



Quality Assurance and Automated Testing

Как устроено автоматизированное тестирование Loginom и
чем оно поможет его развитию



Георгий Киселев

**Frontend Developer,
Loginom Company**



Проблема обеспечения качества

Миллионы строчек кода

- Множество языков программирования
- Десятки дополнительных библиотек
- Богатый и гибкий пользовательский интерфейс



Потребность в автотестах

- Больше функций/кода ▷ Сложнее тестировать вручную
- Больше пользователей ▷ Дороже цена ошибки
- Чаще применение для бизнес-критичных задач ▷
Жестче требования к срокам исправление ошибок ▷
Меньше времени на отладку



График выхода релизов

	Янв	Фев	Мар	Апр	Май	Июн	Июл	Авг	Сен	Окт	Ноя	Дек
2018	6.0.0	6.0.1	6.0.2	6.0.3	6.1.0	6.1.1	6.1.2		6.1.3		6.1.5	
2019				6.2.0		6.2.1	6.2.2			6.2.4		6.2.5
2020		6.3.1				6.3.2		6.3.3		6.3.4	6.4.0	



Почему не сразу?

- Быстрые изменения продукта вначале
- Нестабильный список функционала
- Отсутствие резкого роста сложности продукта
- Терпимые затраты на тестирование



Виды тестирования

- Unit
- Integration
- End-to-end / Functional
- Ручное тестирование
- Canary-тестирование



Подготовительные работы





Критерии выбора инструмента

1. Простота использования
2. Кросбраузерность
3. Javascript/TypeScript для написания тестов
4. Многопоточность
5. Работа браузера без графической оболочки
6. Перезапуск и умное ожидание элементов
7. Drag'n'Drop по координатам
8. Теги
9. Скорость
10. Интерактивная отладка



Из 6-ти кандидатов
выбран TestCafe

TestCafé Studio | DevExpress

Record Configuration: Apple Safari Run Configuration: safari

Feedback

Explorer

- Examples
 - js-examples.js
 - recorded_test_fixture.testcafe
 - A simple test on our example page
 - How to click check boxes and then assert their state
 - How to compare a value in an input
 - How to drag a page element (Drag and drop)
 - How to hover over an element (the mouse cursor is over it)
 - How to implement comparison (Only if the value is not null or undefined)
 - How to press a specified key (Press a key)
 - How to select text in an input (Select text)
 - How to type text into an input (Type text)
 - How to type text into an input, click it, and then assert its value
 - New Test

Welcome x New Test x

New Test

Fixture: Recorded test fixture

http://devexpress.github.io/testcafe/example/

1 T Type Text Selector('#developer-name')

2 C Click Selector('#remote-testing')

3 C Click Selector('#reusing-js-code')

4 C Click Selector('#submit-button')

5 P Press Key meta

6 = Deep Equal ✓

Actual: Selector('#article-header').textContent

Value: "Thank you, Peter!"

Expected: "Thank you, Peter!"

Value: "Thank you, Peter!"

+ Add Timeout + Add Message

Reports

"New Test" test 100% Started at: 07/04/2018 7:28:42 PM Duration: 7s Passed: 0 Failed: 1 Total: 1

Assertions

On-Page Actions

Debug

Browser Actions

Switch Frames

The screenshot shows the TestCafé Studio application window. The top bar includes 'Record Configuration: Apple Safari', 'Run Configuration: safari', and a 'Feedback' button. The main area has tabs for 'Welcome' and 'New Test'. The 'New Test' tab is active, displaying a recorded test fixture for 'Recorded test fixture'. The fixture includes steps like 'Type Text' for '#developer-name', 'Click' for '#remote-testing', and a 'Deep Equal' assertion comparing the actual value of '#article-header' content ('Thank you, Peter!') with the expected value ('"Thank you, Peter!"'). The left sidebar shows an 'Explorer' tree with sections for 'Examples' (including 'js-examples.js' and 'recorded_test_fixture.testcafe' which contains the recorded fixture) and a 'New Test' item. The bottom section shows a 'Reports' summary for the 'New Test' test, indicating 100% completion, a start time of 07/04/2018 7:28:42 PM, a duration of 7s, and results of 0 passed, 1 failed, and 1 total. The right side of the interface contains panels for 'Assertions', 'On-Page Actions', 'Debug', 'Browser Actions', and 'Switch Frames'.



Подготовительные работы

При встраивании автотестов было внесено множество изменений в код платформы и бизнес-процесс разработки с целью:

1. Минимизации доработок, требуемых для тестов
2. Стабильности прохождения тестов
3. Простоте локализации ошибок



Пример проблемы

В Loginom значительная часть кода генерируется автоматические, но для автотестирования необходимо формирование осмысленных имен для элементов управления – селекторов.

```
test("Создание папки с именем, которое дублирует имя уже существующей папки", async t => {
    /** Имя каталога с дублирующим именем */
    const createDoubleNameFolder = "doubleNameFolder";
    /** Имя рабочего каталога, в котором будет проходить проверка */
    const workFolder = "duplicationFolderNames";
    /** Кнопка создания каталога */
    const sBtnCreateFolder = "[data-tid$='FileStorageForm;btnCreateDirectory']";
```

Пример кода автотеста

_filestorage_actions.ts 1 of 17 problems

селектор следует заменить на соответствующий из файла bg/selectors.ts. Стока №665 (common-selectors.ts)

```
await EnterWorkFolder(workFolder); // Создание и вход в рабочий каталог
await t
    .click(S(sBtnCreateFolder)) // Клик по кнопке "Создать каталог..." панели инструментов
    .typeText(s.msgBox.MsgPromtInput, createDoubleNameFolder, { replace: true }) // Ввод имени папки
    .click(S(s.msgBox.Ok)) // Клик на кнопку "OK"
    .click(S(s.filestorage.toolbar.BtnCreateFolder)) // Клик по кнопке "Создать каталог"
    .typeText(s.msgBox.MsgPromtInput, createDoubleNameFolder, { replace: true }) // Ввод имени папки
    .click(S(s.msgBox.Ok)); // Клик на кнопку "OK"
await CheckNotifyMsg(FolderExistsMsg); // Проверка наличия сообщения об ошибке
});
```

Сценарий

^ < > Пакеты Package1 Модуль1 Сценарий

Компоненты

Фильтрация

Импорт
1С Запрос
База данных
Текстовый файл
Deductor Warehouse
Excel файл
Loginom Data файл
XML файл

Трансформация
Группировка
Дата и время
Дополнение данных
Замена
Калькулятор
Кросс-таблица
Объединение
Параметры полей
Разгруппировка
Свертка столбцов
Скользящее окно
Слияние
Соединение
Сортировка
Фильтр строк

Управление
Выполнение узла
Подмодель
Узел-ссылка
Условие
Цикл

Исследование
Автокорреляция
Дубликаты и противоречия

Текстовый файл

Пример выполнения
автотеста

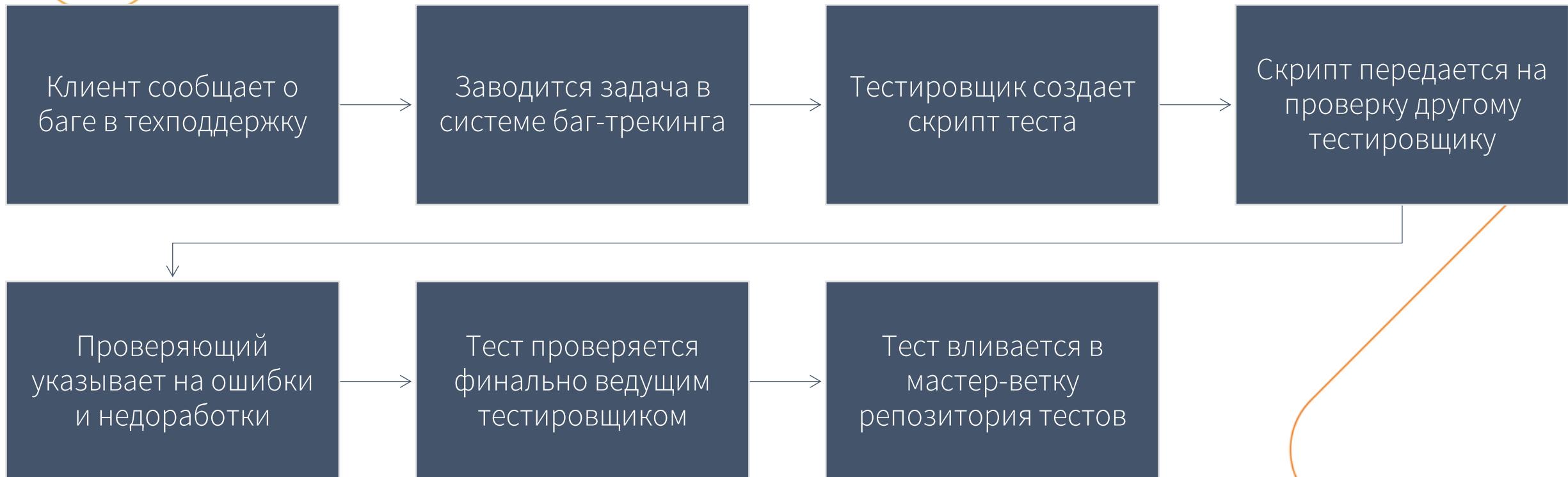


Два вида тестов

1. Приемочные тесты при разработке новых функций
2. Тесты для воспроизведения известных багов



Бизнес-процесс исправления багов, обнаруженных клиентами





Включение в сборщик Loginom

1. Написанный тест добавляется в Continuous Integration разработки Loginom
2. После каждого изменения созданный тест прогоняется, чтобы убедиться, что ошибка больше не возникнет

- E
- H
- B
- L
- D
- I
- A
- C
- G
- P
- S
- ⚙️

```
v dev/loginom!1823#note_130332 - Используемые производные компоненты  
перемещаются, при этом на исходном workflow остаются зависимые узлы, для  
которых эти компоненты не доступны (Error => отменить операцию)  
v dev/loginom!1823#note_130332 - Используемые производные компоненты  
перемещаются, при этом на исходном workflow остаются зависимые узлы, для  
которых эти компоненты не доступны (Error => Повторить принудительно)  
v dev/loginom!1823#note_130483 (вырезание - удаление - вставка в текущий  
модуль)  
v dev/loginom!1823#note_130483 (копирование - удаление - вставка в текущий  
модуль)  
v dev/loginom!1823#note_130483 (вырезание - удаление - вставка в другой  
модуль)  
v dev/loginom!1823#note_130483 (копирование - удаление - вставка в другой  
модуль)  
v dev/loginom!1823#note_130534  
v dev/loginom!1823#note_132231  
v dev/loginom!1823#note_131020 dev/loginom#7396  
v dev/loginom!1823#note_130952 dev/loginom#4843  
v dev/loginom!1823#note_131465 dev/loginom#8108
```

27 passed (31m 33s)

[11:05:37] Finished 'tests' after 32 min

[11:05:37] Starting 'after'...

The "src" option from the configuration file will be ignored.

Running tests in:

- Chrome 84.0.4147.89 / Windows 8.1

Действия после тестов

v Удаляем временный каталог текущего запуска тестов вместе со всеми его
вложенными каталогами/файлами

1 passed (42s)

[11:06:37] Finished 'after' after 1 min

[11:06:37] Finished 'test' after 35 min



e2e-tests-test

Retry

Duration: 35 minutes 44 seconds

Timeout: 16h (from project)



Runner: tests-runner (#35)

Tags: windows chrome firefox

Job artifacts

The artifacts will be removed in 1 week

Keep

Download

Browse

Commit ed846a04 in !73

Вернул старый пакет. Также проходит проверка dev/loginom#8108

Pipeline #38292 for discussions-

Время выполнения всех
тестов – 16 часов



Итоги внедрения автотестов

1. Управляемость процесса улучшения качеством
2. Стабильное снижение количества ошибок
3. Исключение регрессионных ошибок
4. Новые фичи без ухудшения качества
5. Ускорение развития платформы



Жду ваших вопросов

