

**Проект 1: Регрессионное тестирование** по готовым тест-кейсам приложения Mesto — интерактивной веб-страницей, куда можно добавлять фотографии, удалять их и ставить лайки.

1. [Проект в TMS:](#)

2. [Баг-репорты в Яндекс Трекере:](#)

3. Выводы о работе:

Сколько времени ушло на тестирование?

- в общей сложности, со всеми корректировками, порядка 5 часов

Какое тестовое окружение у тебя было?

- Окружение1: Браузер: СБраузер: Chrome v. 108.0.5359.125, ОС: Windows 10 v. 21H2

- Окружение2: Браузер: Chrome v. 109.0.5414.75, ОС: Windows 10 Pro v. 1909

- Окружение3: ОС: Android 10, смартфон: Honor 10i YRY-LXIT. Браузер: Chrome v. 108.0.5359.128

Удалось ли найти баги? Если да, то сколько?

- удалось. 12 багов

Все ли баги помешают пользователю взаимодействовать с приложением?

- нет не все, есть баги с низким или незначительным приоритетом, они не влияют на функциональность, но влияют на общий имидж приложения

Были ли сложные или непонятные тест-кейсы?

Да, долго не могла определиться какой статус присвоить тест-кейсам "При пустом поле "Занятие"/"Имя" кнопка "Сохранить" не активна" и "Сообщение об ошибке при вводе числа символов больше допустимого для поля "Занятие"/"Имя"

Рекомендуешь ли ты выпускать приложение в релиз в таком виде? Почему?

- Нет не рекомендую.

1. В приложении отсутствуют важные блоки - регистрация и авторизация, профиль нужно сначала зарегистрироваться и авторизоваться в нем, чтобы иметь возможность его изменять

2. В приложении есть критические баги, присутствие которых сводит на нет весь смысл существования проекта. Например:

- После внесения новых данных в поля "Имя" и "Занятие" в форме редактирования профиля, информация в профиле на главной странице не обновляется
- В форме редактирования профиля отсутствует возможность изменить фото
- На странице отсутствует кнопка "+" для добавления Нового места
- На карточках места отсутствует кнопка "урна" для удаления карточки

4. Вопросы на рассуждение:

- Представь, что ваша команда планирует очередной релиз приложения Яндекс Самокат. Выбери одну линейную и одну гибкую модель и опиши, как будут проходить разработка и тестирование. За основу возьми следующую функциональность: поиск ближайшей станции зарядки, поиск самоката по сигнализации.

#### **-Инкрементная модель (линейная)**

При работе по инкрементной модели наша команда Яндекс.Самокат на этапе начального планирования соберет все требования к проекту и разделит их на этапы внедрения функционала.

В текущем этапе мы внедряем функционал: поиск ближайшей станции зарядки и поиск самоката по сигнализации.

Реализация внедрения этапа разделена на несколько циклов, в каждом из которых внедряемый на текущем этапе функционал улучшается.

Цикл внедрения функционала состоит из следующих этапов:

- Сбор требований и создание спецификации к функционалу
- Проектирование - составляем подробный план работ, дизайн, порядок тестирования и приемки, требования к документированию процесса и результату разработки, перечень действий для запуска продукта
- Разработка - разработчики пишут код согласно требованиям
- Внедрение (релиз) - функционал внедряется в приложение
- Тестирование внедренного функционала
- Отладка внедренного функционала

В первом цикле мы внедрим основные функции, которые позволят запустить в Яндекс Самокате функционал “поиск ближайшей станции зарядки и поиск самоката по сигнализации” в короткие сроки. Т.о. пользователи смогут уже пользоваться этими функциями, а мы постепенно, в следующих циклах, опираясь на отзывы пользователей, совершенствовать этот функционал, добавляя новые функции до тех пор, пока не будут выполнены все требования к внедренному функционалу,

#### **- Scrum модель (гибкая)**

Основой Scrum является отрезок времени 1-4 недели, в течении которого выполняется работа над продуктом - этот отрезок называется Спринт (Sprint). Длительность Спринта всегда одинакова в течение жизни проекта. По истечении Спринта должна быть получена новая рабочая версия продукта.

В начале проекта команда вместе с заказчиком составляет Бэклог Продукта (Product Backlog) это список всех задач, которые нужно выполнить в течение проекта. Этот список построен по приоритетности задач и может изменяться и дополняться.

В течение спринта команда проводит короткие встречи каждый день, чтобы проделанную работу, список задач, которые сотрудник должен выполнить в течение дня, обсудить затруднения, возникшие в ходе работы.

Этапы спринта:

Планирование спринта (Sprint Backlog). На ближайший спринт наша команда взяла две задачи поиск ближайшей станции зарядки и поиск самоката по сигнализации. Мы оцениваем объем и сложность работы и составляем список задач, которые нужно выполнить.

Реализация. В течение спринта наша команда выполняет задачи: проектирует функциональность и разрабатывает её, затем тестирует и выпускает в релиз.

Демонстрация. Встреча, на которой наша команда покажет готовую версию продукта с внедренными функциями заказчику

Ретроспектива, или ретро. Команда обсудит результаты спринта: что получилось хорошо, какие ошибки нужно исправить и какой опыт учесть в будущем.

- Какие виды тестирования тебе нужно будет выполнить? Почему?

Компонентное тестирование - чтобы протестировать новый функционал.

Интеграционное тестирование - чтобы протестировать взаимодействие интегрированного функционала с другим функционалом программы.

Системное тестирование - чтобы проверить, как работает система в целом после внедрения нового функционала.

## Проект 2: Ретест багов, которые нашли другие тестировщики в сервисе Mesto.

### 1. [Баги в Яндекс Трекере](#):

#### 2. Выводы о работе:

Насколько понятно сформулированы баг-репорты? По ним легко воспроизвести ошибку?  
На мой взгляд - все баг-репорты сформулированы неправильно:

- ни в одном баг-репорте нет описания проблемы, при этом название баг-репорта тоже не дает однозначного понимания, что, где и когда произошло
- только в одном баг репорте есть ссылка на проверяемое приложение
- ни в одном баг-репорте нет приложенного скриншота или скринкаста
- ни в одном баг-репорте не указано окружение

По идее, нужно вернуть все баг-репорты на доработку, но я предположила, что мы работаем в одной команде и в одном окружении и нам более или менее понятно о чем речь.

Все ли ошибки исправлены? - Нет, из 15 багов исправлено 9

Удалось ли тебе найти новые баги? - во втором проекте - нет, все дополнительные баги указаны, которые я нашла, я отразила в первом проекте.

#### 3. Вопросы на рассуждение:

- Как думаешь, может ли появиться другая ошибка на месте исправленной? Почему? - да, может. При исправлении одной ошибки, может пострадать функционал, который находится рядом
- Что ты предпримешь, если при ретесте обнаружишь новый баг? - заведу баг- репорт на новый баг
- Что делать тестировщику, если разработчики не исправили несколько найденных багов? - сообщить о неисправленных багах разработчику. оставить статус "открыт" в баг-репорте
- Может ли случиться такое: разработчик сообщил, что исправил ошибки, но после повторной проверки выяснилось — все баги по-прежнему на месте? Почему? - может, скорее всего - это человеческий фактор. Возможно, программист не сохранил изменения и не перепроверил это.