Тестирование в Ruby

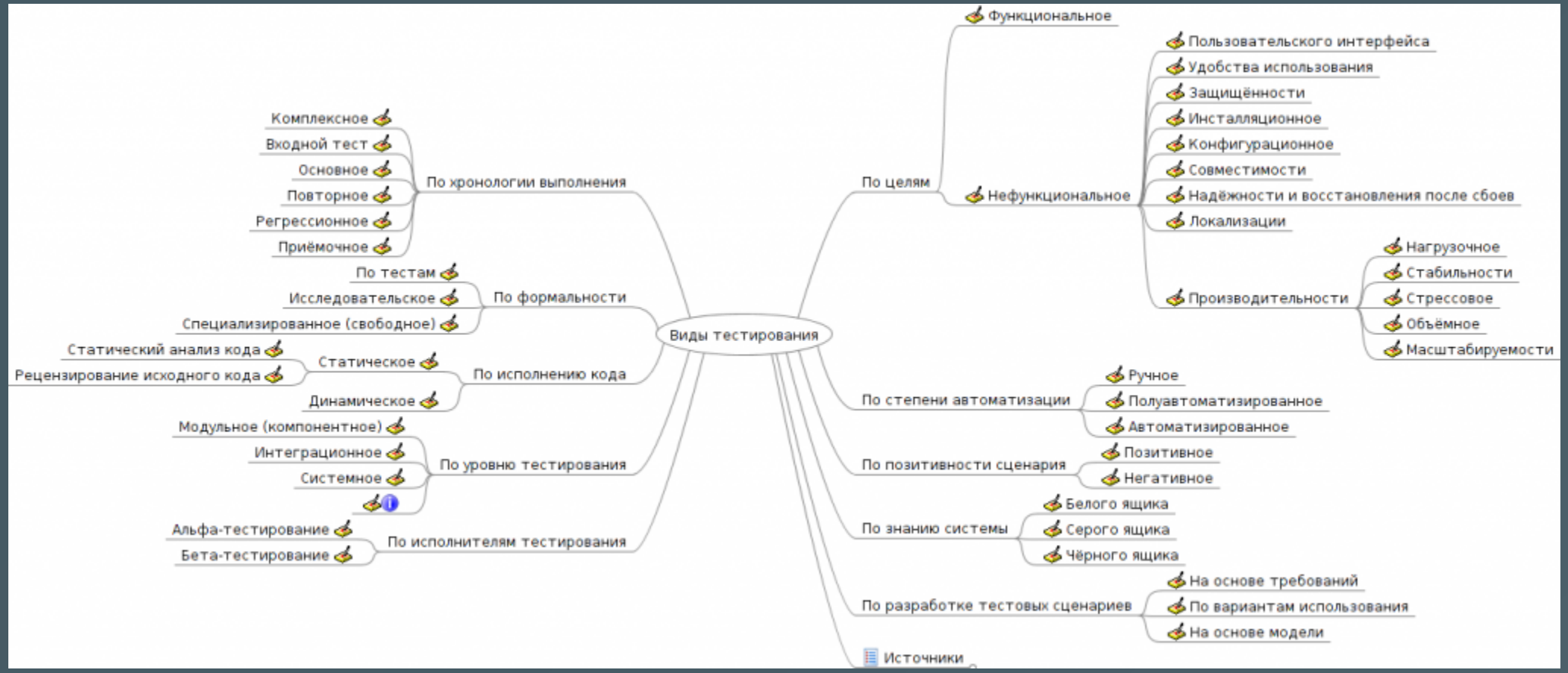
Что сегодня узнаем?

- Зачем нужны тесты?
- Как тестировать код в Ruby?
- Структура тестов
- Вспомогательные инструменты

Зачем тестировать?

- Убедиться, что новый функционал не ломает старый
- Выявить баги на этапе деплоя, а не в проде
- Для согласованной работы команды разработчиков
- Выявить неправильную архитектуру

Виды тестирования



Виды тестирования

- Блочное
- Интеграционное
- Системное
- 

Структура теста

- Начальные условия
- Ожидаемый результат
- Проверка результата

Структура тестов

```ruby
def greeting(name)
  "Hello, #{name}!"
end

name = "Bob"                                # Начальные условия
expected_greeting = "Hello, Bob!"           # Ожидаемый результат

if greeting(name) != expected_greeting  # Проверка результата
  raise "Test failed"
else
  puts "Test passed"
end
```

Фреймворки

	PROJECT SCORE	DOWNLOADS	STARS	FORKS	FIRST RELEASE	LATEST RELEASE	REVERSE DEPENDENCIES
rspec	★ 43.39	★ 536,482,366	2,787	229	2005-08-11	2020-10-30	★ 55,430
minitest	32.14	373,938,855	★ 2,898	★ 459	2008-10-09	2021-02-24	11,534
test-unit	2.47	27,821,933	220	79	2008-03-20	★ 2021-04-19	2,720
yard-doctest	0.30	3,074,144	78	12	2014-06-16	2019-09-12	84
bacon	0.27	350,016	421	54	2008-01-06	2012-12-21	296
wrong	0.23	310,005	431	27	2010-07-06	2013-11-11	56
riot	0.17	124,905	316	26	2009-10-03	2013-10-18	133
testrocket	0.10	5,361	234	7	2011-08-09	2019-10-02	2
rubydoctest	0.08	313,762	96	12	2008-05-25	2014-12-31	7
shindo	0.07	224,629	80	13	2009-10-07	2021-03-01	68
testy	0.04	3,272	97	4	2009-03-28	2009-03-28	1
yard-rspec	0.03	180,401			2009-09-15	2009-09-15	35
micronaut	0.03	7,365	80	2	2009-05-23	2009-08-10	11
fix	0.02	39,698	42	1	★ 2014-08-26	2020-01-16	9
assert	0.02	141,263	10	1	2011-08-14	2021-03-27	127
dtf	0.01	47,837	14	1	2012-05-08	2012-10-20	3
lemon	0.01	36,869	7	0	2009-12-31	2012-03-18	9

Minitest

- Дефолтный фреймворк для Ruby
- Дефолтный фреймворк для Rails
- Легковесный, простой, гибкий
- Два стиля написания тестов - классовый и DSL

Классовый стиль

```ruby
require 'minitest/autorun'

def greeting(name)
  "Hello, #{name}!"
end

class MyTest < Minitest::Test

  def setup; end

  def teardown; end

  def test_if_dummy_method_returns_true
    name = "Bob"                                  # Начальные условия
    expected_greeting = "Hello, Bob!"             # Ожидаемый результат
    assert(greeting(name) == expected_greeting)   # Проверка результата
  end

end
```

DSL стиль

```ruby
require 'minitest/autorun'

def greeting(name)
  "Hello, #{name}!"
end

describe 'dummy method' do

  before do
  end

  after do
  end

  it 'should return true' do
    name = "Bob"                                  # Начальные условия
    expected_greeting = "Hello, Bob!"             # Ожидаемый результат
    assert(greeting(name) == expected_greeting)   # Проверка результата
  end

end
```

Модули Minitest

- minitest/autorun
- minitest/test
- minitest/spec
- minitest/mock
- minitest/benchmark
- minitest/pride

Домашнее задание

```
hexlet program download rails testing
```