Белый ящик
Пандоры

1| Теория

Черный ящик

На слайде черный ящик

Черный ящик

На слайде черный ящик

Но вы его не видите потому что он черный

Черный ящик

Черный ящик

На слайде черный ящик

Но вы его не видите, потому что он черный

Но он есть

Белый ящик

Myers, The Art of Software Testing

The Economics of Testing

To combat the challenges associated with testing economics, you should establish some strategies before beginning. Two of the most prevalent strategies include black-box testing and white-box testing…

Источник: G. Myers, The Art of Software Testing , 2 ed.

Myers, The Art of Software Testing

Определение

White-box testing — A type of testing in which you examine the internal structure of a program.

Источник: G. Myers, The Art of Software Testing , 2 ed.

Myers, The Art of Software Testing

Практика

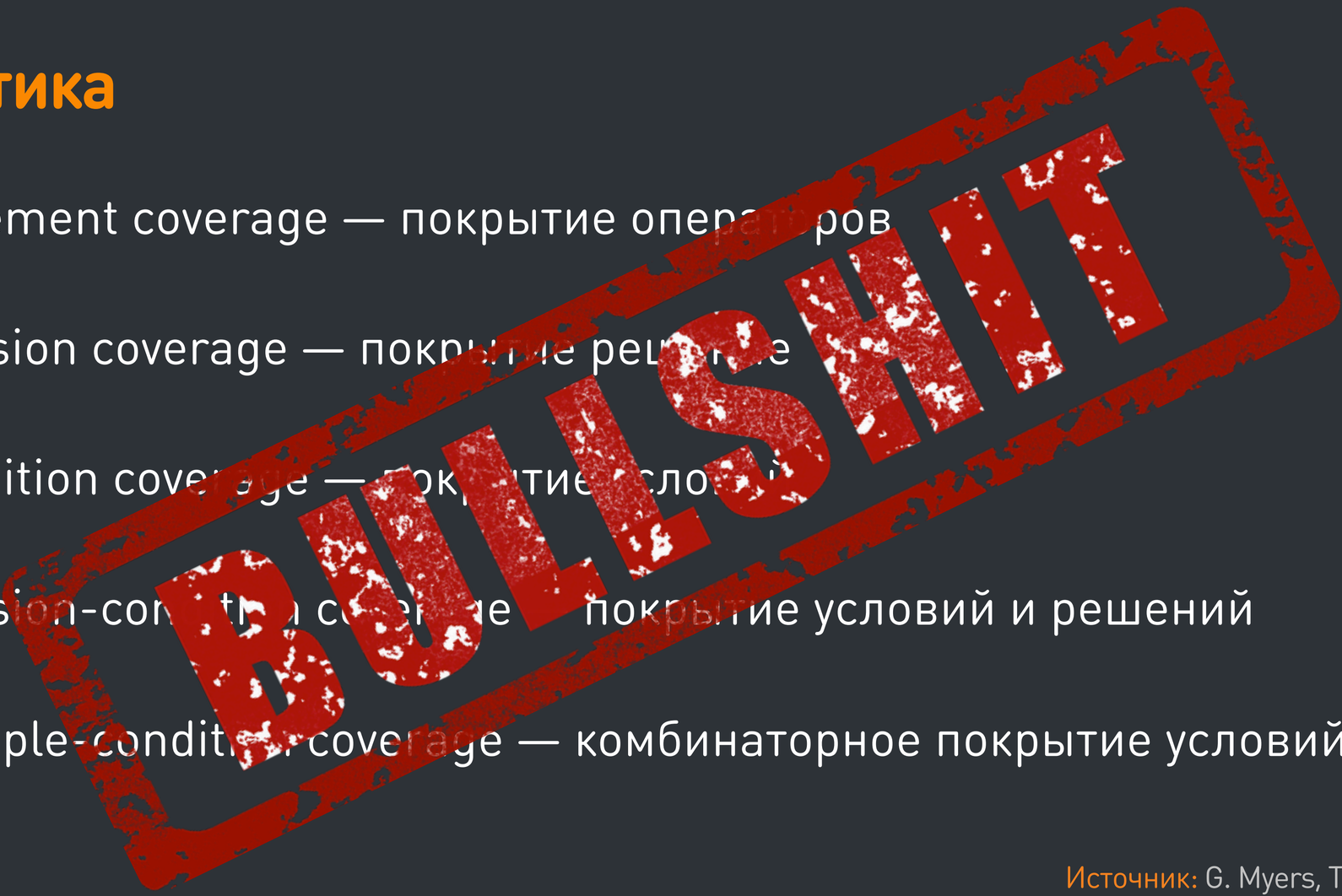
- Statement coverage — покрытие операторов
- Decision coverage — покрытие решение
- Condition coverage — покрытие условий
- Decision-condition coverage — покрытие условий и решений
- Multiple-condition coverage — комбинаторное покрытие условий и решений

Источник: G. Myers, The Art of Software Testing , 2 ed.

Myers, The Art of Software Testing

Практика

- Statement coverage — покрытие операторов
- Decision coverage — покрытие решений
- Condition coverage — покрытие условий
- Decision-condition coverage — покрытие условий и решений
- Multiple-condition coverage — комбинатор

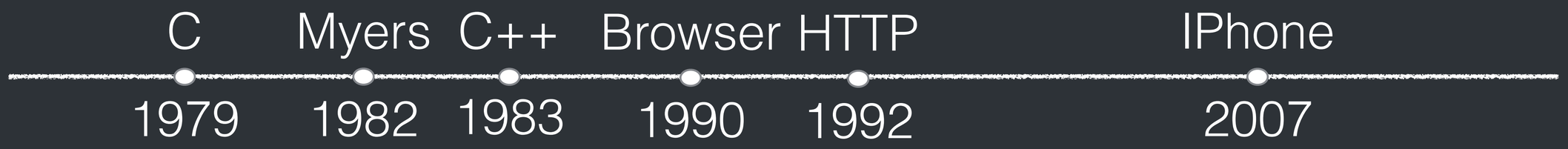


Источник: G. Myers, The Art of Software Testing , 2 ed.

Исторический контекст

Исторический контекст

Исторический контекст



Теория

Как надо

Смотрим в код

Понимаем структуру и зависимости

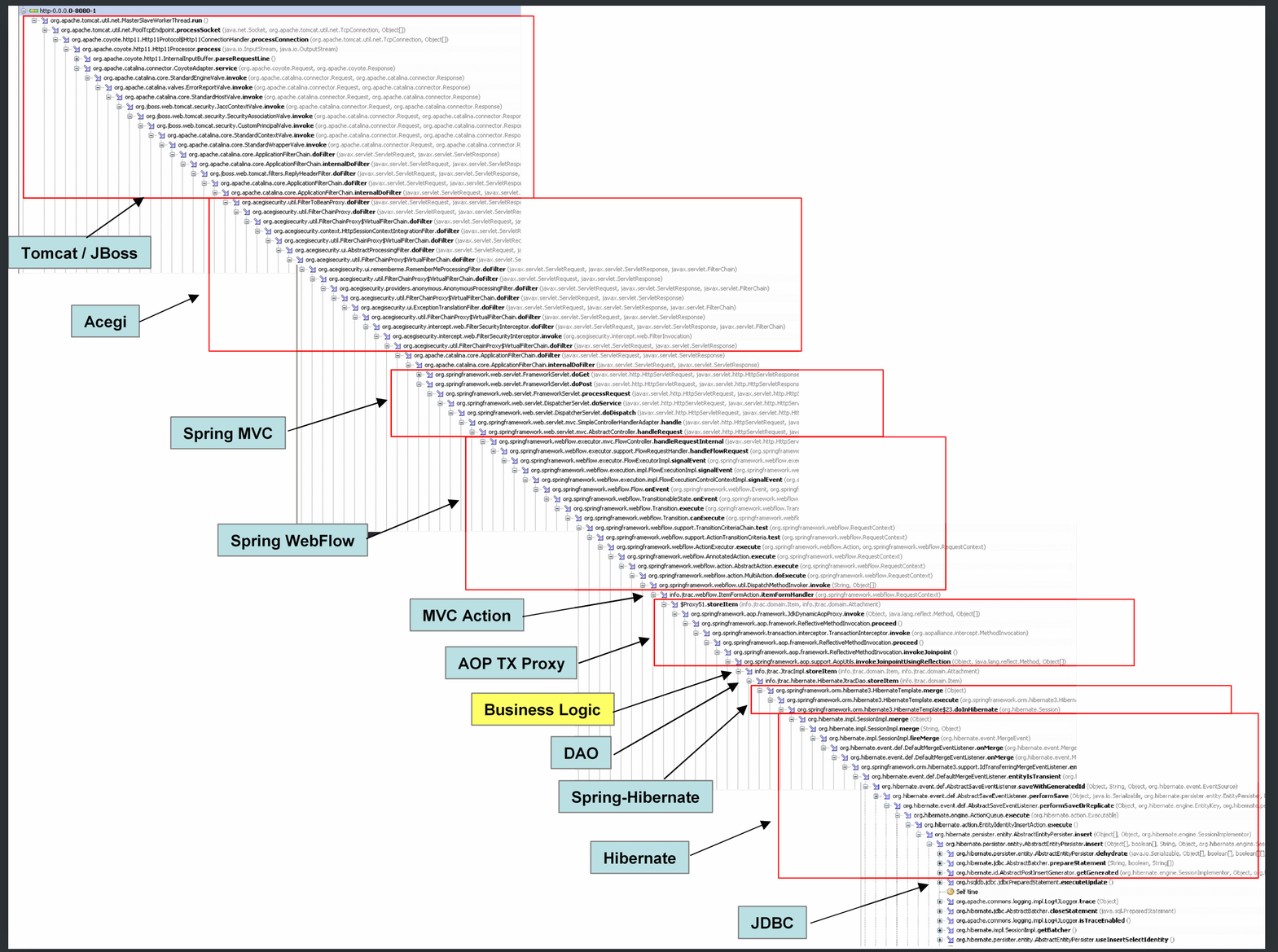
Делаем выводы, задаем вопросы, проектируем тесты

....

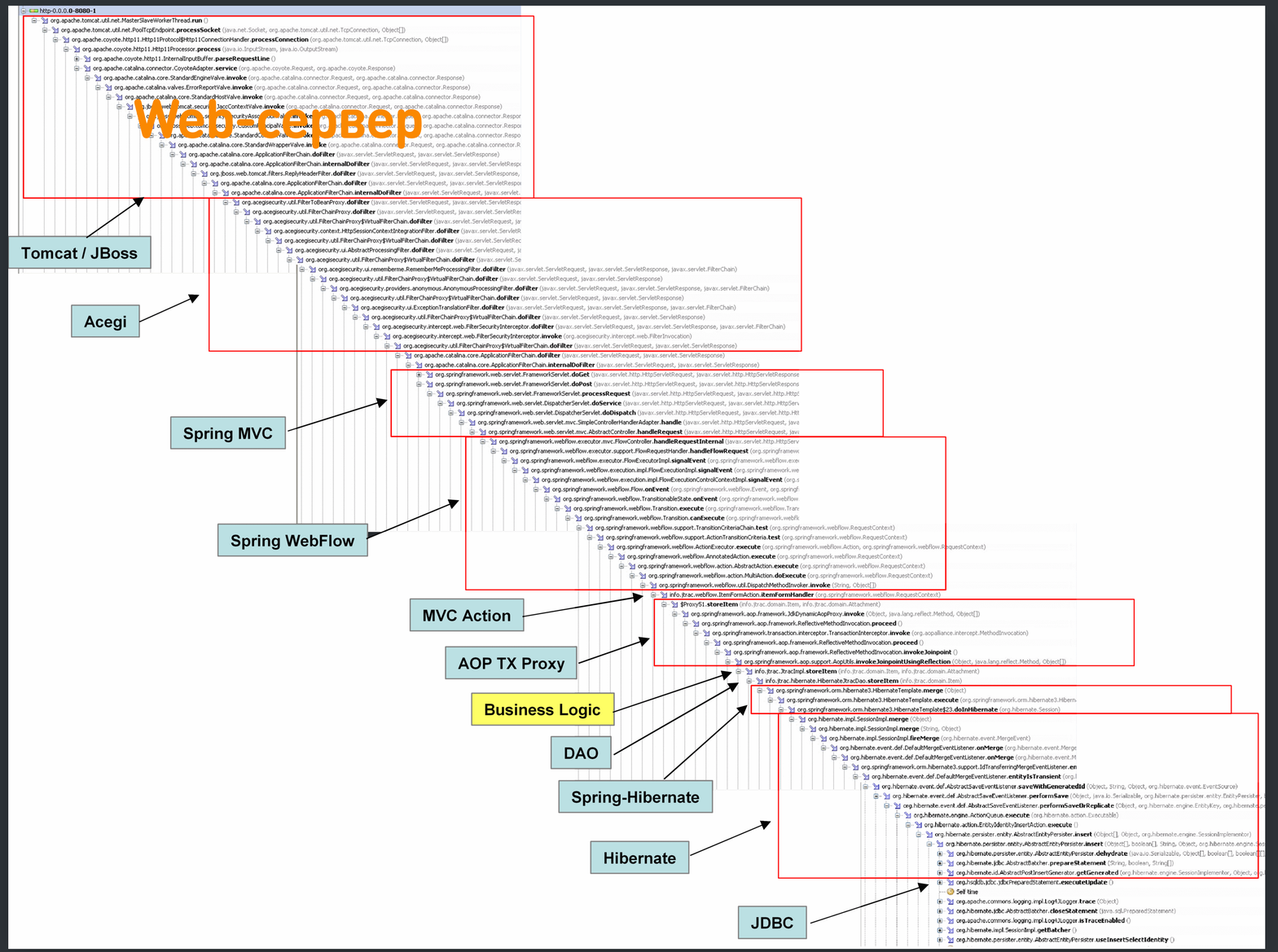
PROFIT!

Зачем ???

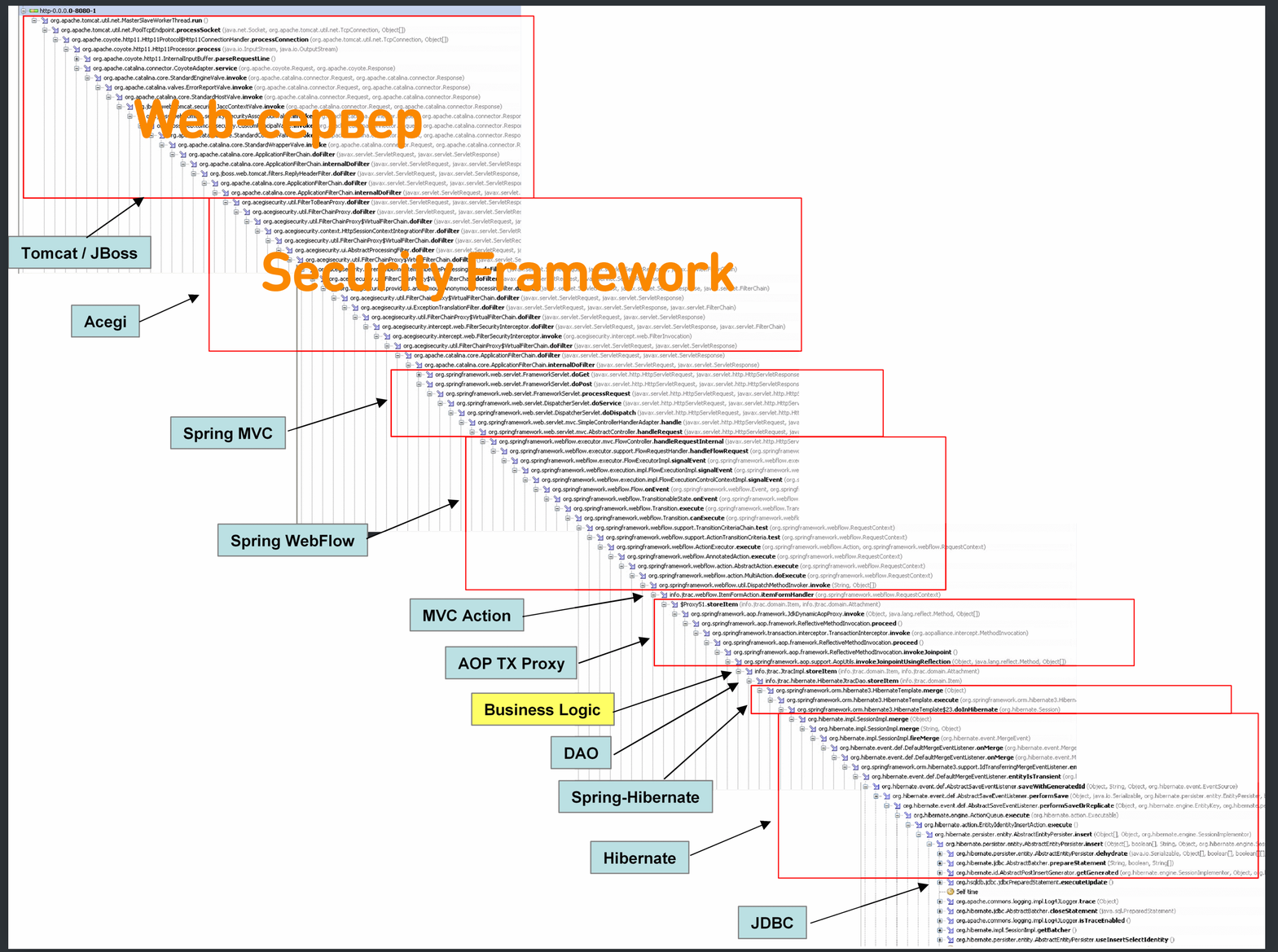
Real Life


Tomcat / JBoss
Acegi
Spring MVC
Spring WebFlow
MVC Action
AOP TX Proxy
Business Logic
DAO
Spring-Hibernate
Hibernate
JDBC

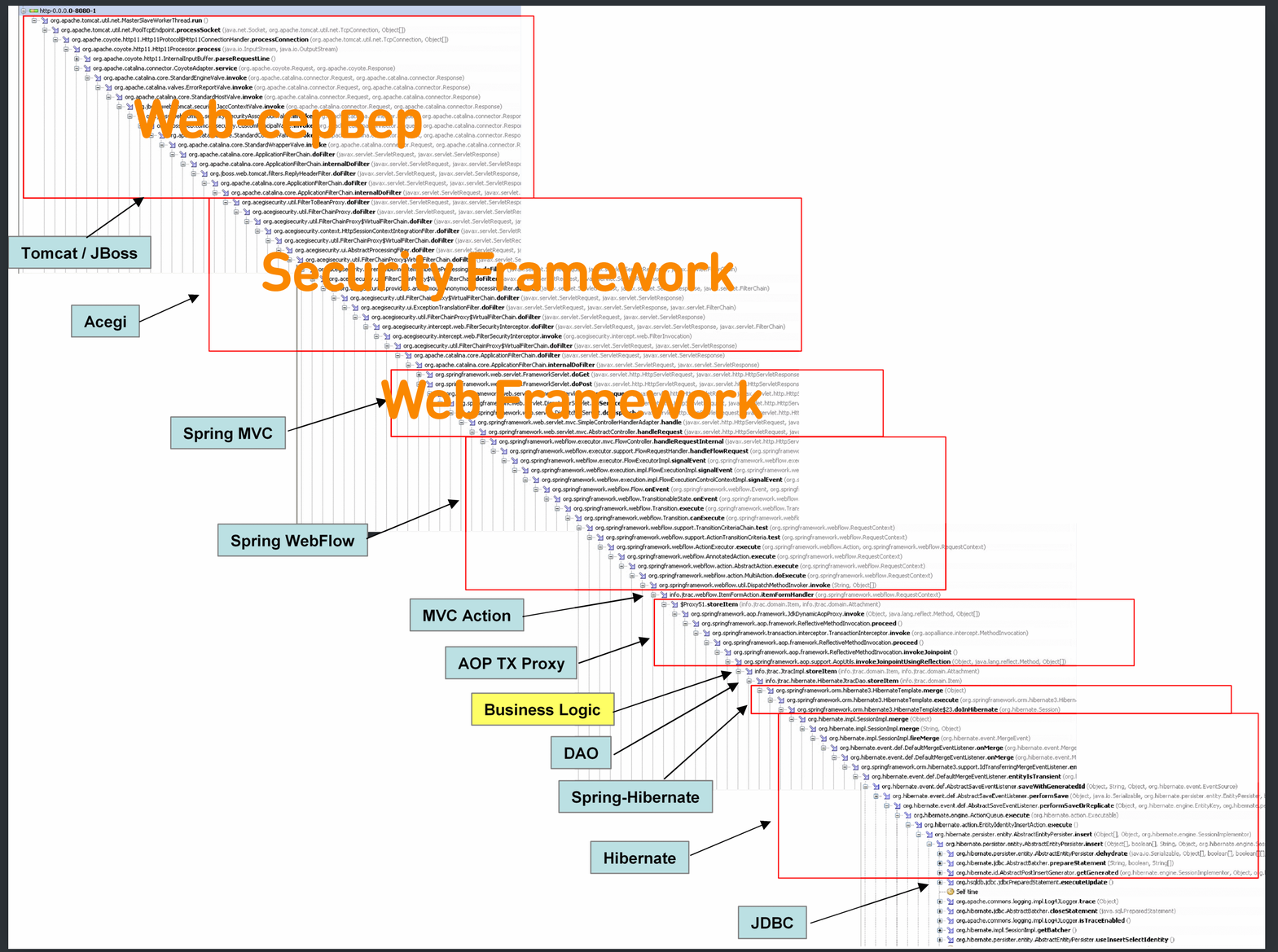
Real Life


Web сервер
Tomcat / JBoss
Acegi
Spring MVC
Spring WebFlow
MVC Action
AOP TX Proxy
Business Logic
DAO
Spring-Hibernate
Hibernate
JDBC

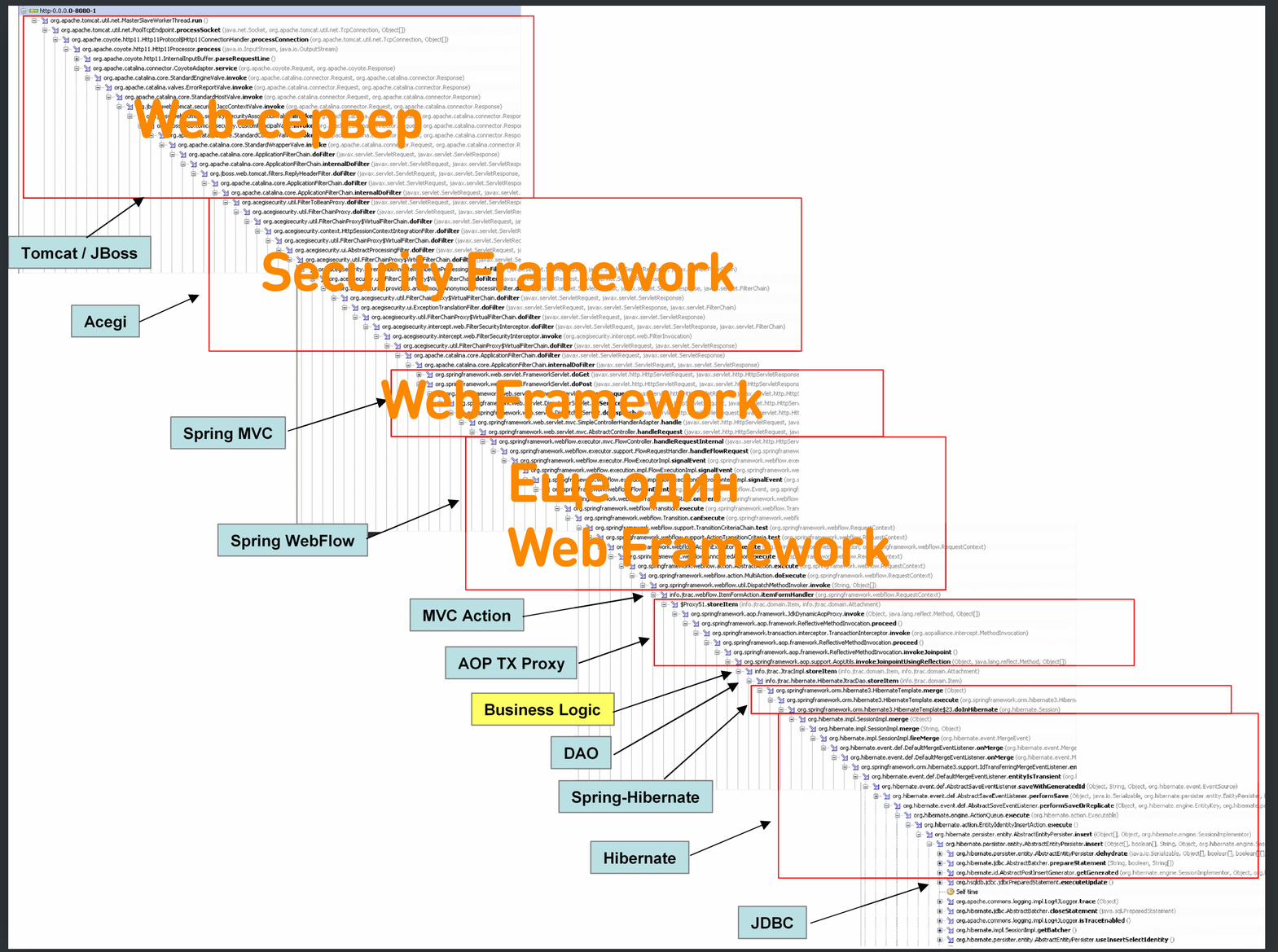
Real Life


Web сервер
Security Framework
Tomcat / JBoss
Acegi
Spring MVC
Spring WebFlow
MVC Action
AOP TX Proxy
Business Logic
DAO
Spring-Hibernate
Hibernate
JDBC

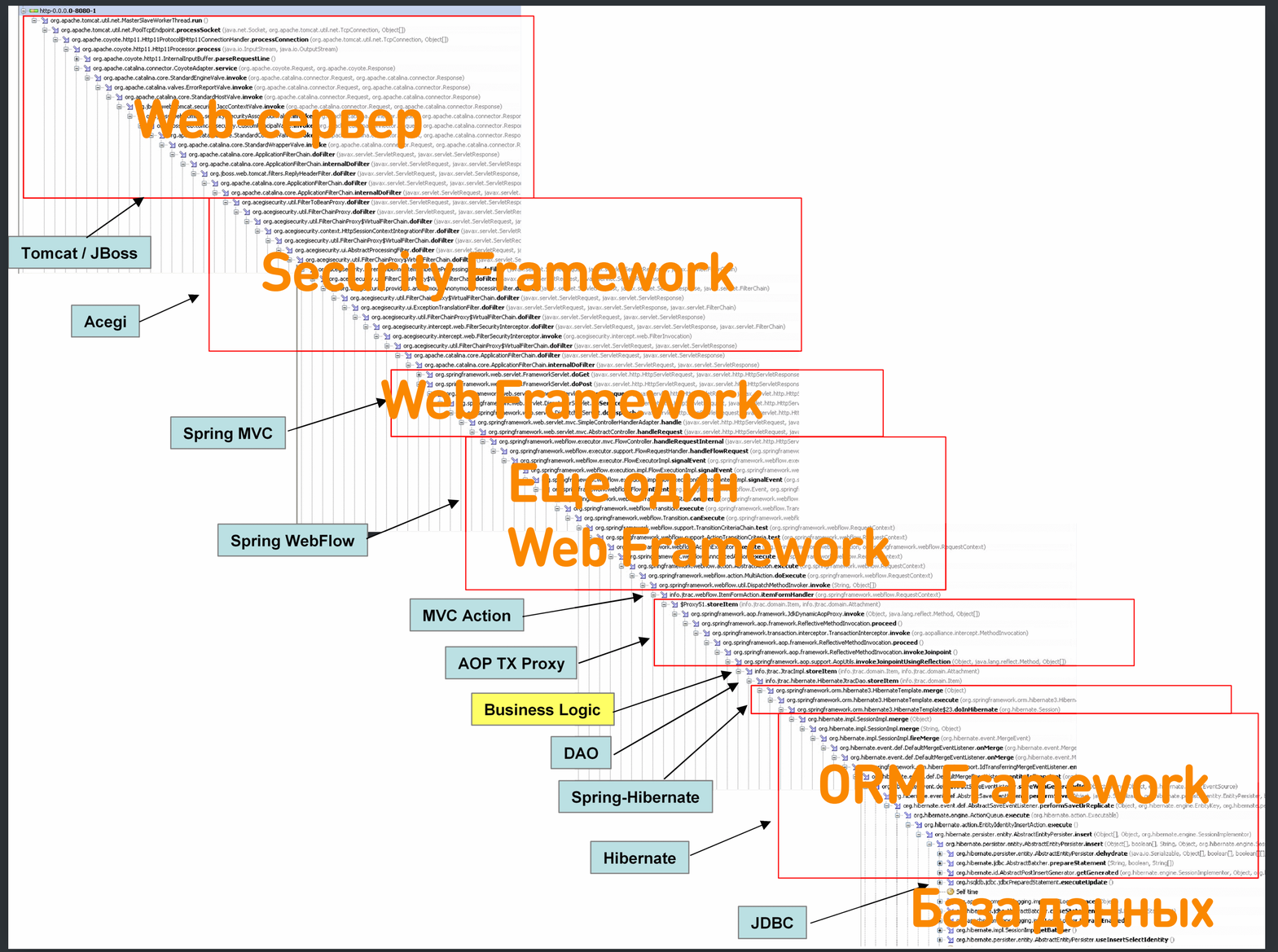
Real Life


Web сервер
Security Framework
Web Framework
Tomcat / JBoss
Acegi
Spring MVC
Spring WebFlow
MVC Action
AOP TX Proxy
Business Logic
DAO
Spring-Hibernate
Hibernate
JDBC

Real Life


Web сервер
Tomcat / JBoss
Security Framework
Acegi
Web Framework
Spring MVC
Еще один
Web Framework
Spring WebFlow
MVC Action
AOP TX Proxy
Business Logic
DAO
Spring-Hibernate
Hibernate
JDBC

Real Life


Web сервер
Tomcat / JBoss
Security Framework
Acegi
Web Framework
Spring MVC
Еще один
Web Framework
Spring WebFlow
MVC Action
AOP TX Proxy
Business Logic
DAO
ORM Framework
Spring-Hibernate
Hibernate
JDBC
Базы данных

2| Практика

Практика

Intro

Готовых рецептов не будет

Практика

Intro

Готовых рецептов не будет

Многое зависит от контекста

Практика

Intro

Готовых рецептов не будет

Многое зависит от контекста

Влияние на команду - must have

3| Easy level

Easy level

Инструменты

- code formatting
- style checks
- static code analysis
- unit tests
- mutation testing

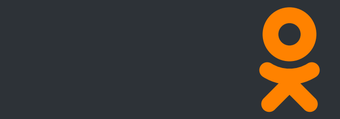
4| Medium level

Medium level

Инструменты

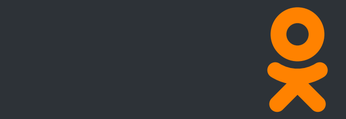
- правильное расположение кода
- правильное оформление кода
- правильные зависимости в коде

Что не так ?



```
@Test
public void testWithWebElement() {
    LoginPage page = new LoginPage(driver);
    page.fillLogin("Heisenbug");
    page.fillPassword("qwe!2#");
    WebElement some = driver.findElement(By.id("rememberMe"));
    some.click();
    page.clickLoginBtn();

}
```



ArchUnit - Demo

Уже было

Источник: https://youtu.be/9mF0zFW7cDQ

Annotation Processing - Demo

Fault Injection Testing

Netflix Chaos Monkey

Simian Army

Gareth Bowles , GTAC 2014

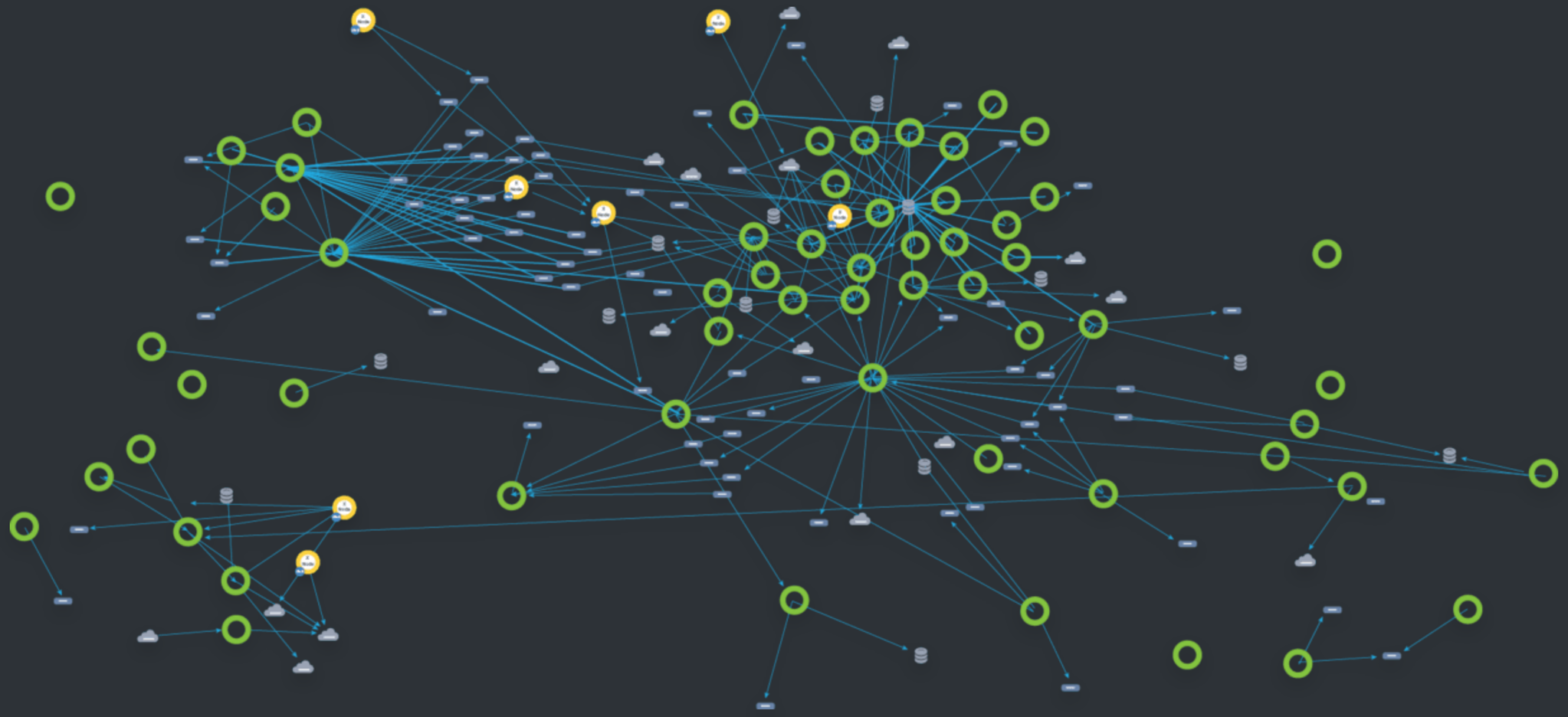
Molly

Peter Alvaro & co. , QCon London 2017

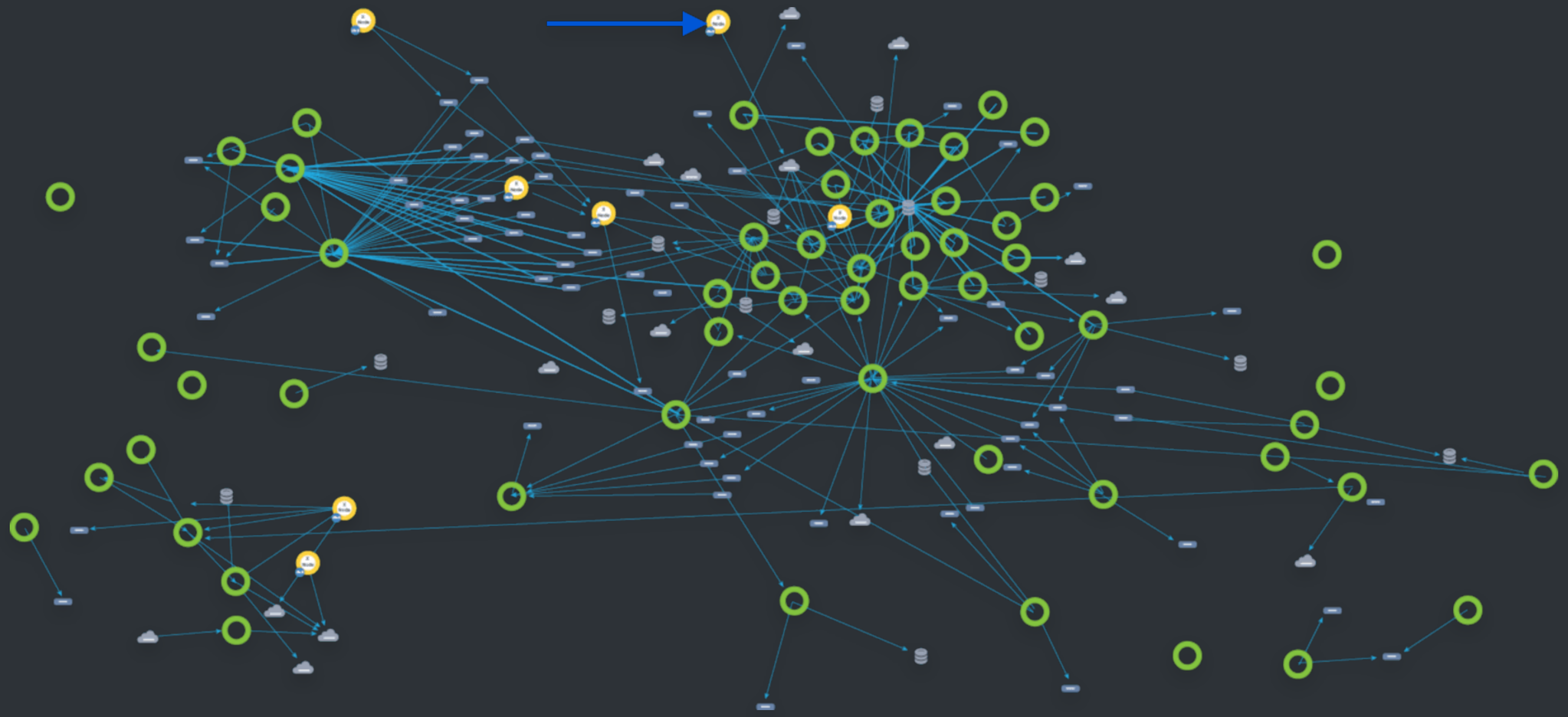
Gremlin

Kolton Andrus, SRECON, 2017

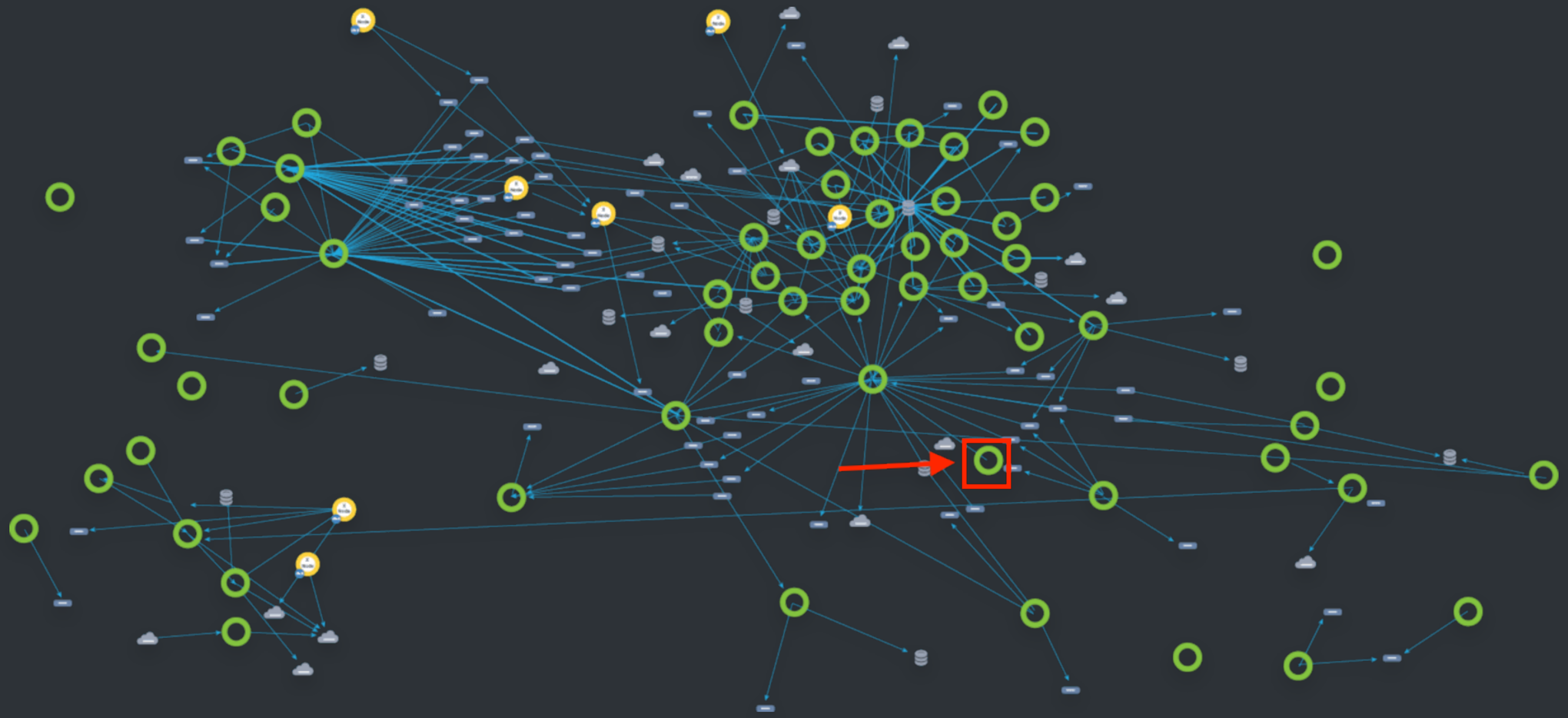
Fault Injection Testing



Fault Injection Testing



Fault Injection Testing



Fault Injection Testing

Агенты

- Latency
- Packet Loss
- Network failures
- CPU Burn
- Disk Burn
- RAM Burn

Fault Injection Testing with Groovy - Demo

4| Nightmare level

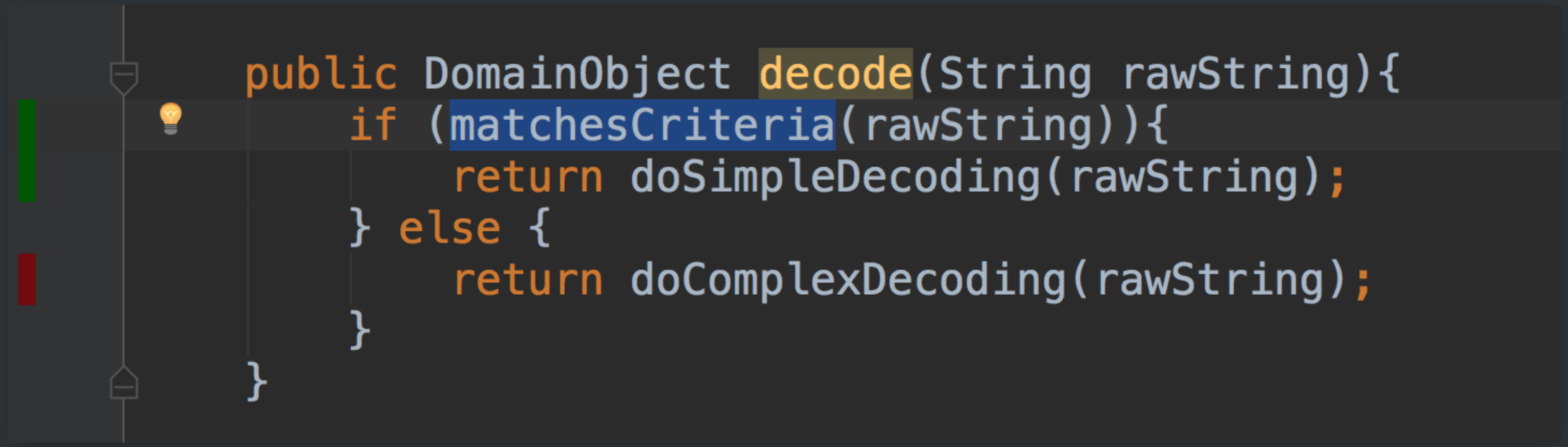
Code Coverage & Co.

Code Coverage

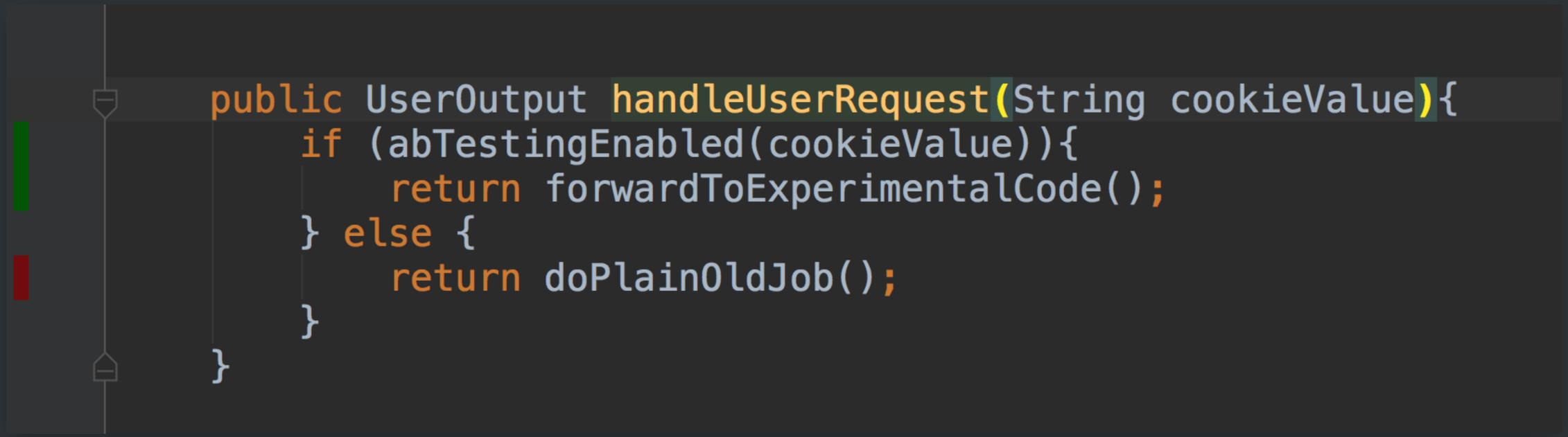
Инструменты

- Jacoco, Cobertura - Java
- OpenCover - .NET
- Coverage - Python
- SimpleCov - Ruby
- OpenCppCoverage - C++
- cover , gcov - Go

Покрытый код



Мертвый код



```
public UserOutput handleUserRequest(String cookieValue){
    if (abTestingEnabled(cookieValue)){
        return forwardToExperimentalCode();
    } else {
        return doPlainOldJob();
    }
}
```

Coverage - Demo

KOVËR (ex. Cover)

RestoreMovieHandler.java

```java
package one.app.community.control.ejb.web.video;

import org.springframework.beans.factory.annotation.Autowired;

import one.discussions.comp.recommender.DiscussionsRecommenderClient;
import one.discussions.comp.recommender.activity.CreateSubjectActivity;
import one.ejb.ValidationException;
import one.ejb.control.EventException;
import one.ejb.control.EventHandlingCtx;
import one.video.entity.album.VideoAlbumId;
import one.video.entity.types.MovieOption;
import one.video.facade.entity.Movie;
import one.video.facade.entity.MovieFull;

public class RestoreMovieHandler extends AMovieEditHandler<RestoreMovieEvent, UpdateMovieResult> {

    private static final long serialVersionUID = 1L;

    @Autowired
    private DiscussionsRecommenderClient discussionsRecommenderClient;

    @Override
    protected void handleEvent(EventHandlingCtx ctx, RestoreMovieEvent event, UpdateMovieResult result)
            throws ValidationException, EventException {
        Movie movieAll = videoService.getMovieAll(event.getMovieId());
        checkMoviePermissions(event.getMovieId(), event.getUserId(), movieAll);

        MovieFull movie;

        if (event.getAlbumId() != null) {
            movie = videoService.unDelete(event.getMovieId(), event.getAlbumId());
        } else {
            movie = videoService.unDelete(event.getMovieId());
        }

        result.setMovie(movie);

        if (!movie.isOptionSet(MovieOption.GROUP_OFFICIAL)) {
            final Long groupId = movie.getGroupId();
            discussionsRecommenderClient.register(movie.getOwnerUserId(), movie.getOwnerUserId(), groupId,
                    videoHelper.toDiscussionId(movie),
                    new CreateSubjectActivity(movie.getCreated()));
        }

        logStat("restoreMovie", movie, event.getUserId(), true);

        ctx.handleNewEvent(new AddMovieToAntiSpamEvent(event.getUserId(), movie));

    }

}
```

RestoreMovieHandler.java

```java
package one.app.community.control.ejb.web.video;

import org.springframework.beans.factory.annotation.Autowired;

import one.discussions.comp.recommender.DiscussionsRecommenderClient;
import one.discussions.comp.recommender.activity.CreateSubjectActivity;
import one.ejb.ValidationException;
import one.ejb.control.EventException;
import one.ejb.control.EventHandlingCtx;
import one.video.entity.album.VideoAlbumId;
import one.video.entity.types.MovieOption;
import one.video.facade.entity.Movie;
import one.video.facade.entity.MovieFull;

public class RestoreMovieHandler extends AMovieEditHandler<RestoreMovieEvent, UpdateMovieResult> {

    private static final long serialVersionUID = 1L;

    @Autowired
    private DiscussionsRecommenderClient discussionsRecommenderClient;

    @Override
    protected void handleEvent(EventHandlingCtx ctx, RestoreMovieEvent event, UpdateMovieResult result)
            throws ValidationException, EventException {
        Movie movieAll = videoService.getMovieAll(event.getMovieId());
        checkMoviePermissions(event.getMovieId(), event.getUserId(), movieAll);

        MovieFull movie;

        if (event.getAlbumId() != null) {
            movie = videoService.unDelete(event.getMovieId(), event.getAlbumId());
        } else {
            movie = videoService.unDelete(event.getMovieId());
        }

        result.setMovie(movie);

        if (!movie.isOptionSet(MovieOption.GROUP_OFFICIAL)) {
            final Long groupId = movie.getGroupId();
            discussionsRecommenderClient.register(movie.getOwnerUserId(), movie.getOwnerUserId(), groupId,
                    videoHelper.toDiscussionId(movie),
                    new CreateSubjectActivity(movie.getCreated()));
        }

        logStat("restoreMovie", movie, event.getUserId(), true);

        ctx.handleNewEvent(new AddMovieToAntiSpamEvent(event.getUserId(), movie));

    }

}
```

RestoreMovieHandler.java

```java
package one.app.community.control.ejb.web.video;

import org.springframework.beans.factory.annotation.Autowired;

import one.discussions.comp.recommender.DiscussionsRecommenderClient;
import one.discussions.comp.recommender.activity.CreateSubjectActivity;
import one.ejb.ValidationException;
import one.ejb.control.EventException;
import one.ejb.control.EventHandlingCtx;
import one.video.entity.album.VideoAlbumId;
import one.video.entity.types.MovieOption;
import one.video.facade.entity.Movie;
import one.video.facade.entity.MovieFull;

public class RestoreMovieHandler extends AMovieEditHandler<RestoreMovieEvent, UpdateMovieResult> {

    private static final long serialVersionUID = 1L;

    @Autowired
    private DiscussionsRecommenderClient discussionsRecommenderClient;

    @Override
    protected void handleEvent(EventHandlingCtx ctx, RestoreMovieEvent event, UpdateMovieResult result)
            throws ValidationException, EventException {
        Movie movieAll = videoService.getMovieAll(event.getMovieId());
        checkMoviePermissions(event.getMovieId(), event.getUserId(), movieAll);

        MovieFull movie;

        if (event.getAlbumId() != null) {
            movie = videoService.unDelete(event.getMovieId(), event.getAlbumId());
        } else {
            movie = videoService.unDelete(event.getMovieId());
        }

        result.setMovie(movie);

        if (!movie.isOptionSet(MovieOption.GROUP_OFFICIAL)) {
            final Long groupId = movie.getGroupId();
            discussionsRecommenderClient.register(movie.getOwnerUserId(), movie.getOwnerUserId(), groupId,
                    videoHelper.toDiscussionId(movie),
                    new CreateSubjectActivity(movie.getCreated()));
        }

        logStat("restoreMovie", movie, event.getUserId(), true);

        ctx.handleNewEvent(new AddMovieToAntiSpamEvent(event.getUserId(), movie));

    }

}
```

Автотесты

RestoreMovieHandler.java

```java
package one.app.community.control.ejb.web.video;

import org.springframework.beans.factory.annotation.Autowired;

import one.discussions.comp.recommender.DiscussionsRecommenderClient;
import one.discussions.comp.recommender.activity.CreateSubjectActivity;
import one.ejb.ValidationException;
import one.ejb.control.EventException;
import one.ejb.control.EventHandlingCtx;
import one.video.entity.album.VideoAlbumId;
import one.video.entity.types.MovieOption;
import one.video.facade.entity.Movie;
import one.video.facade.entity.MovieFull;

public class RestoreMovieHandler extends AMovieEditHandler<RestoreMovieEvent, UpdateMovieResult> {

    private static final long serialVersionUID = 1L;

    @Autowired
    private DiscussionsRecommenderClient discussionsRecommenderClient;

    @Override
    protected void handleEvent(EventHandlingCtx ctx, RestoreMovieEvent event, UpdateMovieResult result)
            throws ValidationException, EventException {
        Movie movieAll = videoService.getMovieAll(event.getMovieId());
        checkMoviePermissions(event.getMovieId(), event.getUserId(), movieAll);

        MovieFull movie;

        if (event.getAlbumId() != null) {
            movie = videoService.unDelete(event.getMovieId(), event.getAlbumId());
        } else {
            movie = videoService.unDelete(event.getMovieId());
        }

        result.setMovie(movie);

        if (!movie.isOptionSet(MovieOption.GROUP_OFFICIAL)) {
            final Long groupId = movie.getGroupId();
            discussionsRecommenderClient.register(movie.getOwnerUserId(), movie.getOwnerUserId(), groupId,
                    videoHelper.toDiscussionId(movie),
                    new CreateSubjectActivity(movie.getCreated()));
        }

        logStat("restoreMovie", movie, event.getUserId(), true);

        ctx.handleNewEvent(new AddMovieToAntiSpamEvent(event.getUserId(), movie));

    }

}
```

Пользователи

Что делать?

	Люди «+»	Люди «-»
АТ «+»	сравнить	подумать
АТ «-»	писать автотесты	удалять код

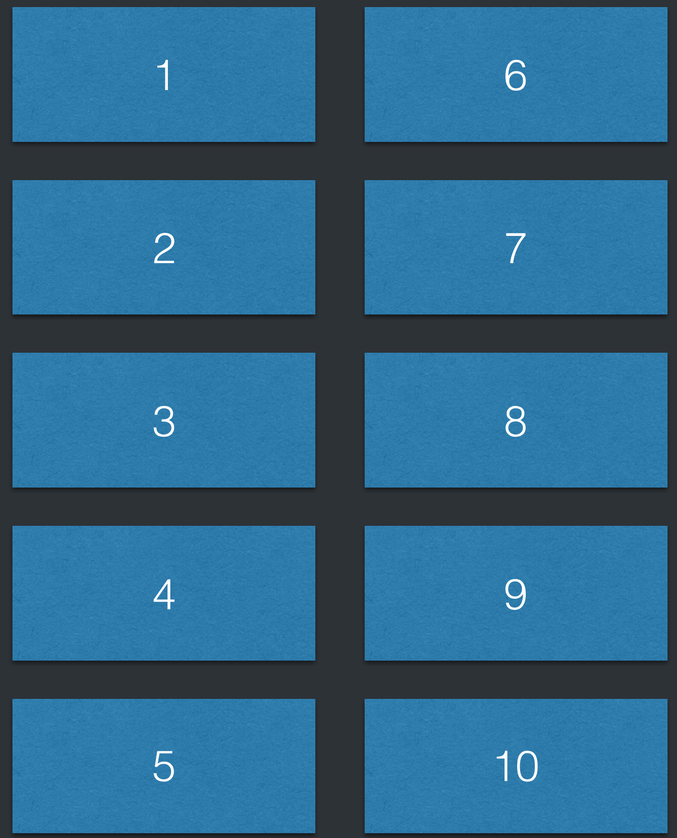
Code Coverage

Что можно сделать с покрытием ?

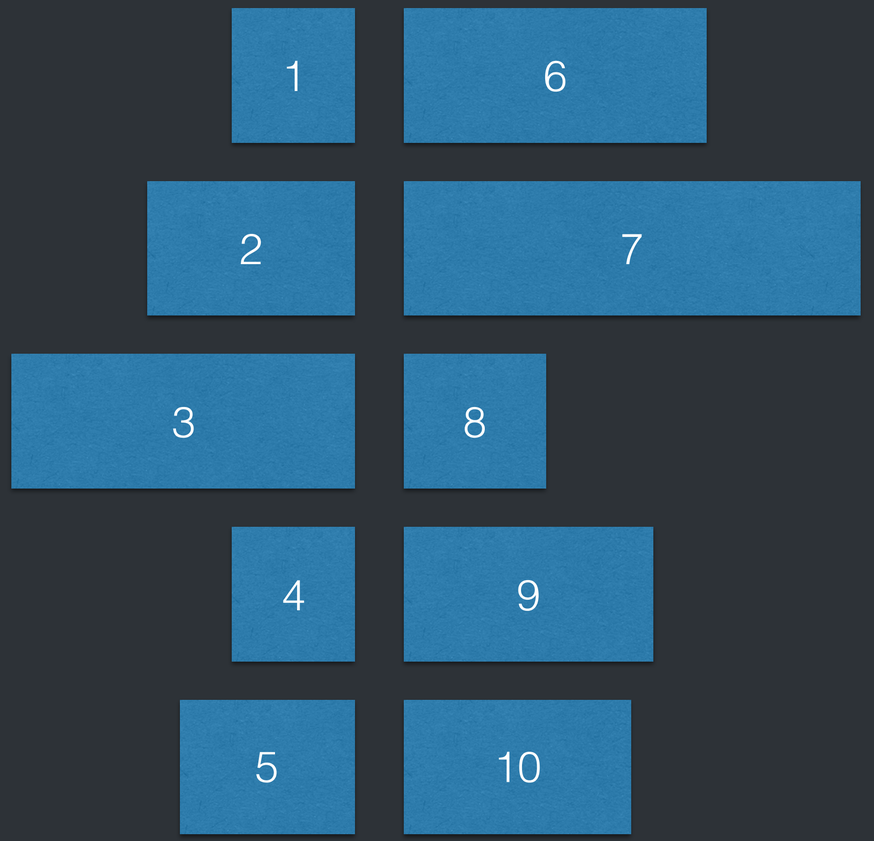
- интроспекция для ручного тестирования
- мерило качества для автотестов
- мертвый код и мертвые фичи

Метаинформация

Задача о рюкзаке



Задача о рюкзаке

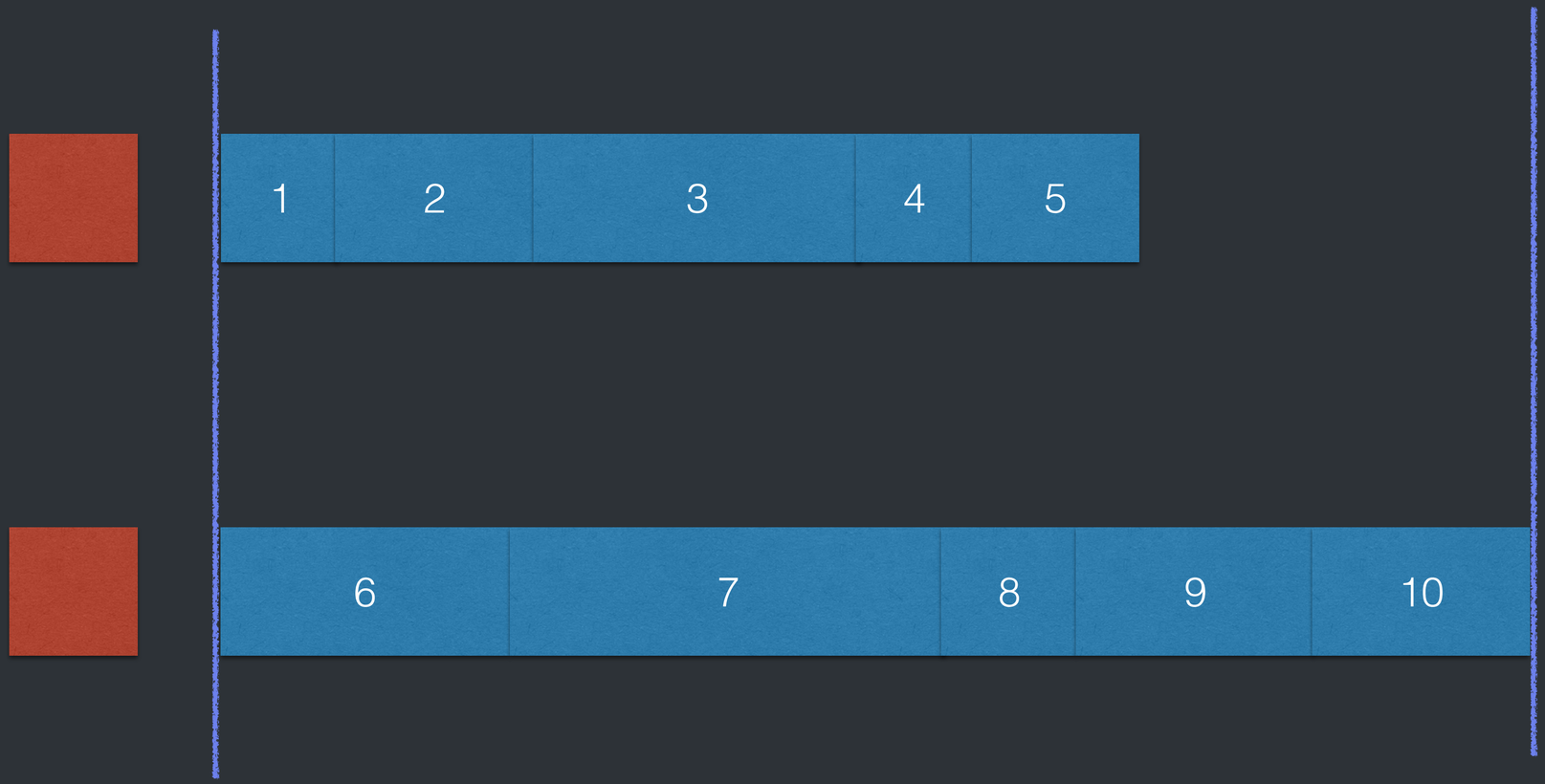


Задача о рюкзаке

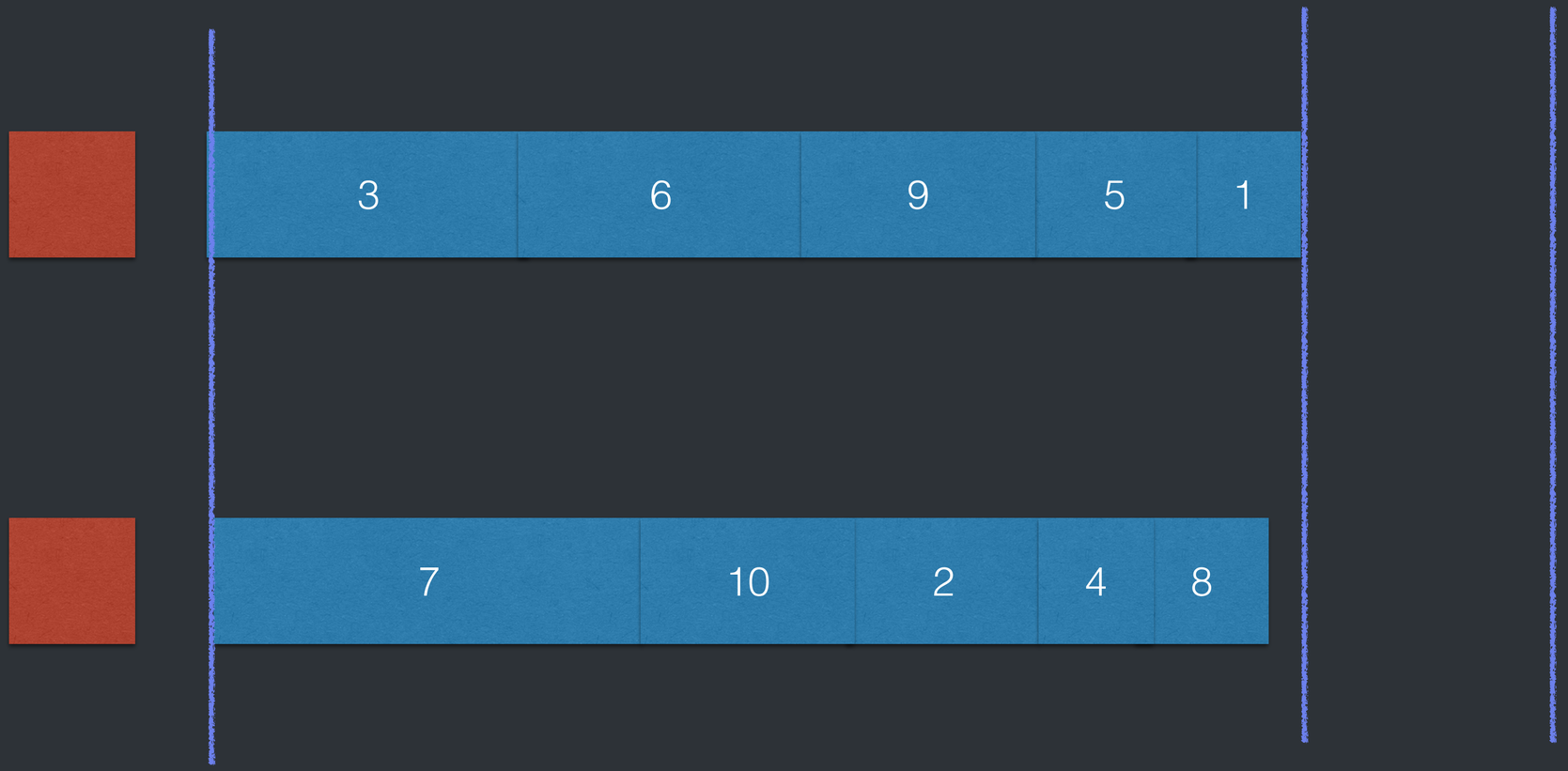
Ресурс для запуска тестов № 1

Ресурс для запуска тестов № 2

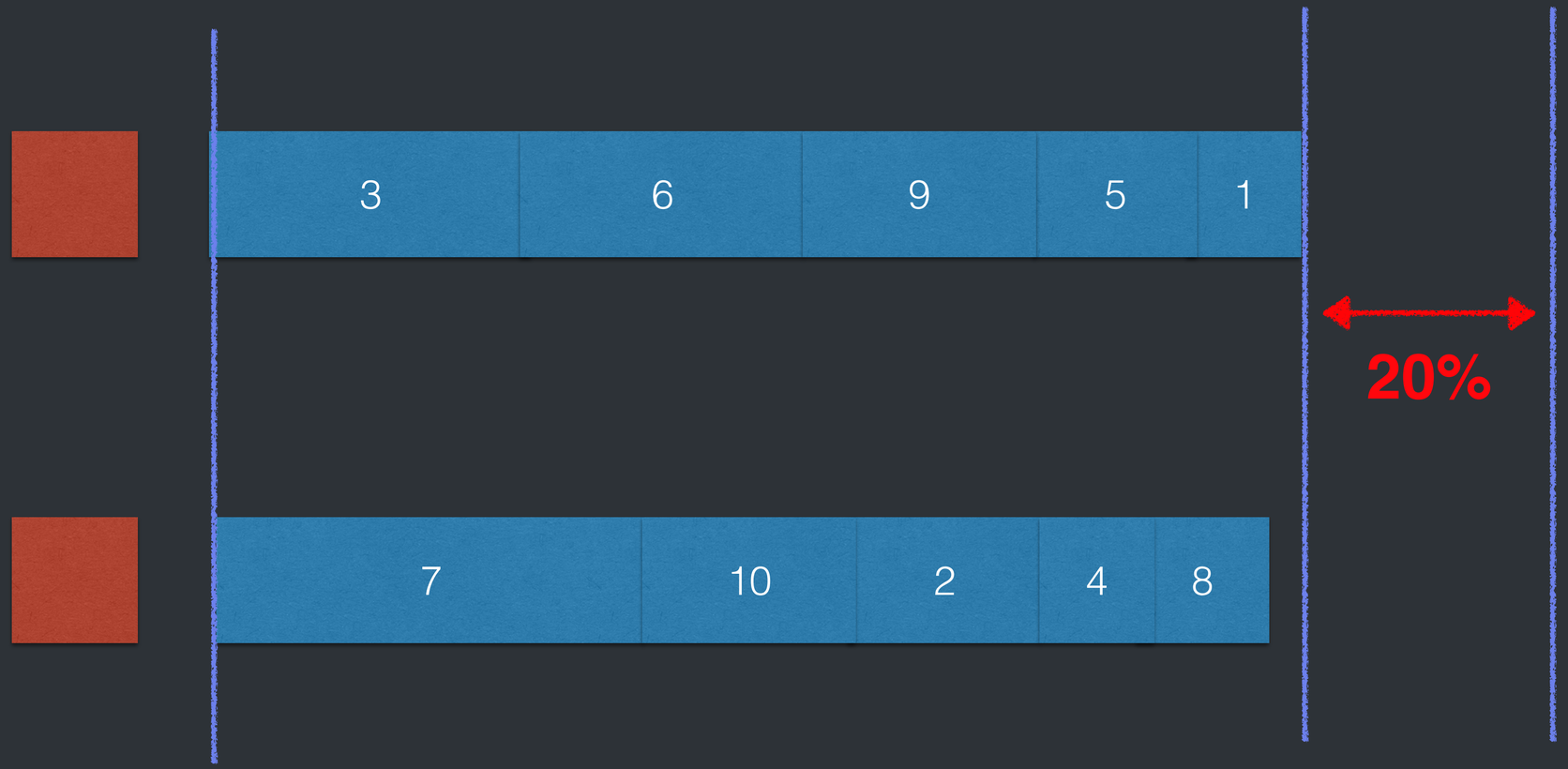
Задача о рюкзаке



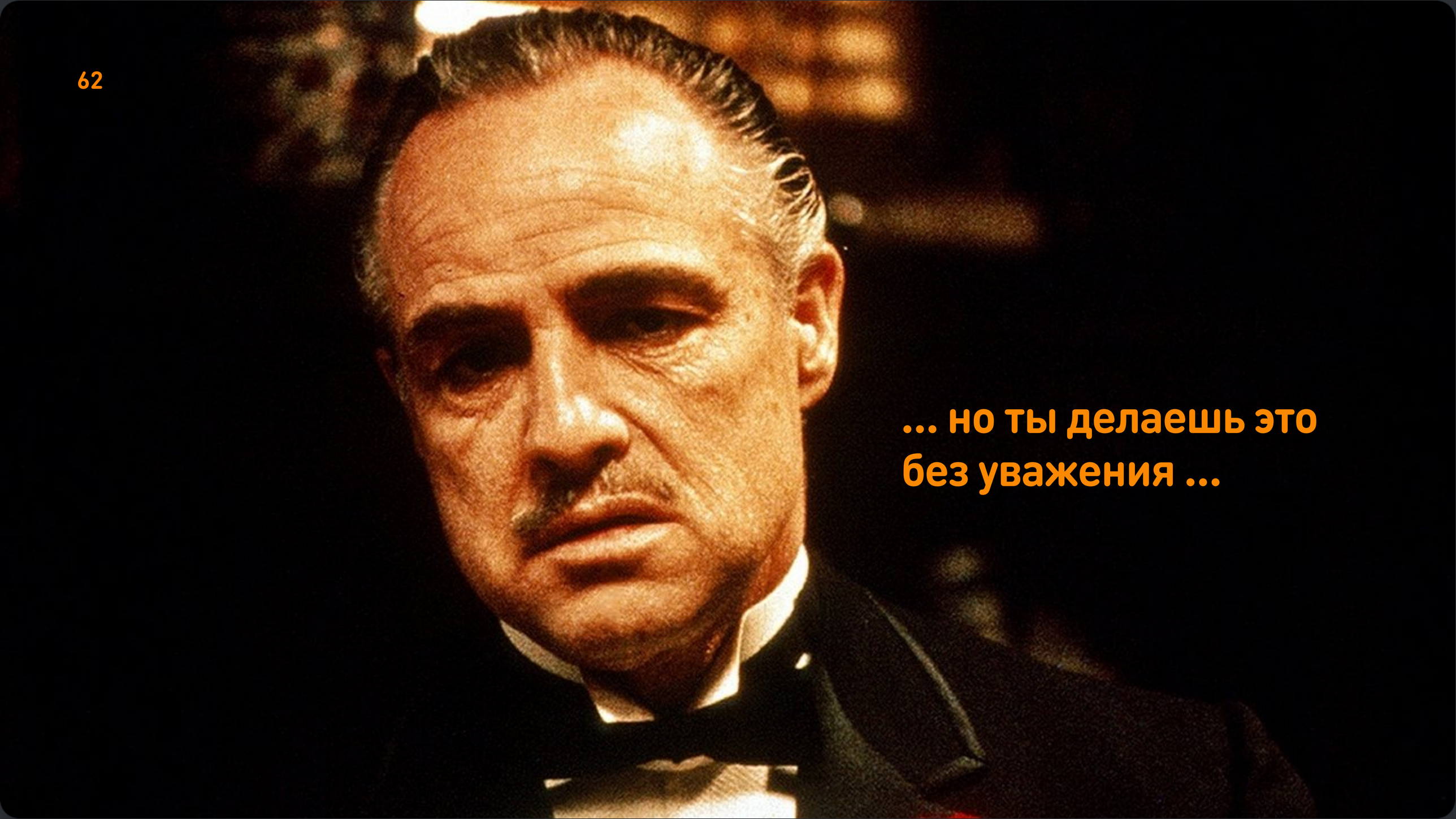
Задача о рюкзаке



Задача о рюкзаке

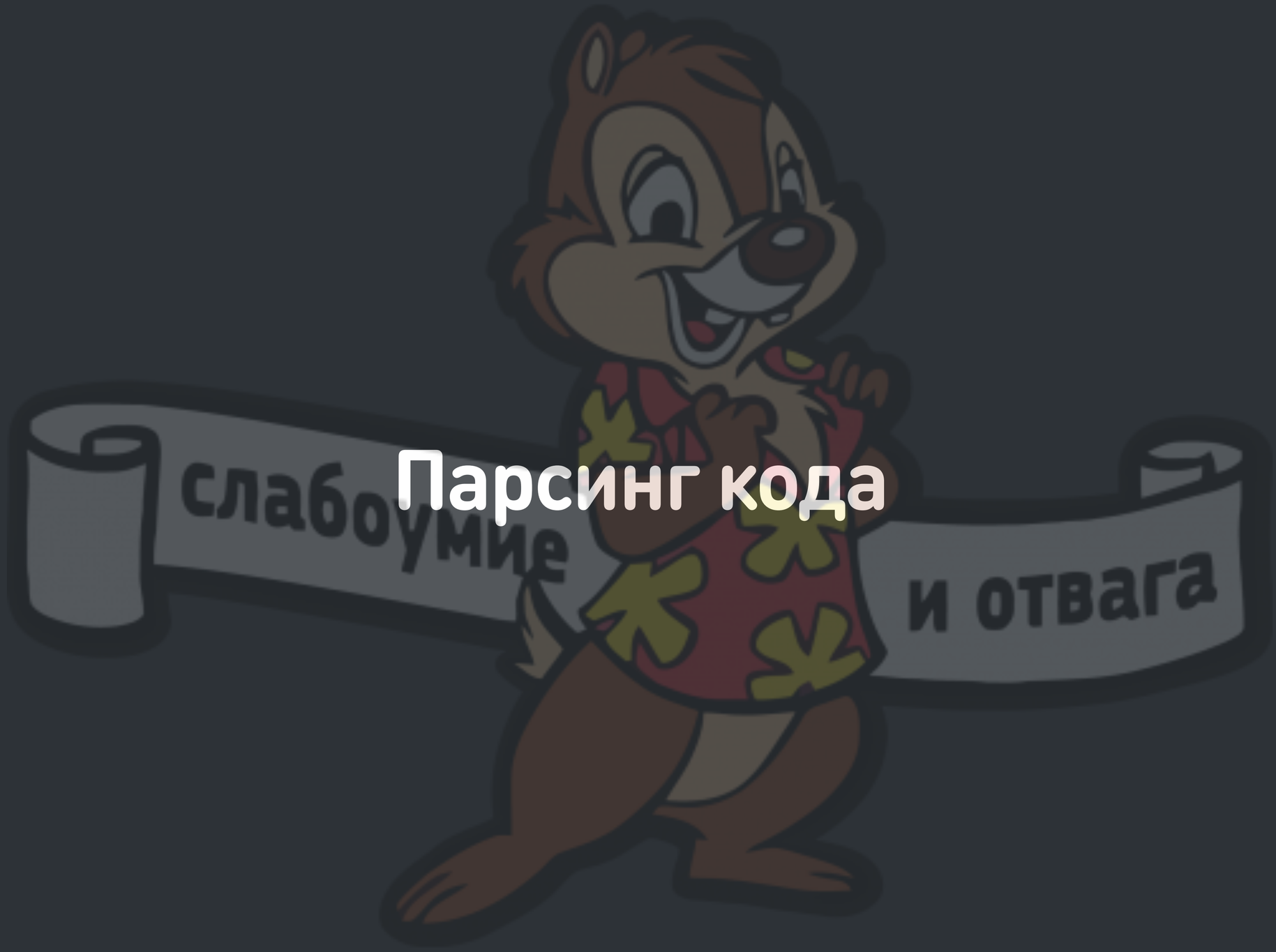



- Ты хочешь запустить
тесты в 200 потоков...



... но ты делаешь это
без уважения ...

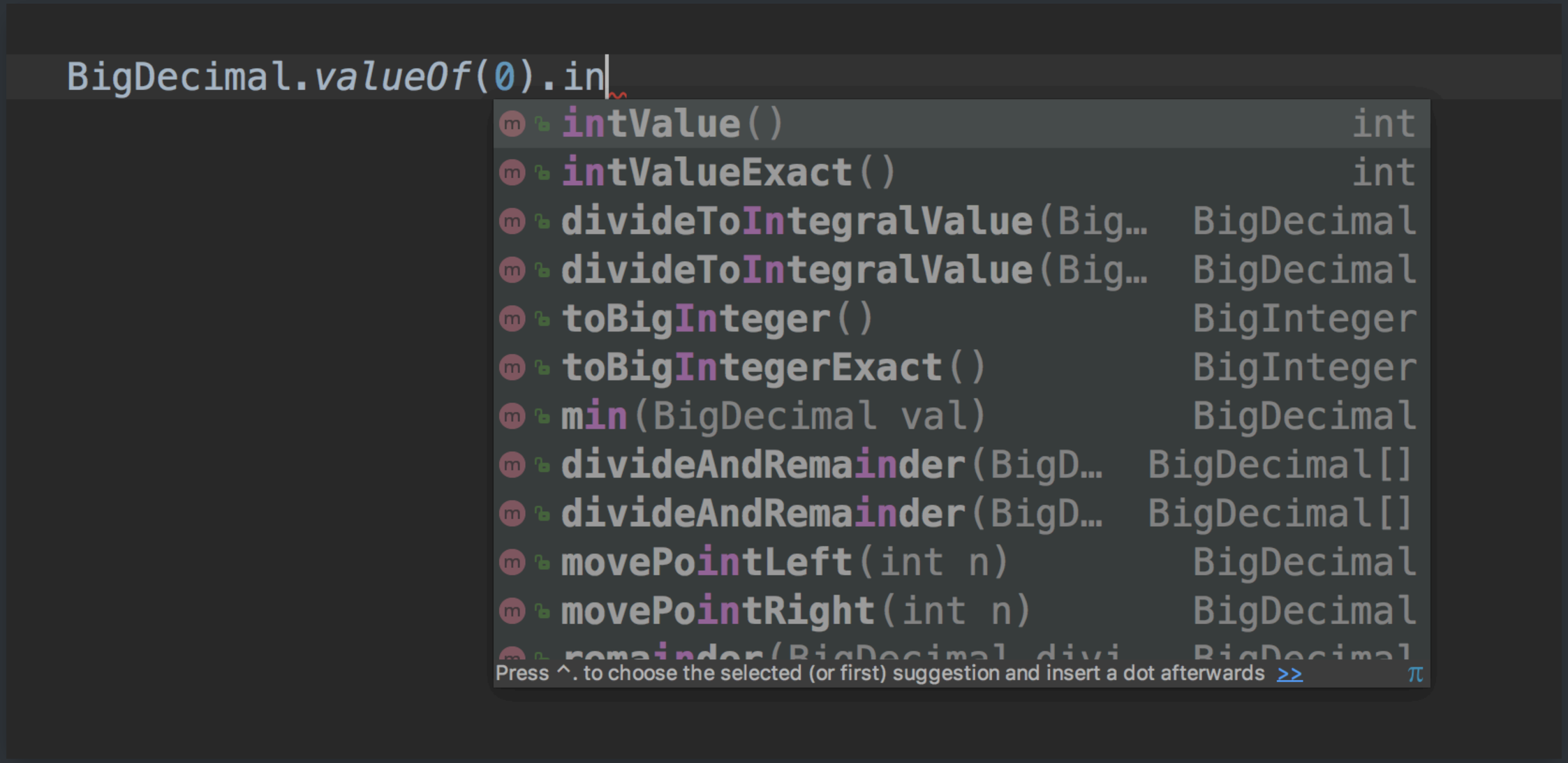

- и тестов у тебя
... всего 50



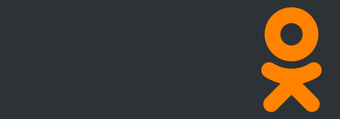
Парсинг кода

Все IDE делают это!

Все IDE делают это!



Парсинга кода, Abstract Syntax Tree - WAT?



```
ASTParser parser = ASTParser.newParser(AST.JLS8);
parser.setSource(source);
parser.setResolveBindings(true);
parser.setKind(ASTParser.K_COMPILATION_UNIT);
ASTNode ast = parser.createAST(progressMonitor);
CompilationUnit cu = (CompilationUnit) ast;
```

Источник: http://bit.ly/2j8gype

BERRIMOR

BERRIMOR

Что умеет делать ?

- выкачивать код из репозиториев
- парсить код
- выделяет метаинформацию
    - считать тесты
    - получать метаинформацию из тестов (тэги, отключенные тесты)
    - знает владельцев тестов

Social Code Analysis

Коллективная ментальная модель - С. Архипенков

...реальность, которая заключена в особой специфике производства программ, по сравнению с любой другой производственной деятельностью, потому что то, что производят программисты – **нематериально, это коллективные ментальные модели, записанные на языке программирования**.

Данные: Сергей Архипенков, Лекции по управлению программными проектами, 2009

Социальный анализ кода?

Что можно сделать ?

- Mining Repository Data to Debug Software Development Teams , Elmar Juergens
- Seven Secrets of Maintainable Codebases , Adam Tornhill
- How Flaky Tests in Continuous Integration: Current Practice at Google and Future Directions, John Micco, Atif Memon

Социальный анализ кода

Что можно узнать ?

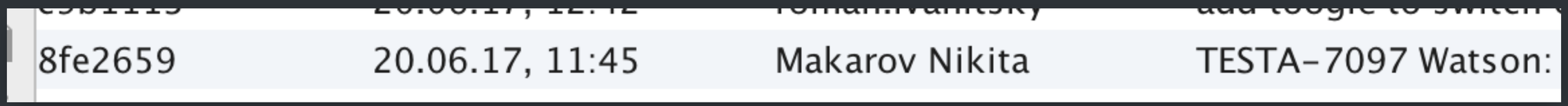
- кто чинит и кто ломает
- неявные связи в коде
- где чаще всего фиксят
- мертвый код и мертвые фичи
- технический долг

Слабая гипотеза о техническом долге

Где у нас закопан технический долг ?

- в баг-трекере есть баги и у них есть ID
- в Git есть коммиты и в комментариях пишут ID таски
- **файлы в Git в которых часто чинят - это технический долг**

Дисциплина - это хорошо!



Kafka

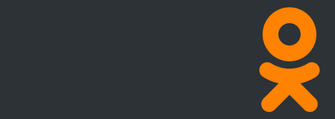
Источники

- https://issues.apache.org/jira/projects/KAFKA
- https://github.com/apache/kafka
- Прекрасное в репозитории http://bit.ly/2yfc7lA и http://bit.ly/2wO5ByG

5| Итоги

Белый ящик - это стратегия!

Читай код!



Читай код, Карл!!!

nikita.makarov@odnoklassniki.ru

@PapaMinos

http://test-failed.blogspot.ru/

одноклассники

Примеры

Medium Level

- ArchUnit Demo - https://goo.gl/kGKF9T
- Annotation Processing Demo
  - Library - https://goo.gl/GcuLC3
  - Test Project - https://goo.gl/eynhCr
- Fault Injection Testing with Groovy Demo - https://goo.gl/SN4ooJ

Примеры

Nightmare Level

- Code Coverage Demo - https://goo.gl/YJfQQB
- Social Coding Analysis - https://goo.gl/jd1WPD