  ../app/myfunc.c
```

# Драйвер запуска тестов

- C++ код!!!

```
#include "tst_test1.h"
```

```
#include "fibonachi_test.h"
```

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

```
int main(int argc, char *argv[])
```

```
{
```

```
    ::testing::InitGoogleTest(&argc, argv);
```

```
    return RUN_ALL_TESTS();
```

```
}
```

# Написание теста

- Заголовочный файл!

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

```
#include <gmock/gmock-matchers.h>
```

```
/* using namespace testing; */
```

```
extern "C" {
```

```
/* подключение деклараций ваших функций */
```

```
    #include "myfunc.h"
```

```
}
```

```
TEST(test_group_name, test_name)
```

```
{
```

```
    /* Подготовка объекта тестирования (запуск) */
```

```
    /* Проверка результата работы и информирование (утверждение) */
```

```
}
```

# Виды утверждений

- Критические отказы начинаются с ASSERT\_, некритические — EXPECT\_
- Простейшие логические
  - ASSERT\_TRUE(condition);
  - ASSERT\_FALSE(condition);
- Сравнение
  - ASSERT\_EQ(expected, actual); — =
  - ASSERT\_NE(val1, val2); — !=
  - ASSERT\_LT(val1, val2); — <
  - ASSERT\_LE(val1, val2); — <=
  - ASSERT\_GT(val1, val2); — >
  - ASSERT\_GE(val1, val2); — >=
- Сравнение строк
  - ASSERT\_STREQ(expected\_str, actual\_str);
  - ASSERT\_STRNE(str1, str2);
  - ASSERT\_STRCASEEQ(expected\_str, actual\_str); — регистронезависимо
  - ASSERT\_STRCASENE(str1, str2); — регистронезависимо



# Виды утверждений

- Сравнение чисел с плавающей точкой
  - `ASSERT_FLOAT_EQ(expected, actual);` — неточное сравнение float
  - `ASSERT_DOUBLE_EQ(expected, actual);` — неточное сравнение double
  - `ASSERT_NEAR(val1, val2, abs_error);` — разница между `val1` и `val2` не превышает погрешность `abs_error`
- Вызов отказа или успеха
  - `SUCCEED();`
  - `FAIL();`
  - `ADD_FAILURE();`
  - `ADD_FAILURE_AT(«file_path», line_number);`

# Запуск тестов в среде Qt SDK

qmake-gtest - Qt Creator

Файл П\_равка С\_борка О\_тладка А\_нализ И\_нструменты О\_кно С\_правка

Проекты <нет документа>

Начало Редактор Дизайн Отладка Проекты Справка

Открытые документы

Результаты тестирования

Test summary: 6 успехов, 0 ошибок.

- PASS Выполнение теста test1
  - PASS test1.suite1 tst\_test1.h 9  
Выполнение заняло 0 ms. tst\_test1.h 9
  - PASS test1.suite2 tst\_test1.h 20  
Выполнение заняло 0 ms. tst\_test1.h 20  
Выполнение теста заняло 0 ms total tst\_test1.h 20
- PASS Выполнение теста test2
  - PASS test2.suite1 tst\_test1.h 15  
Выполнение заняло 0 ms. tst\_test1.h 15  
Выполнение теста заняло 0 ms total tst\_test1.h 15
- PASS Выполнение теста fibonacciTest
  - PASS fibonacciTest.num0 fibonacci\_test.h 10  
Выполнение заняло 0 ms. fibonacci\_test.h 10
  - PASS fibonacciTest.greather2 fibonacci\_test.h 16  
Выполнение заняло 0 ms. fibonacci\_test.h 16
  - PASS fibonacciTest.negative fibonacci\_test.h 21  
Выполнение заняло 0 ms. fibonacci\_test.h 21  
Выполнение теста заняло 0 ms total fibonacci\_test.h 21

Быстрый поиск (...)

1 Проблемы 2 Результаты поиска 3 Вывод приложения 4 Консоль сборки 5 Консоль отладчика 8 Результаты тестирования

# Запуск тестов в консоли

- qmake
- make
- ./tests/tests

```
kulakov@ultrabook: ~/projects/testing/qmake-gtest
Файл Правка Вид Поиск Терминал Справка
kulakov@ultrabook:~/projects/testing/qmake-gtest$ ./tests/tests
[=====] Running 6 tests from 3 test cases.
[-----] Global test environment set-up.
[-----] 2 tests from test1
[ RUN     ] test1.suite1
[       OK ] test1.suite1 (0 ms)
[ RUN     ] test1.suite2
[       OK ] test1.suite2 (0 ms)
[-----] 2 tests from test1 (0 ms total)

[-----] 1 test from test2
[ RUN     ] test2.suite1
[       OK ] test2.suite1 (0 ms)
[-----] 1 test from test2 (0 ms total)

[-----] 3 tests from fibonacciTest
[ RUN     ] fibonacciTest.num0
[       OK ] fibonacciTest.num0 (0 ms)
[ RUN     ] fibonacciTest.greater2
[       OK ] fibonacciTest.greater2 (0 ms)
[ RUN     ] fibonacciTest.negative
[       OK ] fibonacciTest.negative (0 ms)
[-----] 3 tests from fibonacciTest (0 ms total)

[-----] Global test environment tear-down
[=====] 6 tests from 3 test cases ran. (0 ms total)
[ PASSED ] 6 tests.
kulakov@ultrabook:~/projects/testing/qmake-gtest$
```

# Сообщение об ошибке

fibonacci\_test.h (tests @ qmake-gtest) - Qt Creator

Файл Плавка Сборка Отладка Анализ Инструменты Окно Справка

Проекты qmake-gtest  
qmake-gtest.pro  
app  
tests  
tests.pro  
gtest\_dependency  
Заголовочные  
Исходники

```
1 #ifndef FIBONACHI_H
2 #define FIBONACHI_H
3
4 #include <gtest/gtest.h>
5
6 extern "C" {
```

Результаты тестирования

Test summary: 5 успехов, 1 ошибок.

- PASS Выполнение теста test1
- PASS Выполнение теста test2
- FAIL Выполнение теста fibonacciTest
  - PASS fibonacciTest.num0  
Выполнение заняло 0 ms. fibonacci\_test.h 10
  - FAIL fibonacciTest.greather2  
.../gtest-master/tests/fibonacci\_test.h:17: Failure  
Expected equality of these values:  
 fibonacci(5)  
 Which is: 5  
 2  
Выполнение заняло 0 ms. fibonacci\_test.h 16
  - PASS fibonacciTest.negative  
Выполнение заняло 0 ms. fibonacci\_test.h 21

Выполнение теста заняло 0 ms total fibonacci\_test.h 21

Открытые документы  
fibonacci\_test.h

qmak...test  
Отладка

Быстрый поиск (...)

1 Проблемы 2 Результаты поиска 3 Вывод приложения 4 Консоль сборки 5 Консоль отладчика 8 Результаты тестирования

# Сообщение об ошибке

```
kulakov@ultrabook: ~/projects/testing/qmake-gtest
Файл Правка Вид Поиск Терминал Справка
kulakov@ultrabook:~/projects/testing/qmake-gtest$ ./tests/tests
[=====] Running 6 tests from 3 test cases.
[-----] Global test environment set-up.
[-----] 2 tests from test1
[ RUN     ] test1.suite1
[ OK      ] test1.suite1 (0 ms)
[ RUN     ] test1.suite2
[ OK      ] test1.suite2 (0 ms)
[-----] 2 tests from test1 (0 ms total)

[-----] 1 test from test2
[ RUN     ] test2.suite1
[ OK      ] test2.suite1 (0 ms)
[-----] 1 test from test2 (0 ms total)

[-----] 3 tests from fibonacciTest
[ RUN     ] fibonacciTest.num0
[ OK      ] fibonacciTest.num0 (0 ms)
[ RUN     ] fibonacciTest.greater2
fibonacci_test.h:17: Failure
Expected: fibonacci(5)
Which is: 5
To be equal to: 2
[ FAILED  ] fibonacciTest.greater2 (0 ms)
[ RUN     ] fibonacciTest.negative
[ OK      ] fibonacciTest.negative (0 ms)
[-----] 3 tests from fibonacciTest (0 ms total)

[-----] Global test environment tear-down
[=====] 6 tests from 3 test cases ran. (0 ms total)
[ PASSED  ] 5 tests.
[ FAILED  ] 1 test, listed below:
[ FAILED  ] fibonacciTest.greater2

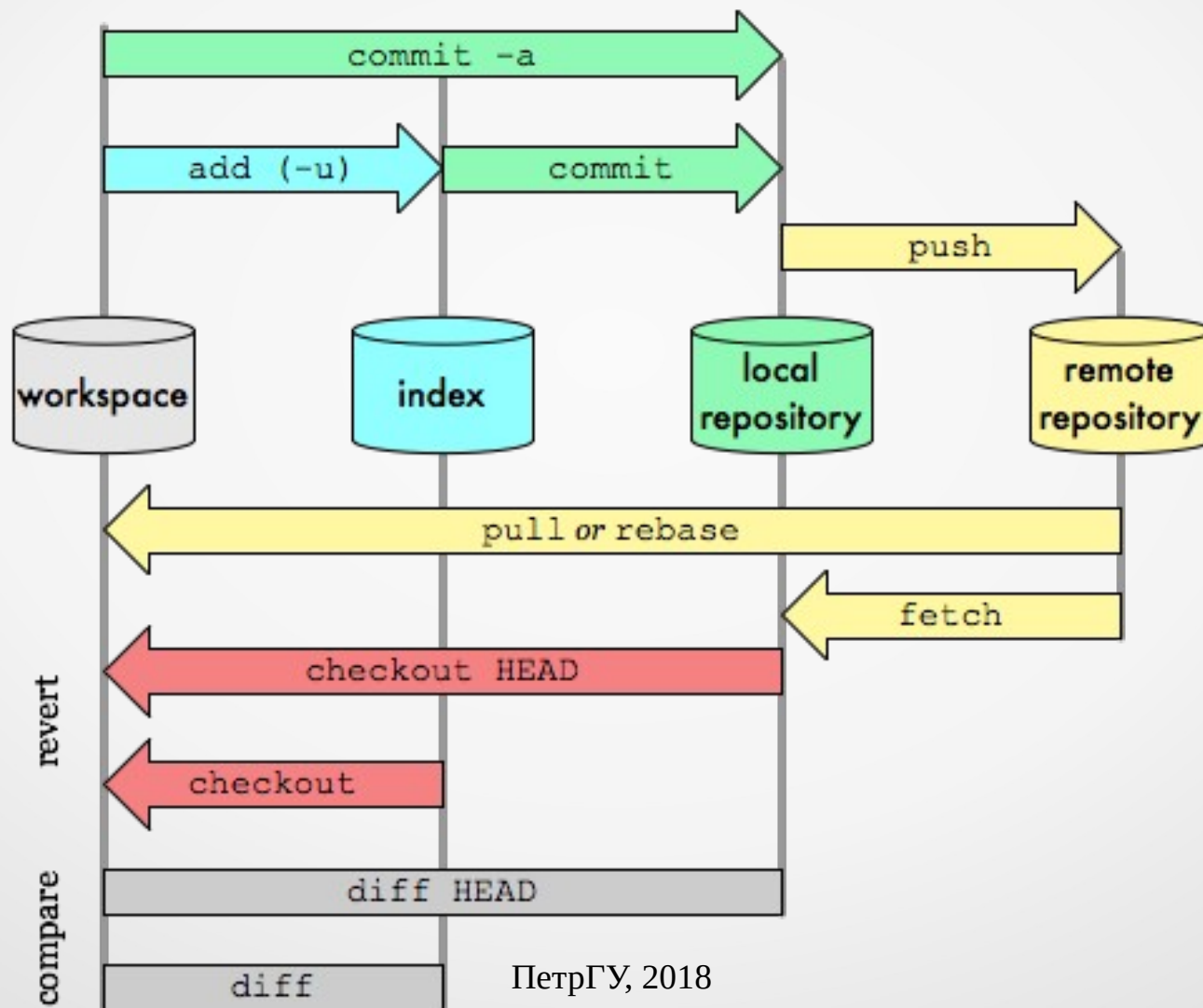
1 FAILED TEST
kulakov@ultrabook:~/projects/testing/qmake-gtest$
```



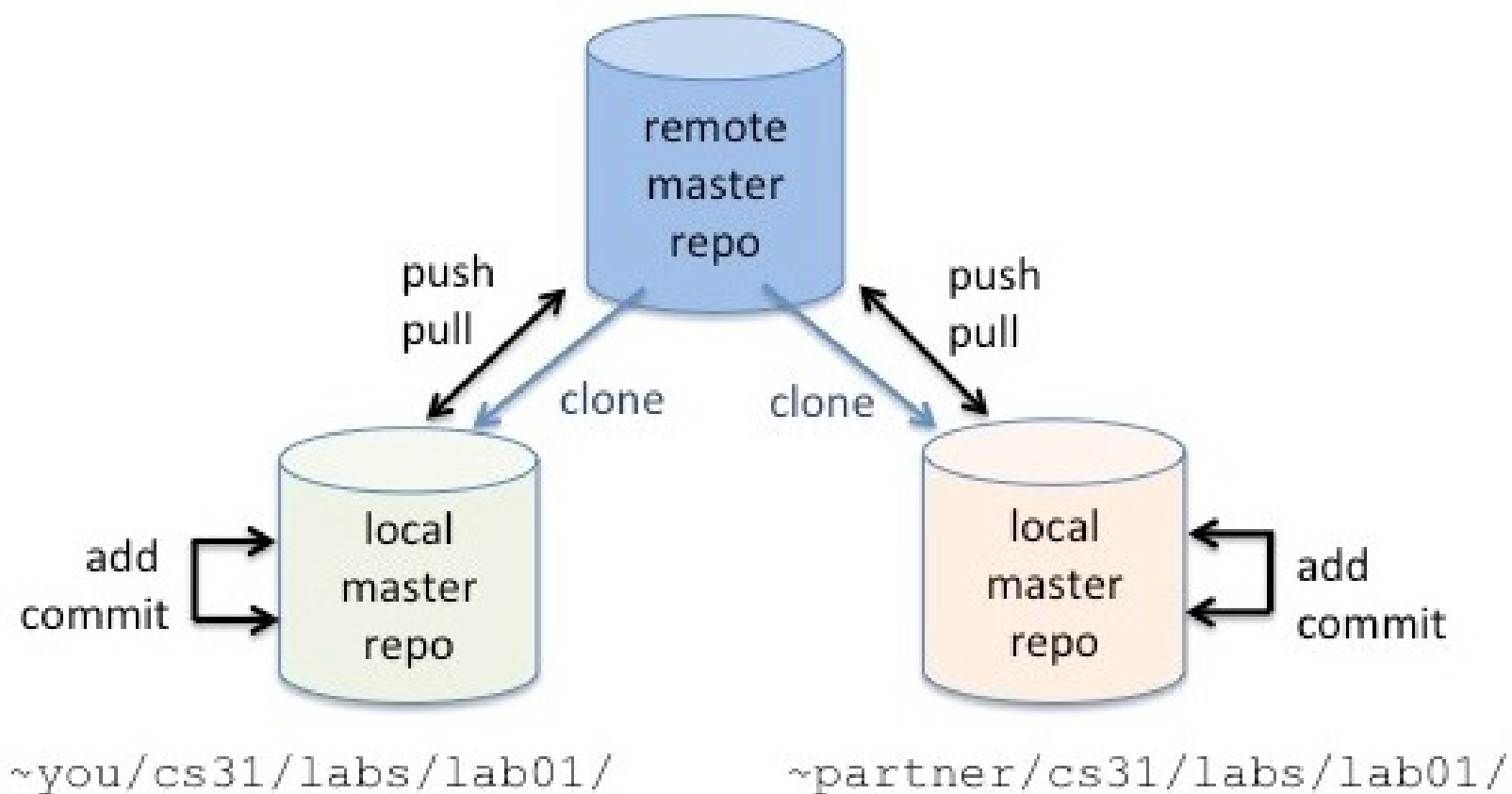
# Git

## Git Data Transport Commands

<http://osteele.com>



# Git



<https://git-scm.com/book/ru/v1/>

# Github.com

<> Code

! Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

📊 Insights

⚙️ Settings

No description, website, or topics provided.

Edit

Manage topics

🕒 2 commits

🌿 1 branch

📦 0 releases

👤 0 contributors

Branch: master ▾


New pull request

Create new file

Upload files

Find file

Clone or download ▾

 <b>Vyacheslav Dimitrov</b> Fixed get code.	Latest commit bcecc2f 2 minutes ago
📁 <a href="#">app</a>	test 9 minutes ago
📁 <a href="#">external/googletest</a>	test 9 minutes ago
📁 <a href="#">tests</a>	test 9 minutes ago
📄 <a href="#">.gitignore</a>	test 9 minutes ago
📄 <a href="#">.gitmodules</a>	test 9 minutes ago
📄 <a href="#">.travis.yml</a>	test 9 minutes ago
📄 <a href="#">README.md</a>	test 9 minutes ago
📄 <a href="#">get_code_cov.sh</a>	Fixed get code. 2 minutes ago
📄 <a href="#">qmake-gtest.pro</a>	test 9 minutes ago

📖 [README.md](#) 



# travis-ci

seekerk / gtest  build passing




[Current](#) [Branches](#) [Build History](#) [Pull Requests](#)

More options 

✓ **master** add sonarcloud badges

✓ #27 passed

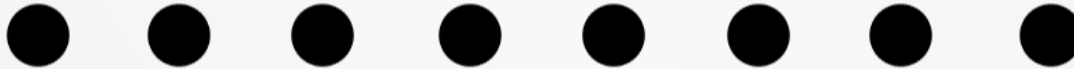
Restart build

- Commit 11923bc 
- Compare 1e01c29...11923bc 
- Branch master 

Ran for 3 min 42 sec  
4 months ago

 Kirill Kulakov

 C++



```
546
547 [-----] Global test environment tear-down
548 [=====] 7 tests from 3 test cases ran. (0 ms total)
549 [ PASSED ] 7 tests.
550
551
552 The command "./tests/tests" exited with 0.
▶ 553 store build cache
▶ 557 $ coveralls --root . -E ".*external.*" -E ".*tests.*"
▶ 1154 $ build-wrapper-linux-x86-64 --out-dir bw-output make clean all
▶ 1180 $ sonar-scanner
1282
1283 Done. Your build exited with 0.
```

cache.2	
after_success.1	1.61s
after_success.2	13.52s
after_success.3	41.78s

Top 

# .travis.yml

language:

- cpp

script:

- qmake
- make
- ./tests/tests

notifications:

email: false

# coveralls

## COVERALLS

Settings  
Repo on GitHub

### SEEKERK / GTEST

100%

DEFAULT BRANCH: MASTER

REPO ADDED  
01 JUN 2017 05:54AM  
UTC

TOTAL FILES  
3

# BUILDS  
20

BADGE  
coverage 100%

TOKEN  
WSa29FSUxvUJka49cNEoh5z

### LAST BUILD ON BRANCH MASTER

BRANCH: MASTER

COMMITTED 21 JUN 2018 - 8:44

COVERAGE REMAINED THE SAME AT 100.0%

BUILD #	BUILD TYPE	COMMITTED BY	COMMIT MESSAGE	RUN DETAILS
27	push travis-ci	seekerk	add sonarcloud badges	9 of 9 relevant lines covered (100.0%) 9.22 hits per line

# coveralls

```
1  #include "level.h"
2  #include <math.h>
3  using namespace std;
4  void Level::Level_opt(int leftBorderVal, int upperBorderVal,           !
5                       int rightBorderVal, int lowerBorderVal,
6                       int heroX, int heroY,
7                       unsigned int heroMaxHealth, unsigned int heroMaxLevel,
8                       int traderX, int traderY)
9  {
10     leftBorder = leftBorderVal;           !
11     upperBorder = upperBorderVal;        !
12     rightBorder = rightBorderVal;        !
13     lowerBorder = lowerBorderVal;        !
14     mainHero = new HeroObject(heroX, heroY, heroMaxHealth, heroMaxLevel); !
15     trader = new TraderObject(traderX, traderY); !
16 }
17 Level::Level(int leftBorderVal, int upperBorderVal,                   36x
18              int rightBorderVal, int lowerBorderVal,
19              int heroX, int heroY,
20              unsigned int heroMaxHealth, unsigned int heroMaxLevel,
21              int traderX, int traderY)                                     36x
22 {
23     leftBorder = leftBorderVal;           36x
```

# Coveralls + .travis.yml

language:

- cpp

**before\_install:**

- **pip install --user cpp-coveralls**

script:

- qmake
- make
- ./tests/tests

**after\_success:**

- **coveralls --root . -E ".\*external.\*" -E ".\*tests.\*"**

# sonarcloud

The screenshot shows the SonarCloud dashboard for a project named 'gtest' by user 'Kirill Kulakov'. The interface includes a navigation bar with 'Overview', 'Issues', 'Security Reports', 'Measures', 'Code', 'Activity', and 'Administration'. The main content area is divided into several sections:

- Quality Gate:** A green badge indicates the quality gate has 'Passed'.
- Bugs:** 1 bug (Error level).
- Vulnerabilities:** 0 vulnerabilities (All level).
- Code Smells:** 40min Debt, 4 Code Smells.
- Coverage:** 0.0% Coverage.
- Duplications:** 0.0% Duplications.

Summary statistics for 'New code: last 30 days' are shown in a yellow box:

- New Bugs: 0 (All level)
- New Vulnerabilities: 0 (All level)
- New Debt: 0 (All level)
- New Code Smells: 0 (All level)

On the right side, the 'About This Project' section shows:

- Public project with no tags.
- 103 Lines of Code (C: 95, C++: 8).
- Project Activity graph showing a sharp drop in activity on June 21, 2018.
- Recent events: 'Quality Profile: Stop using 'Sonar way' (Python)' and 'Project Analyzed' on June 21, 2018.
- Quality Gate: (Default) Sonar way.
- Quality Profiles: (C) Sonar way, (C++) Sonar way.
- Project Key: qmake-gtest (Copy).
- Organization Key: seekerk-github (Copy).

# Sonarcloud + .travis.yml

language:

- cpp

before\_install:

- pip install --user cpp-coveralls

**addons:**

**sonarcloud:**

- organization:** "seekerk-github"

- token:**

  - secure:** \$SONAR\_TOKEN

script:

- qmake

- make

- ./tests/tests

after\_success:

- coveralls --root . -E ".\*external.\*" -E ".\*tests.\*"

- **build-wrapper-linux-x86-64 --out-dir bw-output make clean all**

- **sonar-scanner**

**cache:**

**directories:**

- '\$HOME/.sonar/cache'

notifications:

- email: false