A/B-тестирование рекомендательной системы

Описание и цели проекта: Данный проект посвящен анализу соответствия эффекта от внедрения улучшенной рекомендательной системы ожидаемому (прирост конверсии на каждом из этапов воронки продаж на 5 п.п.) на основании результатов проведенного A/B-тестирования.

Задачи:

1. Провести обзор и предварительную обработку данных (проверить типы данных, пропуски, наличие дубликатов)
2. Проверить данные на соответствие техническому заданию:
    - период набора пользователей в тест и его соответствие требованиям,
    - регион регистрации пользователей (все ли попавшие в тест пользователи представляют целевой регион и составляет ли общее количество пользователей из целевого региона 15% от общего числа пользователей из целевого региона, зарегистрированных в период набора пользователей в тест),
    - динамику набора пользователей в группы теста и равномерность распределения пользователей по группам теста,
    - отсутствие пересечений с другими тестами и отсутствие пользователей, попавших в обе группы теста,
    - недельную цикличность набора пользователей в группы
    - период совершения событий участниками теста на соответствие требованиям;
    - наличие событий для каждого присутствующего в тесте пользователя,
    - горизонт анализа: рассчитать лайфтайм совершенных событий и проверить его на соответствие требованиям
3. Рассмотреть распределение количества событий на пользователя в разрезе групп теста, построить гистограмму распределения этой величины в разрезе групп и сравнить её средние значения между собой у групп теста;
4. Рассмотреть динамику количества событий в группах теста по дням, изучить распределение числа событий по дням и сравнить динамику групп теста между собой.
5. Проверить влияние маркетинговых событий на результаты тестирования
6. Провести анализ воронки продаж, рассчитать конверсию к первому шагу для каждого этапа в разрезе по группам, построить визуализацию
7. Проверить наличие статистической разницы между показателями конверсий на каждом этапе между группами
8. Подвести итоги A/B-тестирования, сделать общее заключение о корректности проведения теста и принять решение о целесообразности внедрения рекомендательной системы

Данные и Техническое Задание:

- Название теста: recommender_system_test;
- Группы: A (контрольная), B (новая платёжная воронка);
- Дата запуска теста: 2020-12-07;
- Дата остановки набора новых пользователей: 2020-12-21;
- Дата остановки теста: 2021-01-04;
- Ожидаемое количество участников теста: 15% новых пользователей из региона EU;

- Назначение теста: тестирование изменений, связанных с внедрением улучшенной рекомендательной системы;
- Ожидаемый эффект: за 14 дней с момента регистрации в системе пользователи покажут улучшение конверсии не менее, чем на 5 процентных пунктов на следующих этапах воронки:
  - из авторизации на сайте в просмотр карточек товаров
  - из авторизации на сайте в просмотры корзины
  - из авторизации на сайте в покупки

- Для анализа предоставлены следующие данные:
  - данные по новым зарегистрированным пользователям в период с 7 по 20 декабря 2020 года
  - данные по событиям новых пользователей в период с 7 декабря по 4 января 2021 года
  - таблица с участниками теста по группам
  - календарь маркетинговых событий на 2020 год

Содержание:

Обзор данных

In [1]:
```
# Библиотеки
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as st
import numpy as np
from statsmodels.stats.proportion import proportions_ztest
from plotly import graph_objects as go
```

In [2]:
```
# Зададим цвета для вывода результатов теста
class color:
    GREEN = '\033[92m'
    RED = '\033[91m'
    END = '\033[0m'
```

In [3]:
```
# Сформируем переменные с данными
marketing_events, events, new_users, participants  = (
        pd.read_csv('https://code.s3.yandex.net/datasets/ab_project_marketing_events.csv
        pd.read_csv('https://code.s3.yandex.net/datasets/final_ab_events.csv', parse_dat
        pd.read_csv('https://code.s3.yandex.net/datasets/final_ab_new_users.csv', parse_
        pd.read_csv('https://code.s3.yandex.net/datasets/final_ab_participants.csv') # y
)
```

Рассмотрим отдельно каждую из переменных.

Календарь маркетинговых событий

In [4]:
```
# Рассмотрим маркетинговые события за 2020 год
marketing_events.sort_values(by='start_dt')
```

Out[4]:

	name	regions	start_dt	finish_dt
6	Chinese New Year Promo	APAC	2020-01-25	2020-02-07
1	St. Valentine's Day Giveaway	EU, CIS, APAC, N.America	2020-02-14	2020-02-16
8	International Women's Day Promo	EU, CIS, APAC	2020-03-08	2020-03-10
2	St. Patric's Day Promo	EU, N.America	2020-03-17	2020-03-19
3	Easter Promo	EU, CIS, APAC, N.America	2020-04-12	2020-04-19
7	Labor day (May 1st) Ads Campaign	EU, CIS, APAC	2020-05-01	2020-05-03
9	Victory Day CIS (May 9th) Event	CIS	2020-05-09	2020-05-11
11	Dragon Boat Festival Giveaway	APAC	2020-06-25	2020-07-01
4	4th of July Promo	N.America	2020-07-04	2020-07-11
13	Chinese Moon Festival	APAC	2020-10-01	2020-10-07
12	Single's Day Gift Promo	APAC	2020-11-11	2020-11-12
5	Black Friday Ads Campaign	EU, CIS, APAC, N.America	2020-11-26	2020-12-01
0	Christmas&New Year Promo	EU, N.America	2020-12-25	2021-01-03
10	CIS New Year Gift Lottery	CIS	2020-12-30	2021-01-07

In [5]:
```
# Проверим типы данных
marketing_events.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype         
---  ------     --------------  -----         
 0   name       14 non-null     object        
 1   regions    14 non-null     object        
 2   start_dt   14 non-null     datetime64[ns]
 3   finish_dt  14 non-null     datetime64[ns]
dtypes: datetime64[ns](2), object(2)
memory usage: 576.0+ bytes
```

Из календаря событий мы видим, что в декабре 2020 года маркетинговые события были. При этом в момент набора пользователей в тест событий не было, но были события в конце декабря, которые могли оказать влияние на поведение этих пользователей.

Действия новых пользователей

In [6]:
```
# Рассмотрим первые строки датафрейма
events.head()
```

Out[6]:

	user_id	event_dt	event_name	details
0	E1BDDCE0DAFA2679	2020-12-07 20:22:03	purchase	99.99
1	7B6452F081F49504	2020-12-07 09:22:53	purchase	9.99
2	9CD9F34546DF254C	2020-12-07 12:59:29	purchase	4.99
3	96F27A054B191457	2020-12-07 04:02:40	purchase	4.99
4	1FD7660FDF94CA1F	2020-12-07 10:15:09	purchase	4.99

In [7]:
```
# Проверим типы данных
events.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440317 entries, 0 to 440316
Data columns (total 4 columns):
 #   Column      Non-Null Count   Dtype         
---  ------      --------------   -----         
 0   user_id     440317 non-null  object        
 1   event_dt    440317 non-null  datetime64[ns]
 2   event_name  440317 non-null  object        
 3   details     62740 non-null   float64       
dtypes: datetime64[ns](1), float64(1), object(2)
memory usage: 13.4+ MB
```

Все типы данных указаны корректно. Но в поле details присутствуют пропуски, поскольку стоимость указывается только для события покупки (purchase).

In [8]:
```
# Рассмотрим, какие события содержатся в нашем датафрейма
events['event_name'].value_counts()
```

Out[8]:
```
login           189552
product_page    125563
purchase         62740
product_cart     62462
Name: event_name, dtype: int64
```

В качестве событий пользователя мы имеем: вход в аккант, просмотр страницы продукта, добавление в корзину и покупка.

In [9]:
```
# Проверим, что данные в поле details заполнены только для purchase
events.groupby('event_name')['details'].agg('count')
```

Out[9]:
```
event_name
login               0
product_cart        0
product_page        0
purchase        62740
Name: details, dtype: int64
```

Действительно, мы видим, что для каждой покупки (purchase) поле со стоимостью (details) заполнено. В то время, как для других событий в этом поле пропуски, что является логичным и допустимым для нас в этом проекте, поэтому эти пропуски мы заполнять не будем.

In [10]:
```
# Проверим наличие дубликатов в данных
events.duplicated().sum()
```

Out[10]:
```
0
```

Полных дубликатов в данных нет. При этом в остальных полях дубликаты возможны, поскольку за период пользователь мог совершить несколько разных покупок.

Новые пользователи

In [11]:
```
# Рассмотрим первые строки датафрейма
new_users.head()
```

Out[11]:

	user_id	first_date	region	device
0	D72A72121175D8BE	2020-12-07	EU	PC

1	F1C668619DFE6E65	2020-12-07	N.America	Android
2	2E1BF1D4C37EA01F	2020-12-07	EU	PC
3	50734A22C0C63768	2020-12-07	EU	iPhone
4	E1BDDCE0DAFA2679	2020-12-07	N.America	iPhone

In [12]:
```
# Проверим типы данных
new_users.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 61733 entries, 0 to 61732
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype         
---  ------      --------------  -----         
 0   user_id     61733 non-null  object        
 1   first_date  61733 non-null  datetime64[ns]
 2   region      61733 non-null  object        
 3   device      61733 non-null  object        
dtypes: datetime64[ns](1), object(3)
memory usage: 1.9+ MB
```

In [13]:
```
# Проверим данные на полные дубликаты
new_users.duplicated().sum()
```

Out[13]:
```
0
```

Типы данных в датафрейме указаны корректно, пропусков и дубликатов нет.

Участники теста

In [14]:
```
# Изучим первые строки датафрейма
participants.head()
```

Out[14]:

	user_id	group	ab_test
0	D1ABA3E2887B6A73	A	recommender_system_test
1	A7A3664BD6242119	A	recommender_system_test
2	DABC14FDDFADD29E	A	recommender_system_test
3	04988C5DF189632E	A	recommender_system_test
4	482F14783456D21B	B	recommender_system_test

In [15]:
```
# Проверим типы данных
participants.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18268 entries, 0 to 18267
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   user_id  18268 non-null  object
 1   group    18268 non-null  object
 2   ab_test  18268 non-null  object
dtypes: object(3)
memory usage: 428.3+ KB
```

In [16]:
```
# Рассмотрим информация о каких тестах содержится в датафрейма
participants['ab_test'].value_counts()
```

Out[16]:
```
interface_eu_test          11567
recommender_system_test     6701
Name: ab_test, dtype: int64
```

Все типы данных корректны, пропусков нет. В датафрейме содержатся данные об участниках сразу 2х тестов: interface_eu_test, recommender_system_test, это необходимо учесть в нашем будущем анализе и убрать учатников другого теста из данных.

In [17]:
```
# Проверим наличие полных дубликатов
participants.duplicated().sum()
```

Out[17]:
```
0
```

In [18]:
```
# Проверим наличие дубликатов в поле с user_id
participants[participants.duplicated(subset='user_id', keep=False)].sort_values(by='user
```

Out[18]:

	user_id	group	ab_test
17892	001064FEAAB631A1	B	interface_eu_test
235	001064FEAAB631A1	B	recommender_system_test
16961	00341D8401F0F665	A	interface_eu_test
2137	00341D8401F0F665	A	recommender_system_test
8143	003B6786B4FF5B03	A	interface_eu_test
...	...	...	...
5213	FFC53FD45DDA5EE8	B	recommender_system_test
5667	FFED90241D04503F	B	recommender_system_test
14158	FFED90241D04503F	B	interface_eu_test
3448	FFF28D02B1EACBE1	B	recommender_system_test
7238	FFF28D02B1EACBE1	A	interface_eu_test

3204 rows × 3 columns

In [19]:
```
# Проверим есть ли дубликаты пользователей только среди recommender_system_test
recom_dup = participants.query('ab_test == "recommender_system_test"')
recom_dup[recom_dup.duplicated(subset='user_id', keep=False)]
```

Out[19]:

	user_id	group	ab_test

В данных нет полных дубликатов, но есть пользователи попавшие в оба теста. Мы рассмотрим и удалим их в следующем разделе. При этом среди пользователей теста recommender_system_test дубликатов нет, а значит нет и пользователей, попавших, как в группу A, так и B.

Вывод: В данном разделе мы получили общее представление о данных, проверили типы данных и наличие пропусков и дубликатов в них. В наших данных нет полных дубликатов, но есть пользователи, которые попали сразу в несколько тестов. В следующем разделе мы рассмотрим все датафреймы более подробно и проверим их на соответствие техническому заданию.

Оценка корректности проведения теста

Согласно нашему техническому заданию в нашем тесте, посвященном тестированию изменений, связанных с внедрением улучшенной рекомендательной системы, должно было участвовать **15% от всех новых пользователей из Европы, зарегистрированных в период с 7 по 20 декабря (2 недели). Полная дата остановки теста - 4 января 2021 года.**

В данном разделе мы проверим все полученные данные на соответствие ТЗ и в случае выявления нарушений проведем их обработку.

Набор новых пользователей в тест

In [20]:
```
# Проверим период набора пользователей в тест и его соответствие ТЗ
print('В датафрейме присутствуют новые пользователи, зарегистрированные с {} по {}'.format
```

```
В датафрейме присутствуют новые пользователи, зарегистрированные с 2020-12-07 по 2020-12-2
3
```

Согласно нашему Техническому заданию в A/B-тесте должны участвовать только новые пользователи, зарегистрированные с 7 по 20 декабря. Удалим пользователей, зарегистрированных позже 20 декабря из датафрейма.

In [21]:
```
# Отфильтруем пользователей, зарегистрированных после 20 декабря
new_users = new_users.loc[new_users['first_date'] <= '2020-12-20']
```

In [22]:
```
# Рассмотрим регион новых пользователей
region = new_users.groupby('region')[['user_id']].agg('count').reset_index()
region['%'] = (region['user_id'] / region['user_id'].sum() * 100).round(2)
region
```

Out[22]:

	region	user_id	%
0	APAC	2551	5.08
1	CIS	2530	5.04
2	EU	37690	75.11
3	N.America	7409	14.76

75% новых пользователей из Европы, еще 25% приходится на жителей других регионов. Поскольку в данном исследовании мы будем анализировать только результаты теста, направленного на изменения в регионе EU, то данные пользователей остальных регионов нам не нужны.

In [23]:
```
# Отфильтруем новых пользователей из других регионов
new_users = new_users.loc[new_users['region'] == 'EU']
```

In [24]:
```
# Проверим оставшиеся даты регистрации новых пользователей
print('В датафрейме присутствуют новые пользователи, зарегистрированные с {} по {}'.format
```

```
В датафрейме присутствуют новые пользователи, зарегистрированные с 2020-12-07 по 2020-12-2
0
```

Обработка данных в датафрейме с новыми пользователями завершена, мы оставили в нем только данные по пользователям из Европы, зарегистрированных в период с 7 по 20 декабря.

Важным требованием нашего тех. задания является то, что в тест должно было попасть 15% таких пользователей. Рассмотрим датафрейм с участниками A/B-теста и проверим так ли это.

Также ранее мы уже выявили, что в это же время проходило другое тестирование и 1602 пользователей попала в оба теста. Группа А - является контрольной, поэтому пользователи, которых в другом тесте состояли в группе А нам подходят, поскольку они тестировали старую версию продукта и на них не оказывалось влияние тестируемых изменений.\ Пользователей, состоявших в группе В второго теста брать в анализ рискованно, т.к. мы не знаем, какие изменения они тестировали и как это может сказаться на результатах нашего теста. Однако, в нашем тесте достаточно небольшое количество пользователей, поэтому мы рассмотрим их количество и распределение по группам более подробно. Если пользователи конкурирующего теста распределены равномерно по группам нашего теста, то получается и оказываемое ими влияние на результаты для каждой группы одинаково.

In [25]:
```
# Создадим переменную с участниками нужного нам теста и с пользователями, попавшими во 2
user_test = participants.query('ab_test == "recommender_system_test"')
b_users = participants.query('ab_test == "interface_eu_test" and group == "B"')['user_id
```

In [26]:
```
# Присоединим к таблице с участниками теста их характеристики и оставим участников, кото
user_test = user_test.merge(new_users, on='user_id', how='inner')
```

In [27]:
```
# Рассмотрим, как распеделились пользователи группы B конкурирующего теста в нашем тесте
b_users_check = (
                 user_test[user_test['user_id'].isin(b_users)]
                          .groupby('group')[['user_id']].agg('nunique')
                          .merge(user_test.groupby('group')[['user_id']].agg('nunique'),
)
b_users_check.columns = ['Участвуют в другом тесте','Участники нашего теста']
b_users_check['%'] = (b_users_check['Участвуют в другом тесте']/b_users_check['Участники
b_users_check
```

Out[27]:

group	Участвуют в другом тесте	Участники нашего теста	%
A	388	3236	11.99
B	311	2432	12.79

Таким образом, мы видим, что среди участников нашего теста 12-12,8% пользователей параллельно принимали участие в другом тесте и при этом состояли в группе В. Проведем статистический тест и рассмотрим статистическую значимость этого различия.

H0: Пользователи конкурирующего теста равномерно распределены по группам;\ **H1:** Пользователи конкурирующего теста неравномерно распределены по группам

In [28]:
```
# Проверим тест и проверим одинаковое ли влияние эти пользователи оказывают на наши груп
other_test = list(b_users_check['Участвуют в другом тесте'].values)
our_test = list(b_users_check['Участники нашего теста'].values)

alpha = 0.05 # статистическая значимость

pvalue = proportions_ztest(other_test, our_test, value = 0)[1]
print('p-value: {}'.format(pvalue))

if pvalue < alpha:
    print(color.RED + 'Отвергаем нулевую гипотезу: пользователи конкурирующего теста нер
else:
    print(color.GREEN +'Не получилось отвергнуть нулевую гипотезу: Пользователи конкурир
```

```
p-value: 0.3659858278795519
Не получилось отвергнуть нулевую гипотезу: Пользователи конкурирующего теста равномерно
 распределены по группам
```

Таким образом, **оказываемое влияние участников конкурирующего теста на группы текущего теста одинаково, а значит мы можем оставить таких пользователей в анализе.**

In [29]:
```
# Посчитаем количество участников теста и проверим его на соответствие ТЗ
print('В A/B-тесте приняло участие {} пользователей из Европы, зарегистрированных с {} п
 что составляет {}% от общего количества новых пользователей из этого региона.'
      .format(len(user_test), user_test['first_date'].min().date(),user_test['first_date
```

```
В A/B-тесте приняло участие 5668 пользователей из Европы, зарегистрированных с 2020-12-0
7 по 2020-12-20, что составляет 15% от общего количества новых пользователей из этого ре
гиона.
```

Таким образом, после нашей обработки данных, мы видим, что требование технического задания выполнено, в нем действительно приняло участие 15% новых пользователей Европы. Проведем статистический тест для дополнительной проверки.

Проведем тест, чтобы проверить действительно ли это статистически значимое различие:\ **H0:** для пользователя из региона EU вероятность попасть в тест составляет 15%;\ **H1:** для пользователя из региона EU вероятность попасть в тест отличается от 15%

In [30]:
```
# Проверим стат значимость вероятности пользователя попасть в тест

alpha = 0.05 # уровень значимости

shidaka_alpha = 1 - (1 - alpha)**(1/2) # поправка Шидака

# Z-тест
pvalue = proportions_ztest(user_test.shape[0], new_users.shape[0], value = 0.15)[1]

print('p-value: {}'.format(pvalue))

if pvalue < shidaka_alpha:
    print(color.RED +'Отвергаем нулевую гипотезу: для пользователя из региона EU вероятн
else:
    print(color.GREEN +'Не получилось отвергнуть нулевую гипотезу: для пользователя из р
```

```
p-value: 0.8344874409156163
Не получилось отвергнуть нулевую гипотезу: для пользователя из региона EU вероятность по
пасть в тест составляет 15%.
```

Таким образом, это **требование ТЗ выполнено.\ В тест попало 15% новых пользователей из Европы и это подтверждено статистическим тестом.**

In [31]:
```
# Рассмотрим распределение пользователей по группам
a_users = user_test.query('group == "A"').shape[0]
print('В группу А попало {}% пользователей'.format(round(a_users/user_test.shape[0]*100)
```

```
В группу А попало 57% пользователей
```

Распределение по группам отлично от 50/50. Проверим, насколько это отличие значимо.

Проведем тест, чтобы проверить действительно ли это статистически значимое различие:\ **H0:** вероятность попасть в группу А для пользователя составляет 50%;\ **H1:** вероятность попасть в группу А для пользователя отличается от 50%

In [32]:
```
# Проверим стат значимость вероятности пользователя попасть в тест

alpha = 0.05 # уровень значимости

shidaka_alpha = 1 - (1 - alpha)**(1/2) # поправка Шидака
```

```
# Z-тест
pvalue = proportions_ztest(a_users, user_test.shape[0], value = 0.5)[1]

print('p-value: {}'.format(pvalue))

if pvalue < shidaka_alpha:
    print(color.RED +'Отвергаем нулевую гипотезу: вероятность попасть в группу А отличае
else:
    print(color.GREEN +'Не получилось отвергнуть нулевую гипотезу: вероятность попасть в
```

```
p-value: 3.907518470766417e-27
Отвергаем нулевую гипотезу: вероятность попасть в группу А отличается от 50%.
```

Таким образом, распределение по группам не равномерно. Оценим размер наименьшей группы, поскольку это важнее распределения.

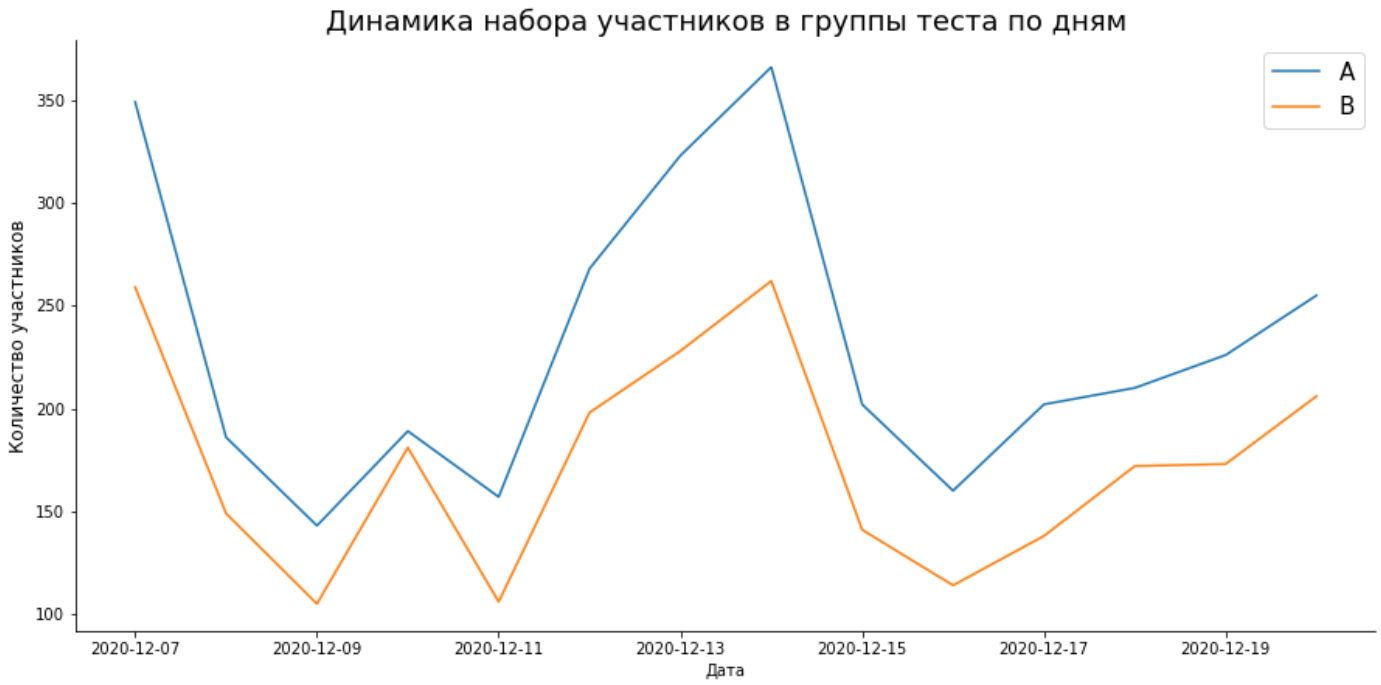
In [33]:
```
# Рассмотрим, сколько пользователей попало в группу B
user_test.query('group == "B"').shape[0]
```

Out[33]:
```
2432
```

По текущим данным базовая конверсия в переход на следующий этап воронки составляет 50%. Ожидаемый эффект от проводимых изменений составляет 5 процентных пунктов. Согласно калькулятору минимальный размер выборки в таком случае должен составлять 1567 человек. В нашей минимальной выборке 2432 человек, а значит этого достаточно.

Рассмотрим динамику набора пользователей в тест.

In [34]:
```
# Рассмотрим динамику набора пользователей в группы по дням
plt.figure(figsize=(15,7))
sns.lineplot(data = user_test.groupby(['first_date','group'])[['user_id']].agg('count').
             x='first_date', y='user_id',hue='group')
plt.title('Динамика набора участников в группы теста по дням', fontsize=18)
plt.xlabel('Дата')
plt.ylabel('Количество участников', fontsize=12)
plt.legend(fontsize=15)
sns.despine()
```



Мы видим, что набор пользователей в тест по дням проходил неравномерно, что связано с

динамикой посещаемости сайта (в выходные дни активность выше). При этом на протяжении практически всего периода (исключение 10 декабря) набор по группам проходил с одинаковой динамикой и в группу B попадало меньше участников.

Оценим также понедельную динамику.

In [35]:
```
# Добавим столбец с номером недели
user_test['first_week'] = (user_test['first_date'].dt.isocalendar().week).astype(str)

# Построим визуализацию
plt.figure(figsize=(15,7))
sns.lineplot(data = user_test.groupby(['first_week','group'])[['user_id']].agg('count').
             x='first_week', y='user_id',hue='group')
plt.title('Динамика набора участников в группы теста по неделям', fontsize=18)
plt.xlabel('Номер недели в году')
plt.ylabel('Количество участников', fontsize=12)
plt.legend(fontsize=15)
sns.despine()
```

Рассматривая понедельную динаку, мы видим, что набор пользователей в группу A проходил равномерно. А вот для группы B мы видим, некоторое снижение, на 2ой неделе в тест попадало меньше пользователей.

Активность пользователей

В данном разделе мы рассмотрим датафрейм с событиями пользователей. Согласно нашему ТЗ в тест должны попасть события новых пользователей до 4 января 2021, но совершенные в пределах 14 дней с момента регистрации пользователя.

In [36]:
```
# Проверим даты совершения событий пользователями
print('В датафрейме присутствуют события пользователей, совершенные с {} по {}'.format(e
```

```
В датафрейме присутствуют события пользователей, совершенные с 2020-12-07 по 2020-12-30
```

В нашем датафрейме представлены события с 7 по 30 декабря, хотя дата остановки теста - 4 января. Это значит, что не все пользователи прожили все необходимые 14 дней лайфтайма с момента регистрации.

In [37]:
```
# Оставим в датафрейме по событиям только нужных пользователей и добавим столбцы с их ха
events = events.merge(user_test, on='user_id', how='inner')

# Оценим результат
events.head()
```

Out[37]:

	user_id	event_dt	event_name	details	group	ab_test	first_date	region	device
0	831887FE7F2D6CBA	2020-12-07 06:50:29	purchase	4.99	A	recommender_system_test	2020-12-07	EU	Android
1	831887FE7F2D6CBA	2020-12-09 02:19:17	purchase	99.99	A	recommender_system_test	2020-12-07	EU	Android
2	831887FE7F2D6CBA	2020-12-07 06:50:30	product_cart	NaN	A	recommender_system_test	2020-12-07	EU	Android
3	831887FE7F2D6CBA	2020-12-08 10:52:27	product_cart	NaN	A	recommender_system_test	2020-12-07	EU	Android
4	831887FE7F2D6CBA	2020-12-09 02:19:17	product_cart	NaN	A	recommender_system_test	2020-12-07	EU	Android

In [38]:
```
# Рассмотрим, сколько пользователей теста совершали события
print('В датафрейме представлены данные по {} пользователям из {} зарегистрированных'
      .format(events['user_id'].nunique(), user_test.shape[0]))
```

```
В датафрейме представлены данные по 3000 пользователям из 5668 зарегистрированных
```

Таким образом из 5668 пользователей, прошедших регистрацию и попавших в тест, только 3000 прошли хотя бы первый этап воронки.\ Рассмотрим, как пользователи без каких либо событий распределены по группам теста.

In [39]:
```
# Рассмотрим, какое количество пользователей не совершали событий в разрезе по группам т
user_with_events = list(events['user_id'].unique())
user_without_events = user_test.query('user_id not in @user_with_events')
user_without_events.groupby('group')[['user_id']].agg('count')
```

Out[39]:

group	user_id
A	1030
B	1638

Мы видим, что пользователи группы B гораздо чаще пользователей группы A не проходили даже первый шаг нашей воронки. Это говорит о том, что разработанная новая рекомендательная система может влиять на переход к первому шагу воронки и работать хуже.

Поскольку в файле по событиям присутствует только чуть больше половины от всех пользователей, участвующих в тесте, создадим для каждого пользователя еще одно событие `registration` и добавим его к датафрейму.

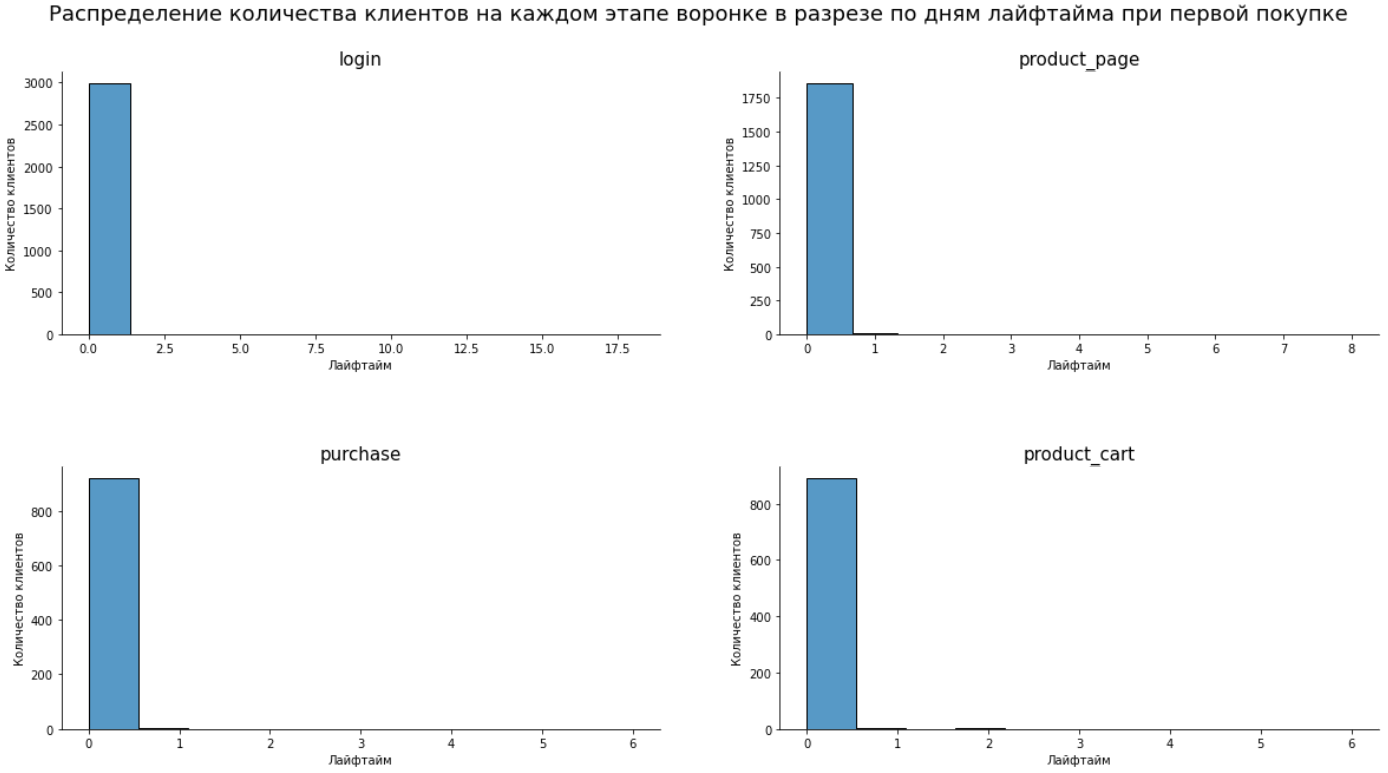
In [40]:
```
# Для каждого пользователя, участвовавшего в тесте добавим событие "registration"
user_test['event_name'] = 'registration'
```

```
# Добавим события с регистрацией пользователей в датафрейм с событиями
events = pd.concat([user_test, events], axis=0)
events.loc[events['event_dt'].isna(),'event_dt'] = events.loc[events['event_dt'].isna(),
```

In [41]:
```
# Рассчитаем день лайфтайма пользователя для каждого, совершенного им события
events['lifetime'] = (events['event_dt'] - events['first_date']).dt.days
```

In [42]:
```
# Для каждого пользователя отберем дату первого события и посмотрим, на какой день лайфт
first_event = events.groupby(['user_id','event_name','group'])[['lifetime']].agg('min').
```

In [43]:
```
# Построим визуализацию с распределением каждого события по дням

# Отберем признаки для диаграмм распределения
features = list(first_event['event_name'].unique())
features.remove('registration')


# Расположим все диаграммы в 2 столбца
num_cols = 2

# Рассчитаем нужное количество строк
if len(features)%num_cols == 0:
    num_rows = len(features)//num_cols
else:
    num_rows = (len(features)//num_cols) + 1

# Определим фигуру и оси
fig, ax = plt.subplots(figsize=(20,10),
                       nrows=num_rows,
                       ncols=num_cols)

# Добавим общее название
fig.suptitle('Распределение количества клиентов на каждом этапе воронке в разрезе по дня
             fontsize=18)

# Напишем цикл для построения диаграмм
for feat in features:
    row = features.index(feat)//num_cols
    col = features.index(feat)%num_cols
    table = first_event.query('event_name == @feat')
    sns.histplot(data=table['lifetime'], ax = ax[row][col])
    ax[row][col].set_title(feat, fontsize=15)
    ax[row][col].set_ylabel('Количество клиентов')
    ax[row][col].set_xlabel('Лайфтайм')
    sns.despine()


plt.subplots_adjust(top=0.9, wspace=0.2, hspace=0.5)
```



Мы видим, что большинство покупок пользователей приходится на первый же день после его регистрации. А значит, несмотря на то, что события в нашем датайфреме представлены только по 30 декабря, что не соответствует ТЗ, согласно диаграммам выше пользователь все равно должен был успеть пройти все этапы воронки.

Однако, мы также видим, что в данных присутствуют события с лайфтаймом более 14 дней, такие события мы в анализе учитывать не будем, посколько мы рассматриваем только эффект спустя 14 дней.

In [44]:
```
# Отфильтруем события, совершенные пользователем после 14 дня лайфтайма
events = events.query('lifetime < 14')
```

In [45]:
```
# Рассмотрим окончательный размер полученных выборо каждой группы
sample_size = (
                events.groupby('group')[['user_id']].agg('nunique')
                      .merge(events.query('event_name != "registration"').groupby('grou
                             right_index=True, left_index=True)
)

sample_size.columns = ['Общее кол-во пол-лей','Кол-во пол-лей, прошедших авторизацию']
sample_size
```

Out[45]:

group	Общее кол-во пол-лей	Кол-во пол-лей, прошедших авторизацию
A	3236	2206
B	2432	794

Вывод:\ В данном разделе мы проверили соответствие всех требований технического задания и можем сделать следующие выводы:

1. **Период набора новых пользователей в тест с 7 по 20 декабря 2020 года** : **Выполнено**. Изначально, в наших данных присутствовали зарегистрированные в период с 7 по 23 декабря, но

мы их отфильтровали под нужное требованиие.

1. **Дата остановки теста - 4 января 2021 года**: **Есть незначительные нарушения**. В данных присутствуют события пользователей только до 30 декабря включительно. Однако, в процессе анализа мы выявили, что большинство переходов по этапам воронки совершаются в первый же день после регистрации, а значит у всех пользователей была возможность пройти воронку полностью.

1. **Ожидаемое количество участников теста - 15% новых пользователей из региона EU** - **Выполнено**. В тест действительно попало 15% новых пользователей Европы, зарегистрированных в период с 7 по 20 октября и это подтверждено статистическим тестом. Среди выявленных нарушений стоит отметить присутствие в тесте пользователей из других регионов, а также попадание пользователей в контрольные группы B сразу нескольких тестов. Пользователи других регионов были удалены. Пользователей, участвующих в группе конкурирующего теста мы оставили, поскольку их количество равномерно распределено по нашим группам, а значит оказываемое ими влияние одинаково.

1. **Эффект от внедрения ожидается в течение 14 дней** - **Есть незначительные нарушения**. Поскольку события пользователей представлены только по 30 декабря, не для всех новых пользователей прошло необходимых 14 дней лайфтайма. Однако, в процессе анализа мы выявили, что большинство переходов по этапам воронки совершаются в первый же день после регистрации, а значит у всех пользователей была возможность пройти воронку полностью. События, совершенные пользователями, после 14 дня лайфтайма не будут учтены в анализе.

1. **Распределение пользователей по группам происходило неравномерно, но минимальный размер выбор достаточен для анализа**. Вероятность попасть в группу A для пользователя составляет 57%. В группу B попало 2121 человек. Согласно калькулятору минимальный размер выборки в таком случае должен составлять 1567 человек. А значит полученные выборки достаточны для анализа.

1. **Набор пользователей в тест происходил каждую неделю равномерно**. Динамика набора в тест по дням соответствует общей динамике регистраций на сайте. Распределение пользователей по группам практически на протяжении всего периода (исключение 10 декабря) также проходил с одинаковой динамикой, при этом большая часть пользователей ежедневно относилась к группе A.

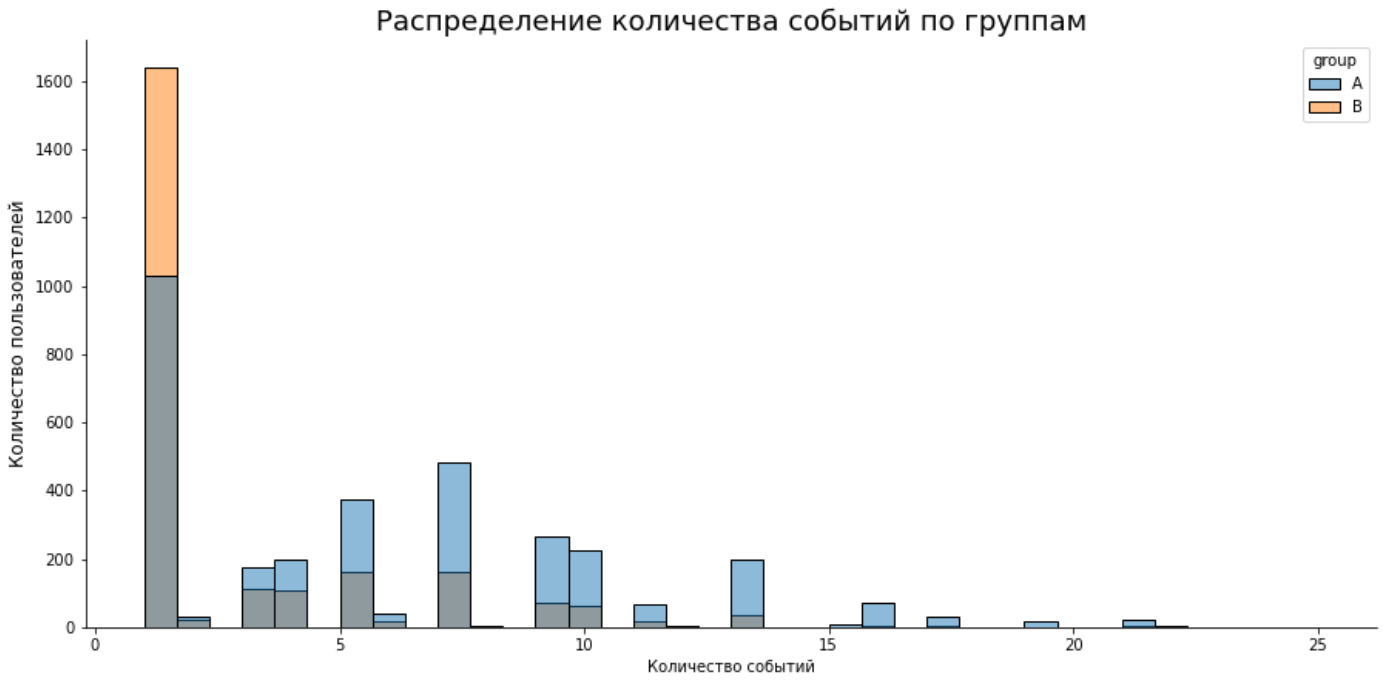
1. **Из 5668 пользователей, прошедших регистрацию и попавших в тест, только 3000 (53%) прошли хотя бы первый этап воронки.** При этом среди пользователей, не дошедших даже до первого этапа, преобладают пользователи из группы B. Мы добавили в датафрейм с событиями новое событие `registration` для каждого пользователя, участвующего в тесте.

Исследовательский анализ данных

Распределение количества событий по группам

In [46]:
```
# Рассмотрим количество событий, совершаемых пользователем каждой группы
plt.figure(figsize=(15,7))
sns.histplot(data = events.groupby(['user_id','group'])[['event_name']].agg('count').res
             x='event_name', hue='group')
```

```
plt.title('Распределение количества событий по группам', fontsize=18)
plt.xlabel('Количество событий')
plt.ylabel('Количество пользователей', fontsize=12)
sns.despine()
```



In [47]:
```
# Рассчитаем среднее значение количества событий для пользователя каждой группы с учетом
(
    events.groupby(['user_id','group'])[['event_name']].agg('count').reset_index()
          .groupby('group')[['event_name']].agg('mean').round(2)
)
```

Out[47]:

group	event_name
A	5.70
B	2.81

Несмотря на то, что количество пользователей в группе B меньше, чем в группе A, среди них наибольшее количество покупателей, которые совершили только 1 событие - регистрация. Среднее значение кол-ва событий, совершаемых пользователем этой группы, составляет 2,8.\ В то время, как **пользователи группы A взаимодействуют с сайтом гораздо активнее, среднее количество событий для этой группы - 5,7.**

In [48]:
```
# Рассчитаем также среднее количество событий без учета этапа регистрации
(
    events.query('event_name != "registration"').groupby(['user_id','group'])[['event_na
          .groupby('group')[['event_name']].agg('mean').round(2)
)
```

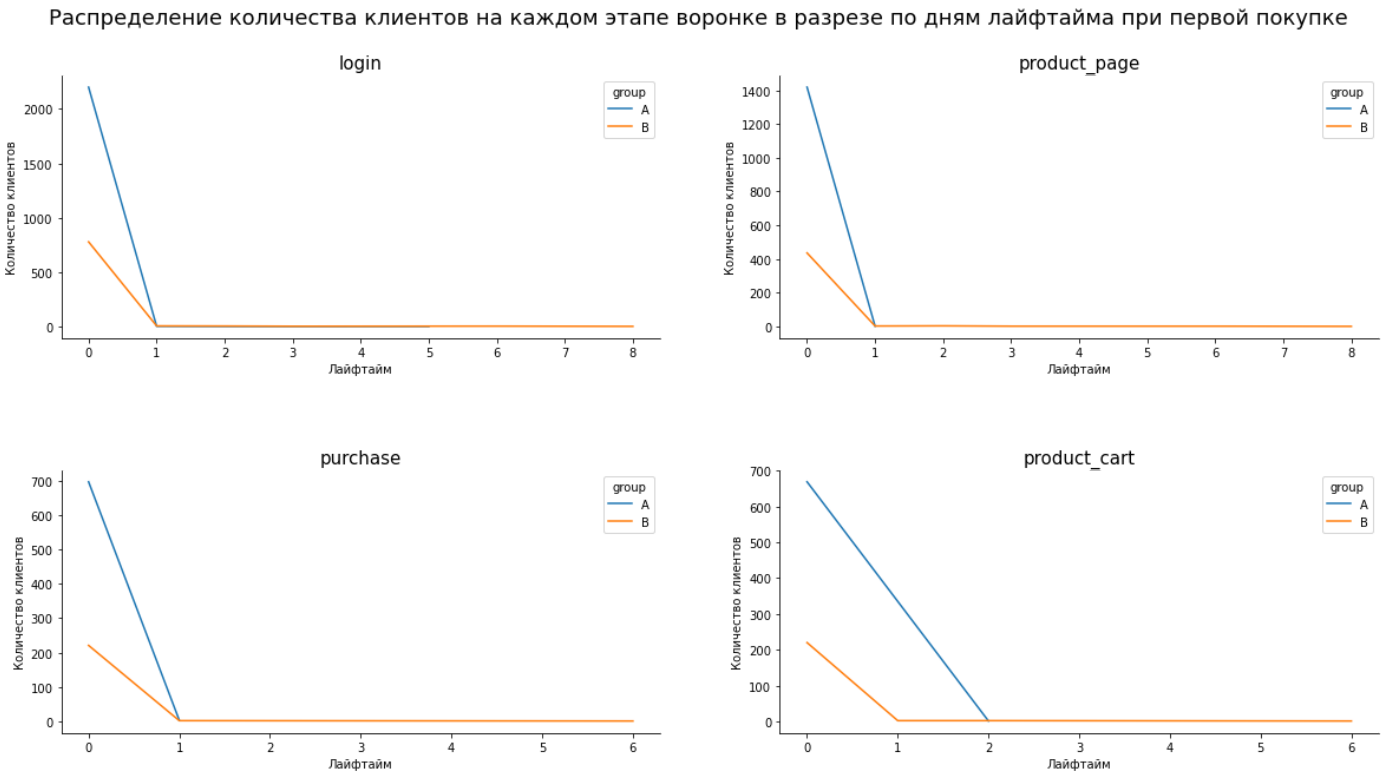
Out[48]:

group	event_name
A	6.89
B	5.54

Если не учитывать этап регистрации и рассматривать среднее количество событий среди

пользователей, которые как минимум прошли авторизацию, мы видим, что разница в количестве событий между группами становится небольшой, но при этом пользователи группы A (в среднем 6,9 событий) по-прежнему более активнее совершают действия на сайте, чем в группе B (5,5 событий).

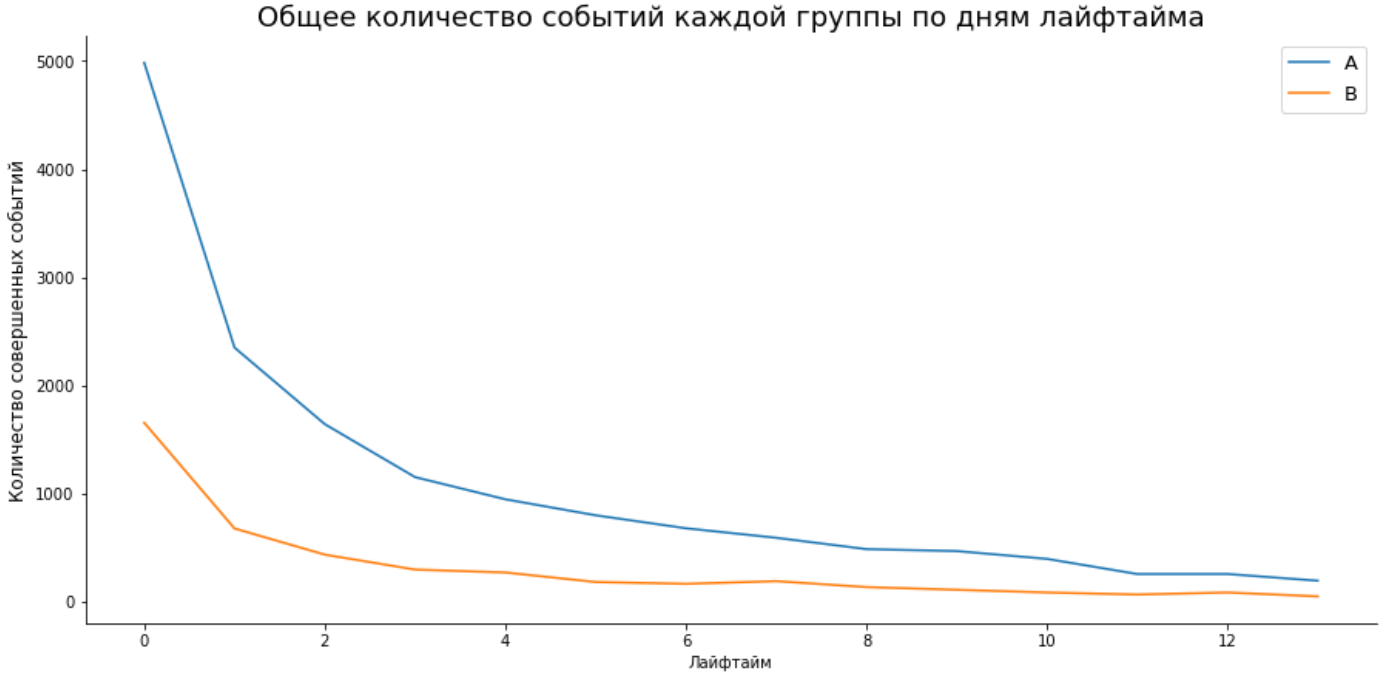
In [49]:
```
# Построим визуализацию с распределением каждого события по дням

# Отберем признаки для диаграмм распределения
features = list(first_event['event_name'].unique())
features.remove('registration')


# Расположим все диаграммы в 2 столбца
num_cols = 2

# Рассчитаем нужное количество строк
if len(features)%num_cols == 0:
    num_rows = len(features)//num_cols
else:
    num_rows = (len(features)//num_cols) + 1

# Определим фигуру и оси
fig, ax = plt.subplots(figsize=(20,10),
                       nrows=num_rows,
                       ncols=num_cols)

# Добавим общее название
fig.suptitle('Распределение количества клиентов на каждом этапе воронке в разрезе по дня
             fontsize=18)

# Напишем цикл для построения диаграмм
for feat in features:
    row = features.index(feat)//num_cols
    col = features.index(feat)%num_cols
    table = first_event.query('event_name == @feat and lifetime <14').groupby(['group','
    sns.lineplot(data=table, x='lifetime', y='event_name',hue='group', ax = ax[row][col]
    ax[row][col].set_title(feat, fontsize=15)
    ax[row][col].set_ylabel('Количество клиентов')
    ax[row][col].set_xlabel('Лайфтайм')
    sns.despine()


plt.subplots_adjust(top=0.9, wspace=0.2, hspace=0.5)
```



Мы видим, что в обеих группах все этапы воронки пользователи проходят в 0-й день лайфтайма, т.е. в день регистрации. Пользователи группы А также часто доходят до этапа корзины на 1-ый день.

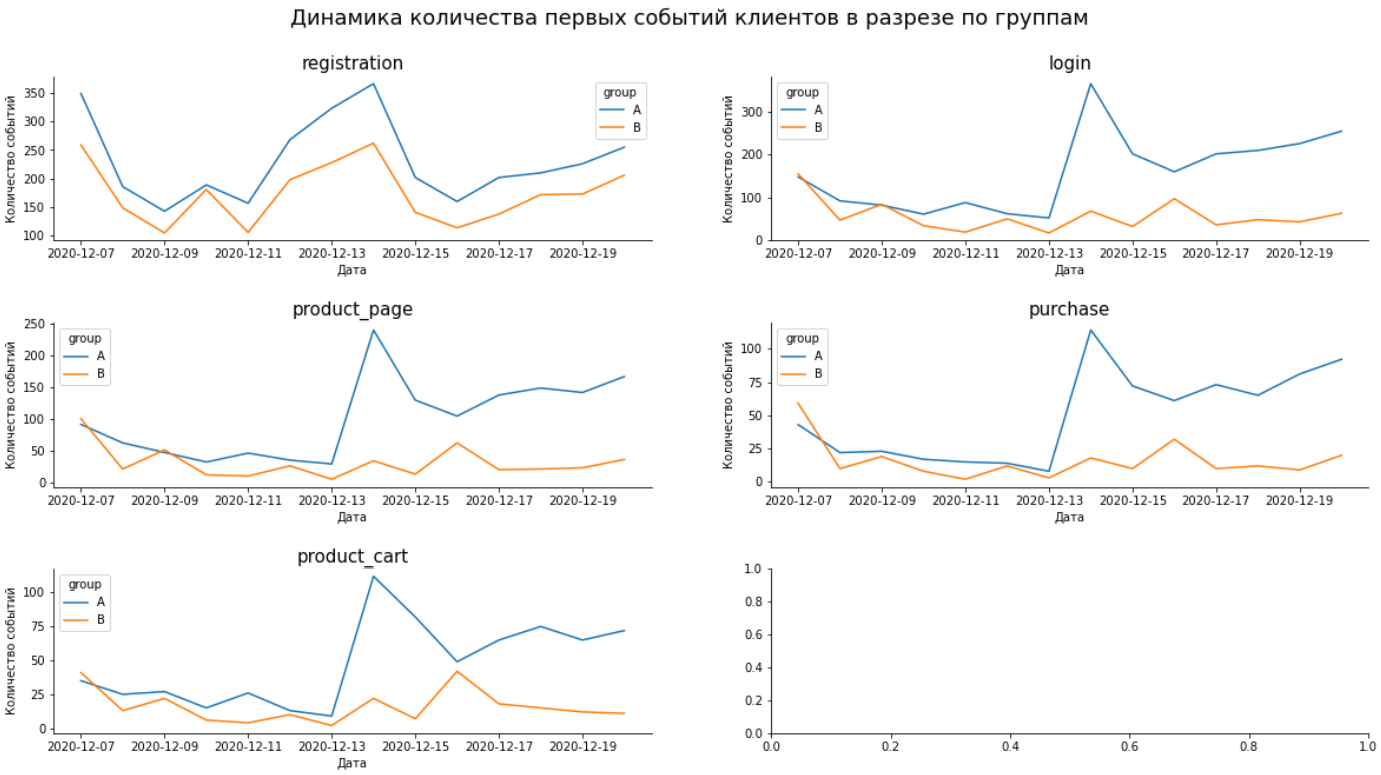
Рассмотрим также динамику всех событий по дням лайфтайма.

In [50]:
```
# Рассмотрим динамику всех событий, совершаемых пользователями по дням лайфтайма, без уч
plt.figure(figsize=(15,7))
sns.lineplot(data = events.query('event_name != "registration"')
                          .groupby(['group','lifetime'])[['event_name']].agg('count').re
             x='lifetime', y='event_name',hue='group')
plt.title('Общее количество событий каждой группы по дням лайфтайма', fontsize=18)
plt.xlabel('Лайфтайм')
plt.ylabel('Количество совершенных событий', fontsize=12)
plt.legend(fontsize=13)
sns.despine()
```



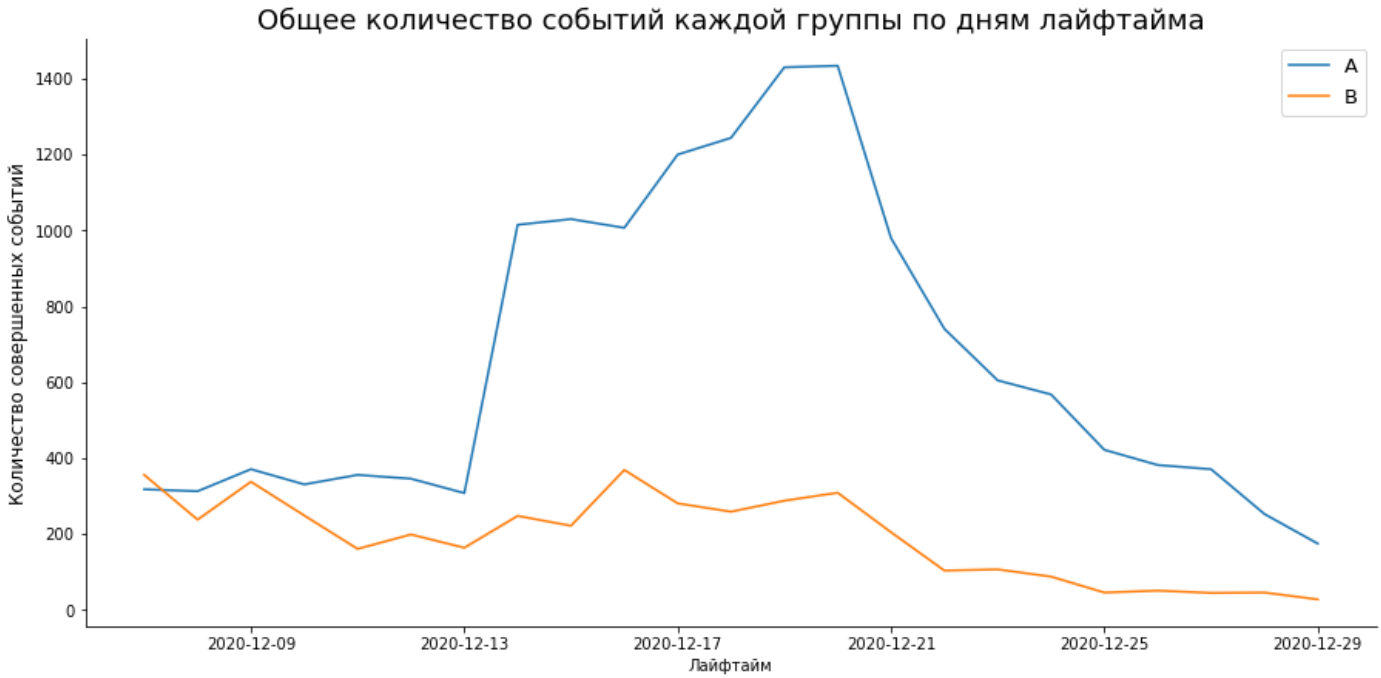
Таким образом, **основная активность пользователей каждой группы приходится на первые дни после регистрации.** При этом активность пользователей группы А выше. **С каждым последующим днем лайфтайма мы видим, что активность пользователей снижается, и при этом результаты группы А все более сопоставимы с результатами группы В.**

In [51]:
```
#Рассмотрим количество событий в разрезе по дням
events['event_date'] = events['event_dt'].dt.date

# Выведем для каждого события каждого пользователя минимальную дату
first_date = events.groupby(['user_id','event_name','group'])[['event_date']].agg('min')

# Построрим график с количеством каждого события по дянм в разрезе по группам
# Отберем признаки для диаграмм распределения
features = list(first_event['event_name'].unique())


# Расположим все диаграммы в 2 столбца
num_cols = 2

# Рассчитаем нужное количество строк
if len(features)%num_cols == 0:
    num_rows = len(features)//num_cols
else:
    num_rows = (len(features)//num_cols) + 1

# Определим фигуру и оси
fig, ax = plt.subplots(figsize=(20,10),
                       nrows=num_rows,
                       ncols=num_cols)

# Добавим общее название
fig.suptitle('Динамика количества первых событий клиентов в разрезе по группам', fontsiz

# Напишем цикл для построения диаграмм
for feat in features:
    row = features.index(feat)//num_cols
    col = features.index(feat)%num_cols
    table = first_date.query('event_name == @feat').groupby(['group','event_date'])[['ev
    sns.lineplot(data=table, x='event_date', y='event_name',hue='group', ax = ax[row][co
    ax[row][col].set_title(feat, fontsize=15)
    ax[row][col].set_ylabel('Количество событий')
    ax[row][col].set_xlabel('Дата')
    sns.despine()


plt.subplots_adjust(top=0.9, wspace=0.2, hspace=0.5)
```



Рассмотрим также общую динамику всех событий по дням теста.

In [52]:
```
# Построим визуализацию с динамикой количества событий по дням теста
plt.figure(figsize=(15,7))
sns.lineplot(data = events.query('event_name != "registration"')
                          .groupby(['group','event_date'])[['event_name']].agg('count').
             x='event_date', y='event_name',hue='group')
plt.title('Общее количество событий каждой группы по дням лайфтайма', fontsize=18)
plt.xlabel('Лайфтайм')
plt.ylabel('Количество совершенных событий', fontsize=12)
plt.legend(fontsize=13)
sns.despine()
```



В первый день тестирования (7 декабря) количество событий каждого на каждом этапе воронке, совершенных пользователями из группы B больше, чем пользователями группы A, учитывая, что группа B состоит из меньшего количества пользователей. Далее до 13 декабря, зная, что группа A

более многочисленная, мы видим в обеих группах похожую динамику, количество событий для каждого этапа менялось не сильно.

Однако, **начиная с 14 декабря и далее мы видим резкий рост количества событий, совершенных группой A, существенно превыщающий результаты пользователей группы B.** Важно, что в этот день действительно отмечался прирост новых пользователей, но он был характерен как для группы A, так и для группы B, а среди пользователей группы B таких результатов нет. Динамика их количества событий мало верьируется на протяжении всего периода.

Проверим, проводились ли какие-либо маркетинговые события в этот период.

In [53]:
```
# Выведем маркетинговые события, проводившиеся во время тестирования

# Для каждого события проверим, проводилось ли оно в Европе
marketing_events['is_eu'] = marketing_events['regions'].apply(lambda x: 'EU' in x.split(

# Для проверки установим дату начала и окончания теста
start_test = events['event_date'].min()
finish_test = events['event_date'].max()

# Проверим на наличие разные виды событий
print('Начались до начала проведения тестирования, закончились во время')
display(marketing_events.query('start_dt <=@ start_test and finish_dt>= @start_test and\
                                finish_dt <= @finish_test and is_eu==True'))

print('Начались до начала проведения тестирования, закончились после')
display(marketing_events.query('start_dt <=@ start_test and finish_dt >= @finish_test an

print('Начались во время проведения тестирования, закончились после')
display(marketing_events.query('start_dt >= @ start_test and start_dt <= @ finish_test a
                                finish_dt >= @finish_test and is_eu==True'))
```

```
Начались до начала проведения тестирования, закончились во время
```

name	regions	start_dt	finish_dt	is_eu

```
Начались до начала проведения тестирования, закончились после
```

name	regions	start_dt	finish_dt	is_eu

```
Начались во время проведения тестирования, закончились после
```

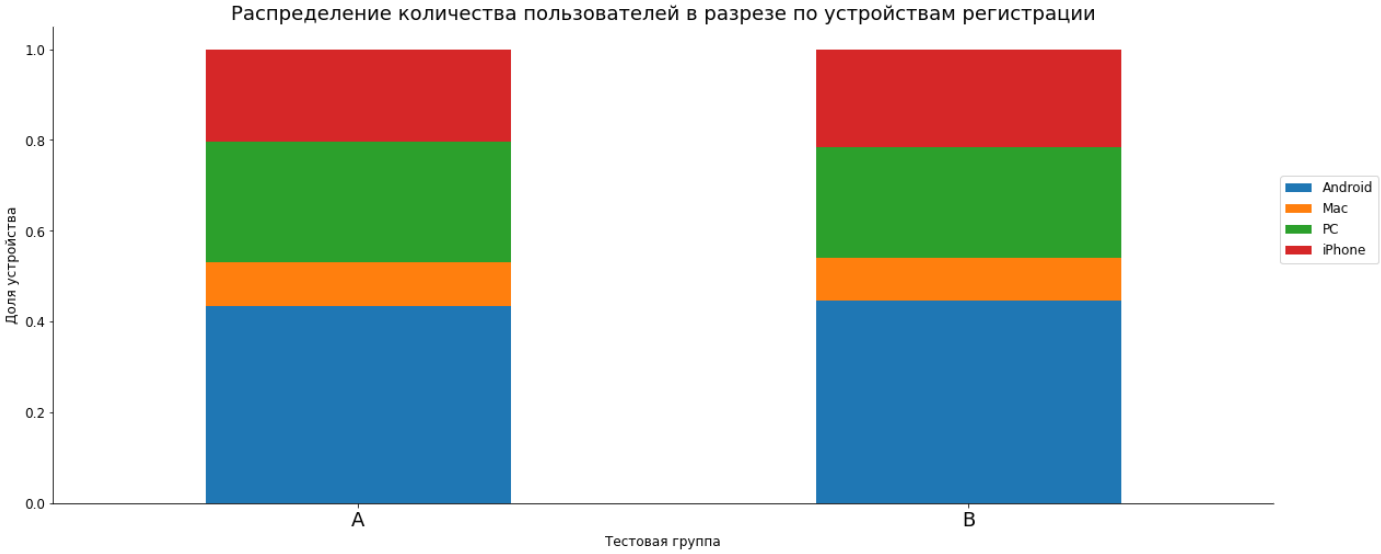
	name	regions	start_dt	finish_dt	is_eu
0	Christmas&New Year Promo	EU, N.America	2020-12-25	2021-01-03	True

25 декабря стартовала промо-акция, подготовленная под Рождество и Новый Год. Это событие совершенно не повлияло на результаты нашего теста, поскольку к этому моменту во-первых уже завершился набор пользователей тест, во-вторых абсолютно все пользователи фактически уже закончили участвовать в нашем тестировании, поскольку ранее мы выявили, что события каждого этапа приходятся на 0-й день лайфтайма и последнее "первое событие" нового пользователи состоялось 20 декабря.

Устройства пользователей и средний чек

In [54]:
```
# Рассмотрим, с каких устройств чаще всего заходили пользователи в каждой группе
device = events.pivot_table(index='device', columns='group',values='user_id', aggfunc='n
device = device / device.sum()
device.T.plot.bar(stacked=True, figsize = (20,8))
plt.title('Распределение количества пользователей в разрезе по устройствам регистрации',
```

```
plt.ylabel('Доля устройства', fontsize=12)
plt.xlabel('Тестовая группа', fontsize=12)
plt.xticks(fontsize = 18, rotation=0)
plt.yticks(fontsize = 12)
plt.legend(bbox_to_anchor=(1.0, 0.7), loc='upper left', fontsize=12)
sns.despine();
```



Из диаграммы, построенной выше, мы видим, что **распределение количества пользователей в разрезе по устройствам, с которого они регистрировались в нашем интернет-магазине между группами одинаково**, а значит не оказывает влияния на результаты нашего тестирования.

Рассмотрим также средний чек пользователя в каждой группе.

In [55]:
```
# Рассчитаем средний чек первой покупки среди пользователей каждой группы
(
    events.sort_values(by='event_date', ascending=False)
          .query('event_name == "purchase"')
          .drop_duplicates(subset='user_id', keep='first')
          .groupby('group')[['details']].agg(['mean','median']).round(2)
)
```

Out[55]:

	details	
	mean	median
group		
A	23.93	4.99
B	23.12	4.99

Средний и медианный чеки первой покупки также различаются не сильно.\ Большая часть пользователей приобретаем самый дешевый товар стоимость 4.99 у.е.

Анализ воронки продаж

Проведем анализ воронки продаж и рассмотрим конверсию на каждом этапе воронки.

In [56]:
```
# Построим сводную таблицу с количеством событий для каждой группы
sales_funnel = (events.pivot_table(index='event_name',columns = 'group', values = 'user_
                      .reindex(['registration','login','product_page','product_cart','pu
                )
```

```
# Рассчитаем конверсию к кол-ву зарегистрированных пользователей
sales_funnel['A_конверсия от регистрации'] = (sales_funnel['A']/sales_funnel.loc[0,'A']*
sales_funnel['B_конверсия от регистрации'] = (sales_funnel['B']/sales_funnel.loc[0,'B']*

# Рассчитаем конверсию к колву-авторизованных пользователей
sales_funnel['A_конверсия от авторизации'] = (sales_funnel['A']/sales_funnel.loc[1,'A']*
sales_funnel['B_конверсия от авторизации'] = (sales_funnel['B']/sales_funnel.loc[1,'B']*
sales_funnel.loc[0,'A_конверсия от авторизации'] = np.nan
sales_funnel.loc[0,'B_конверсия от авторизации'] = np.nan

sales_funnel
```

Out[56]:

	group	event_name	A	B	A_конверсия от регистрации	B_конверсия от регистрации	A_конверсия от авторизации	B_конверсия от авторизации
0		registration	3236	2432	100.00	100.00	NaN	NaN
1		login	2206	793	68.17	32.61	100.00	100.00
2		product_page	1420	448	43.88	18.42	64.37	56.49
3		product_cart	670	225	20.70	9.25	30.37	28.37
4		purchase	700	224	21.63	9.21	31.73	28.25

Результаты конверсии группы B на всех этапах воронки хуже группы A. При этом, если за первый шаг воронки брать регистрацию пользователя, то мы видим, что только 32,6% пользователей группы B провели хотя бы авторизацию на сайте. До стадии покупки дошло только только 9,2% пользователей, прошедших регистрацию, и 28,3% пользователей, совершивших хотя бы одно событие (авторизацию).

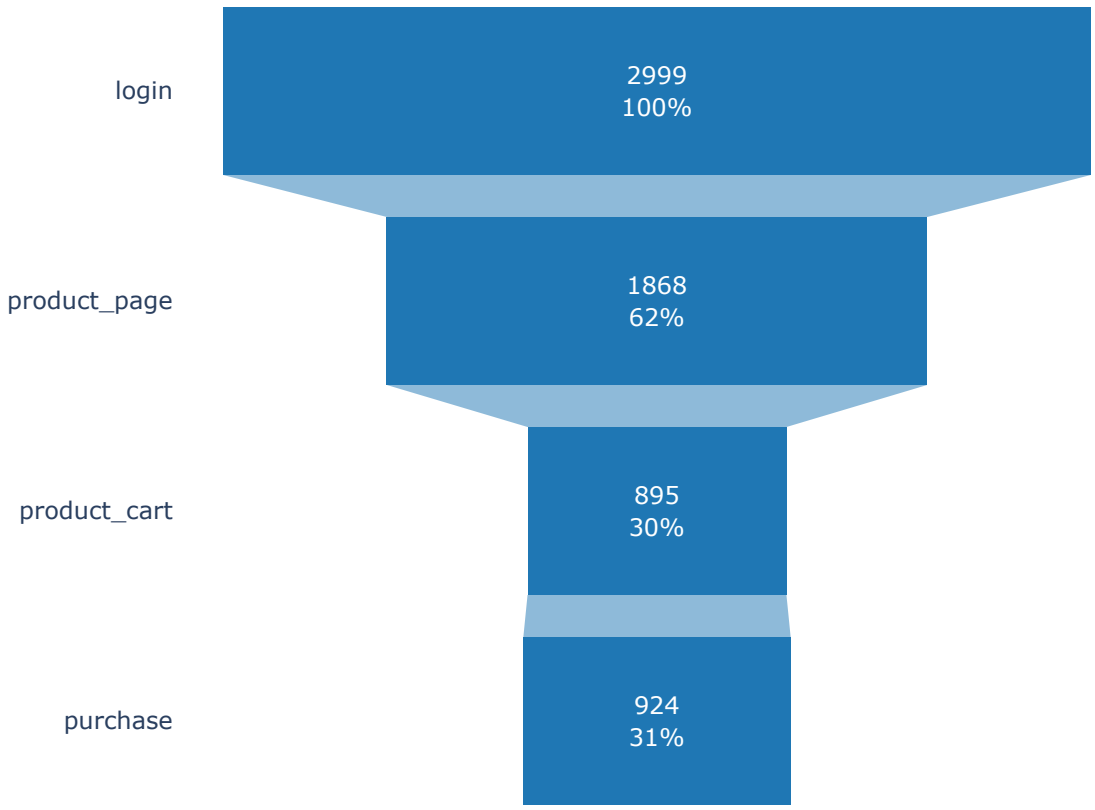
В то же время, среди пользователей группы A авторизацию на сайте совершили 68% зарегистрированных пользователей, а покупку 21,6% из них (и 31,7% от количества авторизовавшихся пользователей).

Важно отметить, что **некоторые шаги воронки (product_page, product_cart) являются опциональными.** Т.е. пользователь может перейти к процессу покупки, не просматривая карточку товара и не кладя его в корзину. За счет этого мы видим, что количество сделок в группе A меньше кол-ва корзин.

Для наглядности визуализируем воронку продаж. Поскольку рекомендательная система начинает работать только после получения данных о предпочтениях пользователей, **конверсия этапа регистрация-авторизация в данном исследовании является для нас скорее информационным и не влияет на действие рекомендательной системы. Поэтому все дальнейшие визуализации и проверку статистических гипотез мы будем от вести начиная с этапа login.**

In [57]:
```
# Подготовим данные для построения визуализации
funnel_data = sales_funnel[['event_name','A','B']].copy()
funnel_data = funnel_data[1:]
funnel_data['total'] = sales_funnel['A']+ sales_funnel['B']

# Визуализируем воронку продаж
fig = go.Figure(go.Funnel( y = funnel_data['event_name'],
                           x = funnel_data['total'],
                           textposition = "inside",
                           textinfo = "value+percent initial",
                          marker = {'color' : '#1F77B4'})
    )
```

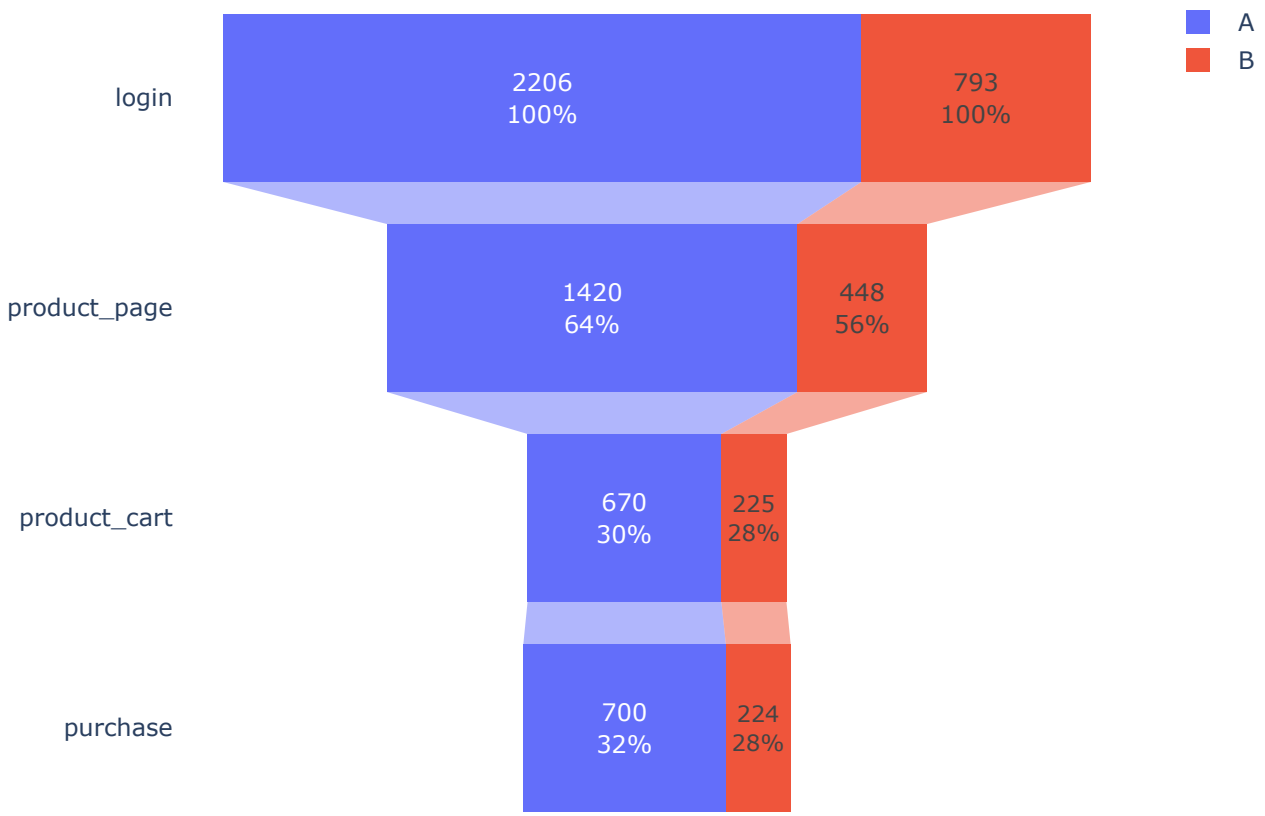
```
fig.update_layout(title={'text': '<b>Воронка продаж </b>', 'font': {'size': 18}},
                  plot_bgcolor="rgba(0,0,0,0)", height=600)
fig.show()
```



В общем виде, среди 2999 участников теста, прошедших после регистрации также авотризацию на сайте, до стадии покупки дошли 30,8%.

In [58]:
```
# Визуализируем воронку продаж в разрезе по группам

fig = go.Figure()

fig.add_trace(go.Funnel(
    name = 'A',
    y = funnel_data['event_name'],
    x = funnel_data['A'],
    textinfo = "value+percent initial"))

fig.add_trace(go.Funnel(
    name = 'B',
    orientation = "h",
    y = funnel_data['event_name'],
    x = funnel_data['B'],
    textposition = "inside",
    textinfo = "value+percent initial"))
```

```
fig.update_layout(title={'text': '<b>Воронка продаж</b>', 'font': {'size': 18}},
                  plot_bgcolor="rgba(0,0,0,0)", height=600)
fig.show()
```



В разрезе по группам мы видим, что результаты группы B на всех этапах воронки хуже группы A. При этом наибольшее отклонение между группами мы видим на этапе авторизация-просмотр карточки товара, только 56% авторизованных пользователей изучили информацию о товаре.

Поскольку в нашей воронке некоторые стадии опциональны, рассмотрим еще один вариант воронки, засчитывающий "проскочившие" стадии как пройденные этапы.

In [59]:
```
# Построим воронку продаж с "виртуальными" стадиями
sales_funnel_virtual = sales_funnel[['event_name','A','B']].copy()

# Для значений этапов "просмотр карточки товара" и "формирование корзины" учтем пропущен
for group in ['A','B']:
    product_page = list(events.query('group == @group and event_name == "product_page"')
    next_stages = len(list(events.query('group == @group and event_name in ["product_car
    correct_pp = len(product_page) + next_stages
    sales_funnel_virtual.loc[2, group] = correct_pp

    product_cart = list(events.query('group == @group and event_name == "product_cart"')
    next_stages = len(list(events.query('group == @group and event_name =="purchase" and
    correct_pc = len(product_cart) + next_stages
    sales_funnel_virtual.loc[3, group] = correct_pc
```

```
# Рассчитаем конверсию к количеству зарегистрированных пользователей
sales_funnel_virtual['A_конверсия от регистрации'] = (sales_funnel_virtual['A']/sales_fu
sales_funnel_virtual['B_конверсия от регистрации'] = (sales_funnel_virtual['B']/sales_fu

# Рассчитаем конверсию к количеству авторизованных пользователей
sales_funnel_virtual['A_конверсия от авторизации'] = (sales_funnel_virtual['A']/sales_fu
sales_funnel_virtual['B_конверсия от авторизации'] = (sales_funnel_virtual['B']/sales_fu

sales_funnel_virtual.loc[0,'A_конверсия от авторизации'] = np.nan
sales_funnel_virtual.loc[0,'B_конверсия от авторизации'] = np.nan

# Рассчитаем конверсию к предыдщему этапу
sales_funnel_virtual['A_конверсия к предыдущему этапу'] = (sales_funnel_virtual['A']/sal
sales_funnel_virtual['B_конверсия к предыдущему этапу'] = (sales_funnel_virtual['B']/sal

sales_funnel_virtual
```

Out[59]:

group	event_name	A	B	A_конверсия от регистрации	B_конверсия от регистрации	A_конверсия от авторизации	B_конверсия от авторизации	A_конверсия к предыдущему этапу	B_ко пред
0	registration	3236	2432	100.00	100.00	NaN	NaN	NaN	
1	login	2206	793	68.17	32.61	100.00	100.00	68.17	
2	product_page	1797	603	55.53	24.79	81.46	76.04	81.46	
3	product_cart	1142	379	35.29	15.58	51.77	47.79	63.55	
4	purchase	700	224	21.63	9.21	31.73	28.25	61.30	

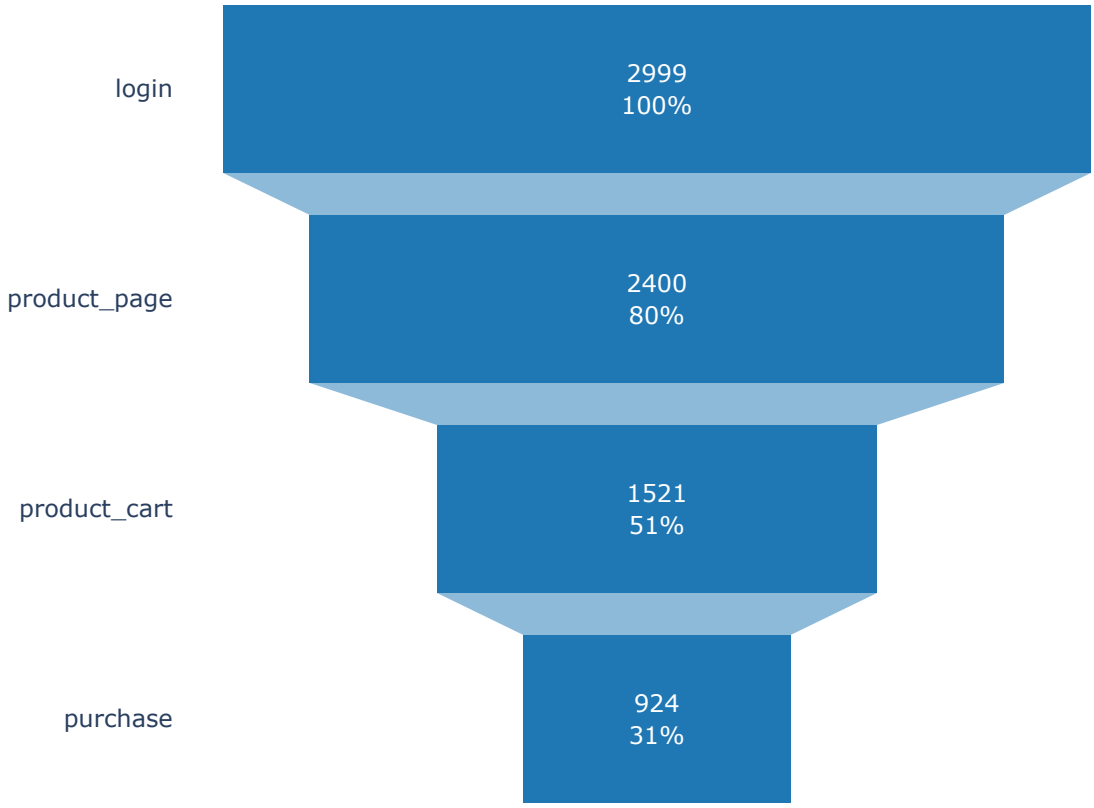
В данной таблице мы учли для пользователей, которые переходят к стадии покупки пропуская некоторые этапы воронки, стадии "просмотр карточки продукта" и "формирование корзины". Таким образом, мы сможем лучше оценить, на каком из этапов теряется большее количество пользователей. В частности, на прошлом шаге мы выявили, что за период теста пользователи группы А 670 раз формировали корзину и совершили 700 покупок. Здесь же мы видим, что **до этапа покупки дошли только 61,3% пользователей из тех, что формировали корзину. Для группы В эта цифра еще немного ниже и составляет 59,1%.**

Однако, учитывая, что базовая конверсия на каждом шаге составляет 50%, мы видим, что обе группы на каждом шаге (кроме этапа регистрация-авторизация в группе В) показали результаты значительно выше этого показателя.

In [60]:

```
# Подготовим данные для визуализиции
funnel_data2 = sales_funnel_virtual[['event_name','A','B']].copy()
funnel_data2 = funnel_data2[1:]
funnel_data2['total'] = sales_funnel_virtual['A']+ sales_funnel_virtual['B']

# Визуалищируем воронку продаж
fig = go.Figure(go.Funnel( y = funnel_data2['event_name'],
                           x = funnel_data2['total'],
                           textposition = "inside",
                           textinfo = "value+percent initial",
                          marker = {'color' : '#1F77B4'})
    )

fig.update_layout(title={'text': '<b>Воронка продаж с учетом виртуальных стадий</b>', 'f
                  plot_bgcolor="rgba(0,0,0,0)", height=600)
fig.show()
```

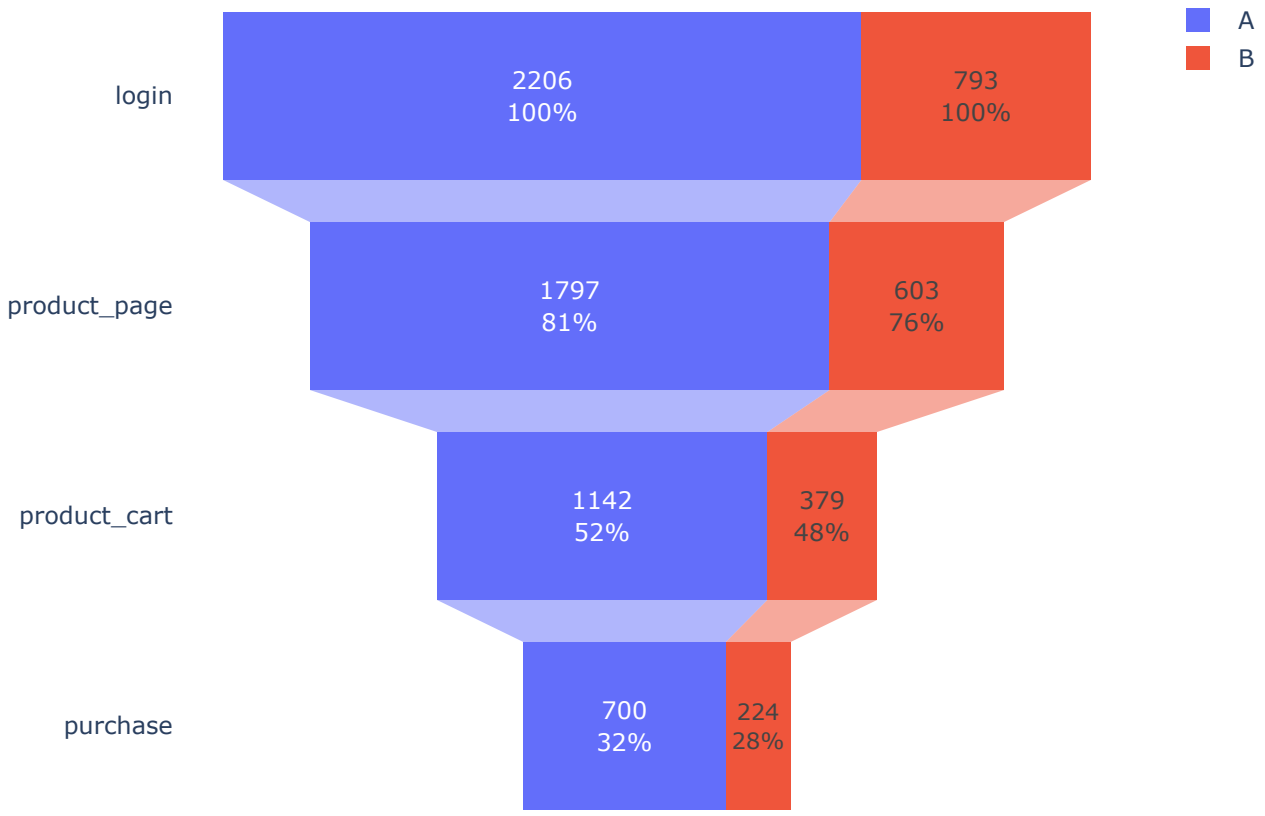
Воронка продаж с учетом виртуальных стадий



В целом, за время проведения нашего теста, до покупки дошли 60,7% пользователей среди тех, кто формировал корзину.

In [61]:
```
# Визуализируем воронку продаж в разрезе по группам

fig = go.Figure()

fig.add_trace(go.Funnel(
    name = 'A',
    y = funnel_data2['event_name'],
    x = funnel_data2['A'],
    textinfo = "value+percent initial"))

fig.add_trace(go.Funnel(
    name = 'B',
    orientation = "h",
    y = funnel_data2['event_name'],
    x = funnel_data2['B'],
    textposition = "inside",
    textinfo = "value+percent initial"))

fig.update_layout(title={'text': '<b>Воронка продаж с учетом виртуальных стадий</b>', 'f
                  plot_bgcolor="rgba(0,0,0,0)", height=600)
fig.show()
```

Воронка продаж с учетом виртуальных стадий



Из визуализации выше мы видим, что результаты группы A лучше результатов группы B на каждом из этапов воронки даже если учитывать "проскочившие" стадии.

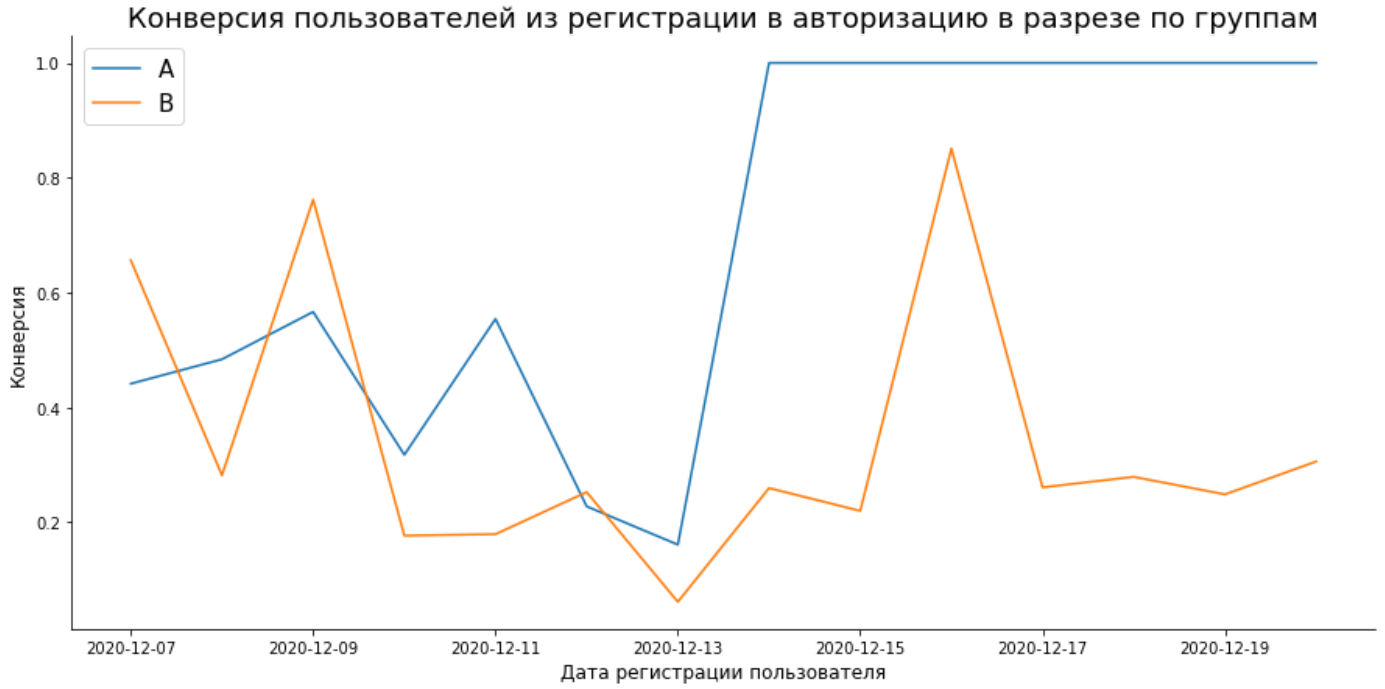
Анализ воронки продаж в разрезе по когортам

Рассмотрим конверсию пользователей по этапам регистрация-авторизация и авторизация-совершение покупки в разрезе по датам регистрации пользователей и накопленным итогом.

In [62]:
```
# Подготовим данные, сгруппируем количество пользователей на каждом этапе в зависимости
cohorts = (
            events.pivot_table(index=['group','first_date'], columns='event_name', values
                  .reset_index()
)

# Рассчитаем показатели конверсии
cohorts['reg-log'] = cohorts['login']/cohorts['registration']
cohorts['log-deal'] = cohorts['purchase']/cohorts['login']

# Для каждой группы посчитаем данные по авторизациям и покупкам накопительным итогом, а
cohorts.loc[cohorts['group'] == "A",'login_cum'] = cohorts.loc[cohorts['group'] == "A",'
cohorts.loc[cohorts['group'] == "B",'login_cum'] = cohorts.loc[cohorts['group'] == "B",'
cohorts.loc[cohorts['group'] == "A",'purchases_cum'] = cohorts.loc[cohorts['group'] == "
cohorts.loc[cohorts['group'] == "B",'purchases_cum'] = cohorts.loc[cohorts['group'] == "
cohorts['log-deal_cum'] = cohorts['purchases_cum']/cohorts['login_cum']
```

In [63]:
```
# Построрим график с конверсией пользователей из регистрации в авторизацию в зависимости
plt.figure(figsize=(15,7))
```

```
sns.lineplot(data=cohorts, x='first_date', y=cohorts['reg-log'],hue='group')
plt.title('Конверсия пользователей из регистрации в авторизацию в разрезе по группам', f
plt.ylabel('Конверсия', fontsize=12)
plt.xlabel('Дата регистрации пользователя', fontsize=12)
plt.legend(fontsize=15)
sns.despine();
```

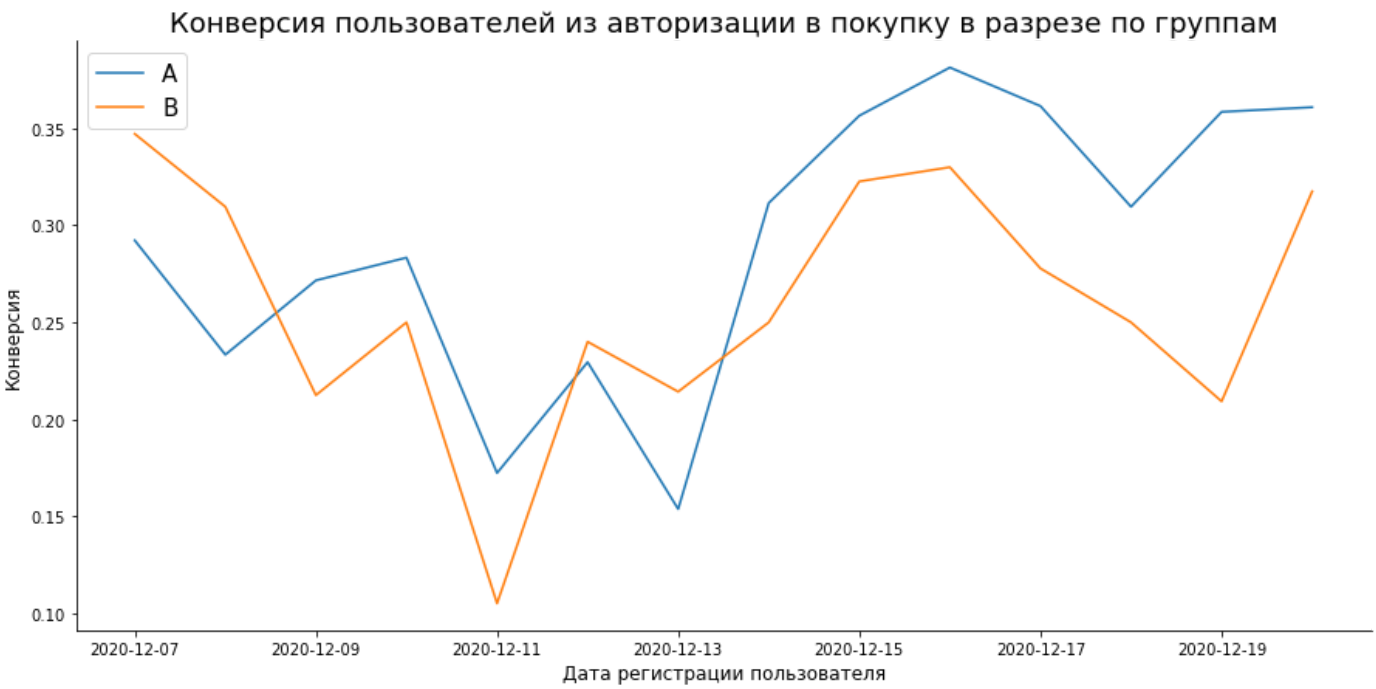


Таким образом, **конверсия из регистрации пользователя в авторизацию на сайте для пользователей, зарегистрированных в период с 7 по 10 декабря сопоставима.**

Однако,далее мы видим резкий рост конверсии пользователей группы А, зарегистрированных в период с 14 декабря, и при этом на протяжении всего дальшнейшего периода она остается стабильной. В группе В таких результатов нет, конверсия пользователей, зарегистрированных после 10 декабря напротив, даже снизилась по сравнению с пользователями, зарегистрированными до этой даты. (исключение - скачок конверсии 16 декабря).

Рассмотрим конверсию пользователей из авторизации на сайте в покупку.

In [64]:
```
# Построрим график с конверсией пользователей из авторизации в покупку в разрезе по кого
plt.figure(figsize=(15,7))
sns.lineplot(data=cohorts, x='first_date', y=cohorts['log-deal'],hue='group')
plt.title('Конверсия пользователей из авторизации в покупку в разрезе по группам', fonts
plt.ylabel('Конверсия', fontsize=12)
plt.xlabel('Дата регистрации пользователя', fontsize=12)
plt.legend(fontsize=15)
sns.despine();
```

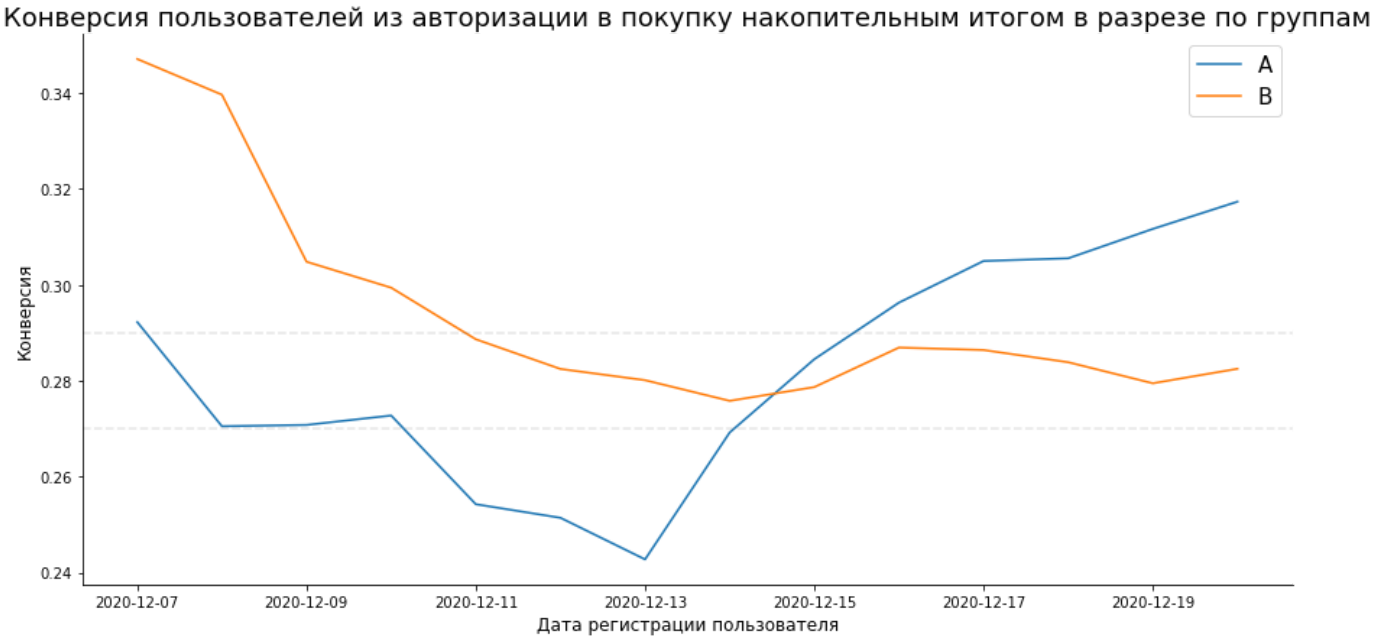


В период с 7 по 13 декабря результаты конверсии между группами варьируются и однозначно определить лидера нельзя.

Однако, далее мы видим, что **пользователи группы А, зарегистрированные после 14 декабря, демонстрируют более высокие показатели конверсии в покупку, по сравнению с результатами пользователей группы В.**

Рассмотрим динамику конверсии накопленным итогом.

In [65]:
```
# Построрим график с конверсией пользователей из авторизации в покупку в разрезе по кого
plt.figure(figsize=(15,7))
sns.lineplot(data=cohorts, x='first_date', y=cohorts['log-deal_cum'],hue='group')
plt.title('Конверсия пользователей из авторизации в покупку накопительным итогом в разре
plt.ylabel('Конверсия', fontsize=12)
plt.xlabel('Дата регистрации пользователя', fontsize=12)
plt.legend(fontsize=15)
plt.axhline(y=0.29, color='grey', linestyle='--', alpha=0.2)
plt.axhline(y=0.27, color='grey', linestyle='--', alpha=0.2)
sns.despine();
```



Рассматривая показатели конверсии накопленным итогом, мы видим, что **до 14 декабря конверсия из авторизации на сайте в покупку пользователей группы B существенно лучше результатов группы A. Хотя при этом обе группы демонстрировали снижение показателя накопленным итогом.**

Однако, **начиная с 13 декабря показатель конверсии группы A начали быстро восстанавливаться, и в результате к концу теста составил 31,7%, что выше, чем в группе B, где, начиная с 12 декабря показатель конверсии стабилизировался и до окончания теста находился около 28%.**

Общий вывод по разделу:

1. **Несмотря на то, что количество пользователей в группе B меньше, чем в группе A, среди них наибольшее количество покупателей, которые совершили только 1 событие - регистрация.** Среднее значение кол-ва событий, совершаемых пользователем этой группы, составляет 2,8.В то время, как **пользователи группы A взаимодействуют с сайтом гораздо активнее, среднее количество событий для этой группы - 5,7.** \ Если не учитывать этап регистрации и рассматривать среднее количество событий среди пользователей, которые как минимум прошли авторизацию, мы видим, что разница в количестве событий между группами становится небольшой, но при этом пользователи группы A по-прежнему более активнее совершают действия на сайте, чем в группе B.

2. **Основная активность пользователей каждой группы приходится на первые дни после регистрации.** При этом активность пользователей группы A выше. **С каждым последующим днем лайфтайма мы видим, что активность пользователей снижается, и при этом результаты группы A все более сопоставимы с результатами группы B.**

1. **В обеих группах все этапы воронки пользователи проходят в 0-й день лайфтайма, т.е. в день регистрации.** Пользователи группы A также часто доходят до этапа корзины на 1-ый день.

1. **Начиная с 14 декабря и далее мы видим резкий рост количества событий, совершенных группой A, существенно превыщающий результаты пользователей группы B. Важно, что в этот день действительно отмечался прирост новых пользователей, но он был характерен как для группы A, так и для группы B, а среди пользователей группы B таких результатов нет.** Динамика их количества событий мало верьируется на протяжении всего периода.

1. **Среди маркетинговых мероприятий нет событий, которые могли бы повлиять на результаты нашего теста.** 25 декабря стартовала промо-акция, подготовленная под Рождество и Новый Год. Это событие совершенно не повлияло на результаты нашего теста, поскольку к этому моменту во-первых уже завершился набор пользователей тест, во-вторых абсолютно все пользователи фактически уже закончили участвовать в нашем тестировании, поскольку ранее мы выявили, что события каждого этапа приходятся на 0-й день лайфтайма и последнее "первое событие" нового пользователи состоялось 20 декабря.

1. **Распределение количества пользователей в разрезе по устройствам, с которого они регистрировались в нашем интернет-магазине между группами одинаково**, а значит не оказывает влияния на результаты нашего тестирования.

1. **Средний и медианный чеки первой покупки также различаются не сильно.** Большая часть пользователей приобретаем самый дешевый товар стоимость 4.99 у.е.

1. В общем виде, **среди 5568 участников теста до стадии покупки дошли 16% зарегистрированных пользователей и 30,8% от числа, прошедших авторизацию на сайте.**

1. **Результаты конверсии группы B на всех этапах воронки хуже группы A.** До стадии покупки дошло только только 9,2% пользователей, прошедших регистрацию, и 28,3% пользователей, совершивших хотя бы одно событие (авторизацию). В то же время, среди пользователей группы A покупку совершили 21,6% зарегистрированных пользователей и 31,7% от количества авторизовавшихся пользователей).

1. **В группе A до этапа покупки дошли только 61,3% пользователей из тех, что формировали корзину. Для группы B эта цифра еще немного ниже и составляет 59,1%.** целом, за время проведения нашего теста, до покупки дошли 60,7% пользователей среди тех, кто формировал корзину.

1. Конверсия из регистрации пользователя в авторизацию на сайте для пользователей, зарегистрированных в период с 7 по 10 декабря сопоставима. После этого периода мы видим **резкий рост конверсии пользователей группы A, зарегистрированных в период с 14 декабря, и при этом на протяжении всего дальшнейшего периода она остается стабильной. В группе B таких результатов нет**, конверсия пользователей, зарегистрированных после 10 декабря напротив, даже снизилась по сравнению с пользователями, зарегистрированными до этой даты. (исключение - скачок конверсии 16 декабря).

1. В период с 7 по 13 декабря результаты конверсии между группами варьируются и однозначно определить лидера нельзя. Однако, далее мы видим, что **пользователи группы A, зарегистрированные после 14 декабря, демонстрируют более высокие показатели конверсии в покупку, по сравнению с результатами пользователей группы B.**

1. Рассматривая показатели конверсии накопленным итогом, мы видим, что **до 14 декабря конверсия из авторизации на сайте в покупку пользователей группы B существенно лучше результатов группы A.** Хотя при этом обе группы демонстрировали снижение показателя накопленным итогом. Однако, **начиная с 13 декабря показатель конверсии группы A начали быстро восстанавливаться, и в результате к концу теста составил 31,7%, что выше, чем в группе B**, где, начиная с 12 декабря показатель конверсии стабилизировался и до окончания теста находился около 28%.

Расчет статистической значимости различий между группами

Согласно нашему техническому заданию ожидаемым эффектом от нашего A/B является рост конверсии из авторизации на сайте в каждый шаг (просмотр карточки товара, формирование корзины, покупка) на 5 процентных пунктов.

In [66]:
```
# Подготовим данные к оценке
result = sales_funnel[['event_name','A','B']].T
result.columns = result.iloc[0]
result = result.drop(result.index[0])
result
```

Out[66]:

event_name	registration	login	product_page	product_cart	purchase

group					
A	3236	2206	1420	670	700
B	2432	793	448	225	224

Мы проведем 3 статистических теста, в которых будем проверять наличие статистической разницы между конверсией из авторизации на сайте в каждый этап воронки.

Сформулируем гипотезы для теста:\ **Нулевая гипотеза:** Нет различий между конверсией из авторизации в этап воронки а между группами A и B\ **Альтернативная гипотеза:** Есть различия между конверсией из авторизации в этап воронки товара между группами A и B

Для проверки гипотез мы будем использовать на основе двухвыборочного Z-критерия. Поскольку мы будем проводить 3 теста на одних данных применим также поправку Шидака.

In [67]:
```
# Отберем необходимые стадии конверсии
stages = ['product_page','product_cart','purchase']

# Напишем цикл, сравнивающей конверсию из авторизации в каждый этап между группами
for stage in stages:
    nobs = list(result['login'].values)
    count = list(result[stage].values)

    alpha = 0.05 # уровень значимости

    shidaka_alpha = 1 - (1 - alpha)**(1/3) # поправка Шидака

    pvalue = proportions_ztest(count, nobs)[1]

    print('Конверсия login - {}'.format(stage))
    print('Конверсия группы A составляет {}%'.format(round((result[stage][0]/result['log
    print('Конверсия группы B составляет {}%'.format(round((result[stage][1]/result['log
    print('Преимущество группы B над A составляет {} п.п.'
                            .format(round((result[stage][1]/result['login'][1] - result

    print('p-value: {}'.format(pvalue))

    if pvalue < shidaka_alpha:
        print(color.RED + 'Отвергаем нулевую гипотезу, наблюдается статистически значима
 в конверсии между результатами групп' + color.END)
    else:
        print(color.GREEN + 'Не получилось отвергнуть нулевую гипотезу, вывод о различии
    print('\n')
```

```
Конверсия login - product_page
Конверсия группы A составляет 64%
Конверсия группы B составляет 56%
Преимущество группы B над A составляет -8 п.п.
p-value: 8.689121794414493e-05
Отвергаем нулевую гипотезу, наблюдается статистически значимая разница в конверсии между
 результатами групп


Конверсия login - product_cart
Конверсия группы A составляет 30%
Конверсия группы B составляет 28%
Преимущество группы B над A составляет -2 п.п.
p-value: 0.29149924742534616
Не получилось отвергнуть нулевую гипотезу, вывод о различии конверсии сделать нельзя


Конверсия login - purchase
```

```
Конверсия группы A составляет 32%
Конверсия группы B составляет 28%
Преимущество группы B над A составляет -3 п.п.
p-value: 0.0683455866854418
Не получилось отвергнуть нулевую гипотезу, вывод о различии конверсии сделать нельзя
```

Таким образом, **z-тест показал наличие статистической разницы между группами в конверсии этапа авторизация на сайте-просмотр карточки товаров, а значит нулевая гипотеза отвергается. При этом коверсия группы B на 8% ниже, чем в группе A.**

На этапе авторизация на сайте-формирование корзины p-value значительно больше уровня стат. значимости, а значит **статистической разницы между результатами групп нет и у нас нет оснований для отвержения нулевой гипотезы. Однако, результаты конверсии группы B на данном этапе на 2% хуже A.**

На этапе воронки авторизация на сайте-покупка p_value составил 0.07. Это выше уровня статистической значимости, которая с учетом поправки Шидака составляет 0.017, а значит **тест говорит об отсутствии статистической разницы между результатами групп. Однако, стоит учитывать, что уровень 0.07 является достаточно небольшим, что может говорит о том, что некоторая разница в данных все же присутствует. При этом, конверсия группы B на 3% ниже, чем в группе A.**

Выводы и рекомендации

1. **Тест был проведен в соответствии с техническим заданием с небольшими, практически незначительными нарушениями.**

- **Период набора новых пользователей в тест с 7 по 20 декабря 2020 года** : **Выполнено**.
- **Дата остановки теста - 4 января 2021 года**: **Есть незначительные нарушения**. В данных присутствуют события пользователей только до 30 декабря включительно. Однако, в процессе анализа мы выявили, что большинство переходов по этапам воронки совершаются в первый же день после регистрации, а значит у всех пользователей была возможность пройти воронку полностью.
- **Ожидаемое количество участников теста - 15% новых пользователей из региона EU** - **Выполнено**. В тест действительно попало 15% новых пользователей Европы, зарегистрированных в период с 7 по 20 октября и это подтверждено статистическим тестом.
- **Распределение пользователей по группам происходило неравномерно, но минимальный размер выбор достаточен для анализа**. Вероятность попасть в группу A для пользователя составляет 57%. В группу B попало 2121 человек (при достаточном минимальном размере выборки в 1567 чел).
- **Набор пользователей в тест происходил каждую неделю равномерно**. Динамика набора в тест по дням соответствует общей динамике регистраций на сайте. Распределение пользователей по группам практически на протяжении всего периода (исключение 10 декабря) также проходил с одинаковой динамикой, при этом большая часть пользователей ежедневно относилась к группе A.
- **Из 5668 пользователей, прошедших регистрацию и попавших в тест, только 3000 (53%) прошли хотя бы первый этап воронки.** При этом среди пользователей, не дошедших даже до

первого этапа, преобладают пользователи из группы B. Мы добавили в датафрейм с событиями новое событие `registration` для каждого пользователя, участвующего в тесте.

1. **Исследовательский анализ даных показал следующие результаты:**

- **Несмотря на то, что количество пользователей в группе B меньше, чем в группе A, среди них наибольшее количество покупателей, которые совершили только 1 событие - регистрация.** Если не учитывать этап регистрации и рассматривать среднее количество событий среди пользователей, которые как минимум прошли авторизацию, мы видим, что разница в количестве событий между группами становится небольшой, но при этом **пользователи группы A (в среднем 6,9 событий)по-прежнему более активнее совершают действия на сайте, чем в группе B (в среднем 5,5 событий).**
- **Основная активность пользователей каждой группы приходится на первые дни после регистрации.** При этом активность пользователей группы A выше. **С каждым последующим днем лайфтайма мы видим, что активность пользователей снижается, и при этом результаты группы A все более сопоставимы с результатами группы B.**
- **В обеих группах все этапы воронки пользователи проходят в 0-й день лайфтайма, т.е. в день регистрации.**
- **Начиная с 14 декабря и далее мы видим резкий рост количества событий, совершенных группой A, существенно превыщающий результаты пользователей группы B. Важно, что в этот день действительно отмечался прирост новых пользователей, но он был характерен как для группы A, так и для группы B, а среди пользователей группы B таких результатов нет.** Динамика их количества событий мало верьируется на протяжении всего периода.
- **Среди маркетинговых мероприятий нет событий, которые могли бы повлиять на результаты нашего теста.** 25 декабря стартовала промо-акция, подготовленная под Рождество и Новый Год. Это событие совершенно не повлияло на результаты нашего теста, поскольку к этому моменту во-первых уже завершился набор пользователей тест, во-вторых абсолютно все пользователи фактически уже закончили участвовать в нашем тестировании, поскольку ранее мы выявили, что события каждого этапа приходятся на 0-й день лайфтайма и последнее "первое событие" нового пользователи состоялось 20 декабря.
- В общем виде, **среди 5568 участников теста до стадии покупки дошли 16% зарегистрированных пользователей и 30,8% от числа, прошедших авторизацию на сайте.**
- **Результаты конверсии группы B на всех этапах воронки хуже группы A.** В группе B до стадии покупки дошло только только 9,2% пользователей, прошедших регистрацию, и 28,3% пользователей, совершивших хотя бы одно событие (авторизацию). В то же время, среди пользователей группы A покупку совершили 21,6% и 31,7% от количества авторизовавшихся пользователей.
- **В группе A до этапа покупки дошли только 61,3% пользователей из тех, что формировали корзину. Для группы B эта цифра еще немного ниже и составляет 59,1%.** целом, за время проведения нашего теста, до покупки дошли 60,7% пользователей среди тех, кто формировал корзину.
- Конверсия из регистрации пользователя в авторизацию на сайте для пользователей, зарегистрированных в период с 7 по 10 декабря сопоставима. После этого периода мы видим **резкий рост конверсии пользователей группы A, зарегистрированных в период с 14 декабря, и при этом на протяжении всего дальшнейшего периода она остается стабильной. В группе B таких результатов нет**, конверсия пользователей, зарегистрированных после 10 декабря напротив, даже снизилась по сравнению с пользователями, зарегистрированными до этой даты. (исключение - скачок конверсии 16 декабря).

- В период с 7 по 13 декабря результаты конверсии между группами варьируются и однозначно определить лидера нельзя. Однако, далее мы видим, что **пользователи группы А, зарегистрированные после 14 декабря, демонстрируют более высокие показатели конверсии в покупку, по сравнению с результатами пользователей группы В.**
- Рассматривая показатели конверсии накопленным итогом, мы видим, что **до 14 декабря конверсия из авторизации на сайте в покупку пользователей группы В существенно лучше результатов группы А.** Хотя при этом обе группы демонстрировали снижение показателя накопленным итогом. Однако, **начиная с 13 декабря показатель конверсии группы А начали быстро восстанавливаться, и в результате к концу теста составил 31,7%, что выше, чем в группе В**, где, начиная с 12 декабря показатель конверсии стабилизировался и до окончания теста находился около 28%.

1. **Расчет статистической значимости показал:**

- **В конверсии этапа авторизация на сайте-просмотр карточки товара присутствует статистическая разница между группами А и В, а значит нулевая гипотеза отвергается. При этом коверсия группы В на 8% ниже, чем в группе А.**
- **На этапе авторизация на сайте-формирование корзины** p-value значительно больше уровня стат. значимости, а значит **статистической разницы между результатами групп нет и у нас нет оснований для отвержения нулевой гипотезы. Однако, результаты конверсии группы В на данном этапе на 2% хуже А.**
- **На этапе воронки авторизация на сайте-покупка p_value составил 0.07.** Это выше уровня статистической значимости, которая с учетом поправки Шидака составляет 0.017, а значит **тест говорит об отсутствии статистической разницы между результатами групп. Однако, стоит учитывать, что уровень 0.07 является достаточно небольшим, что может говорит о том, что некоторая разница в данных все же присутствует. При этом, конверсия группы В на 3% ниже, чем в группе А.**

1. **Результаты A/B-тестирования вполне однозначны. Можно остановить тест и зафиксировать победу группы А. Результаты группы В значительно хуже группы А на этапе авторизация-просмотр карточки товара, незначительно хуже на этапе авторизация-совершение покупки, и примерно равны на этапе авторизация-формирование корзины.**

1. **Новая рекомендательная система работает хуже старой, ее внедрение не рекомендуется.**