

Введение в тестирование

# Автоматизация тестирования

Кулаков Кирилл Александрович

Петрозаводск — 2017

# Будни тестирования

- Проект большой → много тестов → ручной запуск
- Демонстрация работы / качества приложения
- Демонстрация ошибки разработчику
- Регрессионное тестирование
- Несколько проектов одновременно → необходимость переключаться
- Текучка кадров / неопределенный круг разработчиков
- Отчет о проекте начальству / заказчикам

# Задача автоматизации тестирования

- Цель: сокращение издержек тестирования
  - уменьшение ручной работы
  - уменьшение человеческого фактора
  - инструмент для проверки нового кода
  - Составление отчетов руководству → онлайн режим отчета
- Недостатки
  - накладные расходы на создание тестов
  - накладные расходы на автоматизацию
  - накладные расходы на поддержку системы тестирования

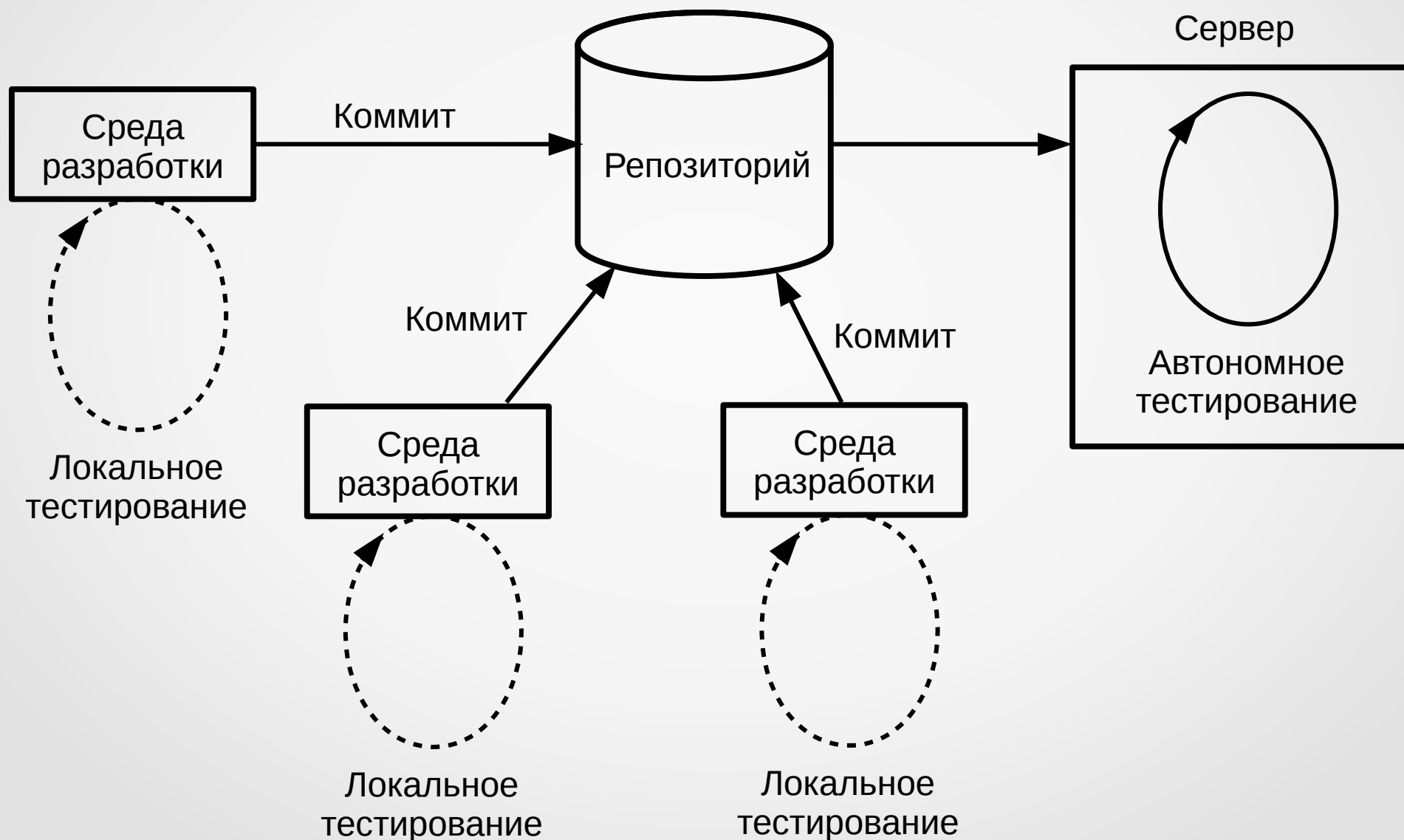
# Варианты автоматизации

- Запуск тестов при необходимости
  - скрипт для ручного запуска
  - разбиение тестов на группы
- Запуск тестов после события
  - триггеры для выбранных событий (коммиты, запросы на слияния, ... )
  - выделенные ресурсы (сервера)
- Систематический запуск тестов
  - планировщик задач (ночные тесты, еженедельные тесты, ... )

# Непрерывная интеграция

- практика разработки программного обеспечения
- слиянии рабочих копий в общую основную ветвь разработки несколько раз в день
- выполнении частых автоматизированных сборок проекта
- Цель: скорейшее выявление потенциальных дефектов и решение интеграционных проблем.

# Непрерывная интеграция



# Непрерывная интеграция

- Преимущества
  - проблемы интеграции выявляются и исправляются быстро, что оказывается дешевле;
  - немедленный прогон модульных тестов для свежих изменений;
  - постоянное наличие текущей стабильной версии вместе с продуктами сборки — для тестирования, демонстрации, и т.д.
  - немедленный эффект от неполного или неработающего кода приучает разработчиков к работе в итеративном режиме с более коротким циклом.

# Непрерывная интеграция

- Недостатки
  - затраты на поддержку работы непрерывной интеграции;
  - потенциальная необходимость в выделенном сервере под нужды непрерывной интеграции;
  - немедленный эффект от неполного или неработающего кода отучает разработчиков от выполнения периодических резервных включений кода в репозиторий.
    - использование ветвей для новых изменений → увеличение расхода ресурсов



# Реализация автоматизации тестирования

- Инструменты:
  - qt creator (среда разработки + локальное тестирование)
  - google test (создание тестов)
  - github (репозиторий кода)
  - travis-ci (непрерывная сборка)

# Создание тестов для автоматического тестирования

- Задача: обеспечить автономность запуска
  - доступность исходных данных
  - доступность окружения объекта тестирования (других модулей / программ / ...)
  - возможность получить результат работы теста
  - независимость тестов друг от друга и от истории запусков
- Выводы
  - все что нужно для теста в репозиторий
  - поиск "маркеров" в результате работы объекта тестирования
  - удаление временных файлов по завершении работы теста

# Travis-CI

- Мониторинг пушей в репозиторий (github)
- Запуск скрипта сборки (.travis.yml) свободным обработчиком на чистой виртуальной машине
- Контроль результата по кодам возврата
- Предоставление бейджей с результатами сборки
- Сайт проекта
  - <https://travis-ci.org> - open source проекты
  - <https://travis-ci.com> - приватные проекты
- Документация <https://docs.travis-ci.com/>

# .travis.yml

- Проверка с помощью `travis-lint`: [github.com/travis-ci/travis-lint](https://github.com/travis-ci/travis-lint)

## **language:**

- **cpp**

## **before\_install:**

- `pip install --user cpp-coveralls`

## **script:**

- **qmake**
- **make**
- **./tests/tests**

## **after\_success:**

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

## **notifications:**

`email: false`

# Результат работы

The screenshot shows a web browser window with the title "seekerk/gtest - Travis CI - Mozilla Firefox". The address bar displays "Travis CI GmbH (DE) | https://travis-ci.org/seekerk/gtest". The page content includes a navigation menu with "Current", "Branches", "Build History", and "Pull Requests". A green checkmark indicates a successful build on the "master" branch for the commit "fix path for shadow builds". The build status is "#16 passed" and it ran for 45 seconds, 4 months ago. The commit was authored and committed by Kirill Kulakov. Below the build summary, there are links for "Job log" and "View config". The job log is expanded, showing system information such as the worker mode, build system details, and network configuration.

seekerk/gtest build passing

Current Branches Build History Pull Requests More options

✓ **master** fix path for shadow builds ↻ #16 passed ↻ Restart build

↻ Commit d90096a ↻  
↻ Compare 0fef4a6..d90096a ↻  
↻ Branch master ↻

Kirill Kulakov authored and committed

[Job log](#) [View config](#)

✕ Remove log ⌵ Raw log

```
1 Worker information worker_info
6 mode of '/usr/local/clang-5.0.0/bin' changed from 0777 (rwxrwxrwx) to 0775 (rwxrwxr-x)
7 Build system information system_info
404
405 removed '/etc/apt/sources.list.d/basho_riak.list'
406 W: http://ppa.launchpad.net/couchdb/stable/ubuntu/dists/trusty/Release.gpg: Signature by key
15866BAFD9BCC4F3C1E0DFC7D69548E1C17EAB57 uses weak digest algorithm (SHA1)
407 127.0.0.1 localhost
408 ::1 ip6-localhost ip6-loopback
409 fe00::0 ip6-localnet
410 ff00::0 ip6-mcastprefix
411 ff02::1 ip6-allnodes
412 ff02::2 ip6-allrouters
413 172.17.0.19 travis-job-seekerkerk-gtest-333145060.travisci.net travis-job-seekerkerk-gtest-333145060
```

# Результат ошибки

Build #6 - seekerk/gtest - Travis CI - Mozilla Firefox

Файл Правка Вид Журнал Закладки Инструменты Справка


Build #6 - seekerk/g x +

Travis CI GmbH (DE) | https://travis-ci.org/seekerk/gtest/b... ☆ ⏴ ⏵

Current Branches Build History Pull Requests > Build #6 More options

**✖ master add gcov** **↪ #6 failed** [Restart build](#)

↪ Commit 37cebe1 [↗](#) [↻](#) Ran for 53 sec  
↪ Compare ece1316..37cebe1 [↗](#) [📅](#) 12 months ago  
↪ Branch master [↗](#)

 Kirill Kulakov authored and committed

[Job log](#) [View config](#)

[Remove log](#) [Raw log](#)

```
1 Worker information worker_info
6 Build system information system_info
73
74 $ export DEBIAN_FRONTEND=noninteractive fix.CVE-2015-7547
110 $ sudo apt-get install libssl1.0.0 update_libssl1.0.0
134 $ git clone --depth=50 --branch=master https://github.com/seekerk/gtest.git seekerk/gtest git.checkout 0.33s
144 $ git submodule update --init --recursive git.submodule 0.84s
155 $ export CXX=g++
156 $ export CC=gcc
157 $ g++ --version
158 g++ (Ubuntu/Linaro 4.6.3-1ubuntu5) 4.6.3
159 Copyright (C) 2011 Free Software Foundation, Inc.
160 This is free software; see the source for copying conditions. There is NO
161 warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

## Build #6 - seekerk/gtest - Travis CI - Mozilla Firefox

Build #6 - seekerk/g x +

Travis CI GmbH (DE) | <https://travis-ci.org/seekerk/gtest/b>

```
227 [ OK ] test1.suite2 (0 ms)
228 [-----] 2 tests from test1 (0 ms total)
229
230 [-----] 1 test from test2
231 [ RUN ] test2.suite1
232 [ OK ] test2.suite1 (0 ms)
233 [-----] 1 test from test2 (0 ms total)
234
235 [-----] 3 tests from fibonacciTest
236 [ RUN ] fibonacciTest.num0
237 [ OK ] fibonacciTest.num0 (0 ms)
238 [ RUN ] fibonacciTest.greater2
239 fibonacci_test.h:17: Failure
240     Expected: fibonacci(5)
241     Which is: 5
242 To be equal to: 1 + 1 + 2 + 3 + 5
243     Which is: 12
244 [ FAILED ] fibonacciTest.greater2 (0 ms)
245 [ RUN ] fibonacciTest.negative
246 fibonacci_test.h:22: Failure
247     Expected: fibonacci(-1)
248     Which is: -1
249 To be equal to: 0
250 [ FAILED ] fibonacciTest.negative (0 ms)
251 [-----] 3 tests from fibonacciTest (0 ms total)
252
253 [-----] Global test environment tear-down
254 [=====] 6 tests from 3 test cases ran. (0 ms total)
255 [ PASSED ] 4 tests.
256 [ FAILED ] 2 tests, listed below:
257 [ FAILED ] fibonacciTest.greater2
258 [ FAILED ] fibonacciTest.negative
259
260 2 FAILED TESTS
261
262
263 The command "./tests/tests" exited with 1.
264 $ /get_code cov.sh
```

# Другие среды для автоматизации

- <https://circleci.com/> - CircleCI
- <https://jenkins.io/> - Jenkins
- <https://www.jetbrains.com/teamcity/> - TeamCity
- <https://codeship.com/> - Codeship
- <https://gitlab.com> - Gitlab



circleci



**Jenkins**



GitLab

