Проект выполнил: Фурсов Павел.
Дата: 25.07.2025

Исследование результатов A/B-теста и поиск инсайтов

Международное мобильное приложение для онлайн-торговли финансовыми активами. Пользователи могут инвестировать деньги в акции, валюту, криптовалюту, биржевые фонды (ETF) и другие активы. Целевая аудитория приложения — начинающие инвесторы. Получить доступ к финансовым рынкам можно со смартфона либо через веб-версию.

В фокусе исследования — мобильное приложение для инвестиций, работающее в нескольких латиноамериканских странах с разным уровнем экономического развития. Возникла гипотеза, что пользователи недостаточно разбираются в рисках разных инструментов, из-за чего склонны покупать высокорисковые активы и часто терпят убытки, теряя интерес к сервису.

Команда предложила углублённый вступительный онбординг, объясняющий особенности и риски различных типов активов. Существуют опасения, что детальный онбординг может отпугнуть пользователей от пополнения счёта и более рискованных инвестиций. Чтобы проверить это, был запущен A/B-эксперимент.

A/B-эксперимент

Новые пользователи, зарегистрировавшиеся в приложении с 2 по 15 июня 2025 года, были поделены поровну на две группы:

- Первая группа прошла привычный вводный онбординг без углублённых материалов.
- Вторая группа получила расширенный онбординг с детальными пояснениями по видам активов и сопутствующим рискам. Далее в течение недели отслеживалась их активность. Задача эксперимента — понять, как более информативный вводный гид влияет на дальнейшее поведение и инвестиционную активность пользователей.

При запуске обновлённого онбординга команда сформулировала несколько предположений:

- Гипотеза роста: обучающий онбординг помогает пользователям лучше понимать принципы инвестирования, поэтому они будут чаще открывать второй депозит.
- Гипотеза риска: информация о возможных потерях и высоких рисках отпугнёт некоторых новичков, особенно самых осторожных, что снизит конверсию в первый депозит.

- Дополнительная гипотеза: после нового онбординга пользователи, которые выбрали высокорискованные активы, будут чаще, чем раньше, возвращаться и открывать второй депозит. При старом онбординге пользователи часто покупали активы с высоким риском без понимания последствий. Это приводило к потерям и оттоку после первого депозита.

Для всесторонней оценки эффекта команда использовала несколько показателей:

- Ключевая метрика — средняя сумма всех депозитов на одного пользователя (включая тех, кто установил приложение или открыл веб-версию).
- Барьерная метрика — конверсия из регистрации в первый депозит.
- Вспомогательная метрика 1 — конверсия из первого депозита во второй.
- Вспомогательная метрика 2 — средняя сумма всех депозитов на пользователя, который открыл хотя бы один депозит.

Ожидалось, что ключевая и барьерная метрики не упадут, а вспомогательные покажут значительный рост.

Задачи

Необходимо провести полный анализ результатов A/B-эксперимента. Но перед этим нужно погрузиться в продукт и изучить исторические данные.

1. Анализ исторических данных

Чтобы понять, для чего была разработана новая фича, необходимо изучить исторические данные:

- Поведение новых пользователей, в том числе динамику привлечения, сегментацию и ключевые этапы воронки действий.
- Метрики, связанные с внесением депозитов, в том числе средние суммы депозитов.

2. Анализ данных A/B-теста

Изучение результатов эксперимента будет состоять из двух этапов:

- Сравнение поведения пользователей в контрольной и тестовой группах, оценка статистической значимости изменений.
- Исследование влияния нового онбординга на поведение платящих пользователей. В исследовании вы будете использовать бутстрап и сосредоточитесь на нижних и верхних перцентилях распределения депозитов.
- Такой подход позволит понять, как обновлённый онбординг повлиял на ключевые бизнес-метрики, найти точки роста и сформулировать рекомендации по улучшению пользовательского опыта и монетизации.

Данные

Будем работать с двумя датасетами:

1. Датасет `/datasets/df_hist.csv` содержит исторические данные о ключевых действиях новых пользователей, привлечённых в период с 1 апреля по 1 июня 2025 года включительно. В датасете собраны действия пользователей до оформления второго депозита.
2. Датасет `/datasets/df_abt.csv` содержит данные A/B-эксперимента — все действия новых пользователей, которые зарегистрировались со 2 по 15 июня 2025 года включительно. Данные собраны в рамках проверки гипотезы о влиянии нового онбординга на поведение и активность пользователей. Пользователи уже распределены по группам A/B-эксперимента.

Общие поля датасетов:

- `user_id` — уникальный идентификатор пользователя;
- `country_code` — код страны пользователя в формате ISO (например, `BR` — Бразилия, `MX` — Мексика, `AR` — Аргентина, `CO` — Колумбия);
- `platform` — устройство, с которого пользователь взаимодействует с продуктом: mobile или web;
- `first_ts` — время первого появления пользователя в системе;
- `first_dt` — дата первого появления пользователя (без времени);
- `event_ts` — время события;
- `event_name` — название события;
- `amount` — сумма пополнения депозита;
- `asset` — тип приобретённого актива;
- `risk_level` — уровень риска актива: low — низкий риск, medium — средний риск, high — высокий риск.

В датасете `/datasets/df_abt.csv` содержатся два дополнительных поля:

- `ab_test` — название A/B-эксперимента;
- `group` — пользовательская группа A/B-эксперимента.

План проекта

1. Загрузка исторических данных и их предобработка
2. Исследовательский анализ исторических данных
3. Исследование результатов A/B эксперимента
4. Анализ изменений суммы депозитов на платящего пользователя
5. Выводы

Часть 1

1. Загрузка исторических данных и их предобработка

Получим основную информацию о данных. Проведем предобработку.

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import Markdown
import numpy as np
from scipy.stats import ttest_ind
from statsmodels.stats.proportion import proportions_ztest
```

In [2]:
```
df = pd.read_csv('/datasets/df_hist.csv')
```

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 238059 entries, 0 to 238058
Data columns (total 10 columns):
 #   Column        Non-Null Count   Dtype  
---  ------        --------------   -----  
 0   user_id       238059 non-null  object 
 1   country_code  238059 non-null  object 
 2   platform      238059 non-null  object 
 3   first_ts      238059 non-null  object 
 4   first_dt      238059 non-null  object 
 5   event_ts      238059 non-null  object 
 6   event_name    238059 non-null  object 
 7   amount        33093 non-null   float64
 8   asset         15392 non-null   object 
 9   risk_level    15392 non-null   object 
dtypes: float64(1), object(9)
memory usage: 18.2+ MB
```

В данных есть несколько столбцов с датами и временем, которые можно привести к datetime для удобства.

Столбцы: first_ts, first_dt, event_ts

In [4]:
```
datetime_cols = ['first_ts', 'first_dt', 'event_ts']
df[datetime_cols] = df[datetime_cols].apply(pd.to_datetime)
```

In [5]:
```
df.dtypes
```

Out[5]:
```
user_id                 object
country_code            object
platform                object
first_ts        datetime64[ns]
first_dt        datetime64[ns]
event_ts        datetime64[ns]
event_name              object
amount                 float64
asset                   object
risk_level              object
dtype: object
```

Данные приведены к нужным типам.

In [6]:
```
df.head()
```

Out[6]:

	user_id	country_code	platform	first_ts	first_dt	event_ts	event_nam
0	548ac59f-656d-4110-80d2-49f0a217f08a	BR	mobile	2025-04-02 19:55:51	2025-04-02	2025-04-02 19:55:51	install / open_we
1	548ac59f-656d-4110-80d2-49f0a217f08a	BR	mobile	2025-04-02 19:55:51	2025-04-02	2025-04-02 19:55:59	introductic
2	548ac59f-656d-4110-80d2-49f0a217f08a	BR	mobile	2025-04-02 19:55:51	2025-04-02	2025-04-02 23:46:06	registratic
3	548ac59f-656d-4110-80d2-49f0a217f08a	BR	mobile	2025-04-02 19:55:51	2025-04-02	2025-04-02 23:46:15	main_pag
4	548ac59f-656d-4110-80d2-49f0a217f08a	BR	mobile	2025-04-02 19:55:51	2025-04-02	2025-04-02 23:47:59	onboarding_complet

In [7]:
```
df.isna().sum()
```

Out[7]:
```
user_id              0
country_code         0
platform             0
first_ts             0
first_dt             0
event_ts             0
event_name           0
amount          204966
asset           222667
risk_level      222667
dtype: int64
```

Данные выглядят нормально, количество пропусков обусловлено тем, что не для каждого действия (эвента) предусмотрена сумма, покупка актива или уровень риска. Эти столбцы заполняются только для эвентов депозита и покупки актива.

In [8]:
```
df.duplicated().sum()
```

Out[8]:
```
0
```

Явных дубликатов нет.

In [9]:
```
df['event_name'].value_counts()
```

Out[9]:
```
introduction           41032
install / open_web     41032
registration           38133
main_page              35040
onboarding_complete    34337
first_deposit          27685
asset_purchase         15392
second_deposit          5408
Name: event_name, dtype: int64
```

In [78]:
```
df[df['event_ts'] < df['first_ts']]
```

Out[78]:

user_id	country_code	platform	first_ts	first_dt	event_ts	event_name	amount	ass

С датами всё корректно, у пользователей нет событий произошедших до их первого появления в приложении.

In [10]:
```
df['platform'].unique()
```

Out[10]:
```
array(['mobile', 'web'], dtype=object)
```

In [11]:
```
df['country_code'].value_counts()
```

Out[11]:
```
BR    71120
MX    65493
CO    57745
AR    43701
Name: country_code, dtype: int64
```

In [12]:
```
df['risk_level'].unique()
```

Out[12]:
```
array([nan, 'medium', 'high', 'low'], dtype=object)
```

In [75]:
```
df['first_ts'].min()
```

Out[75]:
```
Timestamp('2025-04-01 00:00:00')
```

In [77]:
```
df['first_ts'].max()
```

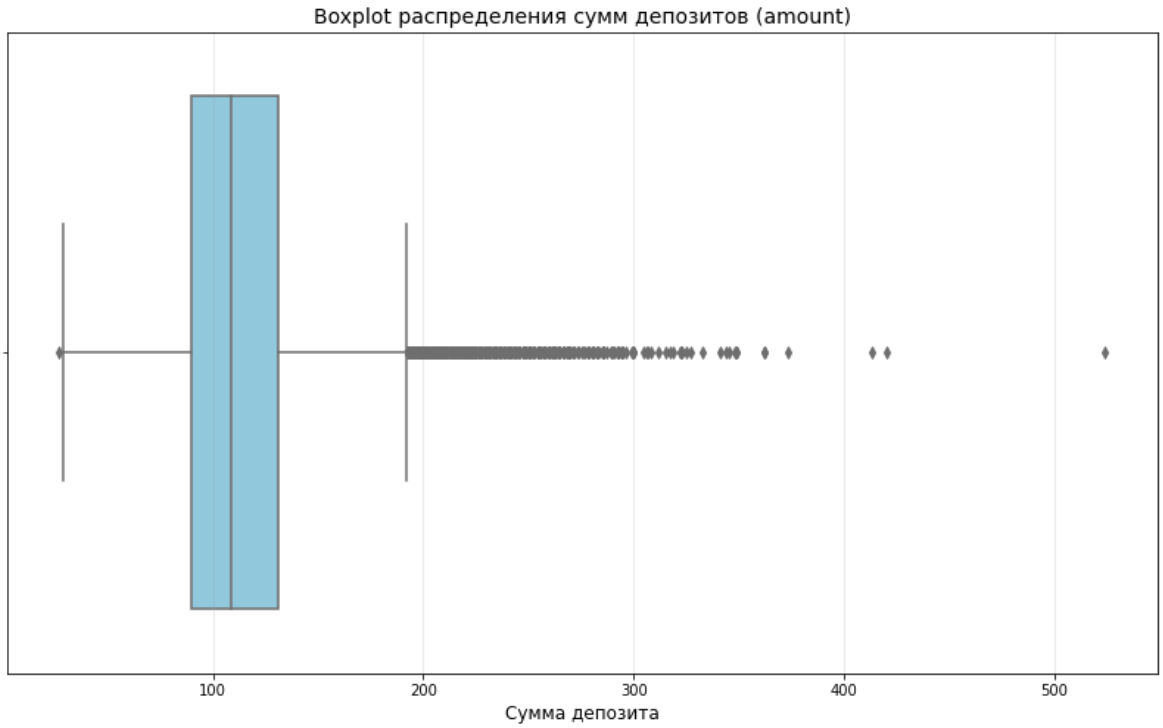
Out[77]:
```
Timestamp('2025-06-01 23:59:52')
```

Данные корректно распределены с 1 апреля по 1 июня включительно.

In [15]:
```
df['amount'].describe()
```

Out[15]:
```
count    33093.000000
mean       113.527967
std         34.472458
min         27.000000
25%         90.000000
50%        109.000000
75%        131.000000
max        524.000000
Name: amount, dtype: float64
```

In [16]:
```
plt.figure(figsize=(14, 8))
sns.boxplot(x=df['amount'], color="skyblue")

plt.title('Boxplot распределения сумм депозитов (amount)', fontsize=14)
plt.xlabel('Сумма депозита', fontsize=12)
plt.grid(axis='x', alpha=0.3)
plt.show()
```

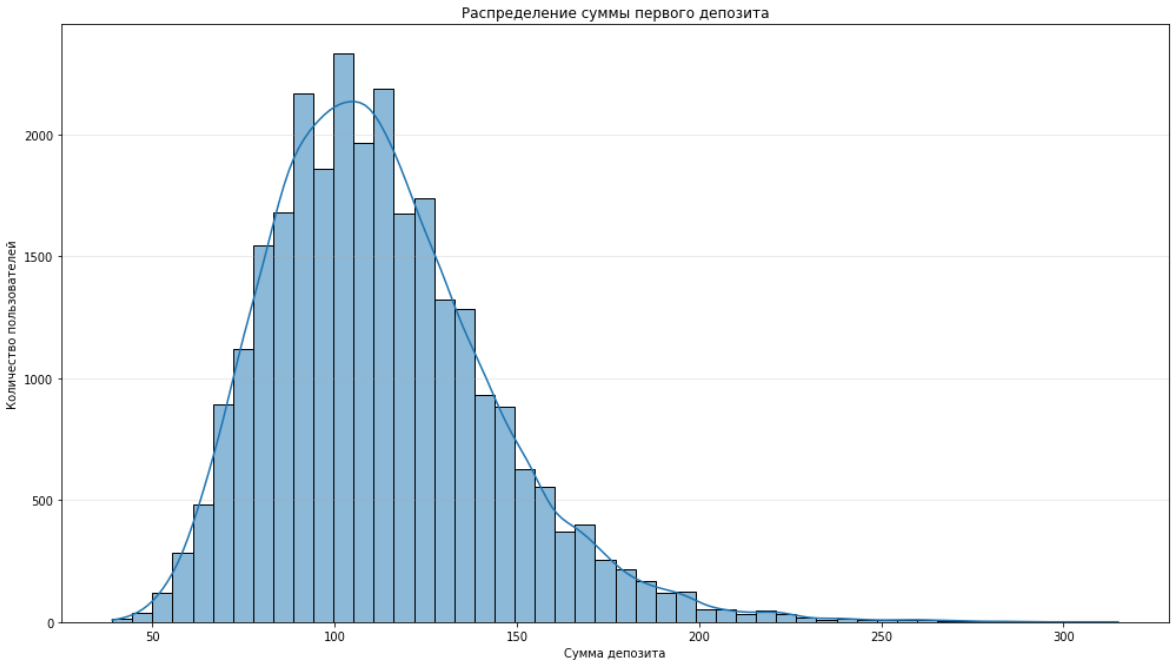


На боксплоте заметны выбросы с высокими значениями до 524 - это нормально для финансовых данных.

In [80]:
```
deposits = df[df['event_name'] == 'first_deposit'].copy()
deposits['amount'].describe()
```

Out[80]:
```
count    27685.000000
mean       112.543399
std         30.764811
min         39.000000
25%         91.000000
50%        109.000000
75%        130.000000
max        315.000000
Name: amount, dtype: float64
```

In [88]:
```
plt.figure(figsize=(14, 8))
sns.histplot(deposits['amount'], bins=50, kde=True)
plt.title('Распределение суммы первого депозита')
plt.xlabel('Сумма депозита')
plt.ylabel("Количество пользователей")
plt.grid(axis='y', alpha=0.3)
plt.tight_layout()
plt.show()
```



Сумма первого депозита распределена почти нормально, небольшой перекос в сторону правого хвоста. Для данных о финансах это логичное распределение, сильных выбросов нет.

Промежуточный вывод

В датафрейм загружено 238059 записей.
Все необходимые столбцы приведены к нужным типам данных.
Пропуски есть только в полях, которые логично заполняются не для всех событий (amount, asset, risk_level), ошибок не выявлено.
Явных дубликатов нет.
Данные (регистрации) распределены с 1 апреля по 1 июня включительно.

Основные события/Путь пользователя:

- Установка приложения / Открытие сайта
- Введение / знакомство
- Регистрация
- Главная страница
- Завершение онбординга
- Первый депозит
- Покупка актива
- Второй депозит

Страны:

- Бразилия
- Мексика
- Колумбия
- Аргентина

Платформы:

- Mobile
- Web

Уровни риска активов:

- low
- medium
- high

Данные по депозитам (amount):

- Общее число транзакций: 33093
- Средняя сумма депозита: 113.5
- Медиана: 109
- Диапазон: от 27 до 524
- Аномально высоких значений нет.

---

2. Исследовательский анализ исторических данных

Анализ новых пользователей.

- Изучим динамику привлечения новых пользователей в приложение.

In [90]:
```
new_users = df.groupby('first_dt')['user_id'].nunique().reset_index(name='users_
new_users['7day_rolling_avg'] = new_users['users_count'].rolling(7).mean()
new_users.head()
```

Out[90]:

	first_dt	users_count	7day_rolling_avg
0	2025-04-01	670	NaN
1	2025-04-02	676	NaN
2	2025-04-03	691	NaN
3	2025-04-04	637	NaN
4	2025-04-05	640	NaN

In [92]:
```
plt.figure(figsize=(14, 8))

sns.lineplot(x='first_dt', y='users_count', data=new_users, label='Ежедневно')
sns.lineplot(x='first_dt', y='7day_rolling_avg', data=new_users, label='7-дневно

plt.xlabel('Дата')
plt.ylabel('Количество новых пользователей')
plt.title('Динамика привлечения новых пользователей')
plt.xticks(rotation=45)
plt.grid(axis='y', alpha=0.3)
plt.tight_layout()
plt.legend()
plt.show()
```

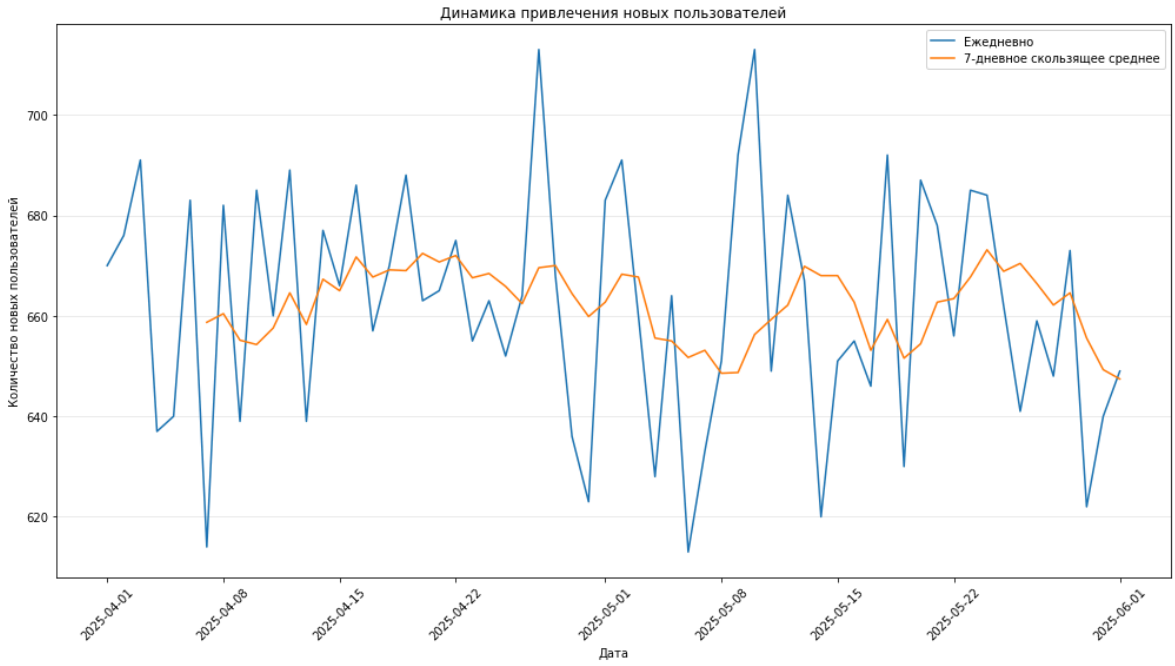
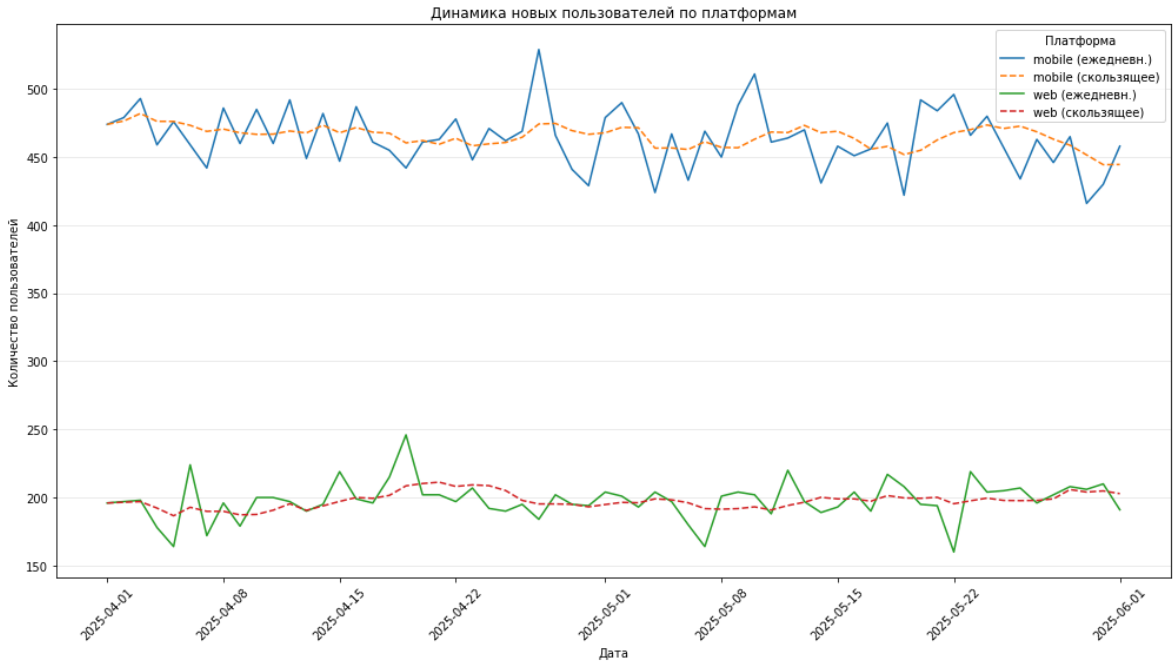


График относительно стабильный, показывает колебания новых пользователей в диапазоне от 610 до 710. Высокая волатильность обусловлена тем, что график для удобства читаемости построен не от 0.

In [128…

```
platform_users = df.groupby(['first_dt', 'platform'])['user_id'].nunique().reset

platform_users['7day_rolling_avg'] = (
    platform_users.groupby('platform')['users_count']
    .transform(lambda x: x.rolling(window=7, min_periods=1).mean())
)

plt.figure(figsize=(14, 8))

for platform in platform_users['platform'].unique():
    data_subset = platform_users[platform_users['platform'] == platform]

    sns.lineplot(
        data=data_subset,
        x='first_dt',
        y='users_count',
        label=f'{platform} (ежедневн.)'
    )

    sns.lineplot(
        data=data_subset,
        x='first_dt',
        y='7day_rolling_avg',
        label=f'{platform} (скользящее)',
        linestyle='--'
    )

plt.title('Динамика новых пользователей по платформам')
plt.xlabel('Дата')
plt.ylabel('Количество пользователей')
plt.xticks(rotation=45)
plt.grid(axis='y', alpha=0.3)
plt.tight_layout()
```

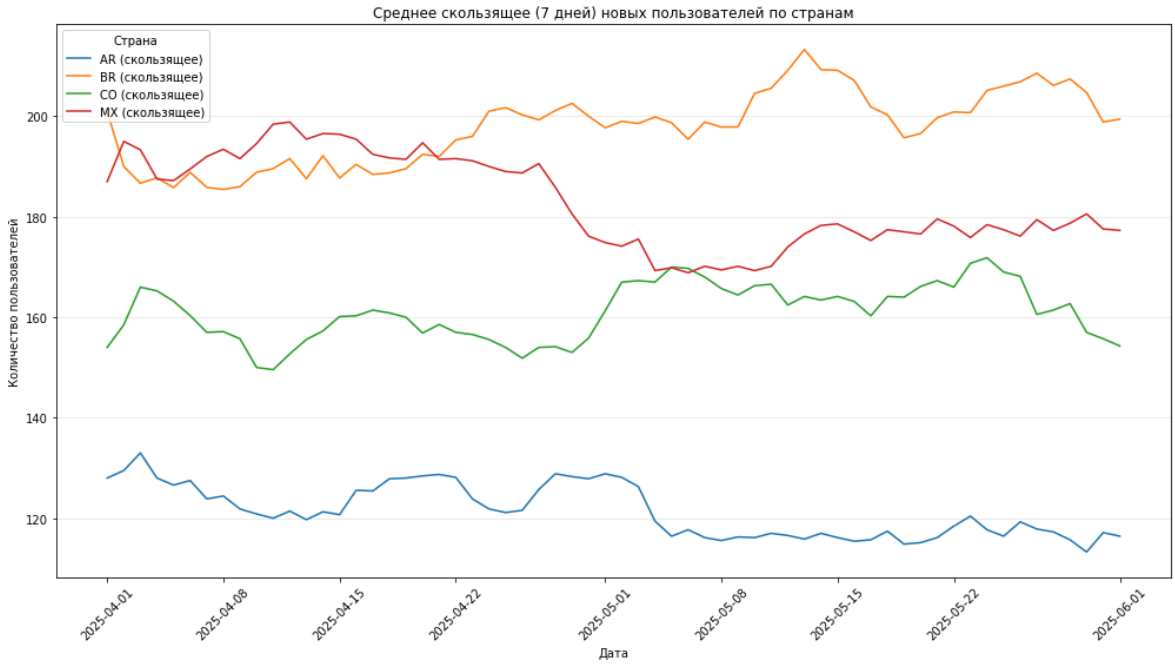
```
plt.legend(title='Платформа')
plt.show()
```



По платформам видим явное преимущество мобильного приложения - в среднем 470 пользователей над веб версией - 200 пользователей.

In [124…

```
country_users = df.groupby(['first_dt', 'country_code'])['user_id'].nunique().re

country_users['7day_rolling_avg'] = (
    country_users.groupby('country_code')['users_count']
    .transform(lambda x: x.rolling(window=7, min_periods=1).mean())
)

plt.figure(figsize=(14, 8))

for country in country_users['country_code'].unique():
    data_subset = country_users[country_users['country_code'] == country]

    sns.lineplot(
        data=data_subset,
        x='first_dt',
        y='7day_rolling_avg',
        label=f'{country} (скользящее)'
    )

plt.title('Среднее скользящее (7 дней) новых пользователей по странам')
plt.xlabel('Дата')
plt.ylabel('Количество пользователей')
plt.xticks(rotation=45)
plt.grid(axis='y', alpha=0.3)
plt.tight_layout()
plt.legend(title='Страна')
plt.show()
```



По странам разница чуть меньше, самая отстающая по пользователям страна - Аргентина, около 120 пользователей в среднем.

Бразилия - 200, Колумбия - 160, Мексика - 180.

Анализ воронок событий.

In [19]:
```
funnel_steps = df['event_name'].unique()
funnel_steps
```

Out[19]:
```
array(['install / open_web', 'introduction', 'registration', 'main_page',
       'onboarding_complete', 'first_deposit', 'asset_purchase',
       'second_deposit'], dtype=object)
```

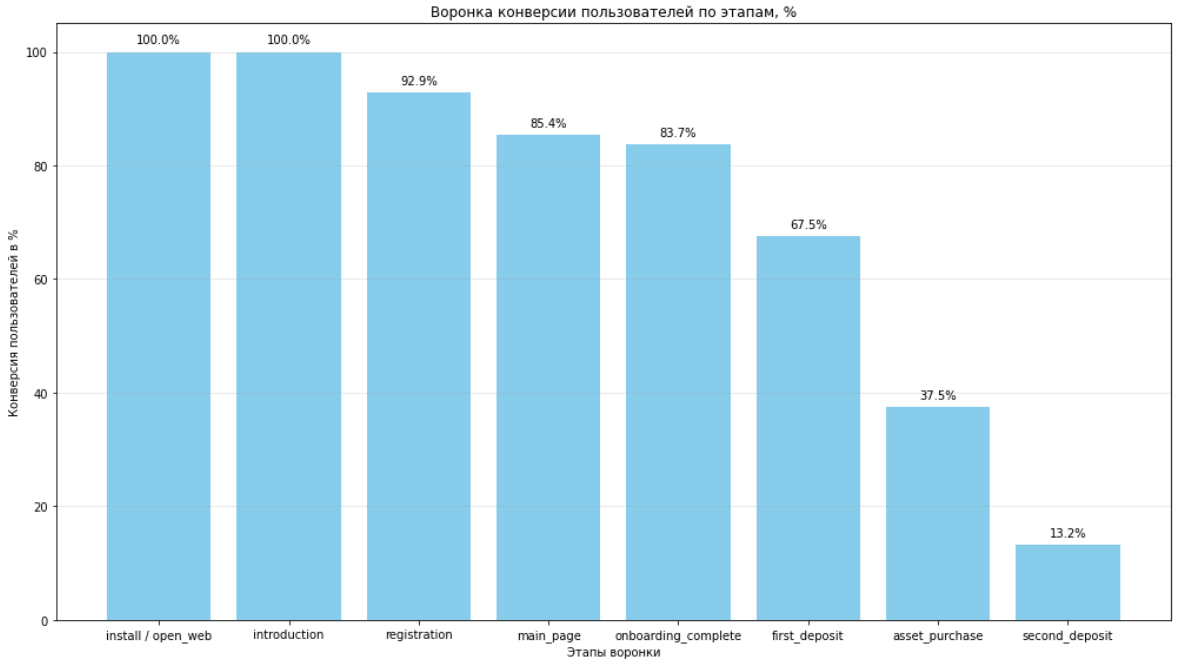
In [140…
```
funnel_df = df.groupby('event_name')['user_id'].nunique().reindex(funnel_steps).
funnel_df['conversion_from_first_%'] = (funnel_df['users_count'] / funnel_df['us
funnel_df['conversion_from_previous_%'] = (funnel_df['users_count'] / funnel_df[

funnel_df
```

Out[140…

	event_name	users_count	conversion_from_first_%	conversion_from_previous_
0	install / open_web	41032	100.0	Na
1	introduction	41032	100.0	100
2	registration	38133	92.9	92
3	main_page	35040	85.4	91
4	onboarding_complete	34337	83.7	98
5	first_deposit	27685	67.5	80
6	asset_purchase	15392	37.5	55
7	second_deposit	5408	13.2	35

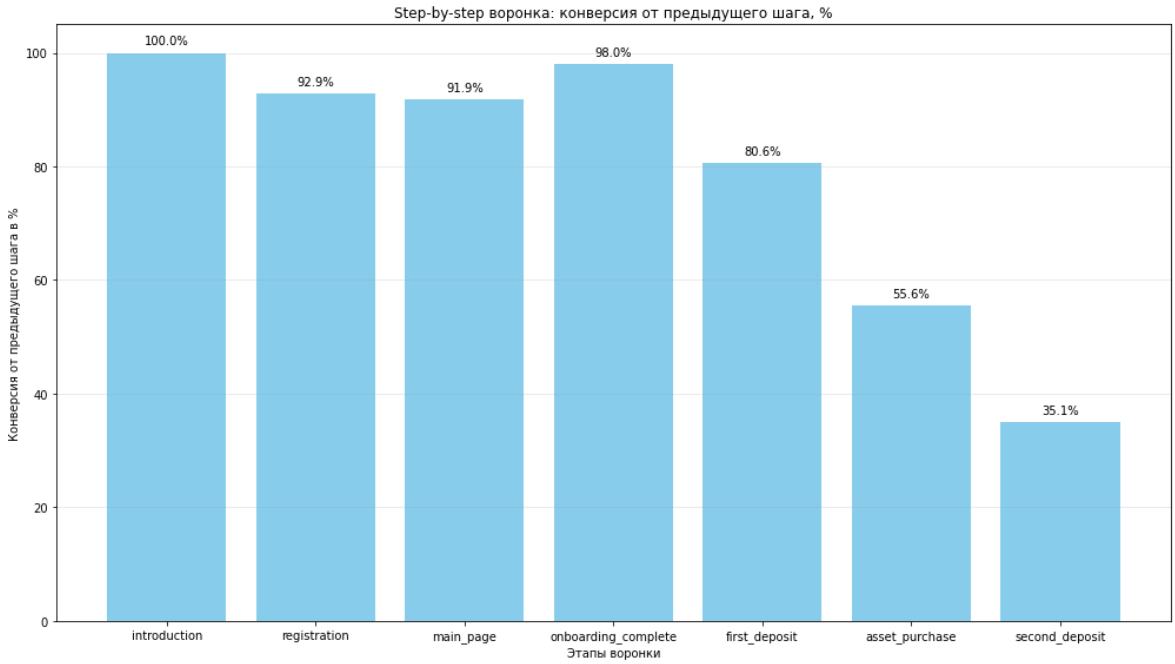
In [135…

```
plt.figure(figsize=(14,8))

bars = plt.bar(funnel_df['event_name'], funnel_df['conversion_from_first_%'], co

plt.xlabel('Этапы воронки')
plt.ylabel('Конверсия пользователей в %')
plt.title('Воронка конверсии пользователей по этапам, %')
plt.grid(axis='y', alpha=0.3)

for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 1, f'{height:.1f}%',
             ha='center', va='bottom', fontsize=10)

plt.tight_layout()
plt.show()
```



In [144…

```
plt.figure(figsize=(14,8))

bars = plt.bar(funnel_df['event_name'], funnel_df['conversion_from_previous_%'],

plt.xlabel('Этапы воронки')
plt.ylabel('Конверсия от предыдущего шага в %')
plt.title('Step-by-step воронка: конверсия от предыдущего шага, %')
plt.grid(axis='y', alpha=0.3)

for bar in bars:
    height = bar.get_height()
    if np.isfinite(height):
        plt.text(bar.get_x() + bar.get_width()/2, height + 1, f'{height:.1f}%',
                 ha='center', va='bottom', fontsize=10)

plt.tight_layout()
plt.show()
```



Проанализировав воронки, можно сделать вывод, что большинство пользователей теряются на следующих этапах:

- после первого депозита только 55.6% совершают покупку актива
- после покупки актива лишь 35.1% возвращаются за вторым депозитом

Это подтверждает гипотезу о потере вовлечённости после первого взаимодействия с активами.
Узкие места - после первого депозита и покупки активов.
Именно здесь новая обучающая фича может повлиять на поведение пользователей.

Влияние уровня риска актива на открытие второго депозита.

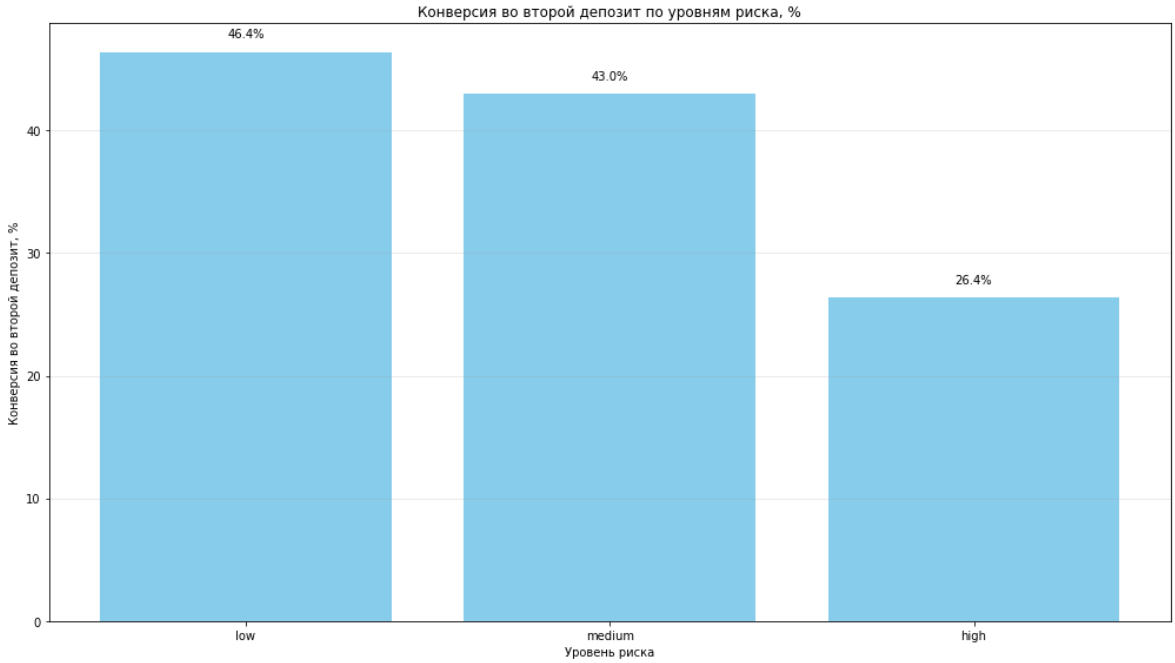
In [23]:
```
assets = df[df['event_name'] == 'asset_purchase']

first_asset = assets.sort_values(['user_id', 'event_ts']).drop_duplicates('user_
risk_conversion_df = first_asset[['user_id', 'risk_level']].copy()

second_deposit_users = df[df['event_name'] == 'second_deposit']['user_id'].uniqu
risk_conversion_df['second_deposit_made'] = risk_conversion_df['user_id'].isin(s

conversion_stats = (
    risk_conversion_df.groupby('risk_level')['second_deposit_made']
    .agg(['count', 'sum'])
    .rename(columns={'count': 'users_with_asset', 'sum': 'users_with_second_depo
).reindex(['low', 'medium', 'high']).reset_index()

conversion_stats['conversion_rate'] = 100 * conversion_stats['users_with_second_

conversion_stats
```

Out[23]:

	risk_level	users_with_asset	users_with_second_deposit	conversion_rate
0	low	2327	1080	46.411689
1	medium	5325	2288	42.967136
2	high	7740	2040	26.356589

In [145…

```
plt.figure(figsize=(14,8))

bars = plt.bar(conversion_stats['risk_level'], conversion_stats['conversion_rate

plt.xlabel('Уровень риска')
plt.ylabel('Конверсия во второй депозит, %')
plt.title('Конверсия во второй депозит по уровням риска, %')
plt.grid(axis='y', alpha=0.3)

for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 1, f'{height:.1f}%',
             ha='center', va='bottom', fontsize=10)

plt.tight_layout()
plt.show()
```



Конверсия во второй депозит существенно зависит от уровня риска первого актива:

- При низком риске: 46.4% пользователей совершают второй депозит.
- При среднем риске: 43%
- При высоком риске лишь 26.4%

Это подтверждает гипотезу: неопытные пользователи, выбравшие активы с высоким риском, чаще разочаровываются и не возвращаются.
Следовательно, обучающий онбординг имеет потенциал снизить отток за счёт повышения осознанности при выборе актива.

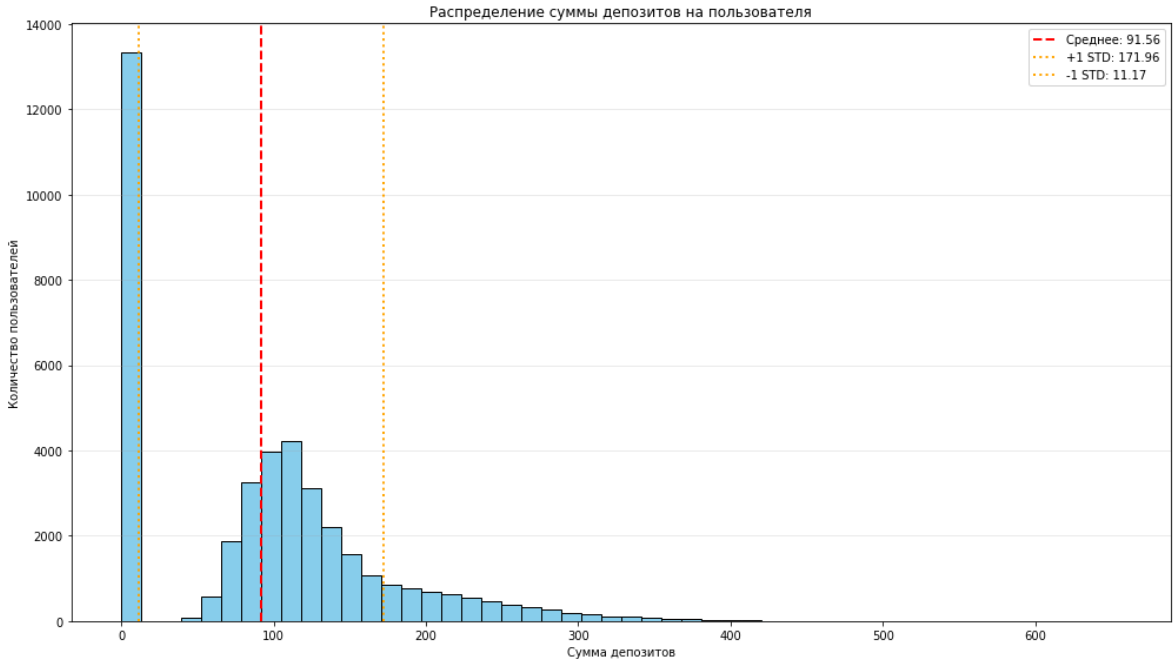
Анализ ключевой метрики на исторических данных.

In [147…
```
all_users = df['user_id'].drop_duplicates().reset_index()

user_deposits = df.groupby('user_id')['amount'].sum().reset_index()
user_deposits = all_users.merge(user_deposits, on='user_id', how='left')
user_deposits['amount'] = user_deposits['amount'].fillna(0)

key_metric = user_deposits['amount'].mean()
std_metric = user_deposits['amount'].std()

display(Markdown(f"**Средняя сумма депозитов на одного пользователя:** {key_metr
display(Markdown(f"**Стандартное отклонение суммы депозитов на одного пользовате
```

Средняя сумма депозитов на одного пользователя: 91.56

Стандартное отклонение суммы депозитов на одного пользователя: 80.39

In [148…
```
plt.figure(figsize=(14, 8))

plt.hist(user_deposits['amount'], bins=50, color='skyblue', edgecolor='black')

plt.axvline(key_metric, color='red', linestyle='dashed', linewidth=2, label=f'Cp
plt.axvline(key_metric + std_metric, color='orange', linestyle='dotted', linewid
plt.axvline(key_metric - std_metric, color='orange', linestyle='dotted', linewid

plt.title('Распределение суммы депозитов на пользователя')
plt.xlabel('Сумма депозитов')
plt.ylabel('Количество пользователей')
plt.legend()
plt.grid(axis='y', alpha=0.3)
plt.tight_layout()
plt.show()
```



Средняя сумма всех депозитов на одного пользователя составляет 91.56

Стандартное отклонение 80.39

Такое большое стандартное отклонение указывает на высокую нестабильность

среднего, поэтому полагаться только на среднее при анализе нельзя, необходимо проанализировать медиану, перцентили и бутстрап интервалы.

При этом значительная доля пользователей не делает ни одного депозита (больше 13000 пользователей).
Большинство пользователей делают депозиты в диапазоне от 11 до 172.
Длинный правый хвост распределения объясняется наличием небольшого числа пользователей с крупными суммами.

Часть 2

3. Исследование результатов A/B эксперимента

Загрузим данные с результатами эксперимента. Проверим корректность данных и проведем предобработку.
Затем проведем анализ результатов A/B-эксперимента.

In [27]:
```
ab_df = pd.read_csv('/datasets/df_abt.csv')
```

In [28]:
```
ab_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54805 entries, 0 to 54804
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   user_id       54805 non-null  object 
 1   country_code  54805 non-null  object 
 2   platform      54805 non-null  object 
 3   first_ts      54805 non-null  object 
 4   first_dt      54805 non-null  object 
 5   event_ts      54805 non-null  object 
 6   event_name    54805 non-null  object 
 7   ab_test       54805 non-null  object 
 8   group         54805 non-null  object 
 9   amount        7843 non-null   float64
 10  asset         3750 non-null   object 
 11  risk_level    3750 non-null   object 
dtypes: float64(1), object(11)
memory usage: 5.0+ MB
```

In [29]:
```
ab_df[datetime_cols] = ab_df[datetime_cols].apply(pd.to_datetime)
```

В датафрейме та же проблема, что и в основном - даты не приведены к нужным типам (в тех же столбцах), поэтому используем тот же список и приводим к datetime.

In [30]:
```
ab_df.dtypes
```

Out[30]:
```
user_id                object
country_code           object
platform               object
first_ts        datetime64[ns]
first_dt        datetime64[ns]
event_ts        datetime64[ns]
event_name             object
ab_test                object
group                  object
amount                float64
asset                  object
risk_level             object
dtype: object
```

In [31]:
```
ab_df.head()
```

Out[31]:

	user_id	country_code	platform	first_ts	first_dt	event_ts	event_nar
0	c430acb5-c6bf-43cf-8e2d-7ab9ce0d2c8a	BR	mobile	2025-06-07 19:55:51	2025-06-07	2025-06-07 19:55:51	install / open_w
1	c430acb5-c6bf-43cf-8e2d-7ab9ce0d2c8a	BR	mobile	2025-06-07 19:55:51	2025-06-07	2025-06-07 19:55:54	introducti
2	c430acb5-c6bf-43cf-8e2d-7ab9ce0d2c8a	BR	mobile	2025-06-07 19:55:51	2025-06-07	2025-06-08 06:18:09	registrati
3	c430acb5-c6bf-43cf-8e2d-7ab9ce0d2c8a	BR	mobile	2025-06-07 19:55:51	2025-06-07	2025-06-08 06:18:13	main_pa
4	c430acb5-c6bf-43cf-8e2d-7ab9ce0d2c8a	BR	mobile	2025-06-07 19:55:51	2025-06-07	2025-06-08 06:33:40	onboarding_comple

In [32]:
```
ab_df.isna().sum()
```

Out[32]:
```
user_id             0
country_code        0
platform            0
first_ts            0
first_dt            0
event_ts            0
event_name          0
ab_test             0
group               0
amount          46962
asset           51055
risk_level      51055
dtype: int64
```

Так же как и в основном датафрейме, отметим, что пропуски в стобцах amoun, asset, risk_level - это нормально.

In [33]:
```
ab_df.duplicated().sum()
```

Out[33]:
```
0
```

Явных дубликатов нет.

In [34]:
```
ab_df['ab_test'].unique()
```

Out[34]:
```
array(['onboarding_test'], dtype=object)
```

В данных только один тест.

In [35]:
```
ab_df['group'].unique()
```

Out[35]:
```
array(['test', 'control'], dtype=object)
```

Пользователи разбиты на две группы - control и test.

In [36]:
```
user_groups = ab_df.groupby('user_id')['group'].nunique().reset_index(name='grou
users_in_both = user_groups[user_groups['group_count'] > 1]

display(Markdown(f"**Пользователей, попавших в обе группы:** {len(users_in_both)
```

Пользователей, попавших в обе группы: 0

In [37]:
```
ab_df[['user_id', 'group']].drop_duplicates().groupby('group')['user_id'].count(
```

Out[37]:
```
group
control    4847
test       4568
Name: user_id, dtype: int64
```

Распределение пользователей выглядит корректно, чуть ниже проверим распределение по платформам и странам.

In [38]:
```
ab_df[~ab_df['amount'].isna()].groupby('group')['user_id'].nunique()
```

Out[38]:
```
group
control    3228
test       2987
Name: user_id, dtype: int64
```

Количество платящих пользователей тоже выглядит корректно.

In [39]:
```
ab_df.groupby('group')['first_ts'].agg(['min', 'max'])
```

Out[39]:

group	min	max
control	2025-06-02 00:27:50	2025-06-15 23:47:04
test	2025-06-02 01:08:14	2025-06-15 23:58:39

Даты в обеих группах соответствуют датам начала и конца A/B теста.

Промежуточный вывод

Данные успешно загружены: 54805 строк, структура и предобработка аналогичны основной таблице событий.
Типы datetime приведены, дубликатов нет, пропуски в столбцах amount, asset, risk_level допустимы и обусловлены логикой событий.

Проверка валидности A/B-теста показала:

- В эксперименте участвует один тест - onboarding_test.
- Всего две группы: control и test.
- Пользователи корректно разделены - пересечений между группами нет.
- Распределение по пользователям: control - 4847, test - 4568
- Платящие пользователи: control - 3228, test - 2987.
- Диапазоны дат в обеих группах соответствуют эксперименту: с 2 по 15 июня 2025 года.

Перед подведением результатов A/B теста также сформулируем основные гипотезы:

- **Нулевая гипотеза**: обновленный онбординг не изменит среднюю сумма всех депозитов на одного пользователя.
- **Альтернативная гипотеза**: обновленный онбординг повысит среднюю сумма всех депозитов на одного пользователя.

Анализ аудитории эксперимента.

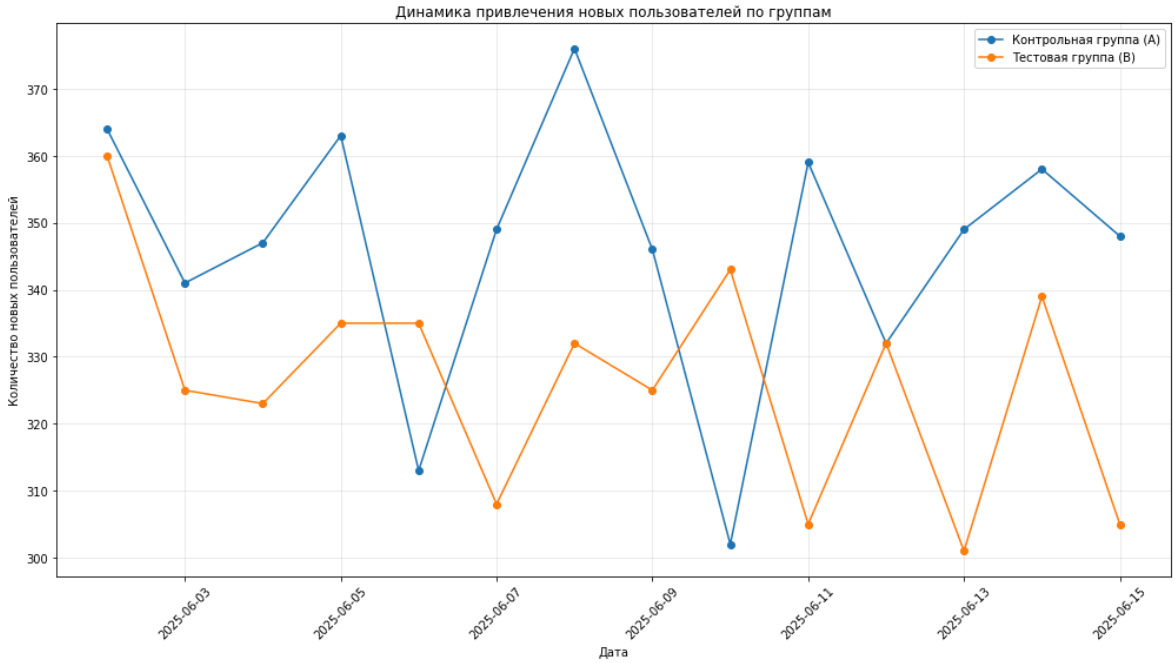
- Проверим корректность распределения новых пользователей по группам A/B-эксперимента.

In [40]:
```
ab_new_users = ab_df.groupby(['group','first_dt'])['user_id'].nunique().reset_in
ab_new_users.head()
```

Out[40]:

	group	first_dt	users_count
0	control	2025-06-02	364
1	control	2025-06-03	341
2	control	2025-06-04	347
3	control	2025-06-05	363
4	control	2025-06-06	313

In [41]:
```
control_new_users = ab_new_users[ab_new_users['group'] == 'control']
test_new_users = ab_new_users[ab_new_users['group'] == 'test']

plt.figure(figsize=(14, 8))
plt.plot(control_new_users['first_dt'], control_new_users['users_count'], label=
plt.plot(test_new_users['first_dt'], test_new_users['users_count'], label='Тесто

plt.title('Динамика привлечения новых пользователей по группам')
plt.xlabel('Дата')
plt.ylabel('Количество новых пользователей')
plt.xticks(rotation=45)
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()
```



Учитывая, что график строится не от 0 (для удобства просмотра), количества распределены корректно - от 300 до 370 пользователей ежедневно.

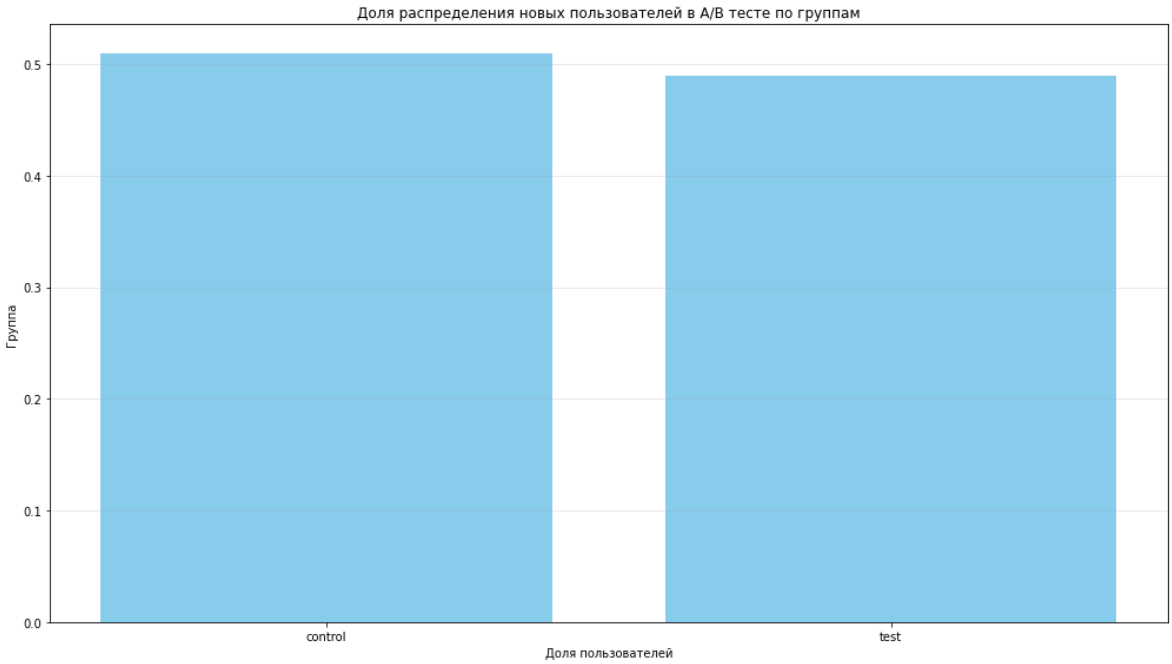
In [42]:
```
ab_new_users_sum = ab_new_users.groupby('group')['users_count'].sum().reset_inde
ab_new_users_sum['share'] = (ab_new_users_sum['users_count'] / ab_new_users_sum[
ab_new_users_sum
```

Out[42]:

	group	users_count	share
0	control	4847	0.51
1	test	4568	0.49

In [160…
```
plt.figure(figsize=(14,8))

plt.bar(ab_new_users_sum['group'], ab_new_users_sum['share'], color='skyblue')

plt.xlabel('Доля пользователей')
plt.ylabel('Группа')
plt.title('Доля распределения новых пользователей в A/B тесте по группам')
plt.grid(axis='y', alpha=0.3)
```

```
plt.tight_layout()
plt.show()
```



Распределение выглядит нормальным, разница в 2% - незначительна, проверим разницу по странам и платформам.

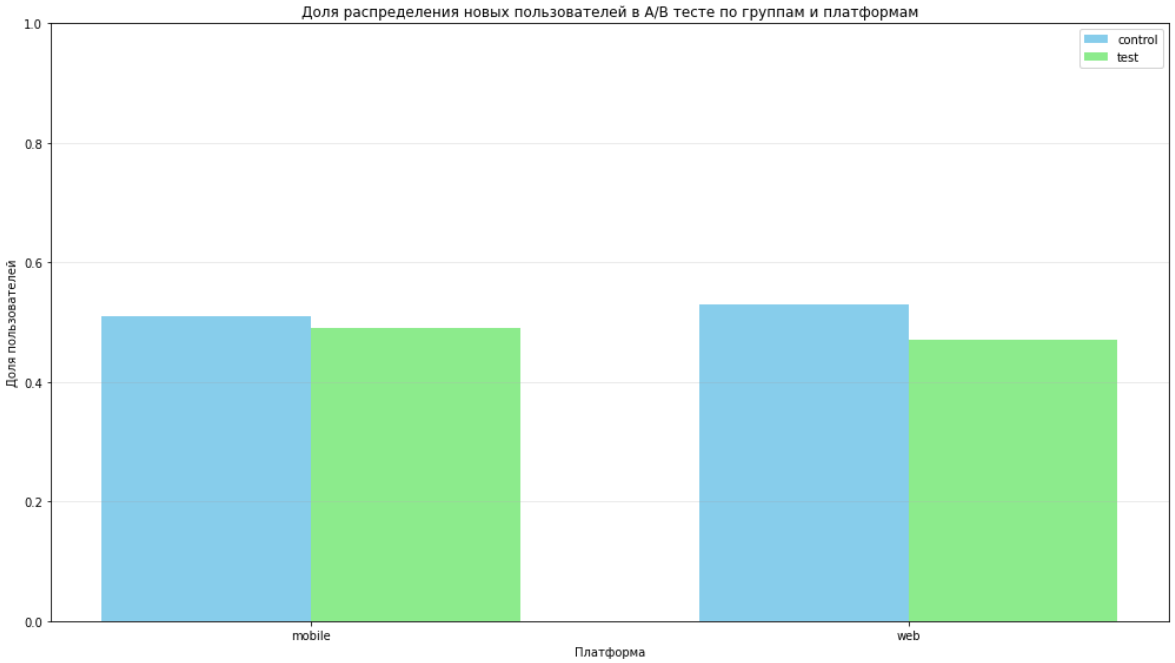
In [44]:
```
platform_dist = ab_df.groupby(['group', 'platform'])['user_id'].nunique().reset_

platform_sums = platform_dist.groupby('platform')['users_count'].sum().reset_ind
platform_dist = platform_dist.merge(platform_sums, on='platform')
platform_dist['share'] = (platform_dist['users_count'] / platform_dist['total_us

platform_dist.sort_values(['platform', 'group'])
```

Out[44]:

	group	platform	users_count	total_users	share
0	control	mobile	3385	6631	0.51
1	test	mobile	3246	6631	0.49
2	control	web	1462	2784	0.53
3	test	web	1322	2784	0.47

In [45]:
```
platforms = platform_dist['platform'].unique()
groups = platform_dist['group'].unique()

x = np.arange(len(platforms))
width = 0.35

control = platform_dist[platform_dist['group'] == 'control']['share'].values
test = platform_dist[platform_dist['group'] == 'test']['share'].values

plt.figure(figsize=(14, 8))
plt.bar(x - width/2, control, width, label='control', color='skyblue')
plt.bar(x + width/2, test, width, label='test', color='lightgreen')

plt.xlabel('Платформа')
plt.ylabel('Доля пользователей')
```

```
plt.title('Доля распределения новых пользователей в A/B тесте по группам и платф
plt.xticks(ticks=x, labels=platforms)
plt.ylim(0, 1)
plt.grid(axis='y', alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()
```



Видим разницу в 6% среди пользователей веб версии, что не является критичным.

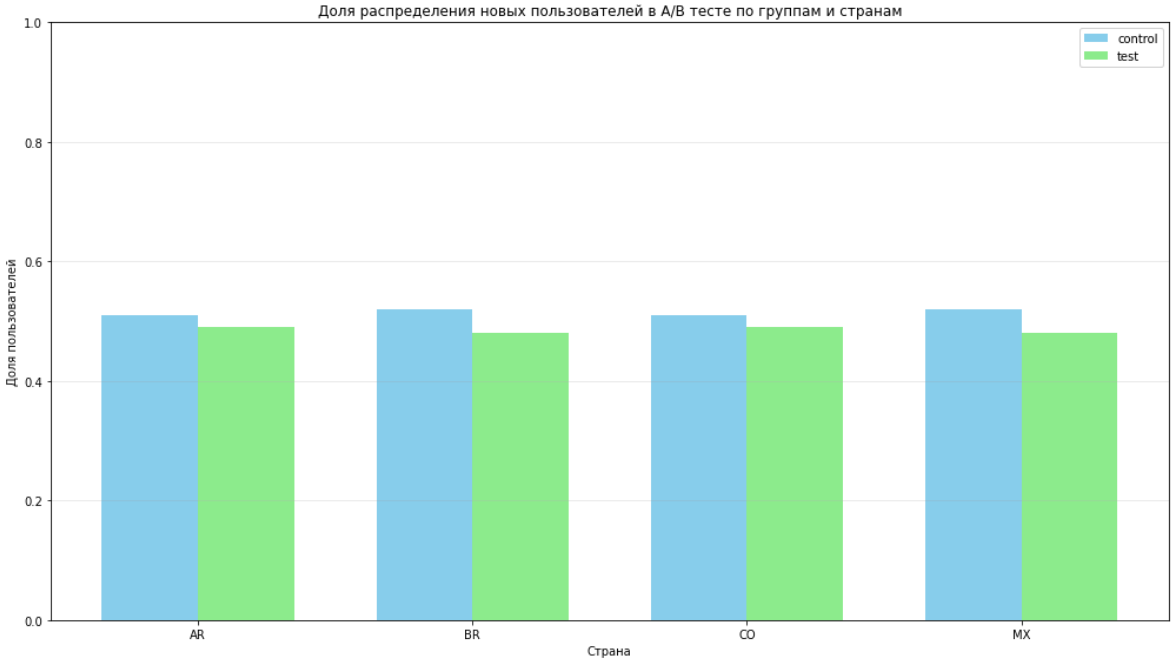
In [46]:
```
country_dist = ab_df.groupby(['group', 'country_code'])['user_id'].nunique().res

country_sums = country_dist.groupby('country_code')['users_count'].sum().reset_i
country_dist = country_dist.merge(country_sums, on='country_code')
country_dist['share'] = (country_dist['users_count'] / country_dist['total_users

country_dist.sort_values(['country_code', 'group'])
```

Out[46]:

	group	country_code	users_count	total_users	share
0	control	AR	756	1481	0.51
1	test	AR	725	1481	0.49
2	control	BR	1543	2991	0.52
3	test	BR	1448	2991	0.48
4	control	CO	1247	2446	0.51
5	test	CO	1199	2446	0.49
6	control	MX	1301	2497	0.52
7	test	MX	1196	2497	0.48

In [47]:
```
countries = country_dist['country_code'].unique()
groups = country_dist['group'].unique()

x = np.arange(len(countries))
```

```
width = 0.35

control = country_dist[country_dist['group'] == 'control']['share'].values
test = country_dist[country_dist['group'] == 'test']['share'].values

plt.figure(figsize=(14, 8))
plt.bar(x - width/2, control, width, label='control', color='skyblue')
plt.bar(x + width/2, test, width, label='test', color='lightgreen')

plt.xlabel('Страна')
plt.ylabel('Доля пользователей')
plt.title('Доля распределения новых пользователей в A/B тесте по группам и стран
plt.xticks(ticks=x, labels=countries)
plt.ylim(0, 1)
plt.grid(axis='y', alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()
```



Распределение по странам показывает небольшое расхождение по всем странам, но как и ранее - не критичное.

Распределение пользователей по группам сбалансировано во всех разрезах (по дате, платформе, стране). Значимых перекосов нет - можно переходить к анализу эффекта от теста.

Сравнение воронок событий.

In [48]:
```
control = ab_df[ab_df['group']=='control']
test = ab_df[ab_df['group']=='test']

control_funnel_df = control.groupby('event_name')['user_id'].nunique().reindex(f
test_funnel_df = test.groupby('event_name')['user_id'].nunique().reindex(funnel_

control_funnel_df['conversion_from_first_%'] = (control_funnel_df['users_count']
test_funnel_df['conversion_from_first_%'] = (test_funnel_df['users_count'] / tes
```

```
control_funnel_df['conversion_from_previous_%'] = (control_funnel_df['users_coun
test_funnel_df['conversion_from_previous_%'] = (test_funnel_df['users_count'] /
```

In [49]:
```
control_funnel_df
```

Out[49]:

	event_name	users_count	conversion_from_first_%	conversion_from_previous_
0	install / open_web	4847	100.0	Na
1	introduction	4847	100.0	100
2	registration	4512	93.1	93
3	main_page	4161	85.8	92
4	onboarding_complete	4063	83.8	97
5	first_deposit	3228	66.6	79
6	asset_purchase	1773	36.6	54
7	second_deposit	654	13.5	36

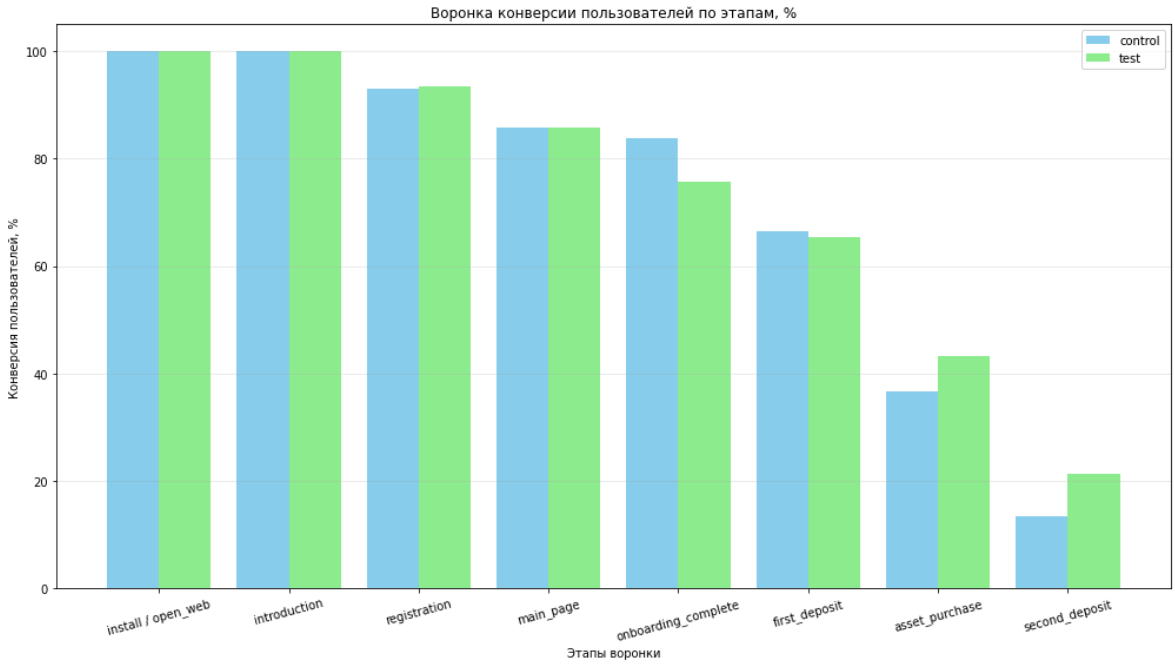
In [50]:
```
test_funnel_df
```

Out[50]:

	event_name	users_count	conversion_from_first_%	conversion_from_previous_
0	install / open_web	4568	100.0	Na
1	introduction	4568	100.0	100
2	registration	4265	93.4	93
3	main_page	3919	85.8	91
4	onboarding_complete	3462	75.8	88
5	first_deposit	2987	65.4	86
6	asset_purchase	1977	43.3	66
7	second_deposit	974	21.3	49

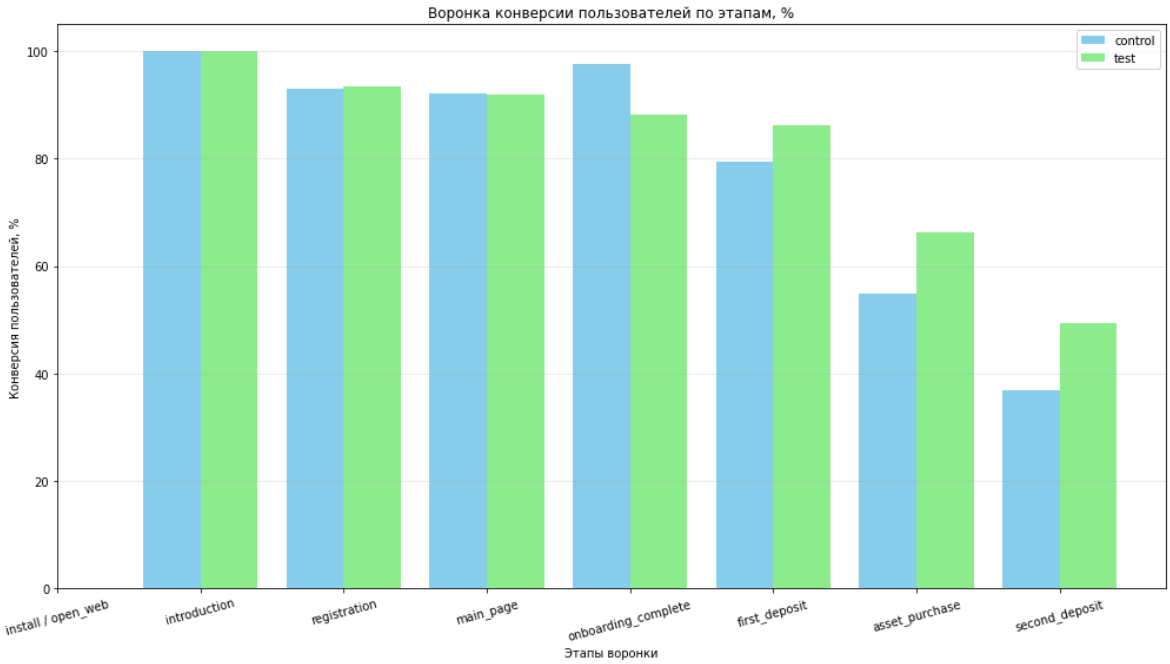
In [51]:
```
events = control_funnel_df['event_name']
x = np.arange(len(events))

width = 0.4

plt.figure(figsize=(14, 8))

plt.bar(x - width/2, control_funnel_df['conversion_from_first_%'], width=width,
plt.bar(x + width/2, test_funnel_df['conversion_from_first_%'], width=width, lab

plt.xticks(x, events, rotation=15)
plt.xlabel('Этапы воронки')
plt.ylabel('Конверсия пользователей, %')
plt.title('Воронка конверсии пользователей по этапам, %')
plt.grid(axis='y', alpha=0.3)
plt.legend()
```

```
plt.tight_layout()
plt.show()
```



На воронке видим падение пользователей тестовой группы на этапе завершения онбординга. Вероятно, онбординг стал сложнее и часть пользователей не стала его завершать.
Количество пользователей, которые в итоге купили актив, а также пользователей, которые внесли второй депозит заметно выросло.
Онбординг: снижение на 8%
Покупка актива: рост на 7%
Второй депозит: рост на 8%

In [52]:
```
events = control_funnel_df['event_name']
x = np.arange(len(events))

width = 0.4

plt.figure(figsize=(14, 8))

plt.bar(x - width/2, control_funnel_df['conversion_from_previous_%'], width=widt
plt.bar(x + width/2, test_funnel_df['conversion_from_previous_%'], width=width,

plt.xticks(x, events, rotation=15)
plt.xlabel('Этапы воронки')
plt.ylabel('Конверсия пользователей, %')
plt.title('Воронка конверсии пользователей по этапам, %')
plt.grid(axis='y', alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()
```



Step-by-step воронка помогает детальнее увидеть разницу между предыдущими этапами:
явный рост пользователей после завершения онбординга в тестовой группе подтверждается.

Промежуточный вывод:

Исходя из текущих данных, можем сделать вывод, что новый онбординг действительно повлиял на количество депозитов и покупки активов среди новых пользователей в тестовой группе.

Стоит отметить, что количество пользователей, прошедших онбординг в тестовой группе **меньше** чем в контрольной: 97% в контрольной и 88% в тестовой. Это может говорить о том, что онбординг в новой версии стал сложнее для прохождения и часть пользователей теряется, но эта ситуация должно компенсироваться ростом покупок и депозитов.
Таким образом отсеиваются пользователи, не готовые к покупкам и риску, а те кто готов - получили в обновлённом онбординге гораздо больше полезной информации и с большей уверенностью вкладывают деньги.

Все последующие этапы имеют лучшую конверсию в тестовой группе:

- первый депозит: 86% (79% в контрольной)
- покупка актива: 66% (55% в контрольной)
- второй депозит: 49% (37% в контрольной)

Влияние новой фичи на конверсию во второй депозит с учётом уровня риска купленного актива.

Сформулируем гипотезы для каждой группы риска:

H0 (нулевая гипотеза): Конверсия во второй депозит одинакова в контрольной и тестовой группах среди пользователей, выбравших активы с высоким риском.
H1 (альтернативная гипотеза): Конверсия во второй депозит отличается между контрольной и тестовой группами среди пользователей с активами высокого риска.

H0 (нулевая гипотеза): Конверсия во второй депозит одинакова в контрольной и тестовой группах среди пользователей, выбравших активы со средним риском.
H1 (альтернативная гипотеза): Конверсия во второй депозит отличается между контрольной и тестовой группами среди пользователей с активами среднего риска.

H0 (нулевая гипотеза): Конверсия во второй депозит одинакова в контрольной и тестовой группах среди пользователей, выбравших активы с низким риском.
H1 (альтернативная гипотеза): Конверсия во второй депозит отличается между контрольной и тестовой группами среди пользователей с активами низкого риска.

In [53]:
```
ab_assets = ab_df[ab_df['event_name'] == 'asset_purchase']

ab_first_asset = ab_assets.sort_values(['user_id', 'event_ts']).drop_duplicates(
ab_risk_conversion_df = ab_first_asset[['user_id', 'group', 'risk_level']].copy(
ab_risk_conversion_df.head()

ab_second_deposit_users = ab_df[ab_df['event_name'] == 'second_deposit']['user_i
ab_risk_conversion_df['second_deposit_made'] = ab_risk_conversion_df['user_id'].

ab_conversion_stats = (
    ab_risk_conversion_df.groupby(['group', 'risk_level'])['second_deposit_made'
    .agg(['count', 'sum'])
    .rename(columns={'count': 'users_with_asset', 'sum': 'users_with_second_depo
    .reset_index()
)

ab_conversion_stats['conversion_rate'] = (
    ab_conversion_stats['users_with_second_deposit'] / ab_conversion_stats['user
)


ab_conversion_stats = ab_conversion_stats.sort_values(['group', 'risk_level']).r
ab_conversion_stats
```

Out[53]:

	group	risk_level	users_with_asset	users_with_second_deposit	conversion_rate
0	control	high	893	247	0.276596
1	control	low	285	142	0.498246
2	control	medium	595	265	0.445378
3	test	high	737	339	0.459973
4	test	low	496	285	0.574597
5	test	medium	744	350	0.470430

In [54]:
```
p_vals = []
for level in ab_conversion_stats['risk_level'].unique():
    ctrl = ab_conversion_stats[(ab_conversion_stats['group']=='control') &
```

```
                                (ab_conversion_stats['risk_level']==level)]
    tst = ab_conversion_stats[(ab_conversion_stats['group']=='test') &
                              (ab_conversion_stats['risk_level']==level)]
    successes = [int(ctrl['users_with_second_deposit']), int(tst['users_with_sec
    trials = [int(ctrl['users_with_asset']), int(tst['users_with_asset'])]
    z_stat, p_value = proportions_ztest(successes, trials)
    p_vals.append({'risk_level': level, 'z_stat': z_stat, 'p_value': p_value})

p_vals_df = pd.DataFrame(p_vals)
p_vals_df
```

Out[54]:

	risk_level	z_stat	p_value
0	high	-7.678906	1.604527e-14
1	low	-2.063426	3.907219e-02
2	medium	-0.914050	3.606905e-01

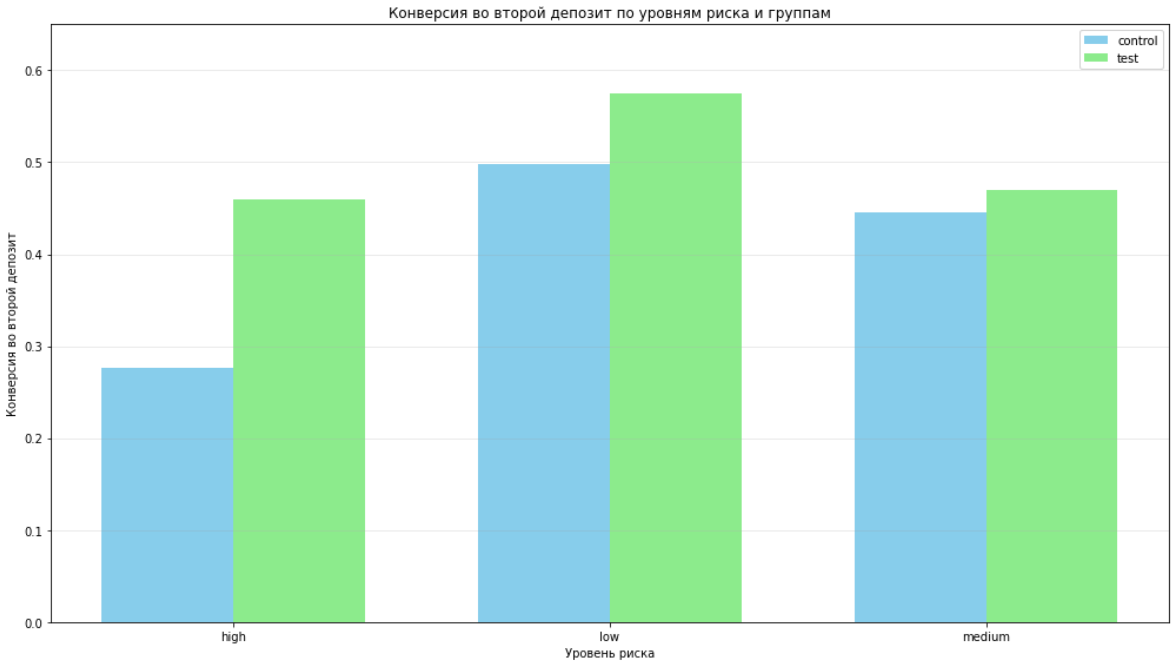
In [55]:
```
for level, p in zip(p_vals_df['risk_level'], p_vals_df['p_value']):
    if p < 0.05:
        display(Markdown(f"**Уровень риска «{level}»: статистически значимая раз
    else:
        display(Markdown(f"**Уровень риска «{level}»: разница не является статис
```

Уровень риска «high»: статистически значимая разница (p = 0.0000)

Уровень риска «low»: статистически значимая разница (p = 0.0391)

Уровень риска «medium»: разница не является статистически значимой (p = 0.3607)

In [56]:
```
pivot_df = ab_conversion_stats.pivot(index='risk_level', columns='group', values
risk_levels = pivot_df.index.tolist()
x = np.arange(len(risk_levels))
width = 0.35

plt.figure(figsize=(14, 8))

plt.bar(x - width/2, pivot_df['control'], width, label='control', color='skyblue
plt.bar(x + width/2, pivot_df['test'], width, label='test', color='lightgreen')

plt.xlabel('Уровень риска')
plt.ylabel('Конверсия во второй депозит')
plt.title('Конверсия во второй депозит по уровням риска и группам')
plt.xticks(x, risk_levels)
plt.ylim(0, 0.65)
plt.grid(axis='y', alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()
```



Вывод по влиянию новой фичи на конверсию во второй депозит с учётом уровня риска:

- Высокий риск: прирост +18%:
  Обучающий онбординг особенно сильно помог тем, кто выбрал активы с высоким риском: они почувствовали себя увереннее и гораздо чаще вернулись за вторым депозитом.
  Альтернативная гипотеза подтверждается, конверсия в тестовой группе отличается.
- Средний риск: прирост +2.51%:
  Для средней категории рисков эффект менее выражен и не является статистически значимым (может быть случайным).
  Нулевая гипотеза подтверждается, конверсия в тестовой группе не отличается.
- Низкий риск: прирост +7.64%:
  Осторожные инвесторы стали более склонны к повторным вложениям после подробного объяснения рисков в новом онбординге.
  Альтернативная гипотеза подтверждается, конверсия в тестовой группе отличается.

Итог: обновлённый онбординг даёт наиболее сильный эффект для пользователей, выбирающих высокорисковые активы, и заметный, но менее мощный для низкорисковых активов. В сегменте среднего риска значимого изменения не произошло.
Это говорит о необходимости разного подхода: усиливать обучающий контент для высокорисковых покупателей и придумать дополнительные стимулы для среднего сегмента.

Анализ метрик A/B-эксперимента.
Вспомним и рассчитаем метрики:

- Ключевая метрика — средняя сумма всех депозитов на одного пользователя (включая тех, кто установил приложение или открыл веб-версию).
- Барьерная метрика — конверсия из регистрации в первый депозит.
- Вспомогательная метрика 1 — конверсия из первого депозита во второй.
- Вспомогательная метрика 2 — средняя сумма всех депозитов на пользователя, который открыл хотя бы один депозит.

После этого проверим статистическую значимость различий метрик между группами эксперимента.

Для ключевой метрики также изучим:

- накопленную динамику изменения по дням эксперимента для каждой группы,
- стабильность p-value во время эксперимента.

Ключевая метрика:

Гипотезы:
H0: средняя сумма всех депозитов на одного пользователя в тестовой группе не отличается от контрольной.
H1: средняя сумма всех депозитов на одного пользователя в тестовой группе отличается от контрольной.

In [57]:
```
all_users = ab_df[['group', 'user_id']].drop_duplicates()

user_deposits = ab_df.groupby(['group', 'user_id'])['amount'].sum().reset_index(
user_deposits = all_users.merge(user_deposits, on=['group', 'user_id'], how='lef

key_metric = user_deposits.groupby('group')['amount'].mean().reset_index(name='a
key_metric
```

Out[57]:

	group	avg_deposit
0	control	90.240149
1	test	92.460814

In [58]:
```
control_key_metric = key_metric[key_metric['group'] == 'control']['avg_deposit']
test_key_metric = key_metric[key_metric['group'] != 'control']['avg_deposit'].va
abs_lift = test_key_metric - control_key_metric
rel_lift = (test_key_metric / control_key_metric - 1) * 100


t1 = user_deposits[user_deposits['group'] == 'control']['amount']
t2 = user_deposits[user_deposits['group'] == 'test']['amount']
_, p_value = ttest_ind(t1, t2)

display(Markdown(f"**Средняя сумма депозитов на одного пользователя по группам:*
display(key_metric.style.format({'avg_deposit': '{:.2f}'}))

display(Markdown(f"**Абсолютное изменение:** {abs_lift:.2f}"))
display(Markdown(f"**Относительное изменение:** {rel_lift:.2f}%"))

if p_value < 0.05:
```

```
    display(Markdown(f"**Разница статистически значима (p-value = {p_value:.4f})
else:
    display(Markdown(f"**Разница не является статистически значимой (p-value = {
```

Средняя сумма депозитов на одного пользователя по группам:

	group	avg_deposit
0	control	90.24
1	test	92.46

Абсолютное изменение: 2.22

Относительное изменение: 2.46%

Разница не является статистически значимой (p-value = 0.3313)

Результаты теста показали, что хоть у тестовой группы средняя сумма всех депозитов на одного пользователя и выше на 2.46%, этот результат не является статистически значимым (p-value = 0.33) и может оказаться случайным.

Нулевая гипотеза подтверждается: средняя сумма всех депозитов на одного пользователя в тестовой группе не отличается от контрольной.

In [163…

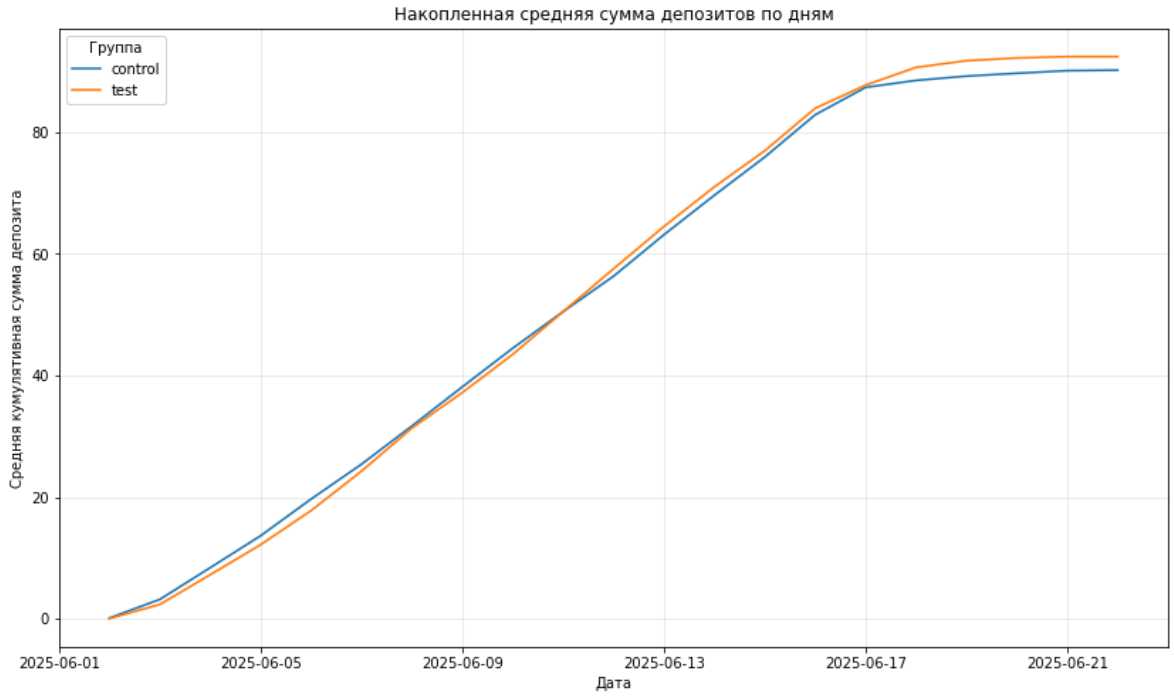
```
ab_df['date'] = ab_df['event_ts'].dt.date
start, end = ab_df['date'].min(), ab_df['date'].max()

idx = pd.MultiIndex.from_product([
    ab_df['user_id'].unique(),
    pd.date_range(start, end).date
], names=['user_id', 'date'])

daily = (
    ab_df.groupby(['user_id','date'])['amount']
         .sum()
         .reindex(idx, fill_value=0)
         .reset_index()
)

daily['cumsum'] = daily.groupby('user_id')['amount'].cumsum()

groups = ab_df[['user_id','group']].drop_duplicates()
daily = daily.merge(groups, on='user_id')
daily_group_avg = daily.groupby(['date','group'])['cumsum'].mean().reset_index()
daily_group_avg.head()
```

Out[163…

	date	group	cumsum
0	2025-06-02	control	0.097380
1	2025-06-02	test	0.013792
2	2025-06-03	control	3.163606
3	2025-06-03	test	2.329247
4	2025-06-04	control	8.380854

In [165…

```
plt.figure(figsize=(14,8))

for grp, grp_df in daily_group_avg.groupby('group'):
    plt.plot(grp_df['date'], grp_df['cumsum'], label=grp)

plt.title('Накопленная средняя сумма депозитов по дням')
plt.xlabel('Дата')
plt.ylabel('Средняя кумулятивная сумма депозита')
plt.legend(title='Группа')
plt.grid(alpha=0.3)
plt.show()
```



На графике видно, что средняя сумма депозитов практически одинакова для всех пользователей.

В начале A/B эксперимента тестовая группа немного отставала, но к концу теста начала опережать контрольную.

Можно сделать вывод, что новый онбординг не ухудшает ключевую метрику.

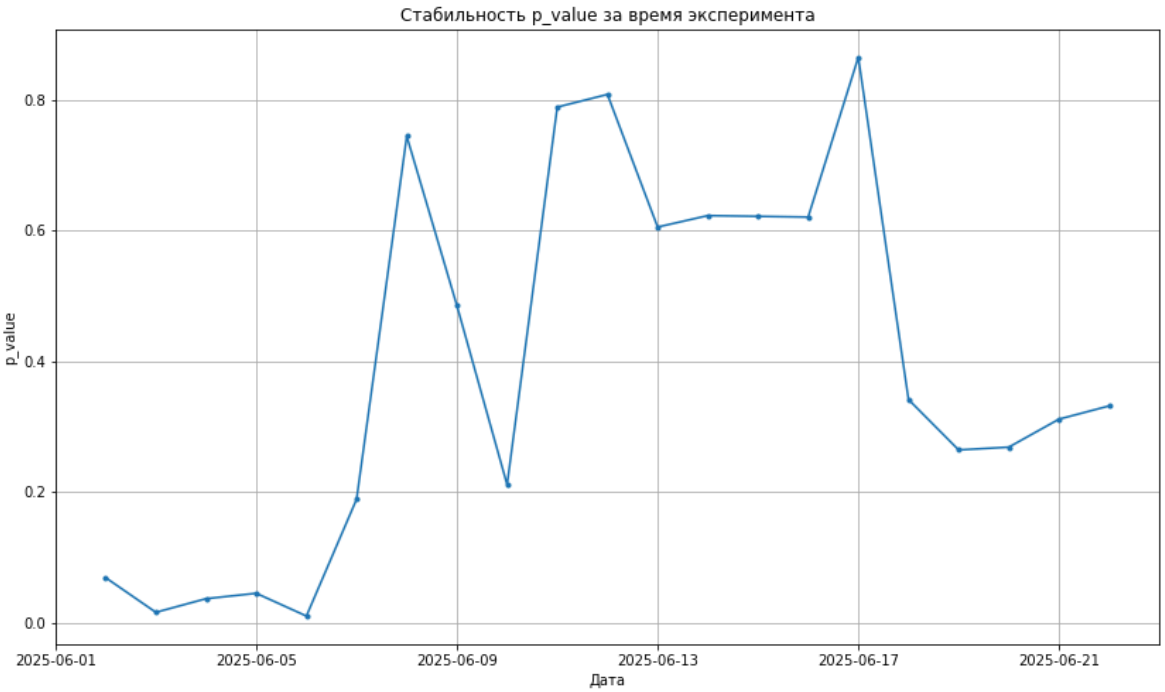
In [61]:

```
results = []

for current_date in sorted(daily_user_sum['date'].unique()):

    subset = daily_user_sum[daily_user_sum['date'] <= current_date]
    last_day = subset.groupby(['user_id'])['date'].max().reset_index()
    data_on_date = subset.merge(last_day, on=['user_id', 'date'], how='inner')

    control_group = data_on_date[data_on_date['group'] == 'control']['cumsum']
    test_group = data_on_date[data_on_date['group'] == 'test']['cumsum']

    _, p_value = ttest_ind(control_group, test_group)
    results.append({'date': current_date, 'p_value': p_value})

result_df = pd.DataFrame(results)
result_df.head(10)
```

Out[61]:

	date	p_value
0	2025-06-02	0.068485
1	2025-06-03	0.015544
2	2025-06-04	0.036423
3	2025-06-05	0.044654
4	2025-06-06	0.009878
5	2025-06-07	0.188542
6	2025-06-08	0.743899
7	2025-06-09	0.484634
8	2025-06-10	0.210852
9	2025-06-11	0.788452

In [62]:
```
plt.figure(figsize=(14, 8))

plt.plot(result_df['date'], result_df['p_value'], marker='.')

plt.title('Стабильность p_value за время эксперимента')
plt.ylabel('p_value')
plt.xlabel('Дата')
plt.grid()
plt.show()
```



Волатильность графика p-value за время проведения эксперимента наглядно показывает проблему подглядывания, особенно в первую неделю теста.

Барьерная метрика:

Гипотезы:

H0: конверсия из регистрации в первый депозит одинакова в тестовой и контрольной группах.

H1: конверсия из регистрации в первый депозит в тестовой группе отличается от контрольной.

In [63]:
```
registered = ab_df[ab_df['event_name'] == 'registration']
first_deposit = ab_df[(ab_df['event_name'] == 'first_deposit') & ab_df['user_id'

reg_count = registered.groupby('group')['user_id'].nunique()
deposit_count = first_deposit.groupby('group')['user_id'].nunique()

results_df = pd.DataFrame({
   'registered': reg_count,
   'first_deposit': deposit_count
}).reset_index()

results_df['CR, %'] = round(results_df['first_deposit'] / results_df['registered
results_df
```

Out[63]:

	group	registered	first_deposit	CR, %
0	control	4512	3228	71.54
1	test	4265	2987	70.04

In [64]:
```
control_barrier_metric = results_df[results_df['group'] == 'control']['CR, %'].v
test_barrier_metric = results_df[results_df['group'] == 'test']['CR, %'].values[
barrier_abs_lift = test_barrier_metric - control_barrier_metric
barrier_rel_lift = (test_barrier_metric / control_barrier_metric - 1) * 100

display(results_df)
display(Markdown(f'**Абсолютное изменение**: {barrier_abs_lift:.2f}%'))
display(Markdown(f'**Относительное изменение**: {barrier_rel_lift:.2f}%'))
```

	group	registered	first_deposit	CR, %
0	control	4512	3228	71.54
1	test	4265	2987	70.04

Абсолютное изменение: -1.50%

Относительное изменение: -2.10%

In [65]:
```
successes = [
    results_df[results_df['group'] == 'control']['first_deposit'].values[0],
    results_df[results_df['group'] == 'test']['first_deposit'].values[0]
]
total = [
    results_df[results_df['group'] == 'control']['registered'].values[0],
    results_df[results_df['group'] == 'test']['registered'].values[0]
]


stat_ztest, p_value_ztest = proportions_ztest(successes, total)
```

```
if p_value_ztest > 0.05:
    display(Markdown(f'pvalue={p_value_ztest:.4f} > 0.05'))
    display(Markdown('**Разница не является статистически значимой**'))
else:
    display(Markdown(f'pvalue={p_value_ztest:.4f} < 0.05'))
    display(Markdown('**Разница статистически значима**'))
```

pvalue=0.1205 > 0.05

Разница не является статистически значимой

Видим ухудшение конверсии пользователей из регистрации в первый депозит, но это может говоришь лишь о большей осведомленности таких пользователей, которые просто решили не рисковать или подробнее изучить тему инвестиций. Тем не менее, такое ухудшение конверсии не является статистически значимым и может быть случайным.

Нулевая гипотеза подтверждается: конверсия из регистрации в первый депозит одинакова в тестовой и контрольной группах.

Вспомогательная метрика 1:

Гипотезы:
H0: конверсия из первого депозита во второй одинакова в тестовой и контрольной группах.
H1: конверсия из первого депозита во второй в тестовой группе отличается от контрольной.

In [66]:
```
first_deposit = ab_df[ab_df['event_name']=='first_deposit']

second_deposit = ab_df[(ab_df['event_name'] == 'second_deposit') & ab_df['user_i

first_deposit_count = first_deposit.groupby('group')['user_id'].nunique()
second_deposit_count = second_deposit.groupby('group')['user_id'].nunique()

cr_results_df = pd.DataFrame({
   'first_deposit': first_deposit_count,
   'second_deposit': second_deposit_count
}).reset_index()

cr_results_df['CR, %'] = round(cr_results_df['second_deposit'] / cr_results_df['

cr_results_df
```

Out[66]:

	group	first_deposit	second_deposit	CR, %
0	control	3228	654	20.26
1	test	2987	974	32.61

In [67]:
```
control_cr_metric = cr_results_df[cr_results_df['group'] == 'control']['CR, %'].
test_cr_metric = cr_results_df[cr_results_df['group'] == 'test']['CR, %'].values
cr_abs_lift = test_cr_metric - control_cr_metric
cr_rel_lift = (test_cr_metric / control_cr_metric - 1) * 100
```

```
display(Markdown(f'**Абсолютное изменение**: {cr_abs_lift:.2f}%'))
display(Markdown(f'**Относительное изменение**: {cr_rel_lift:.2f}%'))
```

Абсолютное изменение: 12.35%

Относительное изменение: 60.96%

In [68]:
```
successes = [
    cr_results_df[cr_results_df['group'] == 'control']['second_deposit'].values[
    cr_results_df[cr_results_df['group'] == 'test']['second_deposit'].values[0]
]
total = [
    cr_results_df[cr_results_df['group'] == 'control']['first_deposit'].values[0
    cr_results_df[cr_results_df['group'] == 'test']['first_deposit'].values[0]
]


stat_ztest, p_value_ztest = proportions_ztest(successes, total)

if p_value_ztest > 0.05:
    display(Markdown(f'pvalue={p_value_ztest:.4f} > 0.05'))
    display(Markdown('**Разница не является статистически значимой**'))
else:
    display(Markdown(f'pvalue={p_value_ztest:.4f} < 0.05'))
    display(Markdown('**Разница статистически значима**'))
```

pvalue=0.0000 < 0.05

Разница статистически значима

In [69]:
```
display(stat_ztest)
display(p_value_ztest)
```

```
-11.061168697061532
1.9355885675373928e-28
```

В конверсии пользователей из первого депозита во второй видим значительные улучшения тестовой группы на 60%. Эта разница является статистически значимой. После прохождения нового онбординга, тестовые пользователи лучше понимают, что ждать от вложений, и не бросают приложение после первой сделки.

Альтернативная гипотеза подтверждается: конверсия из первого депозита во второй в тестовой группе отличается от контрольной.

Вспомогательная метрика 2:

Гипотезы:
H0: средняя сумма всех депозитов на пользователя, который открыл хотя бы один депозит одинакова в тестовой и контрольной группах.
H1: средняя сумма всех депозитов на пользователя, который открыл хотя бы один депозит в тестовой группе отличается от контрольной.

In [70]:
```
user_deposits = ab_df.groupby(['group', 'user_id'])['amount'].sum().reset_index(
positive_deposits = user_deposits[user_deposits['amount'] > 0]

metric2 = positive_deposits.groupby('group')['amount'].mean().reset_index()
metric2.rename(columns={'amount': 'avg_deposit'}, inplace=True)
```

```
metric2
```

Out[70]:

	group	avg_deposit
0	control	135.500000
1	test	141.399732

In [71]:
```
control_metric2 = metric2[metric2['group'] == 'control']['avg_deposit'].values[0
test_metric2 = metric2[metric2['group'] == 'test']['avg_deposit'].values[0]
m2_abs_lift = test_metric2 - control_metric2
m2_rel_lift = (test_metric2 / control_metric2 - 1) * 100

display(Markdown(f'**Абсолютное изменение**: {m2_abs_lift:.2f}%'))
display(Markdown(f'**Относительное изменение**: {m2_rel_lift:.2f}%'))
```

Абсолютное изменение: 5.90%

Относительное изменение: 4.35%

In [72]:
```
control = positive_deposits[positive_deposits['group'] == 'control']['amount']
test = positive_deposits[positive_deposits['group'] == 'test']['amount']

_, p_value = ttest_ind(test, control)

if p_value > 0.05:
    display(Markdown(f'pvalue={p_value:.4f} > 0.05'))
    display(Markdown('**Разница не является статистически значимой**'))
else:
    display(Markdown(f'pvalue={p_value:.4f} < 0.05'))
    display(Markdown('**Разница статистически значима**'))
```

pvalue=0.0347 < 0.05

Разница статистически значима

Средняя сумма депозитов среди платящих пользователей также увеличилась на 5.9% в тестовой группе. Вероятно, новый онбординг придал пользователям уверенности и больше понимания стратегии инвестиций.

Альтернативная гипотеза подтверждается: средняя сумма всех депозитов на пользователя, который открыл хотя бы один депозит в тестовой группе отличается от контрольной.

Вывод

Ключевая метрика:

- Средний депозит на пользователя вырос с 90.24 (control) до 92.46 (test), то есть на +2.46%.
- Разница не является статистически значимой (p = 0.3313), поэтому нельзя однозначно утверждать, что обновлённый онбординг влияет на средний чек всех пользователей.
- Нулевая гипотеза подтверждается: между группами нет разницы.

Барьерная метрика:

- Конверсия упала с 71.54% до 70.04%, но это изменение не является статистически значимым (изменение -1.5%, p = 0.1205).
- Тестовая группа чуть реже совершает первый депозит, но разница тоже не значима. Вероятная причина - более осознанный выбор новичков после знакомства с рисками.
- Нулевая гипотеза подтверждается: между группами нет разницы.

Вспомогательная метрика 1:

- Конверсия выросла с 20.26 % до 32.61 % (изменение +12.35%, p < 0.05).
- Это статистически значимый и большой эффект: пользователи из тестовой группы значительно чаще возвращаются за повторным пополнением счёта.
- Альтернативная гипотеза подтверждается: между группами есть разница.

Вспомогательная метрика 2:

- Средний чек платящих поднялся с 135.50 до 141.40 (изменение +4.35%).
- Растёт, эффект статистически значимый. Вероятно новый онбординг придал пользователям уверенности.
- Альтернативная гипотеза подтверждается: между группами есть разница.

Накопленная динамика:

- Средняя сумма депозитов практически одинакова для всех пользователей, есть небольшое отставание в первую неделю, а затем небольшое преимущество во вторую неделю. Это означает, что онбординг не ухудшает ключевую метрику.

---

4. Анализ изменений суммы депозитов на платящего пользователя

Новая фича могла повлиять на поведение пользователей.

- Пользователи, которые раньше вносили небольшие суммы, могли стать более осторожными, сократить свои вложения или совсем перестать платить. Это отразится в снижении 25-го перцентиля суммы депозитов в тестовой группе.
- Пользователи, которые склонны к более крупным инвестициям, могли сильнее вовлечься в продукт и начать вносить больше средств. Это отразится в росте 75-го перцентиля.

Используя бутстрап, сравним разницы перцентилей суммы всех депозитов на платящего пользователя в контрольной и тестовой группах.

In [73]:
```
np.random.seed(341)

user_deposits = ab_df.groupby(['group', 'user_id'])['amount'].sum().reset_index(
paid_users = user_deposits[user_deposits['amount'] > 0]
```

```
control = paid_users[paid_users['group'] == 'control']['amount'].values
test = paid_users[paid_users['group'] == 'test']['amount'].values

n_iterations = 5000
percentiles = [25, 50, 75]
results = {}

for p in percentiles:
    boot_diffs = []
    for _ in range(n_iterations):
        boot_control = np.random.choice(control, size=len(control), replace=True
        boot_test = np.random.choice(test, size=len(test), replace=True)

        control_p = np.percentile(boot_control, p)
        test_p = np.percentile(boot_test, p)

        boot_diffs.append(test_p - control_p)

    ci = np.percentile(boot_diffs, [2.5, 97.5])
    mean_diff = np.mean(boot_diffs)

    results[p] = {
        'control_p': np.percentile(control, p),
        'test_p': np.percentile(test, p),
        'mean_diff': mean_diff,
        'ci': ci,
        'samples': boot_diffs
    }

for p in percentiles:
    r = results[p]

    text = f"""
**{p}-й перцентиль**
- Control: **{r['control_p']:.2f}**
- Test: **{r['test_p']:.2f}**
- Разница: **{r['mean_diff']:.2f}**
- 95% ДИ: **[{r['ci'][0]:.2f}, {r['ci'][1]:.2f}]**
"""
    display(Markdown(text))
```

25-й перцентиль

- Control: **95.00**
- Test: **48.00**
- Разница: **-46.84**
- 95% ДИ: **[-50.00, -44.00]**

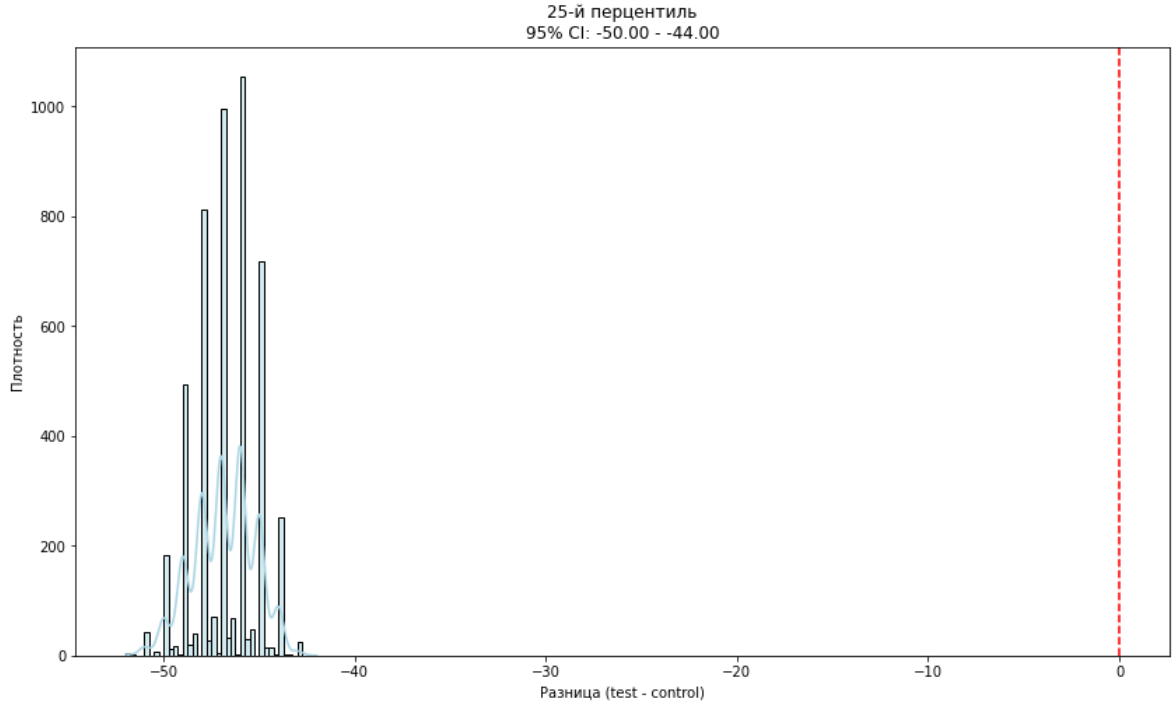
50-й перцентиль

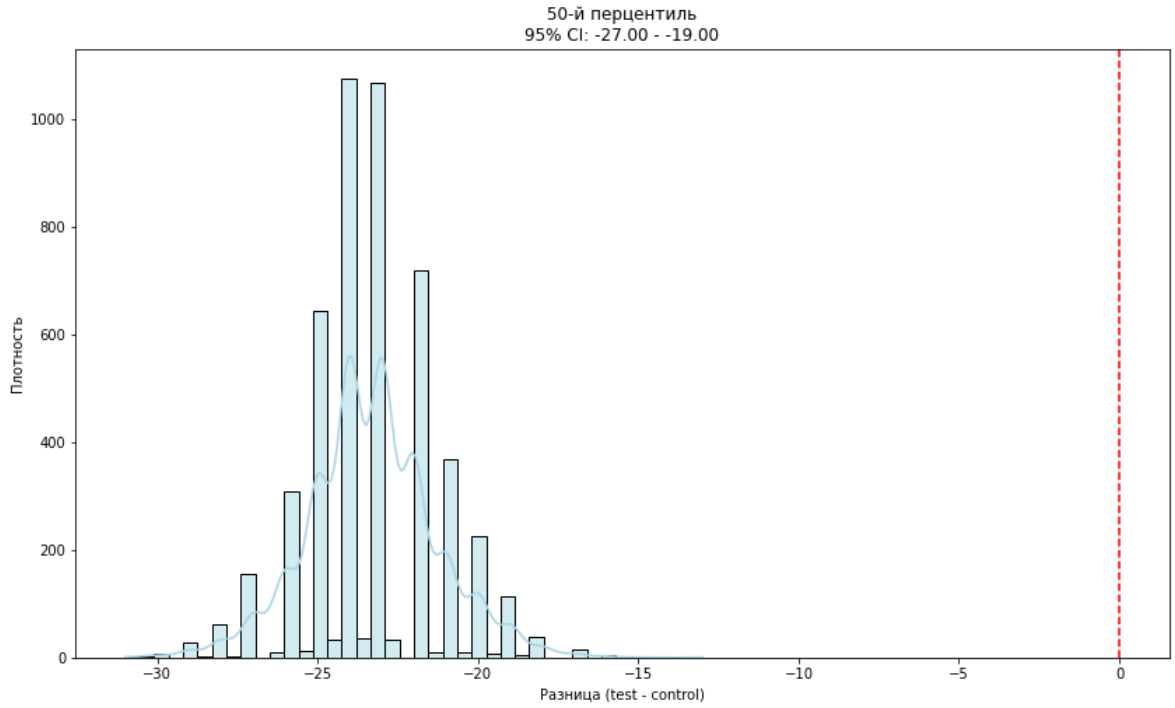
- Control: **118.00**
- Test: **94.00**
- Разница: **-23.32**
- 95% ДИ: **[-27.00, -19.00]**

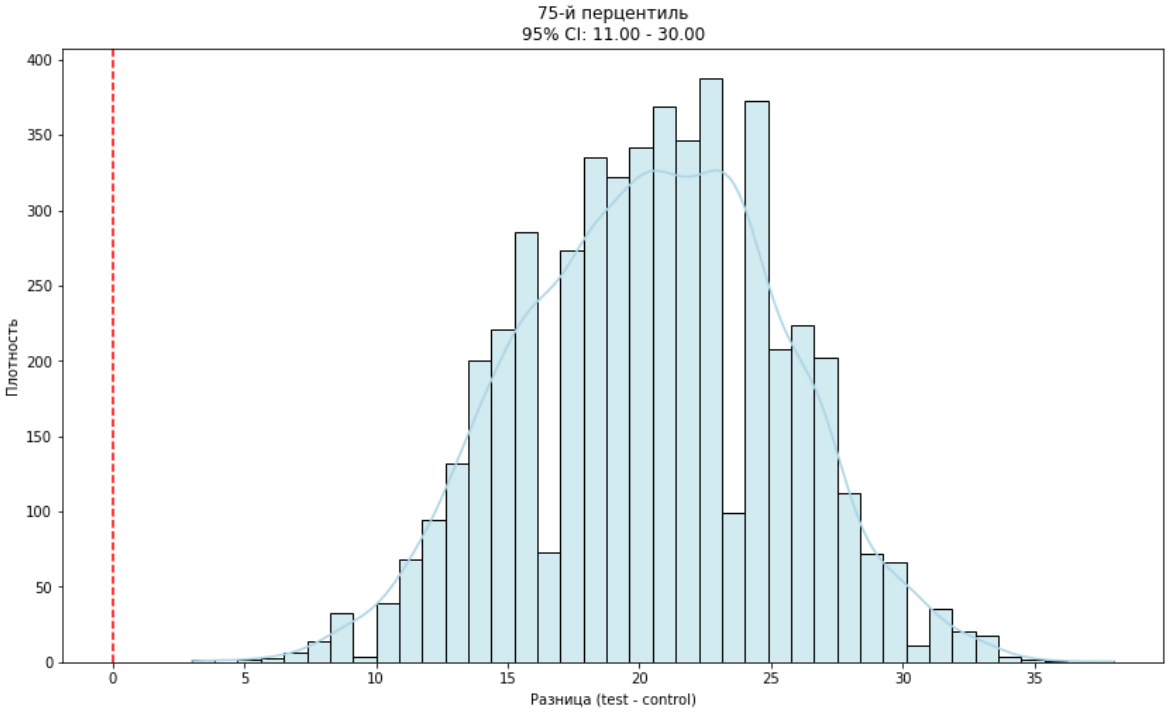
75-й перцентиль

- Control: **159.00**
- Test: **179.50**
- Разница: **20.44**
- 95% ДИ: **[11.00, 30.00]**

In [74]:
```
for p in percentiles:
    plt.figure(figsize=(14, 8))
    sns.histplot(results[p]['samples'], bins=40, kde=True, color='lightblue')
    plt.axvline(0, color='red', linestyle='--')
    plt.title(f'{p}-й перцентиль\n95% CI: {results[p]["ci"][0]:.2f} - {results[p
    plt.xlabel('Разница (test - control)')
    plt.ylabel('Плотность')
    plt.show()
```







По данным бутстрапа можно сделать вывод, что предположение оказалось верным:

- 25 перцентиль платящих пользователей в тестовой группе вносят существенно меньше средств. Разница: -46.84 Доверительный интервал [-50.00, -44.00]
- 50 перцентиль также вносит меньше средств. Разница: -23.32. ДИ: [-27.00, -19.00]
- 75 перцентиль показывает значительные улучшения. Разница: +20.44. ДИ: [11.00, 30.00]

Поскольку все доверительные интервалы не пересекают 0, можно сделать вывод, что данные корректны.

- Обучающий онбординг отпугнул часть наиболее осторожных инвесторов (они стали вносить меньшие суммы или перестали платить), что видно из снижения 25-го и 50-го процентилей.
- Снижение медианы обусловлено тем, что новый онбординг изменил поведение большинства пользователей: теперь они больше осведомлены об активах и разделились на тех, кто сократил свои вложения (или прекратил) и на тех, кто решил выбрать высокорисковые активы.
- При этом самые активные инвесторы наоборот активизировались и увеличили свои вклады (рост 75-го процентиля).
- Это полностью согласуется с гипотезами: информация о рисках фильтрует слабонервных, а более уверенные (или готовые к риску) становятся ещё более вовлечёнными.

---

5. Выводы

Общий вывод:

Метрики:

- Ключевая метрика (средний депозит на пользователя): Рост +2.5%, но статистически не значимо, нет уверенности, что онбординг в целом увеличивает средний чек всех пользователей.
- Барьерная метрика (регистрация в 1-й депозит): Снижение -1.5%, но статистически не значимо, есть риск: подробный онбординг может отсеять часть осторожных новичков.
- Вспомогательная метрика 1 (1-й депозит во 2-й депозит): Рост +12.4%, статистически значимо. Инсайт: обучение заметно повышает повторные вклады.
- Вспомогательная метрика 2 (средний депозит среди платящих): Рост +4.4%, статистически значимо. Вывод: платящие пользователи стали вкладывать немного больше.

Сегментация по риску:

- High-risk: +18.3%
- Low-risk: +7.6%
- Medium-risk: +2.5% (статистически не значимо)

Инсайт: обучение особенно эффективно для самых рискованных инвесторов, а средний сегмент требует дополнительных стимулов.

Бутстрап-анализ перцентилей:

- 25-й и 50-й перцентили существенно упали (пользователи стали вносить меньше или не платят).
- 75-й перцентиль значимо вырос (топ-инвесторы вложили больше).
  Вывод: онбординг фильтрует осторожных, усиливает вовлечённость «профессионалов».

Рекомендации:

- Сделать первые шаги проще: например, предложить начальный депозит или бонус, чтобы не отпугивать осторожных пользователей.
- Усилить обучающий контент и таргетинг на high-risk сегмента, так как наблюдается значительный рост вложений.
- Разработать стимулирующие механики (например, уменьшение комиссии брокера) для medium-risk пользователей.
- Продолжить наблюдение после окончания теста, чтобы полностью отследить отложенные эффекты.

Итог:

Гипотеза роста подтвердилась: обучающий онбординг помогает пользователям лучше понимать принципы инвестирования, поэтому они будут чаще открывать второй депозит.
Гипотеза риска подтвердилась: информация о возможных потерях и высоких

рисках отпугнёт некоторых новичков, особенно самых осторожных, что снизит конверсию в первый депозит. Конверсия действительно немного снизилась. *Дополнительная гипотеза* подтвердилась: после нового онбординга пользователи, которые выбрали высокорискованные активы, будут чаще, чем раньше, возвращаться и открывать второй депозит. При старом онбординге пользователи часто покупали активы с высоким риском без понимания последствий. Это приводило к потерям и оттоку после первого депозита.

Обновлённый онбординг подтвердил все три гипотезы: обучение помогает пользователям чаще делать повторные депозиты.
Но его нужно доработать: упростить первый шаг и лучше адаптировать под пользователей со средним уровнем готовности к инвестициям, чтобы увеличить общую прибыль.

Рекомендуется введение нового онбординга для всех пользователей с небольшой доработкой.
Фича имеет потенциал для усиления LTV и повторных действий, но требует точечной настройки: она усиливает сильных, но может демотивировать осторожных.