

Кейс для бизнес-аналитиков. Т-Банк.

Оглавление

1. Цель проекта и метрики эффективности	1
1.1 Цель проекта:	1
1.2. Метрики для оценки эффективности	2
1.3. Используемые инструменты	2
2. Описание пользователя (Персона).....	2
3. Функциональные требования (User Stories).....	3
4. Нефункциональные требования	4
5. Моделирование бизнес-процесса (схема работы)	4
6. Схема перехода между статусами задачи.....	9
7. Прототипирование интерфейса (описание Wireframes).....	13
Экран 1: Главный экран планировщика (рисунок 4)	14
Экран 2: Детали задачи (рисунок 5).....	16
Процесс создания прототипа в Figma.....	18
8. Приоритизация и дорожная карта (Roadmap).....	19
Этап 1: MVP (Must-have)	19
Этап 2: Release (Should-have)	19
Этап 3: Backlog / Future improvements (Could-have).....	19
9. Итоги и ожидания по проекту «Планировщик задач».....	20
Что мы ожидаем в ближайшей перспективе?	20
К чему мы придем в итоге?.....	20

1. Цель проекта и метрики эффективности

Ключевая гипотеза: Интеграция планировщика задач в экосистему Т-Банка увеличит использование внутренних сервисов, повысит лояльность пользователей и создаст новый источник выручки.

1.1 Цель проекта: Создать и внедрить в мобильное приложение Т-Банка сервис "Планировщик задач", который поможет пользователям управлять повседневными делами (доставка, бронирование, покупки) и предложит для их выполнения релевантные сервисы из каталога Т-Банка с вознаграждением в виде кэшбека 5%.

Бизнес-цели:

1. Увеличение вовлеченности (Engagement): Повысить частоту и продолжительность сессий в приложении, сделав его ежедневным инструментом для пользователя.
2. Рост транзакционной активности (Cross-sell): Увеличить количество покупок через внутренние сервисы Т-Банка.
3. Повышение лояльности (Retention): Укрепить "экосистемный" эффект, снизив отток пользователей к конкурентам за счет уникальной и полезной функциональности.
4. Привлечение новой аудитории: Использовать новую функцию как конкурентное преимущество в маркетинговых коммуникациях.

1.2. Метрики для оценки эффективности

- **1. Количественные:**

- **Adoption Rate:** % активных пользователей (MAU), которые создали хотя бы одну задачу в планировщике за месяц. **Цель:** >15% MAU через 3 месяца после запуска.
- **Feature Stickiness:** Соотношение DAU/MAU для функциональности планировщика. Показывает, насколько регулярно пользователи возвращаются к фиче.
- **Task-to-Service Conversion Rate:** % задач, которые привели к покупке через предложенный сервис Т-Банка. **Цель:** >10%.
- **Attributable Revenue (Дополнительная выручка):** Объем выручки, сгенерированный через сервисы, предложенные в планировщике.
- **ARPU (Average Revenue Per User):** Увеличение среднего дохода на пользователя за счет использования сервисов через планировщик.

- **2. Качественные:**

- **Customer Satisfaction (CSAT/NPS):** Оценка удовлетворенности новой функцией через опросы в приложении. **Цель:** CSAT > 4.5/5.
- **User Feedback Analysis:** Анализ отзывов в App Store/Google Play и обращений в поддержку, касающихся планировщика.
- **Usability Testing Score:** Оценка простоты использования на основе юзабилити-тестов (например, по шкале SUS).

1.3. Используемые инструменты

1. **Draw.io** - сервис для создания диаграмм и схем.
2. **Figma** - онлайн-сервис для разработки интерфейсов и прототипирования.
3. **Google Docs** - онлайн-сервис для работы с текстовыми документами.

Итоговый файл предоставлен на проверку в формате **PDF**.

2. Описание пользователя (Персона)

На начальном этапе мы сфокусируемся на одном ключевом сегменте пользователей.

Пользователь: "Занятый житель мегаполиса"

- **О пользователе:** Живет в крупном городе, работает в режиме многозадачности. Ценит свое время и комфорт. Активно пользуется

смартфоном для решения бытовых и рабочих вопросов. Является клиентом Т-Банка.

- **Цели:**

- Эффективно управлять личным временем и не забывать о важных делах (как рутинных, так и развлекательных).
- Находить выгодные предложения и экономить деньги (кэшбэк, скидки).
- Упростить рутину, используя одно приложение для разных задач вместо пяти.

- **Проблемы и «боли»:**

- «У меня в голове или в разных приложениях (календарь, заметки, напоминания) хранится куча дел: записаться к врачу, заказать продукты на ужин, купить билеты в кино на выходные, не забыть заправить машину».
- «Я часто забываю о мелких, но важных задачах».
- «Поиск лучших предложений для каждой задачи отнимает много времени. Нужно открыть одно приложение для билетов, другое для ресторанов, третье для доставки».
- «Я бы хотела, чтобы мой банк не просто хранил деньги, а помогал мне их тратить с умом и удобством».

3. Функциональные требования (User Stories)

- **Базовое управление задачами:**

- Как пользователь, я хочу **создавать новую задачу** с названием, описанием, датой и временем, чтобы зафиксировать свои планы.
- Как пользователь, я хочу **просматривать список** своих задач (на сегодня, на неделю, всех), чтобы видеть общую картину.
- Как пользователь, я хочу **редактировать** существующую задачу (изменить время, описание), чтобы актуализировать информацию.
- Как пользователь, я хочу **удалять** задачу, если она стала неактуальной.
- Как пользователь, я хочу **отмечать задачу как выполненную**, чтобы визуально отслеживать свой прогресс.

- **Интеграция с сервисами Т-Банка:**

- Как пользователь, когда я создаю задачу (например, "ужин с друзьями"), я хочу, чтобы система **предложила мне релевантные сервисы** (например, "Забронировать столик в ресторане"), чтобы я могла быстро решить свою задачу.
- Как пользователь, я хочу видеть **информацию о кэшбэке 5% на карточке предложенного сервиса**, чтобы это мотивировало меня воспользоваться им.

- Как пользователь, после, например, покупки билетов в театр через предложенный сервис, я хочу, чтобы **электронный билет автоматически прикрепился к задаче**, чтобы его можно было легко найти.
- Как пользователь, после выполнения задачи через сервис Т-Банка, я хочу, чтобы **задача автоматически меняла статус на "Выполнена"**, а я получал уведомление о начисленном кэшбэке.
- **Дополнительные возможности (User Experience):**
 - Как пользователь, я хочу получать **push-уведомления** о предстоящих задачах, чтобы ничего не забыть.
 - Как пользователь, я хочу иметь возможность **поделиться задачей** (например, "Купить билеты в кино") с другим пользователем Т-Банка (например, с мужем) через мессенджер.
 - Как пользователь, я хочу **добавлять важные задачи в "Избранное"**, чтобы они всегда были наверху списка.

4. Нефункциональные требования

- **Производительность:**
 - Время отклика интерфейса (открытие списка задач, создание задачи) не должно превышать 1.5 секунды.
- **Надежность:**
 - Доступность сервиса должна составлять не менее 99.9%.
- **Безопасность:**
 - Все данные пользователя (содержание задач, планы) должны храниться в зашифрованном виде и соответствовать стандартам безопасности банка.
- **Масштабируемость:**
 - Архитектура должна выдерживать нагрузку до 1 млн активных пользователей в месяц без деградации производительности.
- **Совместимость:**
 - Корректное отображение и работа на 2-х последних мажорных версиях iOS и Android.

5. Моделирование бизнес-процесса (схема работы)

Процесс "Создание задачи и выполнение через сервис Т-Банка". Используется нотация BPMN, также представлена текстовая схема.

Участники: Пользователь, Система Т-Банка.

1. **Пользователь:** Открывает раздел "Планировщик задач" в приложении Т-Банка.
2. **Пользователь:** Нажимает "Создать задачу".

3. **3. Пользователь:** Вводит данные: "Купить продукты на неделю", устанавливает дату.
4. **Система:** Сохраняет задачу. На основе ключевых слов ("продукты") и контекста анализирует задачу.
5. **Система:** В деталях задачи отображает блок "Решения от Т-Банка" с предложением "Заказать доставку из ВкусВилл" (сервис-партнер). Указывает выгоду: "Кэшбэк 5%".
6. **Пользователь:** Нажимает на предложение.
7. **Система:** Перенаправляет пользователя в бесшовный интерфейс (web-view или mini-app) сервиса "ВкусВилл" внутри приложения Т-Банка.
8. **Пользователь:** Совершает заказ и оплачивает его картой Т-Банка.
9. **Система:** Получает от сервиса подтверждение об успешной оплате.
10. **Система:** Автоматически меняет статус задачи на "Выполнена".
11. **Система:** Начисляет кэшбэк 5% и отправляет пользователю push-уведомление: "Задача 'Купить продукты' выполнена! Вам начислен кэшбэк 150 ₽".

Улучшение и детализация процесса

Прежде чем строить схему, немного доработаем исходный текстовый сценарий. Он описывает только "счастливый путь" (happy path). В реальности процесс сложнее. Добавим два ключевых момента, которые сделают схему более полной и реалистичной:

1. **Ветвление по результату анализа задачи.** Что, если система не смогла найти релевантный сервис? Например, для задачи "поздравить бабушку с днем рождения".
2. **Ветвление по выбору пользователя.** Что, если пользователь увидел предложение от Т-Банка, но решил им не пользоваться, а просто отметить задачу выполненной вручную?
3. **Ветвление по результату оплаты:** Что, если оплата в сервисе партнера не прошла?
4. **Ветвление по бездействию:** Что, если пользователь перешел в сервис партнера, но ничего не сделал и просто закрыл его? Нужно обработать это "зависание" задачи. Для этого идеально подходит промежуточное событие-таймер (Timer Event).

Эти дополнения превратят линейный процесс в полноценную бизнес-модель с альтернативными сценариями.

Диаграмма создана в сервисе draw.io с использованием нотации BPMN и представлена на **рисунке 1** ниже.

Диаграмма состоит из двух вертикальных дорожек "Пользователь" и "Система Т-Банка". Поток двигается слева направо, несколько раз пересекая границу между дорожками и ветвясь в ключевых точках принятия решений. Существует три возможных финала (End Event): успешное завершение, завершение без предложения от банка и прерывание из-за бездействия.

Шаг 1: Создание Пулов и Дорожек (Pools and Lanes).

1. **Создание Пула (Pool):** Даем ему название "Процесс: От создания задачи до выполнения".

2. **Настройка Дорожек (Lanes):** На диаграмме представлены две дорожки: "Пользователь" и "Система Т-Банка".

Теперь у нас есть основа для диаграммы с четким разделением ответственности.

Шаг 2: Отрисовка основного сценария ("Happy Path").

Начнем с отрисовки нашего исходного процесса.

1. **Начало:** На дорожке "Пользователь" разместили Start Event (тонкий зеленый кружок). Дали ему название - "Открыл планировщик".
2. **Задачи пользователя:**
 - От стартового события провели стрелку к элементу User Task (прямоугольник с иконкой человечка). Дали ему название - "Иницирует создание задачи".
 - Добавили следующую User Task с названием "Вводит данные задачи". Соединили их стрелкой.
3. **Задачи системы:**
 - Теперь процесс переходит на дорожку "Система Т-Банка". Добавили Service Task (прямоугольник с шестеренками). Дали ему название - "Сохраняет и анализирует задачу". Соединили стрелкой "Вводит данные задачи" с этой задачей.

Шаг 3: Добавление альтернативных сценариев (Gateways).

Здесь начинается самое интересное — делаем схему умной.

1. Ветвление № 1: "Сервис найден?"

- После задачи "Сохраняет и анализирует задачу" добавили Exclusive Gateway (ромб со знаком "X"). Подписали его вопросом: **"Релевантный сервис найден?"**.
- От ромба провели две стрелки:
 - **Ветка "Да":** Стрелка ведет к Service Task на дорожке системы: **"Формирует и показывает предложение"**.
 - **Ветка "Нет":** Стрелка ведет к End Event (жирный красный кружок). Подписали End Event — **"Задача создана без предложения"**.

2. Ветвление № 2: "Пользователь воспользуется предложением?"

- После задачи "Формирует и показывает предложение" поток возвращается к пользователю. Добавили на дорожке "Пользователь" еще один Exclusive Gateway. Подписали его: **"Предложение принято?"**.
- От ромба так же провели две стрелки:
 - **Ветка "Да":** Стрелка ведет к User Task: **"Выбирает предложение сервиса"**.
 - **Ветка "Нет":** Стрелка ведет к End Event. Подписали End Event — **"Задача ожидает ручного выполнения"**.

3. Ветвление № 3: "Ожидание оплаты или бездействие".

- После того как пользователь выбрал предложение, система его перенаправляет. Добавили на дорожке **"Система Т-Банка"** Service Task: **"Перенаправляет в интерфейс партнера"**.
- Теперь нам нужно подождать действия от пользователя. Для этого используем Event-Based Gateway (ромб с кругом и пентаграммой внутри). Он означает "ждать одного из нескольких следующих событий".
- От этого шлюза провели две стрелки:
 - **К событию оплаты:** Стрелка ведет к Intermediate Event (двойной кружок с иконкой конверта) с подписью **"Получено подтверждение оплаты"**. Это событие-сообщение (Message Event).
 - **К событию-таймеру:** Стрелка ведет к Intermediate Event (двойной кружок с иконкой часов) с подписью **"Истекло время ожидания (15 мин)"**. Это событие-таймер (Timer Event).
- От **события-таймера** провели стрелку к End Event с подписью **"Процесс прерван (бездействие)"**.
- От **события-сообщения об оплате** продолжаем наш "счастливый путь".

Шаг 4: Завершение процесса.

1. После получения подтверждения оплаты система выполняет несколько действий. Используем **Параллельный Шлюз (Parallel Gateway, ромб с "+")**, чтобы показать, что действия происходят одновременно.

- Добавили на дорожке **"Система Т-Банка"** Parallel Gateway.
- От него провели три стрелки к трем разным задачам:
 1. Service Task: **"Изменяет статус задачи на 'Выполнена'"**.
 2. Service Task: **"Начисляет кэшбэк 5%"**.
 3. Send Task (прямоугольник с закрашенным конвертом): **"Отправляет push-уведомление"**.
- После этих трех задач "собрали" потоки обратно с помощью еще одного Parallel Gateway.

2. **Конец процесса:** От собирающего параллельного шлюза провели стрелку к End Event с подписью **"Процесс успешно завершен"**.

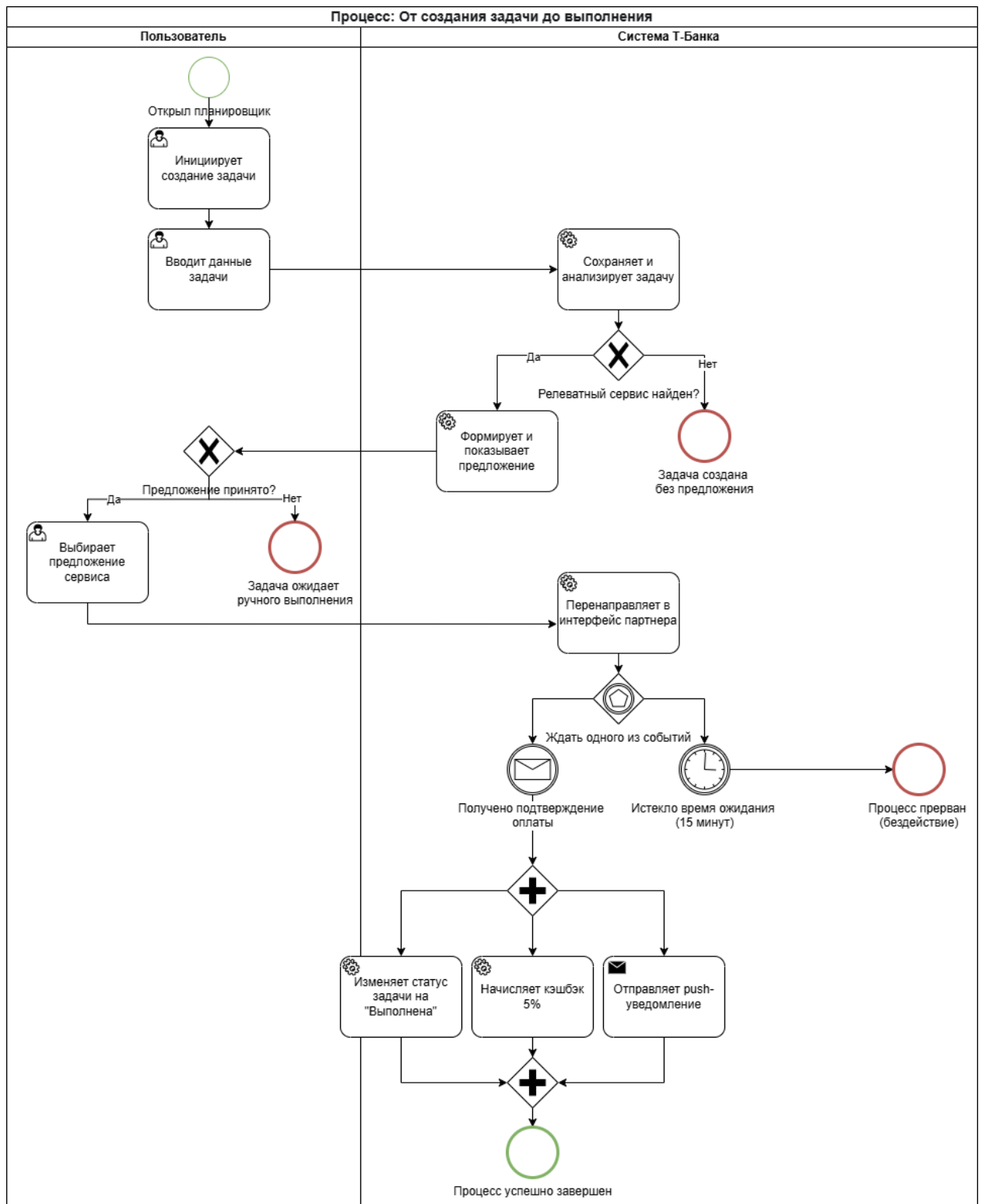


Рисунок 1 - Диаграмма-схема моделирования бизнес-процесса

Пояснения к элементам и логике схемы:

- **Пул/Дорожка**
 - Контейнеры "Пользователь" и "Система Т-Банка". Четко разделяют, кто за какое действие отвечает. Это устраняет двусмысленность.
- **Start/End Event**

- Зеленый и красные круги. Обозначают начало и все возможные исходы процесса. У нас несколько финалов, что отражает реальность.
- **User/Service Task**
 - Прямоугольники. Обозначают конкретные действия, выполняемые либо человеком, либо автоматизированной системой.
- **Exclusive Gateway (X)**
 - Ромбы с "X". **Ключевые точки принятия решений (логика "ЕСЛИ-ТО-ИНАЧЕ")**. Система решает, есть ли подходящий сервис. Пользователь решает, хочет ли он воспользоваться предложением.
- **Event-Based Gateway**
 - Ромб с пентаграммой. **Точка ожидания**. Система не продолжает процесс, пока не случится одно из двух событий: либо придет подтверждение оплаты, либо истечет время. Это решает проблему "зависших" процессов.
- **Timer Event**
 - Круг с часами. **Обработка бездействия**. Если пользователь ушел в сервис и ничего не сделал за 15 минут, процесс автоматически завершается, не оставаясь в подвешенном состоянии.
- **Parallel Gateway (+)**
 - Ромб с "+". **Одновременные действия (логика "И")**. После успешной оплаты система одновременно меняет статус, начисляет кэшбэк и шлет уведомление. Это эффективнее, чем делать их по очереди.
- **Send Task**
 - Прямоугольник с темным конвертом. Специальный тип задачи для отправки информации вовне — в нашем случае, отправка Push-уведомления на устройство пользователя.

6. Схема перехода между статусами задачи

Задачи имеют жизненный цикл, который помогает пользователю отслеживать их состояние.

- **Новая (To Do)**: Начальный статус после создания задачи.
- **В работе (In Progress)**: Пользователь может вручную перевести задачу в этот статус или система переводит ее автоматически, когда пользователь переходит в предложенный сервис.
- **Выполнена (Done)**: Конечный успешный статус. Устанавливается вручную пользователем или автоматически после выполнения действия через сервис Т-Банка.
- **Отменена (Cancelled)**: Конечный неуспешный статус. Устанавливается пользователем вручную.

- **Просрочена (Overdue):** Статус присваивается системой, если дата и время выполнения задачи прошли, а статус не был изменен на "Выполнена" или "Отменена".

Дополнения и улучшения схемы

Добавим несколько логичных переходов, которые могут быть упущены:

- **Из "Просрочена" в "Выполнена":** Пользователь все же может выполнить задачу, даже если она просрочена.
- **Из "Просрочена" в "Отменена":** Если задача просрочена, пользователь может решить ее отменить.
- **Из "Новая" в "Выполнена":** Возможно, пользователь захочет сразу отметить простую задачу выполненной, минуя статус "В работе".

С учетом этих дополнений наша схема будет абсолютно полной.

Диаграмма статуса создана в draw.io с использованием диаграммы состояний и отражает жизненный цикл задачи (**рисунок 2**).

Диаграмма состояний (State Machine Diagram) создана для описания **жизненного цикла одного объекта**. Она показывает, в каких состояниях (статусах) может находиться объект и какие события (триггеры) вызывают переход из одного состояния в другое. Она отвечает на вопрос: *«Как меняется статус задачи с течением времени?»*.

Шаг 1: Обозначение начала и конца жизненного цикла.

1. **Начало:** Создадим Эллипс (Ellipse). Это начальная точка (Initial State).
2. **Конец:** Нарисуем еще один круг. В панели стилей уберем заливку и сделаем рамку толще. Это конечная точка (Final State). Нам понадобится две таких точки, так как у нас два конечных статуса: "Выполнена" и "Отменена".

Шаг 2: Создание состояний (статусов).

1. Создадим Прямоугольник со скругленными углами (Rounded Rectangle). Это классическая фигура для обозначения состояния.
2. Создадим три таких прямоугольника для наших промежуточных состояний. Дадим им названия:
 - Новая (To Do).
 - В работе (In Progress).
 - Просрочена (Overdue).
3. Для конечных статусов создадим еще два прямоугольника. Они будут терминальными, но мы все равно их покажем как состояния, к которым ведут стрелки перед конечной точкой.
 - Выполнена (Done).
 - Отменена (Cancelled).

Шаг 3: Соединение состояний (переходы).

Это самый важный шаг. Мы будем использовать стрелки, чтобы показать переходы, и подписывать их **событием (триггером)**, которое вызывает этот переход.

1. **От начала к "Новой"**: Проведем стрелку от черного круга к состоянию Новая (To Do) и подпишем ее: **[Задача создана]**.

2. **Из "Новая"**:

- Проведем стрелку от Новая к В работе и подпишем ее: **[Пользователь начал работу] / [Система предложила сервис]**.
- Проведем стрелку от Новая к Отменена и подпишем: **[Пользователь отменил]**.
- Проведем стрелку от Новая к Выполнена и подпишем: **[Пользователь выполнил]**.

3. **Из "В работе"**:

- Проведем стрелку от В работе к Выполнена и подпишем: **[Пользователь выполнил] / [Система подтвердила оплату]**.
- Проведем стрелку от В работе к Отменена и подпишем: **[Пользователь отменил]**.

4. **Обработка просрочки (системное событие)**:

- Проведем стрелку от Новая к Просрочена и подпишем: **[Система: наступило время выполнения]**.
- Проведем стрелку от В работе к Просрочена и подпишем: **[Система: наступило время выполнения]**.

5. **Из "Просрочена" (добавленная нами логика)**:

- Проведем стрелку от Просрочена к Выполнена и подпишем: **[Пользователь выполнил]**.
- Проведем стрелку от Просрочена к Отменена и подпишем: **[Пользователь отменил]**.

6. **Завершение жизненного цикла**:

- От состояния Выполнена проведем стрелку к одной из конечных точек (пустой круг с жирной рамкой).
- От состояния Отменена проведем стрелку ко второй конечной точке.

Шаг 4: Визуальное оформление

- Состояния «Новая», «В работе» обозначены нейтральным синим цветом.
- Состояние «Просрочена» обозначено оранжевым (желтым цветом).
- Состояние «Выполнена» обозначено зеленым цветом.
- Состояние «Отменена» обозначено красным цветом.

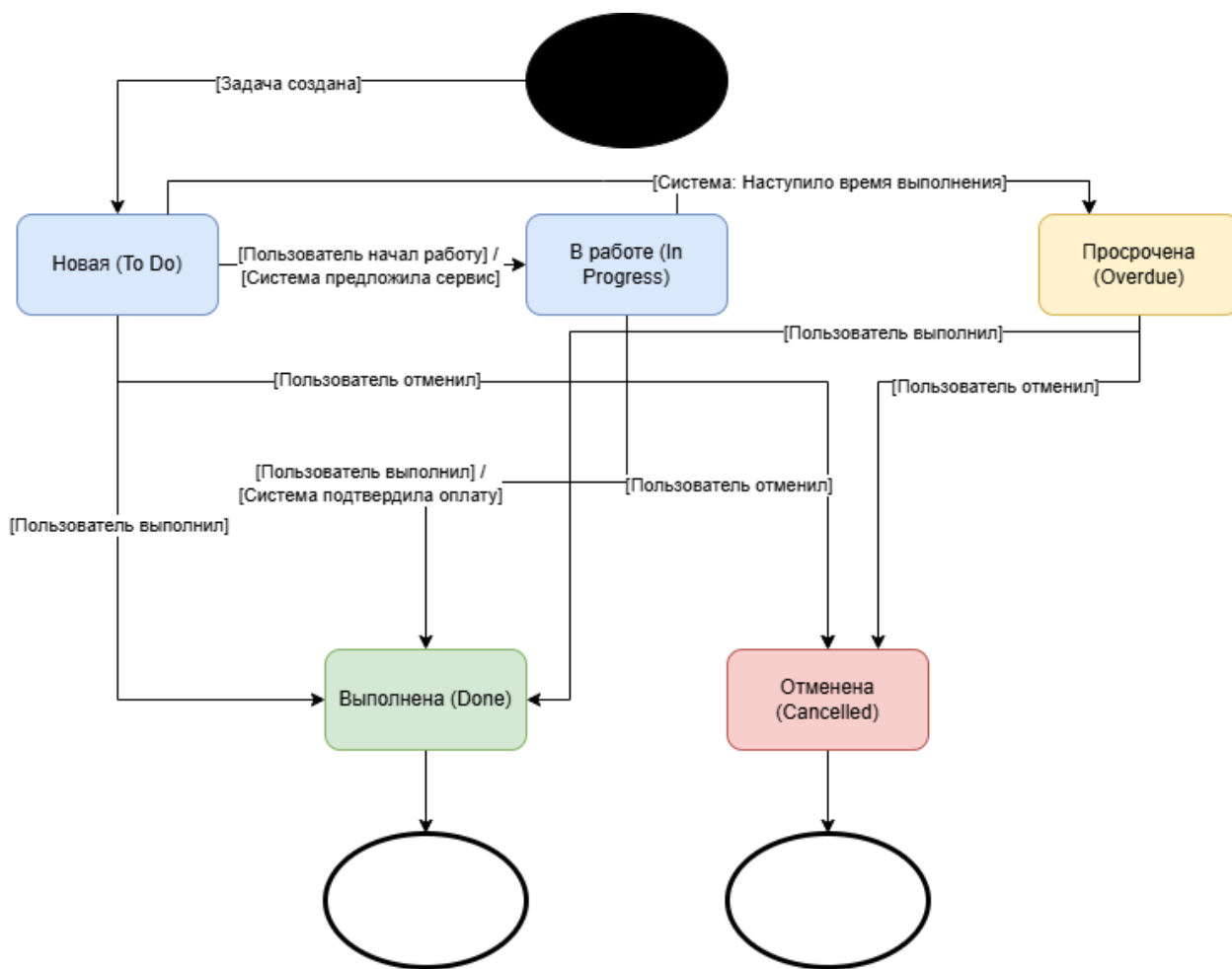


Рисунок 2 - Диаграмма статусов задачи

Пояснения к схеме:

В центре диаграммы расположены три основных прямоугольника: Новая, В работе, Просрочена. Сверху находится черная точка начала, от которой стрелка ведет к статусу Новая. Внизу диаграммы расположены два зеленых и красных прямоугольника Выполнена и Отменена, от которых стрелки ведут к конечным точкам (кругам с жирной рамкой). Все прямоугольники соединены стрелками, каждая из которых подписана событием-триггером в квадратных скобках.

Элементы диаграммы и описания к ним:

- **Состояния (прямоугольники).** Это возможные статусы, в которых может находиться объект "Задача" в любой момент времени.
- **Начальное состояние (черный круг).** Точка, с которой начинается жизненный цикл любой задачи.
- **Конечное состояние (круг с рамкой).** Точка, в которой жизненный цикл задачи считается завершенным. У нас их две, так как есть два финальных статуса.
- **Переходы (стрелки).** Показывают, из какого состояния в какое может перейти задача.

- **События/Триггеры ([текст на стрелках]).** Условие или действие (пользователя или системы), которое инициирует переход из одного состояния в другое.

Таблица 1 - Состояния задач и переходы

Статус	Описание	Возможные переходы
Новая (To Do)	Задача создана, но работа над ней еще не начата.	→ В работе → Выполнена → Отменена → Просрочена
В работе (In Progress)	Пользователь или система инициировали выполнение задачи.	→ Выполнена → Отменена → Просрочена
Выполнена (Done)	Задача успешно завершена. Конечный успешный статус.	→ (Конец)
Отменена (Cancelled)	Задача была отменена пользователем. Конечный неуспешный статус.	→ (Конец)
Просрочена (Overdue)	Время выполнения задачи истекло, но она не была выполнена или отменена.	→ Выполнена → Отменена

Эта диаграмма состояний идеально визуализирует всю логику смены статусов, легко читается разработчиками и тестировщиками, и является стандартом для описания подобных жизненных циклов.

7. Прототипирование интерфейса (описание Wireframes)

Ниже представлено текстовое описание ключевых экранов. Визуальные макеты подготовлены в Figma.

Интерактивный прототип, созданный в Figma, можно посмотреть, кликнув на QR-код или на изображения – рисунок 4 и рисунок 5.



Рисунок 3 - QR-код для просмотра интерактивного прототипа в Figma

Экран 1: Главный экран планировщика (рисунок 4)

- **Заголовок:** "Мои задачи".
- **Табы/фильтры:** "Сегодня", "Неделя", "Все", "Избранное".
- **Список задач:** Каждая задача - это карточка, содержащая:
 - Чек-бокс для быстрого выполнения.
 - Название задачи.
 - Дата и время.
 - Цветовой индикатор статуса (например, серый - новая, зеленый - выполнена, красный - просрочена).
- **Плавающая кнопка (FAB):** Большой "+" в правом нижнем углу для создания новой задачи.

Мои задачи

Сегодня

Неделя

Все

Избранное



Название задачи

Сегодня, 18:00



Название задачи

Сегодня, 18:00



Название задачи

Сегодня, 18:00



Название задачи

Сегодня, 18:00

Рисунок 4 - Главный экран планировщика. Чтобы открыть интерактивный прототип, кликните на изображение выше

Экран 2: Детали задачи (рисунок 5)

- **Заголовок:** Название задачи.
- **Блок информации:**
 - Полное описание.
 - Дата, время.
 - Статус (можно изменить).
- **Блок "Решения от Т-Банка":**
 - Яркий, заметный блок.
 - Горизонтально прокручиваемые карточки с предложениями.
 - **Пример карточки:** Иконка сервиса "Театры", заголовок "Купить билеты", подзаголовок "Вернем 5% кэшбэком", кнопка "Выбрать билеты".
- **Нижнее меню:** "Редактировать", "Поделиться", "Удалить".

← Заправить машину

Заехать на любую заправку по дороге домой,
залить полный бак АИ-95

Дата: Сегодня, 18:00

Статус: Новая

Решения от Т-Банка



Заправиться на АЗС

Вернем 5% кэшбэком

Выбрать АЗС



Поделиться



Удалить



Редактировать

Рисунок 5 - Экран "Детали задачи". Чтобы открыть интерактивный прототип, кликните на изображение выше

Процесс создания прототипа в Figma

Для создания интерактивного прототипа интерфейса планировщика задач был применен системный и компонентный подход, основанный на дизайн-системе Т-Банка.

1. Создание основы (Design System)

В первую очередь была заложена основа дизайн-системы:

- **Стили:** Настроены и добавлены в локальные стили фирменные цвета (желтый, темный фон, статусы) и текстовые стили (заголовки, основной текст, подписи). Это обеспечило целостность элементов и ускорило дальнейшую работу.

2. Компонентный подход

Вместо прямого рисования на экранах все ключевые элементы интерфейса были созданы как компоненты, которые можно использовать несколько раз:

- **Карточка задачи (Task Card):** Создана как компонент с **вариантами (Variants)** для отображения разных состояний (новая, выполнена, просрочена).
- **Плавающая кнопка (FAB):** Сделана компонентом для быстрого добавления на любой экран.
- **Карточка предложения (Offer Card):** Разработан отдельный компонент для блока "Решения от Т-Банка".
- Все компоненты были построены на **Auto Layout**, что обеспечило их адаптивность и легкое редактирование.

3. Сборка экранов

Экраны собирались как конструктор из готовых компонентов:

- **Экран 1: Главный экран планировщика**
 - Создан основной фрейм, внутри которого контент (заголовок, табы, список задач) организован с помощью **вертикального Auto Layout** для управления отступами.
 - Список задач собран из нескольких **экземпляров (instances)** компонента Task Card.
 - Плавающая кнопка FAB была вынесена из основного потока Auto Layout и спозиционирована с помощью **привязок (Constraints: Right, Bottom)** и опции **Fix position when scrolling**, чтобы она оставалась на месте при прокрутке.
- **Экран 2: Детали задачи**
 - **Горизонтальный скролл** для блока "Решения от Т-Банка" был реализован путем помещения горизонтального Auto Layout с карточками внутрь фрейма с включенной опцией **Clip Content** (обрезать содержимое) и настройкой **Horizontal scrolling** в режиме прототипирования.
 - **Фиксированное нижнее меню** было создано как отдельный компонент и, по аналогии с FAB, размещено вне основного потока

контента и "приклеено" к низу экрана с помощью **Constraints (Bottom, Left & Right)**.

4. Прототипирование

В режиме **Prototype** были созданы интерактивные связи между экранами:

- Нажатие на карточку задачи (On tap) ведет на экран деталей с плавной анимацией Move In.
- Нажатие на иконку "назад" возвращает на главный экран с обратной анимацией Move Out.

В результате был получен не просто набор статичных картинок, а целостный, интерактивный прототип, демонстрирующий ключевые пользовательские сценарии и готовый для обсуждения с командой и передачи в разработку.

8. Приоритизация и дорожная карта (Roadmap)

Мы будем использовать подход MoSCoW для определения последовательности реализации.

Этап 1: MVP (Must-have)

- **Цель:** Запустить базовую, но ценную для пользователя версию.
- **Функциональность:**
 - Полный цикл управления задачами.
 - Простая статусная модель: "Новая", "Выполнена".
 - Интеграция с 2-3 ключевыми сервисами (например, Рестораны, Билеты в кино).
 - Отображение информации о кэшбэке и его начисление.
 - Базовые push-уведомления о предстоящих задачах.

Этап 2: Release (Should-have)

- **Цель:** Расширить возможности и улучшить пользовательский опыт.
- **Функциональность:**
 - Расширенная статусная модель ("В работе", "Отменена", "Просрочена").
 - Автоматическое прикрепление документов (билеты, ваучеры).
 - Автоматическое изменение статуса задачи.
 - Расширение каталога интегрированных сервисов (+ Доставка продуктов, + Заправки).
 - Возможность поделиться задачей.
 - Фильтрация и сортировка задач.

Этап 3: Backlog / Future improvements (Could-have)

- **Цель:** Добавить "приятные" фишки для повышения лояльности.

- **Функциональность:**

- Функция "Избранное".
- Категоризация задач (теги: #дом, #работа, #отдых).
- Повторяющиеся задачи (например, "Оплатить интернет" каждый месяц).
- Интеллектуальное распознавание сущностей из текста для автозаполнения полей (NLP).
- Геймификация (ачивки за выполнение N задач).

Следующие шаги:

1. Обсуждение документа с командой и владельцем продукта для сбора обратной связи.
2. Передача документа в отдел дизайна для подготовки детальных макетов (UI/UX).
3. Декомпозиция User Stories из MVP на технические задачи для бэклога разработки.

9. Итоги и ожидания по проекту «Планировщик задач»

Перед нами стоит амбициозная, но очень интересная задача — создать не просто новый функционал, а по-настоящему полезный инструмент, который изменит ежедневные привычки наших пользователей и укрепит их связь с экосистемой Т-Банка.

Что мы ожидаем в ближайшей перспективе?

Наш фокус — быстрый запуск **MVP (минимально жизнеспособного продукта)**. Это означает, что в первую очередь мы сконцентрируемся на самом важном:

1. **Создадим ядро:** Пользователь должен иметь возможность легко создавать, редактировать и выполнять задачи.
2. **Докажем ценность:** Интегрируем 2-3 ключевых сервиса (например, Рестораны, Кино), чтобы продемонстрировать главную идею: «запланировал → получил релевантное предложение → воспользовался с выгодой».
3. **Обеспечим выгоду:** Реализуем механику начисления кэшбэка 5% — это наш ключевой мотиватор для пользователя.

Мы ожидаем, что команда слаженно проработает этот базовый сценарий, чтобы как можно скорее получить первые реальные данные и отзывы от пользователей.

К чему мы придем в итоге?

Наша финальная цель — создать **умного помощника**, органично встроенного в приложение Т-Банка, который:

- **Для пользователя:** Превратит рутинное планирование из головной боли в удобный и выгодный процесс. Ему больше не нужно будет переключаться между десятком приложений — все можно будет решить в одном месте, сэкономив время и деньги.

- **Для бизнеса:** Станет мощным инструментом удержания и вовлечения. Мы не просто дадим пользователю новый «таск-менеджер», а создадим еще один весомый повод оставаться в нашей экосистеме, чаще пользоваться нашими сервисами и воспринимать Т-Банк как "life-style" партнера.
- **Для нас, как команды:** Мы создадим уникальную для рынка фичу, которая напрямую влияет на ключевые бизнес-метрики и делает жизнь наших клиентов лучше.

Проще говоря, мы начинаем с простого и понятного планировщика, а приходим к интеллектуальному ассистенту, который предугадывает потребности пользователя и помогает ему жить проще и выгоднее.