**Давлятшин Роман**

JavaScript Developer

at EPAM, Drill4J

WORKSHOP

# Оцениваем качество и ускоряем тесты на Cypress при помощи Drill4J

## на примере React* приложения

*применимо и к другим frontend фреймворкам

# AGENDA

1. О проблемах
2. Клонируем, запускаем, изучаем example project
3. Интегрируем Drill4J
4. FAQ/Q&A

# ПОКА МЫ РАЗГОВАРИВАЕМ...

Установить

- **Docker**
  https://www.docker.com/
- **Git**
  https://git-scm.com/
- **Node**
  https://nodejs.org/en/
- **Jq** (можно позже, это мелкая утилита)
  https://github.com/stedolan/jq/releases

- **Инструкции/шпаргалка/how-to**
  clone
  https://github.com/Drill4J/heisenbug-moscow-2021-workshop
  cd drill-services, docker-compose up -d

- **Example project**
  clone https://github.com/Drill4J/realworld-react-cypress.git
  cd, npm install

- **@drill4j/js-parser**
  npm i -g @drill4j/js-parser

# О проблемах

## ОДНАЖДЫ В ПРОДАКШН

*камерная трагедия в трех действиях*

~~Не~~выдуманный кусочек жизни любого проекта

герои:

Автоматизаторы - лучшие автоматизаторы

Тестировщики - лучшие ручные тестировщики

Девелоперы - лучшие девелоперы

Проект-менеджер - лучший менеджер

# О проблемах #1

## ДЕЙСТВИЕ ПЕРВОЕ

*Новая фича*

Действующие лица: Девелопер, Автоматизатор тестирования, Ручной тестировщик

Девелопер: Фича А готова!

Автоматизатор: Все тесты зеленые!

...

*через неделю (или две)*

Тестировщик: а что если снять галочку, засабмитить форму, а затем поставить её снова?

Тестировщик: клик-клик!

*приложение крашится*

# О проблемах #2

## ДЕЙСТВИЕ ВТОРОЕ

*Багфикс*

Действующие лица: Девелопер

Девелопер: git commit -m “fix: remove bug”

Девелопер: git push origin dev

Девелопер: надеюсь, это работает.

*запускаются тесты, estimated execution time > 2 часов*

Девелопер: по-о-онятно, да конечно это работает!

*забивает на тест и идет пить кофе/писать код дальше*

...

*тесты фейлятся спустя 2 часа 10 минут*

# О проблемах #3

## ДЕЙСТВИЕ ТРЕТЬЕ

*Запуски по расписанию*

Действующие лица: Девелоперы, Автоматизатор, Все

Автоматизатор: ресурсов мало, коммитов много, поставлю запуски тестов два раза в день

Все: ок

...

*позже*

Девелоперы 1, 2 и 3 (каждый): git push origin dev

...

*тесты запускаются через пару часов, и фейлятся еще через час*

Девелоперы 1, 2 и 3 (хором): понятия не имею, что происходит

# О проблемах #4

## НЕМАЯ СЦЕНА

Действующие лица: все

*предупреждение - возможна излишняя драматизация*

Фикс багов на 90% состоит из “кто, где и когда сломал?”

Автотесты запускаются редко и идут мучительно долго.

Ручные тестировщики гоняются за heisenbug-ами.

Проект-менеджер нервничает.

# Выводы?

В идеале...

1. Тесты должны быть **достаточными**

2. Тесты должны предоставлять **быстрый фидбек**

3. Тесты должны быть **актуальными**

# Как этого добиться?

В идеале...

1. Тесты должны быть **достаточными (*покрывать “все что нужно”*)**

Test plan; Checklist; Test case; Bug report; Traceability matrix

...coverage report?

2. Тесты должны предоставлять **быстрый фидбек (*чем быстрее тем лучше*)**

Делим, приоритезируем, параллелим

3. Тесты должны быть **актуальными (*без лишних повторений, без пустышек*)**

Внимательно следим за тестами

# Проблема “черного ящика”?

# Задаемся вопросами

1. Насколько хороши наши тесты?

   **Узнать что конкретно делает каждый тест**

2. Какие есть риски?

   **Обнаружить измененный, либо новый код**

3. Какие тесты запустить чтобы покрыть риски?

   **Найти оптимальный набор тестов**

4. Какая динамика между билдами?

   **Применять вышеуказанные действия к каждому билду, на постоянной основе**

# Drill4J это open-source инструмент, который...

1

**Отслеживает** выполнение кода внутри приложения во время тестов

*без изменений кода и билда, on-premises, без security рисков*

2

**Строит “карту”** связей тестов с кодом

*просто говоря - записывает какой тест вызвал какой код*

3

**Находит риски** в новых билдах и **предлагает тесты**, которые их покроют

*и проверяет действительно ли все было запущено и покрыто*

# Чего Drill4J не знает и не делает

1. Мы ничего не знаем о state

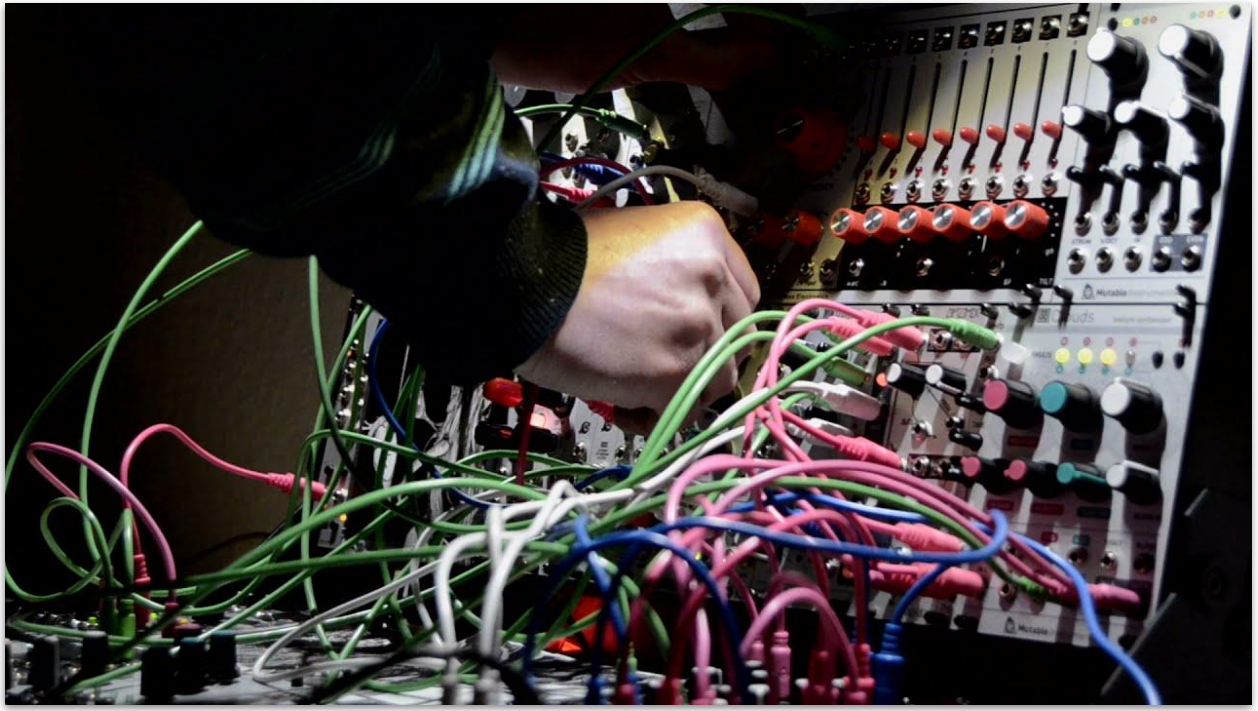
   **Работа кода зависит от времени дня?**

2. Мы не знаем о зависимостях между тестами

   **Один тест регистрирует юзера, а следующий делает логин?**

**Это не то, как работает Drill4J --->**

# Важный disclaimer



Drill4J это развивающийся инструмент

еще не 1.0.0, но мы уже можем приносить пользу :)

# Drill4J доступен для...

**ПЛАТФОРМЫ**

- Java (и JVM-based languages, e.g. Scala)
- WEB (JS/TypeScript, React/Angular/JQuery)
- .NET - WIP 🕑

**БРАУЗЕРЫ**

Backend - Chrome/Chromium/Firefox

WEB - Chrome/Chromium

**TESTING FRAMEWORKS/TOOLS**

- Selenium (Java, Node.js, …?)
- Cypress
- Postman

**CI/CD**

Везде, где можно сделать HTTP запрос

**Микросервисы поддерживаете?**

Да

# Время практики

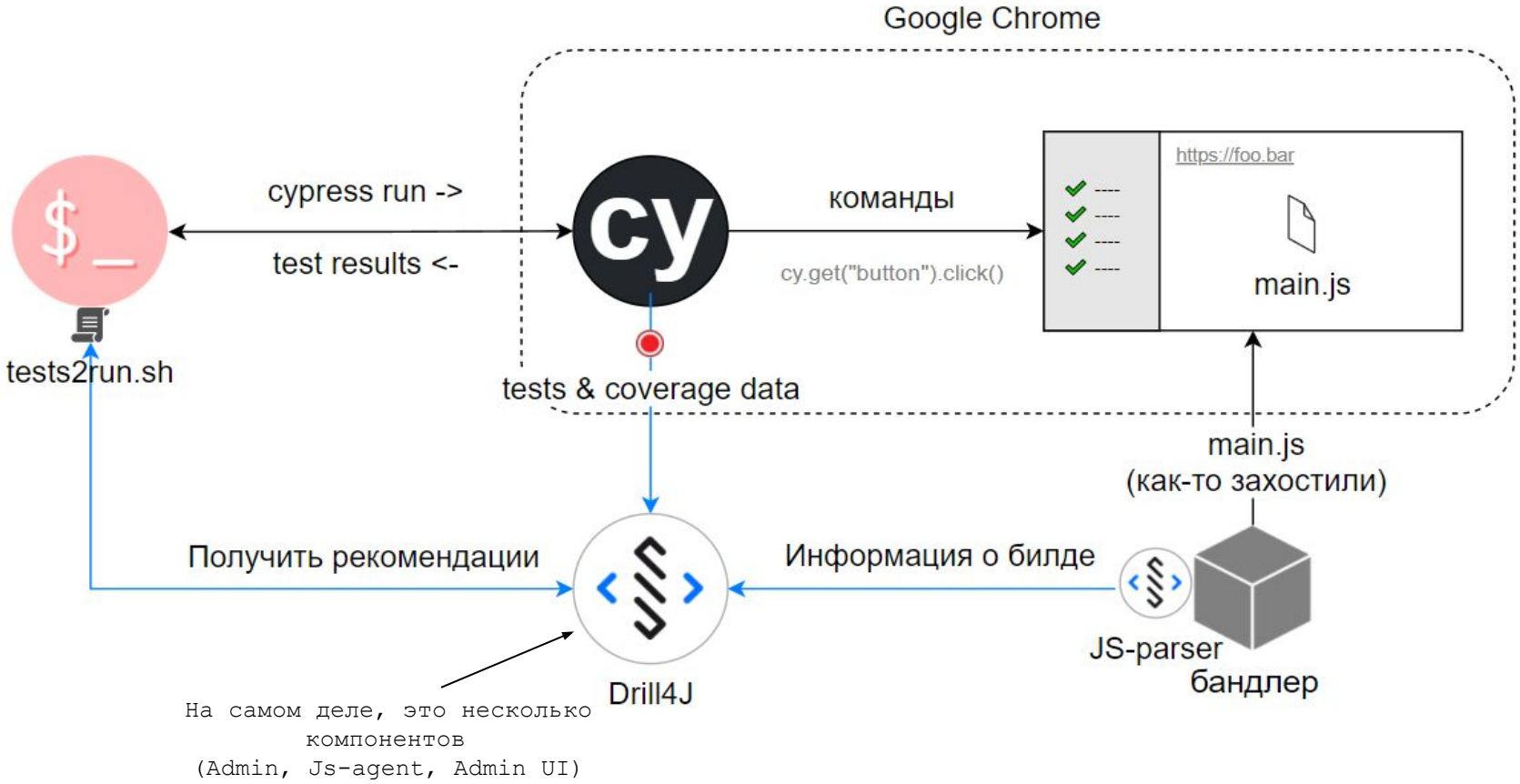
Инструкции/шпаргалка/how-to

https://github.com/Drill4J/heisenbug-moscow-2021-workshop

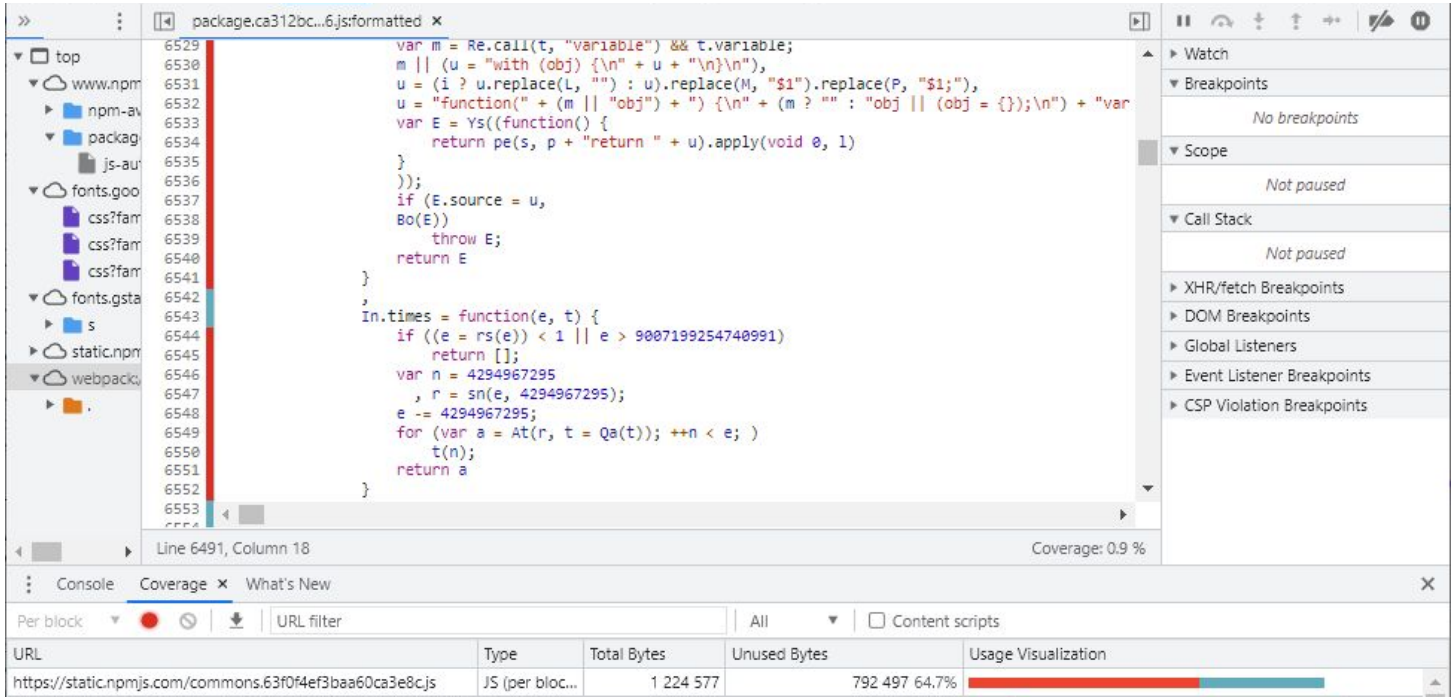
Наш example app

https://github.com/Drill4J/realworld-react-cypress

# Интеграция Drill4J - в картинках



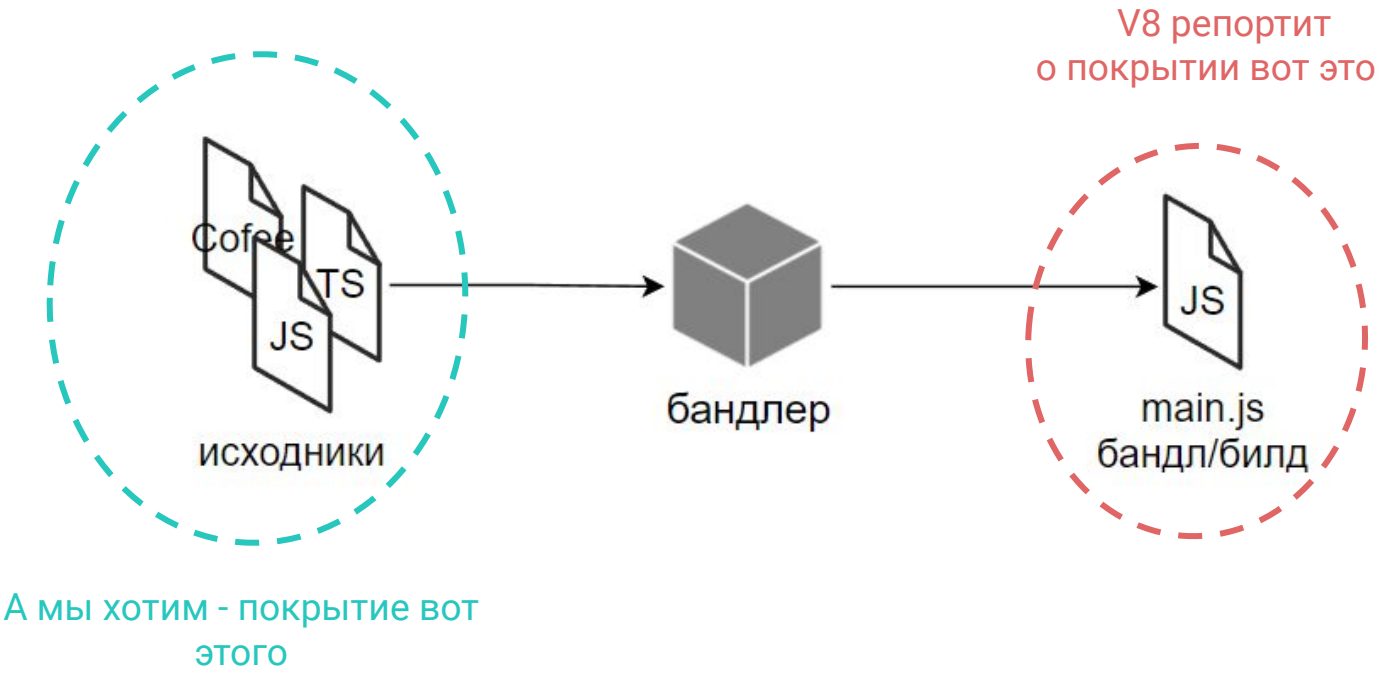
# Что не так с coverage из Chrome/V8?




V8 возвращает **coverage для <u>бандла</u>**, а не для исходников

*>tip: откройте devtools на F12 и попробуйте собрать coverage для любой страницы*

# Мы получаем не совсем то, что нужно!




Drill4J JS-agent отвечает за обратный маппинг

# Итак...

1. Насколько хороши наши тесты?

   ✔️ **Узнать что конкретно делает каждый тест**

2. Какие есть риски?

   ✔️ **Обнаружить измененный, либо новый код**

3. Какие тесты запустить чтобы покрыть риски?

   ✔️ **Найти оптимальный набор тестов**

4. Какая динамика между билдами?

   ✔️ **Применять вышеуказанные действия к каждому билду, на постоянной основе**

# FAQ / Q&A

1. *Влияние на перф приложения?*

   В случае JavaScript все “на совести” движка V8 (это “он” записывает coverage)

   по замерам потребления ресурсов вкладкой Chrome - **оверхед в пределах погрешности***

2. *Можно ли на одном билде гонять одновременно и авто, и ручные тесты?*

   **можно**

3. *Можно ли тестировать одновременно разные билды?*

   **пока нет**, но мы работаем над этим прямо сейчас

4. *Что с ручным тестированием?*

   У нас есть browser-extension для Google Chrome

5. *Поддержка JavaScript для других браузеров/платформ?*

   По-идее любых, где реализован **DevTools** протокол с **Profiler.startPreciseCoverage**

   **Проверено** только с **Chrome** и **Chromium-based** браузерами (нет, Electron и Node.js еще не трогали)

# Ресурсы Drill4J

Вебсайт https://drill4j.github.io

Документация - https://drill4j.github.io/docs/installation/drill-admin

*В процессе развития*

Telegram - https://t.me/drill4j

*Присоединяйтесь, задавайте вопросы, мы любим отвечать :)*

Github - https://github.com/Drill4J

*Ставьте звездочки*

Youtube - https://www.youtube.com/channel/UCJtegUnUHr0bO6icF1CYjKw/videos