ВDD в розовом цвете













Анна Чернышева

Test Automation Lead

Web Testing

facebook.com/anna.chernyshova.79



Continuous Delivery































































































Разработчик автотестов





Ручное тестирование Регрессия PRODUCT



Unit-тесты





Черный ящик для всех. Никто не знал, что именно покрыто unit-тестами, а что не покрыто. A



Автотесты

Unit-тесты



Не все проекты имели автотесты Дублировалось ручным и unit тестированием

> Черный ящик для всех. Никто не знал, что именно покрыто unit-тестами, а что не покрыто.





I

Приемка и регресс

Проходят долго Команда не знает, как и что именно тестируется

Автотесты

Unit-тесты

Черный ящик для всех. Никто не знал, что именно покрыто unit-тестами, а что не покрыто.

Не все проекты имели автотесты Дублировалось ручным и unit тестированием













А как релизить **каждую неделю** и быть уверенным в качестве?



















Как же достичь автоматического регресса во всех продуктах?

















Найти много разработчиков автотестов





.











Научить тестировщиков писать автотесты







Должен быть ИНСТРУМЕНТ, УПРОЩАЮЩИЙ НАПИСАНИЕ АВТОТЕСТОВ для непрограммистов

Автотесты могут писать ВСЕ в команде

• Разработка автотестов НЕ ОТНИМАЕТ много времени









• Тестировщик совместно с разработчиком оценивает покрытие unit-тестами

• Тестировщик или аналитик, который не умеет программировать, сам пишет автотесты на UI

Максимальное время приемки - 30 минут









Структура сценария: Поиск репозиториев в github по ключевому слову И в поле "Поиск" введено значение "<searchQuery>" И выполнено нажатие на клавиатуре "Enter" Тогда страница "Репозитории" загрузилась

Примеры: searchQuery alfalab



Пусть совершен переход на страницу "GitHub" по ссылке "gitHubMainPage" Korдa выполнен GET запрос на URL "{repositorySearchUrl}<searchQuery>". Тогда список репозиториев на странице соответствует ответу сервиса





TESTING 47 DESIGN













Automated testing



Технические решения





Akita - BDD библиотека шагов для автоматизации тестирования







Akita - BDD библиотека шагов втоматизации тестирования ДЛЯ









Аналоги







selenium-cucumber-java

Плюсы:

много реализаций различных пользовательских действий

Минусы:

- шаги на английском
- нет интергационных шагов



Beryllium

Плюсы:

хорошая реализация шагов для тестирования RESTAPI

Минусы:

• шаги на английском

нет шагов для тестирования UI





На чем построили?

Selenide CONCISE UL TESTS IN JAVA

cucumber









•	Li	Leakita-testing-template [TEMPLATE (CHANGE MEIII)] ~/work/akita-test		
		🗖 .g	radle	
	►	🗅 .id	lea	
		🗅 cla	isses	
	►	🗖 gradle		
	►	🖿 lib	S	
	▼	🗅 sr	c	
		V D	anain	
		•	🗖 java	
			entities	
			🔻 🖻 pages	
			🕒 🚡 LoginPage	
			restbodies	
			🔤 templateBody.json	
			🔻 🖻 steps	
			🕒 🚡 TemplateSteps	
			utils	
		►	resources	
		V	test	
		•	ne sources	
			🔻 🖻 features	
			📁 Детальная информация о пополнении депозита.fea	
		🚯 .g	itignore	
		🕑 bu	ild.gradle	
		🖬 gr	adle.properties	
		📑 gr	adlew	
		🗐 gr	adlew.bat	
		🖻 re	adMe.md	
		(se	ttings_gradle	

ng-template



Шаблон проекта

ature





Простой сценарий

Сценарий: Вход в приложение под админом Когда пользователь "admin" ввел логин и пароль Тогда страница "Расходы и Доходы" загрузилась И поле "Расходы" кликабельно





Дано совершен переход на страницу "Авторизация" по ссылке "appUrl"



Простои шаг

Сценарий: Вход в приложение под админом Когда пользователь "admin" ввел логин и пароль Тогда страница "Расходы и Доходы" загрузилась И поле "Расходы" кликабельно

Дано совершен переход на страницу "Авторизация" по ссылке "appUrl"





Что внутри?

akitaScenario_write(" url = " + address); getWebDriver().get(address); loadPage(pageName);



@И("^совершен переход на страницу \"(*)\" по ссылке \"(*)\"\$") public void goToSelectedPageByLink(String pageName, String url){ String address = loadProperty(url, resolveVars(url));





Параметры Шага

akitaScenario.write(" url = " + address); getWebDriver().get(address); loadPage(pageName);



@И("^совершен переход на страницу \"(*)\" по ссылке \"(*)\"\$") public void goToSelectedPageByLink(String pageName, String url){ String address = loadProperty(url0rName, resolveVars(url));



Page Object




Интернет-банк

Вход Регистрация

Логин

Пароль

Восстановить логин и пароль











Определение страницы

@Name("Авторизация")

@FindBy(id = "input_login") @Name("Логин")

@FindBy(id = "input_pass") **@Name("Пароль")**

@FindBy(id = "submit") @Name("Войти")



- public class LoginPage extends AkitaPage {

 - public SelenideElement loginField;
 - public SelenideElement passwordField;
 - public SelenideElement submitButton;



Интернет-банк

Вход Регистрация

Логин

2328128

Пароль

Восстановить логин и пароль

Войти







Дополнительные элементы

@Name("Авторизация") public class LoginPage extends AkitaPage {

@FindBy(id = "input_login") **@Name("**Логин") public SelenideElement loginField;

@FindBy(id = "input pass") **@Name("Пароль")**

@FindBy(id = "submit") @Name("Войти")

@FindBy(id = "validation_error") @Name("Ошибка валидации") @Optional



- public SelenideElement passwordField;
- public SelenideElement submitButton;
- public SelenideElement validationError;





Дополнительные элементы

@Name("Авторизация") public class LoginPage extends AkitaPage {

@FindBy(id = "input_login") @Name("Логин") public SelenideElement loginField;

@FindBy(id = "input pass") @Name("Пароль")

@FindBy(id = "submit") @Name("Войти")

@FindBy(id = "validation_error") @Name("Ошибка валидации") @Optional



- public SelenideElement passwordField;
- public SelenideElement submitButton;
- public SelenideElement validationError;





Обращение к странице из шага

Расходыи,





ДОХ	(ОДЬ)				
ОКТЯБРЬ	НОЯБРЬ	ДЕКАБРЬ	2018 ЯНВАРЬ	ФЕВРАЛЬ	МАРТ	
ļ	Доходы 0₽					
		0 ₽ из 77 ₽	Выбор о за мес о за год	периода		
		0 ₽ из 15 ₽	Счета ЕЕ ₽ · ·	3699 602.85 P		



Обращение к странице из шага

Сценарий: Вход в приложение под админом Когда пользователь "admin" ввел логин и пароль Тогда страница "Расходы и Доходы" загрузилась И поле "Расходы" кликабельно



Допустим совершен переход на страницу "Страница входа"



Обращение к странице из шага

Сценарий: Вход в приложение под админом Когда пользователь "admin" ввел логин и пароль Тогда страница "Расходы и Доходы" загрузилась И поле "Расходы" кликабельно



Допустим совершен переход на страницу "Страница входа"



А как передать данные из одного шага в другой?





A

Сценарий: Отображение суммы расходов Допустим совершен переход на страницу "Расходы и Доходы" И значение поля "Расходы" сохранено в переменную "expenses" Когда выполнено нажатие на поле "Расходы" Тогда страница "Информация о расходах" загрузилась из переменной "expenses"

- И значение поля "Общая сумма расходов" совпадает со значением





И значение поля "Расходы" сохранено в переменную "expenses"

/**

* Сохранение значения элемента в переменную */

@Korдa("^значение (?:элемента|поля) \"(*)\" сохранено в переменную \"(*)\"")
public void storeElementValueInVariable(String elementName, String variableName){
 akitaScenario.setVar(variableName,
 akitaScenario.getCurrentPage().getAnyElementText(elementName));





И значение поля "Pacxoды" сохранено в переменную "expenses"

/**

* Сохранение значения элемента в переменную */ @Когда("^значение (?:элемента|поля) \"(*)\" сохранено в переменную \"(*)\"") public void storeElementValueInVariable(String elementName, String variableName){ akitaScenario.setVar(variableName, akitaScenario.getCurrentPage().getAnyElementText(elementName));



И значение поля "Общая сумма расходов" совпадает со значением из переменной "expenses"

```
/**
```

* Проверка того, что значение из поля совпадает со значением заданной переменной из хранилища */

@Тогда("^значение (?:поля элемента) \"(*)\" совпадает со значением из переменной \"(*)\"\$") public void compareFieldAndVariable(String elementName, String variableName) { String actualValue = akitaScenario.getCurrentPage().getAnyElementText(elementName); String expectedValue = akitaScenario.getVar(variableName).toString(); assertThat(String.format("Значение поля [%s] не совпадает со значением из переменной [%s]", elementName, variableName), actualValue, equalTo(expectedValue));















И значение поля "Общая сумма расходов" совпадает со значением из переменной "expenses"

/**

* Проверка того, что значение из поля совпадает со значением заданной переменной из хранилища */

public void compareFieldAndVariable(String elementName, String variableName) { String expectedValue = akitaScenario.getVar(variableName).toString(); assertThat(String.format("Значение поля [%s] не совпадает со значением из переменной [%s]",



@Тогда("^значение (?:поля|элемента) \"(*)\" совпадает со значением из переменной \"(*)\"\$") String actualValue = akitaScenario.getCurrentPage().getAnyElementText(elementName);

elementName, variableName), actualValue, equalTo(expectedValue));











Решение выражения в шаге











Структура сценария: Добавление наличного расхода Допустим совершен переход на страницу "Расходы и Доходы" Когда выполнено нажатие на "Записать наличный расход" Когда в поле "Сумма" введено значение "<sum>" И выполнено нажатие на "Добавить" И верно, что expenses2=expenses1+sum

Примеры: sum 100



- И значение поля "Pacxоды" сохранено в переменную "expenses1"
- Тогда значение поля "Pacxоды" сохранено в переменную "expenses2"





Структура сценария: Добавление наличного расхода Когда выполнено нажатие на "Записать наличный расход" Когда в поле "Сумма" введено значение "<sum>" И выполнено нажатие на "Добавить" И верно, что expenses2=expenses1+sum

Примеры: SUM 100



```
Допустим совершен переход на страницу "Расходы и Доходы"
И значение поля "Pacxoды" сохранено в переменную "expenses1"
Тогда значение поля "Pacxoды" сохранено в переменную "expenses2"
```







Решение выражения в шаге

/**

- * Проверка выражения на истинность * Например, string1.equals(string2) * OR string.equals("string")
- * Любое Java-выражение, возвращающие boolean */
- @Тогда("^верно, что \"([^\"]*)\"\$") public void expressionExpression(String expression) {



```
akitaScenario.getVars().evaluate("assert(" + expression + ")");
```







Списки элементов





A

Расходы 4 102 ₽			доходы 0 ₽	
Прочие расходы				4 100 ₽ из 50 ₽
A3C				2 ₽
Дом, квартира, д	ача			0 ₽ из 1 200 ₽
Pac	ходы Д	Јоходы		
(+)	Добавитн	ь категорию		
	Meow			
	test10trar	nsactions		
	Автотран	спорт		

	Расходы 4 102 ₽	До	оходы 0₽	
Пр	очие расходы		4 100 ₽ из 50 ₽	
	11 июня в 23:59 Трата наличными		1 000,00 ₽	
	11 июня в 23:59 Трата наличными		1 000,00 ₽	
	10 июня в 23:59 Трата наличными		2 000,00 ₽	
		рпврпвапва	100,00 ₽	
	Кат	тегория		
	П	рочие расход		
Î		Бизнес, прое	КТЫ	
		Благотворит	ельность	
		Бытовая тех	ника	
		Гипермаркет		
		Долги, креди	ТЫ	





.

Выбор элемента из списка

Сценарий: Отображение списка транзакций расходов Когда выбран случайный элемент в списке "Расходы" Тогда список "Транзакции расходов" отображается на странице

> @FindBy(className = "list_element") @Name("Pacxoды") public List<SelenideElement> expencesList;



Допустим совершен переход на страницу "Расходы и Доходы" пользователем "admin"



Выбор элемента из списка

Сценарий: Отображение списка транзакций расходов Допустим совершен переход на страницу "Расходы и Доходы" пользователем "admin" Когда выбран случайный элемент в списке "Расходы" Тогда список "Транзакции расходов" отображается на странице

Когда в списке "Расходы" выбран элемент содержащий текст "АЗС"

Когда в списке "Расходы" выбран 1-й элемент





Проверка элементов списка

Сценарий: Отображение списка транзакций расходов Допустим совершен переход на страницу "Расходы и Доходы" пользователем "admin" Когда выбран случайный элемент в списке "Расходы"

Тогда список "Транзакции расходов" отображается на странице Тогда список "Транзакции расходов" содержит текст "Списание процентов" Тогда список "Транзакции расходов" совпадает со списком "ListFromVar" Тогда список "Транзакции расходов" состоит из элементов из таблицы

Оплата страхового полиса Списание процентов





Текущая дата









2017-12-09

09 january 2017

Текущая дата

09 12 2017

09.12.2017



09-12-2017

09 декабря

09/12/2017

09 декабря 2017

12-09-2017



	Расходыи	1 ДОХОДЫ		Эапись о наличных расходах	
АПРЕЛЬ	МАЙ ИЮНЬ ИЮЛЬ АВГУСТ	СЕНТЯБРЬ ОКТЯБРЬ НОЯБРЬ	ДЕКАБРІ	Сумма	
	Расходы 2 303 ₽	Доходы 0 ₽		Категория	
				Прочие расходы 🔻	
	Прочие расходы	2 233 ₽	ВЫбО о за ме о за го	Заметка Корм для кота	
	Семья и дети	70 ₽	Счета	Когда	
	Управление категориями		EE ₽ 851 9	✓ Из снятых наличных	







/** */

Структура сценария: Попап успеха при добавление расхода Допустим совершен переход на страницу "Расходы и Доходы" Когда выполнено нажатие на "Записать наличный расход" Когда в поле "Сумма" введено значение "<sum>" И элемент "Когда" заполняется текущей датой в формате "dd.MMMM. уууу" И выполнено нажатие на "Добавить" Тогда элемент "Расход добавлен" отображается на странице

Примеры: SUM |100|

```
* Ввод в поле текущей даты в заданном формате
* При неверном формате, используется dd.MM.yyyy
```





Работа с клавиатурой



65



A

Сценарий: Вход в приложение под админом Дано совершен переход на страницу "Авторизация" по ссылке "appUrl" Когда пользователь "admin" ввел логин и пароль И выполнено нажатие на клавиатуре "Enter" Тогда страница "Расходы и Доходы" загрузилась







Сценарий: Вход в приложение под админом Дано совершен переход на страницу "Авторизация" по ссылке "appUrl" Когда пользователь "admin" ввел логин и пароль И выполнено нажатие на клавиатуре "Enter" Тогда страница "Расходы и Доходы" загрузилась

> alt Enter

И выполнено нажатие на сочетание клавиш из таблицы







Тесты на интеграцию с REST





A

Тесты на интеграцию с REST

Структура сценария: Отображение списка расходов Допустим совершен переход на страницу "Pacxoды и Доходы" пользователем "<user>" Когда выполнен GET запрос на URL "{expenses.url}" с headers и parameters из таблицы. Полученный ответ сохранен в переменную "<expensesResponse>" [type |name |value | header|applicationId|{app.id}] header|customerId |<user> | Тогда список расходов на странице соответствует списку из "expensesResponse"

Примеры: user userAdmin





Тесты на интеграцию с REST

Структура сценария: Отображение списка расходов Допустим совершен переход на страницу "Расходы и Доходы" пользователем "<user>" Когда выполнен GET запрос на URL "{expenses.url}" с headers и parameters из таблицы. Полученный ответ сохранен в переменную "<expensesResponse>" type name value header applicationId {app.id} header customerId <user> Тогда список расходов на странице соответствует списку из "expensesResponse"

Примеры: user userAdmin





Тесты на интеграцию с REST ацаб

/**

- * Отправка GET/POST запроса по заданному URL с заданными параметрами. * И в URL, и в значениях в таблице можно использовать переменные как из * application.properties, так и из хранилища переменных из AkitaScenario. * Для этого достаточно заключить переменные в фигурные скобки, например: * http://{hostname}?user={user.name}.

- * Content-Type при необходимости должен быть указан в качестве header. */

@И("^выполнен (GET|POST) запрос на URL \"([^\"]*)\". Полученный ответ сохранен в переменную \"([^\"]*)\"\$")

Ожидается код ответа: (\\d+)\$")

Полученный ответ сохранен в переменную \"([^\"]*)\"\$")



- @И("^выполнен (GET|POST) запрос на URL \"([^\"]*)\" с headers и parameters из таблицы.
- @И("^выполнен (GET|POST) запрос на URL \"([^\"]*)\" с headers и parameters из таблицы.





Тестовые данные





A 72
Тестовые данные

application.properties

waitingAppearTimeout=20000

mainPageUrl=https://alfabank.ru devapi.plans=http://devapi/finhealth-api/plans

userWithManyTransactions.login=123467 userWithManyTransactions.password=000000 userWithManyTransactions.cus=AAAJJJ





#language: ru Функционал: Транзакции расходов

Структура сценария: Отображение списка транзакций расходов в текущем месяце Цопустим совершен переход на страницу "Расходы и Доходы" пользователем "<user>" И элемент "Дата" заполняется текущей датой в формате "dd.MM.yyyy" Когда выбран случайный элемент в списке "Расходы" Тогда список "Транзакции расходов" отображается на странице Когда значение поля "Категория расходов" сохранено в переменную "expenseCateory" И выполнен GET запрос на URL "{expensesUrl}" с headers и parameters из таблицы. Полученный ответ сохранен в переменную "expensesResponse" type name value header applicationId {appID} header customerId <user> Тогда названия и суммы транзакций расходов для категории "expenseCateory" соответствуют транзакциям из Json ответа "expensesResponse"

Примеры: user userAdmin













Запуск тестов







<u>A</u> 75

./gradlew clean generateCucumberReport -Dbrowser=chrome









./gradlew clean generateCucumberReport -Dbrowser=chrome

./gradlew clean gCR -Dbrowser=chrome -Ptag=@smoke







.

./gradlew clean generateCucumberReport -Dbrowser=chrome

./gradlew clean gCR -Dbrowser=chrome -Ptag=@smoke

./gradlew clean gCR -Dbrowser=mobile -Ddevice=iPhone 7







7



Преимущества, которые дает библиотека

- простой АРІ на русском языке
- гибкая конфигурация
- низкий порог входа
- поддержка различных браузеров, в том числе и мобильного
- 60 готовых шагов для ваших тестов



реализация шагов для тестирования интеграции ui и rest













Плагин для параллелизации тестов











Плагин для параллелизации тестовПлагин для генерации отчётов







Плагин для параллелизации тестов







cucumberparallel-testgradle-plugin

раннеры генерируются в папку build проекта

генерирует раннер для каждого .feature файла в директории test/resources





Отчет о тестировании





85



Features Statistics

The following graphs show passing and failing statistics for features



Feature	Passed	Failed
LIGHTS-161 Пополнение депозита с сберегательных счетов	14	0
LIGHTS-162 Пополнение депозита сумма с копейками	14	0
LIGHTS-34 При открытии подставлять минимальную сумму пополнения	9	0
LIGHTS-442 Проверка отображения ошибок pipe	12	0
LIGHTS-56 На странице пополнения вывести шапку депозита вверху	6	0
LIGHTS-6 Отображение страницы пополнения депозита без правой части	12	0





Feature Report

	Steps				Scenarios			Features			
Feature	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
LIGHTS-161 Пополнение депозита с сберегательных счетов	14	0	0	0	0	14	1	0	1	23s 842ms	Passed
											\bigcirc
Функционал LIGHTS-161 Пополнение депозита с сберегательны	ых счетов										
Структура сценария Пополнение депозита с накопительног	о счета 🗸	•									
Hooks >											

Steps V Когда ref депозита с названием "Потенциал" и валютой "usd" для head name value type header applicationId test ALOKOT header customerId OPENED state body Output 1 Output 2 MOCØAQ5ALOKØT10056I0 Тогда совершен переход на страницу "Страница входа" по прямой со Output 1 Когда я залогинился как пользователь "ALOK0T" Тогда страница "Пополнение депозита" загрузилась Тогда элемент "Сумма" отображается на странице Тогда элемент "С какого счета списать" отображается на странице Тогда элемент "Пополнить" отображается на странице И элемент "Заголовок" содержит значение "Пополнить депозит" Когда установлено значение "1000" в поле "Сумма"

	ders и	parameters	из таблиць	сохранен	в переменную	o "ref
--	--------	------------	------------	----------	--------------	--------

сылке = "https://onlineft7.alfaintra.net/deposits-refilling/deposits/{ref}/refill"	2s 874ms
	6s 452ms
	5s 142ms
	068ms
	056ms
	060ms
	112ms
	473ms



3s 166ms

87

Генерация документации





8





Cukedoctor



Плюсы

- удобная генерация документации
- логирование дополнительной информации в отчете

Минусы

- для банка не всегда годится подход с логированием всех тестовых данных
- Maven-плагин





documentation-gradle-plugin

работает как gradle-таск, который запускается после того как отработал таск test. c помощью asciidoctor генерит html, a bdd-тесты вставляются как сниппеты.

Минус: • тестировщик при появлении новых фича-файлов должен прописать путь до него в файле index в папке с документацией, как сниппет







Table of Contents

Алерт

Проверка перехода на страницу заполнения заявки на выпуск КЭП Проверка наименования компании в тексте при смене компании Переход по вкладкам в Налоги КБ Доступность вкладок для нажатия Автоматическое заполнение реквизитов организации данными из EQ

Отображение реквизитов организации на основе ответа от corp-taxes-organizationsapi/{organizationId}#GET

Автоматическое заполнение информации о владельце подписи данными из EQ

Отображение информации о владельце подписи на основе ответа сервиса corp-taxes-profilesapi/{profileId}/personal#GET

Отправка заявки на получение квалифицированной электронной подписи (КЭП) в Калугу-Астрал

Отправка заявки с данными организации и владельца подписи для получения КЭП через вызов сервиса corp-taxes-signatures-api/ **#POST**

Отображение информации о владельце подписи на основе ответа сервиса corp-taxes-profiles-api/{profileId}/personal#GET

- И далее я перешел на вкладку "Документооборот"
- Тогда страница "Создание подписи" загрузилась
- И я открыл блок "Владелец подписи"
- в переменную "signatureOwner"

Отправка заявки с данными организации и владельца подписи для получения КЭП через вызов сервиса corp-taxes-signatures-api/ #POST

- И на странице "Создание подписи" заполнены следующие поля:

Наименование

ИНН

• Дано открыта страница Получение КЭП для пользователя "userUniversal"

• И я нажал на кнопку "Получить подпись" на странице "Документооборот"

• И был вызван сервис с header-параметром profileId = "userUniversal.acus". Результат работы сервиса сохранён

• Тогда данные о "владельце подписи" совпадают с данными в "signatureOwner"

Отправка заявки на получение квалифицированной электронной подписи (КЭП) в Калугу-Астрал

• Дано открыта страница Получение КЭП для пользователя "userUniversal"



Масштабируемость









• единый инструмент для 20+ команд









единый инструмент для 20+ команд

базовая библиотека покрывает 80% потребностей









единый инструмент для 20+ команд базовая библиотека покрывает 80% потребностей первичный onboarding тестировщика - 1 день









единый инструмент для 20+ команд базовая библиотека покрывает 80%

первичный onboarding тестировщика - 1 день

свободная ориентация в инструменте - 2 недели



базовая библиотека покрывает 80% потребностей







приемка + регресс < 30 минут</p>





















которые НЕ УМЕЮТ ПРОГРАММИРОВАТЬ

▶ автотесты могут писать тестировщики или аналитики,

приемка + регресс < 30 минут</p>









• несколько поставок за НЕДЕЛЬНЫЙ спринт

которые НЕ УМЕЮТ ПРОГРАММИРОВАТЬ

▶ автотесты могут писать тестировщики или аналитики,

приемка + регресс < 30 минут</p>



100





.











бизнес хочет BDD с живой документацией на русском











бизнес хочет BDD с живой документацией на русском

в команде нет выделенной позиции тестировщика









бизнес хочет BDD с живой документацией на русском

в команде нет выделенной позиции тестировщика

важно обеспечение единого стека технологий между

большим количеством scrum команд







Скорее всего не подойдет, если











Скорее всего не подойдет, если

бизнесу не нужен BDD









Скорее всего не подойдет, если

бизнесу не нужен BDD

• автоматизацией тестирования занимается отдельная

команда















108




Клонировать git-репозиторий шаблона

github.com/alfa-laboratory/akita-testing-template











Клонировать git-репозиторий шаблона

Придумать тестовый сценарий





github.com/alfa-laboratory/akita-testing-template







Клонировать git-репозиторий шаблона

Придумать тестовый сценарий

• Автоматизировать его



github.com/alfa-laboratory/akita-testing-template







GitHub template

> Telegram @akitaQA



github.com/alfa-laboratory/akita

github.com/alfa-laboratory/akita-testing-



