

Задание 1 «Отчёт о продажах в Телеграм»

Вы работаете аналитиком в стартапе в области обучения взрослых английскому языку. Компания растёт быстро, целый штат маркетологов и продактов придумывает новые механики привлечения пользователей, улучшая коммерческие продукты.

Ещё есть отдел продаж. В нём сотрудники звонят потенциальным клиентам и продают пакеты уроков. Только вот маркетинг не дружит с продажами и красивой сквозной аналитики у Руководителя отдела продаж нет. Нужно ему помочь.

Никакого централизованного DWH или сложного BI - у компании нет, но бизнесу точно нужны основные метрики, причем завтра. Данные об основных событиях СЛМ пользователя записываются в Postgres.

Руководитель отдела продаж просит вас каждый день присылать в его telegram метрики или графики. Они должны помочь ему понимать как идут дела в отделе, так сказать держать руку на пульсе.

Данные

Все таблицы с данными в PostgreSQL. Параметры для подключения:

```
database = "quest-db", user = "rouser", password = "ZI6MVnmi", host = "178.62.242.91", port = "5433"
```

Таблицы:

- events — таблица событий по пользователям и датам
- events_dict — таблица со списком событий их описанием
- payments — транзакции пользователей

Задачи

1. Выберите 3 метрики, которые помогут руководителю отдела продаж контролировать ситуацию ежедневно — всё ли идёт нормально. Объясните свой выбор.
2. Напишите Телеграм Бота (скрипт), который будет отправлять ежедневный отчёт по этим метрикам в Телеграм руководителю. Чтобы показать, как работает бот вставьте его код в файл с ответом и прикрепите скриншот отправленного им сообщения, чтобы было видно от кого это. Под названием отправителя должно быть написано Бот, как на скриншоте ниже.

Анализ данных

Для начала попробуем понять по данным, как происходит работа в отделе. Я добавил продажи(транзакции) к событиям в таблице. И по получившимся данным сделал Сэнкей диаграммы с фактическими сменами статусов для первых 7 событий по каждому user_id. Больше - было бы слишком мелко в файле.

Диаграмма Сэнкей 1 Смены статусов

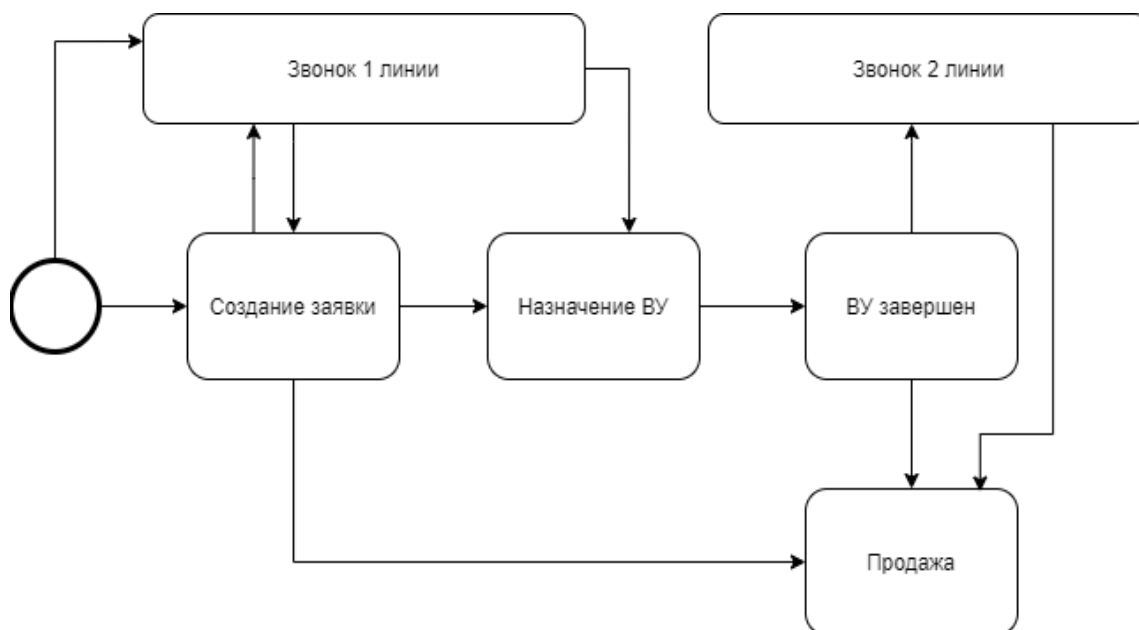
На второй диаграмме оставил только клиентов у которых встречается использование WA или ДУ.

Диаграмма Сэнкей 2 Использование новых каналов взаимодействия

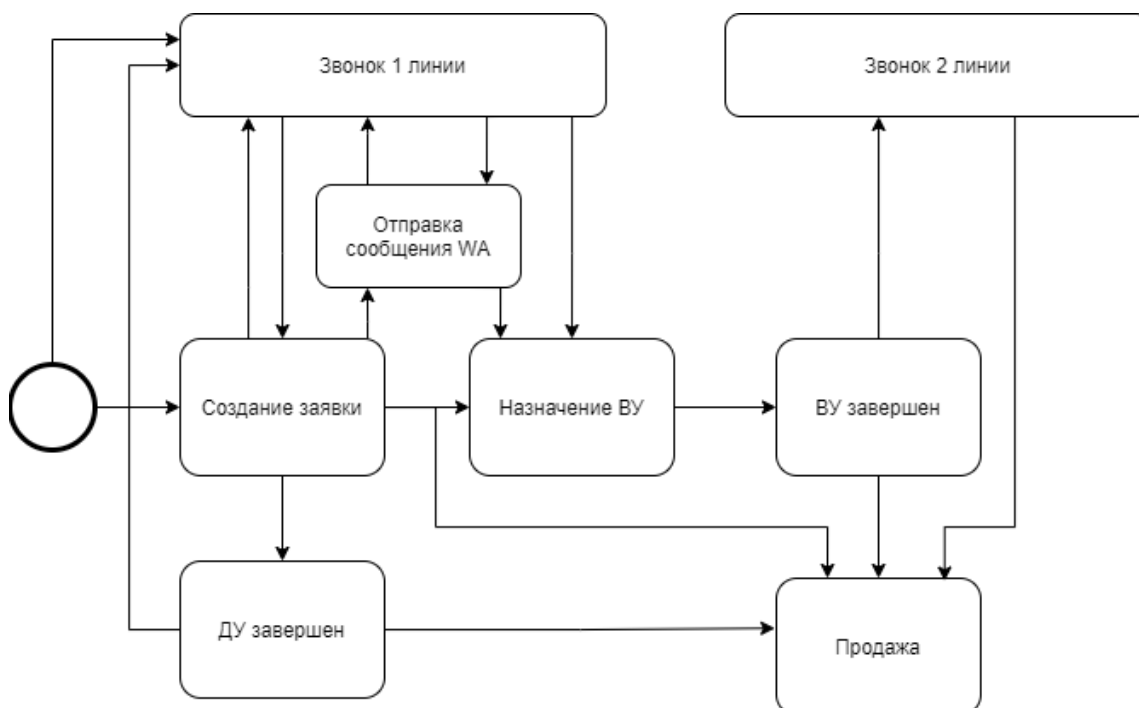
По ним уже можно предполагать, как идет обработка заявок. Хотя, например, цепочка «Создание заявки → Назначение ВУ → Создание заявки → Назначение ВУ → Назначение ВУ → Выход МВУ на

ВУ» без участия первой линии выглядит довольно странно. Думаю, учащиеся могут самостоятельно записываться на вводный урок и потом переносить запись.

Основные события по заявке выглядят так:



Если добавить общение в мессенджерах и демонстрационный урок:



Время обработки

У меня была гипотеза, что можно повысить конверсию, если уменьшить время обработки заявки первой и второй линией. Для проверки я отметил цепочки событий закончившиеся продажей, как успешные. Можно было ожидать, что в таких цепочках до звонка не будет проходить слишком много времени. Распределение времени дозвона выглядит так:

[Время обработки события Назначена задача на звонок 1 линии, часов](#)

[Время обработки события Назначена задача на вторую линию, часов](#)

Первая линия пытается дозвониться в течении ровно трех часов. Если не получилось, есть три варианта: Меняется статус на «Создание заявки», Меняется статус на «Назначение ВУ» или Будут звонить дальше. Отчего это зависит — не понятно.

Вторая линия начинает звонить сразу и продолжает в течение суток.

Судя по всему, время обработки заявок в обоих случаях уже отслеживается и оно выглядит адекватным. Стоит ли все равно включать его в метрики? Думаю, нет.

Метрики

Какой результат требуется от сотрудников отдела?

- Первая линия: обработать заявку и назначить вводный урок
- Методисты: провести вводный урок и сделать продажу. В задании нет четкого указания, что методисты входят в отдел продаж, но предположим, это так.
- Вторая линия: позвонить после вводного урока и сделать продажу

Анализировать сумму продажи или просто количество продаж? Не очень понятно, от чего зависит стоимость. На мой взгляд, в ежедневном режиме удобнее отслеживать количество. А «качество» продаж может не быть стабильным в пределах дня и больше подходит для недельной/месячной отчетности. Как и конверсия на всех этапах.

Еще можно отслеживать количество входящих заявок и назначенных без участия первой линии ВУ. Сотрудники отдела на это повлиять вроде бы не могут, но если что-то сломалось - лучше узнать максимально рано.

Стоит ли отдельно анализировать результаты использования мессенджеров и ДУ? Да, но не в ежедневном режиме. Нужно собрать больше данных и по ним принять решение по развитию этих инструментов.

Расчет метрик

Итак, что стоит отслеживать по итогам каждого дня?

1. Количество вводных уроков назначенных первой линией.

Считается как количество событий «Назначение ВУ» после события «Назначение задачи на звонок 1Л» или «Ученик ответил на звонок оператора 1л»

2. Количество вводных уроков после которых ученик совершил оплату без привлечения второй линии.

Считается как количество событий в таблице payments после события «Успешный ВУ», «Выход МВУ на ВУ», «Переход на ДУ (ДУ начался)» или «ДУ завершен»

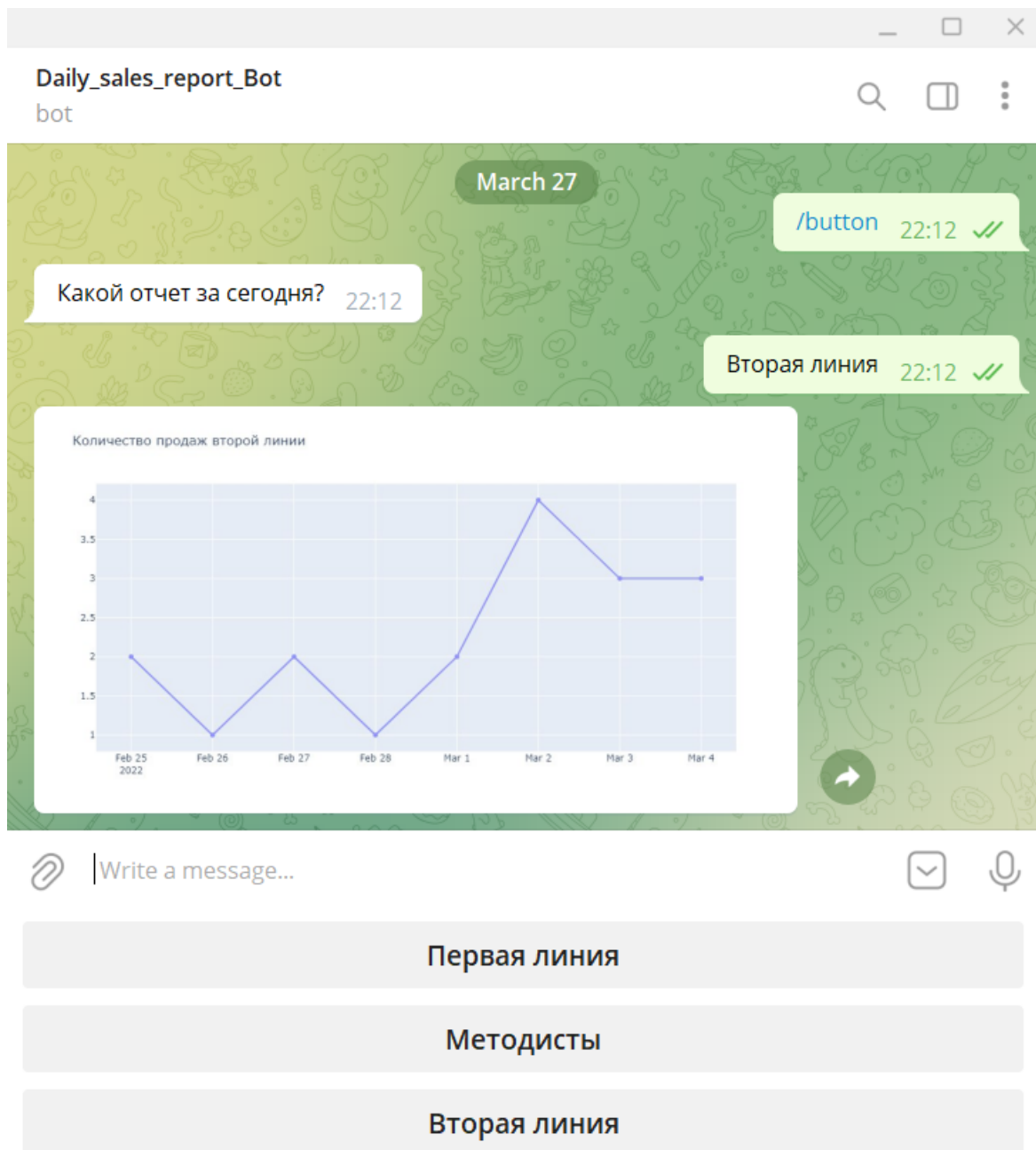
3. Количество вводных уроков после которых ученик совершил оплату с привлечением второй линии.

Считается как количество платежей (событий из таблицы payments) после события «Назначена задача на вторую линию» или «Дозвон 2Л»

Так же стоило бы отслеживать количество назначенных ВУ без участия первой линии. Но, если считать конверсию при ВУ стабильной, его можно прикинуть по соотношению (1) с (2) или (3). Если одно сильно изменилось без другого, стоит проверить количество заявок.

Бот

Пример работы бота



[Код отправки отчетов ботом](#)

Приложения

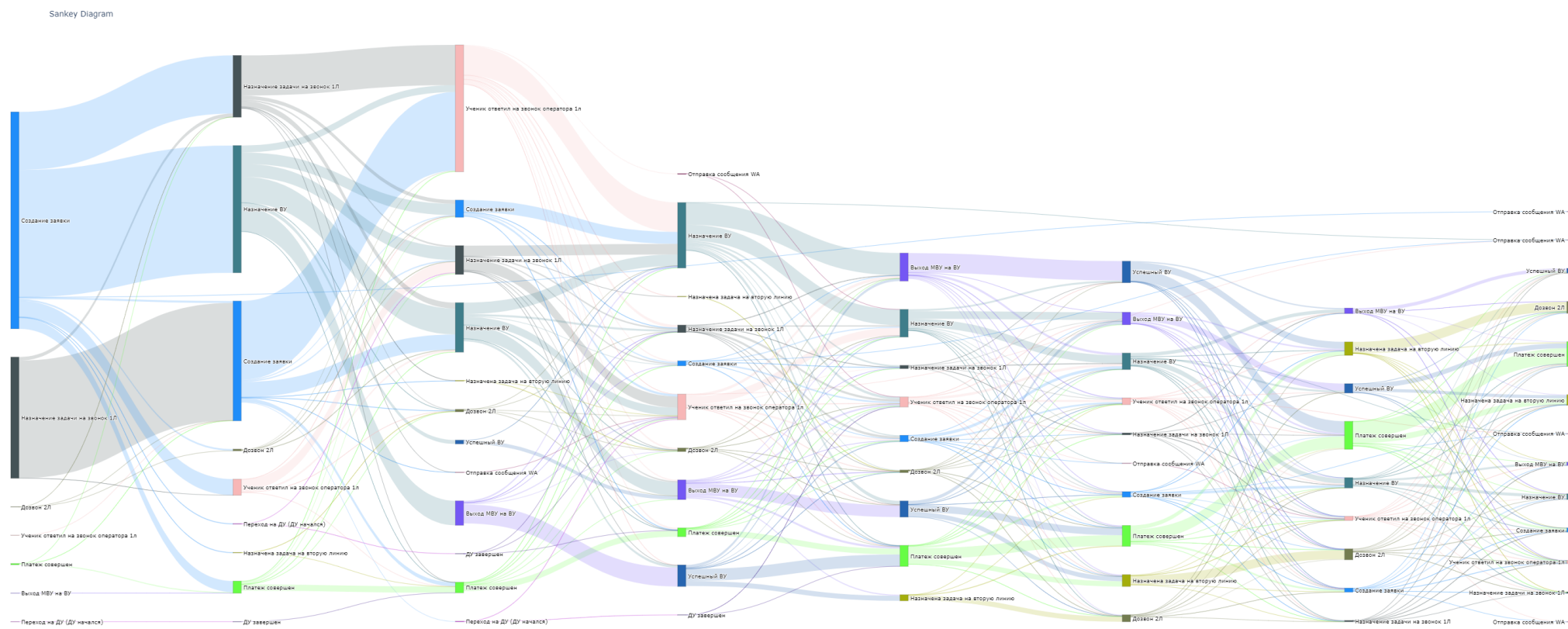


Рисунок 1: Диаграмма Сэнкей 1 Смены статусов

Sankey Diagram

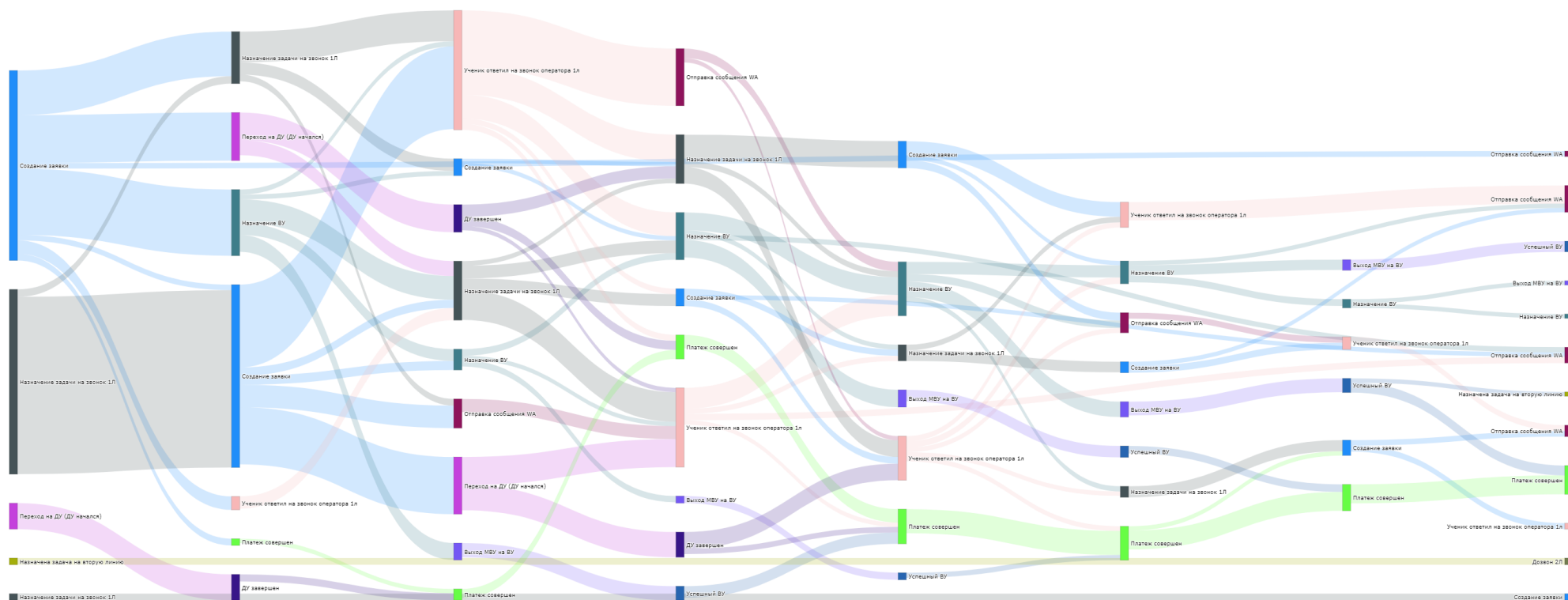
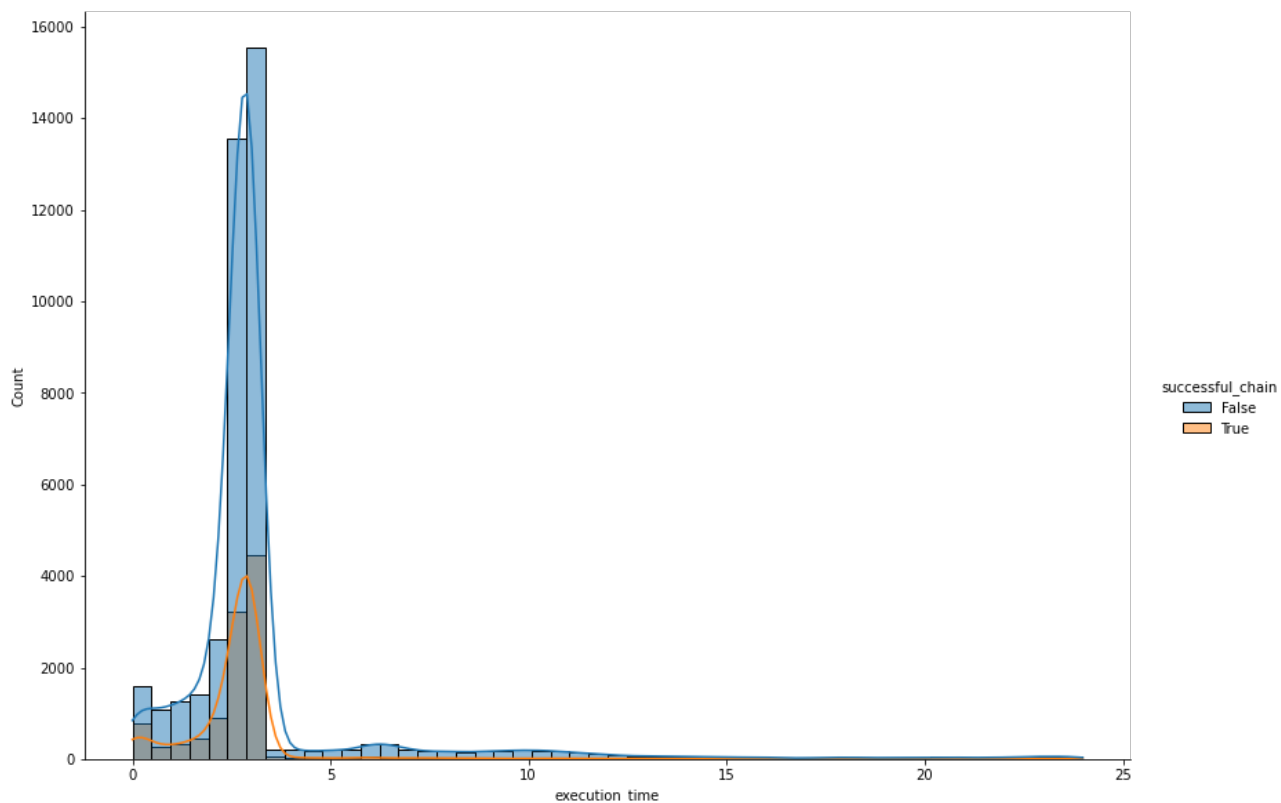
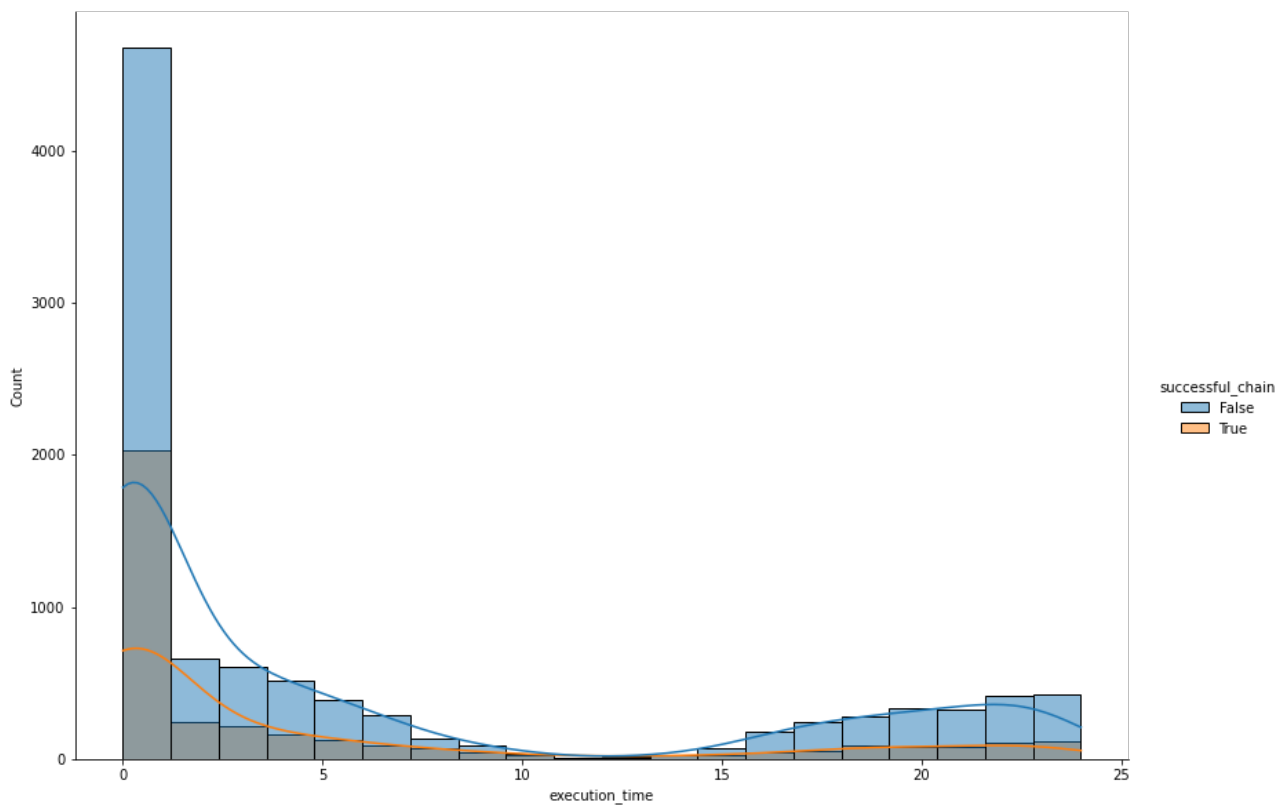


Рисунок 2: Диаграмма Сэнкей 2 Использование новых каналов взаимодействия



Фигура 1: Время обработки события Назначена задача на звонок 1 линии, часов



Фигура 2: Время обработки события Назначена задача на вторую линию, часов

Код отправки отчетов ботом

```
import telebot
from telebot import types

token = "" # Токен убрал
bot=telebot.TeleBot(token)

@bot.message_handler(commands=['button'])
def button_message(message):
    markup=types.ReplyKeyboardMarkup(resize_keyboard=True)
    item1=types.KeyboardButton("Первая линия")
    markup.add(item1)
    item2=types.KeyboardButton("Методисты")
    markup.add(item2)
    item3=types.KeyboardButton("Вторая линия")
    markup.add(item3)
    bot.send_message(message.chat.id,'Какой отчет за сегодня?',reply_markup=markup)
@bot.message_handler(content_types='text')
def message_reply(message):
    if message.text=="Первая линия":
        img = open('plot1.PNG', 'rb')
        bot.send_photo(message.chat.id, img)
    if message.text=="Методисты":
        img = open('plot2.PNG', 'rb')
        bot.send_photo(message.chat.id, img)
    if message.text=="Вторая линия":
        img = open('plot3.PNG', 'rb')
        bot.send_photo(message.chat.id, img)

bot.polling()
```