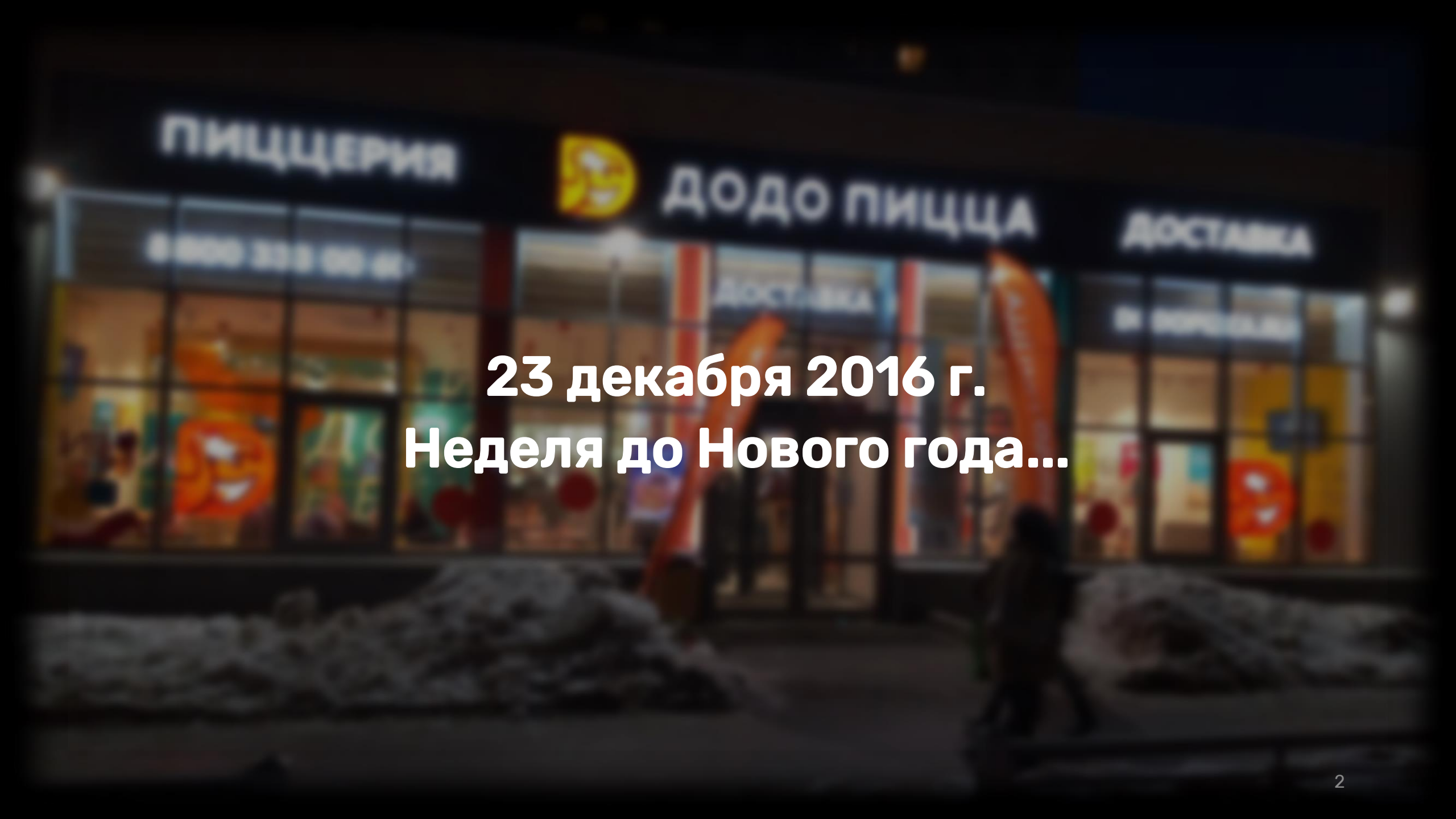
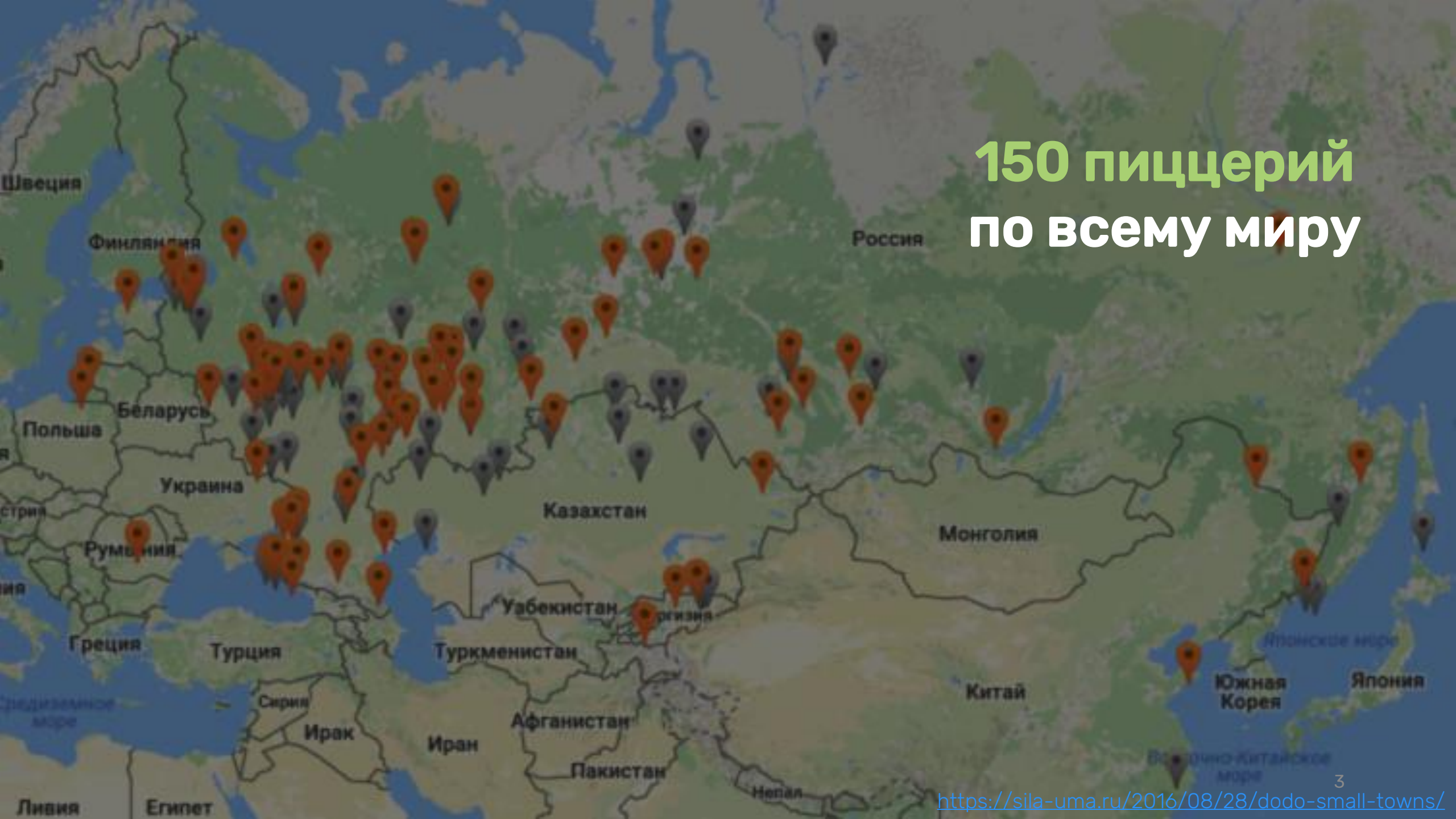
Инъекция тестовых поведений

Как выйти сухим из воды?

Владимир Плизга
ЦФТ



23 декабря 2016 г.
Неделя до Нового года...



https://sila-uma.ru/2016/08/28/dodo-small-towns/

25 декабря
открытие пиццерии
в Санкт-Петербурге

https://images.app.goo.gl/n1vHtL5ZygLWjsNM8

https://vc.ru/flood/20958-dodo-10m

Офтоп Alina Tolmacheva 24 дек 2016 30 746

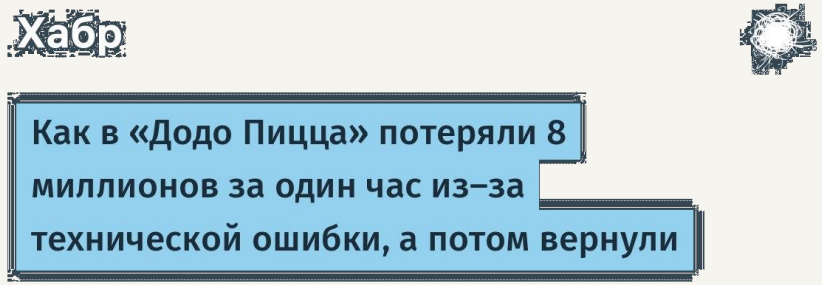
«Додо Пицца» по ошибке перечислила клиентам 10 млн рублей за уже оплаченные заказы

Генеральный директор «Додо пицца» Федор Овчинников рассказал на своих страницах в соцсетях, что из-за технической ошибки российские клиенты сети получили 10 млн рублей за совершенные заказы, которые они уже оплачивали ранее. Пока Овчинников не знает, как решить ситуацию.

“

… оказалось, что фоновая задача смотрит …
*на **реальное** подключение к Яндекс.Кассе.*

… при этом фоновая задача смотрела
*на версию **тестовой** базы.*



Андрей Арефьев

Руководитель направления
электронной коммерции
«Додо Пицца»

Особенно обидно было осознавать, что мы вернули деньги, которых не получали — это были ***тестовые*** *заказы.*

https://vc.ru/flood/21255-dodopizza-money-back

Что тесту хорошо,
то production'у – смерть!

Редкое поведение

Mocks & stubs

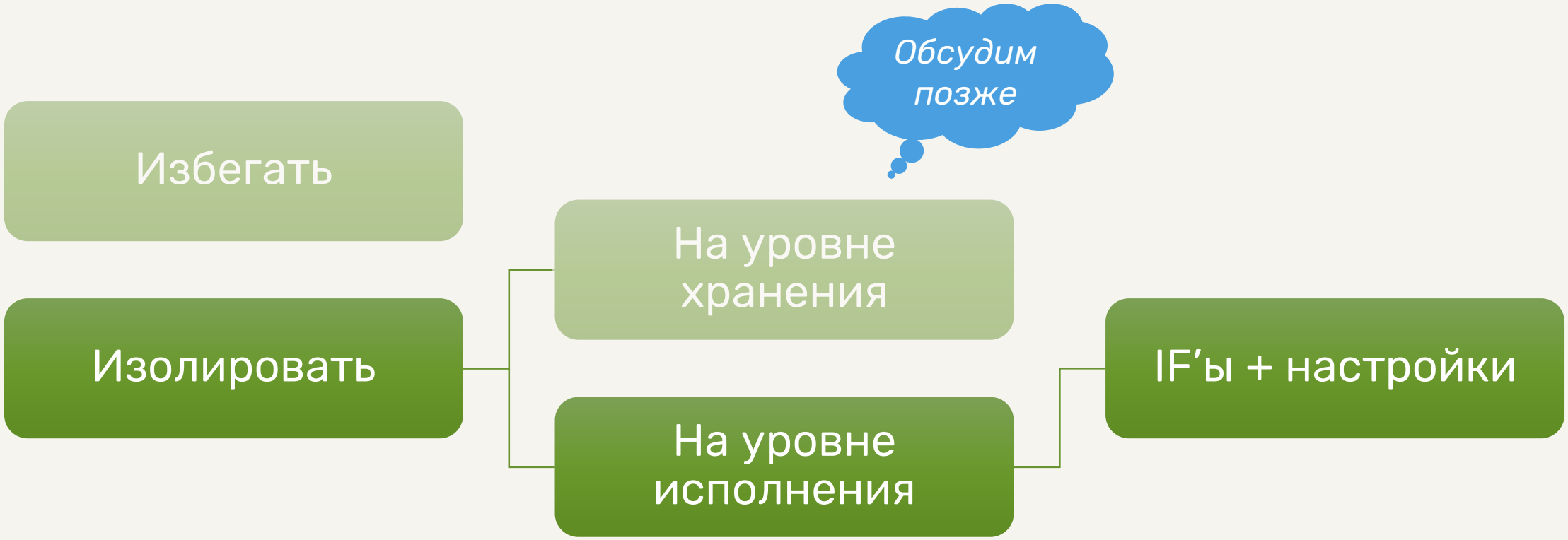
Логирование

Антибезопасность

(добавь своё)

Not for production!

И как быть?


Обсудим позже
Избегать
Изолировать
На уровне хранения
На уровне исполнения
IF’ы + настройки

IF'ы + настройки: Play! Framework

```
// Mode
try {
    mode = Mode.valueOf(configuration.getProperty("application.mode", "DEV").toUpperCase())
```

```
class DefaultMailSystemFactory extends AbstractMailSystemFactory {

    private static final MailSystem LEGACY_MOCK_MAIL_SYSTEM = new LegacyMockMailSystem();
    private static final MailSystem PRODUCTION_MAIL_SYSTEM  = new ProductionMailSystem();

    @Override
    public MailSystem currentMailSystem() {
        if (Play.useDefaultMockMailSystem()) {
            return LEGACY_MOCK_MAIL_SYSTEM;
        } else {
            return PRODUCTION_MAIL_SYSTEM;
        }
    }

}
```

https://github.com/playframework/play1/blob/master/framework/src/play/Play.java

IF'ы + настройки: HotSpot JVM

```
DEBUG_ONLY(if (ArchiveRelocationMode == 1 && use_requested_addr) {
  // This is for simulating mmap failures at the requested address. In debug builds, we do it
  // here (after all archives have possibly been mapped), so we can thoroughly test the code for
  // failure handling (releasing all allocated resource, etc).
  log_info(cds)("ArchiveRelocationMode == 1: always map archive(s) at an alternative address");
  if (static_result == MAP_ARCHIVE_SUCCESS) {
    static_result = MAP_ARCHIVE_MMAP_FAILURE;
  }
  if (dynamic_result == MAP_ARCHIVE_SUCCESS) {
    dynamic_result = MAP_ARCHIVE_MMAP_FAILURE;
  }
});
```

http://hg.openjdk.java.net/jdk/jdk14/file/6c954123ee8d/src/hotspot/share/memory/metaspaceShared.cpp

IF’ы + настройки: резюме

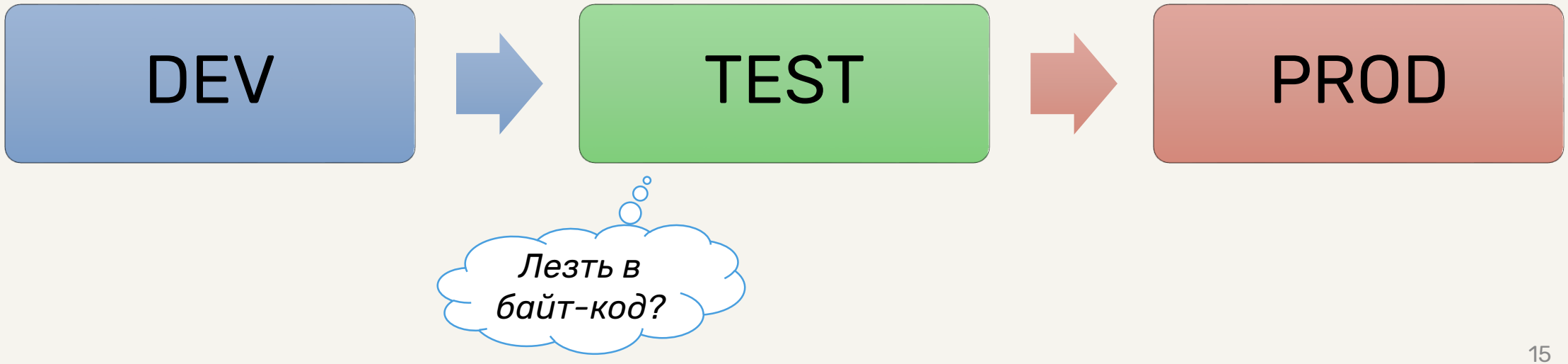
- Применимы **не везде** (например, в библиотеках)
- Нужно предусматривать **заранее**
- Легко **накосячить**

И как быть?


Обсудим позже
Избегать
Изолировать
На уровне хранения
На уровне исполнения
IF’ы + настройки
Side Effect Injection

Ключевая идея Side Effect Injection

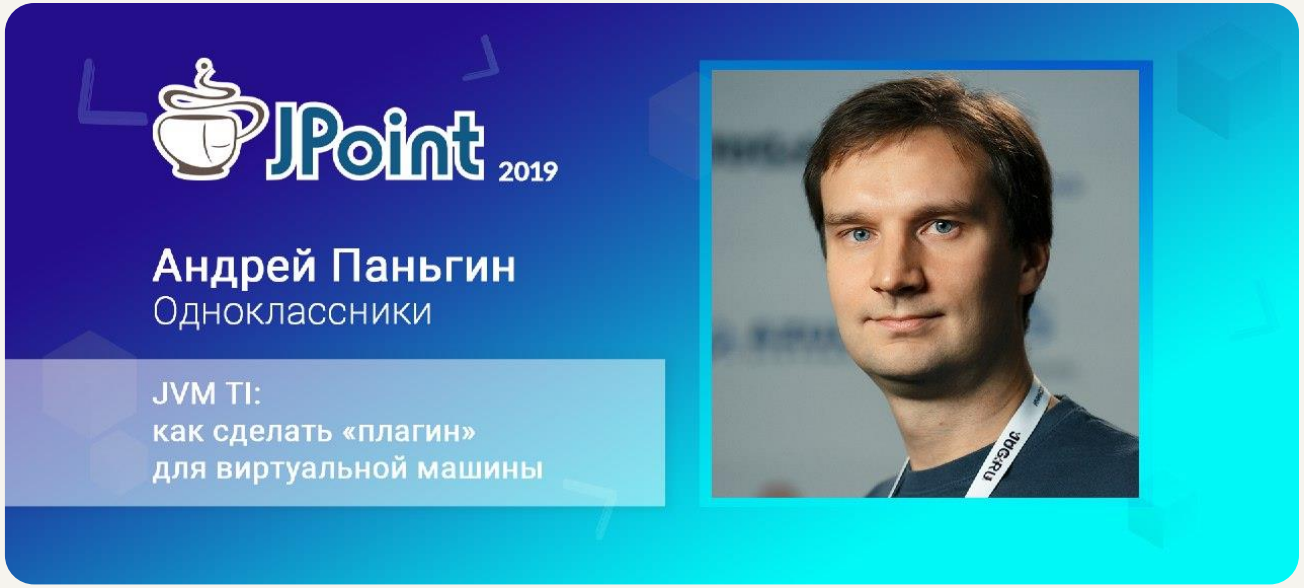
Писать исходный код как обычно, а тестовое поведение **внедрять извне** и только там, **где это нужно**.



Java Virtual Machine Tool Interface

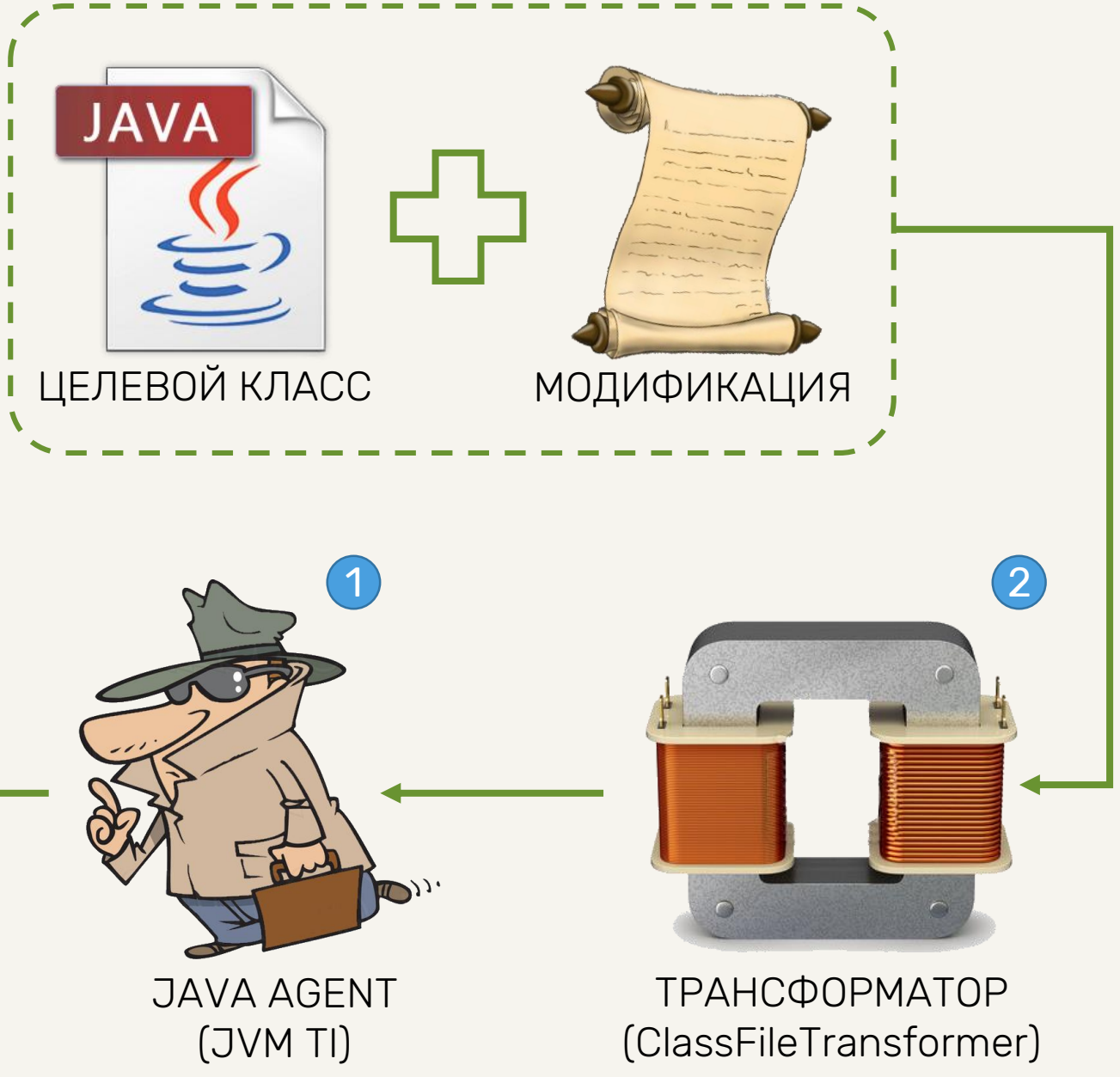
JVM TI позволяет трансформировать байт-код классов во время их загрузки.

- https://docs.oracle.com/en/java/javase/14/docs/specs/jvmti.html
- https://habr.com/ru/company/odnoklassniki/blog/458812/



Side Effect Injection

На пальцах



1 Java agent

MyJavaAgent.java

```
package com.example;
public class MyJavaAgent {
  public static void premain(String agentArgs, Instrumentation inst) {
    // ...
  }
}
```

myagent.jar!/META-INF/MANIFEST.MF

```
Manifest-Version: 1.0
Premain-Class: com.example.MyJavaAgent
```

```
$ java -javaagent:myagent.jar=arg1,arg2,...
```

2 Трансформатор

ClassFileTransformer.java

```
package java.lang.instrument;

public interface ClassFileTransformer {
byte[]
    transform(  ClassLoader          loader,
                String               className,
                Class<?>             classBeingRedefined,
                ProtectionDomain     protectionDomain,
                byte[]               classfileBuffer)
        throws IllegalClassFormatException;
}
```

Принцип один.
Реализаций много.

AspectJ

Byteman

Принцип один.
Реализаций много.

GluonJ

jMint

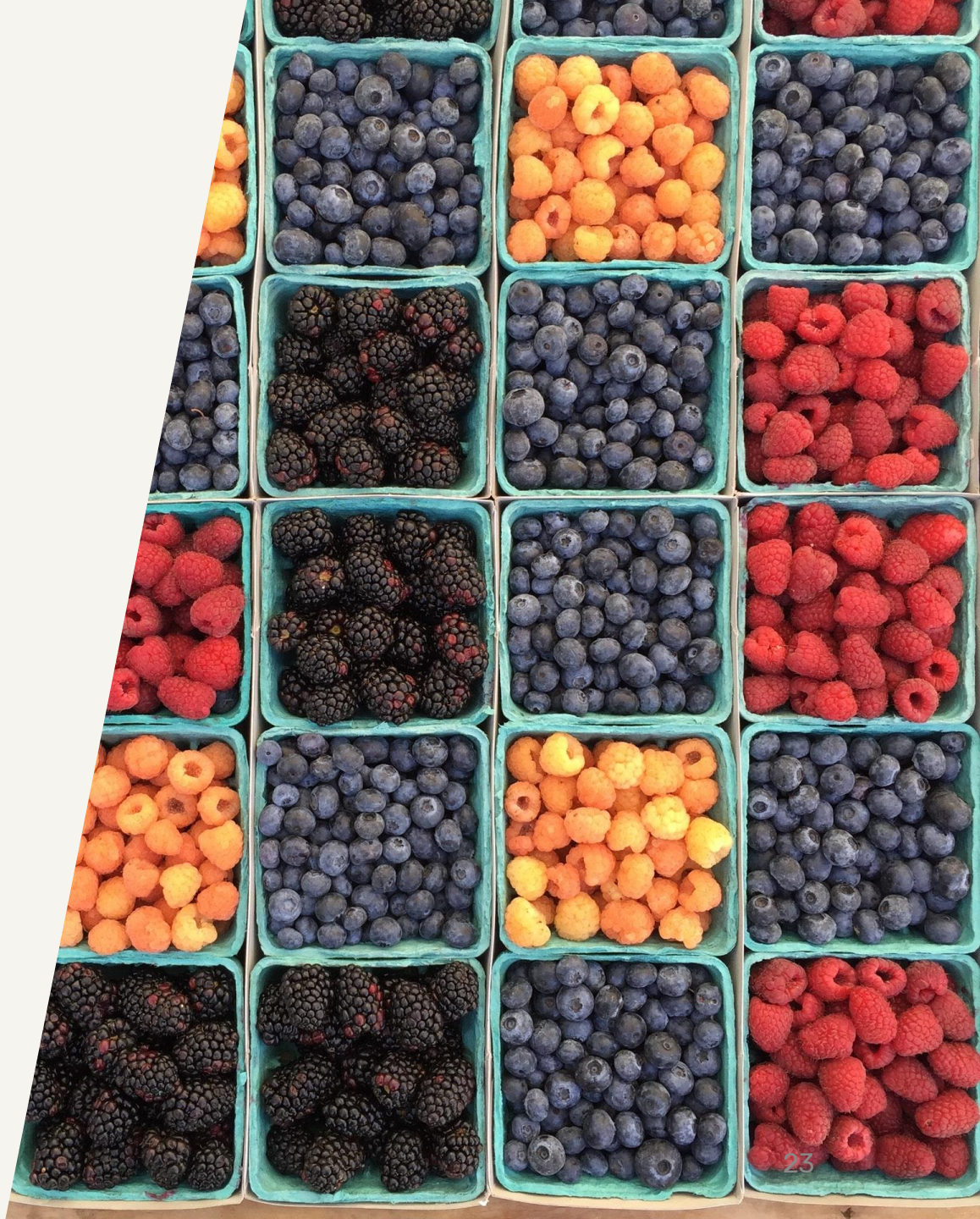
Eclipse AspectJ

Что?	Аспектно-ориентированное **расширение** Java *С версии 5 можно писать и на «чистой» Java* 1
Когда?	Создан в **2001 г.** компанией PARC
Кто?	Живёт под зонтиком **Eclipse Foundation**

“

AspectJ enables
clean modularization
of crosscutting concerns,
such as error checking and
handling, synchronization,
context-sensitive behavior,
performance optimizations,
monitoring and logging,
debugging support,
and multi-object protocols

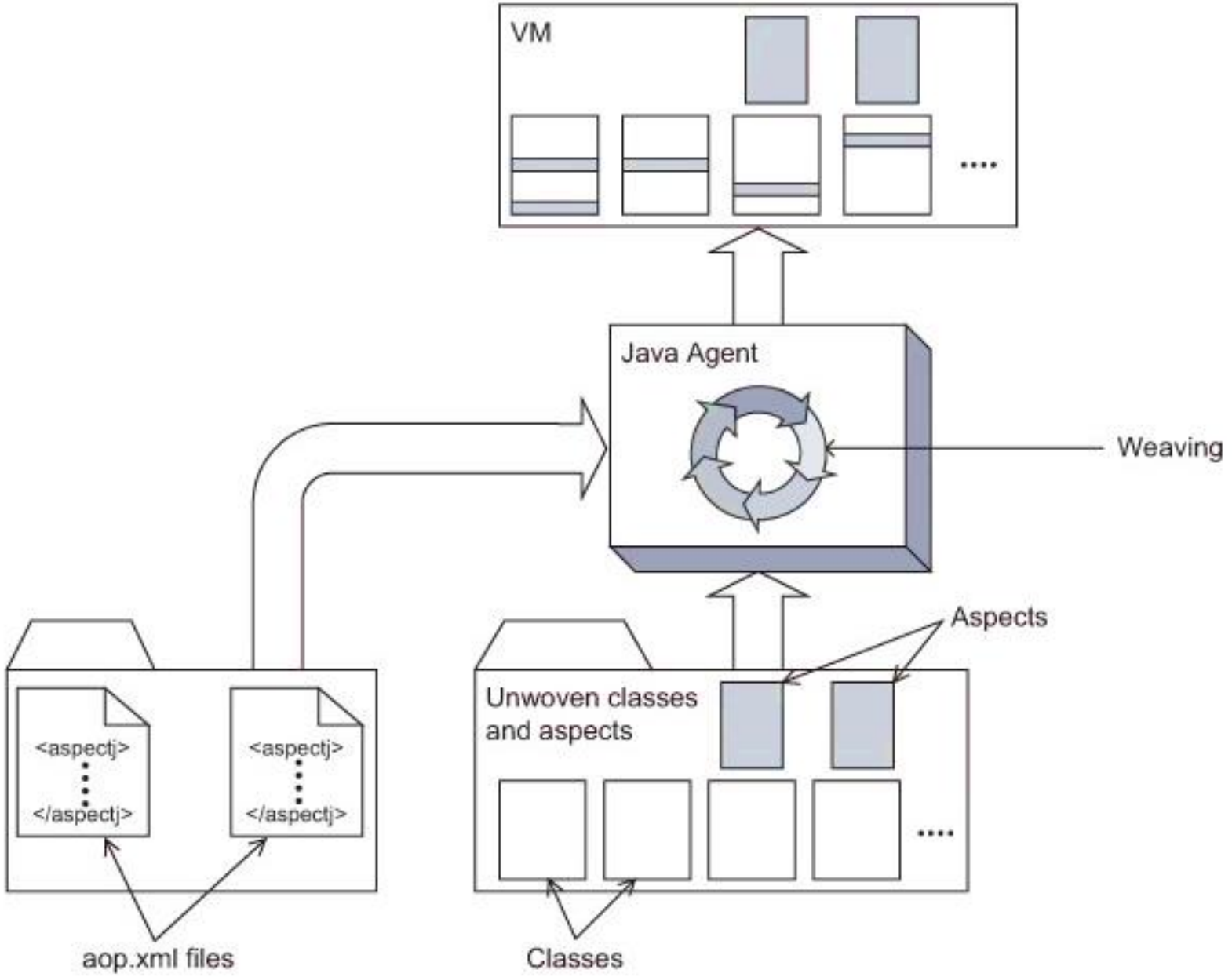
https://www.eclipse.org/aspectj/



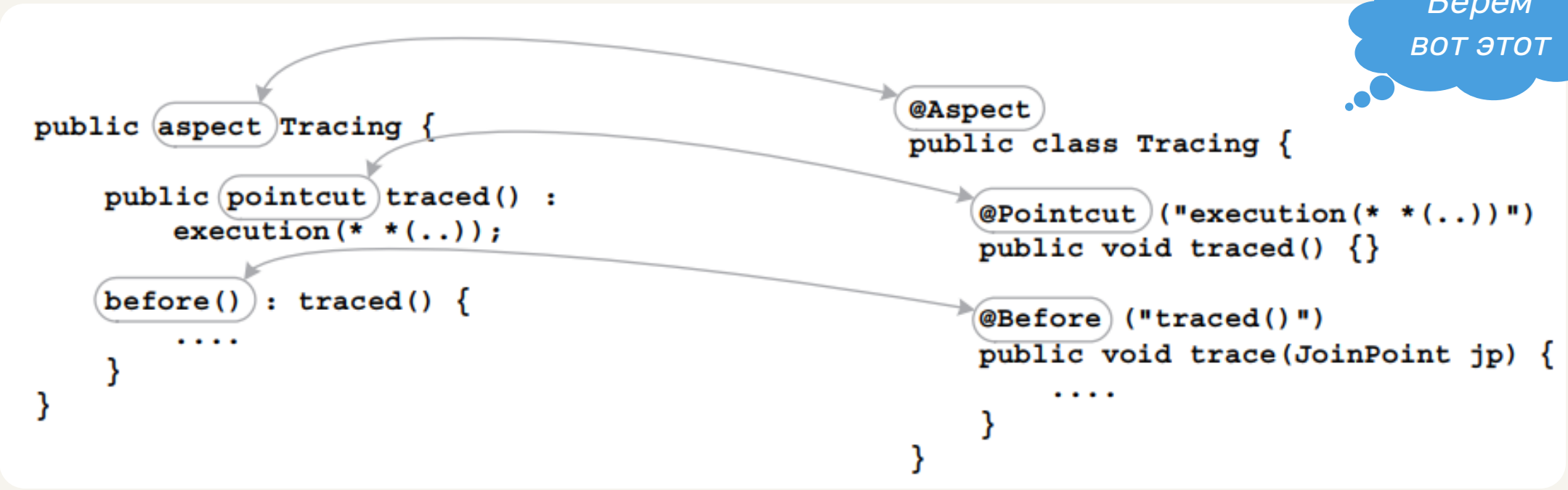
Устройство AspectJ Weaver

По мнению авторов

Источник: книга **AspectJ in Action** (2nd Edition) глава **8.3.1**



1 AspectJ поддерживает 2 формата



Источник: **AspectJ in Action**
(2nd Edition), глава **7.1**

Пора бы и код показать…

AspectJ: заметки

- 👍 Гибкий язык модификаций
- 👍 Отличная поддержка в IDE

- 👎 Нужно хранить вместе с целевым кодом
- 👎 Нужно тянуть `aspectjrt.jar` к себе в classpath

JBoss Byteman

Что?	Инструмент для трассировки, мониторинга, **отладки** и **тестирования** приложений на Java
Когда?	Разрабатывается с **2009 г.** как один из проектов JBoss
Кто?	Спонсируется **Red Hat**

“

It injects Java code into your application … without the need for you to recompile, repackage or even redeploy your application.

https://byteman.jboss.org/

Пора бы и код показать…

Byteman: заметки

- 👍 Легко подключается к приложению
- 👍 Умеет править классы в `java.lang` и т.п.

- 👎 Нужно помнить язык модификаций
- 👎 Нет поддержки в IDE

jMint

Что? — Инструмент для внедрения **отладочных** и **тестовых** поведений в Java приложения

Когда? — Создан в **2016 г.** как личный R&D проект

Кто? — Активно применяется в **ЦФТ** (дирекция PrePaid)

“

Модификации должны описываться так же, как если бы мы делали их прямо в исходном коде.

придумал только что

Дроплет – это модификация в jMint

- Версия целевого класса, содержащая **только модифицирующий код**
- **Игнорирует** модификаторы доступа, поля, наследование, аннотации и любые непомеченные методы
- Может иметь в имени суффикс `Droplet`
- Может создаваться **2-мя способами**:
  - **с нуля**: ничего лишнего, но надо писать руками
  - **из копии**: «ломать не строить», но остается много шума

Как будем делать дроплет?

Напишите номер варианта в нашем чате на YouTube:

С нуля

Из копии

Пора бы и код показать…

jMint: заметки

- 👍 Синтаксис похож на целевой класс
- 👍 Модификации можно хранить где угодно

- 👎 Ущербный язык модификаций
- 👎 Неочевидность отличий от целевого класса

Status
Changes
Workspace
Build with Parameters
Configure
Delete Project
Rebuild Last
Favorite
Move
Job Config History
Rename

Build History trend

find

#112
Wed Jul 22 12:41:45 NOVT 2020

#111
Wed Jul 22 11:12:54 NOVT 2020

#110
Tue Jul 21 09:46:39 NOVT 2020

#109

Project upc2sand

Droplets

Подключить дроплеты:

☑ core/AuthenticateMP.java	
☐ core/BookerImpl.java	
☐ core/CDASClientImpl.java	
☐ core/CardUpc.java	
☑ core/CardUpc_isFirstLogin_Tu	
☐ core/CommonAbstractAmqpP	
☐ core/Consumer.java	
☐ core/ControlRateDecrease.java	
☐ core/ControlRateIncrease.java	
☐ core/CoreContextImpl.java	
☐ core/CoreContextImpl_RT.java	Дроплет предназначен для имитации работы через мобильное приложение. Это необходимо, например, при тестировании
☐ core/Crypter.java	Отключает шифрование паролей
☐ core/EquitiesProvider.java	Исключает тип баланса BNS_DEBT для мобильных платформ.
☑ core/EurekaInstanceConfigProvider.java	Принудительно включает регистрацию комплекса в Eureka по IP-адресу вместо hostName.
☐ core/FRAMOSClientImpl.java	Подменяет статус проверки платежа на фрод после получения ответа от FRAMOS
☐ core/LimitsManagerImpl.java	Подменяет поход в Limon на заведомо успешную проверку лимитов с фейковым ID проверки.
☐ core/LimitsSyncService.java	Подменяет номер счета для загрузки лимитов на заведомо корректный
☐ core/LoyaltyApiClientImpl.java	При загрузке бонусных балансов с использованием REST-API подменяет ean на зарегистрированный в Лояльности.
☐ core/OperationEnricher.java	Логирует сообщения об элементах Истории, игнорируемых при загрузке в Elastic
☐ core/ParameterType.java	При переводах ЗК обеспчечивает передачу в ДП фиксированного курса валюты
☐ core/PayPassManagerImpl.java	@cutpoint INSTEAD
☐ core/PcvrClientImpl.java	Подменяет ответ PCVR
☐ core/QpayAdapterImpl.java	Для 2960264879676 формирует ответ от ДП о превышении лимита и формирует ответ от ДП для входящего перевода ЗК ДП для oid=246886901
☐ core/QpayClientImpl.java	Возвращает ошибку об аресте карты пользователя при попытке заказать paypass
☐ core/QpayClientImplLimitExceeded.java	Возвращает ошибку о превышении максимальной суммы перевода при попытке "Получить наличные" с валютного кошелька (больше или равно 4400 $/€)
☐ core/QpayClientImpl_Block_Arrested.java	Возвращает ошибку об аресте карты по ФЗ-167 пользователя при попытке входа
☐ core/QpayClientImpl_TranStatus.java	Подменяет статусы указанных по oID адресных денежных переводов
☐ core/QpayClientImpl_TranStatus_RT.java	Подменяет статусы указанных по oID адресных денежных переводов
☐ core/QpayModule.java	Подключает имитатор Денежных переводов (QPAY).
☐ core/QpayNameFormProviderDecrease.java	При переводах ЗК уменьшает курс до 40, обеспечивает видимость изменения курса при заполнении формы
☐ core/QpayNameFormProviderIncrease.java	При переводах ЗК увеличивает курс до 70, обеспечивает видимость изменения курса при заполнении формы
☐ core/QpayNameServiceBuilder.java	Добавляет на форму ДП по ФИО статическую надпись, чтобы эта форма явно визуально отличалась от остальных.
☐ core/QpayPhoneFormProvider.java	Для эмуляции ошибок по лимитам от ДП в DP3(QpayFault 4516, 4517, 4518, 4519, 4520 для переводов отправленных на указанные в дроплете ean-ы)

Подключить дроплеты:
Droplets
☑ core/AuthenticateMP.java
☐ core/BookerImpl.java
☐ core/CDASClientImpl.java
☐ core/CardUpc.java

jMint: кишочки

https://toparvion.pro/talk/2018/jbreak/

GluonJ

Что? Расширение Java по мотивам открытых классов **Ruby** и промежуточных типов **AspectJ**

Когда? Разрабатывался в **2016 г.** как личный академический проект

Кто? Автор: Shigeru Chiba – разработчик библиотеки Javassist (⭑ 2800+)

“

… a reviser directly modifies
the definition of an existing class;
it can … override a method
in the target class.

https://github.com/chibash/gluonj

Как выглядит reviser

SayHello.java

```
package sample;
public reviser SayHello extends test.Person {
    public void greet() {
        System.out.println("Hello!");
    }
}
```

```
java -jar GluonJCompiler.jar test/Person.java sample/SayHell
java -jar gluonj.jar test/Person.class sample/SayHello.class
```

Сравнение инструментов

Свойство\инструмент	AspectJ	Byteman	jMint
Модификация приватных методов	✓	✓	✓
Отдельное хранение модификаций	–	✓	✓
Поддержка в IDEA	✓	–	✓[1]
Тот же язык, что у целевого класса	✓[2]	–	✓
Чистый classpath	–	✓	✓
Внедрение в классы JVM	–	✓	–
Добавление новых полей и методов	✓	–	–
Подключение “на лету”	–	✓	–

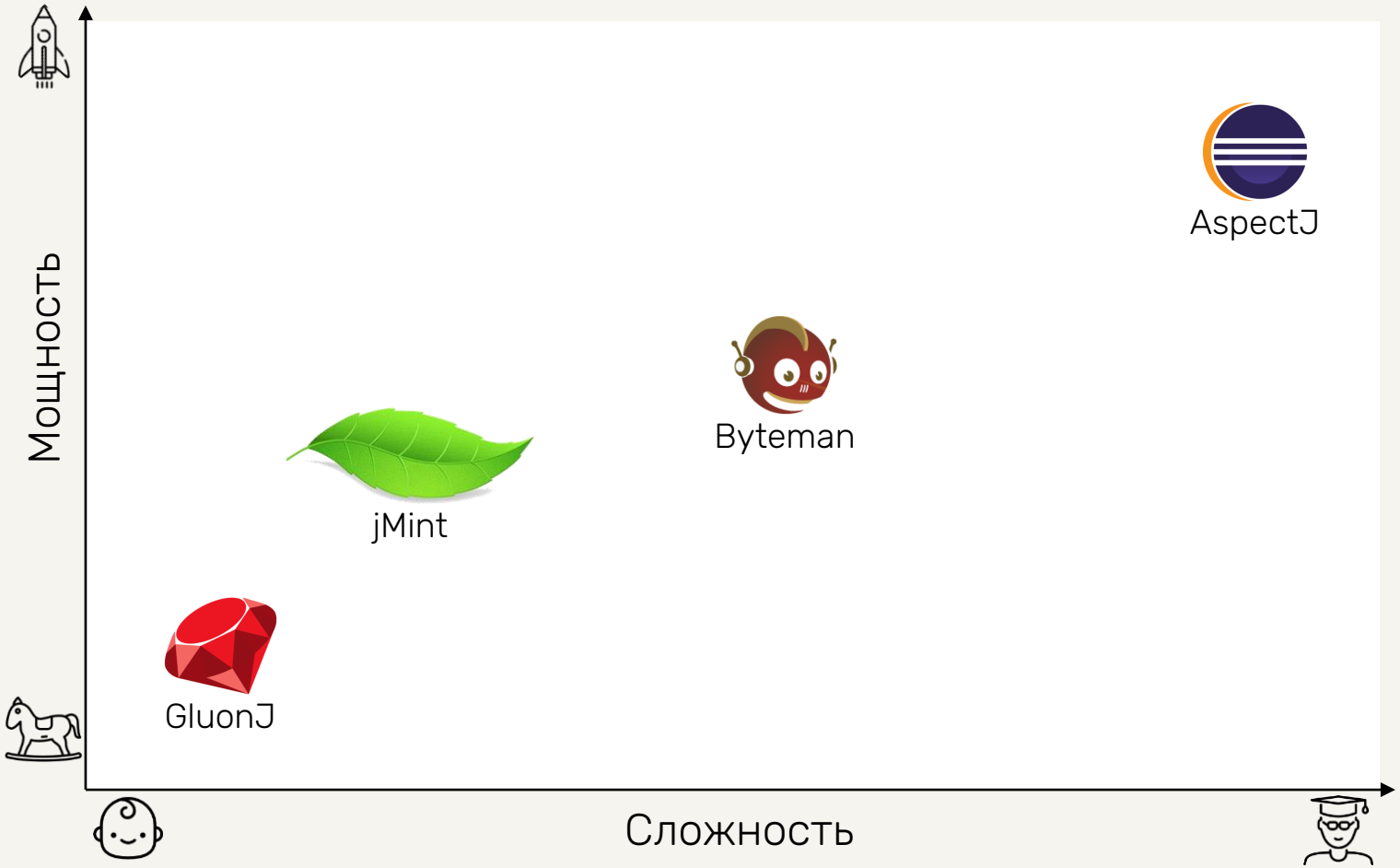
[1] Но без учёта ограничений Javassist
[2] Но не «чистый» AspectJ и выражения в pointcut’ах

Щас будет график

Вкусовщина

Субъективщина

Мощность VS сложность



Side Effect Injection: резюме

- «Боевой» код и настройки остаются **чистыми**
- Модификации должны быть **простыми**

Side Effect Injection + настройки

DevController.java

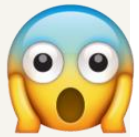
```java
@RestController
@Profile({DEV, TEST})
public class DevController {

  @PostMapping("/dev/reset-locks")
  public ResponseEntity<Void> resetLocks() {
    if (isDisabled()) {
      return ResponseEntity.of(empty());
    }
    // ...
  }

  private boolean isDisabled() {
    return true;
  }
}
```

1 Источники настроек в Spring Boot

1. Devtools global settings properties in the $HOME/.config/spring-boot directory when devtools is active.
2. @TestPropertySource annotations on your tests.
3. properties attribute on your tests.
4. Command line arguments.
5. Properties from SPRING_APPLICATION_JSON (inline JSON embedded in an environment variable or system property).
6. ServletConfig init parameters.
7. ServletContext init parameters.
8. JNDI attributes from java:comp/env.
9. Java System properties (System.getProperties()).
10. OS environment variables.
11. A RandomValuePropertySource that has properties only in random.*.
12. Profile-specific application properties outside of your packaged jar (application-{profile}.properties and YAML variants).
13. Profile-specific application properties packaged inside your jar (application-{profile}.properties and YAML variants).
14. Application properties outside of your packaged jar (application.properties and YAML variants).
15. Application properties packaged inside your jar (application.properties and YAML variants).
16. @PropertySource annotations on your @Configuration classes.
17. Default properties (specified by setting SpringApplication.setDefaultProperties).



https://docs.spring.io/spring-boot/docs/current/reference/html/spring-boot-features.html#boot-features-external-config

2 Дополнительный рубеж защиты

DevControllerDroplet.java

```
package my.app.controller.dev;

public class DevController {

  /**
   * @cutpoint INSTEAD
   */
  private boolean isDisabled() {
    log.warn("DROPLET: Запрет на вызов отладочного контроллера снят");
    return false;
  }
}
```

Side Effect Injection: резюме

- «Боевой» код и настройки остаются **чистыми**
- Модификации должны быть **простыми**
- Модификации лучше хранить **отдельно**
  - *И быть готовым к их устареванию* 😜
- Применять **по назначению**
  - *Оценивая альтернативы*

Всё, что может пойти не так, обязательно пойдёт не так.

Закон Мерфи

AspectJ

www.eclipse.org/aspectj

Byteman

byteman.jboss.org

jMint

github.com/toparvion/jmint

Владимир Плизга
ЦФТ

@toparvion

Toparvion

https://toparvion.pro/

Q&A

Источники материалов

- Photo by William Felker on Unsplash
- Photo by Karolina Grabowska from Pexels
- Photo by Marc Schulte on Unsplash
- Photo by Brett Sayles from Pexels
- https://www.flaticon.com/authors/freepik
- https://www.flaticon.com/packs/archeology-40