

Skillbox

Фитнес-функции архитектуры

Андрей Гордиенков

Solution Architect

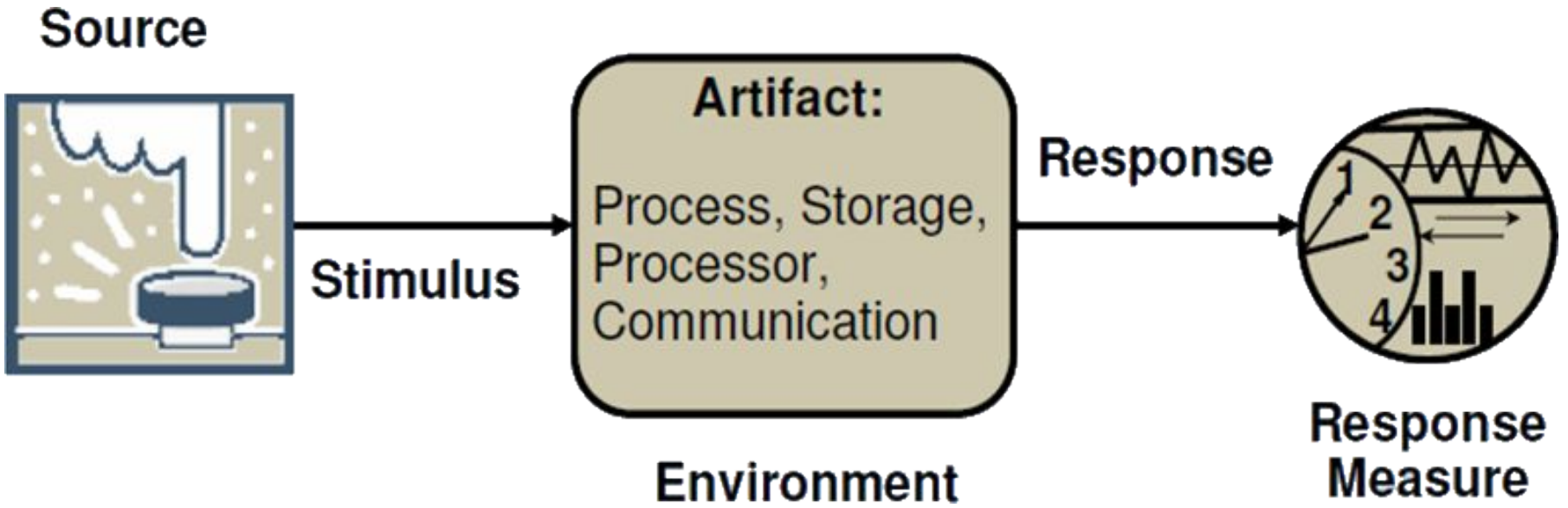
ABAX

В прошлом уроке

- Узнали, как создать концепт архитектуры, отвечающий основным НФТ

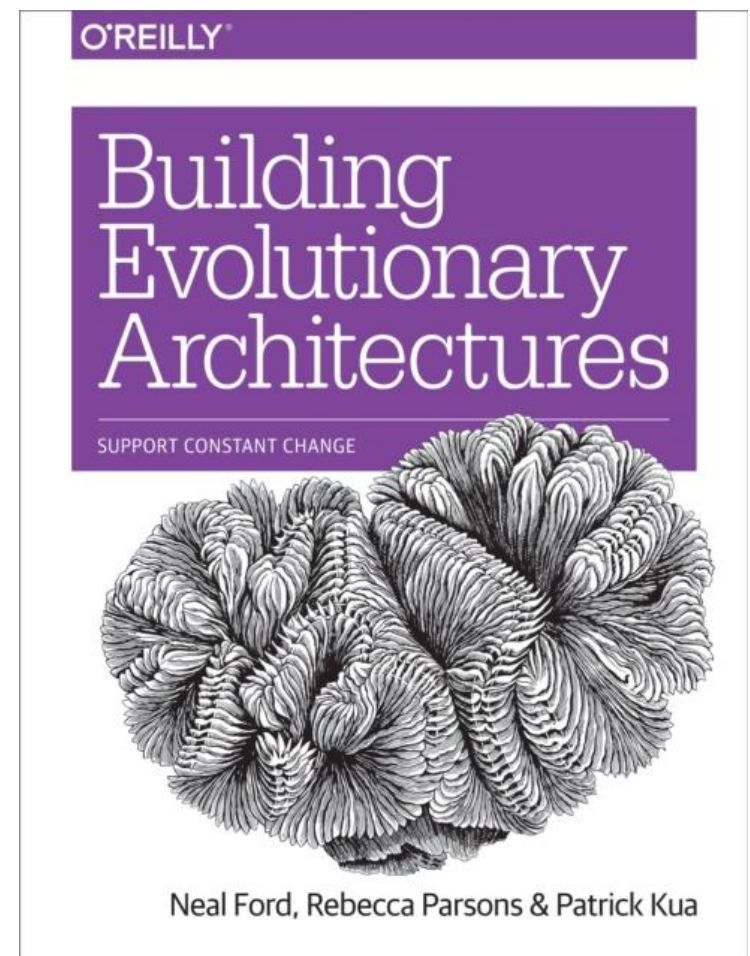
В этом уроке

- Как убедиться в том, что первоначальный план соблюдается в процессе разработки?
- Как следить за трендами изменения НФТ в реальной системе?



Определение

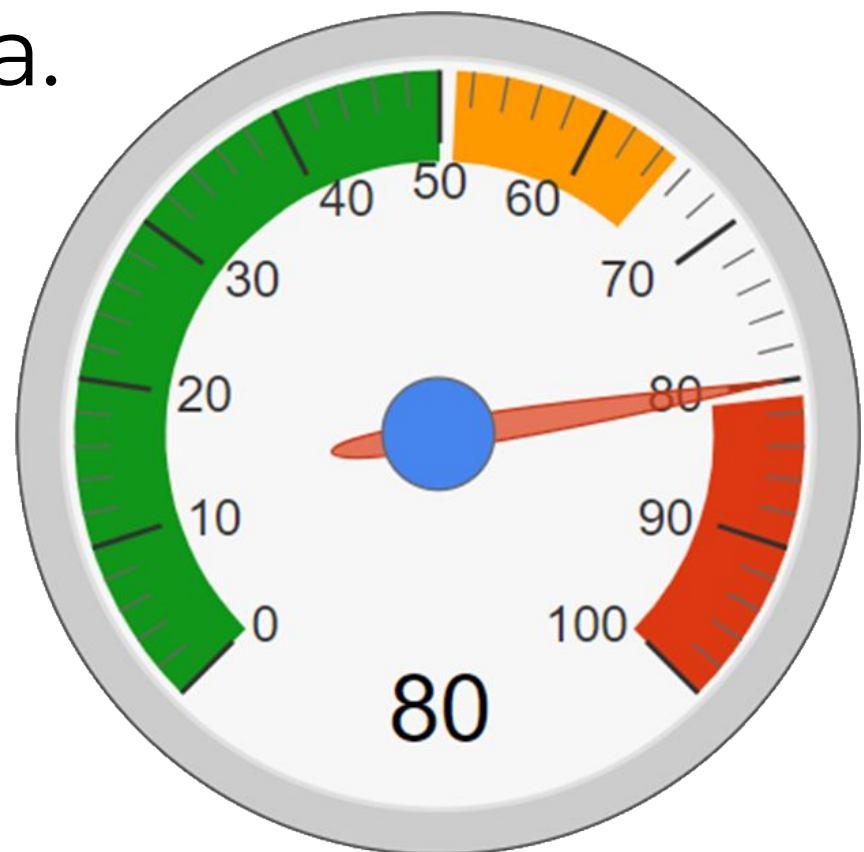
Фитнес-функции обеспечивают объективную оценку соблюдения архитектурных характеристик, которые могут включать в себя существующие критерии проверки, такие как: тестирование модулей, метрики, мониторы и т. д.



Инструментарий

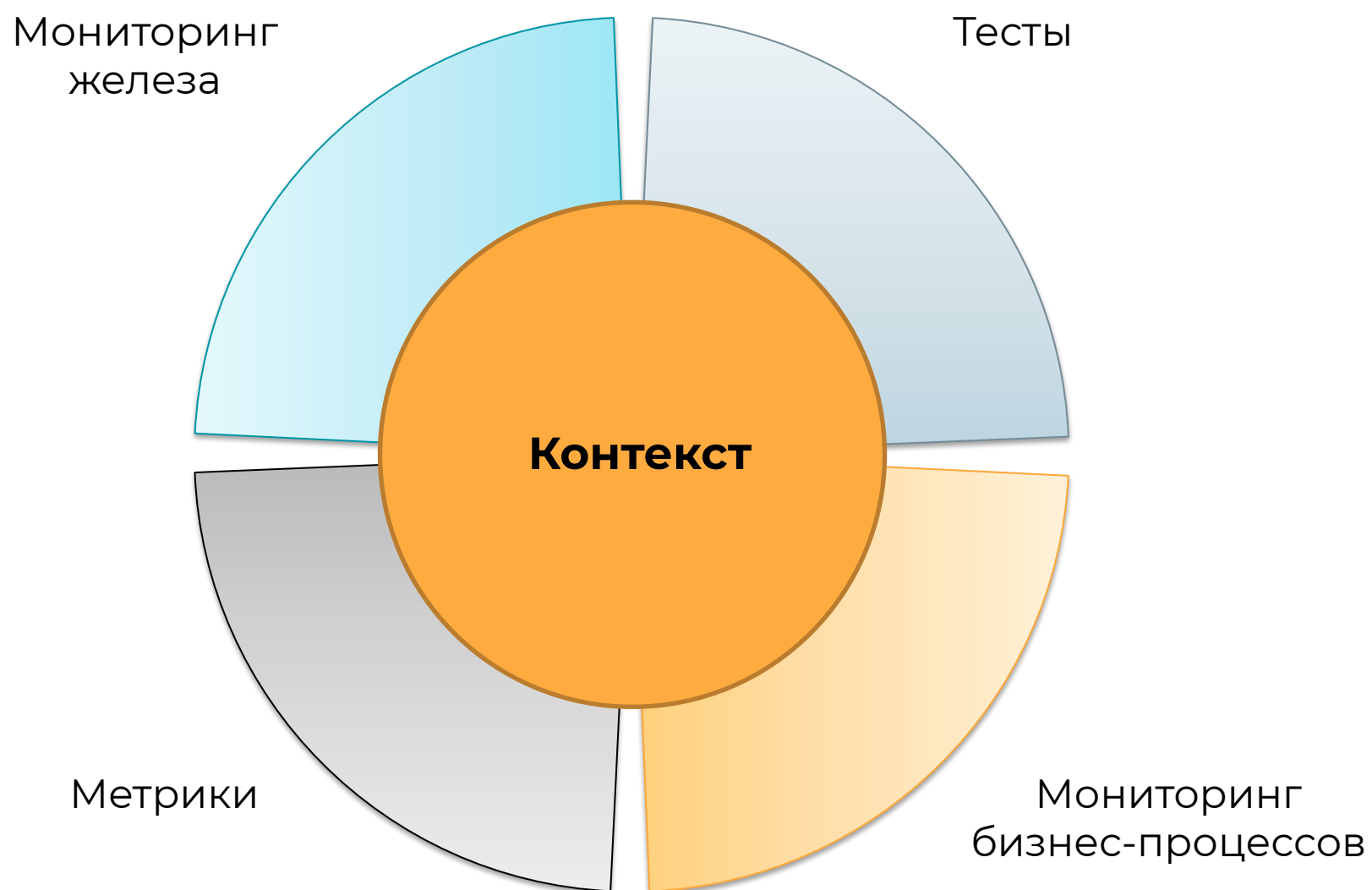
- Тесты — юнит, интеграционные, производительности
- Метрики и их тренды
- Мониторинг железа
- Мониторинг бизнес-процессов

Фитнес-функции строятся для наиболее важных атрибутов качества.



Когнитивная сложность

Примеры



Тесты

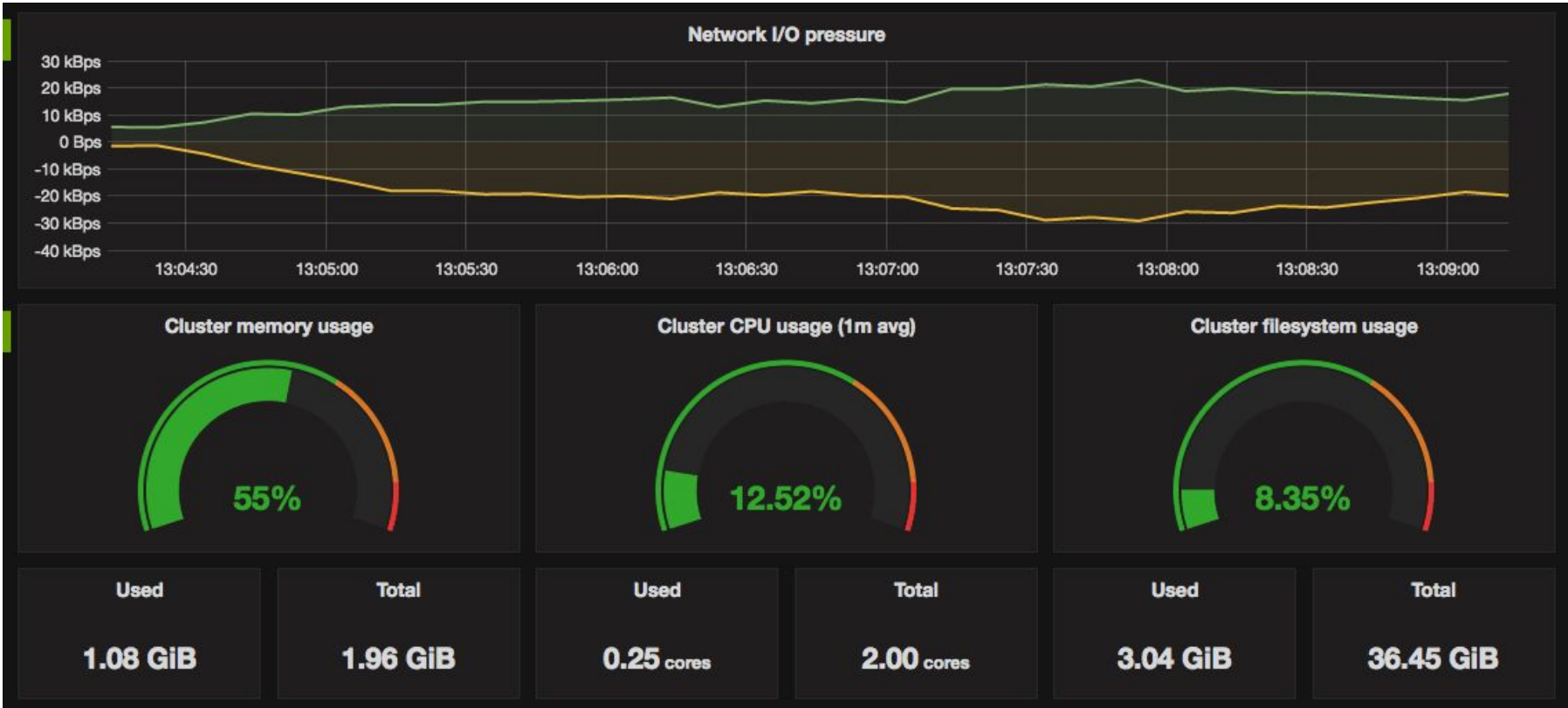
Coverage TreeHot Spots

Type to search

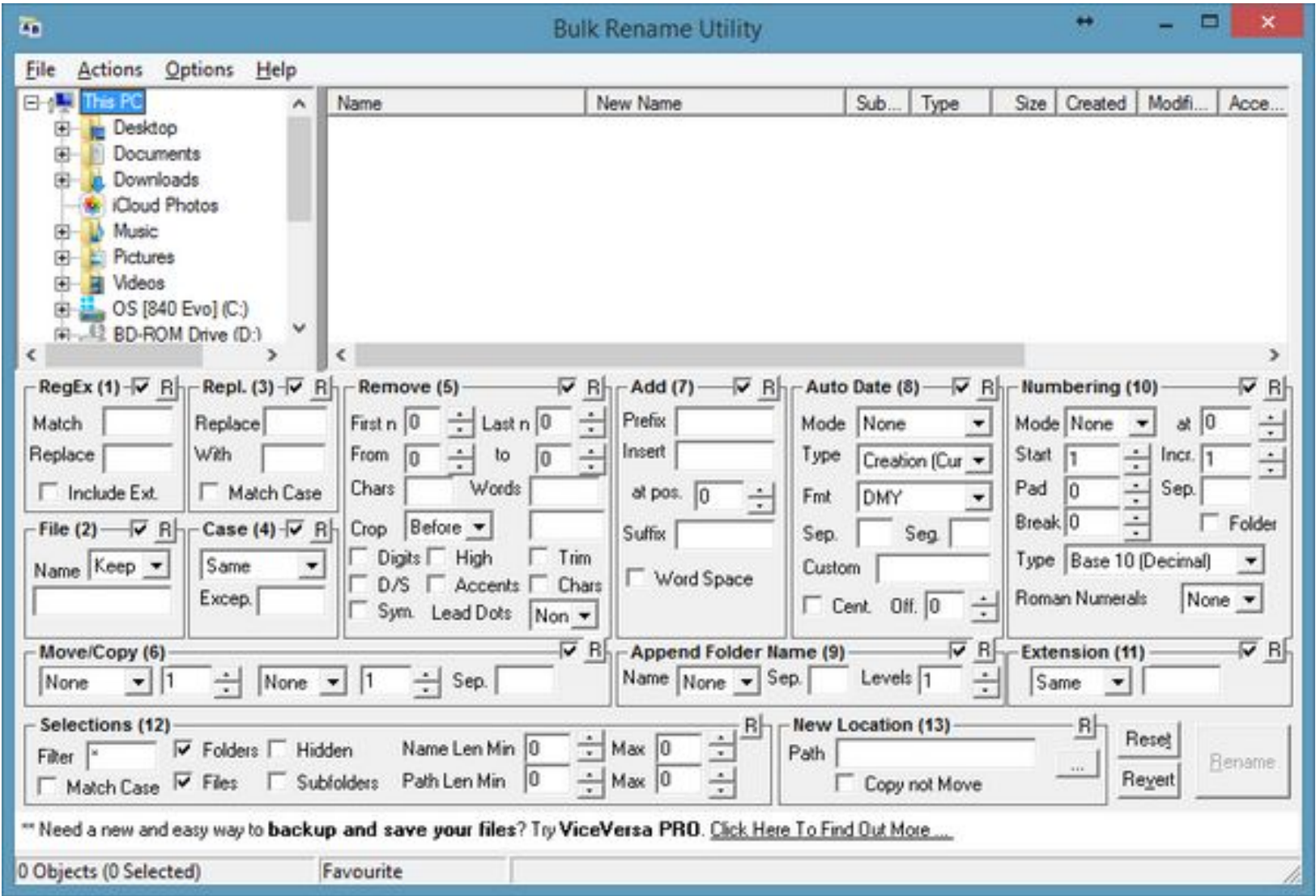
Symbol	Coverage (%) ▲	Uncovered/Total Stmts.
▲ Total	100%	0/23
▲ TestCoverage	100%	0/23
▲ { } Math	100%	0/4
▶ AdvancedCalculator	100%	0/1
▶ Calculator	100%	0/3
▶ { } TestCoverage	100%	0/19

OutputCoverage

Железо



Опыт пользователя



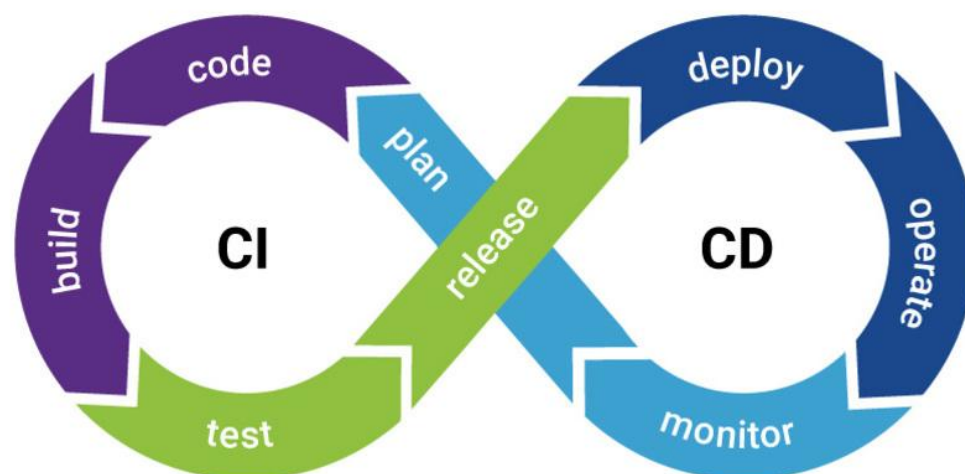
Metric-Driven Development

Избегайте ловушки Metric-Driven Development, когда за метриками теряются смысл процессов и конечная цель.

Архитектура — это не про технологии, это про то, как эффективно решить проблемы бизнеса.

Технические аспекты

- Версия в тестировании не должна отклоняться от других библиотек в продакшне более чем на две минорных версии
- Запрещается использовать пароли открытым текстом в коде
- У вас всегда должна быть стадия тестирования безопасности
- У каждого сервиса настроена своя учётная запись безопасности
- Каждое изменение кода должно иметь как минимум двух проверяющих и на прод уходит образ с тестирования, а не собирается из мастера



Качество кода

- Изменяемость
- Управляемость
- Конфигурируемость

```
describe "Code Quality" do
  it "has test coverage above 90%" do
    expect(quality.get_test_coverage()).to > .9
  end
  it "has maintainability rating of .1 or higher (B)" do
    expect(quality.get_maintainability_rating()).to < .1
  end
end
```

Устойчивость решения

- Отказоустойчивость
- Доступность

```
describe "Resiliency" do
  describe "New Deployment" do
    it "has less than 1% error rate for new deployment" do
      expect(new_deployment.get_error_rate()).to < .01
    end
  end
  describe "Network Latency" do
    it "has less than 5% error rate even if there is network latency" do
      expect(network_tests.get_error_rate()).to < .05
    end
    it "completes a transaction under 10 seconds even if there is network latency" do
      expect(network_tests.get_transaction_time()).to < 10
    end
  end
end
end
```

Возможности наблюдения

- Отладка
- Аудит
- Мониторинг

```
describe "Observability" do
  it "streams metrics" do
    expect(service.has_metrics()).to be(true)
  end
  it "has parseable logs" do
    expect(service.has_logs_in_aggregator()).to be(true)
  end
end
```

Производительность

```
describe "Performance" do
  it "completes a transaction under 10 seconds" do
    expect(transaction.check_transaction_round_trip_time()).to < 10
  end
  it "has less than 10% error rate for 10000 transactions" do
    expect(transaction.check_error_rate_for_transactions(10000)).to < .1
  end
end
```


Аудит и законы

```
describe "Compliance Standards" do
  describe "PII Compliance" do
    it "should not have PII in the logs" do
      expect(logs.has_pii_content()).to_not be(true)
    end
  end
  describe "GDPR Compliance" do
    it "should report types of personal information processed" do
      expect(gdpr.reports_PII_types()).to be(true)
    end
    it "should have been audited in the past year" do
      expect(gdpr.audit_age()).to < 365
    end
  end
end
end
```

Безопасность

```
describe "Security - Code Analysis" do
  describe "Code Analysis" do
    it "should use corporate-approved libraries only" do
      expect(code.only_uses_corporate_libraries()).to be(true)
    end
    it "should not have any of the OWASP Top 10" do
      expect(code.has_owasp_top_10()).to_not be(true)
    end
    it "should not have plaintext secrets in codebase" do
      expect(code.has_secrets_in_codebase()).to_not be(true)
    end
  end
end
describe "CVE Analysis" do
  it "should not use libraries with known vulnerabilities" do
    expect(libraries.have_no_cves()).to be(true)
  end
  it "should not use a container image with known vulnerabilities" do
    expect(container.has_no_cves()).to be(true)
  end
end
end
```

Работоспособность

- Конфигурируемость
- Работоспособность
- Восстановление

```
describe "Operability Standards" do
  describe "Operations Check" do
    it "should have a service runbook" do
      expect(service.has_runbook()).to be(true)
    end
    it "should have a README" do
      expect(service.has_readme()).to be(true)
    end
    it "should have alerts" do
      expect(service.has_alerts()).to be(true)
    end
    it "should have tracing IDs" do
      expect(service.has_tracing_ids()).to be(true)
    end
  end
end
end
```

Расширяйте контекст

- Сколько времени требуется от определения новой функциональности до релиза? Сколько разворачиваний на прод происходит в определённый период и сколько из них неудачные?
- Сколько времени требуется для внесения нового пользователя в систему с момента продажи услуги?
- Сколько запросов на поддержку приходит после обновления в течение определённого срока?
- Как часто и как долго система и её части недоступны вне запланированных сервисных периодов?

Самое важное

Атрибуты качества могут и должны отслеживаться в процессе реализации системы.

Вывод

Фитнес-функции позволяют следить, что реализация системы соответствует первоначальному архитектурному плану.

Итоги модуля

- Способы сбора нефункциональных данных
- Как приоритизировать НФТ?
- Связь НФТ, атрибутов качества и архитектуры
- Отслеживание трендов значений атрибутов качества для НФТ

Что дальше?

Детали проработки архитектуры и адресации NFT с точки зрения разных аспектов.

**Спасибо
за внимание!**