



Максим Щепалин
Тинькофф.Бизнес

Test Impact Analysis

Для Android & JVM



ТИНЬКОФФ

tinkoff.ru

Обо мне

- Платформ-тех разработчик в Тинькофф.Бизнес
- Занимаюсь автоматизациями, инфраструктурой, CI/CD



О чем мы поговорим

О чем мы поговорим

01

Что такое импакт-анализ и зачем он нужен

О чем мы поговорим

01

Что такое импакт-анализ и зачем он нужен

02

Как написать свой импакт-анализ

О чем мы поговорим

01

Что такое импакт-анализ и зачем он нужен

02

Как написать свой импакт-анализ

03

Итоги

Постановка проблемы



Тинькофф.Бизнес 2019 vs 2021



Бизнес требует все больше фич

Тинькофф.Бизнес 2019 vs 2021



Бизнес требует все больше фич



Нанимаем больше разработчиков

Тинькофф.Бизнес 2019 vs 2021



Бизнес требует все больше фич



Нанимаем больше разработчиков



Нужно тестировать больше кода

Тинькофф.Бизнес 2019 vs 2021



Бизнес требует все больше фич



Нанимаем больше разработчиков



Нужно тестировать больше кода



Пишем больше автотестов

Сезон автотестов

Сезон автотестов



Автотесты сильно сокращают регресс

Сезон автотестов

 Автотесты сильно сокращают регресс

 Ни одна фича не попадает в прод без тестов

Сезон автотестов



АВТОТЕСТЫ APP 8:13 PM

Задачи, не покрытые автотестами 📍

[@ext.echernetsov](#) ЕМА-14909

Дружище, напиши тесты, иначе фича не попадет в ближайшую версию.

Сезон автотестов

 Автотесты сильно сокращают регресс

 Ни одна фича не попадает в прод без тестов

 А как их поддерживать?

Сезон автотестов



AutoTest Reporter APP 8:02 AM

Список сломанных автотестов и подробный Allure-репорт

@e.kursakov @ext.vglushen @m.shanko

1 Failed

@k.bersenev @d.larkin @k.lizunova

1 Failed

@ext.nszintsev @e.kursakov @m.shanko

1 Failed

@ext.ekostyshina @a.kazaeva

1 Failed

@n.v.melnik @y.v.barsukova

1 Failed

@n.v.melnik @a.kazaeva

2 Failed

@v.postnov @m.shanko

2 Failed

@v.postnov @g.ulizko

1 Failed

@ext.echernetsov @e.kalugina

1 Failed

1210 tests 1199 success 11 failed (из них 1 crashed)

А если тестов больше?



ТИНЬКОФФ

tinkoff.ru



AutoTest Reporter APP 12:41 AM

Список сломанных UI автотестов на ветке master
подробный отчет в Allure

| | |
|--|-----------|
| @ext.omazhukin @b.kataev | 2 Failed |
| @e.kostyshina @n.sharonov @ext.kvasilchenko @a.kazaeva | 16 Failed |
| @n.zaltsman @ext.agavrilenko | 20 Failed |
| @g.sadovskiy @a.sovetnikov | 7 Failed |
| @ext.vglushen @m.shanko | 24 Failed |
| @ext.echernetsov @e.kalugina | 32 Failed |
| @ext.nszintsev @ext.zshakirov @k.zudov @k.mamakov | 11 Failed |
| @g.brundaev | 21 Failed |
| @a.a.melnik @a.o.pavlova | 10 Failed |
| @ext.nszintsev @g.ulizko | 1 Failed |
| @v.postnov @g.ulizko @e.abdulmanov | 6 Failed |
| @a.galiullin @a.yarovikov @m.tavrova | 20 Failed |
| @vi.v.kharlamov @an.rudenko @a.o.lobanovskiy | 3 Failed |
| @n.v.melnik @y.v.barsukova @d.krivososov | 5 Failed |
| @an.rudenko @p.rybakov | 1 Failed |
| @e.streshnev @n.donskaya | 1 Failed |

2746 tests 2566 success 180 failed (из них 0 crashed)

Поддержка автотестов



В какой-то момент тестов
становится слишком много

Поддержка автотестов



В какой-то момент тестов
становится слишком много



Все тесты прогоняются ночью

Поддержка автотестов



В какой-то момент тестов
становится слишком много



Все тесты прогоняются ночью



Автотесты имеют свойство
протухать

Поддержка автотестов



В какой-то момент тестов
становится слишком много



Все тесты прогоняются ночью



Автотесты имеют свойство
протухать



Протухшие автотесты замедляют
регресс

Решение проблемы



Поддержка автотестов



Прогонять все тесты на пулл-реквестах

Поддержка автотестов



Прогонять все тесты на пулл-реквестах



Прогонять только определенные тесты на пулл-реквестах

Поддержка автотестов



Прогонять все тесты на пулл-реквестах



Прогонять только определенные тесты на пулл-реквестах



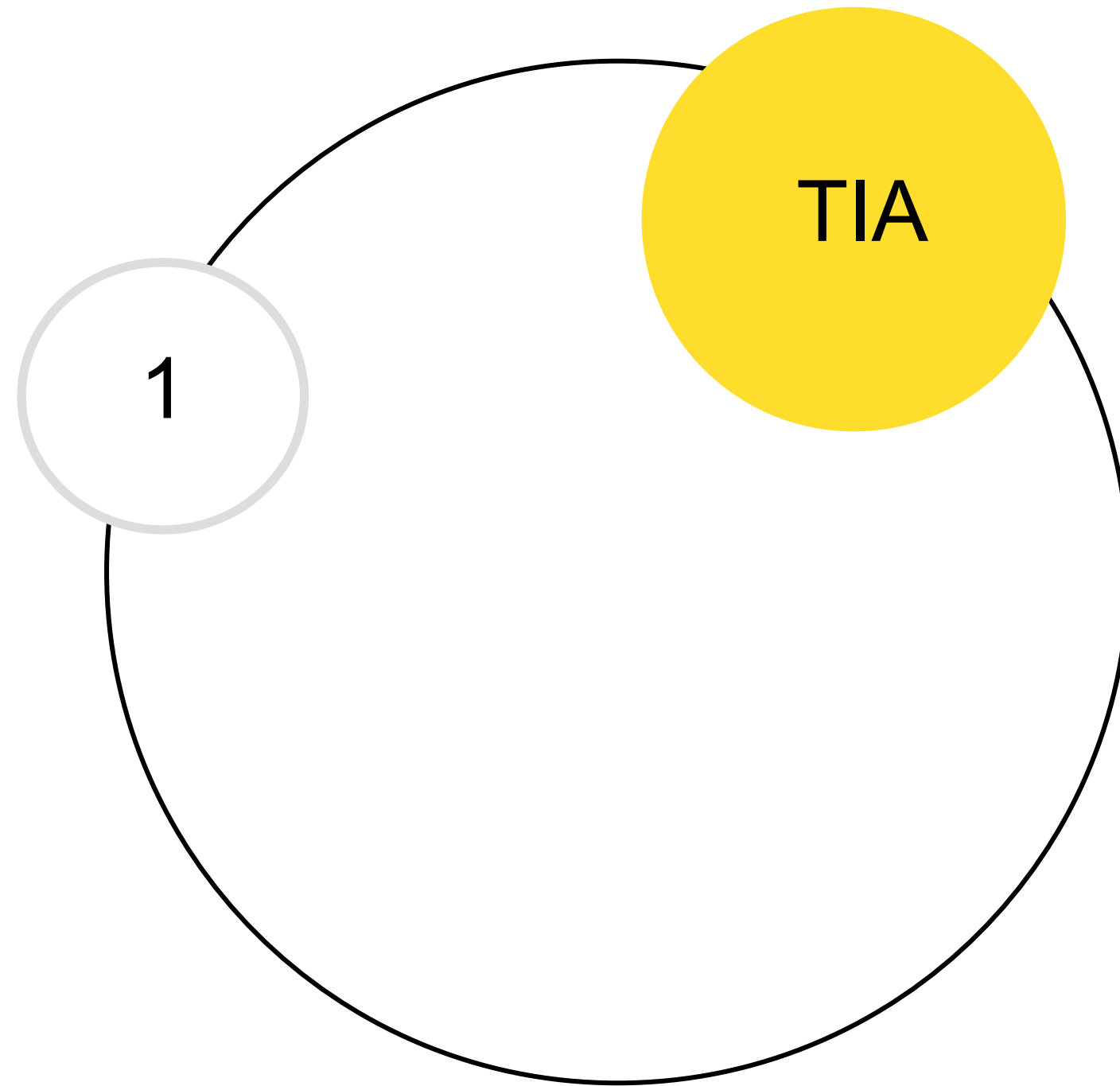
Внедрить импакт-анализ

Тест импакт-анализ

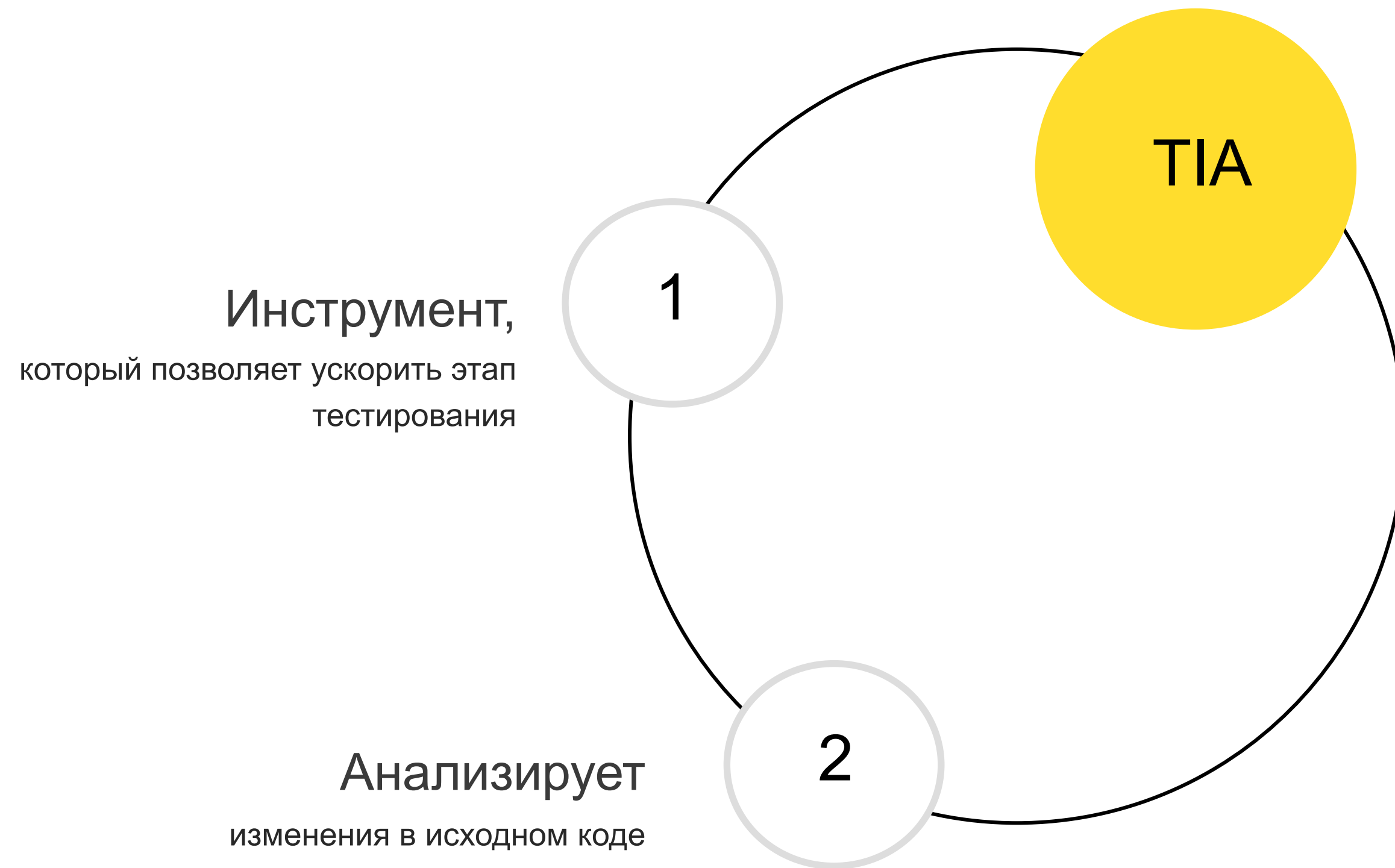


Что такое TIA

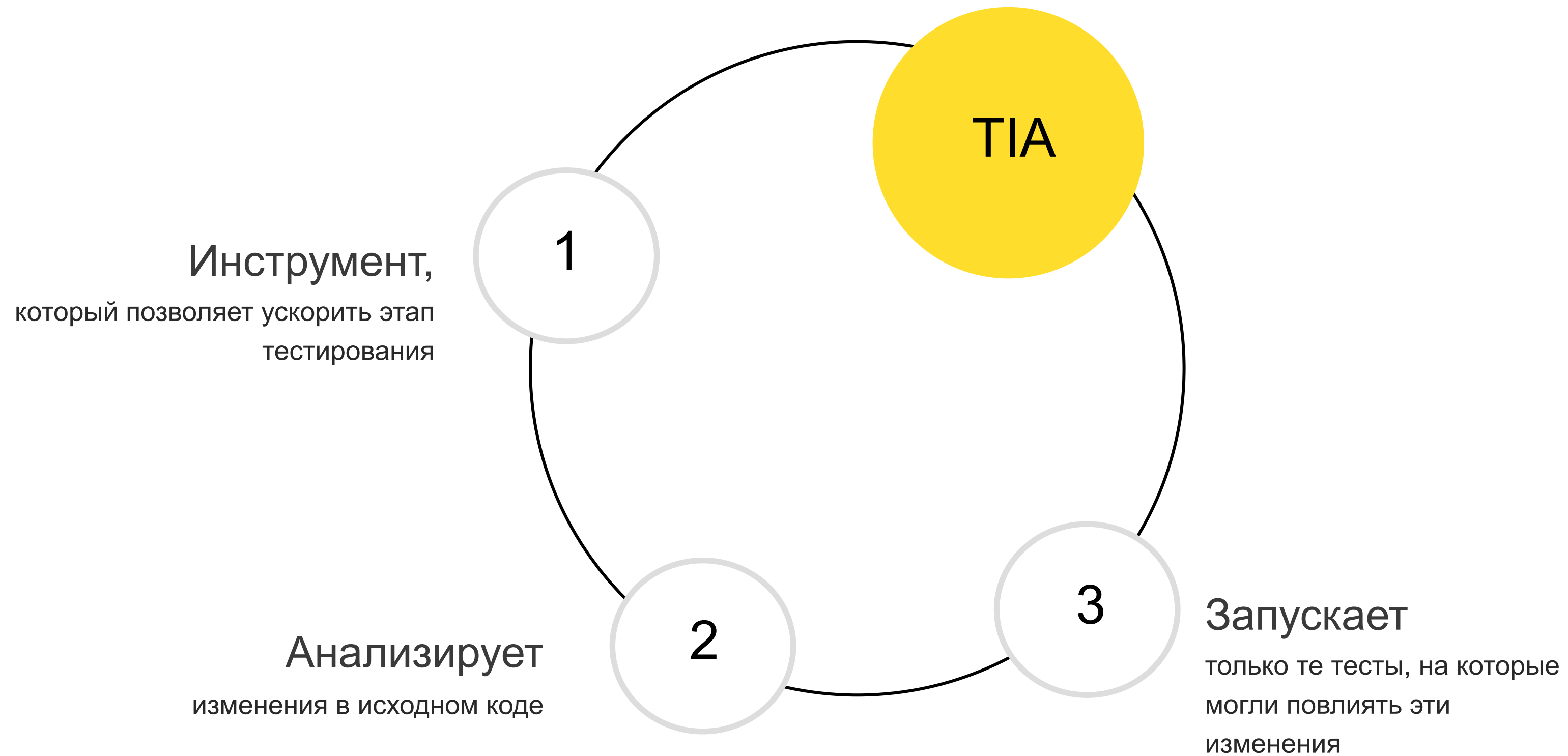
Инструмент,
который позволяет ускорить этап
тестирования



Что такое TIA



Что такое TIA



ContactValidationTest

```
51      waitUntilNotificationIsGone()  
52  
53      -  
54      -  
55      + setValueAndConfirm("01.01.1000")  
56      assert { smartFieldWithError() }  
57      assert { errorNotificationNotShowned() }  
58  }
```

Варианты реализации ТИА



File Detector

File Detector



Смотрим Git Diff

File Detector

Смотрим Git Diff

Вычленяем сет
затронутых модулей

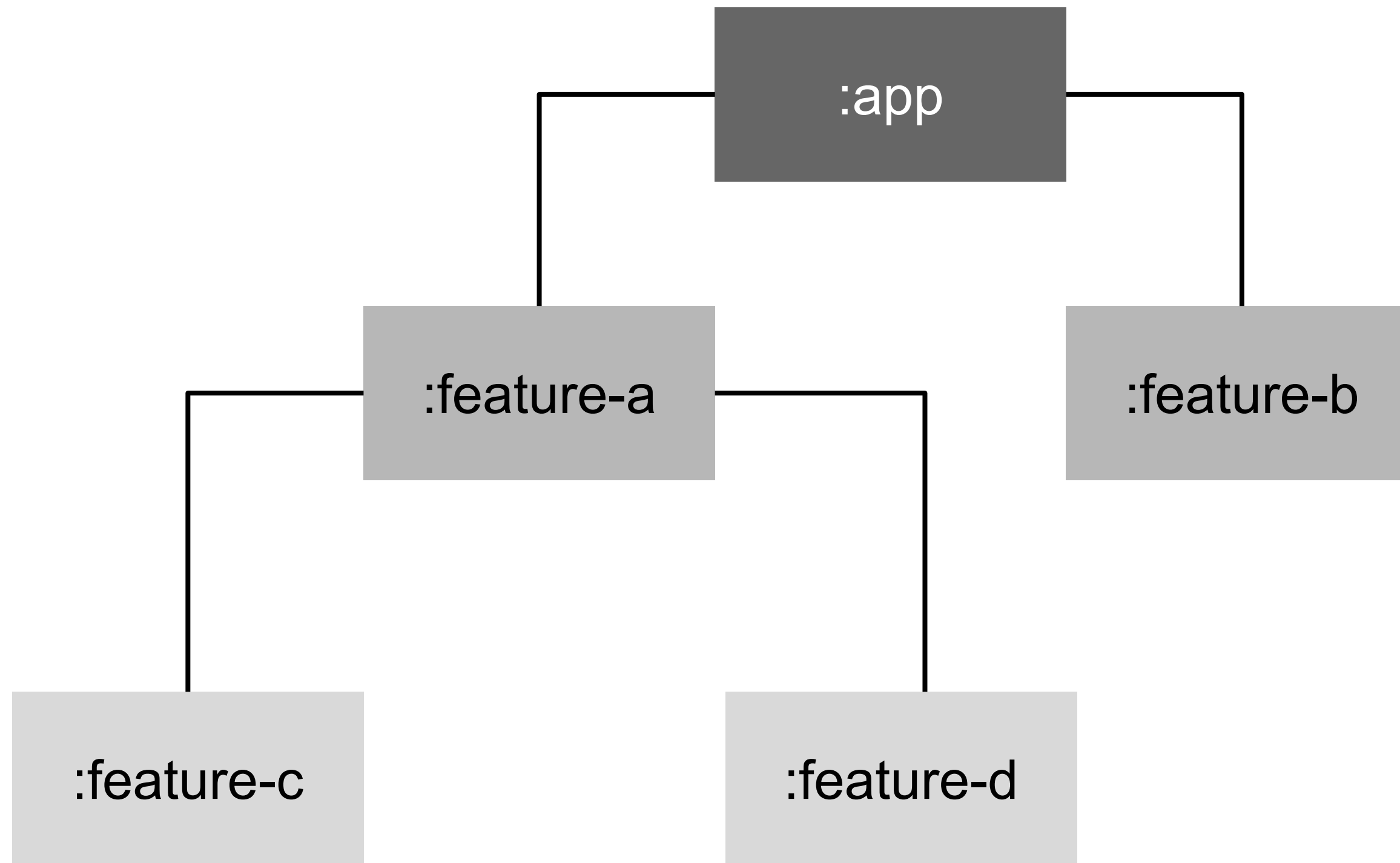
File Detector

Смотрим Git Diff

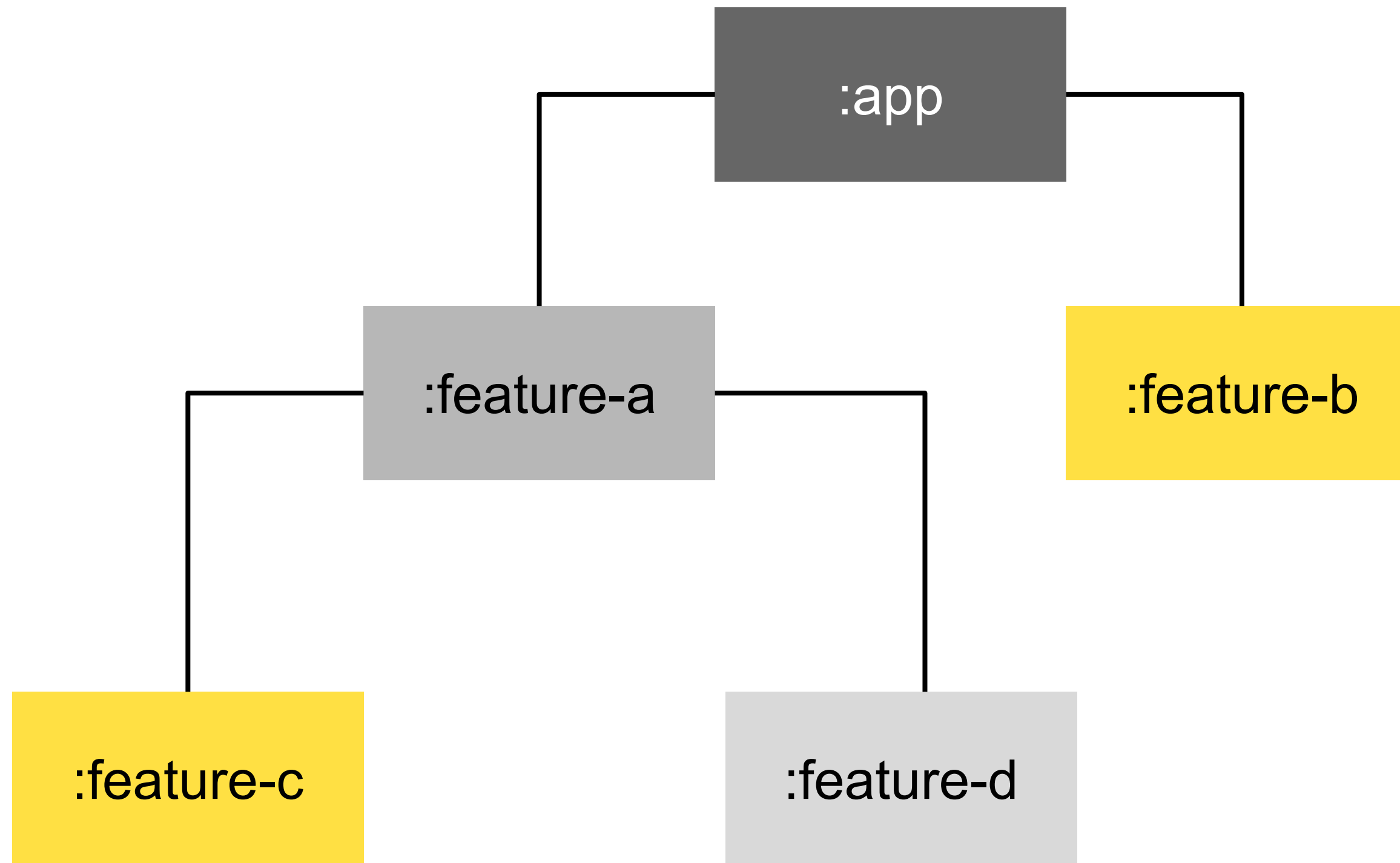
Вычленяем сет
затронутых модулей

Запускаем тесты
только на ЭТИХ
модулях

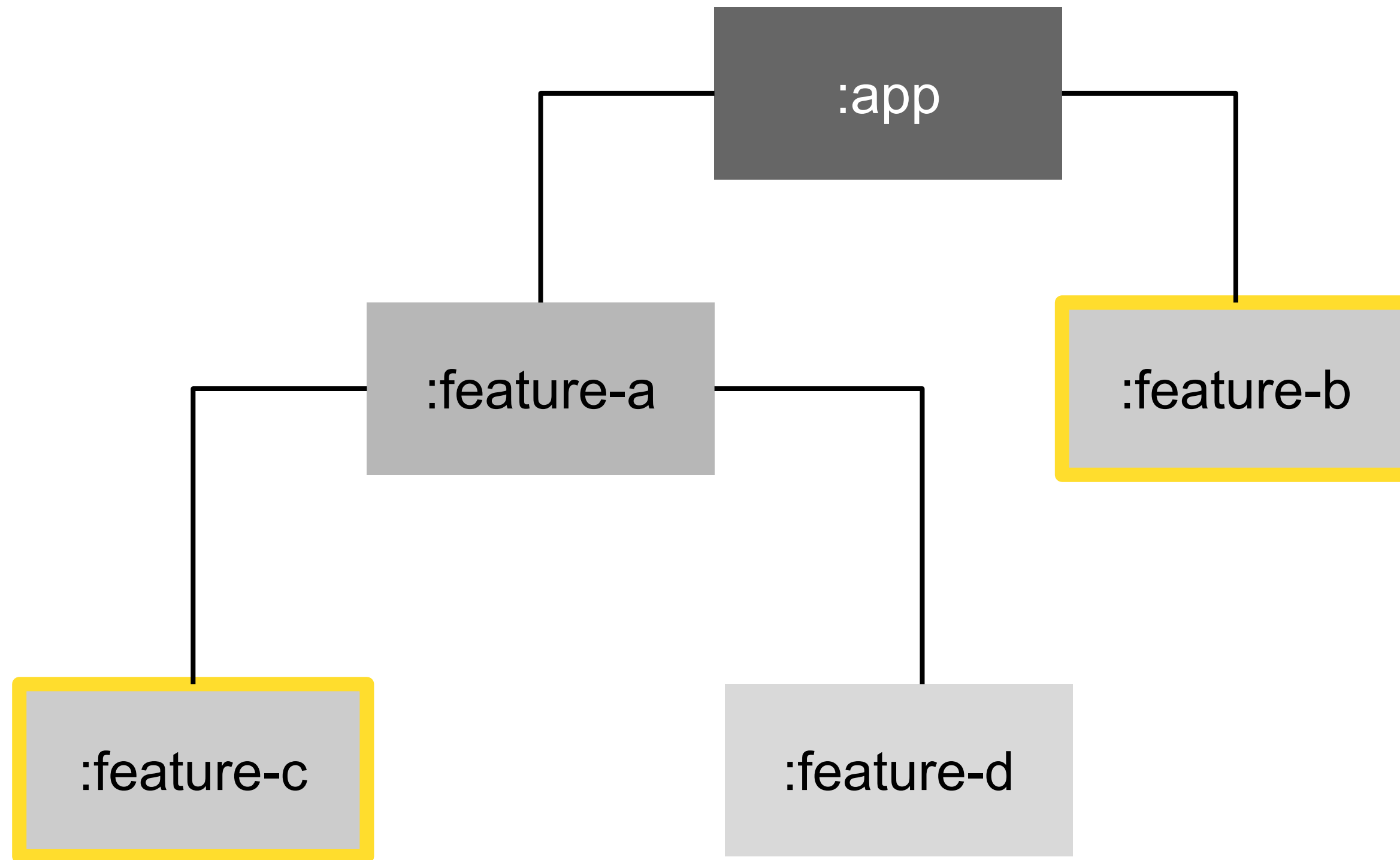
File Detector



File Detector



File Detector



File Detector

```
tasks.register(name: "autoTestChangedModules") { this: Task
    val diff = cmd(cmd: "git show --name-only origin/dev..HEAD")

    val autoTestTasks = diff.asSequence()
        .filter(::isModule)
        .map(::extractAutoTestTask)
        .toSet()

    dependsOn(autoTestTasks)
}
```

File Detector

```
tasks.register(name: "autoTestChangedModules") { this: Task
    val diff = cmd(cmd: "git show --name-only origin/dev..HEAD")

    val autoTestTasks = diff.asSequence()
        .filter(::isModule)
        .map(::extractAutoTestTask)
        .toSet()

    dependsOn(autoTestTasks)
}
```

File Detector

```
tasks.register(name: "autoTestChangedModules") { this: Task
    val diff = cmd(cmd: "git show --name-only origin/dev..HEAD")

    val autoTestTasks = diff.asSequence()
        .filter(::isModule)
        .map(::extractAutoTestTask)
        .toSet()

    dependsOn(autoTestTasks)
}
```

File Detector

```
tasks.register(name: "autoTestChangedModules") { this: Task
    val diff = cmd(cmd: "git show --name-only origin/dev..HEAD")

    val autoTestTasks = diff.asSequence()
        .filter(::isModule)
        .map(::extractAutoTestTask)
        .toSet()

    dependsOn(autoTestTasks)
}
```


File Detector

```
tasks.register(name: "autoTestChangedModules") { this: Task
    val diff = cmd(cmd: "git show --name-only origin/dev..HEAD")

    val autoTestTasks = diff.asSequence()
        .filter(::isModule)
        .map(::extractAutoTestTask)
        .toSet()

    dependsOn(autoTestTasks)
}
```

File Detector

```
tasks.register(name: "autoTestChangedModules") { this: Task
    val diff = cmd(cmd: "git show --name-only origin/dev..HEAD")

    val autoTestTasks = diff.asSequence()
        .filter(::isModule)
        .map(::extractAutoTestTask)
        .toSet()

    dependsOn(autoTestTasks)
}
```

File Detector

Плюсы

-  Простота реализации

File Detector

Плюсы



Простота реализации

Минусы



False positive

File Detector

Плюсы


 Простота реализации

Минусы

 False positive

 False negative

File Detector

 Минусы



TeamCity APP 10:41 AM

Failed - Mobile / Android / Mobile Bank for Business / Testing pipeline / UNIT Tests Pipeline / Run Unit Tests #234

Run Unit Tests

Elapsed: 23m:11s

Reason

Process exited with code 1 (Step: UNIT Tests (Command Line))

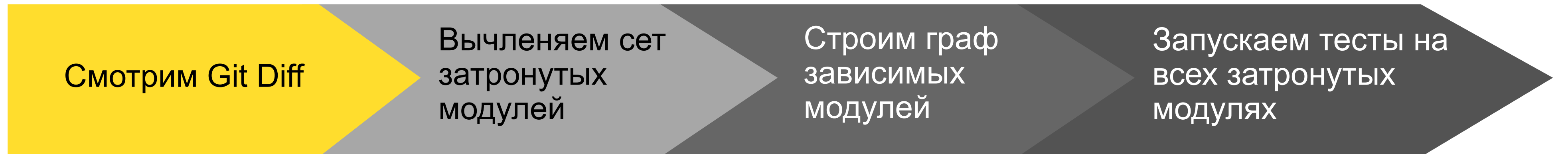
Triggered By

svc_allureserver

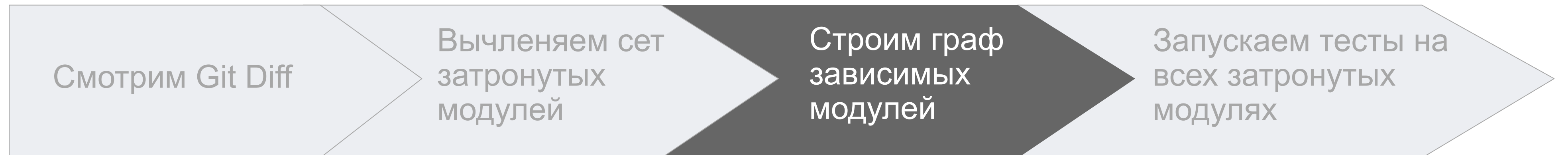


7 replies Last reply 3 months ago

File Detector на максималках



File Detector на максималках



File Detector на максималках



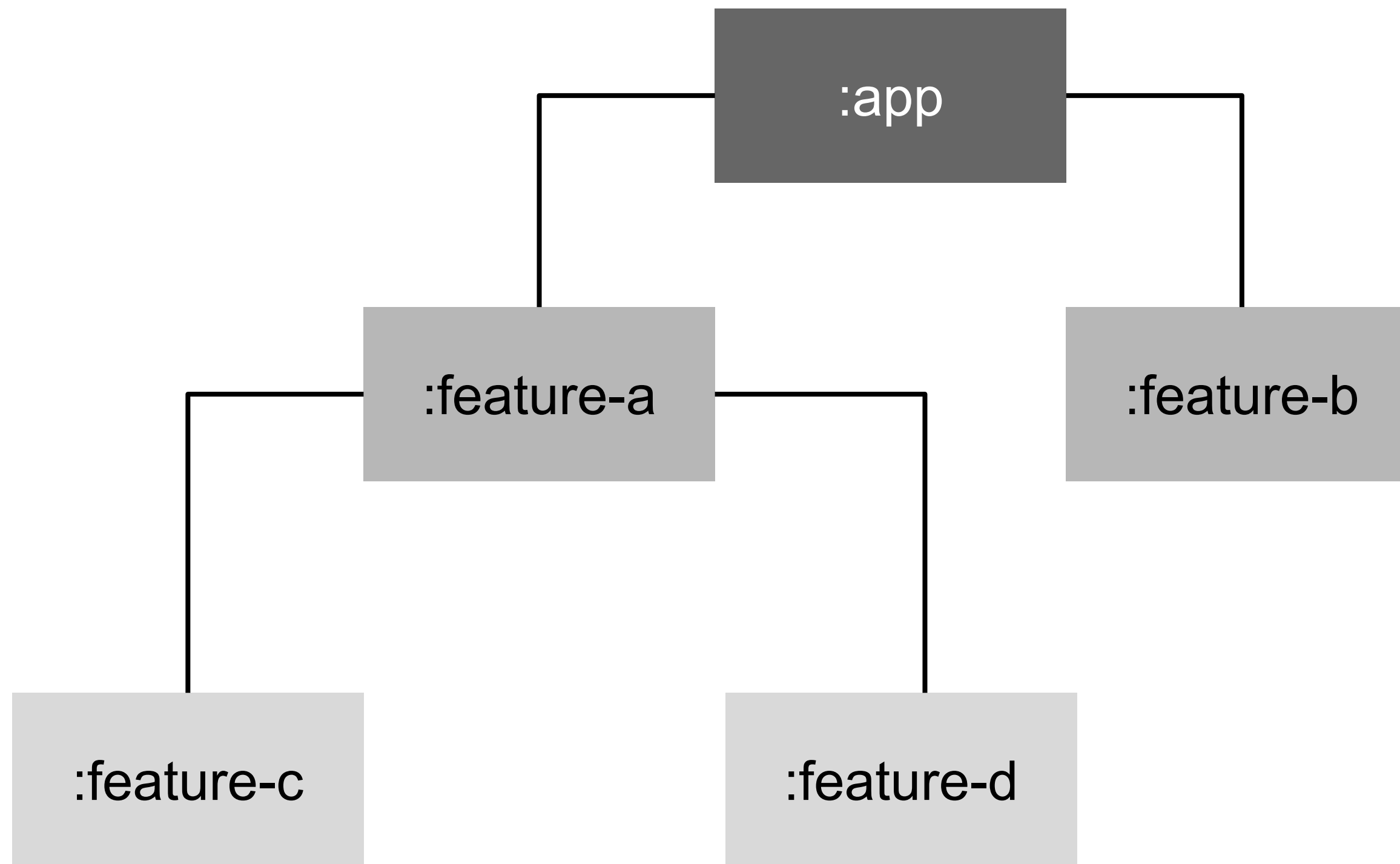
[AffectedModuleDetector](#)



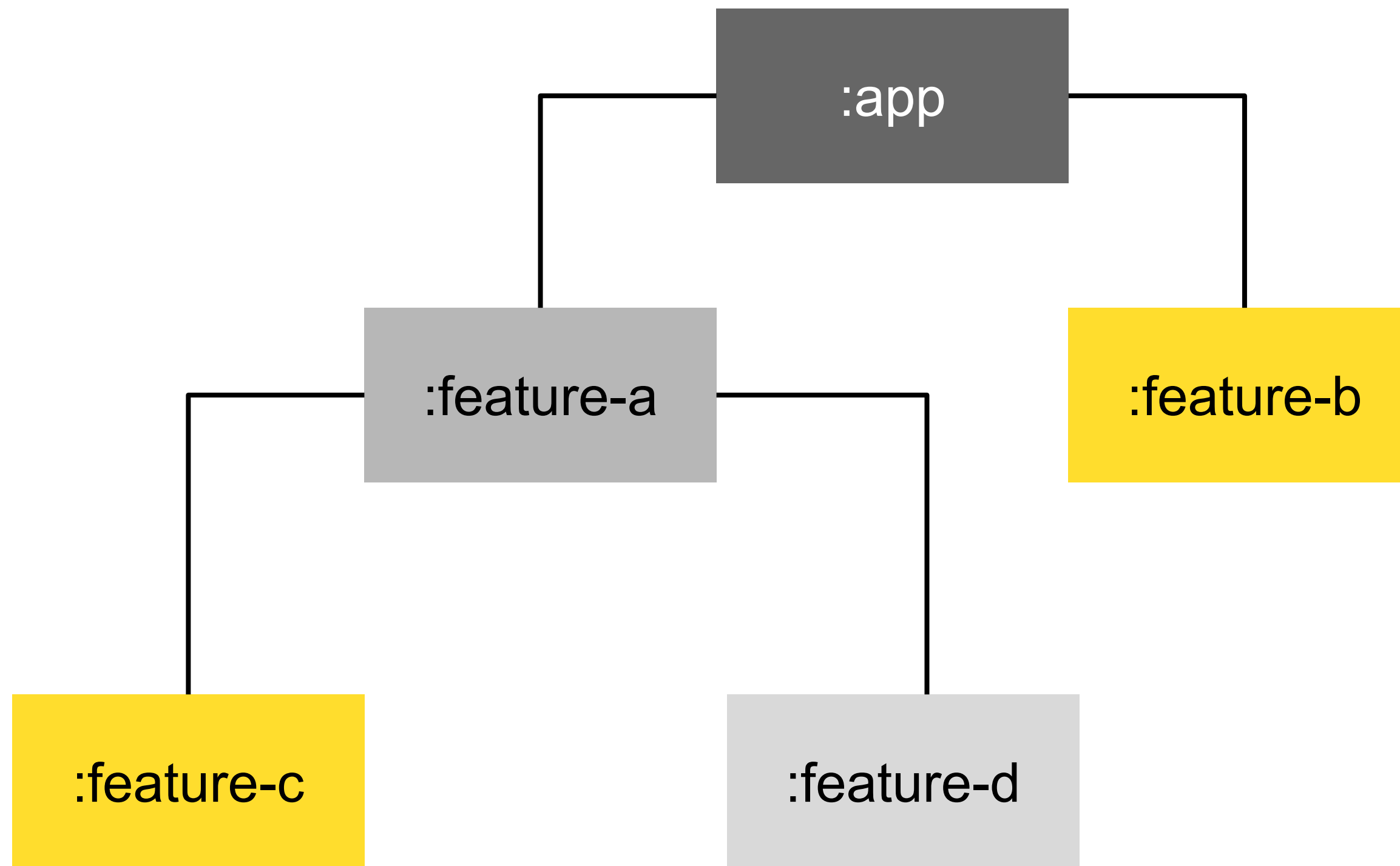
[Avito Android Impact](#)

File Detector на максималках

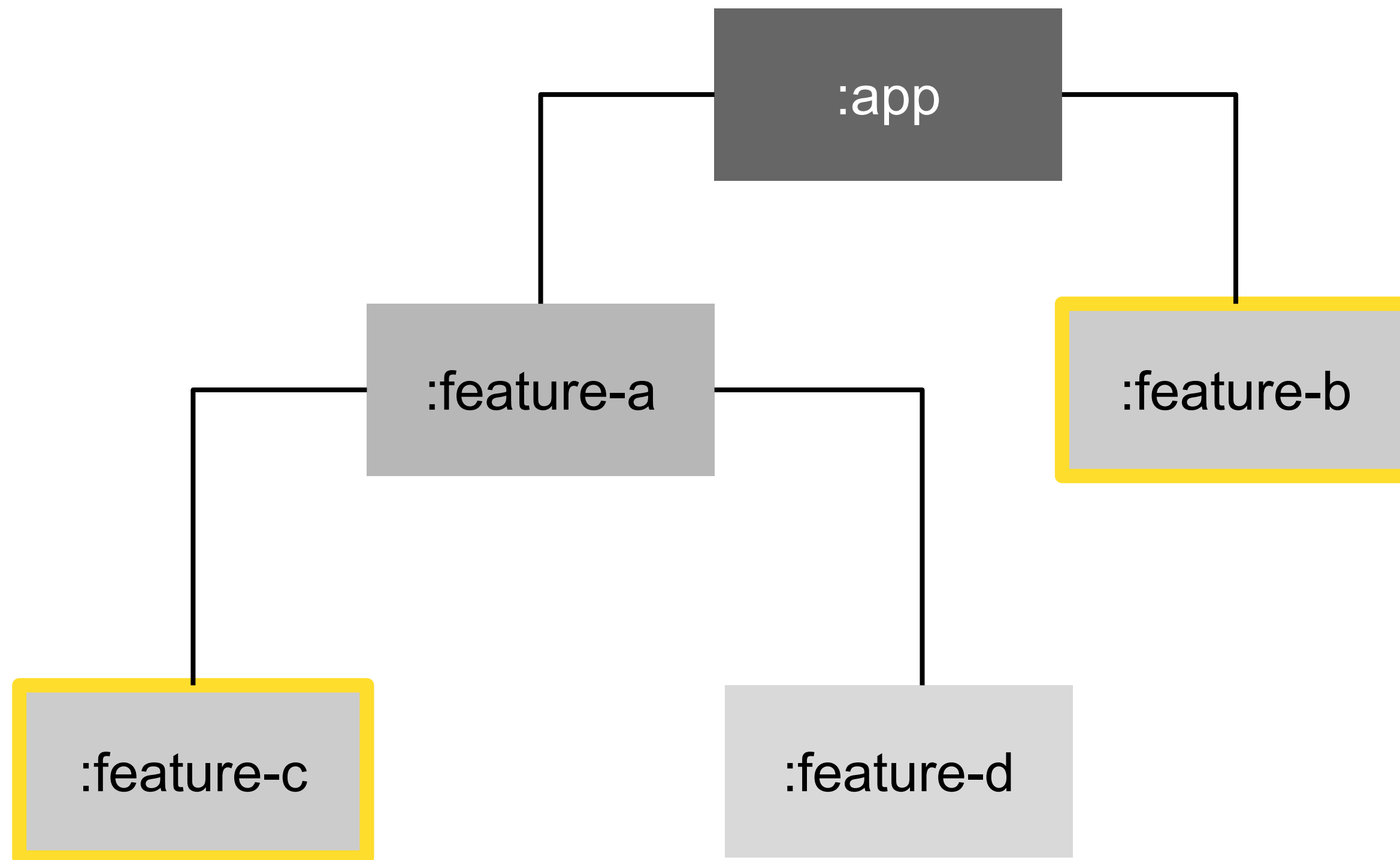
File Detector на максималках



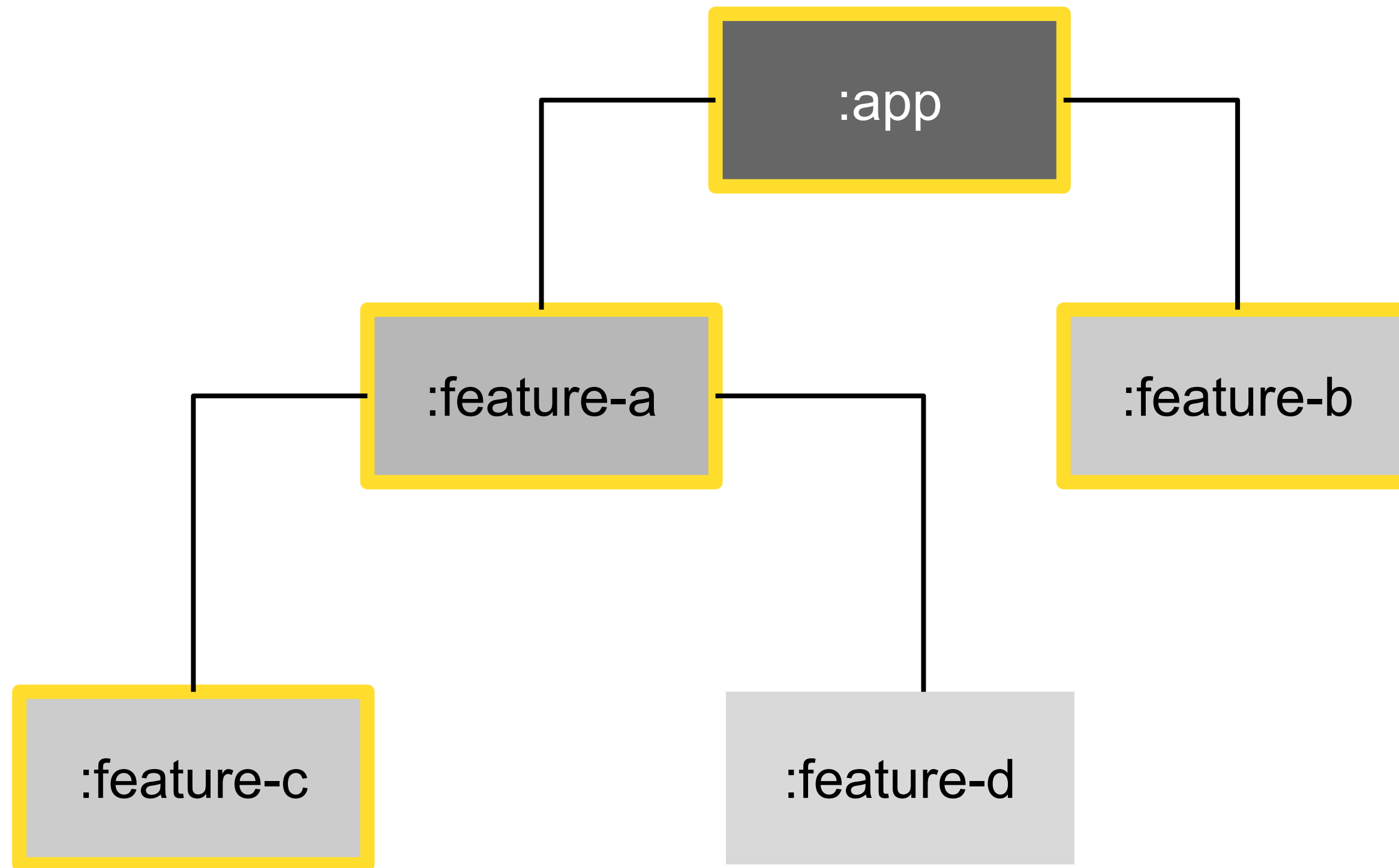
File Detector на максималках



File Detector на максималках



File Detector на максималках



File Detector на максималках


Плюсы



Теперь мы точно запустим все тесты,
ни один не будет пропущен

File Detector на максималках

Плюсы


-  Теперь мы точно запустим все тесты, ни один не будет пропущен

Минусы



-  False positive

File Detector на максималках

Плюсы

-  Теперь мы точно запустим все тесты, ни один не будет пропущен

Минусы

-  False positive
-  Большинство ci-тестов лежат в одном модуле

Другие варианты



Можем связать наши экраны с тестами с помощью аннотаций

```
@Test
@AllureId( value: "412075")
@DisplayName( value: "Заглушка при ошибке метода")
fun notificationsFeedError() = with(rule) { this: SampleAppRule<NotificationsFeedActivity>
    mock { this: MockPromotions
        error()
    }
    startActivity()
    NotificationsFeedPage { this: NotificationsFeedPage
        assertAndCloseDefaultInternetErrorDialog()
        assert { recyclerErrorViewShown() }
    }
}
```

```
@Test
@AllureId( value: "412075")
@DisplayName( value: "Заглушка при ошибке метода")
fun notificationsFeedError() = with(rule) { this: SampleAppRule<NotificationsFeedActivity>
    mock { this: MockPromotions
        error()
    }
    startActivity()
    NotificationsFeedPage { this: NotificationsFeedPage
        assertAndCloseDefaultInternetErrorDialog()
        assert { recyclerErrorViewShown() }
    }
}
```

Связываем PageObject

```
@UiDsl
@LinkedWith(...sources: "NotificationsFeedFragment")
class NotificationsFeedPage {

    companion object {
        const val TAG = "Экран 'Лента уведомлений':"
    }
}
```

Связываем PageObject

```
@UiDsl
@LinkedWith(...sources: "NotificationsFeedFragment")
class NotificationsFeedPage {

    companion object {
        const val TAG = "Экран 'Лента уведомлений':"
```

Другие варианты



Можем связать наши экраны с тестами с помощью аннотаций



Можем прикрутить ML

Другие варианты



Можем связать наши экраны с тестами с помощью аннотаций



Можем прикрутить ML

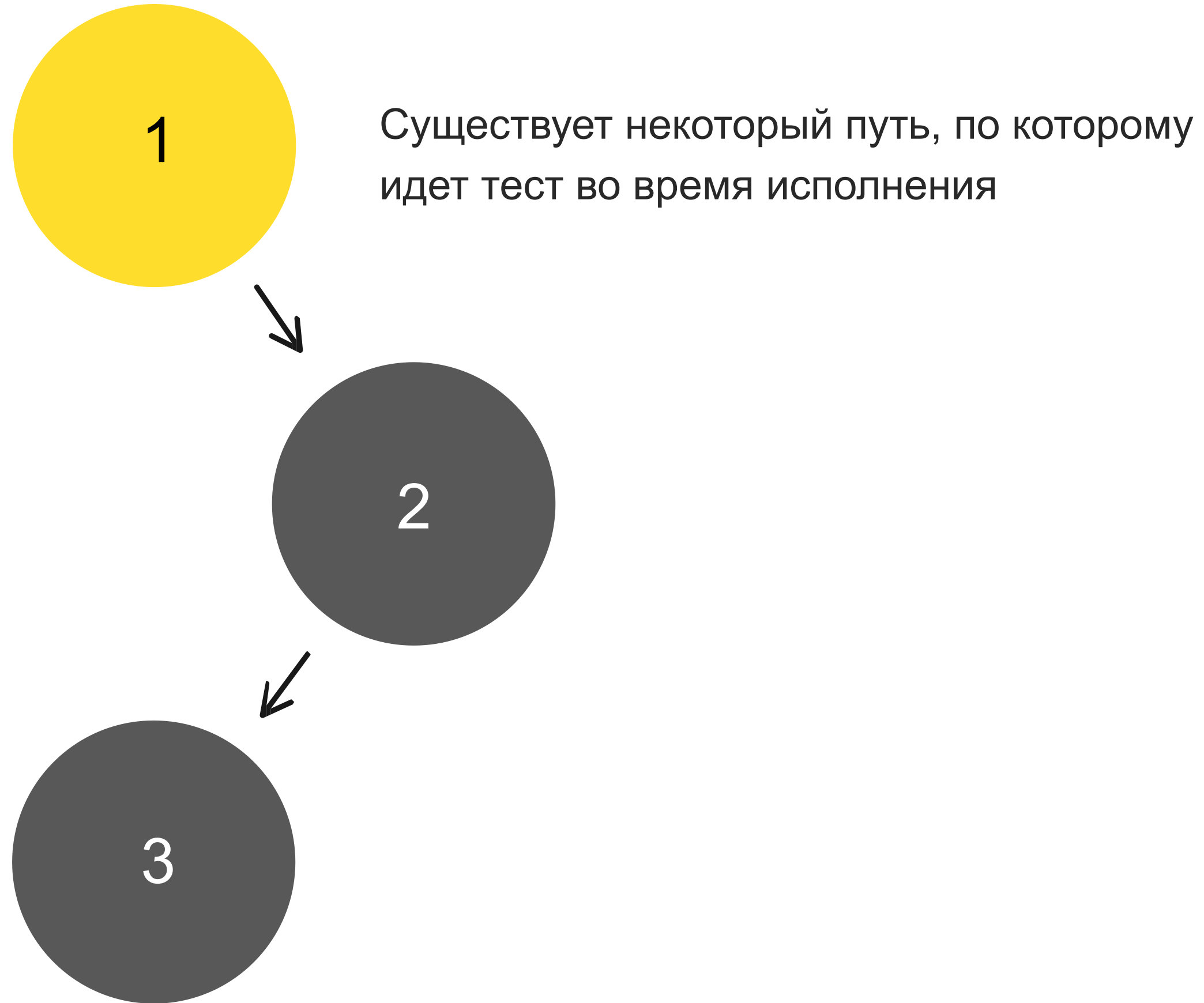


Можем собрать code coverage

Code Coverage Analysis



О чём это?



SomeSampleTest.kt

```
@Test
fun someTest() {
    val testClass = TestClass()

    val testResult = testClass.doReturn2()

    assertEquals(testResult, actual: 2)
}
}
```

SomeSampleTest.kt

```
@Test
fun someTest() {
    val testClass = TestClass()

    val testResult = testClass.doReturn2()

    assertEquals(testResult, actual: 2)
}
}
```

SomeSampleTest.kt

```
@Test
fun someTest() {
    val testClass = TestClass()

    val testResult = testClass.doReturn2()

    assertEquals(testResult, actual: 2)
}
}
```

SomeSampleTest.kt

```
@Test
fun someTest() {
    val testClass = TestClass()

    val testResult = testClass.doReturn2()

    assertEquals(testResult, actual: 2)
}
}
```

TestClass.kt

```
2  class TestClass {  
3  
4      fun doReturn2() = 2  
5  }
```

О чём это?



Code coverage

```
{  
  "SomeSampleTest.kt": [  
    {  
      "TestClass.kt": ["2", "4"]  
    }  
  ]  
}
```

Code coverage

```
{  
  "SomeSampleTest.kt": [  
    {  
      "TestClass.kt": ["2", "4"]  
    }  
  ]  
}
```

← Наш тест

Code coverage

```
{  
  "SomeSampleTest.kt": [  
    {  
      "TestClass.kt": ["2", "4"]  
    }  
  ]  
}
```



Используемый в тесте
класс

Code coverage

```
{  
  "SomeSampleTest.kt": [  
    {  
      "TestClass.kt": ["2", "4"]  
    }  
  ]  
}
```



Покрытые строки в этом классе

Code coverage

```
{  
  "SomeSampleTest.kt": [  
    {  
      "TestClass.kt": ["2", "4"]  
    }  
  ]  
}
```

О чём это?



Как собрать coverage?



Как собрать coverage?



JaCoCo

JaCoCo

Coverage: ...

90% classes, 15% lines covered in package 'org.springframework.samples.pe...

| Element | Class, % | Method, % | Line, % | Branch, % |
|--------------|------------|-----------|-------------|------------|
| model | 100% (3/3) | 0% (0/10) | 23% (3/13) | 100% (0/0) |
| owner | 87% (7/8) | 7% (4/51) | 9% (14/145) | 0% (0/4) |
| system | 100% (3/3) | 60% (3/5) | 75% (6/8) | 100% (0/0) |
| vet | 75% (3/4) | 11% (1/9) | 17% (4/23) | 100% (0/0) |
| visit | 100% (1/1) | 14% (1/7) | 25% (2/8) | 100% (0/0) |
| PetClinic... | 100% (1/1) | 0% (0/1) | 50% (1/2) | 100% (0/0) |

JaCoCo

Плюсы

-  Сбор покрытия из коробки

JaCoCo

Плюсы

 Сбор покрытия из коробки

Минусы

 Не умеет собирать coverage для каждого теста по отдельности

JaCoCo

Плюсы

 Сбор покрытия из коробки

Минусы

 Не умеет собирать coverage для каждого теста по отдельности

 Умеет работать только с JVM

Как выглядит класс изнутри

```
public class HelloWorld {  
  
    public static void main(String... args) {  
        System.out.println("Hello, world!");  
    }  
}
```

Как выглядит класс изнутри



```
javac HelloWorld.java
```

Как выглядит класс изнутри



```
javac HelloWorld.java
```



```
hexdump -C HelloWorld.class > HelloWorld.hexdump
```


HelloWorld.hexdump

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 CA FE BA BE 00 00 00 37 00 1D 0A 00 06 00 0F 09 Êp%...7.....
00000010 00 10 00 11 08 00 12 0A 00 13 00 14 07 00 15 07 .....
00000020 00 16 01 00 06 3C 69 6E 69 74 3E 01 00 03 28 29 .....<init>...()
00000030 56 01 00 04 43 6F 64 65 01 00 0F 4C 69 6E 65 4E V...Code...LineN
00000040 75 6D 62 65 72 54 61 62 6C 65 01 00 04 6D 61 69 umberTable...mai
00000050 6E 01 00 16 28 5B 4C 6A 61 76 61 2F 6C 61 6E 67 n...([Ljava/lang
00000060 2F 53 74 72 69 6E 67 3B 29 56 01 00 0A 53 6F 75 /String;)V...Sou
00000070 72 63 65 46 69 6C 65 01 00 0F 48 65 6C 6C 6F 57 rceFile...HelloW
00000080 6F 72 6C 64 2E 6A 61 76 61 0C 00 07 00 08 07 00 orld.java.....
00000090 17 0C 00 18 00 19 01 00 0D 48 65 6C 6C 6F 2C 20 .....Hello,
000000A0 77 6F 72 6C 64 21 07 00 1A 0C 00 1B 00 1C 01 00 world!.....
000000B0 0A 48 65 6C 6C 6F 57 6F 72 6C 64 01 00 10 6A 61 .HelloWorld...ja
000000C0 76 61 2F 6C 61 6E 67 2F 4F 62 6A 65 63 74 01 00 va/lang/Object..
000000D0 10 6A 61 76 61 2F 6C 61 6E 67 2F 53 79 73 74 65 .java/lang/Syste
000000E0 6D 01 00 03 6F 75 74 01 00 15 4C 6A 61 76 61 2F m...out...Ljava/
000000F0 69 6F 2F 50 72 69 6E 74 53 74 72 65 61 6D 3B 01 io/PrintStream;.
00000100 00 13 6A 61 76 61 2F 69 6F 2F 50 72 69 6E 74 53 ..java/io/PrintS
00000110 74 72 65 61 6D 01 00 07 70 72 69 6E 74 6C 6E 01 tream...println.
00000120 00 15 28 4C 6A 61 76 61 2F 6C 61 6E 67 2F 53 74 ..(Ljava/lang/St
00000130 72 69 6E 67 3B 29 56 00 21 00 05 00 06 00 00 00 ring;)V.!.....
00000140 00 00 02 00 01 00 07 00 08 00 01 00 09 00 00 00 .....
00000150 1D 00 01 00 01 00 00 00 05 2A B7 00 01 B1 00 00 .....*...±..
00000160 00 01 00 0A 00 00 00 06 00 01 00 00 00 02 00 89 .....
00000170 00 0B 00 0C 00 01 00 09 00 00 00 25 00 02 00 01 .....%.
00000180 00 00 00 09 B2 00 02 12 03 B6 00 04 B1 00 00 00 ....²....¶...±...
00000190 01 00 0A 00 00 00 0A 00 02 00 00 05 00 08 00 .....
000001A0 06 00 01 00 0D 00 00 00 02 00 0E .....
```


Как можно модифицировать байт-код

Как можно модифицировать байт-код



Как можно модифицировать байт-код



ASM



AspectJ

Как можно модифицировать байт-код



ASM



AspectJ



ByteBuddy

Как можно модифицировать байт-код



ASM



AspectJ



ByteBuddy



Java Agents

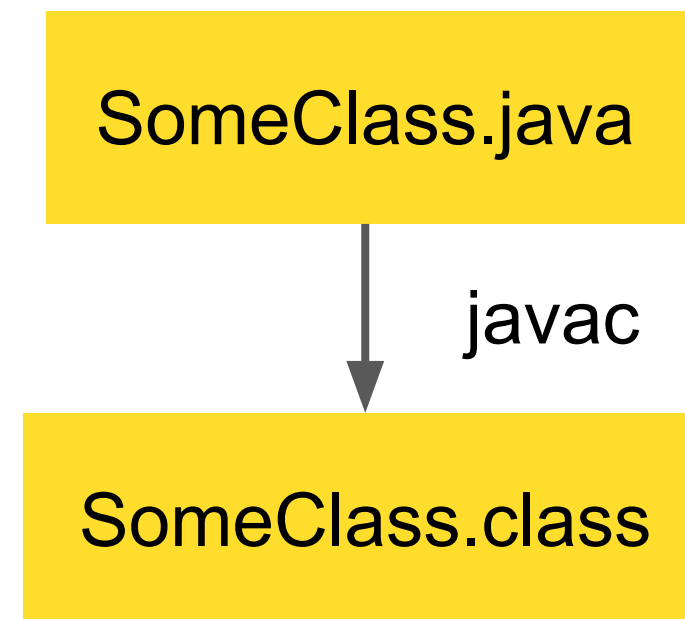
Java Agents



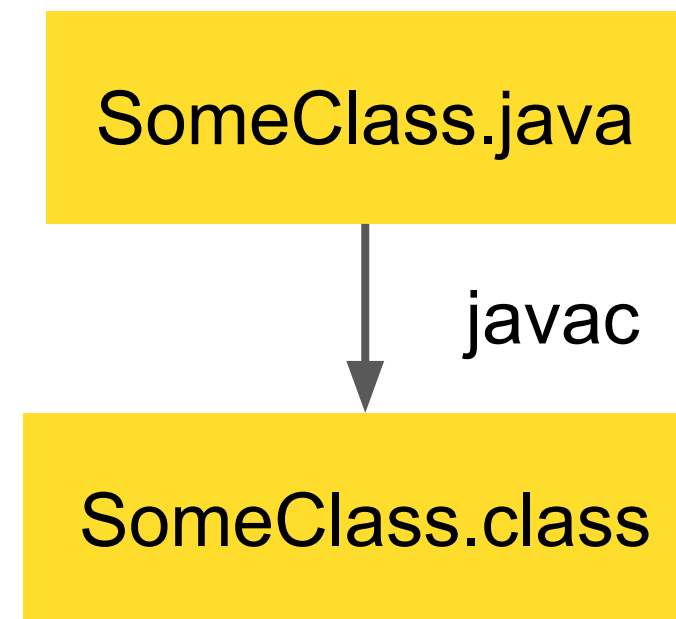
Java Agents

SomeClass.java

Java Agents

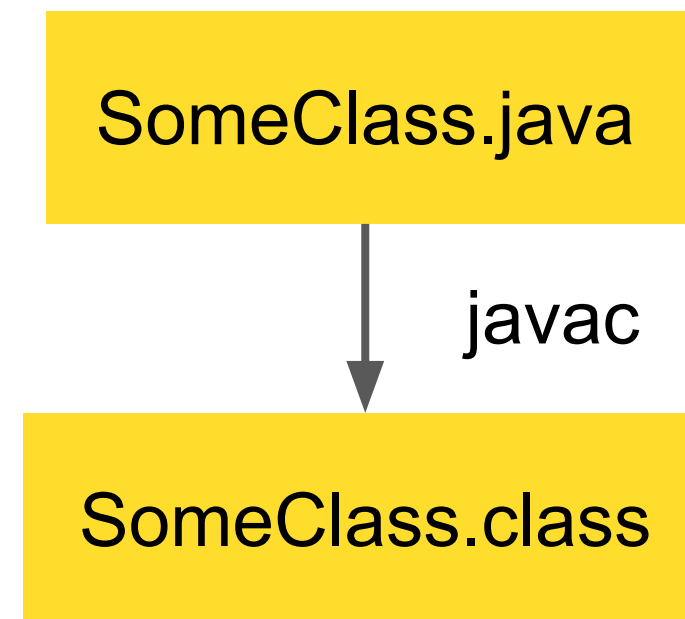


Java Agents



JVM Process

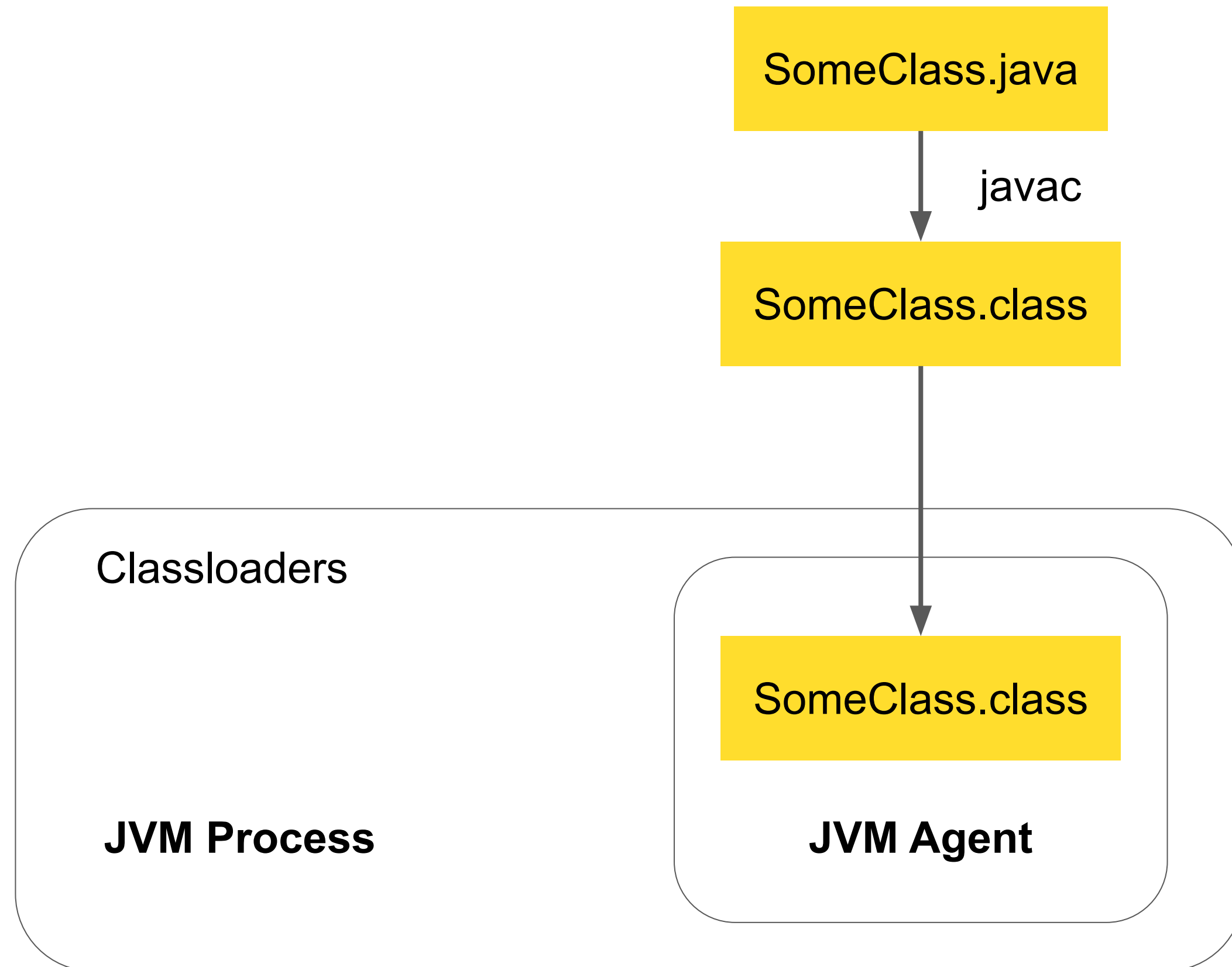
Java Agents



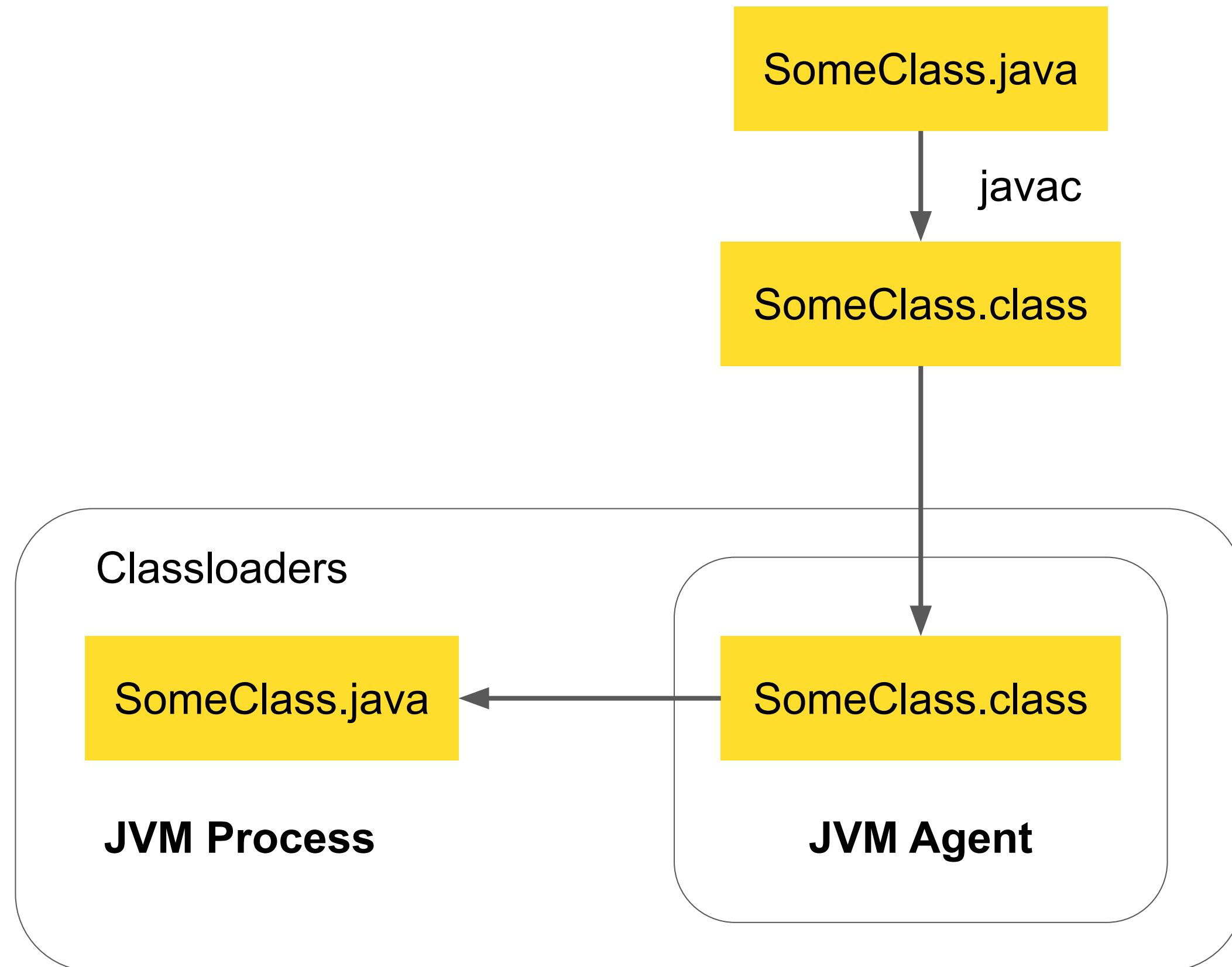
Classloaders

JVM Process

Java Agents



Java Agents



Java Agents



[Rafael Winterhalter – The definite guide to Java agents](#)

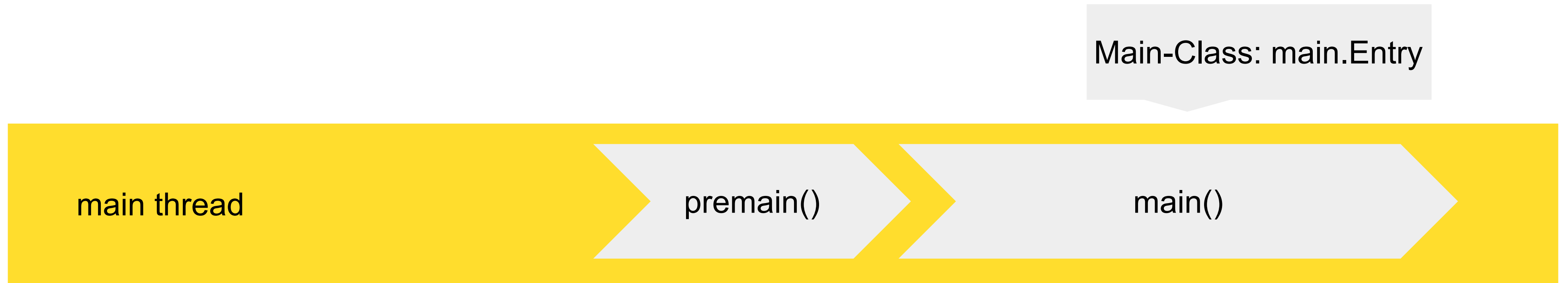
Java Agents



Java Agents



Java Agents



Java Agents

Premain-Class: agent.PreEntry

Main-Class: main.Entry

main thread

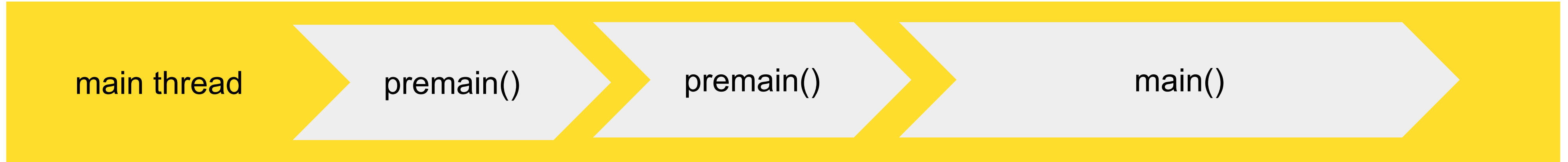
premain()

main()

Java Agents

Premain-Class: agent.PreEntry

Main-Class: main.Entry



Java Agents

Premain-Class: agent.PreEntry

Main-Class: main.Entry

main thread

premain()

premain()

main()

attach thread

agentmain()

Java Agents

Premain-Class: agent.PreEntry

Main-Class: main.Entry

main thread

premain()

premain()

main()

attach thread

agentmain()

Agent-Class: agent.AgentEntry

Java Agents

Premain-Class: agent.PreEntry

Main-Class: main.Entry

main thread

premain()

premain()

main()

attach thread

agentmain()

agentmain()

Agent-Class: agent.AgentEntry

Java Agents



.jar, который может быть запущен
либо статически, либо
динамически

Java Agents



.jar, который может быть запущен
либо статически, либо
динамически



С помощью Instrumentation API может
выполнять преобразования в байт-
коде

Java Agents



.jar, который может быть запущен
либо статически, либо
динамически



С помощью Instrumentation API может
выполнять преобразования в байт-
коде



Можно использовать для
логгеров, профайлеров или для
сбора code coverage

JaCoCo



ТИНЬКОФФ

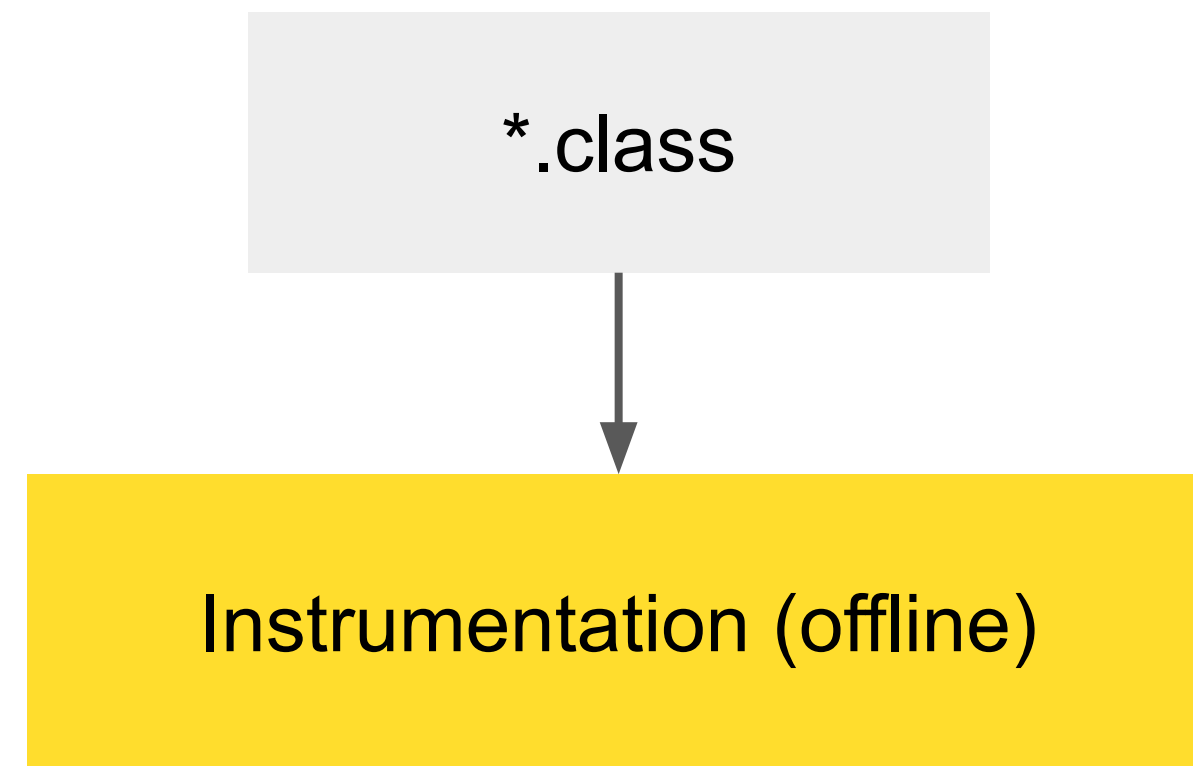
tinkoff.ru

JaCoCo Instrumentation

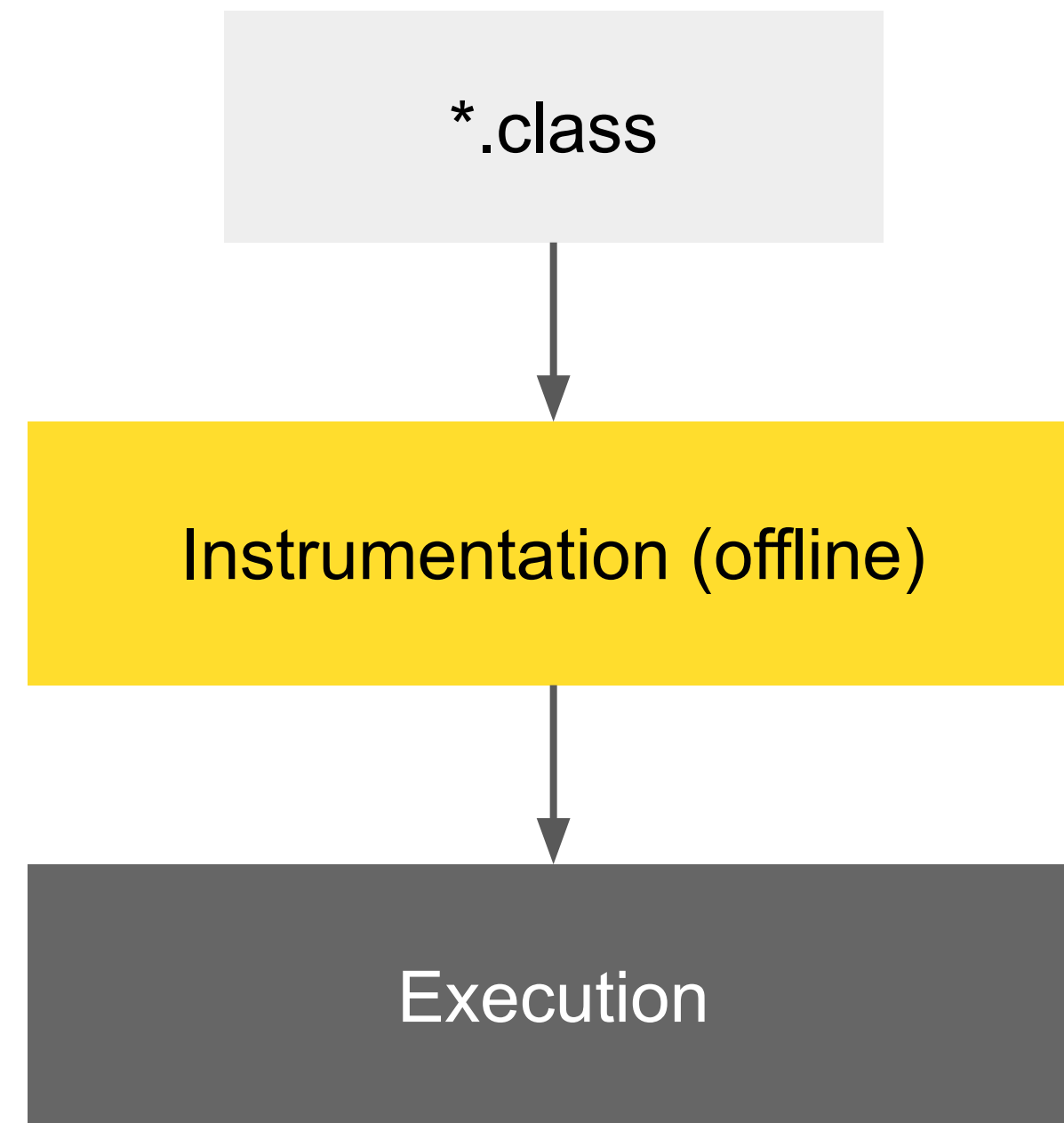
JaCoCo Instrumentation

*.class

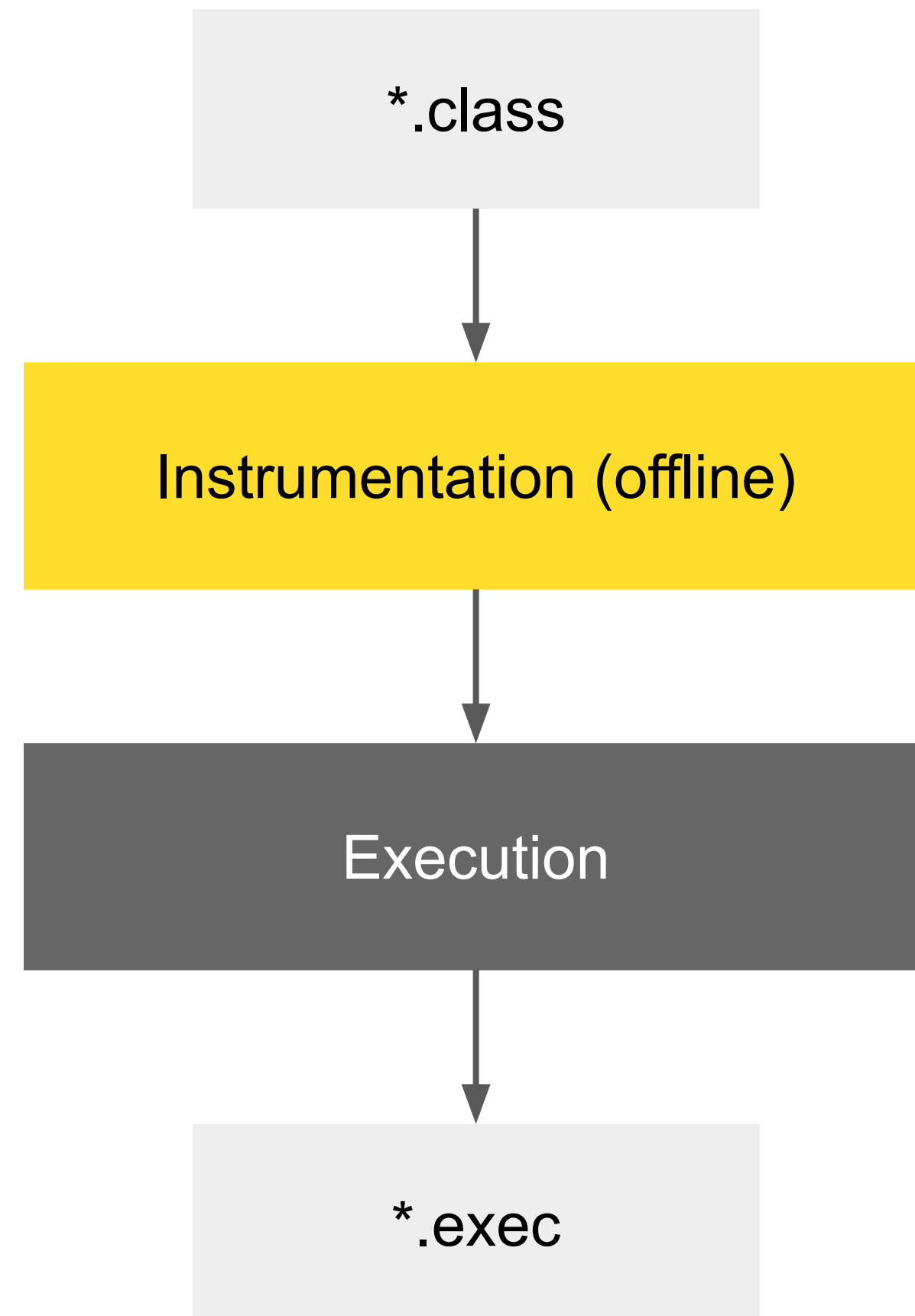
JaCoCo Instrumentation



JaCoCo Instrumentation



JaCoCo Instrumentation



JaCoCo

Для JVM



Использует Java Agents

JaCoCo

Для JVM

 Использует Java Agents

Для Android

 Использует offline-инструментацию

Пишем свой ТИА



ТИНЬКОФФ

tinkoff.ru

Основная идея

Основная идея



Учимся собирать coverage для
каждого отдельно запускаемого
теста

Основная идея



Учимся собирать coverage для каждого отдельно запускаемого теста



Анализируем весь собранный coverage, составляем маппинг-карту (тест → затронутый код)

Основная идея



Учимся собирать coverage для каждого отдельно запускаемого теста



Анализируем весь собранный coverage, составляем маппинг-карту (тест → затронутый код)



Прикручиваем Git

Включаем Coverage

```
android { this: BaseAppModuleExtension
    buildTypes { this: NamedDomainObjectContainer<BuildType>
        getByName(name: "debug") { this: BuildType
            isTestCoverageEnabled = true
            matchingFallbacks += listOf("release")
        }
    }
}
```

Включаем Coverage

```
android { this: BaseAppModuleExtension
    buildTypes { this: NamedDomainObjectContainer<BuildType>
        getByName( name: "debug" ) { this: BuildType
            isTestCoverageEnabled = true
            matchingFallbacks += listOf("release")
        }
    }
}
```

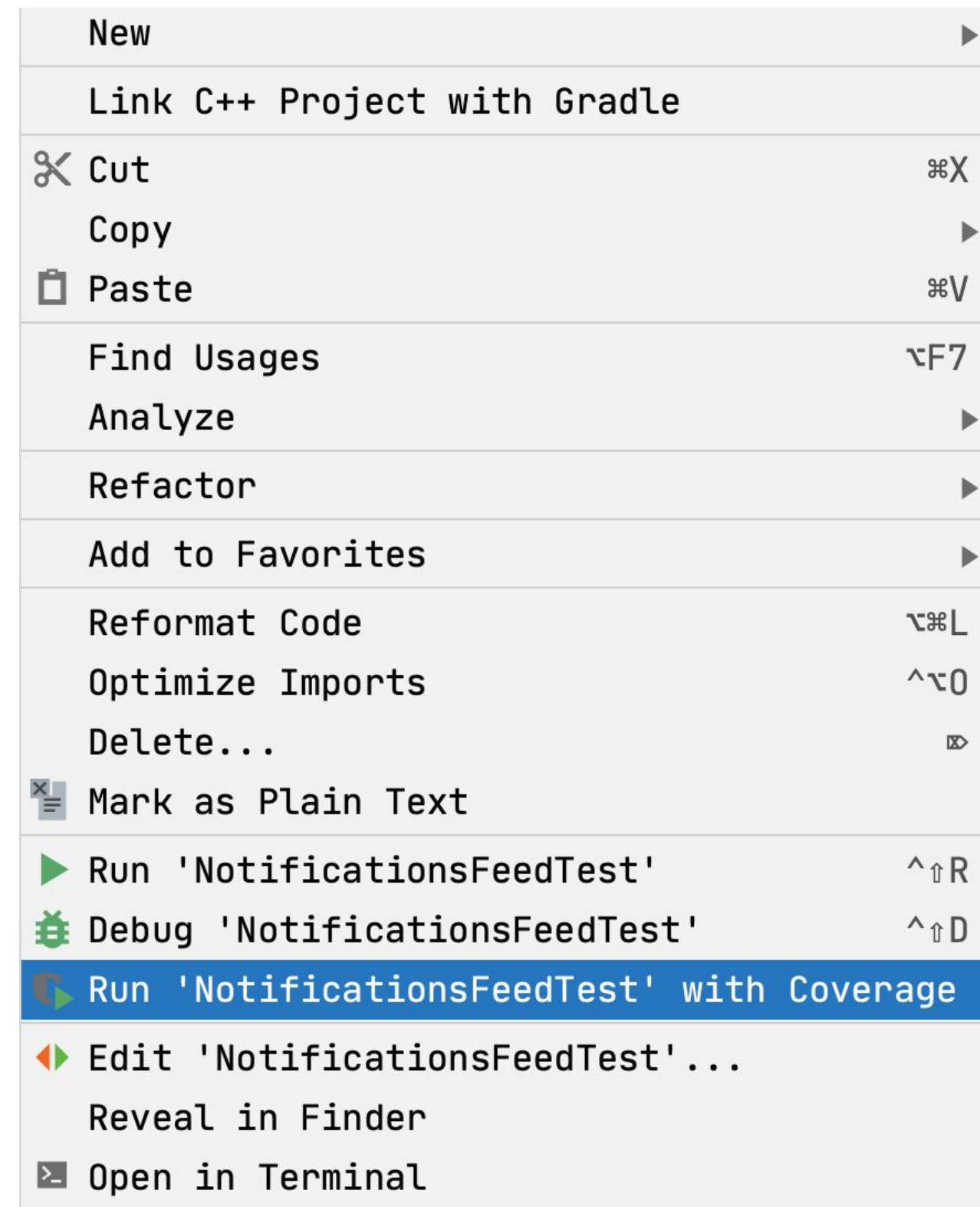
Включаем Coverage

```
android { this: BaseAppModuleExtension
  defaultConfig { this: DefaultConfig
    testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    testInstrumentationRunnerArguments(mapOf("coverage" to "true"))
  }
}
```


Включаем Coverage

```
android { this: BaseAppModuleExtension
  defaultConfig { this: DefaultConfig
    testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    testInstrumentationRunnerArguments(mapOf("coverage" to "true"))
  }
}
```

Run with Coverage



Run with Coverage

```
D/TestExecutor: Adding listener androidx.test.internal.runner.listener.LogRunListener
D/TestExecutor: Adding listener androidx.test.internal.runner.listener.InstrumentationResultPrinter
D/TestExecutor: Adding listener androidx.test.internal.runner.listener.ActivityFinisherRunListener
D/TestExecutor: Adding listener androidx.test.internal.runner.listener.CoverageListener
I/TestRunner: run started: 2 tests
I/TestRunner: started: test(ru.tinkoff.moba.sample.android.SomeSampleAndroidTest)
```

Run with Coverage

```
D/TestExecutor: Adding listener androidx.test.internal.runner.listener.LogRunListener
D/TestExecutor: Adding listener androidx.test.internal.runner.listener.InstrumentationResultPrinter
D/TestExecutor: Adding listener androidx.test.internal.runner.listener.ActivityFinisherRunListener
D/TestExecutor: Adding listener androidx.test.internal.runner.listener.CoverageListener
I/TestRunner: run started: 2 tests
I/TestRunner: started: test(ru.tinkoff.moba.sample.android.SomeSampleAndroidTest)
```


androidx.test.internal.runner.CoverageListener

```
public class CoverageListener extends InstrumentationRunListener {

    private static final File coverageFile = new java.io.File("coverageFilePath");

    private static final String EMMA_RUNTIME_CLASS = "com.vladim.emma.rt.RT";

    private void generateCoverageReport(PrintStream writer, Bundle results) throws ClassNotFoundException

        ClassLoader classLoader = getInstrumentation().getTargetContext().getClassLoader();

        Class<?> emmaRTClass = Class.forName(EMMA_RUNTIME_CLASS, initialize: true, classLoader);

        Method dumpCoverageMethod =
            emmaRTClass.getMethod(
                name: "dumpCoverageData", coverageFile.getClass(), boolean.class, boolean.class);

        dumpCoverageMethod.invoke(obj: null, ...args: coverageFile, false, false);
    }
}
```

androidx.test.internal.runner.CoverageListener

```
public class CoverageListener extends InstrumentationRunListener {

    private static final File coverageFile = new java.io.File(pathname: "coverageFilePath");

    private static final String EMMA_RUNTIME_CLASS = "com.vladim.emma.rt.RT";

    private void generateCoverageReport(PrintStream writer, Bundle results) throws ClassNotFoundException

        ClassLoader classLoader = getInstrumentation().getTargetContext().getClassLoader();

        Class<?> emmaRTClass = Class.forName(EMMA_RUNTIME_CLASS, initialize: true, classLoader);

        Method dumpCoverageMethod =
            emmaRTClass.getMethod(
                name: "dumpCoverageData", coverageFile.getClass(), boolean.class, boolean.class);

        dumpCoverageMethod.invoke(obj: null, ...args: coverageFile, false, false);
    }
}
```

androidx.test.internal.runner.CoverageListener

```
public class CoverageListener extends InstrumentationRunListener {

    private static final File coverageFile = new java.io.File("coverageFilePath");

    private static final String EMMA_RUNTIME_CLASS = "com.vladium.emma.rt.RT";

    private void generateCoverageReport(PrintStream writer, Bundle results) throws ClassNotFoundException {

        ClassLoader classLoader = getInstrumentation().getTargetContext().getClassLoader();

        Class<?> emmaRTClass = Class.forName(EMMA_RUNTIME_CLASS, initialize: true, classLoader);

        Method dumpCoverageMethod =
            emmaRTClass.getMethod(
                name: "dumpCoverageData", coverageFile.getClass(), boolean.class, boolean.class);

        dumpCoverageMethod.invoke(obj: null, ...args: coverageFile, false, false);
    }
}
```


androidx.test.internal.runner.CoverageListener

```
public class CoverageListener extends InstrumentationRunListener {

    private static final File coverageFile = new java.io.File(pathname: "coverageFilePath");

    private static final String EMMA_RUNTIME_CLASS = "com.vladim.emma.rt.RT";

    private void generateCoverageReport(PrintStream writer, Bundle results) throws ClassNotFoundException

        ClassLoader classLoader = getInstrumentation().getTargetContext().getClassLoader();

        Class<?> emmaRTClass = Class.forName(EMMA_RUNTIME_CLASS, initialize: true, classLoader);

        Method dumpCoverageMethod =
            emmaRTClass.getMethod(
                name: "dumpCoverageData", coverageFile.getClass(), boolean.class, boolean.class);

        dumpCoverageMethod.invoke(obj: null, ...args: coverageFile, false, false);
    }
}
```


androidx.test.internal.runner.CoverageListener

```
public class CoverageListener extends InstrumentationRunListener {

    private static final File coverageFile = new java.io.File("coverageFilePath");

    private static final String EMMA_RUNTIME_CLASS = "com.vladium.emma.rt.RT";

    private void generateCoverageReport(PrintStream writer, Bundle results) throws ClassNotFoundException {

        ClassLoader classLoader = getInstrumentation().getTargetContext().getClassLoader();

        Class<?> emmaRTClass = Class.forName(EMMA_RUNTIME_CLASS, initialize: true, classLoader);

        Method dumpCoverageMethod =
            emmaRTClass.getMethod(
                name: "dumpCoverageData", coverageFile.getClass(), boolean.class, boolean.class);

        dumpCoverageMethod.invoke(obj: null, ...args: coverageFile, false, false);
    }
}
```

```
class CollectCoverageRunListener : InstrumentationRunListener() {  
  
    override fun testFinished(description: Description) {  
        val testClass = description.testClass  
        val testMethod = description.methodName  
        val coverageFile = File("$testClass#$testMethod.exec")  
        coverageFile.dumpCoverage()  
    }  
  
    private fun File.dumpCoverage() {  
        val emmaRT = EmmaRT.proxy(instrumentation)  
        emmaRT.dumpCoverageData(output: this, merge = false, stopDataCollection = true)  
    }  
}
```

```
class CollectCoverageRunListener : InstrumentationRunListener() {  
  
    override fun testFinished(description: Description) {  
        val testClass = description.testClass  
        val testMethod = description.methodName  
        val coverageFile = File("$testClass#$testMethod.exec")  
        coverageFile.dumpCoverage()  
    }  
  
    private fun File.dumpCoverage() {  
        val emmaRT = EmmaRT.proxy(instrumentation)  
        emmaRT.dumpCoverageData( output: this, merge = false, stopDataCollection = true)  
    }  
}
```



```
class CollectCoverageRunListener : InstrumentationRunListener() {  
  
    override fun testFinished(description: Description) {  
        val testClass = description.testClass  
        val testMethod = description.methodName  
        val coverageFile = File("$testClass#$testMethod.exec")  
        coverageFile.dumpCoverage()  
    }  
  
    private fun File.dumpCoverage() {  
        val emmaRT = EmmaRT.proxy(instrumentation)  
        emmaRT.dumpCoverageData( output: this, merge = false, stopDataCollection = true)  
    }  
}
```

```
class CollectCoverageRunListener : InstrumentationRunListener() {  
  
    override fun testFinished(description: Description) {  
        val testClass = description.testClass  
        val testMethod = description.methodName  
        val coverageFile = File("$testClass#$testMethod.exec")  
        coverageFile.dumpCoverage()  
    }  
  
    private fun File.dumpCoverage() {  
        val emmaRT = EmmaRT.proxy(instrumentation)  
        emmaRT.dumpCoverageData( output: this, merge = false, stopDataCollection = true)  
    }  
}
```

```
class CollectCoverageRunListener : InstrumentationRunListener() {  
  
    override fun testFinished(description: Description) {  
        val testClass = description.testClass  
        val testMethod = description.methodName  
        val coverageFile = File("$testClass#$testMethod.exec")  
        coverageFile.dumpCoverage()  
    }  
  
    private fun File.dumpCoverage() {  
        val emmaRT = EmmaRT.proxy(instrumentation)  
        emmaRT.dumpCoverageData( output: this, merge = false, stopDataCollection = true)  
    }  
}
```

```
class CollectCoverageRunListener : InstrumentationRunListener() {  
  
    override fun testFinished(description: Description) {  
        val testClass = description.testClass  
        val testMethod = description.methodName  
        val coverageFile = File("$testClass#$testMethod.exec")  
        coverageFile.dumpCoverage()  
    }  
  
    private fun File.dumpCoverage() {  
        val emmaRT = EmmaRT.proxy(instrumentation)  
        emmaRT.dumpCoverageData( output: this, merge = false, stopDataCollection = true)  
    }  
}
```



```
class CollectCoverageRunListener : InstrumentationRunListener() {  
  
    override fun testFinished(description: Description) {  
        val testClass = description.testClass  
        val testMethod = description.methodName  
        val coverageFile = File("$testClass#$testMethod.exec")  
        coverageFile.dumpCoverage()  
    }  
  
    private fun File.dumpCoverage() {  
        val emmaRT = EmmaRT.proxy(instrumentation)  
        emmaRT.dumpCoverageData(output: this, merge = false, stopDataCollection = true)  
    }  
}
```



```

class TiaRunner : AndroidJUnitRunner() {

    override fun onCreate(arguments: Bundle) {
        super.onCreate(arguments.bindListener<CollectCoverageRunListener>())
    }
}

inline fun <reified T : InstrumentationRunListener> Bundle.bindListener() = apply {
    val listenerArg = listOfNotNull(
        getCharSequence("listener"),
        T::class.java.name
    ).joinToString(separator = ",")
    putCharSequence("listener", listenerArg) ^apply
}

```

```

class TiaRunner : AndroidJUnitRunner() {

    override fun onCreate(arguments: Bundle) {
        super.onCreate(arguments.bindListener<CollectCoverageRunListener>())
    }
}

inline fun <reified T : InstrumentationRunListener> Bundle.bindListener() = apply {
    val listenerArg = listOfNotNull(
        getCharSequence("listener"),
        T::class.java.name
    ).joinToString(separator = ",")
    putCharSequence("listener", listenerArg) ^apply
}

```

```

class TiaRunner : AndroidJUnitRunner() {

    override fun onCreate(arguments: Bundle) {
        super.onCreate(arguments.bindListener<CollectCoverageRunListener>())
    }
}

inline fun <reified T : InstrumentationRunListener> Bundle.bindListener() = apply {
    val listenerArg = listOfNotNull(
        getCharSequence("listener"),
        T::class.java.name
    ).joinToString(separator = ",")
    putCharSequence("listener", listenerArg) ^apply
}

```

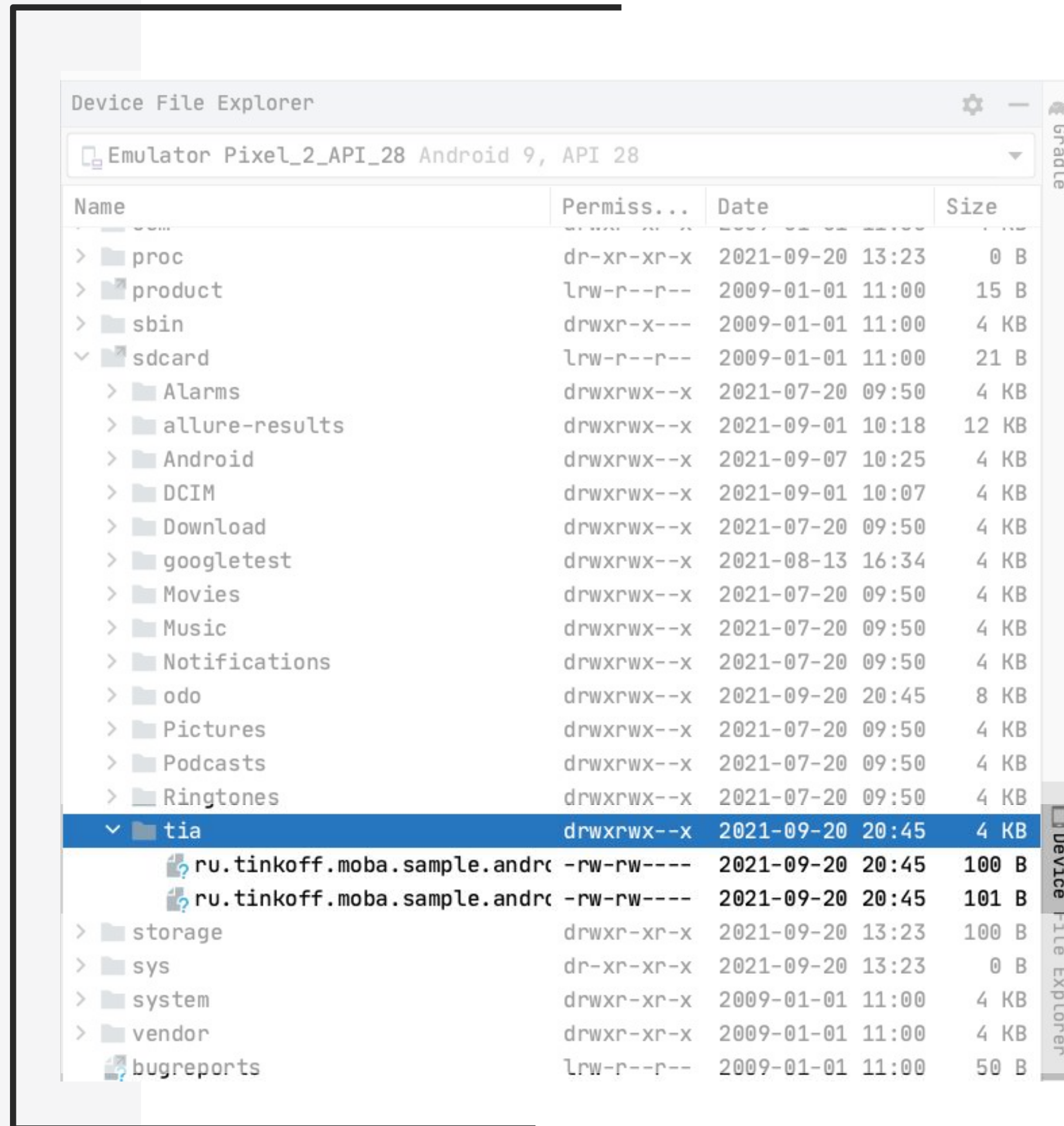

Device File Explorer

Emulator Pixel_2_API_28 Android 9, API 28

| Name | Permiss... | Date | Size |
|------------------------------|------------|------------------|-------|
| > proc | dr-xr-xr-x | 2021-09-20 13:23 | 0 B |
| > product | lrw-r--r-- | 2009-01-01 11:00 | 15 B |
| > sbin | drwxr-x--- | 2009-01-01 11:00 | 4 KB |
| ▼ sdcard | lrw-r--r-- | 2009-01-01 11:00 | 21 B |
| > Alarms | drwxrwx--x | 2021-07-20 09:50 | 4 KB |
| > allure-results | drwxrwx--x | 2021-09-01 10:18 | 12 KB |
| > Android | drwxrwx--x | 2021-09-07 10:25 | 4 KB |
| > DCIM | drwxrwx--x | 2021-09-01 10:07 | 4 KB |
| > Download | drwxrwx--x | 2021-07-20 09:50 | 4 KB |
| > googletest | drwxrwx--x | 2021-08-13 16:34 | 4 KB |
| > Movies | drwxrwx--x | 2021-07-20 09:50 | 4 KB |
| > Music | drwxrwx--x | 2021-07-20 09:50 | 4 KB |
| > Notifications | drwxrwx--x | 2021-07-20 09:50 | 4 KB |
| > odo | drwxrwx--x | 2021-09-20 20:45 | 8 KB |
| > Pictures | drwxrwx--x | 2021-07-20 09:50 | 4 KB |
| > Podcasts | drwxrwx--x | 2021-07-20 09:50 | 4 KB |
| > Ringtones | drwxrwx--x | 2021-07-20 09:50 | 4 KB |
| ▼ tia | drwxrwx--x | 2021-09-20 20:45 | 4 KB |
| ru.tinkoff.moba.sample.andro | -rw-rw---- | 2021-09-20 20:45 | 100 B |
| ru.tinkoff.moba.sample.andro | -rw-rw---- | 2021-09-20 20:45 | 101 B |
| > storage | drwxr-xr-x | 2021-09-20 13:23 | 100 B |
| > sys | dr-xr-xr-x | 2021-09-20 13:23 | 0 B |
| > system | drwxr-xr-x | 2009-01-01 11:00 | 4 KB |
| > vendor | drwxr-xr-x | 2009-01-01 11:00 | 4 KB |
| bugreports | lrw-r--r-- | 2009-01-01 11:00 | 50 B |

Gradle

Device File Explorer



Выгружаем Coverage



Необходимо выгрузить
.exes-файлы из устройства

Выгружаем Coverage

Необходимо выгрузить
.exes-файлы из устройства

Локально: `adb pull`

Выгружаем Coverage

Необходимо выгрузить
.exes-файлы из устройства

Локально: `adb pull`

На CI: поднимаем
AdbServer


Выгружаем Coverage



[Kaspresso AdbTerminal](#)

Coverage dump

▼  tia


 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#addNewSmsChannel.exec

 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#channels#412220.exec

 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#channels.exec

Coverage dump

▼  tia

 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#addNewSmsChannel.exec

 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#channels#412220.exec

 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#channels.exec

Coverage dump

▼  tia

 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#addNewSmsChannel.exec

 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#channels#412220.exec

 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#channels.exec

Coverage dump

▼  tia

 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#addNewSmsChannel.exec

 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#channels#412220.exec

 ru.tinkoff.sme.feature.notifications.test.NotificationsTest#channels.exec

Анализируем coverage

```
class TiaAnalyzer(  
    private val execFileLoader: ExecFileLoader  
) {  
  
    fun analyze(source: File): List<ImpactedFile> {  
        execFileLoader.load(source)  
  
        Analyzer(execFileLoader.executionDataStore) { coverage ->  
            ImpactedFile(  
                origin = coverage.fullClassName,  
                hits = coverage.coveredLines  
            )  
        }  
    }  
}
```

Анализируем coverage

```
class TiaAnalyzer(  
    private val execFileLoader: ExecFileLoader  
) {  
  
    fun analyze(source: File): List<ImpactedFile> {  
        execFileLoader.load(source)  
  
        Analyzer(execFileLoader.executionDataStore) { coverage ->  
            ImpactedFile(  
                origin = coverage.fullClassName,  
                hits = coverage.coveredLines  
            )  
        }  
    }  
}
```

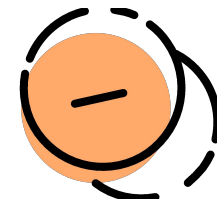
Анализируем coverage

```
class TiaAnalyzer(  
    private val execFileLoader: ExecFileLoader  
) {  
  
    fun analyze(source: File): List<ImpactedFile> {  
        execFileLoader.load(source)  
  
        Analyzer(execFileLoader.executionDataStore) { coverage ->  
            ImpactedFile(  
                origin = coverage.fullClassName,  
                hits = coverage.coveredLines  
            )  
        }  
    }  
}
```


Анализируем coverage

```
class TiaAnalyzer(  
    private val execFileLoader: ExecFileLoader  
) {  
  
    fun analyze(source: File): List<ImpactedFile> {  
        execFileLoader.load(source)  
  
        Analyzer(execFileLoader.executionDataStore) { coverage ->  
            ImpactedFile(  
                origin = coverage.fullClassName,  
                hits = coverage.coveredLines  
            )  
        }  
    }  
}
```

Проблема сбора покрытия



AndroidJUnitRunner запускает все
тесты в одном процессе
instrumentation

Проблема сбора покрытия



AndroidJUnitRunner запускает все тесты в одном процессе instrumentation



Покрытие одного теста перемешивается с покрытием другого

Проблема сбора покрытия



AndroidJUnitRunner запускает все тесты в одном процессе instrumentation



Покрытие одного теста перемешивается с покрытием другого



Чтобы coverage-анализ работал корректно, тесты должны быть изолированы друг от друга

Android Orchestrator

Android Orchestrator



Запускает каждый тест в своем
внутреннем Instrumentation

Android Orchestrator



Запускает каждый тест в своем
внутреннем Instrumentation



На каждый тест заводится свой
AndroidJUnitRunner

Android Orchestrator



Запускает каждый тест в своем внутреннем Instrumentation



На каждый тест заводится свой AndroidJUnitRunner



Решает проблему с тем, что каждый новый coverage наслаивается на предыдущий

Включаем Orchestrator

```
private fun BaseExtension.enableOrchestrator() {  
    testOptions { this: TestOptions  
        execution = "ANDROIDX_TEST_ORCHESTRATOR"  
    }  
}
```

```
{
  "ru.tinkoff.sme.feature.payments.ui.PaymentActivity": {
    "hits": {
      "11": [
        "48110",
        "48111"
      ],
      "15": [
        "48110"
      ]
    }
  },
  "ru.tinkoff.sme.feature.payments.di.PaymentComponent": {
    "hits": {
      "16": [
        "48110",
        "48111"
      ],
      "17": [
        "48110"
      ]
    }
  }
}
```

```
{
  "ru.tinkoff.sme.feature.payments.ui.PaymentActivity": {
    "hits": {
      "11": [
        "48110",
        "48111"
      ],
      "15": [
        "48110"
      ]
    }
  },
  "ru.tinkoff.sme.feature.payments.di.PaymentComponent": {
    "hits": {
      "16": [
        "48110",
        "48111"
      ],
      "17": [
        "48110"
      ]
    }
  }
}
```

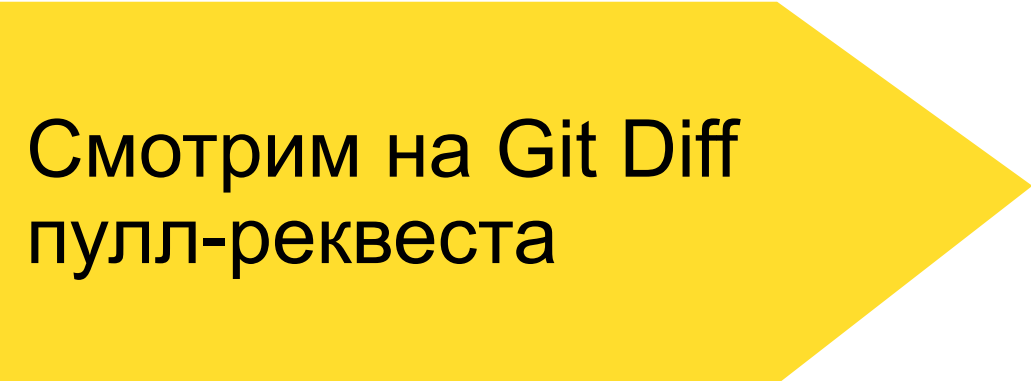
```
{
  "ru.tinkoff.sme.feature.payments.ui.PaymentActivity": {
    "hits": {
      "11": [
        "48110",
        "48111"
      ],
      "15": [
        "48110"
      ]
    }
  },
  "ru.tinkoff.sme.feature.payments.di.PaymentComponent": {
    "hits": {
      "16": [
        "48110",
        "48111"
      ],
      "17": [
        "48110"
      ]
    }
  }
}
```

```
{
  "ru.tinkoff.sme.feature.payments.ui.PaymentActivity": {
    "hits": {
      "11": [
        "48110",
        "48111"
      ],
      "15": [
        "48110"
      ]
    }
  },
  "ru.tinkoff.sme.feature.payments.di.PaymentComponent": {
    "hits": {
      "16": [
        "48110",
        "48111"
      ],
      "17": [
        "48110"
      ]
    }
  }
}
```

Прикручиваем Git



Прикручиваем Git



Смотрим на Git Diff
пулл-реквеста

Прикручиваем Git

Смотрим на Git Diff
пулл-реквеста

Вычленяем
измененные
файлы и строки

Прикручиваем Git

Смотрим на Git Diff
пулл-реквеста

Вычленяем
измененные
файлы и строки

Достаем
coverage-карту

Прикручиваем Git

Смотрим на Git Diff
пулл-реквеста

Вычленяем
измененные
файлы и строки

Достаем
coverage-карту

Получаем список
тестов, которые
нужно запустить

LoanApplicationFragment.kt

di

- LoansApplicationComponent.kt
- LoansApplicationDependencies.kt
- LoansApplicationModule.kt

71 71

72 72

73 -

73 +

74 74

```
private val overnightAnalytics by fastLazy {
```

```
    getComponent<LoansApplicationComponent>().overnightAnalytics
```

```
    getComponent<LoansApplicationComponent>().overnightOfferAnalytics
```

```
}
```

Git

```
{
  "ru.tinkoff.sme.feature.loans.ui.LoanApplicationFragment": {
    "hits": {
      "56": [
        "43001", "45234"
      ],
      "73": [
        "43001"
      ]
    }
  }
}
```

Git

```
{
  "ru.tinkoff.sme.feature.loans.ui.LoanApplicationFragment": {
    "hits": {
      "56": [
        "43001", "45234"
      ],
      "73": [
        "43001"
      ]
    }
  }
}
```

Git

```
{
  "ru.tinkoff.sme.feature.loans.ui.LoanApplicationFragment": {
    "hits": {
      "56": [
        "43001", "45234"
      ],
      "73": [
        "43001"
      ]
    }
  }
}
```

ИТОГИ

ИТОГИ



Наследуемся от
InstrumentationRunListener'a

ИТОГИ



Наследуемся от
`InstrumentationRunListener`'а



Для сбора coverage используем
JaCoCo

ИТОГИ



Наследуемся от
InstrumentationRunListener'а



Для сбора coverage используем
JaCoCo



Используем **Orchestrator**

Нюансы использования



ТИНЬКОФФ

tinkoff.ru

Нюансы ИСПОЛЬЗОВАНИЯ



Что делать с Android-вызовами?

Android-ВЫЗОВЫ

```
fun getText(context: Context): String {  
    return if (isCurrencyContractsRedesignEnabled()) {  
        context.resources.getString(  
            R.string.currency_contract_details_doc_submission_description_2,  
            docSubmission.cutOffDate.toDateMonthFormat()  
        )  
    } else {  
        "Требуются документы до {docSubmission.cutOffDate.toDate...}"  
    }  
}
```

Android-ВЫЗОВЫ

```
fun getText(context: Context): String {  
    return if (isCurrencyContractsRedesignEnabled()) {  
        context.resources.getString(  
            R.string.currency_contract_details_doc_submission_description_2,  
            docSubmission.cutOffDate.toDateMonthFormat()  
        )  
    } else {  
        "Требуются документы до {docSubmission.cutOffDate.toDate...}"  
    }  
}
```

Нюансы ИСПОЛЬЗОВАНИЯ



Что делать с Android вызовами?

– Используем `jvmtiAgents`

Нюансы ИСПОЛЬЗОВАНИЯ



Что делать с Android вызовами?

- Используем `jvmtiAgents`
- Ставим брейкпоинт на `AssetManager.getResourceValue`

Нюансы ИСПОЛЬЗОВАНИЯ



Что делать с Android вызовами?



Тестов может быть слишком много

Нюансы ИСПОЛЬЗОВАНИЯ



Что делать с Android вызовами?



Тестов может быть слишком много



Мигающие тесты

Спасибо за внимание

Щепалин Максим, Тинькофф.Бизнес

t.me/lounvhx

twitter.com/lounvhx



ТИНЬКОФФ

tinkoff.ru