

# Руководство по работе с UX Rocket

## Оглавление

Введение .....	3
Сбор данных .....	3
Сбор данных на сайте .....	4
Просмотр сырых данных .....	6
Подключение SDK в мобильное приложение .....	6
Интеграция с Google Big Query .....	7
Интеграция с Яндекс.Директ .....	7
Сквозная аналитика .....	7
Модели атрибуции .....	8
Отчет «Анализ трафика» .....	9
Клиентская аналитика .....	10
Диаграмма Сэнки .....	10
Путешествие в прошлое .....	11
Путешествие в будущее .....	11
Дашборд «Воронка» .....	11
Агрегаты по сессиям пользователей .....	15
Аналитические виджеты .....	16
AI-модель для выявления проблем и точек роста .....	17
A/B тестирование и персонализация .....	19
Подготовка к A/B тестированию .....	19
Жизненный цикл кампаний .....	19
Кампании управления контентом .....	20
Элементы и варианты .....	21
Правила показа вариантов .....	23
Целевые действия .....	24
Настройка скриптов для сайта .....	24
Многостраничные кампании .....	25
Отладка кампаний .....	25
Публикация кампаний .....	27
Анализ результатов .....	27
Статистическая оценка показателей теста .....	28

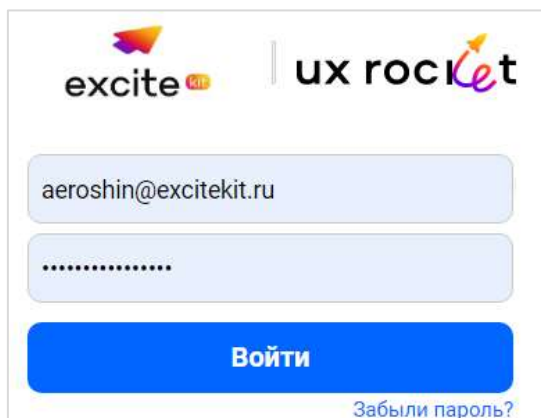
Рекомендации .....	30
Профиль клиента.....	31
Типы идентификаторов клиента.....	31
Идентификация в цифровых каналах.....	32
Fingerprint и кросс-браузерная идентификации .....	33
Карточка профиля клиента.....	33
Объединение профилей .....	35
Перенос идентификаторов между профилями .....	36
Разрешение конфликтов идентификации .....	37
Правила создания событий.....	37
Загрузка офлайн данных.....	38
Подключение SDK.....	39
Основные функции SDK .....	40
Поддерживаемые платформы .....	40
Начало работы с SDK .....	40
Подключение библиотеки SDK .....	40
Инициализация.....	41
Установка значений по умолчанию.....	42
Геолокация.....	43
Сбор сведений о поведении пользователя.....	43
Применение вариантов таргетинга.....	45
Сбор сведений о взаимодействии с персонализированным контентом .....	47

## Введение

Программный продукт UX Rocket состоит из административного модуля и личного кабинета клиента. Заведение аккаунта для нового клиента выполняется в административном модуле. Дальнейшие операции по добавлению пользователей в аккаунт, администрированию прав доступа, настройке сбора данных и доступ к отчетам производятся в личном кабинете клиента по адресу <https://lk.uxrocket.ru>.

## Сбор данных

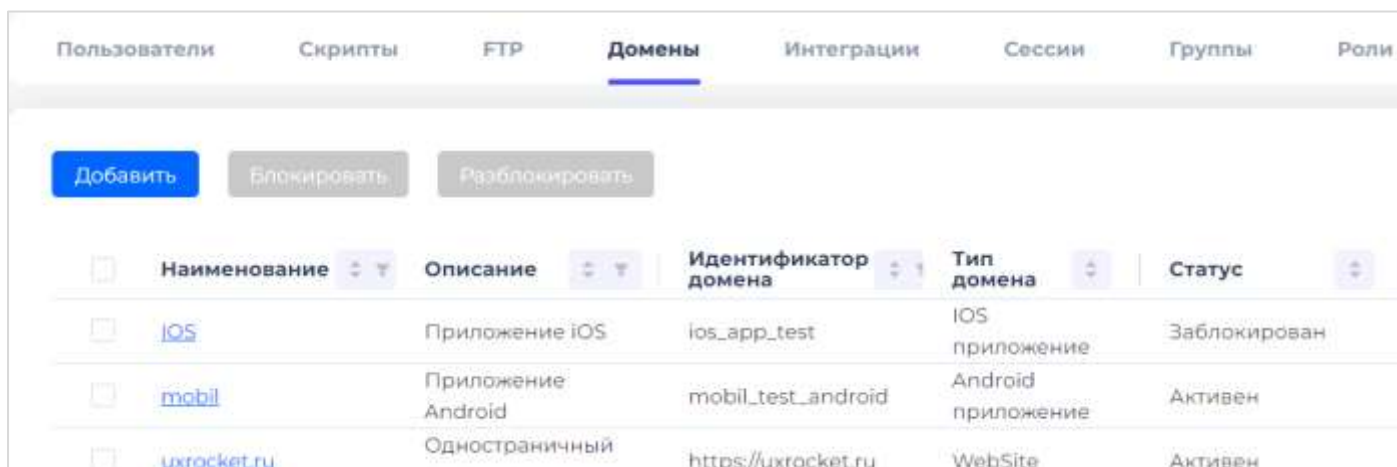
Для входа в аккаунт перейдите по ссылке <https://lk.uxrocket.ru> и введите логин и пароль.



В качестве логина система использует email пользователя. Для смены пароля выберите пункт «Забыли пароль», и система направит вам на почту письмо со ссылкой для ввода нового пароля.

В рамках одного аккаунта Вы можете настроить сбор данных с одного или нескольких сайтов и с одного или нескольких мобильных приложений.

Прежде всего перейдите в меню Настройки, раздел «Домены» и заведите свои домены.



	Наименование	Описание	Идентификатор домена	Тип домена	Статус
<input type="checkbox"/>	<a href="#">ios</a>	Приложение iOS	ios_app_test	iOS приложение	Заблокирован
<input type="checkbox"/>	<a href="#">mobil</a>	Приложение Android	mobil_test_android	Android приложение	Активен
<input type="checkbox"/>	<a href="#">uxrocket.ru</a>	Одностраничный	https://uxrocket.ru	WebSite	Активен

Для подключения сайта к программному продукту UX Rocket необходимо перейти на вкладку «Скрипты», скопировать скрипт и установить его на сайте в разделе head согласно следующим рекомендациям:

*«При установке тега в HTML-код сайта, разместите код как можно ближе к началу страницы, внутри тега <head></head>. Если сайт разработан на react, то добавлять тег надо в код сайта без использования tag-менеджера. Если на сайте установлен gdpr-скрипт, то в интерфейсе администратора разрешите вызов скрипта UX Rocket. Если на сайте используются системы Google Analytics или Яндекс Метрика, то тег UX Rocket необходимо разместить после тегов данных систем.»*

## Сбор данных на сайте

Важное преимущество UX Rocket состоит в том, что сбор данных о поведении пользователей на сайте происходит автоматически сразу после установки тега на сайт. Вам не надо вручную отправлять в систему какие-то события и цели, как при работе с продуктами Yandex и Google. Вы экономите время сотрудников на разметке сайта (она больше не нужна) и выигрываете на полноте и качестве разметки (система соберёт все данные и исключит человеческий фактор).

При автоматическом сборе данных важно сделать эти данные понятными для конечных пользователей. В нашем случае описание должно быть понятно для сотрудников отдела маркетинга и цифровизации. На рисунке ниже показано, как выглядит запись о том, что пользователь на странице «/demo» (пусть от корня сайта) нажал на кнопку (тип действия = buttons) с названием «Отправить заявку».

Дата	Пользователь	Url	Тип действия	Действие
01.03.2023 08:01:19	1677657674182.1207918321	/demo	buttons	Отправить заявку

То есть маркетолог сможет анализировать собранные данные без предварительного обучения и без помощи ИТ специалистов.

Каждая запись в таблице «Сырые данные» представляет собой одно действие пользователя: переход пользователя на страницу, выбор пункта меню, клик по кнопке и т.д. Перечень отслеживаемых действий настраивается в административном модуле. По умолчанию система собирает следующие действия посетителей сайта:

- openpage – переход на страницу;
- closepage – закрытие страницы;
- buttons – нажатие на кнопку;
- links – клик на гиперссылку;
- lists – выбор элемента в выпадающем списке.

Можно подключить сбор дополнительных событий, например,

- show – показ всплывающего элемента (диалоги, баннеры).
- click – клик по элементу на странице (например, баннер);
- change – нажатие на радиокнопки и переключатели (checkbox и radiobutton);
- field – данные из поле ввода при потере фокуса (переход на другое поле).

Важно понимать, что систему UX Rocket можно подключить как облачный сервис или развернуть на серверах вашей компании. При сборе данных в наш облачный сервис Вы можете собирать любые данные о поведении пользователей, кроме данных, сбор и управление которыми регламентируется специальными законодательными актами (персональные данные, медицинские данные и прочее).

Собирать персональные данные Вы можете при развертывании системы на серверах вашей компании. Пример сбора персональных данных показан на картинке ниже.

Дата	Пользователь	Url	Тип действия	Действие
01.03.2023 08:41:02	1652350139825.5778491095	/demo	fields	Иванова м

Каждая запись в таблице «Сырые данные» содержит метку времени, идентификатор сессии и пользователя, группу полей для идентификации события (url, тип действия, действие), стандартные и пользовательские атрибуты.

Стандартные атрибуты – это местоположение, характеристики устройства, UTM-метки и некоторые другие данные. Система автоматически собирает и сохраняет стандартные атрибуты. Какие-либо настройки для сбора стандартных атрибутов не требуются.

Пользовательские атрибуты – это 30 дополнительных текстовых полей (длиной до 64 символов), в которые вы можете сохранить дополнительную информацию о посетителе и/или его поведении на сайте. Сохранить можно любую информацию: значения из полей на странице сайта, значения полей в куки, значения полей в url, значения полей из внешних сервисов (при интеграции UX Rocket со сторонним ПО или вашей CRM-системой). Например, можно сохранять уровень участия в программе лояльности, предпочтительный способ оплаты, чтобы смотреть аналитические отчёты в соответствующих разрезах.

На рисунке ниже показано окно для настройки пользовательского атрибута.

Редактировать атрибут

Название: utm\_kuki

Название для поиска: Куки из UTM

Источник: Хранилище сессии

Название в источнике: utm\_camp

Привязываемое поле: Attr10

Описание: Куки из UTM

Сохранить Отменить

Получение данных из сессионных куки и из URL не требует программирования. Получение данных из других источников требует написания java-script (далее – скрипт) в меню «Общие скрипты». Логика работы «Общих скриптов» следующая: система выполняет скрипт при загрузке страницы. Скрипт вычисляет или получает значение пользовательского атрибута и сохраняет его в куки. Данные из куки можно использовать для заполнения пользовательских атрибутов.

Использование скриптов для заполнения куки позволяет получить пользовательский атрибут не только в момент определённого события на сайте, но и позднее. Рассмотрим два примера.

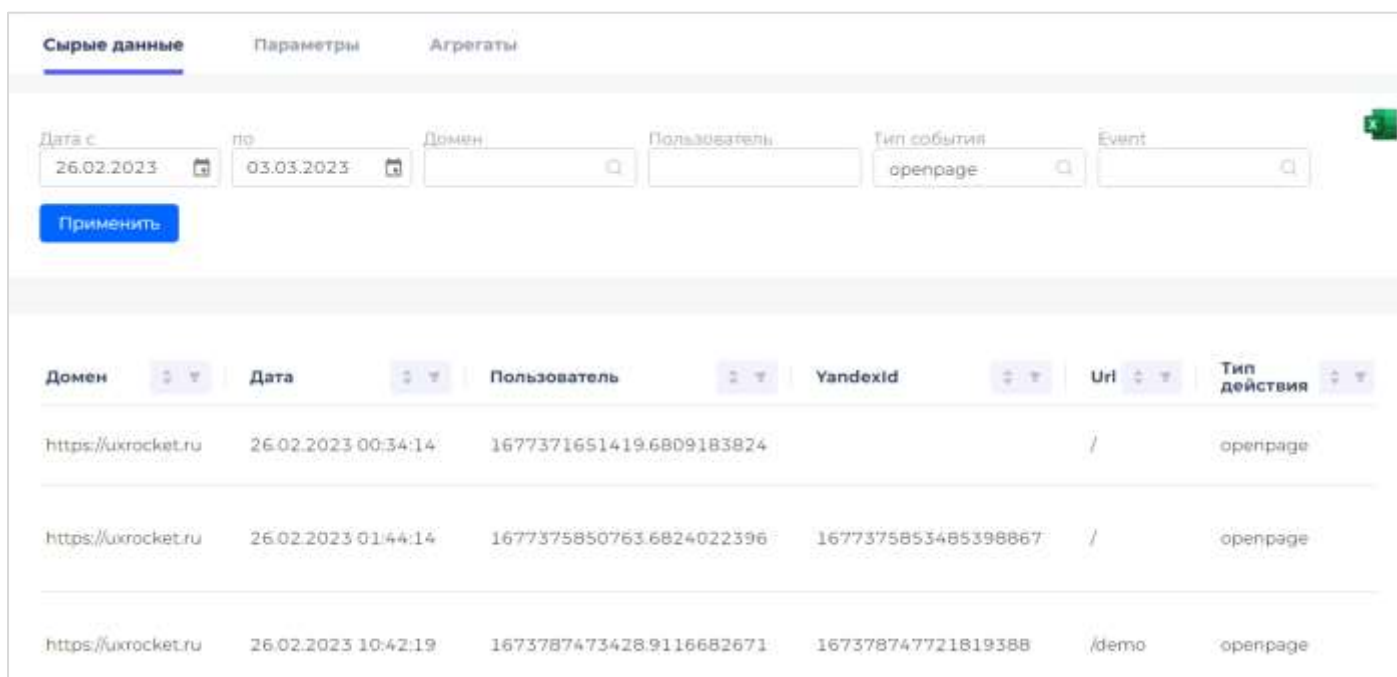
Пример №1. Перед переходом на платёжную страницу можно сохранить в куки сумму и номер заказа, а на странице «спасибо за покупку» считать данные из куки о сумме покупки, номере заказа и купленных категориях товара и сохранить их совместно с фактом успешной оплаты. Это упрощает построение аналитических отчётов.

Пример №2. Давайте сохраним значение стандартного атрибута `utm_campaign` в куки и свяжем с ним пользовательское поле `utm_kuki`. Теперь на каждой странице сайта можно получить данные о рекламной кампании, из которой посетитель перешёл на сайт. В частности, по полю `utm_kuki` можно настроить правила показа для А/В тестов.

Обратите внимание, что использование «Общих скриптов» позволяет получать расширенную аналитику о поведении посетителей сайта без каких-либо изменений программного кода сайта.

## Просмотр сырых данных

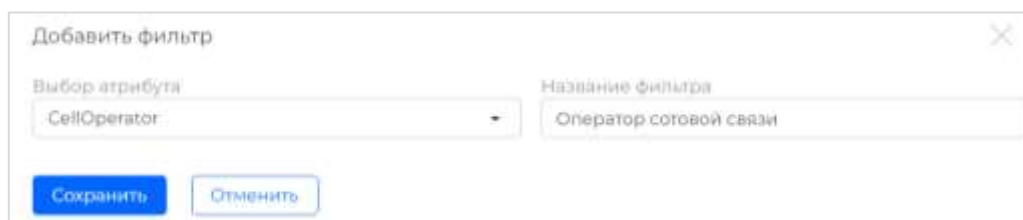
В интерфейсе UX Rocket можно просмотреть интересующий фрагмент таблицы сырых данных, а также выгрузить его в файл Excel для дополнительного анализа или проверки. Для этого следует перейти в меню «Сбор данных – Сырые данные», выбрать требуемый промежуток времени и заполнить значения фильтров.



Домен	Дата	Пользователь	Yandexid	Uri	Тип действия
https://uxrocket.ru	26.02.2023 00:34:14	1677371651419.6809183824		/	openpage
https://uxrocket.ru	26.02.2023 01:44:14	1677375850763.6824022396	1677375853485398867	/	openpage
https://uxrocket.ru	26.02.2023 10:42:19	1673787473428.9116682671	167378747721819388	/demo	openpage

На экран выводится фрагмент данных размером не более 1000 записей, но выгрузить можно до 100 тысяч записей, попадающих под условия фильтра.

Для получения нужной выборки данных предусмотрена настройка фильтров. Фильтры можно добавить, перейдя на вкладку «Параметры». Фильтром может служить любое интересующее поле таблицы, в которое записываются данные.



Добавить фильтр

Выбор атрибута: CellOperator

Название фильтра: Оператор сотовой связи

Сохранить Отменить

## Подключение SDK в мобильное приложение

Для подключения мобильного приложения к программному продукту UX Rocket перейдите в раздел «Подключение SDK».

Сбор данных для мобильного приложения имеет два основных отличия от сбора данных на сайте:

- 1) На сайте тег UX Rocket автоматически вызывает скрипт и собирает данные о действиях посетителя, а в мобильном приложении при подключении SDK программист мобильного приложения должен прописать в коде вызов метода сохранения данных в UX Rocket при каждом отслеживаемом действии (открытии страницы, нажатии кнопок и так далее);
- 2) Так как в мобильном приложении нет url, системные названия страниц не видны, а идентификаторы конкретных полей имеют системные имена, понятные только разработчикам приложения, то вместо тройки полей (url, тип действия, действие) мы используем события, по аналогии с событиями в Yandex и Google. Имя события мы сохраняем в поле «Действие».

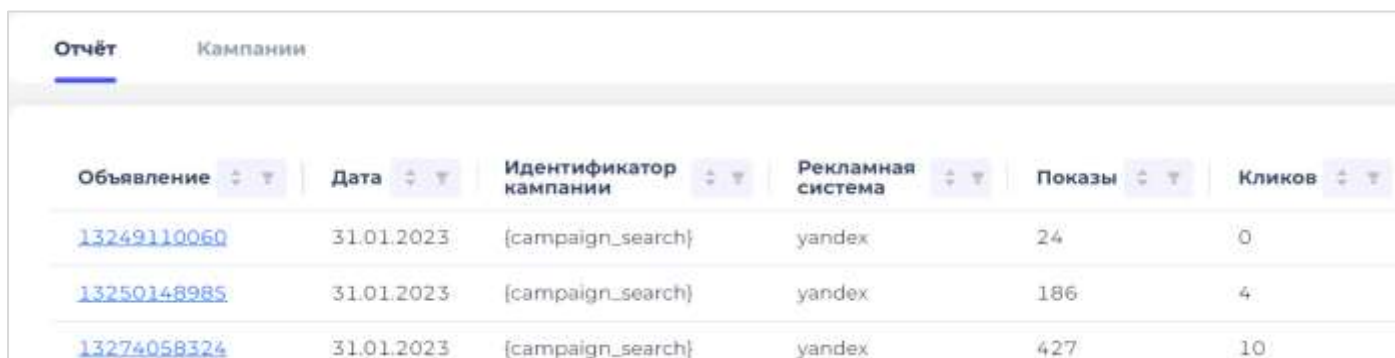
SDK UX Rocket для мобильного приложения автоматически заполняет стандартные атрибуты, кеширует параметры пользователя и вызовы к API (при отсутствии связи событие будет сохранено локально и передано позднее). Пользовательские атрибуты заполняет/сбрасывает программист мобильного приложения.

## Интеграция с Google Big Query

В программном продукте UX Rocket реализована загрузка данных из Google Big Query. Это интересно для тех клиентов, кто строил аналитику в решении Google, но после закрытия платного аккаунта в Google хотел бы сохранить исторические данные. Настройка коннектора в меню «Сбор данных -> Google BigQuery» позволяет настроить коннектор и выборочно перегрузить данные в систему UX Rocket для использования их в аналитических отчётах за прошлые периоды.

## Интеграция с Яндекс.Директ

В программном продукте UX Rocket можно настроить интеграцию с вашим аккаунтом Яндекс.Директ. Это позволит сопоставить данные о показах, кликах, затратах на рекламу и фактических покупках клиентов на сайте. Сопоставления обобщенных данные о поведении на сайте и о показе рекламы в Яндекс настроено с помощью UTM-меток.



Объявление	Дата	Идентификатор кампании	Рекламная система	Показы	Кликов
<a href="#">13249110060</a>	31.01.2023	{campaign_search}	yandex	24	0
<a href="#">13250148985</a>	31.01.2023	{campaign_search}	yandex	186	4
<a href="#">13274058324</a>	31.01.2023	{campaign_search}	yandex	427	10

## Сквозная аналитика

Сквозная аналитика — это метод анализа маркетинговых инвестиций, прослеживающий путь покупателя от первого посещения сайта и до покупки товара (последующих покупок). Программный продукт UX Rocket включает в себя инструментарий сквозной аналитики. Построение отчёта по сквозной аналитике проходит в несколько шагов.

Прежде всего система агрегирует детальные данные о поведении посетителей в записи по сессиям. Каждой сессии по набору правил будет присвоен источник трафика. Источники трафика бывают платные и бесплатные и делятся на несколько групп:

Тип источника трафика		
+	<a href="#">Рекламные системы</a>	
+	<a href="#">Социальные сети</a>	
+	<a href="#">Мессенджеры</a>	
+	<a href="#">Рекомендательные системы</a>	
[-]	<a href="#">Поисковые системы</a>	
Наименование	Платный	
<a href="#">поиск Яндекс</a>	Нет	
<a href="#">поиск Google</a>	Нет	
<a href="#">Mail.ru</a>	Нет	

Правила определения источника трафика можно настроить в административном модуле системы. Если Вы пользуетесь облачной версией системы, то получить информацию о текущих настройках правил и/или изменить правила можно в службе поддержки UX Rocket. Если Вы используете корпоративную версию, то экранная форма для настройки источников трафика выглядит следующим образом:

Группа источников Поисковые системы	Название поиск Яндекс	Платный Нет		
<input type="button" value="Сохранить"/>				
<input type="button" value="Добавить"/> <input type="button" value="Удалить"/>				
Правило	Тип правила	utm-source	Тип правила	utm-medium
<a href="#">Правило №625</a>	Равно	yandex	Равно	organic

Кроме источника трафика в таблице сессий будут заполнены код клиента, utm-метки, сумма покупки в пределах данной сессии и ряд других полей.

Теперь настроим автоматическую загрузку данных о маркетинговых затратах по каналам или вручную загрузим эти данные за соответствующий период.

Система UX Rocket имеет стандартные коннекторы для некоторых рекламных кабинетов. Например, коннектор с Яндекс.Директ, позволяет в автоматическом режиме загружать затраты на рекламу с детализацией до кампаний, рекламных объявлений и ключевых слов. Если Вы не хотите использовать стандартные коннектор, то данные о затратах на рекламу можно загрузить из Excel. Просмотреть ранее загруженные в систему расходы на рекламу можно в меню «Атрибуция -> Расходы на рекламу».

## Модели атрибуции

При наличии нескольких рекламных касаний, предшествующих продаже, сумму покупки можно распределить между рекламными каналами различными способами. Атрибуция трафика — это распределение ценности между источниками трафика, которые привели к конверсии. То есть, это определение, какую роль сыграл каждый из источников, с которых клиент приходил на сайт перед покупкой.

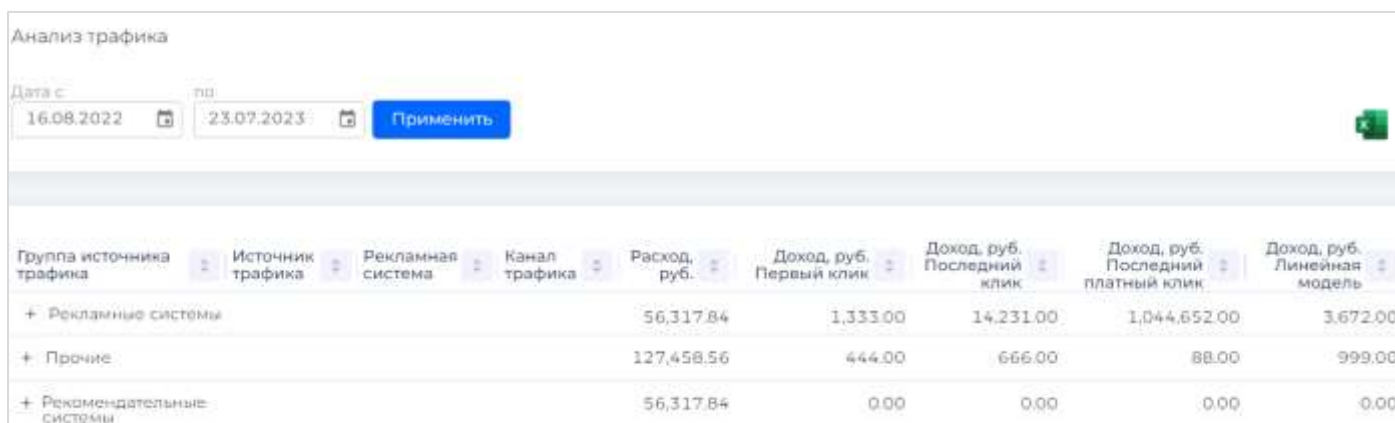


В UX Rocket реализовано 4 модели атрибуции:

- 1) «первый клик» (first click) – считаем, что доход принес первый заход на сайт, когда пользователь узнал о нашем товаре или услуге;
- 2) «последний клик» (last click) – доход присваиваем последнему заходу пользователя, когда он совершил покупку;
- 3) «последний платный клик» (last indirect click) – доход присваиваем последнему перед покупкой заходу из платного источника;
- 4) «линейная» – доход распределяем в равных долях между всеми сессиями пользователя, приведшими к покупке.

## Отчет «Анализ трафика»

С помощью отчета «Анализ трафика» можно проанализировать вклад всех источников трафика в полученный доход с применением различных моделей атрибуции. В отчёте можно сравнить расходы на рекламу с доходами по различным моделям атрибуции по источникам трафика и рекламным кампаниям.



Дата с	по	Группа источника трафика	Источник трафика	Рекламная система	Канал трафика	Расход, руб.	Доход, руб. Первый клик	Доход, руб. Последний клик	Доход, руб. Последний платный клик	Доход, руб. Линейная модель
16.08.2022	23.07.2023	+ Рекламные системы				56,317.84	1,333.00	14,231.00	1,044,652.00	3,672.00
		+ Прочие				127,458.56	444.00	666.00	88.00	999.00
		+ Рекомендательные системы				56,317.84	0.00	0.00	0.00	0.00

Детальный анализ кампаний по рекламным объявлениям и поисковым словам удобнее делать в Excel, так как в отличие от web-форм в отчётах Excel можно закреплять заголовки таблиц, больше возможностей для фильтрации данных.

Существуют различные алгоритмы для анализа эффективности инвестиций на основании полученных данных. Например, Вы смотрите все каналы, где доход по модели «Первый клик» меньше расхода. Получается, что реклама приводит клиентов, которые не мотивированы к покупке. Надо оптимизировать кампанию/канал или отключать. Если канал по первому клику прибыльный, то переходим к анализу по следующей модели атрибуции.

В результате последовательного анализа по разным моделям атрибуции Вы сможете выявить перечень кампаний, которые работают хорошо: в них можно добавить денег и доход от продажкратно вырастет. Также Вы выявите кампании и каналы, которые надо оптимизировать или отключить, потому что они тратят больше, чем приносят дохода.

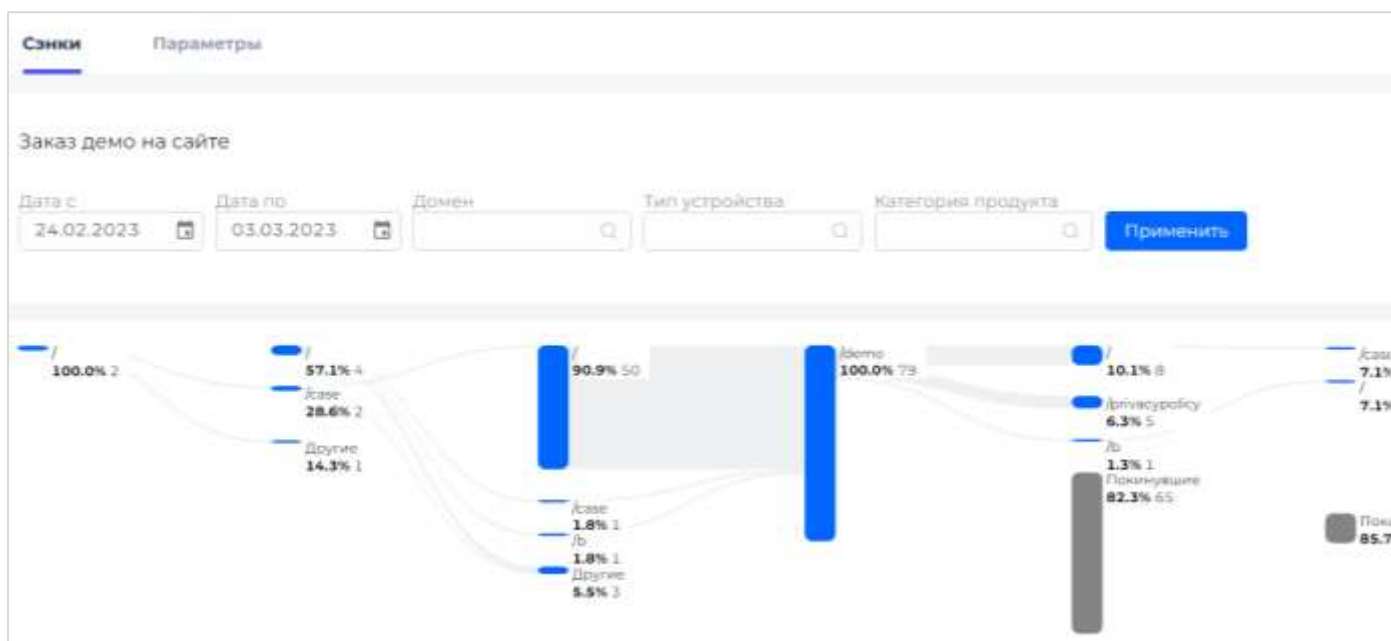
## Клиентская аналитика

Программный продукт UX Rocket включает несколько типов аналитических отчётов для анализа клиентского поведения на сайте и в мобильном приложении. Мы рассматриваем клиентский путь в цифровых ресурсах (сайт, МП и так далее) как путешествие, позволяющее клиенту достичь своей цели – найти и купить интересующий клиента продукт, а компании – получить доход от продажи. Как и в любом путешествии клиенты могут поджидать трудности (например, кнопка оформить заказ неактивна) и ускорители (например, автозаполнение полей заказа).

Обзор аналитических инструментов мы начнём знакомством с диаграммами Сэнки.

### Диаграмма Сэнки

Диаграмма Сэнки предназначена для наглядного отражения потоков посетителей, путешествующих по сайту в течение определенного периода времени (см. рисунок ниже).



При построении диаграммы вы выбираете интересующее вас событие: переход на страницу сайта или действие на странице. Затем надо определить направление путешествия (вперёд, назад или в обе стороны) и количество событий из путешествия клиента в соответствующих направлениях. Далее указываем степень детализации (количество событий в шаге)

Наименование диаграммы	Основное событие	Шаги до	шаги после	Количество событий в шаге
Заказ демо на сайте	/demo	4	4	5

и настраиваем фильтры для выбора исходных событий (сегмент интересующих нас посетителей).

Дата с	Дата по	Домен	Тип устройства	Категория продукта
24.02.2023	03.03.2023	uxrocket.ru	Mobile Phone	

Что интересного можно узнать на этой диаграмме? Как правило, диаграмму используют для решения двух типов задач: увидеть успешные пути клиента в определённую точку сайта и увидеть пути клиентов с определённой страницы сайта. Рассмотрим эти задачи более подробно.

## Путешествие в прошлое

Предположим, мы хотим просмотреть все основные пути, по которым посетители сайта приходят к покупке определённой категории товара или продукта. Зачем это нужно? Чтобы сравнить, по какому пути конверсия выше и **перенаправить пользователей с менее конверсионного пути на более конверсионный**. Например, многие сайты при оформлении заказа очень настойчиво просят посетителя авторизоваться для оформления покупки, однако процент успешных покупок после авторизации меньше, чем без авторизации. Следовательно, если дать посетителям продолжить покупку без авторизации, то продажи вырастут без каких-либо операционных затрат в будущем!

В UX Rocket вы можете увидеть путь клиента к покупке, если выберете страницу «Спасибо за покупку», в качестве направления укажете «N шагов до покупки», а в качестве значений фильтров укажете интересующий вас сегмент покупателей (тип устройства, категорию купленного продукта и так далее) или оставите значение фильтра пустым (если нужна укрупненная картина по покупкам без детализации).

## Путешествие в будущее

Посетителей сайта, как правило, много, поэтому анализировать путь отдельного клиента (в меню «Сбор данных -> Сырые данные» или в Яндекс Вебвизор) имеет смысл только тогда, когда клиент является «среднестатистическим», а точнее выполняет те же действия на сайте, что и большой сегмент (группа) других посетителей сайта. Как нам найти таких клиентов? Построить диаграмму Сэнки на некоторое число шагов вперёд от интересующего нас события. Например, мы видим, что большое число клиентов вместо перехода на страницу оплаты прекращают процесс покупки. Возможно, на странице оплаты предлагается ввести промо-код, и клиенты ищут его на сайте, потом в интернете и забывают завершить покупку.

В UX Rocket вы можете провести первичный анализ, если выберете интересующую вас страницу и построите диаграмму Сэнки на несколько шагов вперёд. Для любого из переходов с большой долей трафика можно выгрузить в Excel список выполнивших действие клиентов и детально рассмотреть их путь в меню «Сбор данных -> Сырые данные» (фильтр по посетителю или сессии). Кроме того, диаграмма позволяет понять куда пошли посетители, а также сформировать перечень поведенческих гипотез для проверки в других аналитических отчётах.

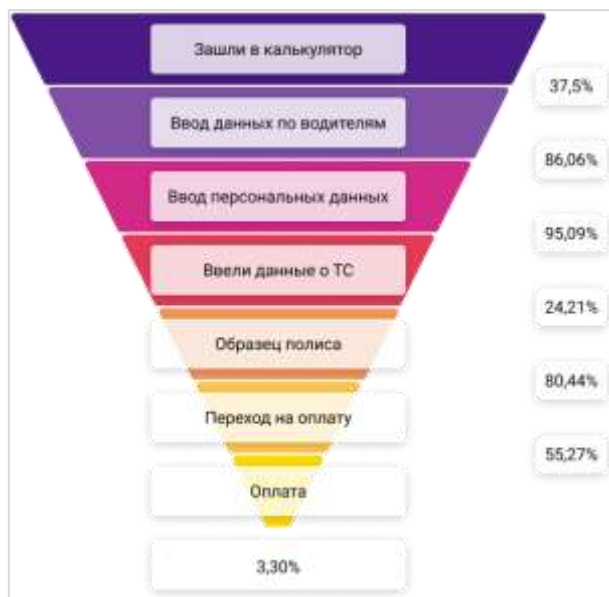
## Дашборд «Воронка»

При покупке в онлайн клиент проходит определённую последовательность шагов, немного разных в зависимости от сферы деятельности владельца сайта. Например, в интернет-магазине надо посмотреть товар, добавить его в корзину, перейти в корзину, оформить заказ, выбрать способ оплаты и оплатить. Вы никак не сможете оплатить заказ, пока не добавите товар в корзину, но вы можете добавить товар в корзину и уйти с сайта без покупки. Таким образом мы можем говорить о воронке продаж как наборе шагов, которые надо пройти клиенту после того, как он начал покупку<sup>1</sup>.

---

<sup>1</sup> Не все действия клиента на сайте будут шагами воронки. Например, просмотр информации о компании, поиск товара и некоторые другие шаги не являются обязательными. Клиент часто выполняет такие шаги задолго до того, как у него возникло намерение о покупке.

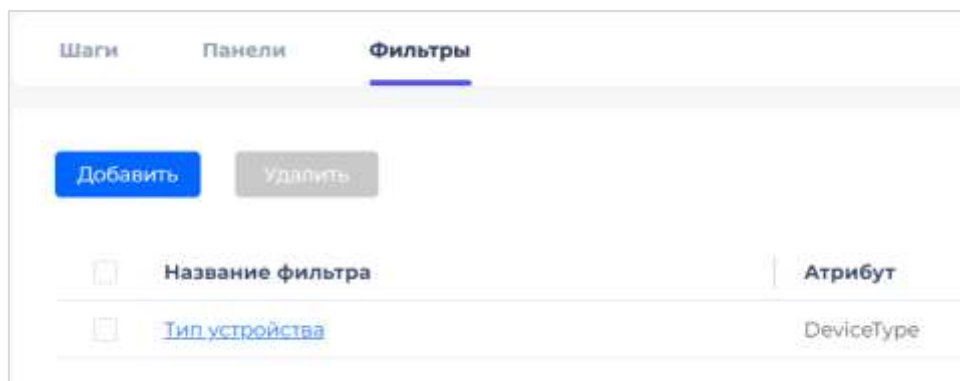
Ниже приведём пример воронки для продажи полисов по страхованию автомобильного транспорта.



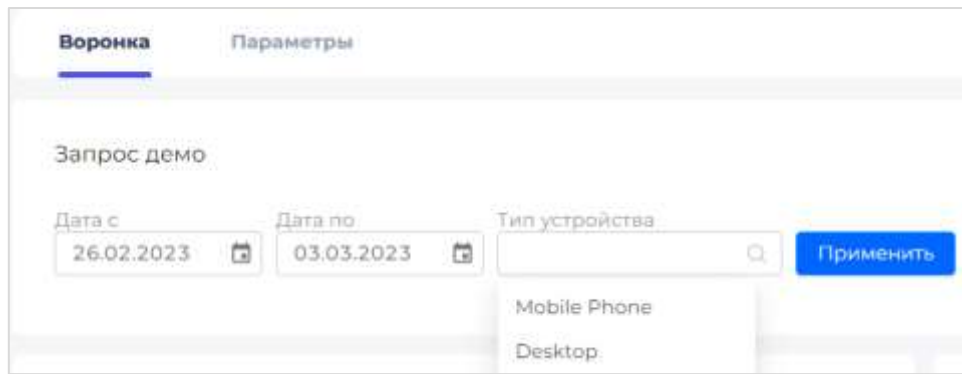
В данном случае, каждый шаг воронки – это страница сайта (URL), на которую перешёл посетитель сайта в процессе оформления полиса: заполнение полей для расчёта, просмотр рассчитанной суммы и оплата полиса. Внизу воронки мы видим цифру 3,30% — это процент посетителей сайта, кто дошёл от ввода номера автомобиля до оплаты полиса. Справа после каждого шага мы также видим проценты – это доля людей, которые успешно перешли с данного шага на следующий.

Использование воронок **помогает найти узкие места в работе вашего сайта или мобильного приложения**. В дальнейшем надо подумать о возможных причинах такого поведения клиентов (в нашем случае отказ от продолжения покупки) и проверить гипотезы в отчётах или путём A/B-тестирования.

На рисунке выше можно говорить о трёх узких местах в процессе продажи полисов: две трети уходят с первого шага воронки, три четверти уходят после получения расчёта стоимости и почти половина уходит с платёжной страницы без оплаты. Как пример, давайте проведём анализ по первому узкому месту. Наша первая гипотеза состоит в том, что оформлять полис со смартфона неудобно, поэтому большинство пользователей с телефонами отказываются от покупки. Как проверить такую гипотезу? Давайте настроим фильтр по типу устройства. Фильтры можно добавить, перейдя на вкладку «Параметры». Фильтром может служить любое интересующее поле таблицы, в которое записываются данные.

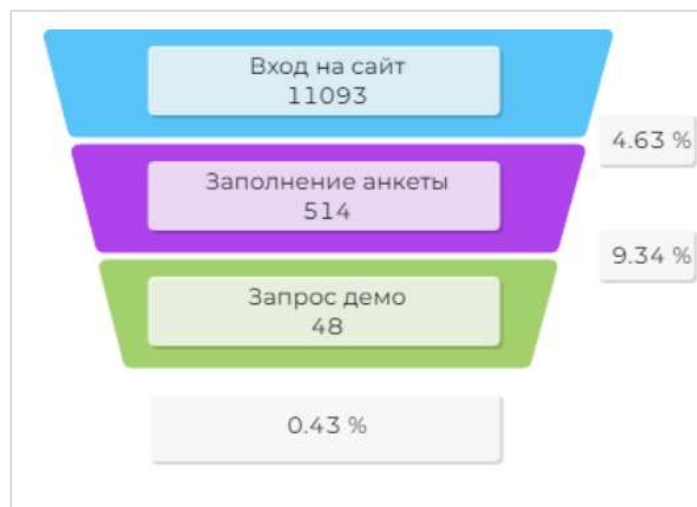


Теперь можно построить воронки отдельно для компьютеров и для мобильных устройств и сравнить потери трафика на первом шаге к покупке.



Сравнивая данные воронок для разных устройств мы видим, что для компьютеров потеря трафика на первом шаге 40%, а для мобильных устройств более 80%. Следовательно, для повышения продаж надо сделать более удобной ввод данных с мобильных устройств.

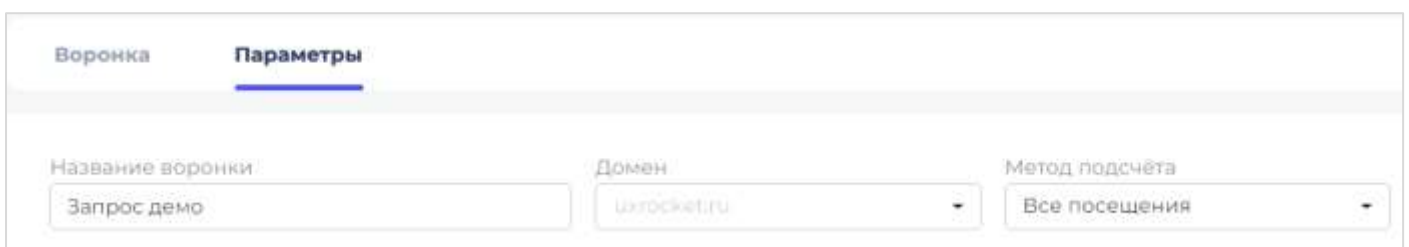
Рассмотрим ещё один пример. На рисунке ниже показана воронка из трёх шагов по продаже программного обеспечения.



Для каждого шага воронки указано название шага и количество посетителей сайта. Справа указан процент посетителей, перешедших на следующий шаг. Легко видеть, что число людей, заполнивших анкету примерно в 10 раз больше, чем число заказавших демонстрацию. В данном случае, стоит изучить вопрос поведения на сайте тех, кто заполнил анкету, но не стал продолжать заказывать демо. Выполнить такое исследование можно в меню «Сбор данных -> Сырые данные».

Важно отметить, что воронка **UX Rocket** – это инструмент анализа данных для маркетолога, а не просто написанный программистом фрагмент кода. Вы можете без каких-либо технических навыков построить любое число воронок и анализировать данные в нужных вам разрезах.

Давайте рассмотрим процесс построения воронки более подробно. При построении воронки вы выбираете домен (в личный кабинет можно подключить несколько сайтов и мобильных приложений) и способ подсчета записей для шагов воронки: все посещения или уникальные посещения.



В нижней части экрана настройки вы вводите шаги воронки, аналитические панели и фильтры.

Шаги		Панели	Фильтры		
<input type="button" value="Добавить"/> <input type="button" value="Удалить"/>					
<input type="checkbox"/>	Название шага	Страница сайта	Тип действия	Действие	Цвет
<input type="checkbox"/>	Вход на сайт	/	openpage		#59C4F7
<input type="checkbox"/>	Заполнение анкеты	/	fields		#AF43EB
<input type="checkbox"/>	Запрос демо	/	buttons	Получить материалы	#A3D16E

Для добавления шага надо написать его названия, выбирать цвет на диаграмме, указать страницу сайта (URL), тип действия пользователя (на рисунке ниже buttons – это нажатие на кнопку) и название кнопки (что написано на этой кнопке на странице сайта). **Никаких технических знаний для анализа данных не требуется!**

Добавить шаг воронки ✕

Название шага  Цвет

Страница сайта  Маска

Тип действия  Действие

Аналогичным образом вы добавляете все шаги воронки и их последовательность. Если Вам требуется построить воронку для определенного сегмента пользователей, укажите на каждом шаге воронки необходимые характеристики пользователей из состава стандартных и/или пользовательских атрибутов. Пример ниже показывает фрагмент настройки воронки для жителей Екатеринбурга.

Обновить шаг воронки ✕

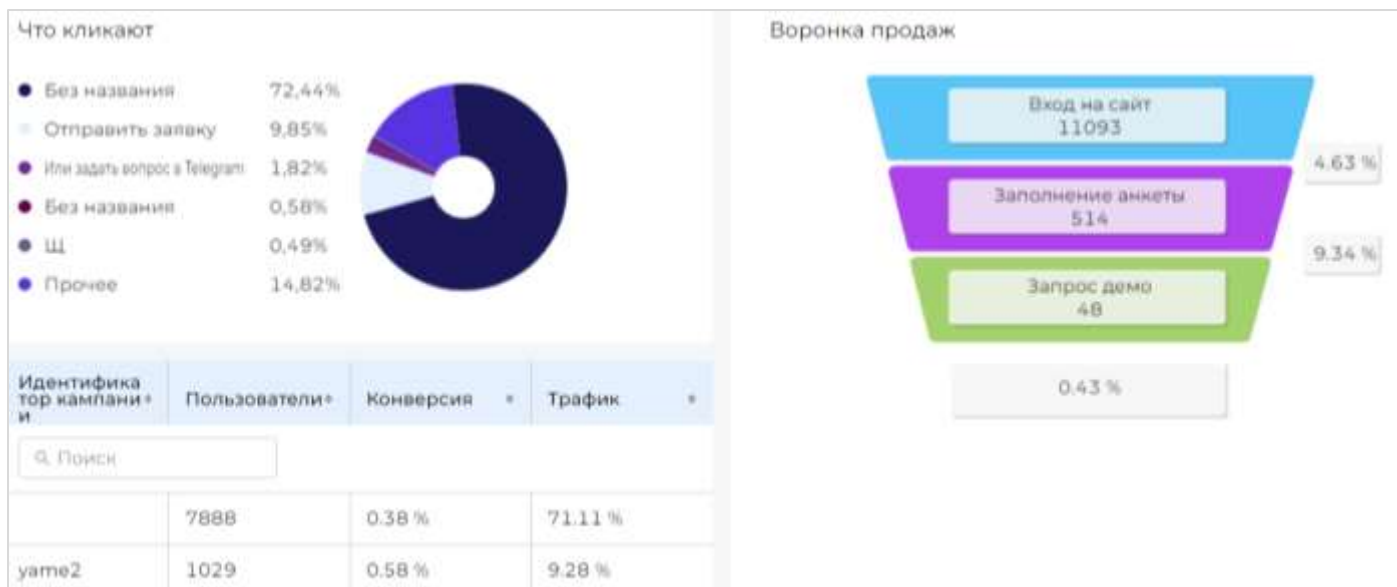
Название шага  Цвет

Страница сайта  Маска

Тип действия  Действие

Атрибут 1  Значение

В разделе панелей вы можете добавить табличные или графические расшифровки по структуре трафика на входе в воронку или на любом из её шагов. В том числе, можно посмотреть какие действия, в экранной форме выполняют клиенты для любого из шагов воронки. На экране ниже показано из каких рекламных каналов попадают в воронку наши клиенты и что они делают после заполнения анкеты.



Воронки для мобильных приложений выглядят аналогичным образом. Единственное отличие состоит в том, что вместо тройки полей (url, тип действия, действие) мы используем события.

## Агрегаты по сессиям пользователей

При построении аналитических отчётов можно использовать как детальные данные о действиях посетителей сайта и мобильного приложения, так и агрегированные данные по сессиям.

При построении некоторых отчётов, например, число посетителей сайта, сумма покупок и так далее, удобнее использовать агрегированные данные по сессиям. UX Rocket обрабатывает детальные данные о действиях на сайте и, с задержкой в несколько минут, считает агрегированные данные по сессиям.

В меню «Сбор данных – Сессии» можно просмотреть и выгрузки в Excel данных по сессиям.

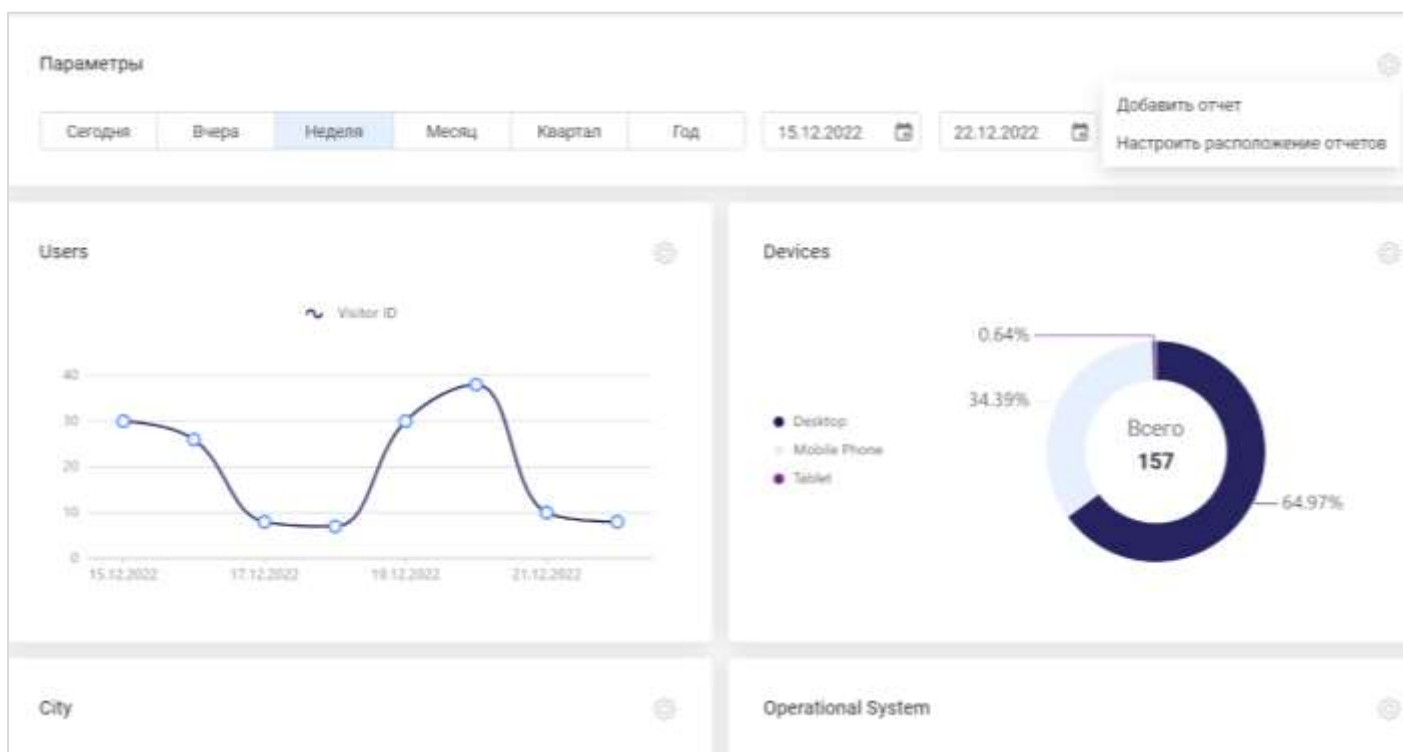
**Сессии** | Параметры

Дата с: 03.03.2023 | Дата по: 03.03.2023 | Дрмен: | Пользователь: | Применить

Идентификатор	Посетитель	Начало сессии	Тип пользователя	Тип устройства	ОС
165871787046670400523221	16517054888169441929133	03.03.2023 13:11:54	Returned	Desktop	Windows 10
165871787046670400523221	16517054888169441929133	03.03.2023 13:11:54	Returned	Desktop	Windows 10
167774011395668699995810	16777401139547982306203	03.03.2023 09:56:56	New	Desktop	Windows

## Аналитические виджеты

В UX Rocket можно настроить виджеты – быстрые графические отчёты для просмотра значения или динамики одного показателей. Для работы с виджетами перейдите в меню «Сводка». Сводка представляет собой панель настраиваемых отчетов, которые в графическом или табличном виде представляют данные набора Сессии.



Доступны следующие виды графиков/диаграмм:

- Линейный график
- Круговая диаграмма
- Круговая диаграмма стиль 2 (кольцо)
- Горизонтальная столбчатая диаграмма
- Вертикальная столбчатая диаграмма
- Область

Можно изменить порядок вывода отчетов, настроить отображение каждого из них на всю ширину либо на ½ ширины экрана.

Использование виджетов – это простой способ проанализировать работу вашего сайта на верхнем уровне: посмотреть статистику посещений, кликов, покупок и так далее.



## AI-модель для выявления проблем и точек роста

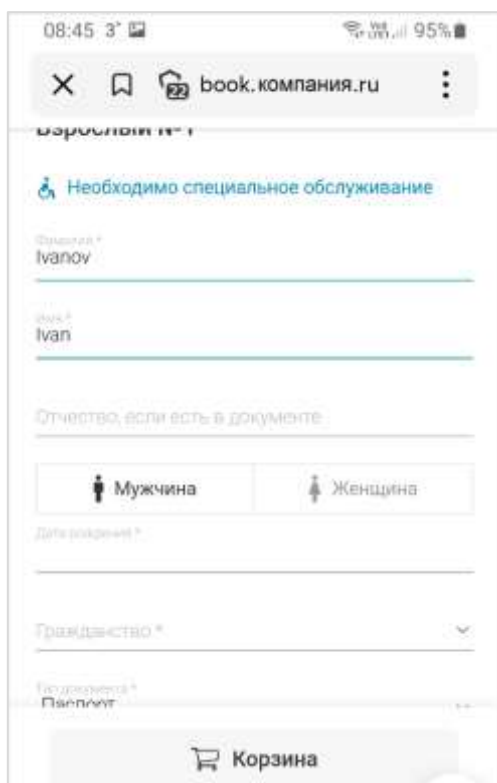
Как известно, доход компании в цифровом канале зависит от трафика, конверсии и суммы чека. Если с течением времени сумма чека меняется незначительно, то для повышения продаж надо увеличить трафик и/или конверсию (процент посетителей с покупкой).



Увеличить трафик можно за счёт рекламы, но это дорого и не всегда эффективно.

Почему дорого? Потому что увеличение ставок в рекламных кабинетах повысит трафик, но конкуренты тоже поднимут ставки, и в итоге трафик вернётся к первоначальным значениям, только платить за него все будут дороже.

Почему неэффективно? Давайте рассмотрим пример. Ниже показан экран покупки билета для смартфона. Клиент выбрал авиарейс и теперь хочет ввести ФИО, день рождения и паспортные данные. Казалось бы, что может быть сложного в заполнении этого экрана? Однако на Android покупки были почти нулевыми, тогда как на iPhone клиенты успешно покупали билеты. Проблема была в заполнении дня рождения. Ввод даты рождения с клавиатуры не был предусмотрен, а выбор даты в календаре оказался трудоёмким: надо ежемесячно прокрутить экран сотни раз (возраст \* 12 месяцев).



Очевидно, что без решения проблемы с календарём нет смысла привлекать клиентов рекламой: почти все пришедшие бросят покупку билета как неадекватно трудоёмкий процесс и купят позже с компьютера или купят у конкурентов. **Значит надо исправлять проблемные места, а уже потом повышать трафик!** Отличный план, но, чтобы найти проблемные места, нужны детальные данные и хороший аналитик. Здесь Вам на помощь приходит AI модель UX Rocket.

Ручной алгоритма поиска узких мест выглядит следующим образом:

- 1) Строим воронку по интересующему нас пути;
- 2) Выбираем стандартные и пользовательские параметры, которые могут влиять на конверсию;

- 3) Настраиваем в воронке соответствующие фильтры;
- 4) Применяем различные значения фильтров или их комбинацию и сравниваем со средней конверсией без фильтра.

Надо отметить, что ручной алгоритм выглядит достаточно трудоёмким, поэтому в систему интегрирована аналитическая модель, которая автоматизирует поиск узких мест. В меню «Рекомендации – Сегменты в воронке» выбираем воронку для анализа, уровень значимости при анализе данных и перечень параметров для передачи в модель.

Название  
Анализ воронки "Запрос демо" от 03-02-2023

Воронка: Запрос демо | Уникальные посещения: Да

Дата с: 01.01.2023 | по: 03.02.2023

Минимальный размер сегмента, %: 5 | Минимальное отклонение от ср.зн.: 5

Атрибуты: Browser x City x DeviceType x UserType x UtmCampaign x Resolution x

По выбранным параметрам система проводит автоматический расчёт, в рамках которого рассматривает наличие взаимосвязи между значениями атрибутов и конверсией по воронке в целом, а также между значениями атрибутов и потерей трафика на каждом из шагов.

Ниже представлен пример автоматического расчёта для воронки из 7 шагов (см. рисунок ниже, приведена выдержка, полный перечень рекомендации порядка 200 позиций). Видно, что для браузера Safari доля успешно завершивших покупок 4,39%, а в среднем по всем типам браузеров покупают 6,35% посетителей. Также есть проблемы с покупками на мобильном устройстве (Атрибут 1= «DeviceType» и Значение 1 = «Mobile Phone») и с покупками у новых посетителей (3% купивших среди новых против 10% при повторном посещении).

Атрибут 1	Значение 1	Атрибут 2	Значение 2	Шаг (0-N)	Доля от общего трафика на шаге	Доля завершивших	Средняя заверш
Browser	Safari			0-7	19.60%	4.39%	6.35%
Browser	Yandex			0-7	6.10%	11.35%	6.35%
DeviceType	Desktop			0-7	47.10%	8.73%	6.35%
DeviceType	Mobile Phone			0-7	51.58%	4.27%	6.35%
UserType	New			0-7	55.74%	3.02%	6.35%
UserType	Returned			0-7	44.25%	10.53%	6.35%
Browser	Chrome	DeviceType	Desktop	0-7	38.78%	7.77%	6.35%
Browser	Chrome	DeviceType	Mobile Phone	0-7	30.26%	4.33%	6.35%

Результаты анализа будут полезны при решении нескольких типов задач, например,

- 1) Поиск технических ошибок в работе сайта или мобильного приложения;
- 2) Адаптация сайта под конкретные сегменты посетителей для повышения продаж;
- 3) Анализ динамики показателей перехода и конверсии в течение времени (или по событиям).

# А/В тестирование и персонализация

## Подготовка к А/В тестированию

Важными функциональными модулями UX Rocket является сбор данных о поведении пользователей на сайте / в мобильном приложении и клиентская аналитика. Вы можете посмотреть «по каким путям посетители сайта приходят к покупке товара», «как выглядит воронка продаж», «где на пути к покупке возникает потеря трафика» и так далее. Данные доступны как в целом, так и по конкретным сегментам посетителей сайта.

Обратите внимание, что построение отчётов не требует предварительной работы на разметке сайта или настройке множества событий. Сбор данных об открытии страниц, кликах по ссылкам, в меню, нажатие на кнопки, показ всплывающих окон и ряда других событий начинается автоматически после установки тега на сайт или подключения SDK в мобильное приложение.

Вы можете построить необходимый вам отчёт и на основании исторических данных оценить возможный эффект от планируемых изменений на сайте. Например, одна из гипотез состоит в том, что «оформление заказа в один клик повышает конверсию, поэтому надо сделать эту кнопку более заметной для посетителей сайта». Исторические данные в UX Rocket позволяют Вам сравнить конверсию по двум путям: «кликнул кнопку и купил» и «купил без клика по кнопке». Вы видите лучший путь, понимаете возможный процент прироста конверсии (на основании исторических данных) и можете направить посетителей в нужную Вам сторону.

**Резюме:** перед проведением А/В и MVT-тестов желательно собрать и проанализировать информацию о поведении различных групп посетителей сайта и/или пользователей мобильного приложения. Это поможет выявить проблемные точки, а также сформировать и/или ранжировать ранее подготовленные гипотезы. Как результат, Вы сократите сроки и стоимость А/В-тестирования и получите лучший финансовый результат.

## Жизненный цикл кампаний

В UX Rocket бизнес-объект «Кампания» отвечает за тестирование и персонализацию контента. Данный бизнес-объект позволяет изменять внешний вид и расположение существующих элементов сайта и мобильного приложения, а для сайта позволяет добавлять на страницу новые элементы (с помощью javascript). Жизненный цикл по работе с кампаниями показан на схеме ниже:



Каждый аккаунт UX Rocket позволяет работать с двумя средами: среда разработки и продуктивная среда. В экранной форме «Кампании разработка» настраиваем А/В или MVT-тест. Проверяем корректность настроек в визуальном отладчике на сайте. Проверка новых/изменённых кампаний

выполняется в среде разработки и видна только пользователям с соответствующими правами в личном кабинете. Для переноса успешно проверенной кампании в продуктив её надо опубликовать. После публикации посетители сайта видят один из настроенных вариантов А/В или MVT-теста в соответствии с правилами показа и вероятностями отображения вариантов. Одному и тому же пользователю всегда показываем один и тот же вариант. Идентификацию пользователя выполняем по куки. Результаты работы кампании можно посмотреть в экранной форме «Кампании прод». По результатам анализа можно предпринять следующие действия:

- А. внести исправление в настройки кампании и опубликовать изменения;
- В. выбрать вариант победитель и переключить на него 100% трафика;
- С. остановить кампанию и отправить в архив.

Последний вариант может быть выбран в двух случаях:

- 1) проверенные варианты оказались хуже варианта по умолчанию и нет смысла направлять на них трафик;
- 2) один из вариантов оказался настолько успешным, что его надо перенести в код сайта, чтобы блокировщики рекламы и java-скриптов, используемые отдельными посетителями сайта, не помешали показу лучшего варианта.

## Кампании управления контентом

Кампании UX Rocket могут быть двух типов: для сайта и для мобильного приложения. Общая схема работы кампании в обоих случаях одинакова (см. рисунок ниже), но настройка кампании для сайта и мобильного приложения немного различается.



Управление контентом на сайте происходит с помощью java-скриптов. Система UX Rocket передаёт необходимые скрипты на сайт. Браузер пользователя исполняет полученные скрипты, и в результате происходит изменение внешнего вида страницы сайта, а также сохранение в базу данных статистики, данных по кликам, покупкам пользователя и так далее.

В UX Rocket можно использовать следующие типы скриптов:

- Скрипты вариантов – скрипт содержит программный код для отображения одного из вариантов внешнего вида страницы или её элемента(ов).
- Скрипты кампаний – позволяют установить значение пользовательских атрибутов для конкретной кампании. Атрибуты можно использовать при настройке правил показа и при работе с отчётами для получения детальной аналитики.
- Скрипты действий – скрипт содержит программный код для сохранения в базу данных какого-то из отслеживаемых событий или значений.

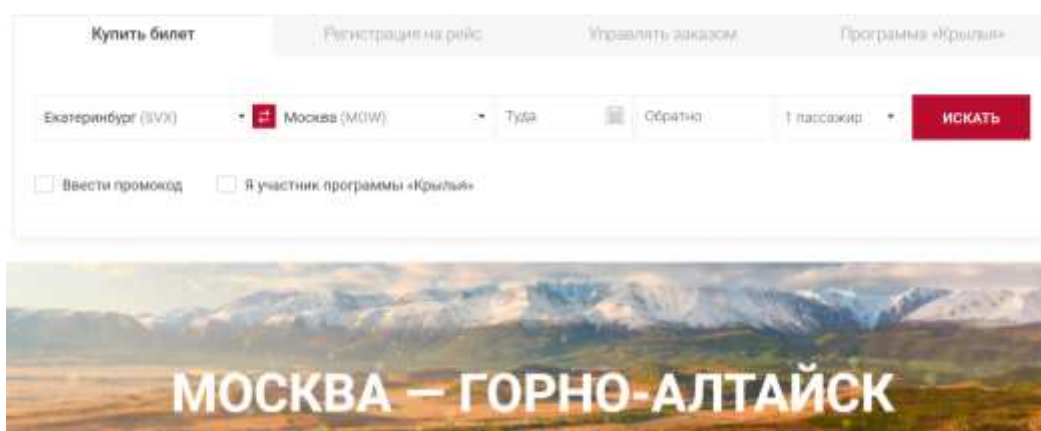
Управление контентом в мобильном приложении не использует java-скрипты, так как мобильное приложение не может исполнять сторонний код. Вместо скриптов вариантов Вы получаете доступ к редактированию свойства страниц, полей, картинок, кнопок и других существующих элементов мобильного приложения. В мобильном приложении создавать новые элементы для А/В-тестирования средствами UX Rocket нельзя. Сначала надо запрограммировать новый элемент в мобильном приложении, а уже потом можно управлять его видимостью и свойствами в UX Rocket.

## Элементы и варианты

Цель кампании по тестированию – это показ несколько разных варианты контента для посетителей сайта и определение победителя по одной из метрик (клики, покупки, другое).

В кампании UX Rocket для визуального отображения разных вариантов контента надо настроить элементы и варианты. Элемент кампании — это логическое понятие, которое описывает один или группу визуальных элементов на сайте или в мобильном приложении. Вариант – это один из способов визуального отображения элемента.

В качестве примера рассмотрим следующую страницу сайта.



Предположим, вам надо определить какая надпись на кнопке «Искать» или «Купить» приведёт к большему числу нажатий на кнопку. Тогда в кампании можно завести элемент с названием «Button» и два варианта показа, отличающиеся надписью на кнопке – «Искать» и «Купить». В нашем примере мы могли бы назвать элемент не «Button», а «Element\_1», это не влияет на работу кампании. Наименование элементов и вариантов необходимо заводить латинскими символами (допустимы буквы латинского алфавита, цифры и нижнее подчеркивание).

При проведении А/В-тестов надо завести один элемент и несколько вариантов для него. В UX Rocket можно настроить до 10 вариантов для каждого элемента. В рамках каждого варианта количество изменяемых свойств у визуальных компонент (надписи, кнопки, ссылки, рисунки и так далее) сайта или мобильного приложения не ограничено.

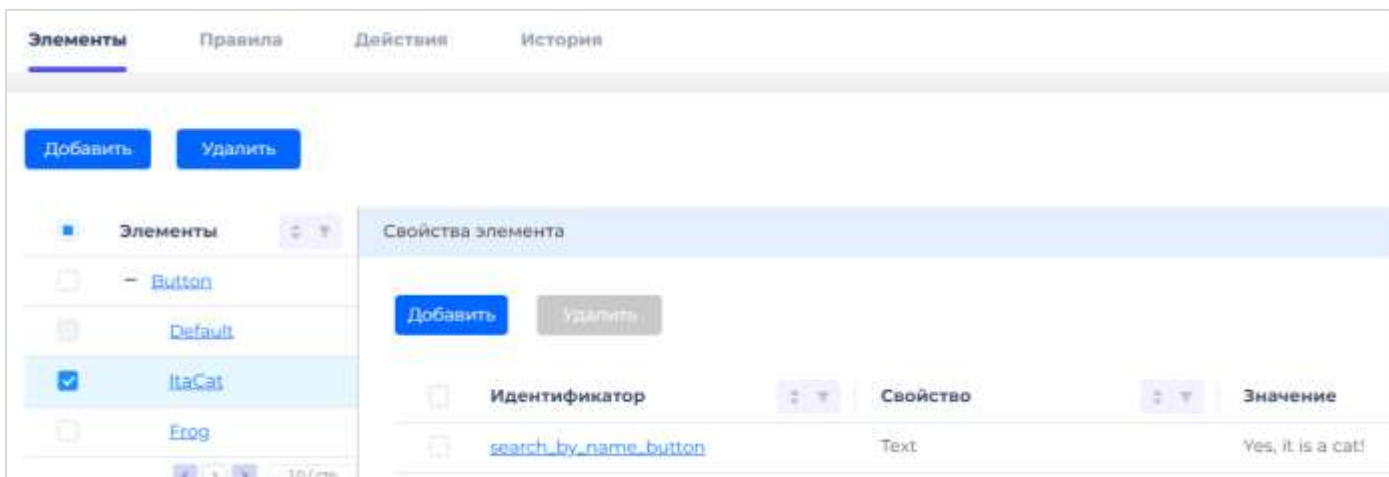
Предположим, что одновременно с тестированием надписи на кнопке мы хотим сравнить конверсию с тремя вариантами картинки: горы, река и кемпинг. В этом случае у нас два варианта у первого элемента и три варианта у второго элемента. Так как тестировать будем одновременно, то получается  $3 \times 2 = 6$  вариантов пользовательского опыта:

- Кнопка «Купить» и «горы»;
- Кнопка «Купить» и «река»;
- Кнопка «Купить» и «кемпинг»;
- Кнопка «Искать» и «горы»;
- Кнопка «Искать» и «река»;
- Кнопка «Искать» и «кемпинг».

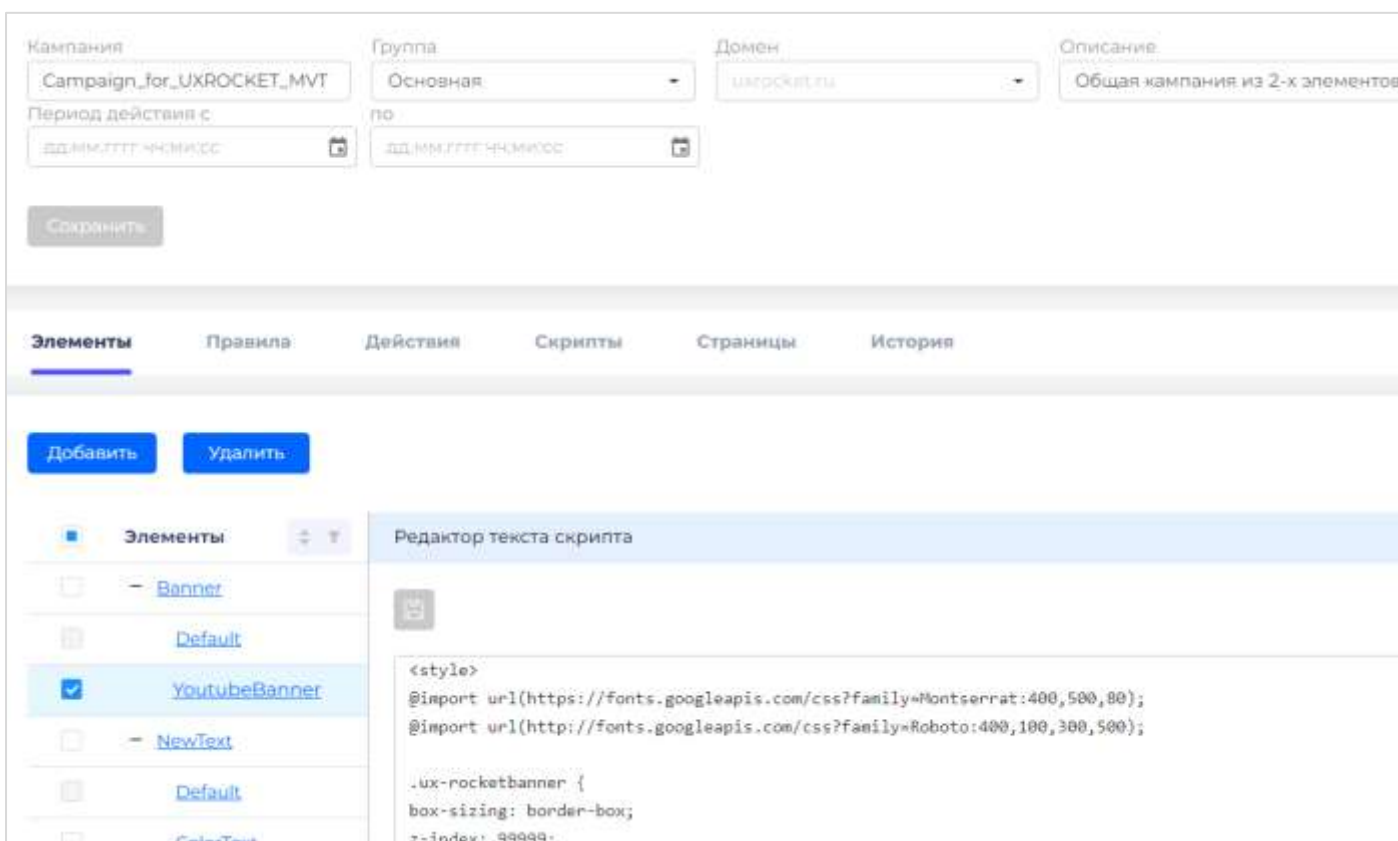
Такие тесты называются многовариантными или «MVT тестирование» (multi variant testing). Вы настраиваете перечень вариантов, а система формирует все доступные сочетания по разным элементам.

При проведении многовариантных тестов – MVT-тестирование – необходимо завести несколько элементов и варианты к каждому из элементов. В одной кампании можно завести до 10 элементов, в каждом из элементов можно завести от 2 до 10 вариантов.

На рисунке ниже показан экран системы с настройками для проведения A/B-теста для мобильного приложения: разные варианты надписи на кнопке.



На рисунке ниже показан экран системы с настройками для проведения MVT-теста на сайте: сравнение эффекта с двумя одновременно идущими экспериментами на странице: изменение описания продукта и всплывающее окно (баннер) со ссылкой на вебинар.




## Правила показа вариантов

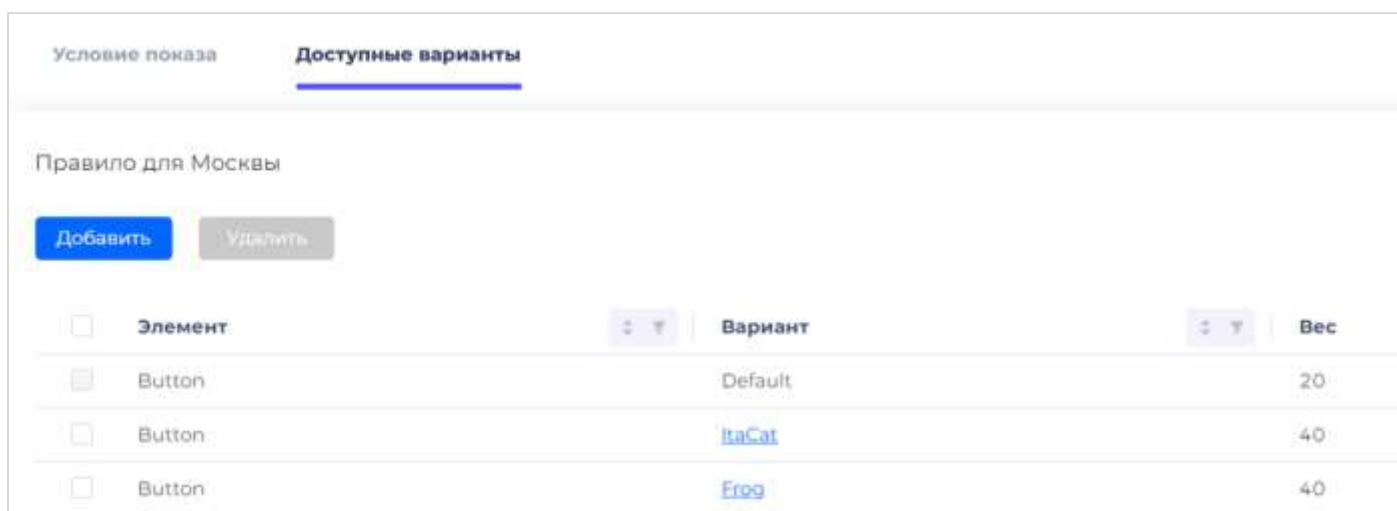
В UX Rocket кампании по управления контентом можно запускать как для всех посетителей сайта, так и для конкретных групп посетителей. Ограничить аудиторию для тестирования можно с помощью правил показа вариантов.

В кампании может быть от 1 до 10 правил показа. Правила имеют название, приоритет (порядок проверки), а также содержат условия показа и доступные к показу варианты.

Предположим, вам надо провести A/B тестирование только для посетителей из Москвы. В этом случае Вы заводите одно правило и настраиваете в нём одно условие показа: геолокация пользователя – это город Москва (см. рисунок ниже).



Дополнительно вы указываете долю посетителей сайта, которые увидят каждый из вариантов. На экране ниже настроено 3 варианта названием кнопки в одном из окон мобильного приложения и весами (доля посетителей для показа) 20, 30 и 50%.



<input type="checkbox"/>	Элемент	Вариант	Вес
<input type="checkbox"/>	Button	Default	20
<input type="checkbox"/>	Button	<a href="#">ItaCat</a>	40
<input type="checkbox"/>	Button	<a href="#">Frog</a>	40

Если в кампании настроено более одного правила, то они проверяются последовательно в порядке убывания приоритета (самым высоким считается первый приоритет). Если для правила все условия показа выполнены, то варианты к показу и их веса будут выбраны из этого правила, а дальнейший просмотр правил будет прекращён. Если параметры текущей сессии не попадают ни под одно правило, то данный посетитель не участвует в тестировании.

В программе реализован следующий алгоритм определения варианта к показу:

- 1) Получить с сайта или из мобильного приложения параметры сессии и пользователя.
- 2) Определить подходящее правило (первое правило, где выполнены все условия показа);
- 3) Если ранее посетитель видел один из вариантов (доступный в выбранном правиле), то показываем ему тот же самый вариант, иначе бросаем монетку (генерация случайного числа от 0 до 1) и по результатам определяем вариант к показу.

Обратите внимание, что в разных правилах Вы можете сравнивать различные варианты контента. Предположим, Вы тестируете четыре варианта баннеров: два для мужчин и два для женщин. Тогда Вы

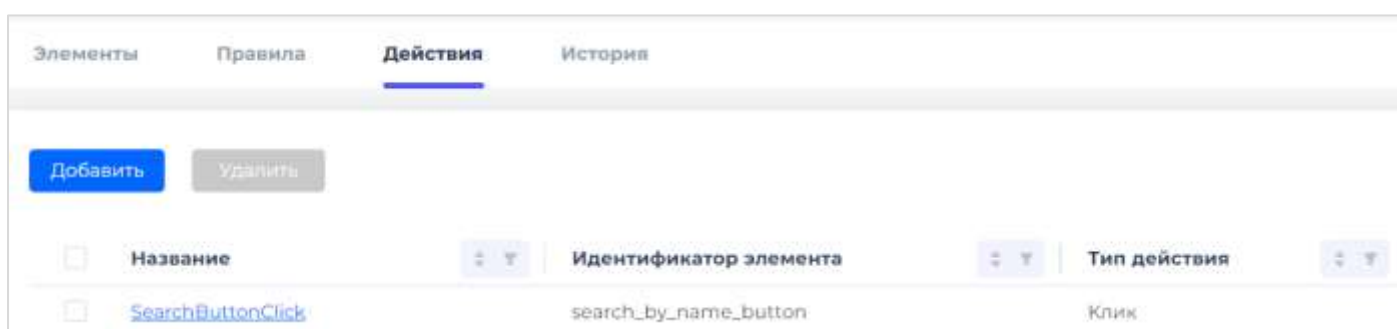
заходите два правила. Первое правило проверяет, что на сайте мужчина, и если ДА, то показывает один из двух баннеров для мужчин. Второе правило проверяет, что на сайте женщина, и если ДА, то показывает один из двух баннеров для женщин.

Примечание: при настройке правил показа можно использовать стандартные параметры сессии (геолокация, тип устройства, операционная система, модель смартфона и некоторые другие) и пользовательские атрибуты. Пользовательские атрибуты настраиваются для аккаунта в целом, всего можно настроить до 30 пользовательских атрибутов. (см. меню «Сегменты -> Атрибуты»)

## Целевые действия

Для определения кампании-победителя (для A/B и MVT-тестов) и суммы достигнутого финансового результата необходимо настроить целевые действия.

В мобильном приложении целевым действием может быть переход на страницу (например, диалоговое окно или страница «Спасибо за покупку») или клик по одному из элементов интерфейса (обычно нажатие на кнопку). Выбор соответствующего значения доступен в выпадающем списке.



На сайте перечень отслеживаемых событий значительно шире, поэтому для настройки события используются java-скрипты.

После настройки действий система сможет автоматически измерять процент конверсии по каждому из вариантов (как число совершивших целевое действие делёное на число просмотревших вариант) и сумму покупки (сумма покупки или сумма дополнительной покупки или иное).

Посмотреть показатели конверсии и сумму покупки Вы можете в экранной форме «Кампании прод».

## Настройка скриптов для сайта

*Важно: Положения данного раздела применимы только к кампаниям для сайта. Для мобильного приложения скрипты не используются (заполнение переменных происходит путём написания программного кода при подключении SDK).*

При работе с сайтом кампаниям надо получать значения стандартных и пользовательских параметров. Для решения этой задачи в программе настраивают два типа java-скриптов: общие скрипты и скрипты кампаний. Общие скрипты можно посмотреть и изменить в меню «Сегменты -> Общие скрипты». Скрипты кампаний можно посмотреть и изменить в окне «Кампании разработка» на вкладке «Скрипты». В системе реализован следующий порядок выполнения скриптов: при открытии страницы сайта система в произвольном порядке выполняет общие скрипты, привязанные к данной странице, а потом в асинхронном режиме выполняет скрипты кампания, привязанные к данной странице.

Обычно скрипт вычисляет стандартные параметры сессии и значения атрибутов и сохраняет их в куки (или наоборот очищает их значение в куки).



На рисунке ниже показано, как выглядит настройка общего скрипта. Встроенный редактор позволяет добавить текст скрипта, просмотреть его и выполнить необходимые корректировки.



Для скрипта указывается домен (у клиента может быть несколько сайтов или микро-сайтов) и страницы либо маски страниц, на которых скрипт должен отработать при открытии страницы.



После завершения настройки скрипты проверяются в режиме отладки и публикуются для выполнения в продуктивной среде. Работающий скрипт может быть отредактирован и опубликован в новой редакции. В UX Rocket можно посмотреть историю публикаций и предыдущие версии скрипта.

Принципы настройки и использования скриптов кампании аналогичны общим скриптам, но они выполняются в рамках жизненного цикла конкретной кампании.

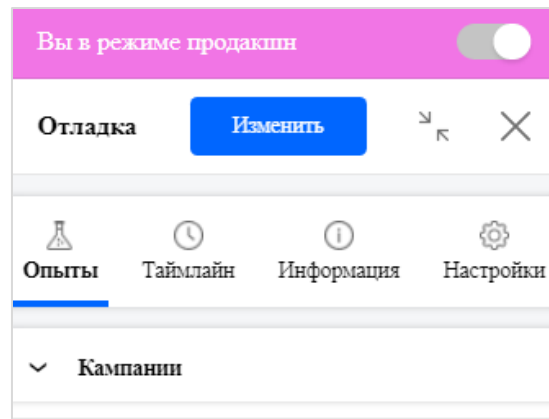
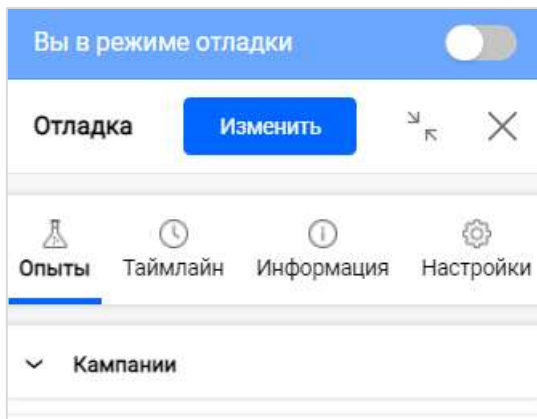
## Многостраничные кампании

В UX Rocket можно настраивать одностраничные и многостраничные кампании. В рамках многостраничных кампаний можно согласованно менять контент сразу на нескольких страницах сайта. Мы рекомендуем переходить к настройке многостраничных компаний после того, как Вы пройдете соответствующий учебный курс с нашим специалистом.

## Отладка кампаний

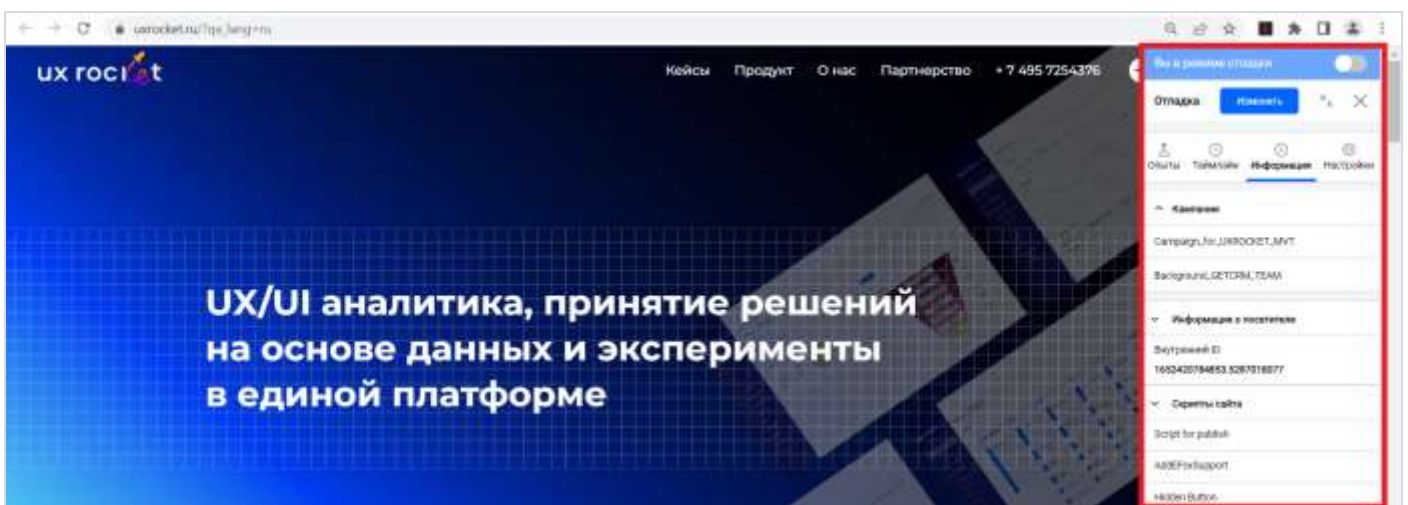
На момент написания документа отладчик доступен только для работы с кампаниями для сайта. Запустить отладчик можно из личного кабинета меню «А/В тестирование -> Отладка».

Как вы помните, UX Rocket работает с двумя средами: среда разработки и продуктивная среда. Поэтому отладчик имеет переключать между средами (см. рисунок ниже).



То есть Вы можете отлаживать кампании как до публикации, так и после публикации.

Для работы с отладчиком необходимы права доступа к меню «Отладка». После выбора пункта меню «Отладка» вы выбираете сайт, на котором хотите запустить отладчик. Внешний вид окна отладчика показан на рисунке ниже.



Окно «Отладчика» имеет 4 вкладки и поддерживает следующие функции:

- Просмотр списка активных кампаний и показанных вариантов – вкладка «Опыты»;
- Просмотр информации о списке кампаний на странице, списке общих скриптов и значения атрибутов – вкладка «Информация»;
- Изменение значения атрибутов и отображение варианта кампании для новых значений атрибутов – кнопка «Изменить», далее вкладка Атрибуты;
- Изменение значения варианта и принудительный показ страницы с данным вариантом (вне зависимости от его соответствия правилам показа) – кнопка «Изменить», далее вкладка Варианты;
- Сброс всех ранее выполненных в отладке настроек (система очищает куки отладчика);
- Изменение местоположение окна «Отладка» на экране и настройка перечня событий для отображения в Таймлайн – вкладка «Настройки»;
- Просмотр хронологии отображения вариантов и выполнения действий – вкладка «Таймлайн»;
- Переключение между средой разработки и продуктивной средой.

Обращаем внимание, что работа с кампаниями в отладчике не попадает в отчёты по кампаниям. Это сделано для того, чтобы тестирование кампаний в продуктивной среде не искажало результаты тестирования.

## Публикация кампаний

После настройки и проверки работы кампании в среде разработки её надо опубликовать для начала тестирования гипотез и/или отображения персонального контента для посетителей сайта.

На экранной форме публикации можно остановить ранее запущенные кампании или опубликовать новую версию кампании (значение поля «Обновление доступно» ДА означает, что после публикации кампании на продуктив для этой кампании в среде разработки были внесены исправления по сравнению с текущими настройками кампании в продуктиве).

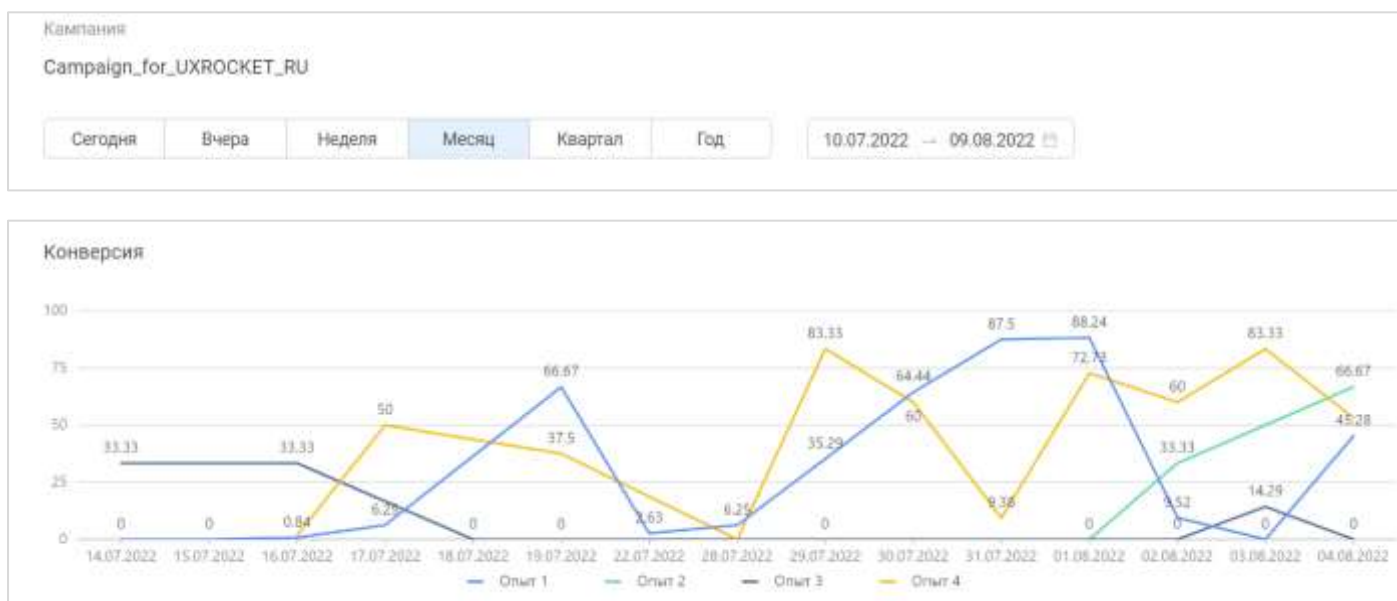
Публикация						
<input type="button" value="Опубликовать"/> <input type="button" value="Приостановить"/> <input type="button" value="Завершить"/>						
Название	Статус	Дата	Автор	Обновление доступно	История	
Background_GETCRM_TEAM	В Разработке	16.05.2022	UXROCKET	Да	<a href="#">История</a>	
Campaign_for_UXROCKET_MVT	Опубликована	16.11.2022	UXROCKET	Да	<a href="#">История</a>	

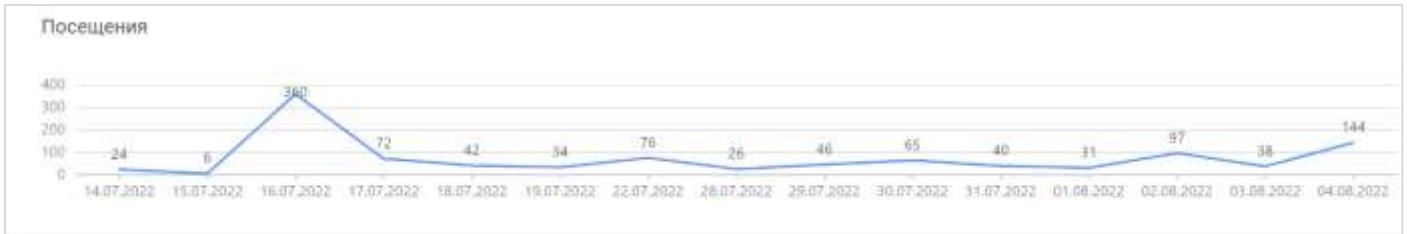
Система хранит историю публикаций с указанием даты публикации/остановки кампании, автора публикации и описание причины. Просмотреть данные можно по гиперссылке «История».

## Анализ результатов

Результаты работы опубликованной ранее кампании можно посмотреть в меню «А/В тестирование -> Кампании прод». В этом окне показан список опубликованных кампаний. Из списка можно перейти в экранную форму отчёта и по конкретной кампании посмотреть результаты её работы за интересующий период. В частности, можно определить: «достигнута ли заданная достоверность теста» и «какой вариант стал победителем». Ответ на подобные вопросы даёт встроенная статистическая модель оценки результатов теста, которая подробно рассмотрена в разделе ниже.

Экранная форма результатов работы кампании состоит из 4-х областей: область фильтра, интерактивный график с результатами тестирования вариантов, график с информацией о количестве посещений и таблица со сводными данными об эффективности (см. рисунок ниже).





Опыт	Посещения	Целевые действия	Конверсия	Погрешность	Доход
Опыт 1	724	144	19,89%	0,00%	8064
Опыт 2	53	9	16,98%	0,00%	946
Опыт 3	Опыт 4	33	80,49%	0,00%	1411
Опыт 4	Banner: Default BackGround: GetCRMTeam	88	31,10%	0,00%	3506
Итого:		274	37,12%	0,00%	13927

Отчёт о работе кампании построен на основании детальной таблицы результатов. В этой таблице хранятся записи о показе каждого варианта по каждой из кампаний, а также данные о достижении целей по каждой из кампаний. Одна строка детальной таблицы результатов – это запись о показе конкретных вариантов для одной отдельно взятой кампании (для каждого перехода на страницу или обновления) или данные о достижении одной из целей кампаний. Например, если на странице запущено 5 кампаний, то при переходе на страницу в таблице статистики будет сохранено 5 записей (отдельная запись для каждой кампании).

При А/В тестировании каждый из Опытов в отчёте соответствует одному из вариантов тестирования. При MVT-тестировании Опытом посетителя будем комбинация тестов, которая была показана при отображении страницы. При наведении мышки на каждый из опытов Вы можете посмотреть какая комбинация вариантов ему соответствовала.

## Статистическая оценка показателей теста

Важной частью UX Rocket является встроенный инструмент для расчета, анализа и прогнозирования статистических показателей теста.

Каждый опыт является гипотезой, которую мы проверяем в процессе тестирования. Вспомним приведенный выше пример с вариантами надписи на кнопке - «Искать» или «Купить». Мы хотим понять, какой эффект даст в перспективе каждый из вариантов. Критерием для выявления лучшей гипотезы будем считать величину конверсии, полученную в результате её применения.

Для выбора лучшего опыта из нескольких производится их попарное сравнение с эталонным. Среди тестируемых на текущий момент опытов всегда будет наихудший – это опыт, показавший на момент расчета наименьшую конверсию. Опыт с наименьшей конверсией можно принять за эталонный и попарно сравнить с ним остальные опыты.

Чтобы учесть влияние случайных отклонений, UX Rocket определяет статистическую достоверность полученных результатов. Это делается путём расчета показателя «Вероятность ошибки» и его сравнения с заданным критерием, который составляет 5%. Расчет достоверности производится за весь активный период работы кампании, с момента её публикации. Если при сравнении двух опытов получена вероятность ошибки менее 5%, результат сравнения считается достоверным, и опыт с большей конверсией становится победителем. При достижении достоверных результатов сравнения нескольких опытов победителем будет вариант с максимальной конверсией.

Рисунок ниже иллюстрирует пример, когда из двух опытов выявлен уверенный победитель, с вероятностью ошибки, близкой к нулю. При наведении курсора на соответствующее значение можно посмотреть статистику посещений и целевых действий, а также конверсию за весь период работы кампании.

Опыт	Посещения	Целевые действия	Конверсия	Вероятность ошибки	Должительность
Опыт 1	86654	52352	60.41%	0.00 %	
Опыт 2	550	274	49.82%	N/A	
<b>Итого:</b>	<b>87204</b>	<b>52626</b>	<b>55.12%</b>		

Опыт 1

Посещения: 86654

Целевые действия: 52352

Конверсия: 60.41 %

Вероятность ошибки: 0.00 %

Если при сравнении двух опытов получена вероятность ошибки 5% и более, то необходимая достоверность теста не достигнута. В этом случае статистическая модель рассчитывает, сколько времени требуется, чтобы получить достоверный результат. Если срок оказывается слишком большим, то решение о дальнейших действиях принимается экспертным путем: гипотезы могут быть пересмотрены, изменены правила показа, кампания может быть завершена и т.д.

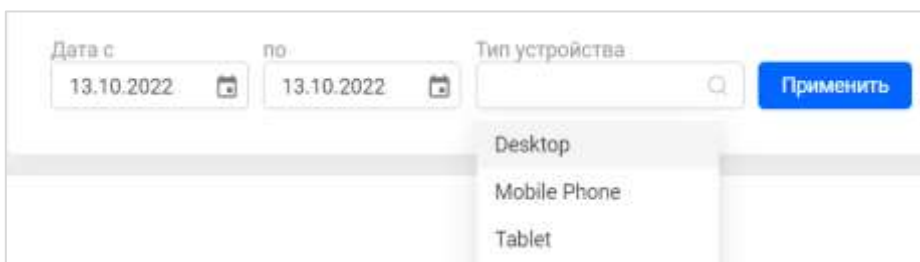
Достоверность результатов тестирования можно посмотреть в двух режимах:

- регулярное задание считает достоверность для кампании в целом и выдаёт рекомендации;
- в окне просмотра результатов работы кампании пользователь может выбрать любой срез по стандартным и пользовательским атрибутам и увидеть результаты тестирования в этом разрезе.

Как пример проведём анализ достоверности A/B теста в разрезе типа пользовательского устройства: мобильное устройство и десктоп. Для этого добавим этот фильтр в параметрах настройки кампании



и посмотрим результат работы кампании сначала в целом, а потом по типу устройства.



Мы видим, что для кампании в целом достигнуто статистически значимое различие между опытами.

Данные					
Опыт	Посещения	Целевые действия	Конверсия	Вероятность ошибки	Прогноз продолжительности
Опыт 1	403	215	53.35%	0.00 %	N/A
Опыт 2	414	191	46.14%	N/A	N/A
<b>Итого:</b>	<b>817</b>	<b>406</b>	<b>49.74%</b>		

Но если анализировать показатели с учетом типа устройства, то картина будет отличаться. Для мобильных устройств наша гипотеза работает хорошо:

Данные					
Опыт	Посещения	Целевые действия	Конверсия	Вероятность ошибки	Прогноз продолжительности
Опыт 1	161	45	27.95%	0.00 %	N/A
Опыт 2	184	46	25%	N/A	N/A
<b>Итого:</b>	<b>345</b>	<b>91</b>	<b>26.48%</b>		

Для настольных компьютеров результат нельзя считать достоверным, поскольку выборка данных недостаточна (вероятность ошибки 6.8%).

Данные					
Опыт	Посещения	Целевые действия	Конверсия	Вероятность ошибки	Прогноз продолжительности
Опыт 1	236	166	70.34%	6.80 %	2 дн. 21 ч.
Опыт 2	229	145	63.32%	N/A	N/A
<b>Итого:</b>	<b>465</b>	<b>311</b>	<b>66.83%</b>		

Таким образом, для настольных компьютеров надо продолжить проверку гипотезы до достижения статистической значимости.

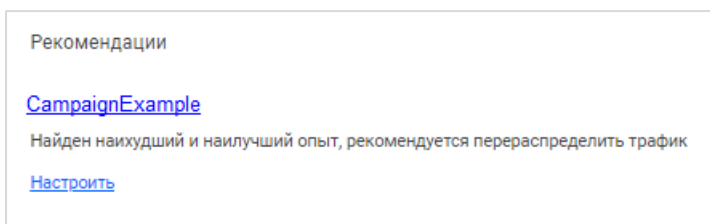
Мы рекомендуем для всех A/B тестов проводить детальный анализ результатов по ключевым сегментам, не ограничивая рассмотрение результатом работы кампании в целом. Это позволяет улучшать успешные тесты и находить интересные инсайты в неудачных тестах (в целом гипотеза не оправдалась, но есть сегмент клиентов, где достигнут большой прирост в конверсии).

## Рекомендации

На практике, клиенты запускают много тестов, но редко посещают раздел статистики. В результате легко пропустить момент, когда результаты теста показывают отличный результат или наоборот быстро уходят в минус. Преимуществом системы UX Rocket являются автоматические рекомендации по изменению правил показа в момент достижения 95% достоверности тестов.

UX Rocket следит за результатами A/B тестирования и выдаст свои рекомендации прежде, чем неудачный тест приведёт к финансовым потерям для вашего бизнеса.

Для просмотра рекомендаций необходимо перейти в меню «A/B тестирование -> Рекомендации». На экране будет показан список доступных рекомендаций по всем активным кампаниям. В рекомендации сообщается о выявлении наихудших и наилучших опытов и необходимости перераспределения трафика между вариантами.



По клику на гиперссылке вы можете перейти от рекомендации к просмотру результатов в карточке кампании, а по клику на ссылке «Настроить» - на экран «Кампании разработка» (при наличии соответствующих полномочий).

При большом количестве работающих кампаний функционал рекомендаций удобен тем, что позволяет оперативно отреагировать на результаты тестов и внести необходимые изменения.

## Профиль клиента

В данном разделе описаны возможности UX Rocket по идентификации клиентов, объединению профилей, загрузке данных, настройке онлайн и офлайн событий, расчёту метрик.

Настройка клиентского профиля позволяет расширить возможности UX Rocket по клиентской аналитике и A/B-тестированию, упрощает персонализацию цифрового контента и активацию данных.

**Обращаем внимание, что модуль клиентского профиля не входит в базовую поставку UX Rocket и подлежит отдельному лицензированию.**

Для работы с персональными данными рекомендовано использовать Enterprise версию продукта или хэшировать их при загрузке в систему.

## Типы идентификаторов клиента

Для идентификации клиентов можно использовать различные идентификаторы, например: телефон, email, код в CRM, код в браузере (куки) и т.п. Идентификаторы можно разделить на два типа:

- жесткий идентификатор (не может изменяться/переходить от одного клиента к другому);
- мягкий идентификатор (может изменяться/переходить от одного клиента к другому).

Деление условно в том смысле, что каждая компания сама решает какой из идентификаторов считать неизменным. Например, первоначально телефон считали основным идентификатором клиента (если клиент сменил телефон, то это новый клиент), но потом бизнес попросил считать основным идентификатором код из CRM (чтобы при смене телефона история операций сохранялась). То есть, сначала телефон был жестким идентификатором, а потом телефон стал мягким, а код CRM – жёстким.

Типы идентификаторов и определение мягкости и жёсткости настраивает служба поддержки UX Rocket в административном модуле системы. Конечные пользователи видят текущие настройки, но не могут их изменить без обращения в службу поддержки. На рисунке ниже показаны настройки типов идентификаторов для демонстрационной базы.

Идентификаторы				
Наименование	Тип	Нижний регистр	Важность	Хэш
ВКонтакте	Жёсткий	Нет	10	Нет
Телефон	Жёсткий	Да	20	Нет
Почта	Мягкий	Да	30	Нет
Куки	Мягкий	Нет	40	Нет

В одном аккаунте можно определить несколько жёстких идентификаторов (например, facebook\_id и vk\_id), но каждый клиент имеет только одно значение для каждого типа жесткого идентификатора (нельзя иметь два vk\_id, если это жёсткий идентификатор).

В одном аккаунте клиент может иметь несколько мягких идентификаторов одного типа (например, куки у одного клиента в разных браузерах и на разных устройствах будут разными). У одного клиента количество мягких идентификаторов одного типа ограничено значением 50. При появлении у клиента нового мягкого идентификатора одного типа сверх 50 значений мы добавляем новый идентификатор и одновременно удаляем идентификатор, который не использовался дольше всего.

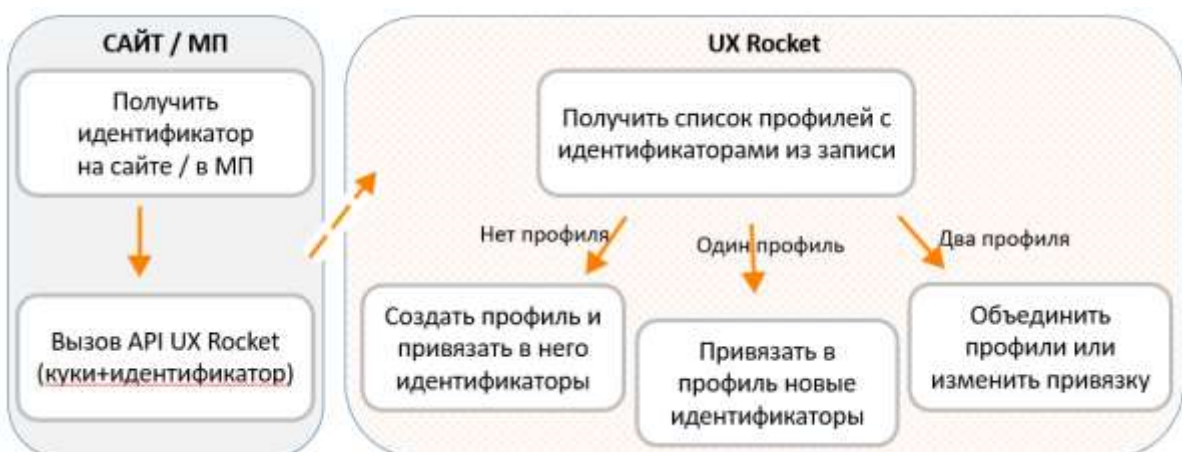
В ряде случаев желательно приводить идентификаторы к нижнему регистру. Например, без нижнего регистра почтовые адреса [Aa@mail.ru](mailto:Aa@mail.ru) и [aa@mail.ru](mailto:aa@mail.ru) будут разными идентификаторами и могут привести к двум разным записям клиентов. За приведение к нижнему регистру отвечает одноимённое поле в интерфейсе системы.

При создании нового профиля клиента необходимо указать хотя бы один идентификатор (мягкий или жёсткий), который не закреплён за другим клиентом. Нельзя удалить/переименовать какой-либо идентификатор после его присвоения клиенту. Исключение составляют две процедуры: объединение клиентов и разрешение конфликтов идентификации.

## Идентификация в цифровых каналах

Существует ряд сценариев по идентификации клиента в цифровых каналах. Например, клиент входит в личный кабинет на сайте или в МП, переходит на сайт со специальным пикселем из рассылки, заказывает звонок менеджера, подписывается на рассылку новостей и так далее.

Во всех этих сценариях надо получить на стороне сайта или МП уникальный идентификатор и через API сохранить его в UX Rocket (точнее сохранить пару «куки»-«уникальный идентификатор»).





После идентификации клиента вы сможете:

- узнавать клиента при повторных посещениях сайта без логина в личный кабинет;
- видеть сквозную историю по клиенту в разных браузерах и на разных устройствах;
- сопоставлять онлайн и офлайн данные по клиенту;
- использовать данные клиентского профиля для персонализации и А/В-тестирования.

## Fingerprint и кросс-браузерная идентификации

При входе посетителя на сайт UX Rocket проверяет есть ли данные об этом посетителе. Для этого система ищет в session storage или в куки ключ gcUserId. Если поле не заполнено, то система сгенерирует уникальный код и запишет его в этот ключ. Если поле заполнено, то значение из поля будет передаваться в UX Rocket как код посетителя (куки). В некоторых ситуациях данный алгоритм не позволит узнать клиента при повторном посещении сайта:

- браузер запущен в режиме инкогнито;
- браузер принудительно чистит данные сессии;
- клиент почистил историю (куки и данные сайтов);
- повторный заход выполнен из другого браузера.

В перечисленных выше случаях, как правило, можно узнать посетителя по цифровому идентификатору его устройства - fingerprint.

В UX Rocket реализован расчёт цифрового отпечатка устройства, что позволяет с высокой вероятностью узнавать повторные заходы клиента на сайт в режиме инкогнито, после удаления куки, при входе из другого браузера (за исключением FireFox, доля которого на рынке всего ~ 3%) и в некоторых других случаях.

Для расчёта fingerprint по посетителям сайта и сопоставлению профилей на основании этого идентификатора необходимо обратиться в службу поддержки UX Rocket для подключения данного функционала. **Обращаем внимание, что поддержка fingerprint не входит в базовую поставку UX Rocket и подлежит отдельному лицензированию.**

## Карточка профиля клиента

В данном разделе мы называем клиентом физическое лицо, взаимодействующее с цифровыми ресурсами организации и точками её продаж в офлайн для получения информации о товарах, их покупки и последующего сервисного обслуживания.

Профиль клиента в UX Rocket содержит информация о клиенте, которую удалось собрать в цифровых ресурсах, загрузить из сторонних систем или рассчитать (события, метрики и сегменты) на основании ранее перечисленных данных.

Информацию из профиля клиента можно передавать на сайт для персонализации контента, передавать в сторонние системы для активации данных (например, в рекламный кабинет Яндекс), а можно использовать при А/В-тестировании или построении аналитических отчётов.

В UX Rocket экранная форма «Профиль клиента» состоит из двух вкладок:

- Вкладка «Профиль» со списком всех клиентов и возможностью поиска клиента по любому идентификатору (телефон, email, куки и так далее);
- Вкладка «Параметры» с настройками внешнего вида карточки клиента.

Обращение	Телефон	Почта	Статус	Описание
<a href="#">Посетитель:1686904923824.9598894196</a>			Активен	
<a href="#">Посетитель:1686902779755.9146195163</a>			Активен	
<a href="#">Посетитель:1686902126236.8102004231</a>			Активен	

Из списка клиентов можно по гиперссылке перейти в карточку клиента, где представлена краткая информация по клиенту, детальные данные по клиенту, идентификаторы клиента, события, метрики и сегменты. Внешний вид карточки клиента показан на рисунке ниже.

**Обзор** | Детально | Метрики | Сегменты | Покупки | Идентификаторы

Обращение: Воробьев Михаил

Дата рождения: 17.02.1979

Пол: Мужской

Телефон: 77696013392

Почта: vorobev.mikhail@gm...

**События**

Месц | Год | Все

Москва  
12:29 16 июня

☀️ 26.7°C 034%

Покупки всего 0% ↑

**28**

Доход всего

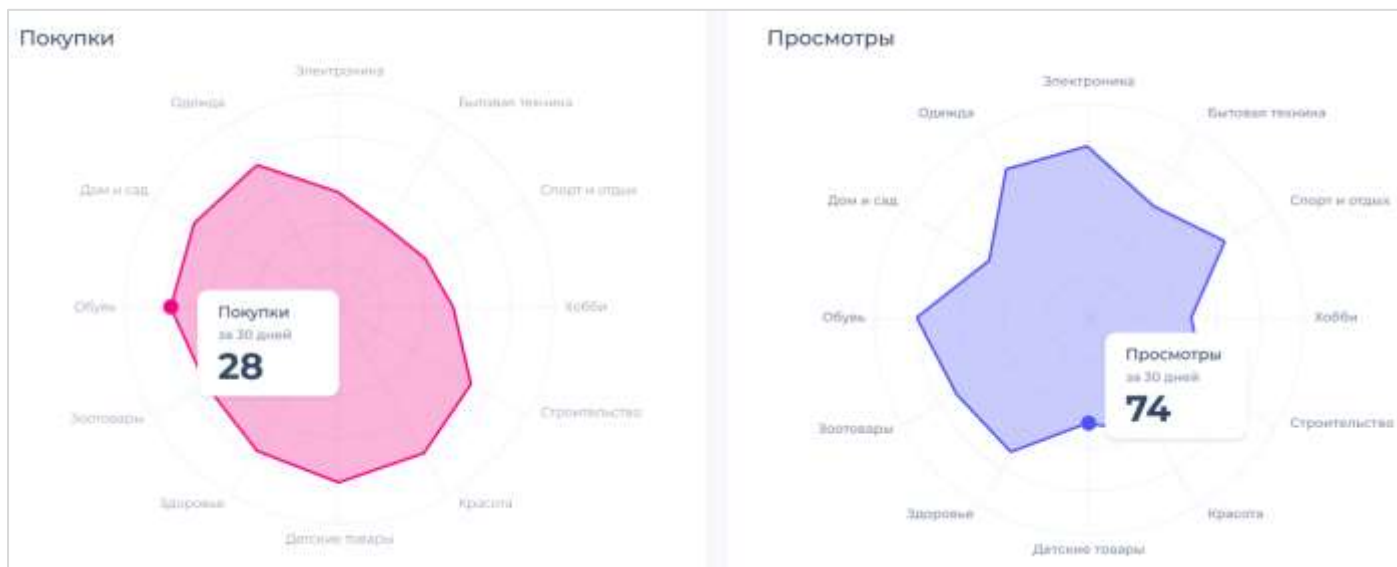
**120 000**

Событие	Время	Описание
<b>В Событий 10 апреля 2023</b>		
Покупка - заголовок чека	14:21	Покупка онлайн, 5000р
Покупка - позиция чека	14:21	Покупка /zhenskaja-maika-shorty, 2000р
Покупка - позиция чека	14:21	Покупка /futbolka-zhenskaja-modellini, 3000р
Просмотр страницы	14:19	Спасибо за покупку /payment_ok
Просмотр страницы	14:18	Корзина /cart
Просмотр страницы	14:18	Просмотр товара /zhenskaja-maika-shorty
Просмотр страницы	14:18	Просмотр товара /futbolka-zhenskaja-modellini
Просмотр страницы	14:18	Просмотр товара /futbolka-zhenskaja773

На экране «Обзор» показаны:

- основные поля клиента (обращение, день рождения, пол, телефон, email);
- график и таблица по событиям клиентского профиля (настройка событий описана в разделе «Правила создания событий»);
- виджет с температурой, связанный с ползунком графика событий;
- ключевые метрики по клиенту, опционально с индикатором динамики изменения показателя или индикатором сегмента (настройка на вкладке «Параметр» в списке клиентов).

На вкладке покупок можно посмотреть историю покупок клиента и графически сопоставить просмотры товара на сайте с офлайн и онлайн покупками.



На вкладке идентификаторы клиента: собранный онлайн и загруженные из других систем, в том числе перенесённые из других карточек при объединении профилей и/или разрешении конфликтов.

На вкладке метрики представлены значения метрик по клиенту, расчётных и прогнозных. Показана динамика метрик за последние 12 месяцев.

На вкладке сегментов показаны все сегменты, в которые был отнесён клиент. Сегменты можно загружать из других систем или рассчитывать в UX Rocket на основании данных по клиенту, его поведению на сайте, покупкам, прочим событиям и метрикам.

## Объединение профилей

Клиент может посещать сайт с разных устройств или из разных браузеров. Для примера рассмотрим следующую цепочку действий: клиент впервые зашёл на сайт с телефона, затем зашёл с компьютера и зарегистрировался на сайте (получил доступ в личный кабинет), затем вошёл в личный кабинет с телефона. Теперь посмотрим какие данные будут в профиле клиента в UX Rocket:

- 1) при входе с телефона система создаёт новый профиль с куки (далее профиль\_1 с идентификатором куки\_1).
- 2) при входе с компьютера система создаёт новый профиль с куки (далее профиль\_2 с идентификатором куки\_2).
- 3) при регистрации на сайте система добавляет данные о телефоне в профиль клиента (в профиль\_2 добавлен идентификатор телефон\_2).
- 4) при входе в личный кабинет с телефона система пробует добавить телефон\_2 в профиль\_1 и понимает, что это невозможно, так как телефон\_2 уже закреплён за профиль\_2.

Как мы видим на шаге 4 возникла ситуация, когда система не может сохранить идентификатор в профиль из-за наличия для одного клиента двух профилей. Решить проблему можно путём объединения профиль\_1 и профиль\_2.

В объединении обычно участвуют два профиля, но могут участвовать и более двух профилей. Например, при регистрации покупке на сайте в таблицу сопоставления идентификаторов добавлена тройка «visitor – email – телефон». Если ранее в системе было три отдельных профиля: с visitor, с email и с телефон, то возникает необходимость объединить одновременно три профиля.

При объединении профилей система а) объединяет карточки профилей, б) переносит исторические данные в самый ранний профиль. Алгоритм объединения профилей состоит из следующих шагов:

- Идентификаторы из исходного профиля переносим в целевой профиль.
  - ✓ Если при объединении в целевом профиле возникнет два значения для одного жёсткого идентификатора, то данный исходный профиль не объединяем (но можем продолжить объединение с последующими профилями, если необходимо).
  - ✓ Если после переноса мягких идентификаторов по любому из типов мягких идентификаторов в целевом профиле будет более 50 записей, то удаляем идентификаторы, которые не использовались дольше всего.
- Непустые свойства от исходного профиля заполняем в целевом профиле.
- Перемещаем события из исходного профиля в целевой профиль.
- Перемещаем покупки из исходного профиля в целевой профиль.
- Перемещаем сегменты из исходного профиля в целевой профиль.
- Не удаляем метрики у исходного профиля (иначе поедут отчёты за прошлые периоды).

**Операция объединения клиентов необратима, поэтому на этапе настройки системы важно правильно настроить типы и приоритеты идентификаторов, а также перечень и параметры событий идентификации клиентов на сайте и в МП.**

## Перенос идентификаторов между профилями

В разделе «Типы идентификаторов клиента» указано, что мягкий идентификатор может переходить от одного клиента к другому. Ниже будут рассмотрены примеры, когда возникает необходимость переноса идентификатора из одного профиля в другой, и упрощённый алгоритм такого переноса.

Допустим, что на одном физическом устройстве (компьютере) в разные моменты времени происходит идентификация разных посетителей сайта. Например, сначала жена зашла в свой личный кабинет, а потом муж. В этом случае есть два разных профиля (с разными жесткими идентификаторами) и одна кука, так как и муж и жена заходят на сайт из одного браузера.

Достоверно определить кто конкретно зашёл на сайт – муж или жена – можно только после логина в личный кабинет (подтверждение пользователя через ввод пароля, авторизацию через ВКонтакте, авторизацию по токену сайта государственных услуг и другими схожими способами). Но большинство сессий на сайте – это анонимные сессии, где посетитель не проходит авторизацию. К какому из профилей мы хотели бы прикрепить события из анонимных сессий? В принципе, можно было бы не привязывать ни в один из профилей, но тогда поведенческие данные станут недоступными.

**Лучшая практика состоит в том, чтобы переносить мягкий идентификатор (в данном примере кука) в тот профиль клиента, который последний раз заходил в личный кабинет.** Например, кука была в профиле мужа, а жена зашла в личный кабинет. Тогда исторические события в профиле мужа должны остаться у мужа, а куку надо перенести в профиль жены, чтобы новые события собирать в профиль жены. Данный подход реализован в системе UX Rocket.

Из приведённого выше следует, что таблицу сырых данных нельзя связать с профилем клиента ни напрямую, ни через какую-либо промежуточную таблицу, так как перенос куки в другой профиль приведёт к перемещению истории. Следовательно, логика хранения в профиле действий на сайте будет следующей: фоновый процесс по определённым правилам создаёт запись в таблице событий клиента на основании записи о действии на сайте или в МП. При перемещении куки в другой профиль за мужем

останутся его записи о поведении на сайте, но куки перейдут к жене, поэтому новые записи о действиях на сайте будут добавляться в profile\_event для профиля жены.

Перенос мягкого идентификатора из одного профиля в другой можно выполнять неограниченное число раз.

## Разрешение конфликтов идентификации

В некоторых случаях объединение существующих профилей по приведенным выше правилам невозможно, так как оба профиля имеют жесткие идентификаторы одного типа. Тогда если в записи об идентификации клиента указаны мягкие идентификаторы, то конфликт можно попробовать разрешить путём переноса мягкого идентификатора из одного профиля в другой. В примере выше перенос куки из профиля мужа в профиль жены разрешает конфликт между последней записью идентификации (куки-профиль\_жены) и идентификатором куки в таблице profile\_identity у мужа.

Важно понимать, что перенос идентификатора не устраняет все противоречия в данных, а согласует конкретную запись об идентификации с распределением мягких идентификаторов по профилям. Прошлые события, естественно, станут НЕ согласованы с итоговым распределением идентификаторов (в прошлом есть запись об идентификации мужа «куки-профиль\_мужа», которое после переноса куки в профиль жены станет не согласованным). **UX Rocket согласует событие идентификации в настоящем с распределением мягких идентификаторов по профилям.**

Иногда существует несколько способов разрешения одного и того же конфликта путём переноса разных типов мягких идентификаторов. При наличии нескольких способов разрешения конфликта надо переносить менее важные идентификаторы (минимальное значение приоритета).

Для иллюстрации рассмотрим пример с перемещением куки от одного профиля к другому приведён в таблице ниже:

Идентификаторы: vk (жесткий), куки (мягкий) Клиенты: c1 с {'vk': '1', 'куки': ['1', '3']} и c2 с {'vk': '2', 'куки': '2'} Запись об идентификации: {'vk': '2', 'куки': '1'} Ожидаемый результат: c1 с {'vk': '1', 'куки': '3'} и c2 с {'vk': '2', 'куки': ['2', '1']}
--

В некоторых случаях перемещение мягких идентификаторов не может разрешить конфликт. В этих случаях используется та же стратегия, что и при конфликтах, вызванных жесткими идентификаторами.

## Правила создания событий

События клиентского профиля – это то, что происходит между клиентом и бизнесом. Например, клиент посмотрел товар, добавил товар в корзину, получил email, перешёл по рекламе и так далее.

В UX Rocket можно создавать события двумя способами:

- загружать через API из сторонних систем (CRM, eCommerce и так далее);
- создавать в системе в режиме реального времени на основании онлайн и офлайн данных.

Для каждого события в системе надо указать перечень заполняемых атрибутов. Например, для посещения страницы сайта настроим сохранение атрибутов url и тип событие на странице, а для покупки товара настроим сохранение суммы покупки, количества товара, код товара и точки продажи.

На экране ниже показано правило автоматического создания событий с названием «Просмотр страницы» на основании данных о поведении пользователя на сайте. В данном примере в качестве

источника данных выбрана таблица с данными о поведении на сайте и в МП. Перечень полей, их значение и условия отбора записей можно описать на встроенном языке.

Обновить событие

Id	Системный код	Наименование	Активен
2	1	Просмотр страницы	<input checked="" type="checkbox"/>

Источник: События с сайта и МП

Перечень полей: name, url, event, event\_context

Значения полей: 'Просмотр страницы' || url, url, event, event\_context

Условия отбора записей в исходной таблице: event\_context != 'openpage'

Сохранить Отменить

Обратите внимание, что система может создавать несколько разных событий на основании одной и той же записи в первичной таблице. Например, открытие страницы на сайте с utm-меткой можно преобразовать в два события: показ рекламы в Яндекс (от него пришла utm-метка, но сам он не передаёт событие в UX Rocket) и открытие страницы сайта.

В текущей версии система считает все правила создания событий независимыми и может обрабатывать параллельно или последовательно (задать порядок выполнения правил нельзя). То есть нельзя создавать производные события на основании ранее созданных событий другого типа.

Создание событий происходит в пакетном задании, частота запуска задания в облаке – 1 раз в 5 минут, в Enterprise версии частоту запуска пакетного задания можно настроить.

## Загрузка офлайн данных

Описание API для загрузки данных доступно по ссылке <https://api.uxrocket.ru/swagger/index.html>.

Для получения доступа обратитесь в службу поддержки UX Rocket.

## Подключение SDK

Решение UX Rocket позволяет настраивать сбор данных о поведении пользователя в мобильном приложении (далее МП), проводить А/В-тестирование и персонализировать внешний вид страниц под целевые группы пользователей. Используется следующая схема:



При сборе данных МП передаёт в UX Rocket информацию об открытии страниц, нажатии кнопок, кликах в меню, переходах по ссылкам, покупках и других событиях, а UX Rocket сохраняет поток событий на сервере и предоставляет маркетологу инструменты для построения отчётов и исследования данных. При А/В-тестировании и персонализации мобильное приложение сообщает UX Rocket об открытии новых страниц, а UX Rocket возвращает данные о перечне визуальных элементов МП, которые надо изменить, и новых значениях свойств для этих элементов. Список тестируемых кампаний и результаты тестирования можно посмотреть в личном кабинете в режиме реального времени.

**Важно, что все настройки по А/В-тестированию, запуск и остановку тестов можно выполнить силами отдела маркетинга без привлечения ИТ-специалистов и без публикации мобильного приложения в магазинах.**

Для удобства работы с API UX Rocket разработан SDK, который решает следующие задачи:

- Автоматическое заполнение большинства параметров устройства;
- Кеширование параметров пользователя, чтобы не указывать их при каждом вызове;
- Кеширование обращений к API при отсутствии интернета и последующая передача данных в фоновом режиме;
- Кеширование параметров таргетинга и персонализации для лучшего клиентского опыта.

Обычно маркетологам нужна информация о поведении пользователей МП как в целом, так и в разрезе конкретных групп пользователей. Поэтому детальные данные о поведении пользователя в приложении и данные А/В-тестирования содержат следующие поля: идентификатор клиента, название и версия МП, параметры устройства (модель телефона, размер экрана, версия ОС, версия браузера и т.д.), геолокация (страна, город), параметры пользователя (перечень параметров настраивается в UX Rocket, например, уровень участника в программе лояльности, город проживания и т.п.) и некоторые другие поля.

## Основные функции SDK

- Сбор сведений о поведении пользователя в мобильном приложении: переходы между экранами и взаимодействие с элементами интерфейса;
- Персонализация отображаемого контента в разрезе целевых групп пользователей;
- Сбор сведений о взаимодействии с персонализированным контентом и связанных с этим целевых действиях (например, покупка в приложении).

## Поддерживаемые платформы

SDK поддерживает работу на платформах и поставляется в следующих форматах:

- Android – AAR
- iOS – CocoaPods

## Начало работы с SDK

1. Для подключения к API необходимо узнать URL (`service_url`) по которому оно расположено. Если UX Rocket развёрнут на локальных серверах вашей компании, то адрес для подключения нужно выяснить у системного администратора. Если используется облачная подписка UX Rocket, то адреса для подключения:
  - Продуктивная среда: <https://api.uxrocket.ru/mobile>
  - Тестовая среда: <https://apidev.uxrocket.ru/mobile>

Описание методов приведено в Swagger: `{service_url}/swagger`

2. Для получения списка всех доступных параметров пользователя, которые нужно передавать в API, вызовите сервис `{service_url}/GetParams`. Как правило набор этих параметров фиксированный и редко дополняется, поэтому нет необходимости вызывать метод из приложения. Достаточно получить список параметров и их коды при подключении SDK и отслеживать появление новых параметров при выпуске новых версий мобильного приложения.
3. В тех случаях, когда вы по объективным причинам не можете заполнить значение одного или нескольких параметров (например, до успешного логина нельзя определить уровень участника в программе лояльности) следует передавать пустые значения.

## Подключение библиотеки SDK

### Android

В `build.gradle` добавить

```
android {
    ...
    ...
    repositories {
        google()
        mavenCentral()
        maven {
            name = "LibraryPackages"
            url = uri("https://maven.pkg.github.com/excitekit/uxrocket-
sdk-android")
        }
    }
}
```



```

    }
}
dependencies {
    ...
    ...
    implementation 'com.uxrocket:android-sdk:{version}'
}

```

## iOS

### SPM (Swift Package Manager)

В Xcode:

1. File → Add Packages
2. Указать адрес репозитория <https://github.com/excitekit/uxrocket-sdk-ios>
3. Dependency Rule минимальная версия 1.0.0
4. Нажать Add Package

### CocoaPods

1. В папке проекта, в файле podfile указать путь к месту, где будет находиться SDK

```

target 'YourAppName' do
    pod 'UXRocketSDK', :path => 'Path_to_sdk'
end

```

2. В терминале зайти в папку проекта и выполнить команду `pod install`

## Инициализация

Для инициализации SDK необходимо вызвать метод SDK `configure`, передав в него параметры авторизации:

- `authKey` – код Клиента. В примере `2JIJ67L7CS`
- `appRocketId` – идентификатор вашего приложения, в примере `test_ios_sdk_uxrocket` и `test_android_sdk_uxrocket`
- `serverEnvironment` – строка подключения к API UX Rocket

### Android

```

override fun onCreate() {
    super.onCreate()
    instance = this
    UXRocket.configure(
        authKey = "2JIJ67L7CS",
        appRocketId = "test_android_sdk_uxrocket",
        serverEnvironment = UXRocketServer.develop(),
        applicationContext = applicationContext
    )
}

```

## iOS

```
func application(application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    UXRocket.configure(
        withAuthKey: "2JIJ67L7CS",
        rocketAppID: "test_ios_sdk_uxrocket",
        serverEnvironment: .develop)

    return true
}
```

## Установка значений по умолчанию

SDK автоматически собирает такие параметры, как разрешение экрана, версия ОС, модель телефона и т.д. Для сбора некоторых параметров нет необходимости запрашивать их у устройства каждый раз, достаточно установить их при инициализации SDK. Для установки значений по умолчанию используется метод `setDefaultParameters`. Параметры, установленные этим методом, будут использоваться в дальнейшем другими методами при отправке данных в UX Rocket.

- Метод `setParams` принимает `id` и `value` для `params`, которые будут сохранены в SDK и устанавливаться по умолчанию в методах, если параметры с соответствующими `id` не заполнены.
- Данные хранятся в пределах сессии
- После завершения сессии сохраненные значения удаляются.
- Разработчик МП может сохранять при помощи метода любые значения и переменные
- Корректность значений методом не проверяется

Входные данные от МП:

Параметр	Описание	Тип данных
<code>id</code>	Идентификатор параметра	<code>int</code>
<code>value</code>	Значение параметра	<code>varchar</code>

При корректном сохранении в SDK значений по умолчанию возвращается код «OK», в противном случае «Error».

## Android

```
UXRocket.setDefaultParams(
    listOf(
        AttributeParameter("2", "Красный"),
        AttributeParameter("3", "Черный")
    )
)
```

## iOS

```
UXRocket.setDefaultParameters(
[
    .init(id: 1, value: "Желтый"),
    .init(id: 2, value: "true"),
    .init(id: 3, value: "false")
])
```

## Геолокация

Также происходит сбор сведений о местоположении пользователя. Ваше приложение может не иметь доступа к геолокации устройства или запрашивать её лишь в определённом случае. Для того чтобы SDK не запрашивало эти сведения каждый раз, вы можете передавать название города и страны в SDK в удобный момент, например при старте приложения, после инициализации или когда определения местоположения требуется по бизнес-процессу.

- Метод принимает значения `city` и `country`, когда их надо сменить при смене геолокации.
- Данные хранятся в пределах сессии
- После завершения сессии сохранённые значения удаляются.
- Корректность значений методом не проверяется

### Android

```
UXRocket.setCountryAndCity("Russia", "Moscow")
```

### iOS

```
UXRocket.setCountryAndCity("Russia", "Moscow")
```

## Сбор сведений о поведении пользователя

SDK может регистрировать различные типы событий. Для регистрации события следует вызвать метод отправки события, в который передать характеристики наступившего события.

### Android

Регистрация события:

```
UXRocket.logEvent(  
    itemIdentifier = "идентификатор или имя элемента",  
    itemName = "надпись на элементе",  
    event = ContextEvent.Type  
)
```

Регистрация события:

### iOS

```
UXRocket.logEvent(  
    event: EventContext.Type,  
    itemName: "надпись на элементе",  
    itemIdentifier: "идентификатор или имя элемента",  
    parameters: [AttributeParameter] = []  
)
```

## Описание параметров

Значение `ContextEvent.Type`

Значение	Описание
<code>install</code>	Установка приложения.
<code>openpage</code>	Открытие очередного экрана приложения

mainmenu	Переход в главное меню
buttons	Нажатие на кнопку
popup	Отображение pop-up окна в приложении
links	Переход по внешней ссылке
input	Ввод данных в редактируемое поле

### Набор атрибутов события

Большинство параметров собирает сам SDK, они описаны в [таблице](#) ниже. Ряд событий, связанных с таргетингом нужно передавать явно при вызове метода logEvent.

Значение AttributeParameter является массивом пар ключ (int) – значение (string). Массив может принимать до 30 пар ключ-значение. Список параметров, которые можно передавать, определяется списком пользовательских полей, полученных при инициализации из метода GetAppParams.

Метод следует вызывать при взаимодействии пользователя с элементом интерфейса. Перечень отслеживаемых элементов согласовывается с маркетологом на этапе внедрения SDK.

Примеры вызова:

#### iOS

```
UXRocket.logEvent (
    event: .button,
    itemName: sender.currentTitle,
    itemIdentificator: sender.accessibilityIdentifier,
    parameters: [.init(id: 2, value: "234")]
)
```

#### Android

```
UXRocket.logEvent (
    itemIdentificator = "nav_search_button",
    itemName = "Nav search button pressed",
    event = ContextEvent.MAIN_MENU
)
```

### Список параметров, которые надо передавать явно

Значение	Описание
event	Тип действия. Допускаются любые значения. Если в UX Rocket не будет найдено подходящего события, то запись не сохраняется. Базовый перечень значений: install, openpage, mainmenu, buttons, popup, links, input. На проекте могут быть настроены пользовательские значения событий.
item_name	Надпись на элементе МП, если применимо (например, надпись на кнопке)
item	Идентификатор элемента в интерфейсе МП
parameters	Необязательное поле. Заполняется массивом параметров, если необходимо их сохранение для дальнейшего анализа

### Список параметров, собираемых SDK автоматически

Значение	Описание
----------	----------

app_rocket_id	Идентификатор приложения для UX Rocket
captured_dt	Дата-время наступления события
device_id	Для IOS – IFV. Для Android – google_aid
os_name	Имя операционной системы на устройстве пользователя. Возможные значения: ios   android
os_version	Версия операционной системы на устройстве пользователя.
device_model	Модель устройства (например, Galaxy S6)
device_type	Тип устройства. Возможные значения: phone   tablet   unknown.
device_locale	Язык интерфейса устройства. Например, en us
app_version_name	Версия приложения в виде, как указана разработчиком.
app_package_name	Имя пакета для Android или Bundle ID для Ios
connection_type	Тип подключения устройства. Возможные значения: wifi   cell   unknown
operator_name	Имя оператора сотовой связи
resolution	Разрешение экрана пользователя

## Применение вариантов таргетинга

При отрисовке страницы мобильного приложения необходимо получить варианты таргетинга. Для этого надо вызвать метод «getUIConfiguration» и проанализировать его ответ. В ответе содержится несколько массивов variantAttrs. Из этих массивов нужно взять данные для персонализации.

Для дальнейшего поиска элементов, изменение которых требуется отслеживать, разработчик должен назначить им персональные идентификаторы, например Id или Name. Перечень отслеживаемых элементов согласовывается с маркетологом на этапе внедрения SDK.

Для каждого элемента массива на странице ищется элемент интерфейса с идентификатором равным значению поля item. Если такой элемент найден, то его атрибутам следует присвоить значения, содержащиеся во вложенном массиве attributes.

Пример:

```
[
  {
    "id": 7,
    "variants": [
      {
        "id": 50,
        "element_id": 24,
        "variant_attrs": [
          {
            "item": "FourthVC_LeftButton",
            "attributes": [
              {
                "attribute": "text",
                "value": [
                  {
                    "value": "Много текста, держи еще
два, а потом три, пятый за деревом",
                    "state": ["highlighted"]
                  },
                  {
                    "value": "Тут будет не так много
текста"
```

```

        ]
    }
}
],
{
    "id": 52,
    "element_id": 25,
    "variant_attrs": [
        {
            "item": "FourthVC_RightButton",
            "attributes": [
                {
                    "attribute": "text",
                    "value": "Push me, please!"
                },
                {
                    "attribute": "text-color",
                    "value": "#000000"
                }
            ]
        }
    ]
}
],
"actions": [
    {
        "id": 23,
        "name": "ClickButton",
        "item": "ViewController_FirstButton",
        "action_type": 1,
        "counting_type": 1
    },
    {
        "id": 24,
        "name": "SecondSumLabel",
        "item": "FourthVC_SecondSumLabel",
        "action_type": 0,
        "counting_type": 2
    },
    {
        "id": 25,
        "name": "ClickImage",
        "item": "FourthVC_ImageView",
        "action_type": 1,
        "counting_type": 1
    }
]
}
]
]

```

В указанном примере, помимо прочего, для элемента интерфейса с идентификатором «FourthVC\_RightButton» необходимо:

- атрибуту "text" присвоить значение "Push me, please!"
- атрибуту "text-color" присвоить значение "#000000"

Аналогичным образом необходимо обработать пришедшие в ответе метода массивы «actions». Каждый элемент массива указывает на целевое действие (например, тап по кнопке), настроенное маркетологом для сохранения.

Перечень отслеживаемых действий согласовывается с маркетологом на этапе внедрения SDK.

Примеры:

### *Android*

```
val params = listOf(AttributeParameter(id = "1", value = "190"))
val customizingItems = listOf(binding.homeBannersButton,
binding.addCustomBannerButton)
    val fragmentName = "HomePage"
UXRocket.getUIConfiguration(
    activityOrFragmentName = fragmentName,
    parameters = params,
    callback = {
        UXRocket.customizeItems(
            items = customizingItems,
            it,
            activityOrFragmentName = fragmentName
        )
        UXRocket.processActions(
            items = customizingItems,
            campaigns = it,
            activityOrFragmentName = fragmentName,
            parameters = params
        )
    })
```

### *iOS*

```
UXRocket.getUIConfiguration(
    forItem: vcIdentifier,
    parameters: [.init(id: 2, value: "2")]
) { response in
    self.alert.dismiss(animated: true)

    guard let response = response else { return }

    UXRocket.customizeItems(elementsToCustomize, with:
response)
    UXRocket.logCampaignEvent(viewControllerID: vcIdentifier,
totalValue: Int(self.secondSumLabel.text ?? "150") ?? 150)
}
```

## Сбор сведений о взаимодействии с персонализированным контентом

Для проведения анализа эффективности маркетинговой кампании требуется собрать сведения о взаимодействии пользователя с элементами интерфейса, изменёнными в результате применения таргетинга.

Для сбора сведений разработчик должен реализовать отслеживание взаимодействие с элементами, список которых был согласован с маркетологом. Таргетинг не обязательно может изменять все

согласованные элементы, этот список может меняться динамически и разработчику не требуется выбирать только те элементы, которые подверглись таргетингу в данный момент, событие отслеживания должно выполняться при взаимодействии с каждым элементом согласованного списка, независимо от применения к нему таргетинга.

При взаимодействии с каждым таким элементом необходимо вызвать функцию `logCampaignEvent` передав в неё набор атрибутов.

Вызов функции отслеживания таргетинга во многом аналогичен отслеживанию поведению пользователя в приложении.

## *Android*

Вызывается в Callback-е метода `getUIConfiguration`

```
UXRocket.getUIConfiguration(  
    activityOrFragmentName = "HomePage",  
    parameters = params,  
    callback = {  
...  
UXRocket.logCampaignEvent("HomePage", "home_banners_button", totalValue =  
1)  
...  
}
```

## *iOS*

```
UXRocket.getUIConfiguration(  
    forItem: vcIdentifier,  
    parameters: [.init(id: 2, value: "2")]  
    ) { response in  
    self.alert.dismiss(animated: true)  
  
    guard let response = response else { return }  
  
    UXRocket.customizeItems(elementsToCustomize, with:  
response)  
    UXRocket.logCampaignEvent(viewControllerID: vcIdentifier,  
totalValue: Int(self.secondSumLabel.text ?? "150") ?? 150)  
}
```