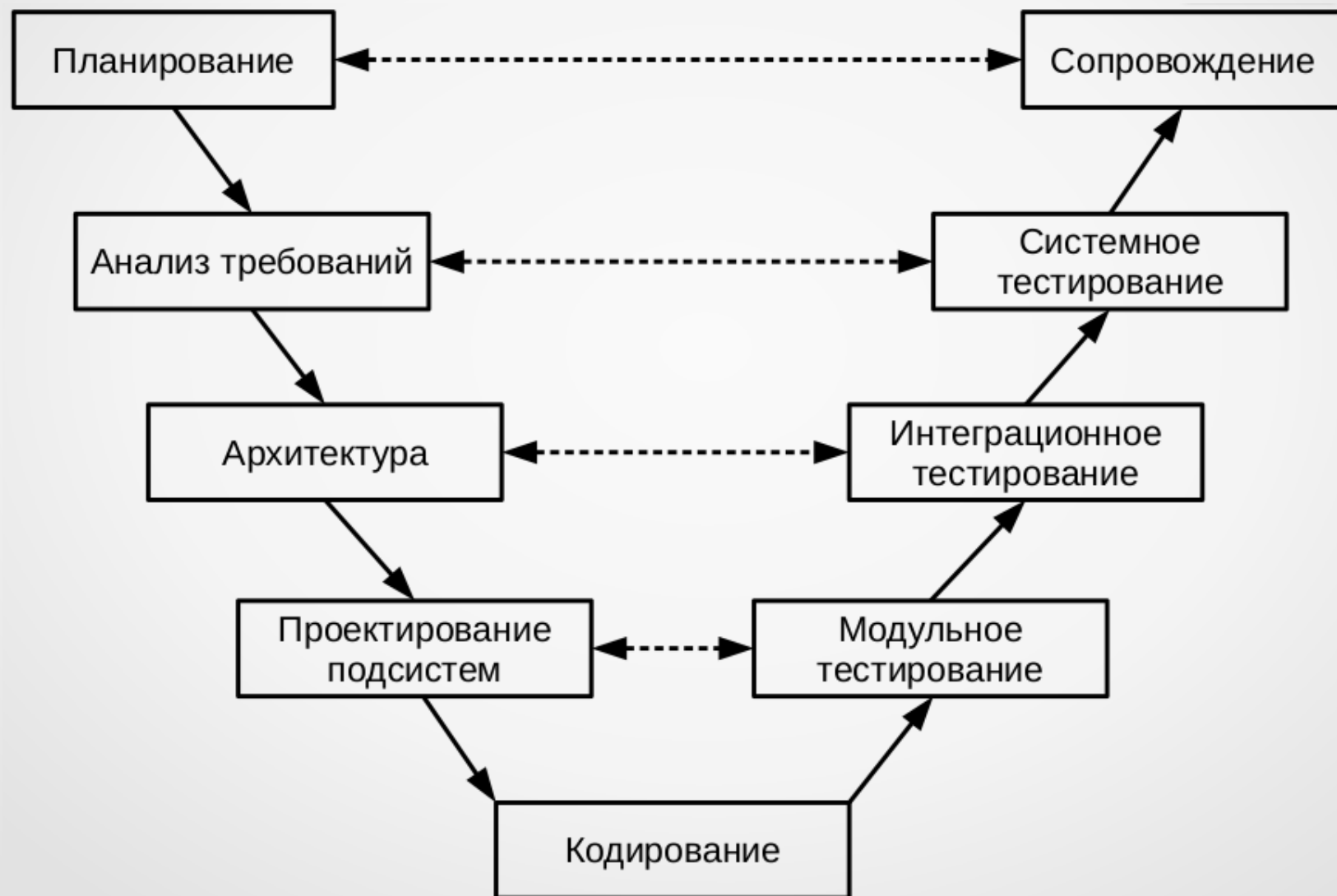


Основы разработки ПО

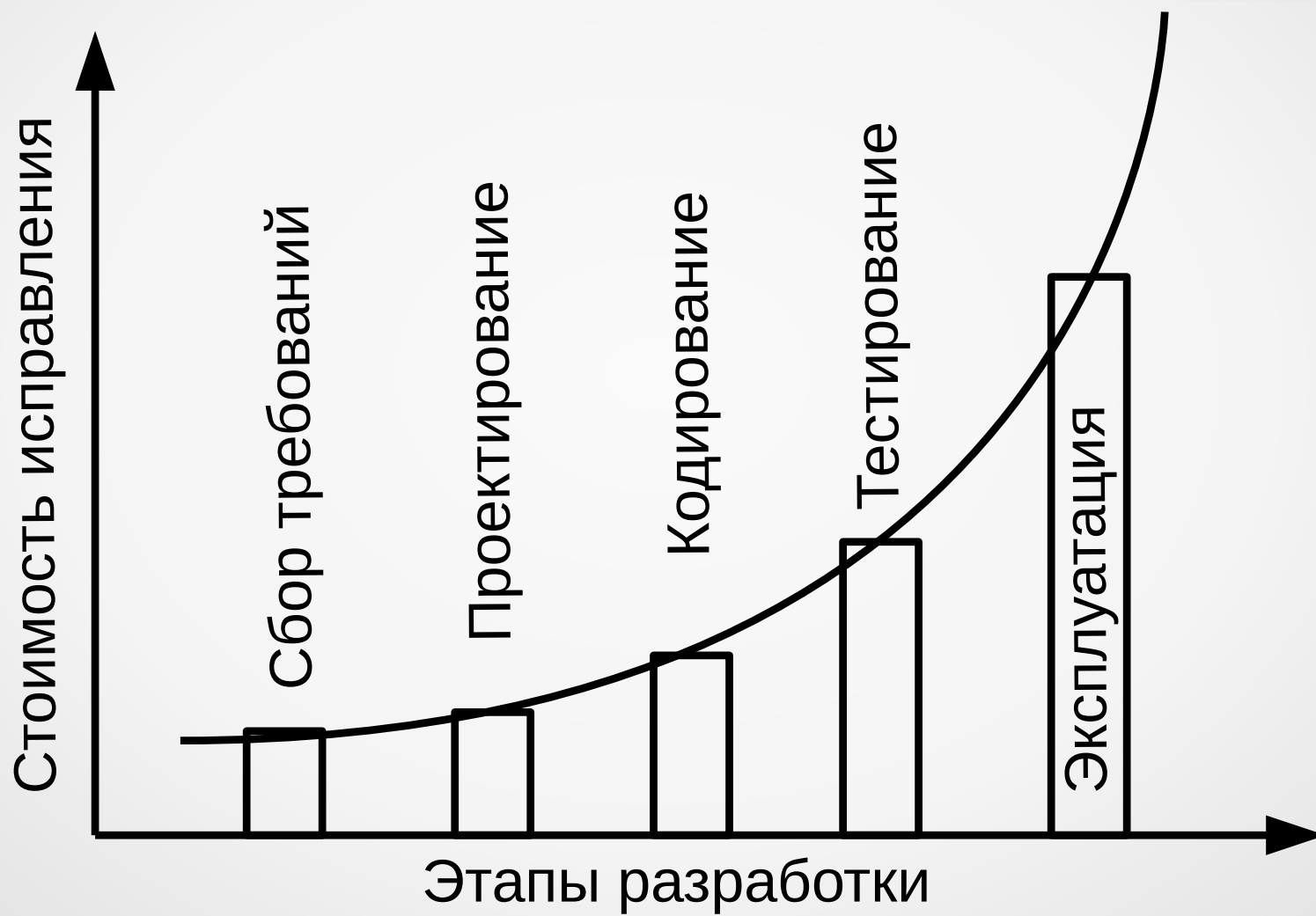
Обзор программного инструментария для разработки ПО

Кулаков Кирилл Александрович

V-модель разработки ПО

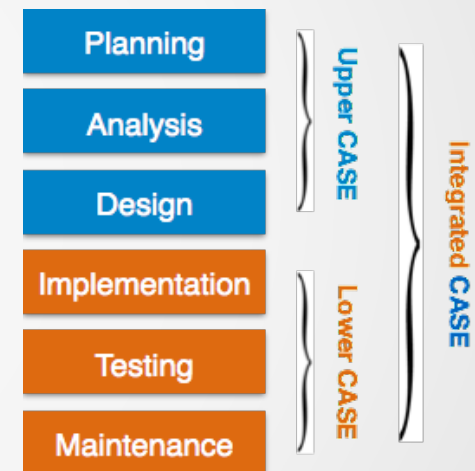


Стоимость исправления ошибки



Инструментальные средства

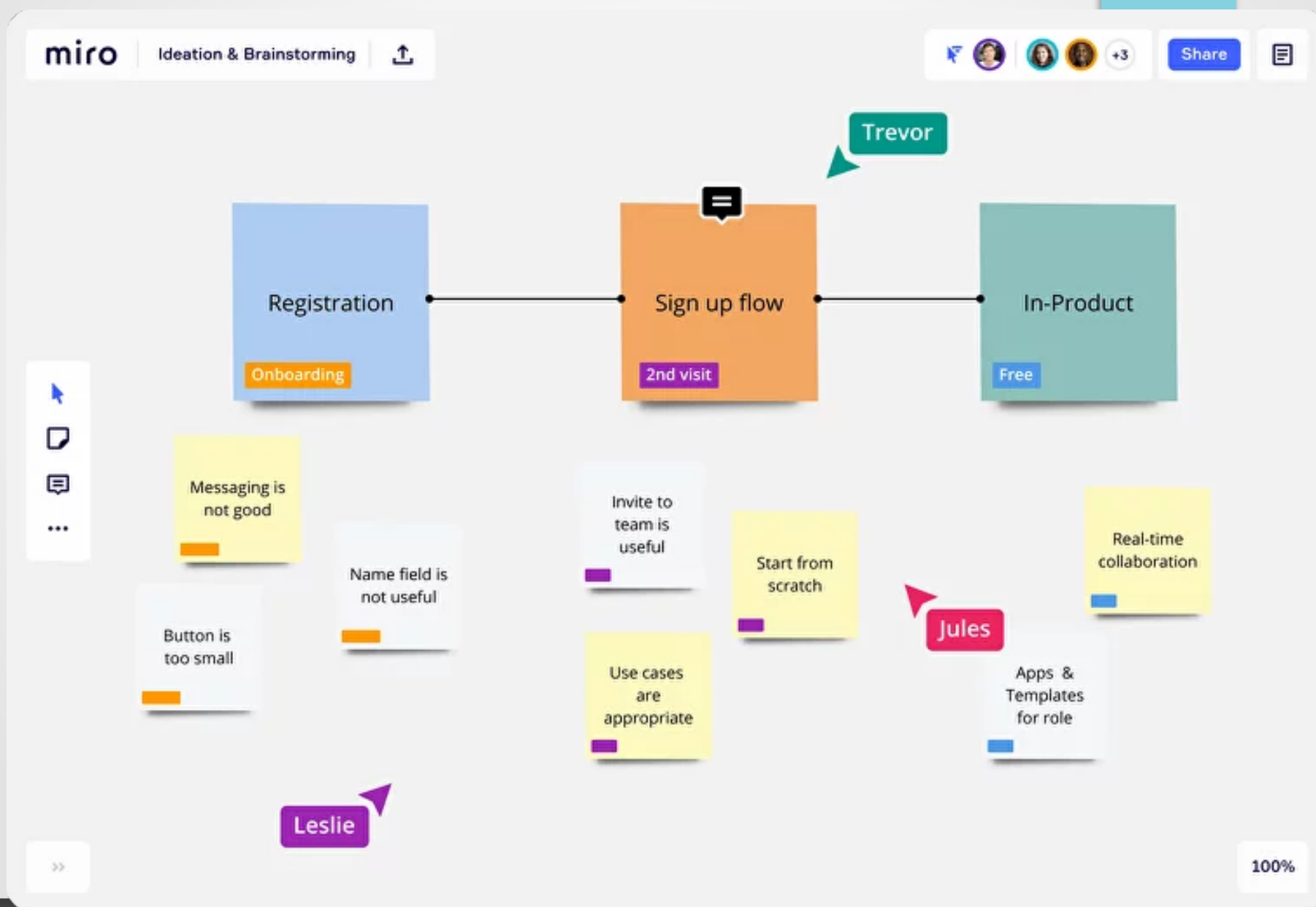
- Computer Aided Software Engineering (CASE)
- Автоматизация выполнения монотонных, рутинных операций
- Единообразиие результата
- Контроль работы
- Ускорение выполнения операций
- Пример классификации:
 - Верхние (Upper) CASE средства — планирование
 - Средние (Middle) CASE средства — проектирование
 - Нижние (Lower) CASE средства — реализация и тестирование



Средства планирования

- Организация взаимодействия между участниками проекта
 - Коммуникация
 - Передача артефактов
 - Совместная работа
 - Фиксация результатов
- Мессенджеры (Skype, Telegram, Сбер Jazz,...)
- Сетевые сервисы (Google docs, Miro, ...)

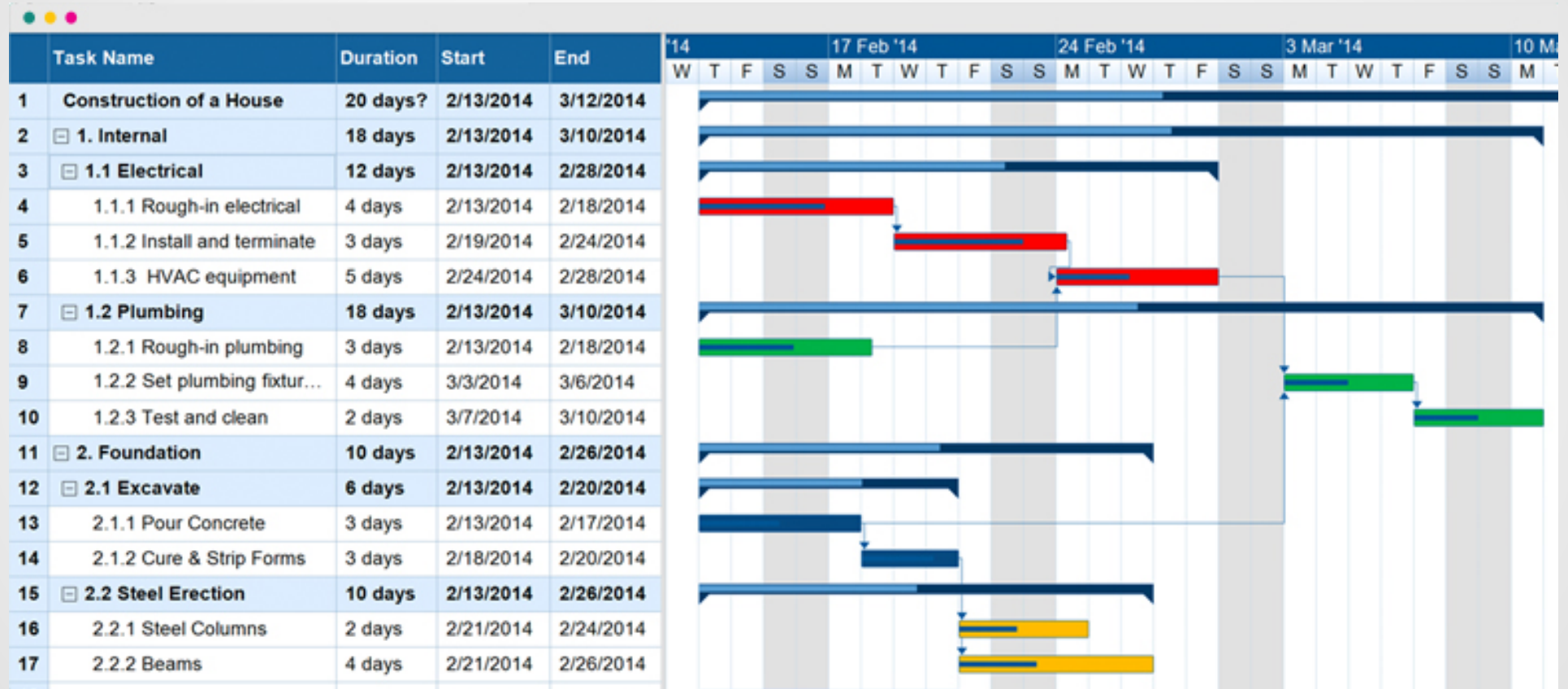
Средства планирования



Средства планирования

- Планировщики задач
 - Организация работ
 - Определение графика работ
 - Организация взаимодействия
- Диаграммы Ганта и Петра
- Встроенные планировщики задач
- Сетевые сервисы (Trello, ...)


Средства планирования



Средства планирования

Taco's Tacos ☆ Taco & Co. Free Team Visible Show Menu

Resources



Financials & Growth Data
☰ 🔗 5

2017 Goals And KPIs
☰ 🔗 2

Brand Guide
☰ 🔗 1


Employee Manual
☰ 🔗 1

Add a card...

To Do

Build A Better Burrito: 7 Layers To Success
☰ ☑ 0/7

Nacho Ordinary Birthday - Event Space Rentals
☰




Taco Drone Delivery Service
🕒 Nov 10 🔗 3

Superbowl Ad - "Super Salad Bowls"
🕒 Dec 12 ☰

Add a card...

Doing

The Taco Truck World Tour
🕒 Oct 5 ☰



Operation "Awesome Sauce" - A Recipe For Profit
🕒 Oct 18 ☰ 🔗 3 ☑ 2/5

#NoFiller Instagram Campaign
🔗 3

Global Franchise Opportunities
☰ ☑ 4/9

Add a card...


Done

Focus Group: Corn vs. Flour Tortillas
☰

New Swag: Socks, Scarves & Salsa
🔗 5

Eco Friendly Utensils & Napkins
☑ 3/3

Update Yelp Listing
☰ 🔗 1



Grand Opening Celebration
🕒 Aug 11, 2016

Add a card...

Средства планирования

- Разработка и анализ требований
 - Редактирование документов и артефактов
 - Совместная работа
 - Версионирование
 - «Мозговые штурмы»
- Текстовые редакторы (Microsoft Word, ...)
- Графические редакторы (GIMP, ...)
- Сетевые сервисы (Google docs, Miro, ...)

Средства планирования

- Создание базы знаний
 - Накопление знаний
 - Структурирование знаний
 - Обмен знаниями
- CMS системы (Wiki, ...)
- Форумы (phpBB, ...)
- Сетевые диски (Гугл диск, Яндекс диск, ...)
- Сетевые сервисы (Google docs, Miro, ...)

Средства планирования

 [Log in / create account](#)

Set `$wgLogo`
to the URL
path to your
own logo
image.

Page [Discussion](#)

[Read](#)

[Edit](#)

[View history](#)

Main Page

MediaWiki has been successfully installed.

Consult the [User's Guide](#) for information on using the wiki software.

Getting started

[\[edit\]](#)

- [Configuration settings list](#)
- [MediaWiki FAQ](#)
- [MediaWiki release mailing list](#)

This page was last modified on 10 August 2011, at 10:57.

This page has been accessed 2 times.

[Privacy policy](#) [About My wiki](#) [Disclaimers](#)



Navigation

[Main page](#)
[Community portal](#)
[Current events](#)
[Recent changes](#)
[Random page](#)
[Help](#)

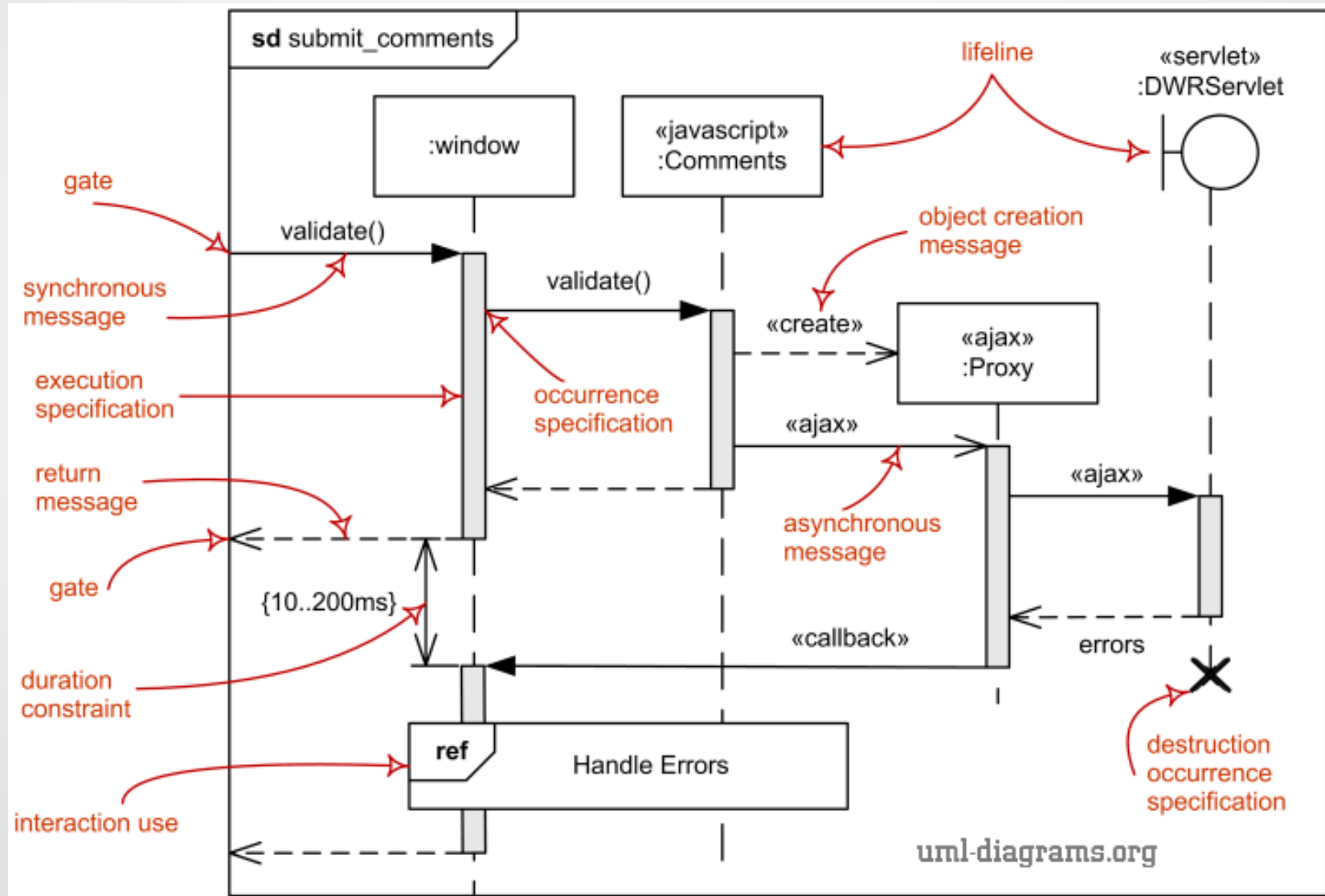
Toolbox

[What links here](#)
[Related changes](#)
[Special pages](#)
[Printable version](#)
[Permanent link](#)

Средства проектирования

- Редакторы диаграмм
 - Следование требуемой нотации
 - Визуальное отображение модели
 - Возможность совместной работы
 - Возможность генерации представлений (кода)
- Универсальные графические редакторы (GIMP, Paint, ...)
- Многопрофильные графические редакторы (dia)
- Специализированные редакторы (BPwin/ERwin, Visual Paradigm, ...)

Средства проектирования



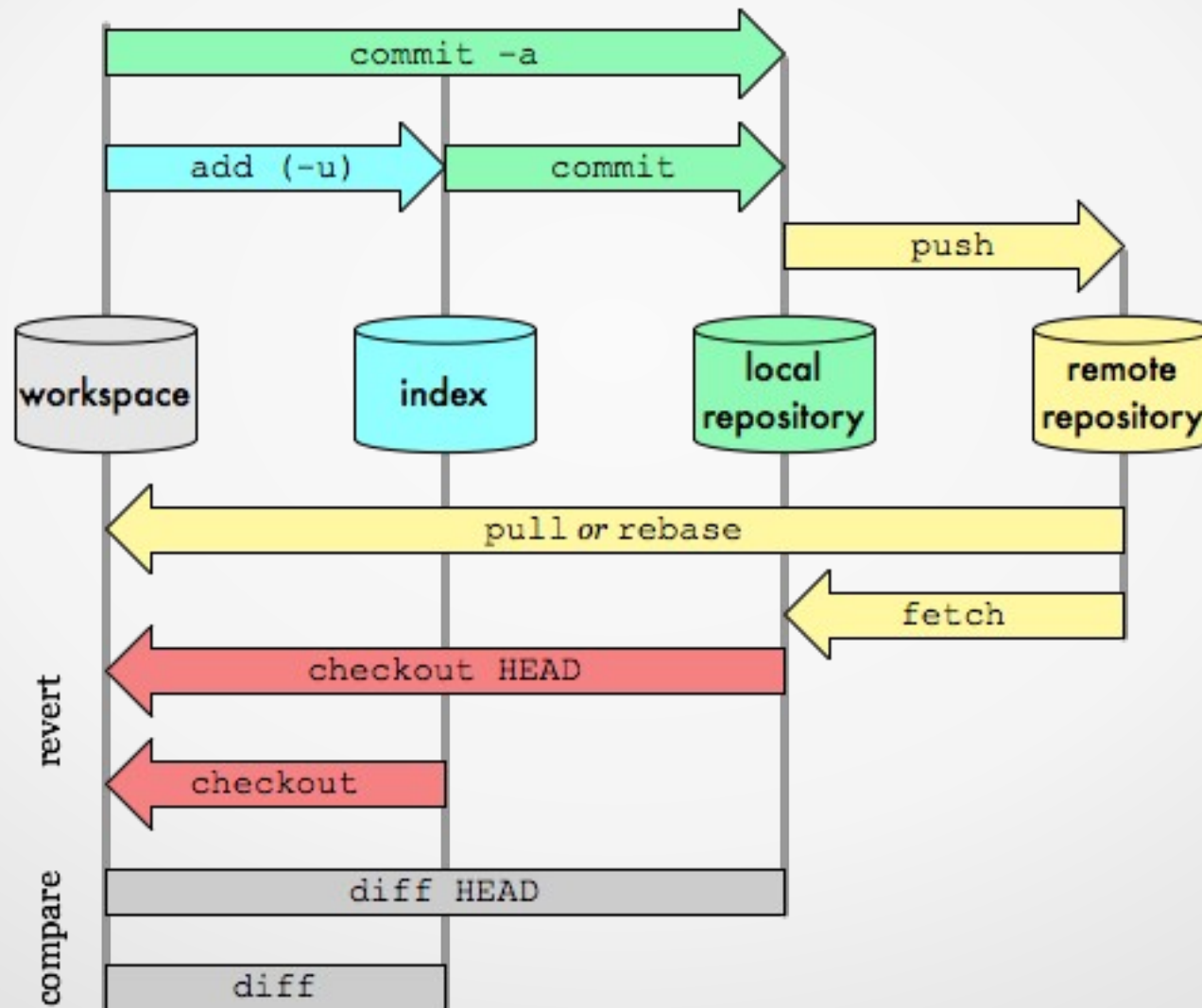
Средства проектирования

- Системы контроля версий
 - Отслеживание изменений данных
 - Совместная работа
 - История версий
- CVN, CVS, Git, Mercurial, Bazaar, ...

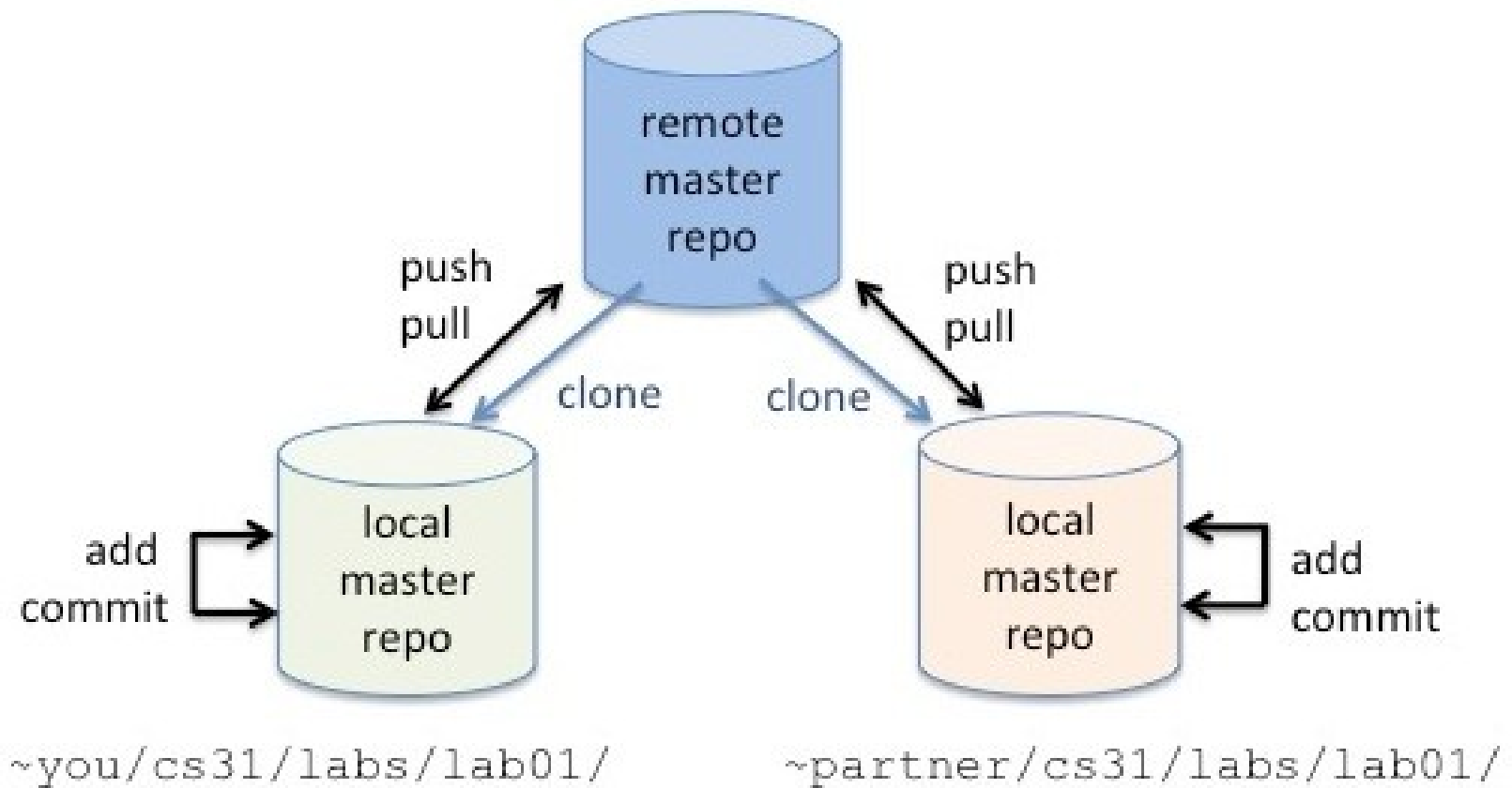
Git

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



Vyacheslav Dimitrov Fixed get code.

Latest commit bcecc2f 2 minutes ago



app

test

9 minutes ago



external/googletest

test

9 minutes ago



tests

test

9 minutes ago



.gitignore

test

9 minutes ago



.gitmodules

test

9 minutes ago



.travis.yml

test

9 minutes ago



README.md

test

9 minutes ago



get_code_cov.sh

Fixed get code.

2 minutes ago



qmake-gtest.pro

test

9 minutes ago



README.md








Github: Issues

<> Code **Issues** 10 Pull requests 1 Discussions Actions Projects Wiki Security Insights Settings

Filters Labels 6 Milestones 0 [New issue](#)

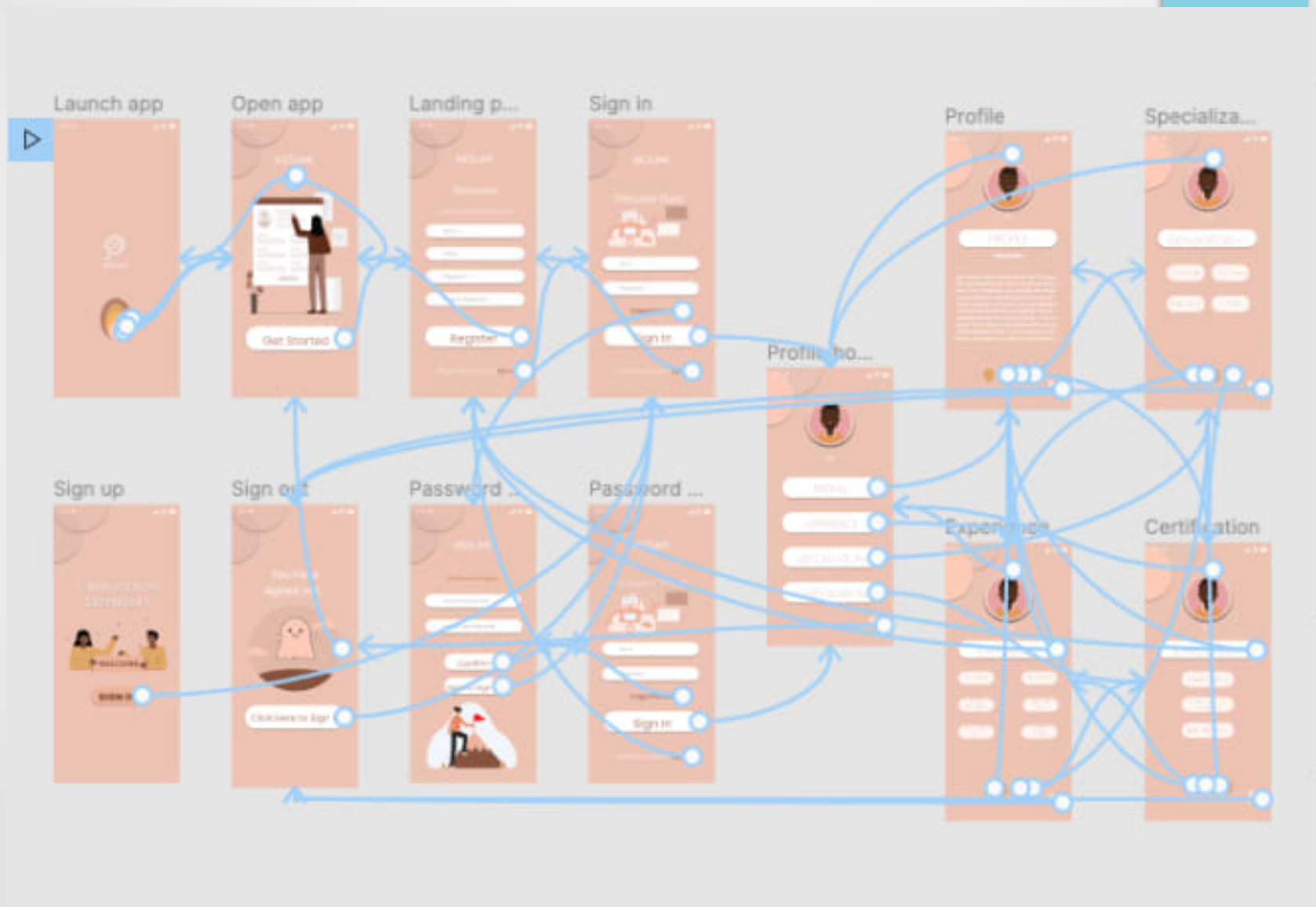
10 Open ✓ 27 Closed Author Label Projects Milestones Assignee Sort

- Отображать флаг публикации для категорий точек** enhancement  2
#36 opened on 20 Oct 2015 by seekerk
- Права админа и доверенного пользователя не проверяются для точек** 
#35 opened on 19 Jun 2015 by seekerk
- Сортировка треков и полигонов на сервере** enhancement 
#34 opened on 26 May 2015 by ivashov
- web: при небольшом смещении точки не обновляются** bug
#32 opened on 28 Apr 2015 by seekerk
- Получение точек и треков с учетом языка** enhancement 
#29 opened on 20 Apr 2015 by ivashov
- Рейтинг для точек в obstacles по умолчанию 0** 
#28 opened on 6 Apr 2015 by ivashov
- отслеживание сессий для GA**
#24 opened on 31 Mar 2015 by seekerk
- отслеживание уникальных пользователей в GA**
#23 opened on 31 Mar 2015 by seekerk
- Обработка пустого поля URL**
#20 opened on 16 Mar 2015 by seekerk
- Показывать радиус точки на карте**
#18 opened on 16 Mar 2015 by seekerk

Средства проектирования

- Разработка прототипов (UI/UX)
 - Быстрое создание модели
 - Визуальное представление
 - Возможность реализации примитивной логики
- Figma, InVision Studio, ...

Пример разработки прототипа



Средства реализации и тестирования

- Компиляторы/интерпретаторы
 - Перевод кода в понятный машине язык
 - Проверка синтаксиса
 - Проверка логики
 - Оптимизация кода
- Gcc, mingw, python, ...

Средства реализации и тестирования

- Инструменты автоматизации сборки
 - Контроль изменений кода
 - Автоматизация компиляции
 - Контроль за последовательностью действий
 - Выполнение дополнительных операций
- Gnu make
- Cmake, qmake, autotools, ...
- Ant, gradle, ...

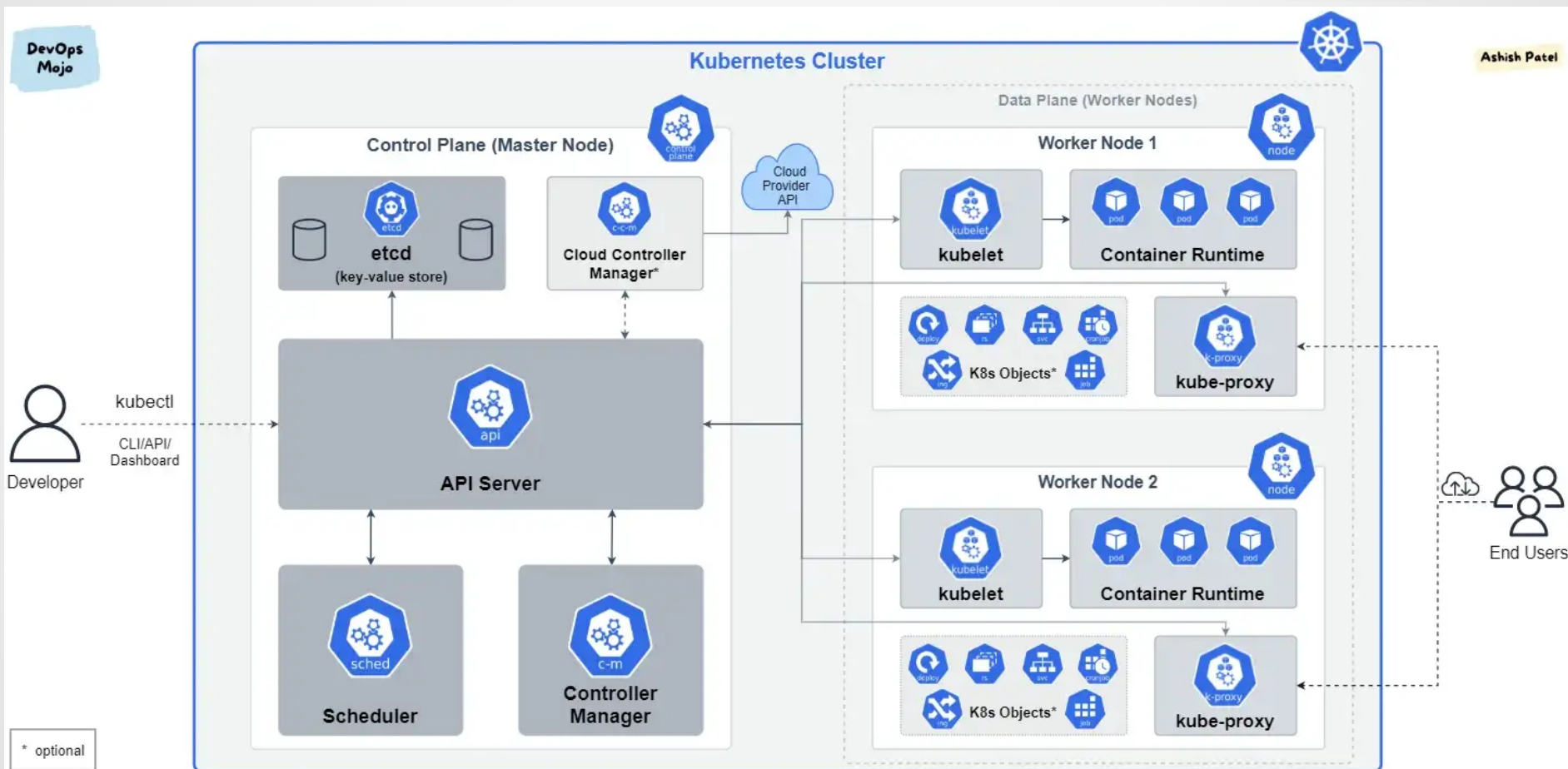
Средства реализации и тестирования

- Инструменты тестирования
 - Проверка кода
 - Проверка программных интерфейсов
- Встраиваемые инструменты — Google test, testunit, ...
- Внешние инструменты — Apache JMeter, ...

Средства реализации и тестирования

- Инструменты развертывания
 - Доставка приложения до места назначения
 - Контроль зависимостей
 - Контроль обновлений
 - Управление работой приложения
- Механизм контейнеров — Docker
- Механизм пакетов — deb, rpm, msi

Средства реализации и тестирования



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
 - fibonacci_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 ANSI C 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

The screenshot displays the Qt Creator IDE interface for a project named 'qmake-gtest'. The main window shows the 'Results of testing' (Результаты тестирования) panel, which contains a 'Test summary' (Test summary: 6 successes, 0 errors) and a detailed list of test results. The results are organized into three main categories: 'test1', 'test2', and 'fibonacciTest'. Each category shows a 'PASS' status and a list of sub-tests with their respective execution times and file locations.

Открыть документ

- Файл > Открыть файл или проект (Ctrl+O)
- Файл > Недавние файлы

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.greater2	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

Открытые документы

qmak..._test
Отладка

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

The screenshot shows the Qt Creator IDE with the following components:

- Project Explorer:** Shows a project named 'qmake-gtest' with sub-projects 'app' and 'tests'. The 'tests' sub-project contains 'tests.pro', 'gtest_dependency', 'Заголовочные', and 'Исходники'.
- Editor:** Displays the file 'fibonacci_test.h' with the following code:

```
1 #ifndef FIBONACHI_H
2 #define FIBONACHI_H
3
4 #include <gtest/gtest.h>
5
6 extern "C" {
```
- Test Results Panel:** Shows the following test summary:
 - Test summary: 5 успехов, 1 ошибок.
 - PASS: Выполнение теста test1
 - PASS: Выполнение теста test2
 - FAIL: Выполнение теста fibonacciTest
 - PASS: fibonacciTest.num0 (fibonacci_test.h 10)
 - Выполнение заняло 0 ms. (fibonacci_test.h 10)
 - FAIL: fibonacciTest.greater2 (fibonacci_test.h 17)
 - Expected equality of these values:
 - fibonacci(5)
 - Which is: 5
 - 2
 - 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)
- Open Documents:** Shows 'fibonacci_test.h'.
- Bottom Panel:** Shows a search bar and a tab for '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$
```

Github Actions

seekerk / gtest Public

Unpin Unwatch 2 Fork 14 Star 1

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Actions

New workflow

All workflows

CI/CD GitHub Actions

All workflows

Showing runs from all workflows

Filter workflow runs

34 workflow runs


	Event	Status	Branch	Actor
Update test-action.yml CI/CD GitHub Actions #33: Commit 76167ab pushed by seekerk	master	7 months ago 4m 1s	...	
manual CI/CD GitHub Actions #32: Commit e851057 pushed by seekerk	master	8 months ago 34s	...	
fix badges CI/CD GitHub Actions #31: Commit c08c431 pushed by seekerk	master	8 months ago 2m 18s	...	
job sequence CI/CD GitHub Actions #30: Commit f9b24e3 pushed by seekerk	master	8 months ago 2m 16s	...	
fix CI/CD GitHub Actions #29: Commit 354d7da pushed by seekerk	master	8 months ago 1m 28s	...	
fix sonarcloud badges CI/CD GitHub Actions #28: Commit df35efd pushed by seekerk	master	8 months ago 1m 16s	...	
fix name CI/CD GitHub Actions #27: Commit 19b2921 pushed by seekerk	master	8 months ago 1m 32s	...	
fix badge CI/CD GitHub Actions #26: Commit f902a6d pushed by seekerk	master	8 months ago 1m 33s	...	

Github Actions

seekerk / **gtest** Public Unpin Unwatch 2 Fork 14 Star 1

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

master **gtest** / .github / workflows / test-action.yml View runs Go to file ...

 **seekerk** Update test-action.yml ... ✓ Latest commit 76167ab on 17 Mar [History](#)

1 contributor

69 lines (66 sloc) | 2.95 KB Raw Blame ✎ 🗑

```
1 name: CI/CD GitHub Actions
2 on: [push]
3 jobs:
4   gtest-tests:
5     runs-on: ubuntu-latest
6     steps:
7       - uses: actions/checkout@v2
8       - run: git submodule init
9       - run: git submodule update
10      - run: pip install --user cpp-coveralls
11      - run: sudo apt-get update -y && sudo apt-get install -y qt5-qmake qt5-default
12      - run: qmake
13      - run: make
14      - run: ./tests/tests
15
16      - name: Coveralls
17        run: coveralls -t ${ secrets.coveralls_token } --root . -E ".*external.*" -E ".*tests.*"
18
19
20 sonarcloud-check:
21   needs: gtest-tests
22   runs-on: ubuntu-latest
23   env:
24     SONAR_SCANNER_VERSION: 4.6.1.2450 # Find the latest version in the "Linux" link on this page:
25                                     # https://sonarcloud.io/documentation/analysis/scan/sonarscanner/
26     SONAR_SERVER_URL: "https://sonarcloud.io"
```


coveralls

COVERALLS

- ☰
- +
- i
- 🔒
- ?
- 👤
- 🔌

- ⚙️ [Settings](#)
- 📁 [Repo on GitHub](#)

 **SEEKERK / GTEST**

100%


DEFAULT BRANCH: MASTER

REPO ADDED 01 JUN 2017 05:54AM UTC	TOTAL FILES 3	# BUILDS 20 <small>LAST ↻</small>	BADGE coverage 100% <small>EMBED ▾</small>	TOKEN WSa29PSUxvUJka49cNEoh5f <small>REGEN ↻</small>
---	-------------------------	--	---	---

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 + github actions

- Использование gcov библиотеки при сборке
- Отправка статистик на сервер

```
9   - run: git submodule update
10  - run: pip install --user cpp-coveralls
11  - run: sudo apt-get update -y && sudo apt-get install -y qt5-qmake qt5-default
12  - run: qmake
13  - run: make
14  - run: ./tests/tests
15
16  - name: Coveralls
17    run: coveralls -t ${{ secrets.coveralls_token }} --root . -E ".*external.*" -E ".*tests.*"
18
19
```

Статическое тестирование

- Статическое тестирование — анализ формальными методами без выполнения тестируемой программы неверных конструкций или неверных отношений объектов программы (ошибки формального задания)
 - ручное исполнение
 - компиляторы
 - среда программирования
 - специальные инструменты

Цели и задачи

- Цель: выявление расхождений на раннем этапе в различных представлениях проекта (код, документация, ...)
- Исходные данные:
 - ТЗ
 - Стандарты, спецификации, гайдлайны
 - Проектные ожидания
- Задачи:
 - анализ представления
 - поиск несоответствий между текущим состоянием и ожиданием (например, между кодом и стандартом кодирования)
 - локализация проблемного участка

Ручное тестирование

- Выбирается объект тестирования (например, код модуля)
- Назначается ответственный за тестирование
- Проводится ручной анализ объекта (просмотр кода)
- Результаты оформляются в протокол
- Плюсы:
 - минимум ложных срабатываний
 - учет множества критериев
- Минусы:
 - очень долго и дорого
 - человеческий фактор

Ручное тестирование

- Критерии оценки: набор(ы) правил
- Пример: правила Скотта Майерса (Скотт Майерс — Эффективное использование C++. 55 верных способов улучшить структуру и код ваших программ)
 - Правило 2: Предпочитайте `const`, `enum` и `inline` использованию `#define`
 - Правило 3: Везде, где только можно используйте `const`
 - Правило 4: Прежде чем использовать объекты, убедитесь, что они инициализированы
 - Правило 7: Объявляйте деструкторы виртуальными в полиморфном базовом классе
 - Правило 9: Никогда не вызывайте виртуальные функции в конструкторе или деструкторе
 - Правило 18: Проектируйте интерфейсы так, что их легко было использовать правильно и трудно – неправильно
 - ...

Компилятор

- Ошибки компиляции
- Предупреждения
- Опции компиляции
 - `-Werr='тип предупреждения'` — воспринимать предупреждение как ошибку. Для настройщиков самураев - `Werr` без параметров или `-Werror`
 - `-Wpedantic` строгое соответствие стандартам ISO C и ISO C++
 - `-Wall` включение предупреждений
 - `-Wextra` включение дополнительных предупреждений

Компилятор

- **-Weffc++** must have опция. Не включается с помощью **-Wextra** или **-Wall** и содержит проверку рекомендаций Скотта Мейерса (<http://www.aristeia.com/ddjpaper1.html>):
 - Item 11: Define a copy constructor and an assignment operator for classes with dynamically allocated memory.
 - Item 12: Prefer initialization to assignment in constructors.
 - Item 14: Make destructors virtual in base classes.
 - Item 15: Have «operator=» return a reference to *this.
 - Item 23: Don't try to return a reference when you must return an object.
 - Item 6: Distinguish between prefix and postfix forms of increment and decrement operators.
 - Item 7: Never overload "&&", "||", or "!,".
- **-Woverloaded-virtual** — перегрузка виртуальных функций выглядит плохо.
- **-Wctor-dtor-privacy** — неиспользуемые классы — с приватными конструкторами и деструктором
- **-Wnon-virtual-dtor** — не виртуальный деструктор
- **-Wold-style-cast** — приведение в стиле C — это плохо
- **-Wconversion -Wsign-conversion** — ворнинг о преобразовании типа. при котором значение может измениться. Как ни странно, не входит в **-Wall**
- **-Wunreachable-code** — код, который никогда не будет выполнен


Среда программирования

- Динамическое отслеживание состояние кода
- Автоподстановка элементов/значений
- Подсветка/пояснение ошибочных конструкций
- Автозамена часто встречаемых ошибочных конструкций (пример: `a.b` на `a->b`)
- Оформление кода
- Отслеживание изменений

Специальные средства

- Более строгая формализация кода
- Набор тестов на наличие "здорового смысла" в коде
- Аналог работы компилятора с максимальными параметрами и минимальной производительностью
 - ***lint** Набор средств для разных языков программирования (splint)
 - **Coverity** Статический анализатор (<http://www.coverity.com/>)
 - **Cppcheck** Бесплатный анализатор на основе регулярных выражений (<http://cppcheck.sourceforge.net/>)
 - **Clang** Компилятор со встроенным статическим анализатором (<http://clang.llvm.org/>)
 - **Frama-C** Анализатор программ на языки Си (<http://frama-c.com/>)
 - **Goanna** Анализатор программ на языки Си/Си++ (<http://www.redlizards.com/>)
 - **PC-Lint** Весьма гибкий инструмент статического анализа (<http://www.gimpel.com/>)
 - ...

sonarcloud

sonarcloud  My Projects My Issues

Explore 





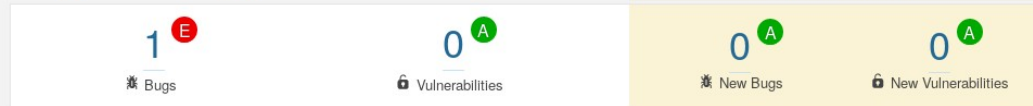
 Kirill Kulakov /  gtest  master 


June 21, 2018, 11:48 AM Version 1.0

[Overview](#) [Issues](#) [Security Reports](#) [Measures](#) [Code](#) [Activity](#) [Administration](#)

Quality Gate  **Passed**

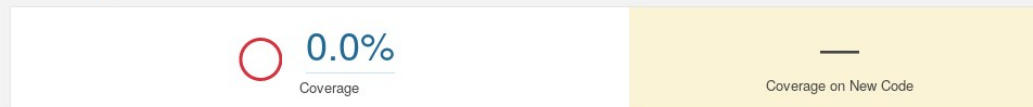
Bugs  Vulnerabilities 



Code Smells 



Coverage 



Duplications 



About This Project

[Public](#)

No tags

XS 103
Lines of Code

C 95
C++ 8

Project Activity



June 21, 2018
1.0

June 21, 2018
Quality Profile: **Stop using 'Sonar way'**
(Python)

June 21, 2018
Project Analyzed
[Show More](#)

Quality Gate

(Default) [Sonar way](#)

Quality Profiles

(C) [Sonar way](#)
(C++) [Sonar way](#)

Project Key

[Copy](#)

Organization Key

[Copy](#)

Sonarcloud + github actions

- Создание проекта на sonarcloud
- Следование инструкциям
- Создание файла sonar-project.properties:

21 lines (18 sloc) | 756 Bytes

```
1 # must be unique in a given SonarQube instance
2 sonar.projectKey=seekerk_gtest
3 sonar.organization=seekerk-github
4 # this is the name and version displayed in the SonarQube UI. Was mandatory prior to SonarQube 6.1.
5 #sonar.projectName=gtest
6 #sonar.projectVersion=1.0
7
8 # Path is relative to the sonar-project.properties file. Replace "\" by "/" on Windows.
9 # This property is optional if sonar.modules is set.
10 sonar.sources=.
11 sonar.exclusions=external/**
12 sonar.cfamily.gcov.reportsPath=.
13
14 # Encoding of the source code. Default is default system encoding
15 #sonar.sourceEncoding=UTF-8
16
17 #sonar.cfamily.build-wrapper-output=bw-output
18 #sonar.host.url=https://sonarcloud.io
19 sonar.cfamily.threads=2
20 sonar.cfamily.cache.enabled=true
21 sonar.cfamily.cache.path=tmp-cache
```

Профилитрование: Valgrind

- Valgrind — инструментальное программное обеспечение, предназначенное для отладки использования памяти, обнаружения утечек памяти, а также профилирования
 - отслеживание работы с памятью (memcheck)
 - отслеживание вызовов (callgrind)
 - отслеживание работы с кэшем (cachegrind)
 - ...
- Запуск:
`valgrind <app_file>`

Valgrind (нет утечек)

Valgrind (нет утечек)

```
kulakov@ultrabook:~/projects/testing/build-qmake-gtest-Desktop-Debug/app$ valgrind ./app
```

```
==31923== Memcheck, a memory error detector
```

```
==31923== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
```

```
==31923== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
```

```
==31923== Command: ./app
```

```
==31923==
```

```
Hello World!
```

```
==31923==
```

```
==31923== HEAP SUMMARY:
```

```
==31923== in use at exit: 0 bytes in 0 blocks
```

```
==31923== total heap usage: 4 allocs, 4 frees, 2,208 bytes allocated
```

```
==31923==
```

```
==31923== All heap blocks were freed -- no leaks are possible
```

```
==31923==
```

```
==31923== For counts of detected and suppressed errors, rerun with: -v
```

```
==31923== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

valgrind (утечки)

```
==32656== Memcheck, a memory error detector
==32656== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==32656== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
==32656== Command: ./app
==32656==
Hello World!
==32656==
==32656== HEAP SUMMARY:
==32656== in use at exit: 1,000 bytes in 1 blocks
==32656== total heap usage: 5 allocs, 4 frees, 3,208 bytes allocated
==32656==
==32656== LEAK SUMMARY:
==32656== definitely lost: 1,000 bytes in 1 blocks
==32656== indirectly lost: 0 bytes in 0 blocks
==32656== possibly lost: 0 bytes in 0 blocks
==32656== still reachable: 0 bytes in 0 blocks
==32656== suppressed: 0 bytes in 0 blocks
==32656== Rerun with --leak-check=full to see details of leaked memory
==32656==
==32656== For counts of detected and suppressed errors, rerun with: -v
==32656== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```


valgrind (детали утечки)

- kulakov@ultrabook:~/projects/testing/build-qmake-gtest-Desktop-Debug/app\$ valgrind --leak-check=full ./app
- ==32759== Memcheck, a memory error detector
- ==32759== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
- ==32759== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
- ==32759== Command: ./app
- ==32759==
- Hello World!
- ==32759==
- ==32759== HEAP SUMMARY:
- ==32759== in use at exit: 1,000 bytes in 1 blocks
- ==32759== total heap usage: 5 allocs, 4 frees, 3,208 bytes allocated
- ==32759==
- ==32759== 1,000 bytes in 1 blocks are definitely lost in loss record 1 of 1
- ==32759== at 0x4C2DB8F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
- ==32759== by 0x400EA2: myfunc (myfunc.c:6)
- ==32759== by 0x400E63: main (main.c:8)
- ==32759==
- ...

callgrind

- Запуск

```
kulakov@ultrabook:~/projects/testing/build-qmake-gtest-Desktop-Debug/app$ valgrind
```

```
--tool=callgrind ./app
```

```
==446== Callgrind, a call-graph generating cache profiler
```

```
==446== Copyright (C) 2002-2015, and GNU GPL'd, by Josef Weidendorfer et al.
```

```
==446== Using Valgrind-3.11.0 and LibVEX; rerun with -h for copyright info
```

```
==446== Command: ./app
```

```
==446==
```

```
==446== For interactive control, run 'callgrind_control -h'.
```

```
Hello World!
```

```
==446==
```

```
==446== Events : Ir
```

```
==446== Collected : 239342
```

```
==446==
```

```
==446== I refs: 239,342
```

callgrind

- Анализ

```
kulakov@ultrabook:~/projects/testing/build-qmake-gtest-Desktop-Debug/app$ callgrind_annotate
```

```
Reading data from 'callgrind.out.421'...
```

```
.....
```

```
-----  
lr file:function  
-----
```

```
56,475 /build/glibc-Cl5G7W/glibc-2.23/elf/dl-addr.c:_dl_addr [/lib/x86_64-linux-gnu/libc-2.23.so]  
33,910 /build/glibc-Cl5G7W/glibc-2.23/elf/dl-lookup.c:do_lookup_x [/lib/x86_64-linux-gnu/ld-2.23.so]  
21,802 /build/glibc-Cl5G7W/glibc-2.23/elf/dl-lookup.c:_dl_lookup_symbol_x  
[/lib/x86_64-linux-gnu/ld-  
2.23.so]  
20,616 ???:gcov_do_dump [/home/kulakov/projects/testing/build-qmake-gtest-Desktop-  
Debug/app/app]  
17,553 /build/glibc-Cl5G7W/glibc-2.23/elf/./sysdeps/x86_64/dl-machine.h:_dl_relocate_object
```