

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**

**Департамент информационных и компьютерных систем**

Янович Яков Валерьевич

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**бакалаврская работа**

**АВТОМАТИЗАЦИЯ УЧЕТА ЗАДАЧ ПРОЕКТОВ НА ПРИМЕРЕ**

**ООО «ВОСТОК ИТ» Г. ВЛАДИВОСТОК**

по направлению подготовки 09.03.03 Прикладная информатика

Прикладная информатика в экономике

г. Владивосток

2025

|  |  |
| --- | --- |
| В материалах данной выпускной  квалификационной работы не содержатся  сведения, составляющие государственную  тайну, и сведения, подлежащие экспортному  контролю  Уполномоченный по экспортному контролю  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ И. Л. Артемьева  *подпись И.О. Фамилия*  «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. | Автор работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,  *подпись*  группа Б9121-09.03.03пиэ    «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.  Руководитель ВКР старший преподаватель  должность, ученое звание  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. И. Шувалова  *подпись И.О. Фамилия*  «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.  Консультант(ы)\*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *подпись И.О. Фамилия*  «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |
|  | Назначен рецензент\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *ученое звание*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *фамилия, имя, отчество* |
| Защищена в ГЭК с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Секретарь ГЭК  \_\_\_\_\_\_\_\_\_\_\_\_\_ Г. Н. Цыганова  *подпись И.О. Фамилия*  «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. | **«Допустить к защите»**  Директор Департамента информационных и компьютерных систем  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *ученая степень, ученое звание*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Н. Федорец  *подпись И.О. Фамилия*  «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |

Аннотация

Выпускная квалификационная работа посвящена разработке программного средства для автоматизации учета задач проектов ООО «Восток ИТ». Целью работы является повышение эффективности управления проектной деятельностью за счёт автоматического учета задач, интеграции с сервисом Asana и формирования отчетности.

В работе применены методы функционально-ориентированного проектирования, моделирования бизнес-процессов и информационных потоков. Разработанное веб-приложение реализовано на языке. Программное средство снижает трудозатраты, повышает точность отчетов и исключает дублирование данных.

Объем работы – 69 страниц, Рисунок 23 рисунка, Таблица 18 таблиц и 5 приложений.

**Оглавление**

[Аннотация 3](#_Toc200812149)

[Введение 6](#_Toc200812150)

[1 Аналитическая часть 8](#_Toc200812151)

[1.1 Технико-экономическая характеристика ООО «Восток ИТ» 8](#_Toc200812152)

[1.2 Экономическая сущность комплекса экономических информационных задач 12](#_Toc200812153)

[1.3 Обоснование выбора технологии проектирования 13](#_Toc200812154)

[1.3.1 Функционально-ориентированное проектирование 13](#_Toc200812155)

[1.3.2 Объектно-ориентированное проектирование 14](#_Toc200812156)

[1.3.3 Выбор технологии проектирования 15](#_Toc200812157)

[1.4 Обоснование необходимости и цели использования вычислительной техники для решения задачи 15](#_Toc200812158)

[1.4.1 Обоснование автоматизации решения задачи 15](#_Toc200812159)

[1.4.2 Цели и задачи автоматизации 20](#_Toc200812160)

[1.4.3 Анализ существующих разработок и обоснование выбора технологии проектирования 21](#_Toc200812161)

[1.4.4 Предлагаемая модель бизнес-процессов 24](#_Toc200812162)

[1.5 Постановка задачи 26](#_Toc200812163)

[1.5.1 Общая характеристика организации задачи на ЭВМ 26](#_Toc200812164)

[1.5.2 Характеристика выходной информации 27](#_Toc200812165)

[1.5.3 Характеристика входной информации 30](#_Toc200812166)

[2 Проектная часть 33](#_Toc200812167)

[2.1 Календарное планирование 33](#_Toc200812168)

[2.1.1 Содержание проекта 33](#_Toc200812169)

[2.1.2 Оценка длительности 33](#_Toc200812170)

[2.1.3 Планирование ресурсов и расчет затрат 34](#_Toc200812171)

[2.1.4 Оптимизация 35](#_Toc200812172)

[2.1.5 Идентификация рисков и разработка стратегии их смягчения 36](#_Toc200812173)

[2.1.6 Стоимость проекта 38](#_Toc200812174)

[2.2 Информационное обеспечение задачи 39](#_Toc200812175)

[2.2.1 Информационная модель её описание 39](#_Toc200812176)

[2.2.2 Характеристика базы данных 40](#_Toc200812177)

[2.3 Программное обеспечение задачи 44](#_Toc200812178)

[2.3.1 Дерево функций 44](#_Toc200812179)

[2.3.2 Структурная схема 45](#_Toc200812180)

[2.3.3 Алгоритмы модулей 46](#_Toc200812181)

[2.3.4 Общие положения 52](#_Toc200812182)

[2.4 Описание интерфейса системы 53](#_Toc200812183)

[2.4.1 Характеристика входных форм 54](#_Toc200812184)

[2.4.2 Характеристика результатных форм 55](#_Toc200812185)

[2.5 Технологическое обеспечение задачи 58](#_Toc200812186)

[2.6 Обоснование экономической эффективности проекта 59](#_Toc200812187)

[2.6.1 Выбор метода расчета экономической эффективности 59](#_Toc200812188)

[2.6.2 Расчет экономической эффективности 60](#_Toc200812189)

[2.7 Описание контрольного примера реализации проекта 63](#_Toc200812190)

[Заключение 65](#_Toc200812191)

[Список литературы 67](#_Toc200812192)

[Приложение А 70](#_Toc200812193)

Введение

В условиях растущей конкуренции на рынке ИТ-услуг компании сталкиваются с необходимостью повышения эффективности управления проектами и оптимизации внутренних процессов. Это связано с высокой динамикой задач, множественностью одновременно реализуемых проектов и необходимостью гибко распределять ресурсы. Общество с ограниченной ответственностью «Восток ИТ» (далее – ООО «Восток ИТ») использует инструменты управления задачами, однако доля ручной обработки данных остается высокой, что снижает эффективность и увеличивает риск ошибок.

Данная выпускная квалификационная работа посвящена разработке программного средства для автоматизации учета проектных задач, интегрированного с сервисом Asana, который уже используется на предприятии. Актуальность темы обусловлена необходимостью повышения точности анализа задач, исключения дублирования информации и сокращения времени, затрачиваемого на подготовку отчетности. Предлагаемое решение направлено на устранение выявленных проблем за счет автоматизации процессов сбора, хранения и обработки данных о задачах.

Целью работы является разработка программного средства для учета дополнительных задач сотрудников, интегрированного с сервисом Asana, и способного формировать сводные отчеты, хранить историю задач и обеспечивать прозрачность загрузки сотрудников. Для достижения поставленной цели в работе решаются следующие задачи:

* анализ текущих бизнес-процессов и выявление проблем;
* выбор способа решения проблемы;
* построение модели информационной системы;
* разработка архитектуры и алгоритмов программного решения;
* оценка экономической эффективности;
* разработка программного средства.

В качестве методической основы были использованы подходы функционально-ориентированного проектирования, нотации IDEF0 и DFD для моделирования бизнес-процессов и информационных потоков. Программное средство разработано с применением языка программирования Python и ориентировано на использование внутри компании.

Практическая значимость полученных результатов заключается в возможности внедрения разработанного решения в ООО «Восток ИТ», что позволит сократить трудозатраты сотрудников отдела управления проектами, минимизировать количество ошибок в отчетах и повысить качество управления проектной деятельностью компании.

1. Аналитическая часть
   1. Технико-экономическая характеристика ООО «Восток ИТ»

ООО «Первый Бит» основано в Москве в 1997 году. Компания имеет более 100 офисов в 9 странах мира и занимается внедрением систем управления базами данных (СУБД), разработкой и поддержкой корпоративных систем хранения данных, а также внедрением ERP-систем, BI-систем, CRM-систем и других программных продуктов.

Система менеджмента ООО «Первый Бит» сертифицирована на соответствие международному стандарту качества ISO 9001:2015.

ООО «Первый Бит» обладает всеми партнерскими статусами 1С, такими как 1С:Франчайзинг, 1С:Центр Компетенции ERP, 1С:Центр Компетенции по Документообороту и другие, позволяющие оказывать широкий спектр услуг [17].

Во Владивостоке ООО «Первый Бит» представлено дочерней компанией ООО «Восток ИТ».

ООО «Восток ИТ» помогает клиентам в адаптации к актуальным рыночным требованиям за счет разработки, внедрения и сопровождения программных средств на платформе 1С, а также оптимизации бизнес-процессов и цифровизации документооборота. Реализация данных решений способствует минимизации временных затрат на обработку документов и повышению точности учетных операций клиентов.

Организационная структура ООО «Восток ИТ» включает:

* руководитель офиса – планирование и управление развитием;
* администрация – помощь в управлении и продвижении:
  + менеджер по персоналу – подбор и работа с персоналом., подчиняется руководителю офиса;
  + маркетолог – разработка маркетинговых стратегий., подчиняется руководителю офиса;
* отдел проектов – управление и координация выполнения проектов:
  + руководитель отдела проектов – управление работами и контроль их выполнения., подчиняется руководителю офиса;
  + менеджер проектов – участие в реализации и координации проектов., подчиняется руководителю отдела проектов;
  + бизнес-аналитик – проведение анализ клиентов для формирования проекта., подчиняется руководителю отдела проектов;
* отдел продаж – привлечение новых клиентов и поддержка существующих:
  + менеджер по продажам – проведение переговоров с клиентами, заключение договоров., подчиняется руководителю офиса;
* отдел внедрения – разработка, установка и настройка программных средств:
  + руководитель отдела внедрения – управление работами., подчиняется руководителю офиса;
  + инженер по внедрению – поддержка внедрения и обновления программных средств., подчиняется руководителю отдела внедрения;
  + консультант технической поддержки – консультация клиентов и обучение., подчиняется руководителю отдела внедрения;
  + программист – программирование, реализация программных решений., подчиняется руководителю отдела внедрения.

Схема организационной структуры ООО «Восток ИТ» представлена на рисунке Рисунок 1.

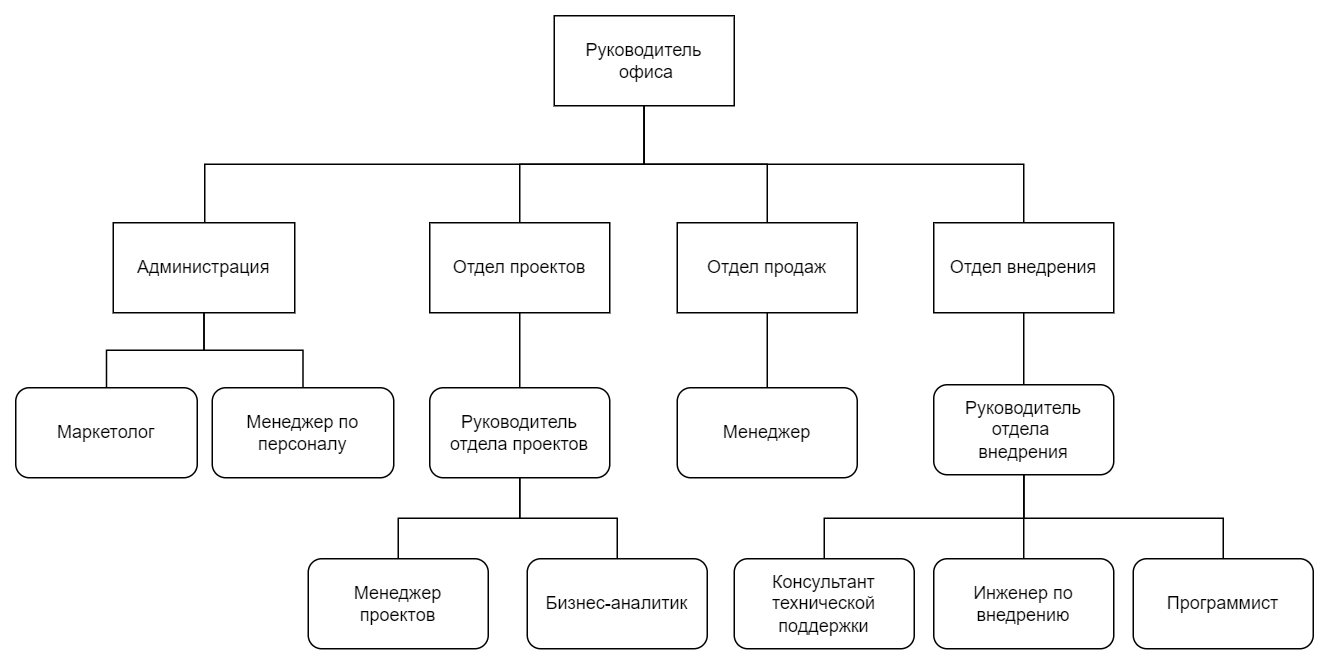


Рисунок 1 – Организационная структура ООО «Восток ИТ»

Этапы основной деятельности ООО «Восток ИТ»:

* работа с клиентами;
* проектирование решений;
* управление проектными задачами;
* разработка и внедрение продуктов 1С;
* сопровождение и поддержка клиентов.

Первым этапом является работа с клиентами. Специалисты собирают требования и пожелания клиентов. Основная задача этого этапа — понять, какие задачи клиент хочет решить с помощью 1С, и заключить договор о предоставлении услуг. В этом процессе используется система К7 – CRM-система, разработанная ООО «Первый Бит». На этом этапе в нее вносятся данные о клиенте, его требованиях и заключенном договоре.

Следующим этапом является проектирования решений. На основании собранных требований и анализа бизнес-процессов клиента разрабатывается архитектура будущих программных средств, составляется проектная документация, определяются конфигурации 1С, которые будут использоваться, и планируются возможные интеграции с внешними системами клиента. Дополнительная информация, полученная в ходе анализа его бизнес-процессов, а также составленная проектная документация вносится в систему К7.

Третий важный этап — управление проектными задачами. Руководители проектов формируют проектные команды, распределяют задачи, контролируют сроки исполнения и координируют работу команды. Важной задачей этого этапа является учет нагрузки на отдел разработки, особенно в случае скрытых задач — проблем и требований, которые не были явно озвучены клиентом или выявлены в ходе бизнес-анализа, но необходимы для полноценного внедрения программного средства. В этом процессе используются следующие системы:

* Asana – платформа для управления проектами на основе канбан досок., используется для распределения и описания задач для сотрудников отдела внедрения;
* Яндекс Формы – сервис, предоставляющий возможность создания и распространения интерактивных форм, используемый для ввода и хранения информации о дополнительных задачах, обнаруженных отделом внедрения., сервис предоставляет возможность экспорта данных в виде электронных таблиц;
* Microsoft Excel – редактор электронных таблиц, используемый для обработки экспортированных из Яндекс Форм данных о дополнительных задачах., в Excel создаются и настраиваются сводные таблицы, дополнительные поля с формулами и визуальные элементы для анализа данных;
* К7 – полученные при анализе данные вносятся в CRM-систему для дальнейшего использования при заключении будущих договоров или корректировки существующих.

После этого команда разработчиков приступает к реализации. Они программируют и настраивают модули и конфигурацию 1С в соответствии с проектом. Проводится тестирование модулей и конфигураций и их внедрение в инфраструктуру клиента. В ходе работы программисты сообщают отделу управления проектами о дополнительных задачах проекта, которые могут быть обнаружены в процессе разработки. В процессе используется Asana для получения задач и отметки о выполнении.

Заключительным этапом является сопровождение и поддержка. ООО «Восток ИТ» предоставляет техническую поддержку, устанавливает обновления, создает инструкции по работе с программным средством, проводит консультации и обучение сотрудников клиента. Также происходит сбор дополнительных требований и пожеланий клиента о модификации сопровождаемых проектов. Информация о требованиях и пожеланиях вносится в К7.

* 1. Экономическая сущность комплекса экономических информационных задач

Сотрудники отдела управления проектами сталкиваются с проблемой высокой трудоемкости в анализе дополнительных задач, так как анализ происходит с высокой долей ручной обработки. В ходе практики на предприятии была поставлена задача автоматизировать процессы анализа данных о дополнительных задачах в отделе управления проектами, с помощью программного средства, способного к интеграции с сервисом Asana и предоставлению сводных отчетов о задачах.

Комплекс экономических информационных задач, рассматриваемый в данной работе, относится к области управления проектной и операционной деятельностью сотрудников. Основная цель такого комплекса задач – обеспечение эффективного сбора, хранения, обработки и анализа информации о выполняемых задачах рабочих процессов компании. Предметной областью является отдел управления проектами ООО «Восток ИТ». Целью работы является автоматизация процесса анализа дополнительных задач путем разработки программного средства, интегрированного с сервисом Asana.

Экономическая сущность данного класса задач заключается в оптимизации управления трудовыми ресурсами предприятия. Эффективный учет задач и их анализ позволяют:

* своевременно выявлять проблемы в распределении нагрузки;
* сокращать время выполнения задач;
* минимизировать затраты, связанные с неэффективным использованием рабочего времени.

Внедрение средства автоматизации учета и анализа задач способствует достижению следующих экономических эффектов:

* повышение прозрачности выполнения проектов;
* снижение издержек на управление персоналом;
* улучшение качества обслуживания клиентов за счет более эффективного распределения ресурсов.

Управление задачами без применения автоматизированных средств приводит к значительным рискам: росту количества ошибок, увеличению временных затрат, возникновению перегрузок сотрудников или их недозагрузки, что отрицательно сказывается на финансовых результатах предприятия.

Таким образом, автоматизация анализа задач сотрудников открывает дополнительные возможности для повышения эффективности бизнеса и делает необходимым внедрение специализированных программных средств.

* 1. Обоснование выбора технологии проектирования

Были рассмотрены две технологии проектирования: функционально-ориентированное проектирование и объектно-ориентированное проектирование (представленное методологией RUP).

* + 1. Функционально-ориентированное проектирование

Функционально-ориентированное проектирование основывается на разбиении системы на отдельные функции или процедуры, каждая из которых выполняет конкретную задачу. Такой подход позволяет чётко представить, какие операции выполняются, как они связаны между собой и в какой последовательности происходят. Это способствует глубокой декомпозиции деятельности предприятия и помогает выявить ключевые процессы и задачи.

Одним из ключевых преимуществ данного подхода является его простота, наглядность и ориентированность на анализ бизнес-логики. На ранних стадиях проектирования он облегчает понимание предметной области, формализацию требований и выявление проблемных зон. Распространённой методологией, реализующей этот подход, является IDEF0, в рамках которой создаются функциональные диаграммы, упрощающие моделирование процессов и постановку задач.

Несмотря на то, что функциональный подход не предполагает строгой формализации архитектуры и может ограничивать возможности повторного использования кода, его применение оправдано в проектах, где ключевым является понимание процессов, чёткое описание функциональных зависимостей и поддержка логической целостности. Благодаря своей универсальности, он также может быть адаптирован для реализации программной части, с фокусом на чёткую организацию функций и модулей [12].

* + 1. Объектно-ориентированное проектирование

Объектно-ориентированное проектирование предполагает моделирование системы в виде совокупности взаимодействующих объектов, которые сочетают в себе данные и поведение. Это позволяет организовать программную систему как структуру взаимосвязанных, автономных компонентов. Такой подход способствует модульности, повторному использованию и масштабируемости решений, что особенно ценно при создании крупных и развиваемых программных систем.

Методология RUP (Rational Unified Process) представляет собой итеративную модель разработки, основанную на объектно-ориентированном подходе. Она предполагает поэтапное уточнение требований, проектирование архитектуры и реализацию программных компонентов. Одним из достоинств RUP является активное применение UML-диаграмм, упрощающих коммуникацию между участниками проекта и документацию архитектурных решений.

Однако объектно-ориентированное проектирование требует значительных усилий на этапе архитектурного планирования, технического анализа и поддержки строгой организационной дисциплины. При необходимости анализа процессов и выявления функциональных зависимостей, данный подход может излишне усложнять модель и затруднять восприятие [5].

* + 1. Выбор технологии проектирования

Сравнительный анализ показал, что функционально-ориентированное проектирование более эффективно соответствует целям работы на всех ее этапах — от анализа бизнес-процессов до реализации программного решения. Простота и наглядность функционального подхода позволяют чётко моделировать деятельность предприятия, выявлять ключевые функции и устанавливать взаимосвязи между ними. Это облегчает постановку задач, построение архитектуры и реализацию решений в виде модулей, организованных вокруг чётко определённых функций.

Объектно-ориентированный подход, несмотря на его архитектурную мощность и модульность, требует более глубокой проработки структуры данных и взаимодействий между объектами. Это делает его менее удобным для задач, связанных с анализом и визуализацией процессов, особенно на начальных этапах.

Таким образом в рамках работы принято решение использовать функционально-ориентированный подход, который обеспечит более логичную и понятную структуру проекта.

* 1. Обоснование необходимости и цели использования вычислительной техники для решения задачи
     1. Обоснование автоматизации решения задачи

ООО «Восток ИТ» занимается разработкой и внедрением программного обеспечения на базе 1С: изучает требования клиентов, модифицирует их ПО, устанавливает обновления, обучает клиентов использованию внесенных изменений. Для более подробного анализа деятельности ООО «Восток ИТ» построена модель бизнес-процессов.

Контекстный уровень модели бизнес-процессов в нотации IDEF0 представлен на рисунке Рисунок 2.

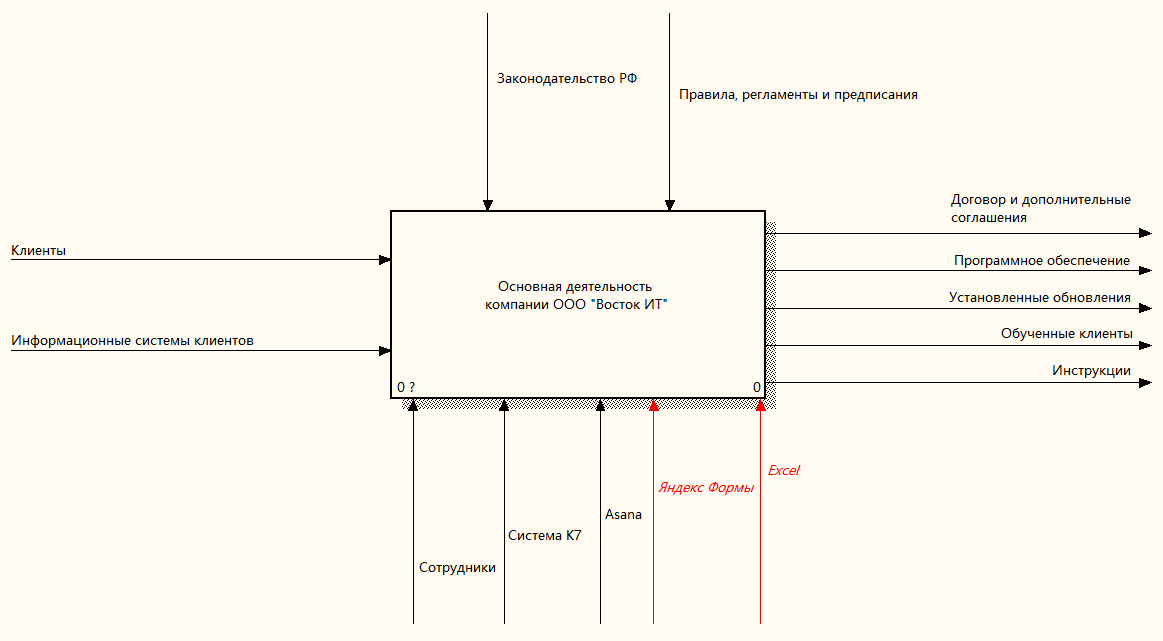


Рисунок 2 – Контекстная диаграмма модели бизнес-процессов ООО «Восток ИТ»

Среди выделенных интерфейсных дуг представлены «Яндекс Формы» и «Excel». Эти механизмы требуют значительной доли ручной обработки данных. Для более подробного анализа проведена декомпозиция контекстного уровня диаграммы.

Декомпозиция контекстной диаграммы представлена на рисунке Рисунок 3.

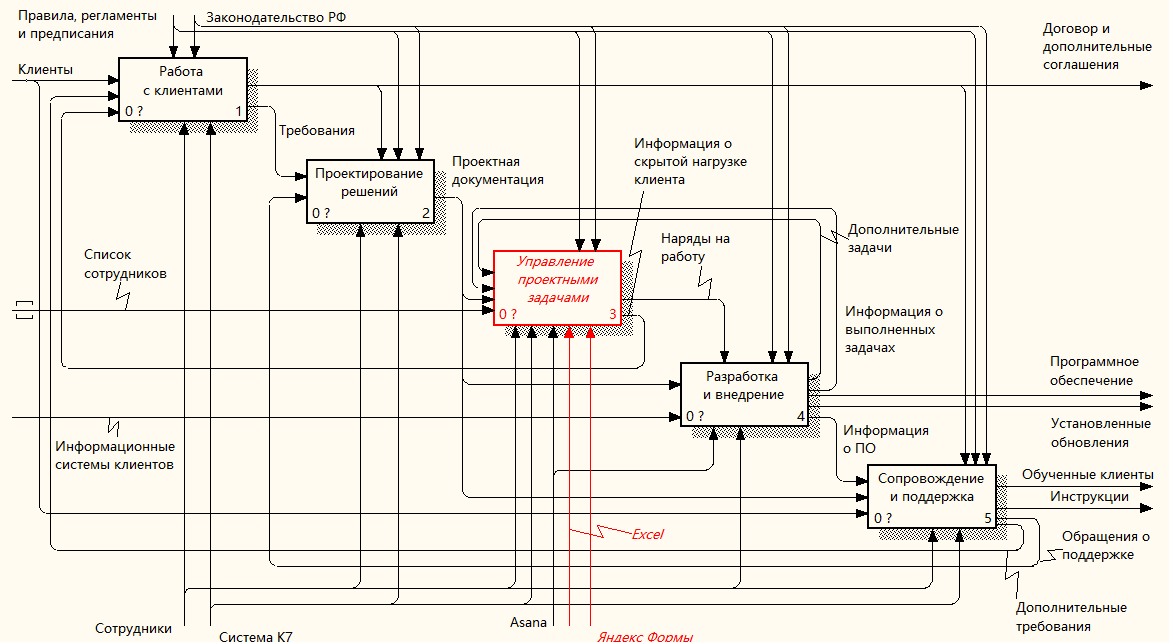


Рисунок 3 – Декомпозиция контекстной диаграммы

На декомпозиции контекстной диаграммы видно, что программные средства и сервисы со значительной долей ручной обработки используются в процессе «Управление проектными задачами». Для анализа выполнена декомпозиция процесса «Управление проектными задачами».

Декомпозиция процессов «Работа с клиентами», «Проектирование решений», «Разработка и внедрение» и «Сопровождение и поддержка» представлены на рисунках А.1 – А.4.

Процесс «Управление проектными задачами» начинается с формирования проектных команд. Далее задачи из проектной документации проекта распределяются между сотрудниками отдела разработки и внедрения. Важным этапом является анализ дополнительных задач, которые поступают от отдела разработки и внедрения. Полученные в ходе анализа данные используются в процессе распределения задач для создания более корректной и равномерной нагрузки на сотрудников. Декомпозиция процесса «Управление проектными задачами» представлена на рисунке Рисунок 4.

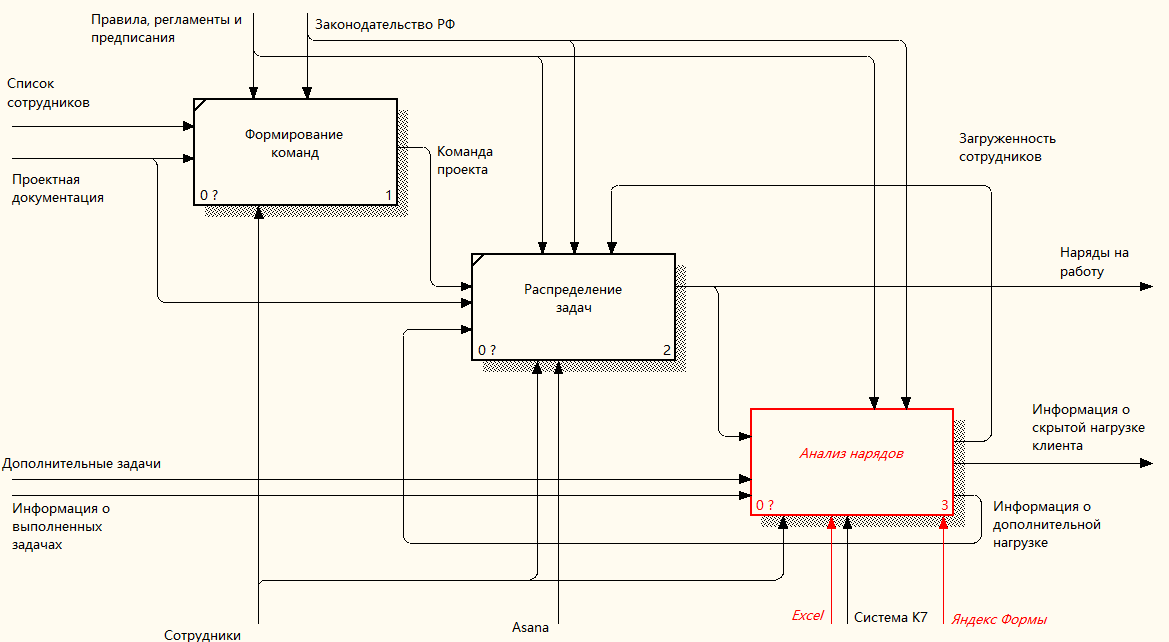


Рисунок 4 – Декомпозиция процесса «Управление проектными задачами»

Наиболее нагружен ручной обработкой процесс «Анализ нарядов». Для более глубокого анализа выполнена декомпозиция процесса «Анализ нарядов». Декомпозиция процесса «Анализ нарядов» представлена на рисунке Рисунок 5.

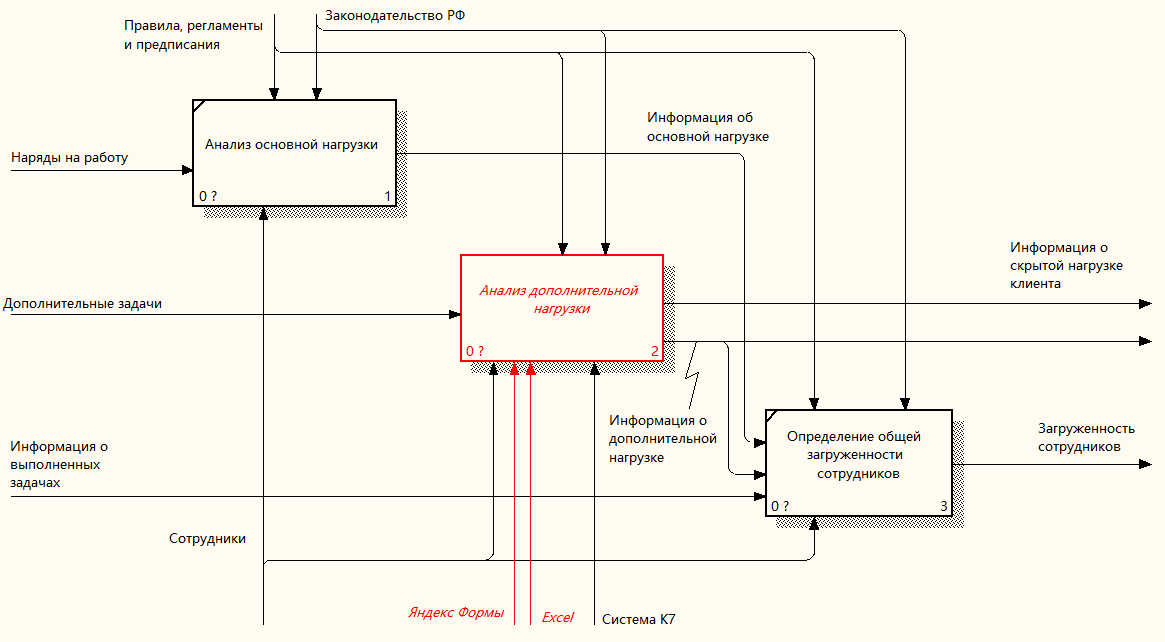


Рисунок 5 – Декомпозиция процесса «Анализ нарядов»

Выявлено, что оба высоконагруженных ручной обработкой программных решения используются в процессе «Анализ дополнительной нагрузки». Для более глубокого анализа выполнена декомпозиция процесса «Анализ дополнительной нагрузки». Декомпозиция процесса «Анализ дополнительной нагрузки» представлена на рисунке Рисунок 6.

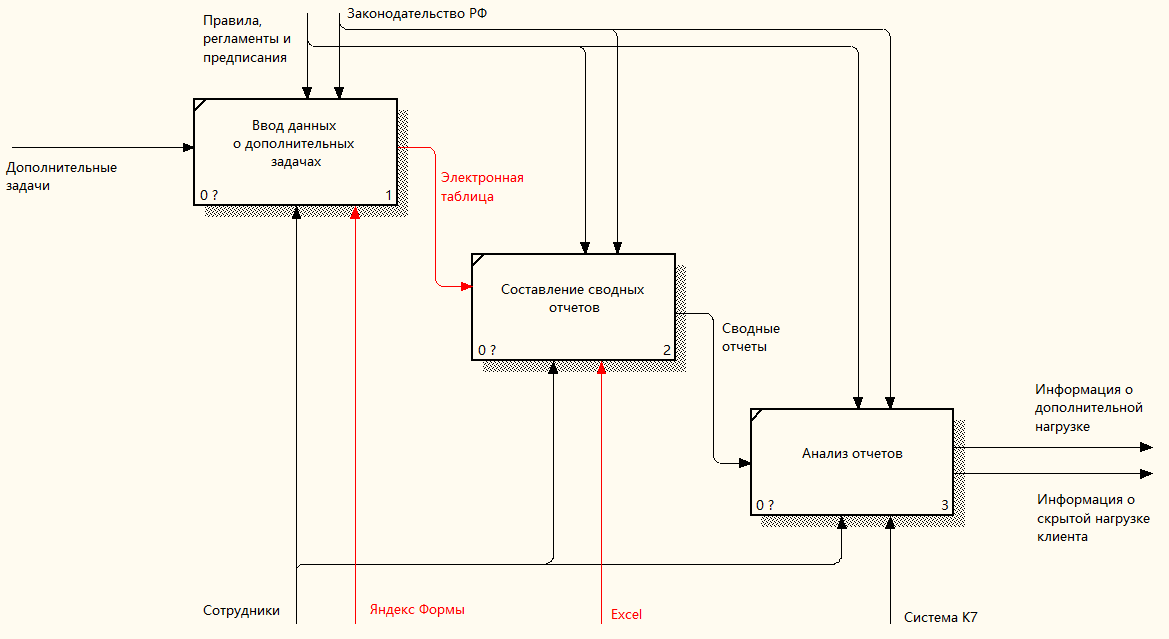


Рисунок 6 – Декомпозиция процесса «Анализ нарядов»

В текущем процессе анализа дополнительной нагрузки сотрудники вручную вводят информацию через Яндекс Формы, затем экспортируют её в Excel для составления отчетов и проведения анализа. Это приводит к ряду существенных проблем:

* увеличение длительности процесса: сотрудники тратят время на скачивание данных из Яндекс Форм и перенос их в Excel, что замедляет процесс получения аналитики и снижает общую продуктивность;
* риск ошибок из-за человеческого фактора: отчеты в Excel формируются вручную: создаются диаграммы, сводные таблицы, прописываются формулы и добавляются поля без использования шаблонов или аналитических панелей., это ограничивает глубину анализа и увеличивает вероятность ошибок.;
* дублирование работы: данные о задачах не интегрированы напрямую с сервисом Asana., в результате одну и ту же информацию приходится вводить как в Яндекс Формы, так и в Asana., также из-за отсутствия интеграции отсутствует возможность отслеживания состояния задач.

Последствия данных проблем:

* высокая вероятность ошибок в отчетах;
* низкая эффективность анализа;
* потери производительности сотрудников;
* вероятность расхождения в данных между сервисами.

Также в текущих процессах управления проектами отсутствует деятельность по анализу основного распределения задач по проектам. Для отслеживания текущих задач используется сервис канбан досок Asana, но он не предоставляет возможностей глубокого анализа и хранения исторической информации. Отсутствие анализа приводит к сложностям в расчете нагрузки на сотрудников и может привести к некорректному распределению задач, в результате чего часть сотрудников будет перегружена, а часть недогружена.

* + 1. Цели и задачи автоматизации

Цель автоматизации – снизить вероятность ошибок в отчетах, повысить эффективность и оперативность анализа, снизить трудозатраты на анализ и исключить дублирование данных. Для решения предлагается разработка программного средства, автоматически импортирующее данные из Asana, и обеспечивающее создание отчетов, хранение данных и анализ как текущих, так и выполненных задач.

Предлагаемое решение позволит:

* минимизировать человеческий фактор при составлении отчетов: программное средство автоматически формирует отчеты по запросу, без необходимости вручную вводить формулы, выбора полей для отчетов и диаграмм;
* снизить длительность процесса: благодаря автоматической загрузке и обработке данных исключается ручной экспорт и составление отчетов, что ускоряет получение аналитики;
* исключить дублирование данных: программное средство получает данные напрямую из Asana, устраняя необходимость использования Яндекс Форм и ручного ввода информации в нескольких местах;
* обеспечить глубокий и гибкий анализ: импорт всех данных из Asana позволит анализировать как дополнительные, так и основные задачи проекта;
* добавить хранение и анализ исторических данных: новая программное средство сохраняет данные о задачах даже после их удаления из Asana, что позволяет формировать отчеты по уже завершённым задачам и отслеживать изменения во времени.

Ожидаемый практический эффект для предприятия:

* более точный анализ нагрузки на сотрудников;
* эффективное распределение задач между ними;
* уменьшение количества перегруженных или недогруженных специалистов;
* снижение времени на обслуживание клиентов;
* повышение доверия и удовлетворённости клиентов за счёт более стабильного и прогнозируемого качества обслуживания.
  + 1. Анализ существующих разработок и обоснование выбора технологии проектирования

Для решения задач управления проектами существует несколько распространенных инструментов:

Asana — это сервис для организации работы над проектами различного масштаба. По заявлению разработчиков основным преимуществом является визуальное представление задач в виде списков, досок и календарей, что облегчает отслеживание прогресса и распределение обязанностей. Сервис предлагает базовые функции по планированию, постановке задач и отслеживанию сроков. Asana позиционируется как инструмент для маркетинговых, дизайнерских, административных и управленческих командах, где важна визуализация процессов и простота взаимодействия [18].

Jira — сервис управления задачами и проектами от Atlassian, изначально разработанный для команд программистов и технических специалистов. Приложение поддерживает гибкие методологии разработки (Agile, Scrum) и включает функции для ведения спринтов, управления бэклогом, отслеживания ошибок, релизов и пользовательских запросов. Jira отличается высокой степенью настройки рабочих процессов, возможностью автоматизации, построения отчетов и интеграцией с другими продуктами Atlassian (например, Confluence, Bitbucket). Позиционируется для IT-среды, особенно компаний, практикующих DevOps и гибкие методологии разработки [19].

ПланФикс — сервис, сочетающий функции CRM, учета задач и средством для управления внутренними процессами компании. Основным отличием разработчики выделяют гибкость в настройке бизнес-логики, включая шаблоны задач, автоматические действия, пользовательские формы и поля. Подходит как для организации проектной работы, так и для регулярного операционного учета, ведения клиентов и документооборота. ПланФикс позиционируется для малого и среднего бизнеса, а также компаний, где требуется комплексное управление разными видами деятельности в одном сервисе [20].

Для сравнения сервисов выделены следующие показатели:

* канбан доски – наличие визуального отображения задач в виде канбан досок;
* командное управление задачами – возможность создания команд, в которых каждый участник имеет возможность изменять задачи;
* простота интерфейса – интуитивность интерфейса и возможность использования без дополнительного обучения;
* создание дополнительных полей задач – возможность создавать для задач дополнительные поля (например наименование клиента, для которого выполняется задача);
* базовая аналитика – наличие инструментов для отслеживания количества задач, прогресса их выполнения, нагрузки на сотрудников, и т.д;
* аналитика по дополнительным полям – возможность формирования аналитических отчетов по дополнительным полям задач (например количество задач по каждому клиенту);
* формирование отчетов по параметрам – возможность быстрой настройки значений, по которым будет проводиться фильтрация данных отчета;
* хранение истории задач – возможность хранения задач, отмеченных как завершенные и их аналитики.

Сравнение существующих инструментов управления проектами представлено в таблице 1. Для оценки использован балльно-рейтинговый метод – каждому показателю назначен вес в долях единицы. Каждому сервису присвоен балл по каждому показателю по десятибалльной шкале, где 0 – не соответствует показателю, 10 – полностью соответствует показателю.

Таблица 1 – Сравнение инструментов управления проектами

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Показатель | Вес показателя | Asana | Jira | ПланФикс |
| Канбан доски | 1 | 10 | 10 | 10 |
| Командное управление задачами | 0,7 | 10 | 10 | 10 |
| Простота интерфейса | 0,8 | 10 | 7 | 4 |
| Создание дополнительных полей задач | 0,8 | 2 | 8 | 10 |
| Базовая аналитика | 0,3 | 1 | 7 | 9 |
| Аналитика по дополнительным полям | 0,9 | 0 | 5 | 10 |
| Формирование отчетов по параметрам | 0,7 | 0 | 3 | 3 |
| Хранение истории задач | 0,4 | 2 | 7 | 7 |
| Итог (JЭТУ) | | 27,7 | 40,5 | 44,8 |

Расчет итогов произведен по формуле (1):

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

где JЭТУ – комплексный показатель эксплуатационно-технического уровня сервиса по группе показателей;

n – число рассматриваемых показателей;

Вi – коэффициент весомости i-го показателя;

Xi – оценка соответствия сервиса показателю.

ООО «Восток ИТ» в настоящее время использует Asana для управления задачами и проектами, однако по ключевым критериям — таким как создание дополнительных полей и аналитика — более подходящим сервисом является ПланФикс. Тем не менее, руководство приняло решение не менять текущий инструмент, чтобы избежать дополнительных затрат, сложностей миграции и необходимости переобучения сотрудников.

Для решения ранее упомянутых проблем, таких как сложность аналитики и использование ручной обработки, руководством была поставлена задача разработки собственного инструмента для автоматического импорта данных из Asana и их последующего анализа.

* + 1. Предлагаемая модель бизнес-процессов

Была построена модель бизнес-процессов ООО «Восток ИТ» КАК-БУДЕТ (TO-BE) в нотации IDEF0. Изменения выделены в процессе «Управление проектными задачами». Декомпозиция процесса «Управление проектными задачами» представлена на рисунке Рисунок 7.

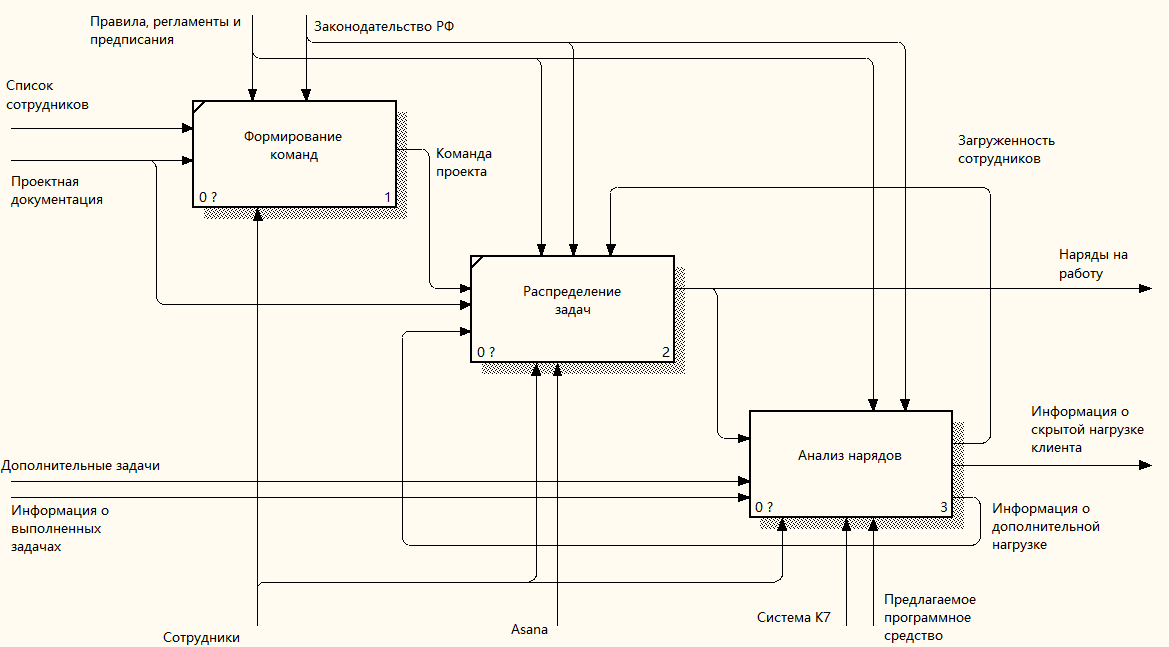


Рисунок 7 – Декомпозиция процесса «Управление проектными задачами»

Изменения коснулись процесса «Анализ нарядов». Декомпозиция процесса «Анализ нарядов» представлена на рисунке Рисунок 8.

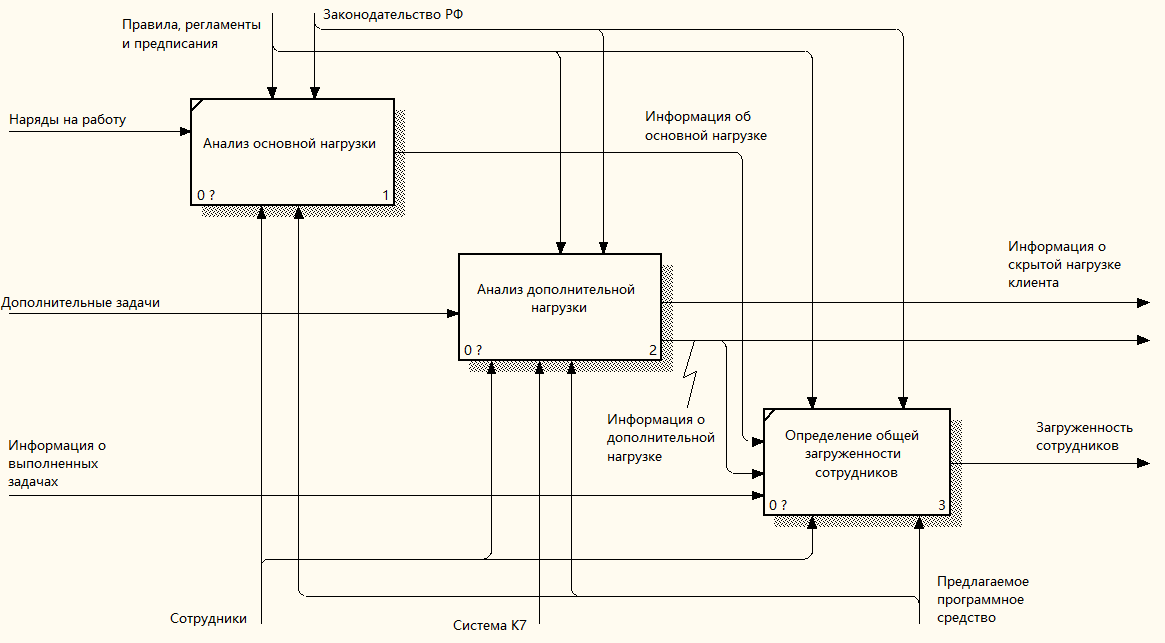


Рисунок 8 – Декомпозиция процесса «Анализ нарядов»

Предлагаемое программное средство используется во всех процессах, так как предоставляет функциональные возможности по учету, отслеживанию и формировании отчетности по всем проектным задачам. Процесс «Анализ дополнительной нагрузки», который имел выделенные ранее проблемы декомпозирован. Декомпозиция процесса «Анализ дополнительной нагрузки» представлена на рисунке Рисунок 9.

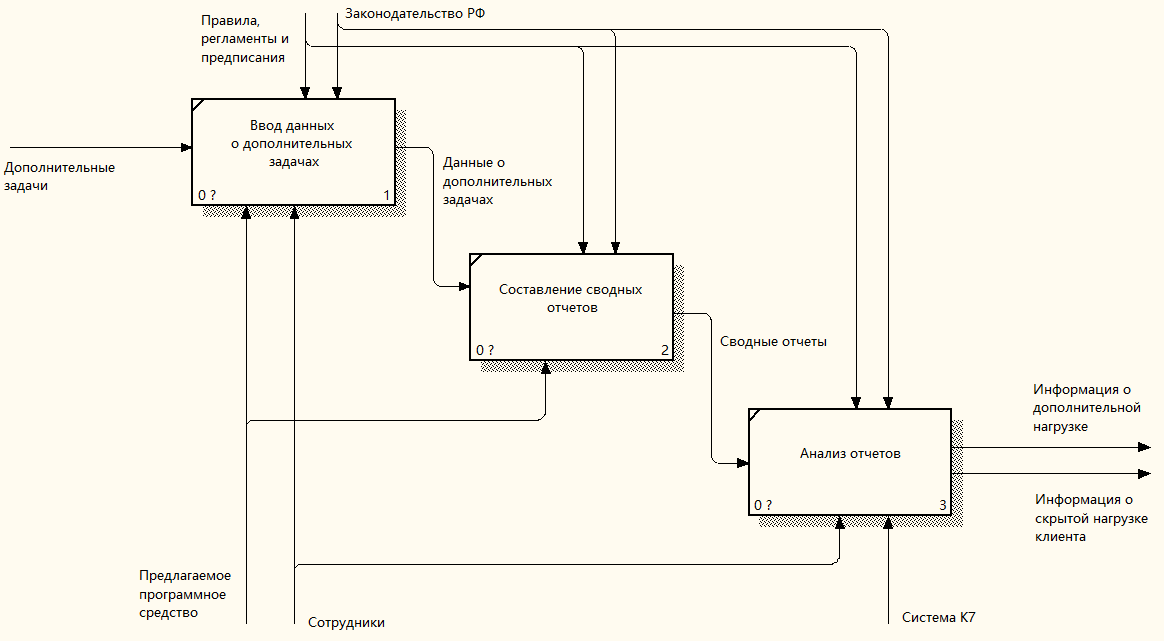


Рисунок 9 – Декомпозиция процесса «Анализ дополнительной нагрузки»

Использование предлагаемого программного средства позволило отказаться от программных решений с высокой долей ручной обработки Яндекс Формы и Excel. Процесс «Составление сводных отчетов» теперь выполняется автоматически, без необходимости сотрудникам вручную переносить данные между Яндекс Формами и Excel, и вручную составлять и настраивать отчеты в Excel, что решает выявленные ранее проблемы высокой длительности, трудоемкости и вероятности ошибок в процессе.

* 1. Постановка задачи
     1. Общая характеристика организации задачи на ЭВМ

Целью автоматизации является снижение вероятности ошибок в отчетах, повышение эффективности и оперативности анализа, снижение трудозатрат на анализ и исключение дублирования данных.

Назначением задачи является автоматизированный учет задач сотрудников, статусов их выполнения и формирования отчетности.

Автоматизируется процесс управления проектными задачами. В настоящее время процесс анализа дополнительных задач требует высокой доли ручной обработки данных, что увеличивает длительность процесса и вероятность ошибок. Для решения проблем целесообразно автоматизировать процесс с использованием программных средств.

Периодичность и продолжительность работы программного средства: постоянный прием новых задач и статусов, отчетность по запросу.

Условия прекращения автоматизированного выполнения задачи: критические ошибки программного средства или инфраструктуры.

Связи с другими задачами: интеграция с сервисом Asana.

Входная информация представлена дополнительными и основными задачами по нарядам. Основные задачи создаются в Asana. Дополнительные задачи вводятся в форму и отправляются в Asana для отображения на канбан доске. Основные задачи создаются напрямую в сервисе Asana. Информация об изменении статусов как основных, так и дополнительных задач должна извлекаться из Asana с помощью API, предоставляемого сервисом Asana. Вся информация о задачах и статусах должна сохраняться в локальной базе данных. Результатная информация должна выводиться на экран в виде отчетов.

* + 1. Характеристика выходной информации

Программное средство формирует выходную информацию в виде отчетов и служебных сообщений, которые отражают результаты анализа данных и выполнения операций. В качестве выходных данных предоставляются отчеты о распределении нагрузки между сотрудниками, сведения о дополнительных задачах по клиентам, а также технические сообщения для интеграции с внешним сервисом Asana.

Перечень и описание выходных сообщений представлены в таблице Таблица 2.

Таблица 2 – Выходные сообщения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Идентификатор | Наименование | Периодичность | Сроки выдачи | Формат данных |
| OUT01 | Отчет о нагрузке на сотрудников за период | По запросу | До 10 секунд | Отчет |
| OUT02 | Отчет о детальной нагрузке на сотрудника за период | По запросу | До 10 секунд | Отчет |
| OUT03 | Отчет о дополнительных задачах по клиентам за период | По запросу | До 10 секунд | Отчет |
| OUT04 | Детальный отчет о дополнительных задачах по клиенту за период | По запросу | До 10 секунд | Отчет |
| OUT05 | Запрос на создание задачи Asana | При вводе дополнительной задачи | До 10 секунд | JSON |

Перечень и описание структурных единиц информации выходных сообщений представлены в таблице Таблица 3.

Таблица 3 – Структурные единицы выходных сообщений

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | Идентификатор сообщения | Формат данных | Получатель | Идентификатор получателя |
| ФИО сотрудника | OUT01, OUT02 | Строка | Менеджер по проектам | S01 |
| Общее количество задач сотрудника | OUT01, OUT02 | Целое число | Менеджер по проектам | S01 |
| Количество завершенных задач сотрудника | OUT01, OUT02 | Целое число | Менеджер по проектам | S01 |
| Количество незавершенных задач сотрудника | OUT01, OUT02 | Целое число | Менеджер по проектам | S01 |
| Количество дополнительных задач на сотруднике | OUT01, OUT02 | Целое число | Менеджер по проектам | S01 |
| Среднее время выполнения задачи сотрудником | OUT01, OUT02 | Время в формате «часы:минуты» | Менеджер по проектам | S01 |
| Наименование статуса | OUT02 | Строка | Менеджер по проектам | S01 |
| Количество задач в статусе | OUT02 | Целое число | Менеджер по проектам | S01 |
| Заголовок задачи | OUT02, OUT04 | Строка | Менеджер по проектам | S01 |
| Текст задачи | OUT02, OUT04 | Строка | Менеджер по проектам | S01 |
| Длительность нахождение в статусе | OUT02 | Время в формате «часы:минуты» | Менеджер по проектам | S01 |
| Наименование клиента | OUT03, OUT04 | Строка | Менеджер по проектам | S01 |
| Количество дополнительных задач по клиенту | OUT03, OUT04 | Целое число | Менеджер по проектам | S01 |
| Количество сотрудников в дополнительных задачах клиента | OUT03, OUT04 | Целое число | Менеджер по проектам | S01 |
| Средняя продолжительность выполненных дополнительных задач | OUT03, OUT04 | Целое число | Менеджер по проектам | S01 |
| Идентификатор наряда в К7 | OUT04 | Строка | Менеджер по проектам | S01 |
| Ответственный по наряду | OUT04 | Строка | Менеджер по проектам | S01 |
| Исполнитель задачи | OUT04 | Строка | Менеджер по проектам | S01 |
| Текст задачи | OUT05 | Строка | Asana | S02 |
| Идентификатор проекта Asana | OUT05 | Строка | Asana | S02 |

Алгоритмы расчета результатных показателей представлены в таблице Таблица 4.

Таблица 4 – Алгоритмы расчета

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование результатного показателя | Идентификатор | Алгоритм расчета |
| 1 | Общее количество задач | Task\_total | COUNT(\*) по всем задачам, назначенным на сотрудника |
| 2 | Количество завершенных задач | Task\_done | COUNT(\*) WHERE deleted = True |
| 3 | Количество незавершенных задач | Task\_pending | COUNT(\*) WHERE deleted = False |
| 4 | Количество дополнительных задач | Task\_extra | COUNT(\*) WHERE additional\_info <> None |
| 5 | Среднее время выполнения задачи | Avg\_duration | AVG(DATEDIFF(end\_date, start\_date)) WHERE deleted = True |
| 6 | Количество дополнительных задач | Extra\_count | COUNT(\*) WHERE additional\_info <> None |
| 7 | Количество сотрудников, задействованных в задачах | Staff\_count | COUNT(DISTINCT employee\_id) WHERE client = Client\_name |
| 8 | Средняя продолжительность дополнительных задач | Avg\_extra\_time | AVG(DATEDIFF(end\_date, start\_date)) WHERE additional\_info <> None |

* + 1. Характеристика входной информации

К числу входных данных относятся сведения о новых задачах, параметры для формирования отчетов, а также конфигурационные данные для интеграции с Asana. Эти данные вводятся пользователями через экранные формы программного средства и поступают в структурированном формате (JSON) от сервиса Asana.

Перечень и описание входных сообщений представлены в таблице Таблица 5.

Таблица 5 – Входные сообщения

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Наименование | Сроки поступления | Формат данных |
| IN01 | Дополнительная задача | При обнаружении дополнительной задачи | Ввод в форму |
| IN02 | Событие Asana | При запросе событий Asana | JSON |
| IN03 | Данные новой задачи Asana | После отправки запроса за создание задачи Asana | JSON |
| IN04 | Период для формирования отчета | При запросе отчета | Ввод в форму |
| IN05 | Данные для интеграции Asana | При первичной настройке программного средства | Ввод в форму |
| IN06 | Список проектов Asana | После ввода токена | JSON |
| IN07 | Список пользователей в проекте Asana | При настройке списка сотрудников | JSON |
| IN08 | ФИО сотрудников | При настройке списка сотрудников | Ввод в форму |

Перечень и описание структурных единиц информации входных сообщений представлены в таблице Таблица 6.

Таблица 6 – Структурные единицы входных сообщений

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | Идентификатор сообщения | Формат данных | Источник | Идентификатор источника |
| ФИО программиста | IN01, IN08 | Строка | Менеджер по проектам | S01 |
| Номер наряда | IN01 | Целое число | Менеджер по проектам | S01 |
| Менеджер по наряду | IN01 | Строка | Менеджер по проектам | S01 |
| Наименование клиента | IN01 | Строка | Менеджер по проектам | S01 |
| Описание задачи | IN01 | Строка | Менеджер по проектам | S01 |
| Идентификатор ресурса | IN02 | Строка | Asana | S02 |
| Тип ресурса | IN02 | Строка | Asana | S02 |
| Тип события | IN02 | Строка | Asana | S02 |
| Идентификатор зависимого объекта | IN02 | Строка | Asana | S02 |
| Тип зависимого объекта | IN02 | Строка | Asana | S02 |
| Идентификатор задачи | IN03 | Строка | Asana | S02 |
| Дата начала | IN04 | Дата | Менеджер по проектам | S01 |
| Дата окончания | IN04 | Дата | Менеджер по проектам | S01 |
| Идентификатор проекта Asana | IN05, IN06 | Строка | Менеджер по проектам, Asana | S01, S02 |
| API токен Asana | IN05 | Строка | Менеджер по проектам | S01 |
| Наименование проекта Asana | IN06 | Строка | Asana | S02 |
| Идентификатор пользователя Asana | IN07 | Строка | Asana | S02 |
| Имя пользователя Asana | IN07 | Строка | Asana | S02 |

1. Проектная часть
   1. Календарное планирование
      1. Содержание проекта

В соответствии со стандартом «ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств», а также методологией RUP (Rational Unified Process) выделены основные этапы проекта и сформирован состав работ. Этапы проекта представлены в таблице Таблица 7. Задачи проекта представлены в таблице Б.1.

Таблица 7 – Этапы проекта

|  |  |
| --- | --- |
| Наименование | Описание |
| Составление плана проекта | Планирование проекта: определение состава работ, длительности, бюджета и рисков |
| Анализ требований и проектирование | Определение требований и проектирование программного средства |
| Разработка | Разработка: программирование и верстка |
| Тестирование и исправление ошибок | Тестирование и исправление ошибок |
| Внедрение | Подготовка и запуск программного средства |
| Сдача проекта | Представление результатов заказчику, передача инструкций |

* + 1. Оценка длительности

Для определение ожидаемой длительности задач применен метод PERT. Ожидаемая длительность рассчитывается по формуле (2):

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

где Tож – ожидаемая длительность;

Tmin – оптимистичная оценка: кратчайшая продолжительность работы, с учетом того, что во время выполнения задачи не возникнет факторов, которые могут увеличить длительность выполнения;

Tнв – реалистичная оценка: наиболее вероятная продолжительность работы;

Tmax – пессимистичная оценка: наибольшая продолжительность работы, с учетом того, что во время выполнения задачи возникнет множество факторов, которые могут увеличить длительность выполнения.

Оценка продолжительности этапов представлены в таблице Таблица 8. Подробные оценки по задачам представлены в таблице Б.2.

Таблица 8 – Ожидаемая продолжительность этапов

|  |  |
| --- | --- |
| Наименование | Ожидаемая продолжительность, дней |
| Составление плана проекта | 8 |
| Анализ требований и проектирование | 47 |
| Разработка | 54 |
| Тестирование и исправление ошибок | 48 |
| Внедрение | 12 |
| Сдача проекта | 1 |
| Итог | 170 |

* + 1. Планирование ресурсов и расчет затрат

Список ресурсов проекта представлен в таблице Таблица 9.

Таблица 9 - Список ресурсов

|  |  |  |
| --- | --- | --- |
| Наименование | Тип | Описание |
| Исполнитель | Трудовой | Основной исполнитель проекта. Выполняет основные задачи проекта: планирование, анализ, разработка, тестирование и внедрение |
| Заказчик | Трудовой | Представитель заказчика. Участвует в задачах определения требований и сдачи проекта |
| Руководитель дипломной работы | Трудовой | Преподаватель университета. Курирует проект и проводит необходимые консультации по работе |
| Ноутбук | Материальный | Ноутбук, с помощью которого производится выполнение проекта |

Для расчета стандартных ставок трудовых ресурсов используется среднее значение предлагаемых зарплат на соответствующие должности в регионе Приморский край:

Ресурс «Исполнитель»: так как исполнитель является студентом университета, выполняющим выпускную квалификационную работу, стандартная ставка ресурса 0 рублей в час.

Ресурс «Заказчик»: вакансии «Руководитель проектов», «Менеджер отдела проектов» и подобные. Стандартная ставка 350 рублей в час.

Ресурс «Руководитель дипломной работы»: вакансии «Преподаватель». Стандартная ставка 815 рублей в час.

Стоимость материальных ресурсов:

Ресурс «Ноутбук»: в качестве стоимости ресурса используется величина амортизации за период проекта. Стоимость ресурса 5000 рублей.

Ресурсы проекта в среде Microsoft Project представлены на рисунке Б.1.

* + 1. Оптимизация

Для оптимизации плана проекта применен метод критического пути. Проведен анализ и внесены изменения для сокращения длительности проекта.

Анализ этапов «Составление плана проекта» и «Анализ требований и проектирование» показал, что их необходимо выполнять последовательно, так как именно на этапе «Составление плана проекта» формируется список задач для этапа «Анализ требований и проектирование», а также для других этапов, что не позволяет выполнять этап «Составление плана проекта» параллельно с другими этапами.

Анализ этапов «Анализ требований и проектирование» и «Разработка» показал, что эти этапы необходимо выполнять последовательно, так как на этапе «Анализ требований и проектирование» составляется проект программного средства, на основе которого будет выполняться разработка.

Внутри этапа «Анализ требований и проектирование» возможно выполнять задачу «Проектирование интерфейса» после задачи «Проектирование архитектуры» с запаздыванием в 1 день, так как именно в ходе выполнения задачи «Проектирование интерфейса» формируется разделение программного средства по уровням и определяются пакеты, нуждающиеся в интерфейсе, который будет разрабатываться в задаче «Проектирование интерфейса». Это изменение сокращает длительность проекта на 6 дней.

Анализ этапов «Разработка» и «Тестирование и исправление ошибок» показал, что эти этапы после задачи «Backend программирование» возможно выполнять параллельно, так как именно задача «Backend программирование» является наиболее крупной в этапе и в ней ожидается наибольшее число потенциальных ошибок, которые возможно обнаружить и исправить. Это изменение сокращает количество задач критического пути на 3 задачи и сокращает длительность на 25 дней.

Анализ этапов «Разработка», «Тестирование и исправление ошибок» и «Внедрение» показал, что этап «Внедрение» необходимо выполнять последовательно этапам «Разработка» и «Тестирование и исправление ошибок», так как в ходе их выполнения разрабатывается и отлаживается программное средства, внедрение которого будет проходить в этапе «Внедрение».

Оптимизация позволила сократить длительность проекта на 30 дней. Проект заканчивается за 56 дней до крайнего срока, что создает резерв для реагирования на неизвестные риски. Оптимизированный план представлен на рисунках А.11 – А.13.

Оптимизированный проект в среде Microsoft Project представлен на рисунках Б.2 – Б.4.

* + 1. Идентификация рисков и разработка стратегии их смягчения

Основным риском в расписании является задача «Backend программирование», имеющая длительность 30 дней. Использование задач с большой длительностью снижает качество планирования и повышает количество неточностей. Для облегчения контроля задачу необходимо разделить на более короткие задачи. Задачи, полученные путем декомпозиции задачи «Backend программирование» представлены в таблице Таблица 10.

Таблица 10 – Декомпозиция задачи «Backend программирование»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование | Описание | Оптимистичная оценка, Tmin, дней | Реалистичная оценка, Tнв, дней | Пессимистичная оценка, Tmax, дней | Ожидаемая продолжительность, Tож, дней |
| Разработка основных классов | Разработка классов, полученных на этапе проектирования | 2 | 4 | 12 | 5 |
| Разработка бизнес-логики | Разработка функций и методов, обеспечивающих функционал | 10 | 15 | 26 | 15 |
| Разработка взаимодействия с СУБД | Разработка функций и методов, обеспечивающих взаимодействие программного средства с системой управления базами данных (СУБД) | 5 | 10 | 15 | 10 |

Ресурсный риск связан с ресурсом «Заказчик». Доступность ресурса неизвестна, существует риск недоступности ресурса в необходимое время. Параметры риска:

* вероятность возникновения: 0,4;
* потенциальное влияние: 1;
* ранг риска: 0,4 – средний;
* стратегия смягчения: увеличение длительности задачи «Интервьюирование заказчика» с 2 до 5 дней.
* план реакции: при возникновении риска возможно увеличение длительности задачи «Интервьюирование заказчика» на срок, необходимы для выполнения задачи. Это также увеличит длительность проекта на соответствующий срок, но наличие достаточного резерва времени не позволит этому повлиять на итоговые сроки.

Ресурсные риски связаны с ресурсом «Исполнитель». Ресурс имеет большой объем и долю работ проекта, что делает проект зависимым от ресурса. Параметры риска:

* вероятность возникновения: 0,05;
* потенциальное влияние: 1;
* ранг риска: 0,05 – низкий;
* стратегия смягчения: нет.
* план реакции: реакцией на возникновение риска должна быть передача проекта другому специалисту. При необходимости поиска специалиста добавляется задача «Поиск исполнителя» с ожидаемой длительностью 7 дней. Однако, так как проект выполняется в рамках выпускной квалификационной работы, замена исполнителя невозможна.
  + 1. Стоимость проекта

Общая стоимость проекта составляет 11 979 рублей.

Стоимость проекта по этапам представлена в таблице Таблица 11.

Таблица 11 – Стоимость по этапам

|  |  |
| --- | --- |
| Наименование | Стоимость, рублей |
| Составление плана проекта | 0 |
| Анализ требований и проектирование | 8 441 |
| Разработка | 0 |
| Тестирование и исправление ошибок | 0 |
| Внедрение | 763 |
| Сдача проекта | 280 |

Стоимость проекта по ресурсам представлена в таблице Таблица 12.

Таблица 12 – Стоимость по ресурсам

|  |  |  |
| --- | --- | --- |
| Наименование | Тип | Стоимость, рублей |
| Исполнитель | Трудовой | 0 |
| Заказчик | Трудовой | 2 443 |
| Руководитель дипломной работы | Трудовой | 7 041 |
| Ноутбук | Материальный | 5 000 |

Несмотря на расчетную стоимость проекта в размере 14 485 рублей, его реализация в рамках выпускной квалификационной работы не предполагает прямых финансовых затрат для заказчика. Единственные издержки со стороны заказчика связаны с временными ресурсами, что составляет 2 443 рубля.

* 1. Информационное обеспечение задачи
     1. Информационная модель её описание

Была построена модель информационных потоков в нотации DFD. Взаимодействующие с программным средством сущности:

* пользователь: вводит данные для интеграции, сотрудников и дополнительные задачи, получает отчеты;
* Asana: предоставляет список пользователей и данные о событиях, получает запросы на создание задач.

Контекстный уровень модели информационных поток представлена на рисунке Рисунок 10.

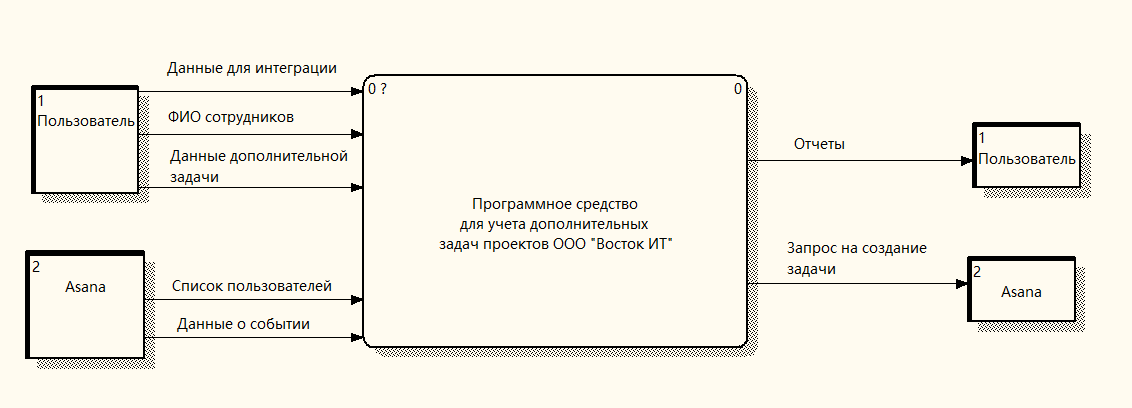


Рисунок 10 – Контекстный уровень модели информационных потоков

Декомпозиция контекстного уровня представлена на рисунке Рисунок 11.

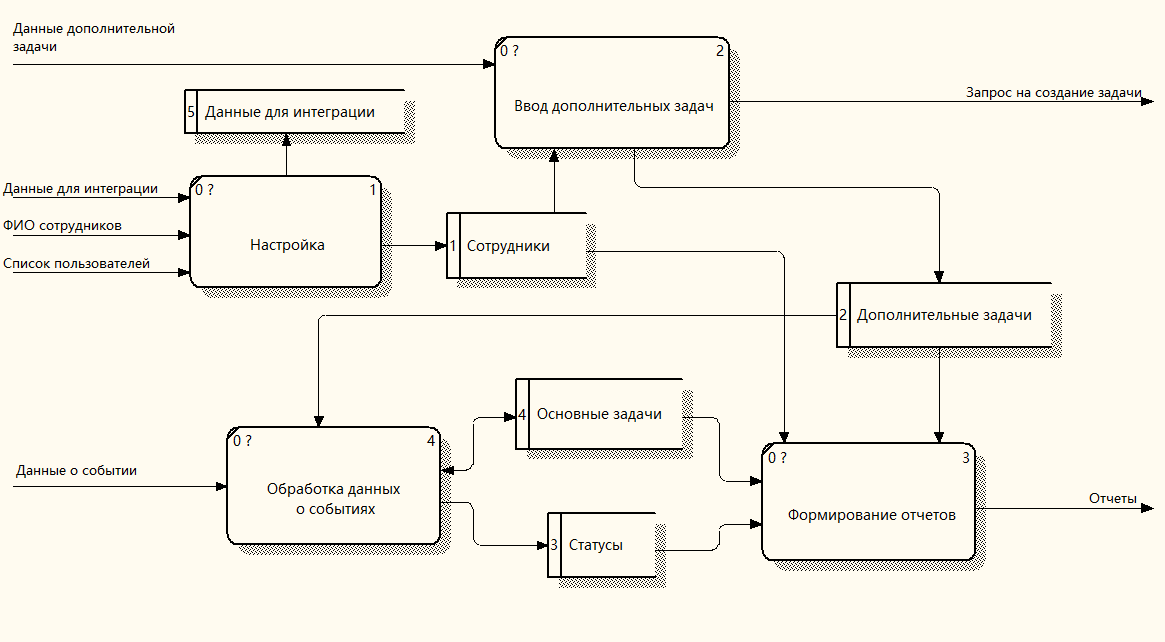


Рисунок 11 – Декомпозиция контекстного уровня модели

информационных потоков

Пользователь настраивает программное средство, вводя данные для интеграции с сервисом Asana и список сотрудников. Далее он вводит данные о дополнительных задачах, которые отправляются в сервис Asana. Сервис Asana сообщает о событиях (таких как перемещение добавление новой задачи, перемещение задач в колонках или их завершение), на основе которых сохраняется информация о задачах и их статусах. Сохраненная информация о сотрудниках, дополнительных задачах, основных задачах и статусах используется далее для формирования отчетов, которые пользователь может запросить.

* + 1. Характеристика базы данных

На основе выделенных хранилищ модели информационных потоков построена инфологическая диаграмма «сущность-связь». Инфологическая модель базы данных представлена на рисунке Рисунок 12.

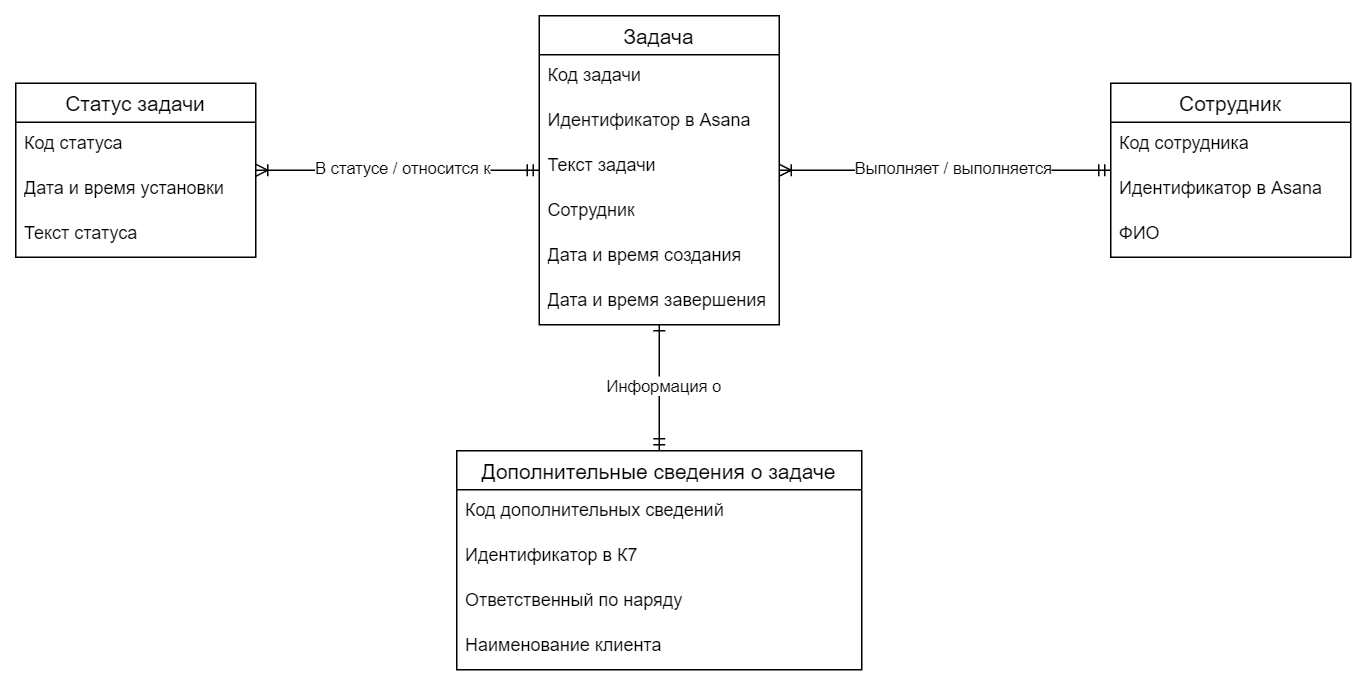


Рисунок 12 – Инфологическая диаграмма «сущность-связь»

Выделены следующие сущности:

* задача – хранение информации о задачах, общей как для основных, так и дополнительных;
* дополнительные сведения о задаче – используется для указания специальной информации о дополнительных задачах;
* сотрудник – используется для хранения информации о сотрудниках и связи с их идентификаторами в Asana;
* статус задачи – используется для хранения информации о статусах задач, изменяемых в Asana.

Для реализации проекта наиболее подходящей системой управления базами данных (СУБД) является SQLite, которая обладает рядом преимуществ:

* встраиваемость: SQLite представляет собой встраиваемую СУБД, не требующую отдельного серверного процесса, что значительно упрощает развёртывание и снижает нагрузку на ресурсы;
* стандартизация: SQLite поддерживает стандартный синтаксис SQL, обеспечивая совместимость с большинством языков программирования через унифицированные интерфейсы;
* надежность: SQLite поддерживает транзакции с соблюдением ACID-свойств (атомарность, согласованность, изолированность, долговечность), что необходимо для обеспечения целостности данных;
* простота: отсутствие встроенных механизмов разграничения доступа является преимуществом, поскольку в данном проекте не требуется сложная модель прав. Это упрощает архитектуру решения и снижает риск ошибок, связанных с управлением доступом.

Для определения состава и взаимосвязей таблиц, отражающих содержание информационных сущностей инфологической модели в системе управления базами данных SQLite построена даталогическая модель в нотации IDEF1X. Даталогическая модель представлена на рисунке Рисунок 13. Кроме выделенных инфологических сущностей добавлена обособленная таблица «Settings» для хранения настроек и параметров программного средства в формате «ключ-значение».

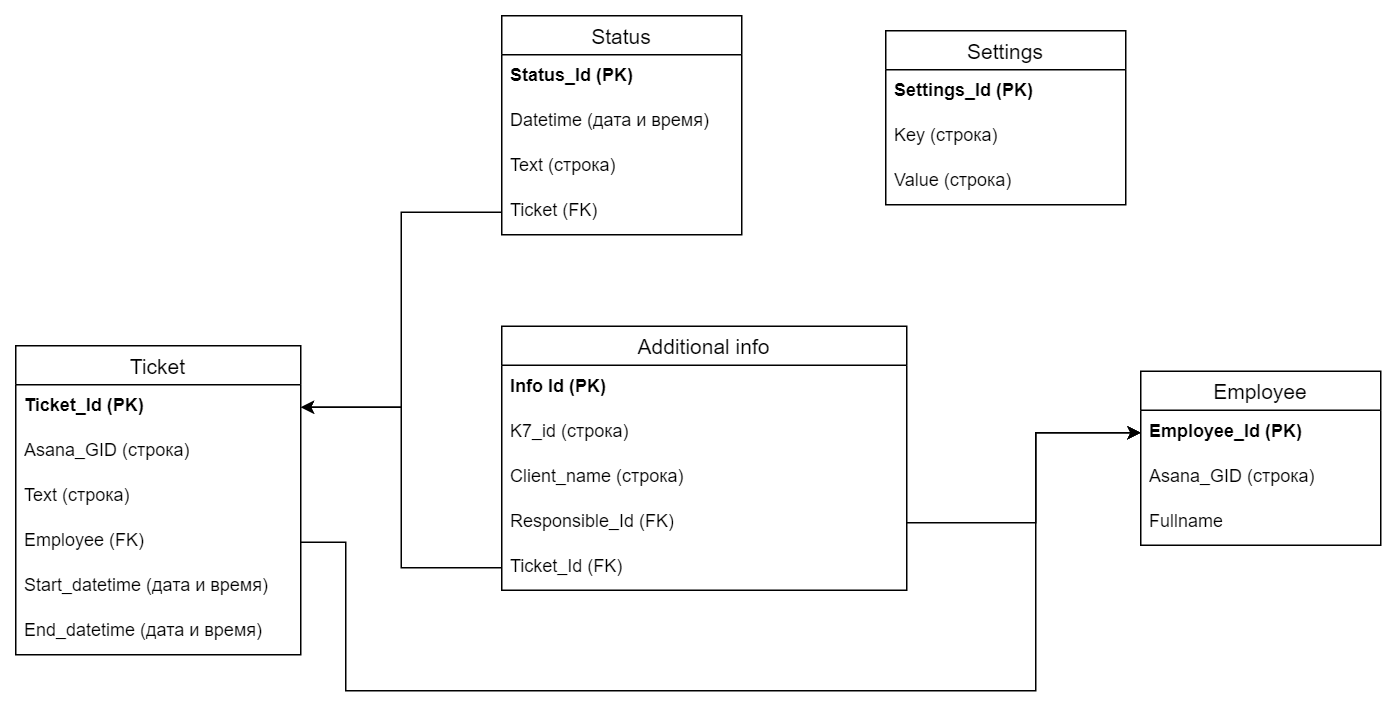


Рисунок 13 – Даталогическая модель базы данных

Схема таблицы «Ticket» представлена в таблице Таблица 13.

Таблица 13 – Схема таблицы «Ticket»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Ticket\_Id | Целое число, первичный ключ | Уникальный идентификатор задачи |
| Asana\_GID | Строка, уникальное | Идентификатор задачи в сервисе Asana |
| Text | Строка | Текст задачи |
| Employee | Целое число, внешний ключ | Идентификатор сотрудника, которому назначена задача |
| Start\_datetime | Дата и время | Дата и время создания задачи |
| End\_datetime | Дата и время | Дата и время завершения задачи |

Схема таблицы «Additional info» представлена в таблице Таблица 14.

Таблица 14 – Схема таблицы «Additional info»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Info\_Id | Целое число, первичный ключ | Уникальный идентификатор дополнительных данных о задаче |
| K7\_id | Строка | Идентификатор наряда в CRM-системе K7 |
| Client name | Строка | Наименование клиента, с которым связана задача |
| Responsible\_Id | Целое число, внешний ключ | Идентификатор сотрудника, ответственного за наряд К7 |
| Ticket\_Id | Целое число, внешний ключ | Идентификатор задачи, к которой относятся дополнительные данные |

Схема таблицы «Status» представлена в таблице Таблица 15.

Таблица 15 – Схема таблицы «Status»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Status\_Id | Целое число, первичный ключ | Уникальный идентификатор статуса |
| Datetime | Дата и время | Дата и время установки статуса |
| Text | Строка | Текст статуса |
| Ticket\_Id | Целое число, внешний ключ | Идентификатор задачи, которой установлен статуса |

Схема таблицы «Employee» представлена в таблице Таблица 16.

Таблица 16 – Схема таблицы «Employee»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Employee\_Id | Целое число, первичный ключ | Уникальный идентификатор сотрудника |
| Asana\_GID | Строка, уникальное | Идентификатор пользователя в системе Asana |
| Fullname | Строка | ФИО сотрудника |

Схема таблицы «Settings» представлена в таблице Таблица 17.

Таблица 17 – Схема таблицы «Settings»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Settings\_Id | Целое число, первичный ключ | Уникальный идентификатор параметра настроек |
| Key | Строка, уникальное | Наименование параметра |
| Value | Строка | Значение параметра |

* 1. Программное обеспечение задачи
     1. Дерево функций

Функционал программного средства учета дополнительных задач проектов ООО «Восток ИТ»:

* ввод дополнительной задачи: ввод данных об обнаруженной дополнительной задаче проекта;
* формирование отчета о нагрузке: формирование и просмотр отчета об общем количестве и типах задач по сотрудникам;
* формирование детального отчета о нагрузке: формирование и просмотр отчета о количестве, статусах и типах задач сотрудника;
* формирование отчета о клиентах: формирование и просмотр отчета о количестве дополнительных задач по клиентам;
* формирование детального отчета о клиенте: формирование и просмотр отчета о дополнительных задачах клиента;
* ввод данных для интеграции с Asana: ввод токена интеграции Asana и выбор проекта Asana для отслеживания;
* ввод списка сотрудников: ввод данных о сотрудниках, для связи их с пользователями Asana;
* обработка событий Asana: запрос, получение и обработка событий из сервиса Asana о действиях с задачами;
* повторная попытка отправки дополнительной задачи: если при вводе дополнительной задачи не удалось связаться с сервисом Asana.

Дерево функций программного средства представлено на рисунке В.1.

* + 1. Структурная схема

Программное средство учета дополнительных задач проектов ООО «Восток ИТ» будет отдельным веб-приложением.

Взаимодействие с пользовательским интерфейсом происходит через протокол HTTP, при этом передаются страницы гипертекстовой разметки (HTML) и каскадные таблицы стилей (CSS). Обмен данными с серверами Asana происходит также через протокол HTTP, с использованием предоставляемого сервисом Asana REST API, при этом данные передаются в формате JSON.

Программное средство запускается на сервере компании ООО «Восток ИТ» и работает непрерывно. Завершение работы программного средства предполагается только по внешним причинам: завершение процесса на уровне операционной системы, выключение сервера и т.д.

Для взаимодействия с базой данных используется встраиваемая система управления базами данных SQLite. Хранение базы данных происходит в виде файла в файловом хранилище сервера, на котором запущено программное средство.

Структурная схема программного средства представлена на рисунке Рисунок 14.

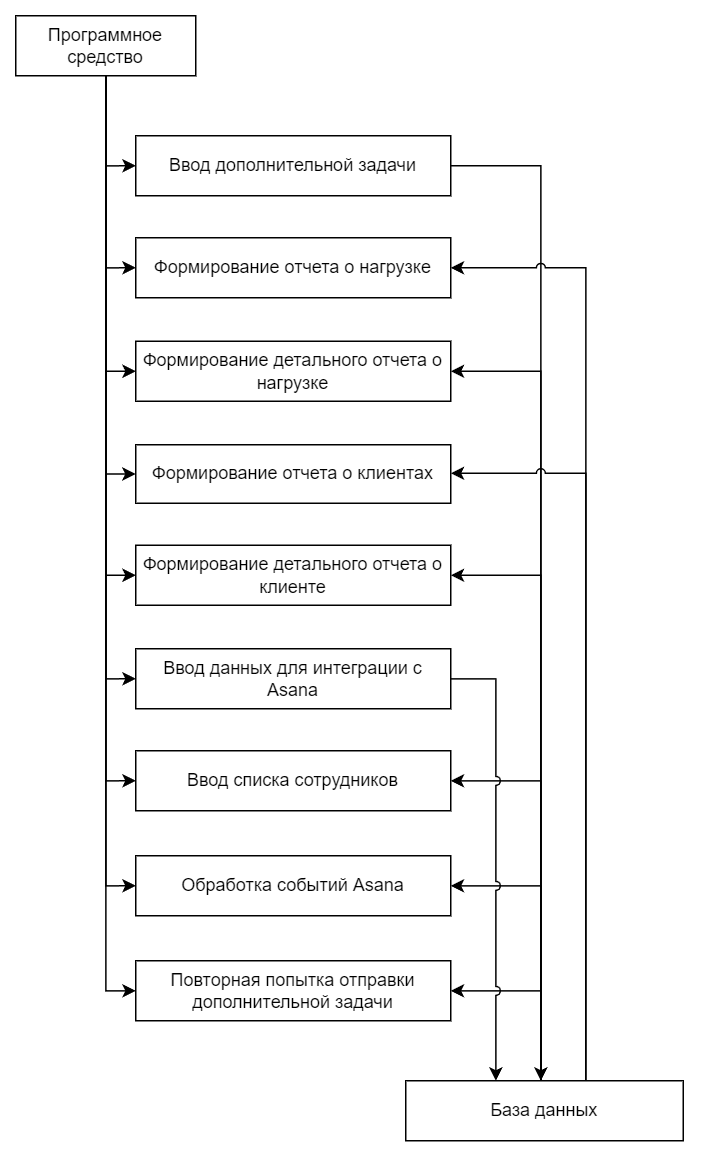


Рисунок 14 – Структурная схема

* + 1. Алгоритмы модулей

Алгоритм обработки событий Asana: выполняется запрос к сервису Asana для получения событий в основном проекте. Если запрос завершился с ошибкой, причина ошибки записывается в файл логов и алгоритм завершается. Если запрос успешен, но не имеет событий, алгоритм завершается. Если события есть, для каждого события происходит обработка в зависимости от типа события:

* удаление: задача в сервисе Asana была удалена – программное средство помечает соответствующую задачу в базе данных как удаленную;
* перемещение между колонками: задача в сервисе Asana была перемещена из одной колонки в другую – программное средство записывает в базу данных новый статус, связанный с соответствующей задачей;
* создание новой: в сервисе Asana создана новая задача – программное средство создает в базе данных новую задачу.

Блок-схема алгоритма обработки событий Asana представлен на рисунке Рисунок 15.

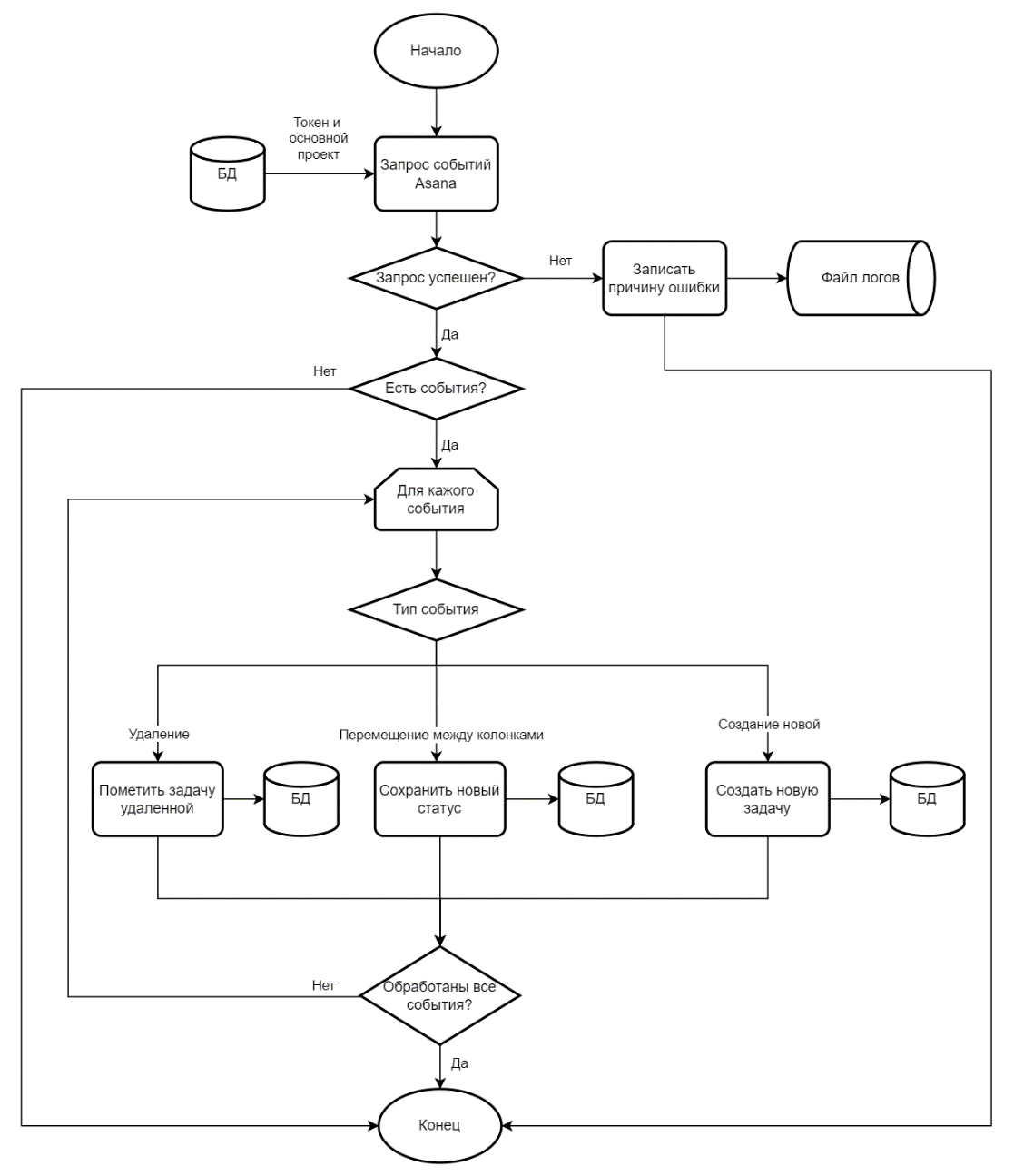


Рисунок 15 – Блок-схема алгоритма обработки событий Asana

Алгоритм настройки интеграции: отображается форма ввода токена Asana. Пользователь вводит токен. Выполняется запрос к сервису Asana для получения списка доступных пользователю проектов. Если запрос завершается с ошибкой, выводится сообщение о причине ошибки: неверный токен или недоступность сервиса Asana. Если запрос завершен удачно и получен список проектов, в базу данных сохраняется токен Asana и отображается выпадающий список для выбора основного проекта. Пользователь выбирает проект. В базу данных сохраняется идентификатор проекта в сервисе Asana.

Блок-схема алгоритма настройки интеграции представлена на рисунке Рисунок 16.

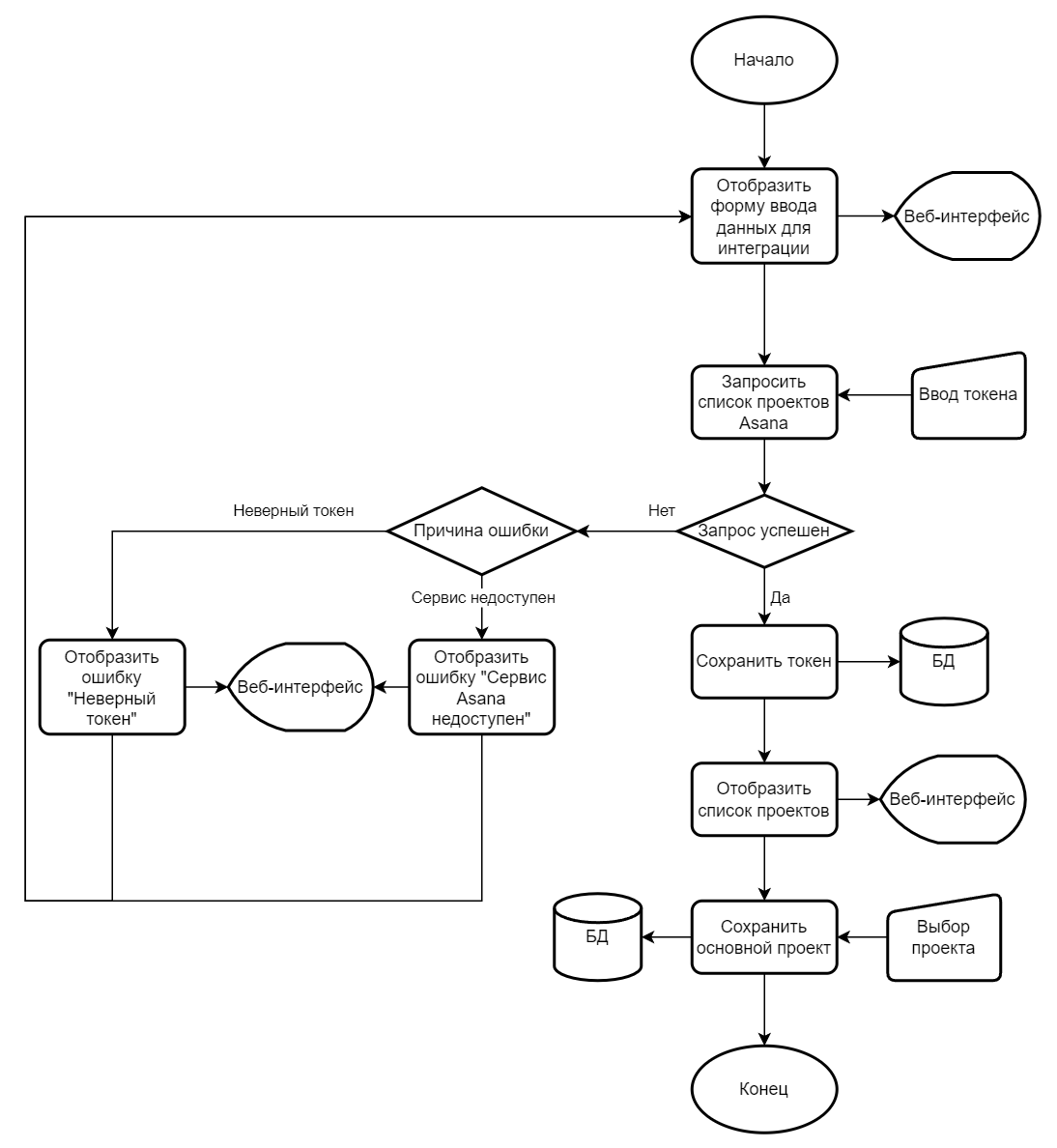


Рисунок 16 – Блок-схема алгоритма настройки интеграции

Алгоритм настройки сотрудников: выполняется запрос к сервису Asana для получения списка пользователей в основном проекте. Если запрос завершается с ошибкой, в пользовательском интерфейсе отображается информация об ошибке и алгоритм завершается. Если запрос выполнен успешно и получен список пользователей, отображается каждый пользователь и поле ввода ФИО сотрудника. Пользователь вводит ФИО соответствующих пользователей. Информация о сотрудниках сохраняется в базу данных.

Блок-схема алгоритма настройки сотрудников представлена на рисунке Рисунок 17.

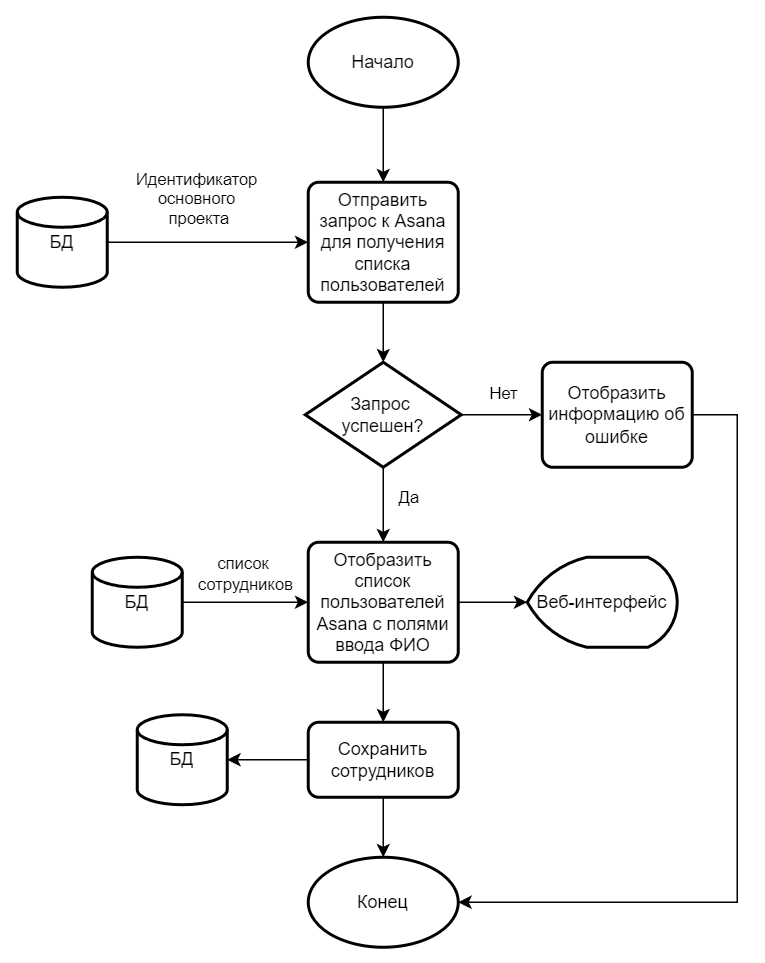


Рисунок 17 – Блок-схема алгоритма настройки сотрудников

Алгоритм ввода дополнительной задачи: отображается форма ввода дополнительной задачи с выпадающем списком сотрудников. Пользователь вводит данные дополнительной задачи. Задача и данные о ней сохраняются в базу данных. Выполняется запрос к сервису Asana для создания задачи. Если запрос завершается неудачей, причина ошибки записывается в файл логов и алгоритм завершается. Если запрос завершается успешно, в базу данных для задачи записывается идентификатор задачи из сервиса Asana.

Блок-схема алгоритма ввода дополнительной задачи представлена на рисунке Рисунок 18.

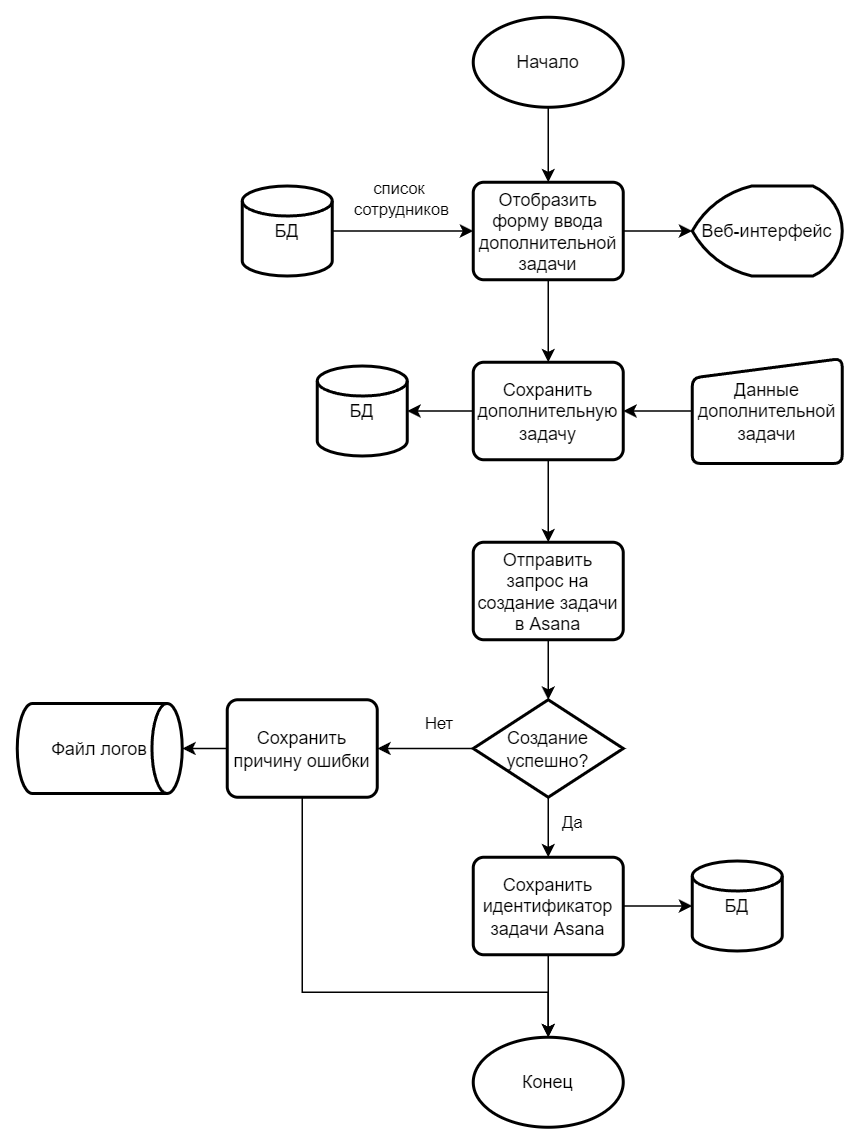


Рисунок 18 – Блок-схема алгоритма ввода дополнительной задачи

Алгоритм попытки повторной отправки дополнительной задачи в сервис Asana: из базы данных получается список задач без идентификатора Asana. Если список пуст, алгоритм завершается. Если в списке есть задачи, то для каждой задачи происходит запрос на создание в сервисе Asana. Если запрос завершается неудаче, в файл логов записывается причина ошибки. Если запрос завершается успешно, в базу данных для задачи записывается идентификатор задачи из сервиса Asana.

Блок-схема алгоритма попытки повторной отправки дополнительной задачи в сервис Asana представлен на рисунке Рисунок 19.

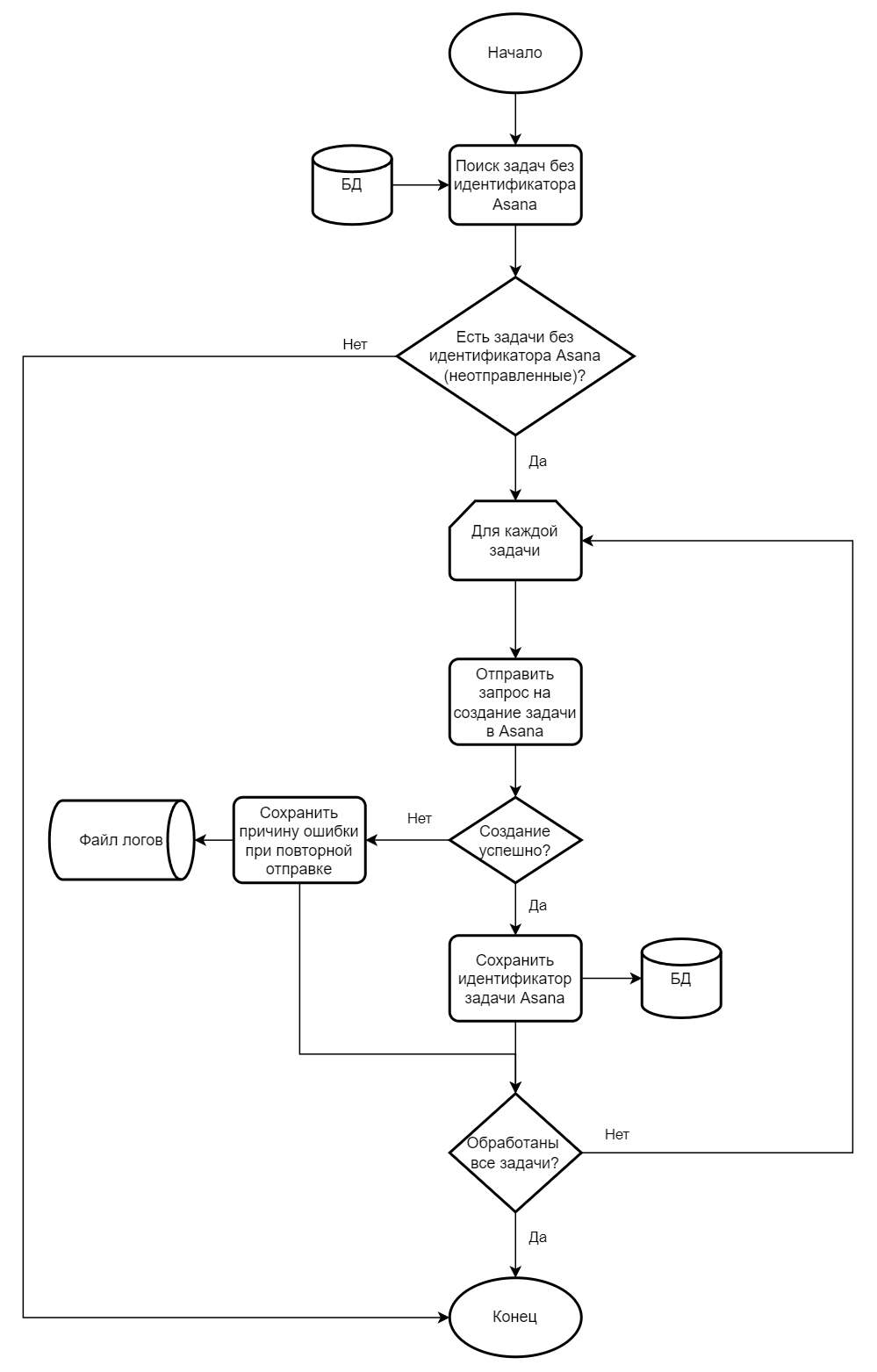


Рисунок 19 – Блок-схема алгоритма попытки повторной

отправки дополнительной задачи

Алгоритм формирования отчета: для всех приведенных отчетов алгоритм формирования схож за исключением данных, выбираемых из базы данных. Отображается форма отчета с выпадающим списком сотрудников или клиентов. Пользователь вводит период формирования отчета и выбирает клиента или сотрудника. Из базы данных выбираются необходимые данные и рассчитывается сводная информация. Отчет отображается в пользовательском интерфейсе.

Блок-схема алгоритма формирования отчета представлена на рисунке Рисунок 20.

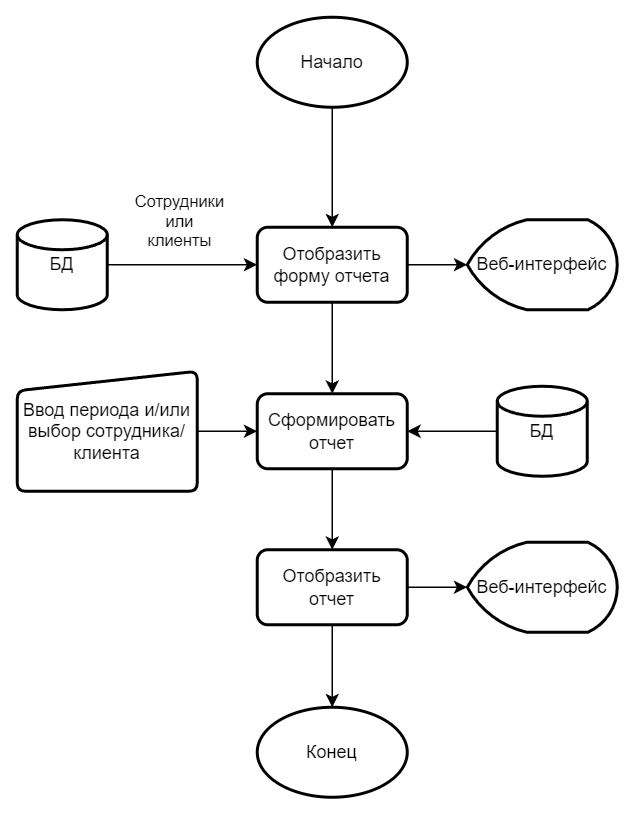


Рисунок 20 – Блок-схема алгоритма формирования отчета

* + 1. Общие положения

Для разработки программного средства, осуществляющего запросы к сервису Asana и выдаче сводных отчетов, был выбран язык программирования Python версии 3.13.2. Этот язык был выбран в качестве основного языка благодаря его высокой читаемости, обширной экосистеме библиотек и активному сообществу разработчиков. Эти факторы значительно ускоряют процесс разработки и упрощают поддержку кода. Python обладает мощными инструментами для работы с HTTP-запросами и асинхронным программированием, что критически важно для взаимодействия с внешними API.

Были использованы библиотеки FastAPI, aiohttp, Jinja2 и SQLAlchemy. Библиотека FastAPI была использована в качестве основы для создания веб-интерфейса. FastAPI также обеспечивает простую интеграцию с другими библиотеками, что упрощает обработку входящих и исходящих HTTP-запросов. Для выполнения асинхронных HTTP-запросов к Asana была выбрана библиотека aiohttp, которая предоставляет удобный и эффективный инструментарий для работы с сетевыми запросами в асинхронном режиме, минимизируя задержки и повышая отзывчивость системы. Библиотека для шаблонизации Jinja2 была применена для генерации отчетов и пользовательского интерфейса в виде динамического HTML-контента. Гибкость и простота интеграции с FastAPI делают её оптимальным выбором для рендеринга веб-страниц. Для взаимодействия с базой данных выбрана библиотека SQLAlchemy, которая предоставляет мощный и гибкий инструментарий для работы с реляционными базами данных. Выбор SQLAlchemy обусловлен её поддержкой как низкоуровневого SQL-синтаксиса, так и высокоуровневого объектно-ориентированного подхода через ORM, что существенно упрощает разработку и делает код более читаемым.

* 1. Описание интерфейса системы

Пользовательский интерфейс программного средства реализован в виде веб-приложения, что обеспечивает удобный и доступный способ взаимодействия с системой через стандартный браузер. Такой подход исключает необходимость установки дополнительного клиентского ПО, снижая порог вхождения для пользователей и упрощая развёртывание решения. Веб-интерфейс также обеспечивает кроссплатформенную совместимость, позволяя работать с системой с любых устройств, без ограничений по операционной системе.

Карта веб-приложения представлена на рисунке Рисунок 21.

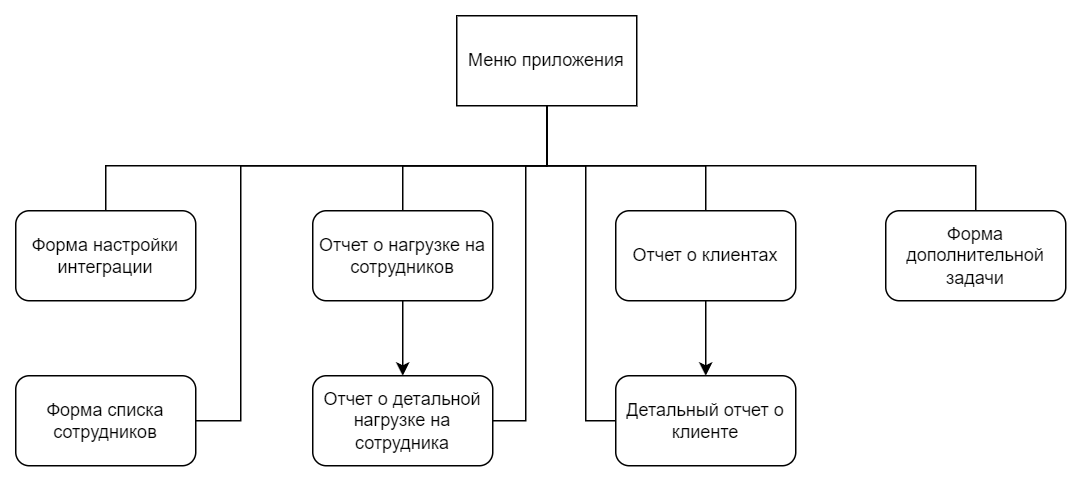


Рисунок 21 – Карта веб-приложения

Пользовательский интерфейс программного средства имеет единую структуру, состоящую из двух логических разделов:

* навигационная панель – обеспечивает переход между страницами системы., меню является статическим и сохраняет единый вид на всех страницах, что способствует удобству ориентации пользователя;
* контентная область – содержит основное содержимое страницы, включая интерактивные элементы (формы ввода данных, отчеты и другие компоненты в зависимости от функционального назначения страницы).

При доступе на коренной адрес происходит переадресация: если данные для интеграции с сервисом Asana уже введены, то на страницу отчета о нагрузке, иначе на страницу настройки интеграции.

* + 1. Характеристика входных форм

Для ввода нормативно-справочной и входной оперативной информации используются экранные формы для ввода данных для интеграции с сервисом Asana, ввода ФИО сотрудников и ввода дополнительной задачи.

Форма ввода данных для интеграции с сервисом Asana. Форма представлена на рисунке В.1. Форма включает:

* поле ввода токена Asana;
* выбор основного отслеживаемого проекта – выпадающий список доступных проектов Asana (список проектов запрашивается у сервиса Asana после ввода токена).

Форма ввода ФИО сотрудников. Форма представлена на рисунке В.2. Форма делает запрос к сервису Asana для получения списка пользователей, находящихся в основном проекте, выбранном в форме ввода данных для интеграции, и выводит для каждого пользователя его имя в сервисе Asana и поле ввода ФИО. Если данные для интеграции не были введены, форма показывает предупреждение вместо основных полей.

Форма ввода дополнительной задачи. Форма представлена на рисунке В.3. Форма включает:

* сотрудник, назначаемый на задачу – выпадающий список (использует справочник сотрудников);
* поле ввода идентификатора наряда из CRM-системы К7;
* поле ввода наименования клиента;
* сотрудник, ответственный за наряд;
* описание задач.
  + 1. Характеристика результатных форм

Результатная информация представлена отчетами об общей нагрузке на сотрудников за период, детальной нагрузке на сотрудника за период, о дополнительных задачах по клиентам за период и детальный отчет о дополнительной нагрузке по клиенту за период.

Отчет о нагрузке на сотрудников за период: предназначен для анализа распределения задач между сотрудниками за указанный период. Используется для выявления перегруженных специалистов и более равномерного распределения рабочих задач. Отчет формируется по запросу пользователя за произвольно указанный период. Представляется в виде вывода в интерфейс системы. Формируется на основе таблиц «Employee», «Ticket», «Status» и «Additional info». Отчет представлен на рисунке В.4.

Отчет включает поля:

* ФИО сотрудника;
* общее количество задач;
* количество завершенных задач;
* количество незавершенных задач;
* количество дополнительных задач;
* среднее время выполнения задачи.

Отчет о детальное нагрузке на сотрудника за период: предназначен для анализа нагрузки на сотрудника. Используется для детального анализа задач сотрудника по статусам, срокам исполнения и самому описанию задачи. Отчет формируется по запросу пользователя за произвольно указанный период. Представляется в виде вывода в интерфейс системы. Формируется на основе таблиц «Employee», «Ticket», «Status» и «Additional info». Отчет представлен на рисунке В.5.

Отчет включает поля:

* ФИО сотрудника;
* общее количество задач;
* количество завершенных задач;
* количество незавершенных задач;
* среднее время выполнения задачи;
* наименование статуса;
* количество задач в статусе;
* количество дополнительных задач;
* заголовок задачи;
* текст задачи;
* длительность нахождения в статусе.

Отчет о дополнительных задачах по клиентам за период: используется для анализа активности по нестандартным, дополнительным задачам, связанным с конкретными клиентами. Это позволяет выявлять клиентов, требующих чрезмерного внимания или имеющих скрытые проблемы в обслуживании. Отчет формируется по запросу пользователя за произвольно указанный период. Представляется в виде вывода в интерфейс системы. Формируется на основе таблиц «Employee», «Ticket», «Status» и «Additional info». Отчет представлен на рисунке В.6.

Отчет содержит следующие поля:

* название клиента;
* количество дополнительных задач;
* количество сотрудников, задействованных в дополнительных задачах клиента;
* средняя продолжительность дополнительных задач.

Детальный отчет о дополнительных задачах по клиенту за период: используется для анализа дополнительных задач клиентам. Это позволяет выявить причины сложности работы с данным клиентом. Отчет формируется по запросу пользователя за произвольно указанный период. Представляется в виде вывода в интерфейс системы. Формируется на основе таблиц «Employee», «Ticket», «Status» и «Additional info». Отчет представлен на рисунке В.7.

Отчет содержит следующие поля:

* название клиента;
* общее количество дополнительных задач;
* количество выполненных дополнительных задач;
* количество невыполненных дополнительных задач;
* количество задействованных сотрудников;
* заголовок задачи;
* описание задачи;
* идентификатор наряда в CRM-системе К7;
* ответственный по наряду;
* исполнитель задачи;
* время выполнения выполненной задачи.
  1. Технологическое обеспечение задачи

Организация технологического процесса работы программного средства учета дополнительных задач проектов:

Первым действием по работе с программным средством является настройка интеграции с сервисом Asana.

После ввода данных для интеграции с сервисом Asana создается цикл, одновременно запрашивающий события сервиса Asana и обрабатывающий повторные отправки дополнительных задач. Цикл выполняется каждые 5 секунд. Одновременно с циклом происходит обработка запросов от пользователя на формирование отчетов или ввод данных.

Благодаря использованию Python библиотеки FastAPI параллельное выполнение достигается за счет асинхронности, которая позволяет эффективно обрабатывать множество запросов без блокировки основного потока приложения: когда один процесс ожидает (например, ответа от базы данных или запроса к внешнему сервису), среда выполнения моментально переключается на обработку других процессов, что значительно повышает производительность и отзывчивость системы. Это является серьезным преимуществом для веб-приложения, позволяя быстро реагировать на действия пользователей.

Технологическая схема обработки событий из Asana представлена на рисунке Рисунок 22.

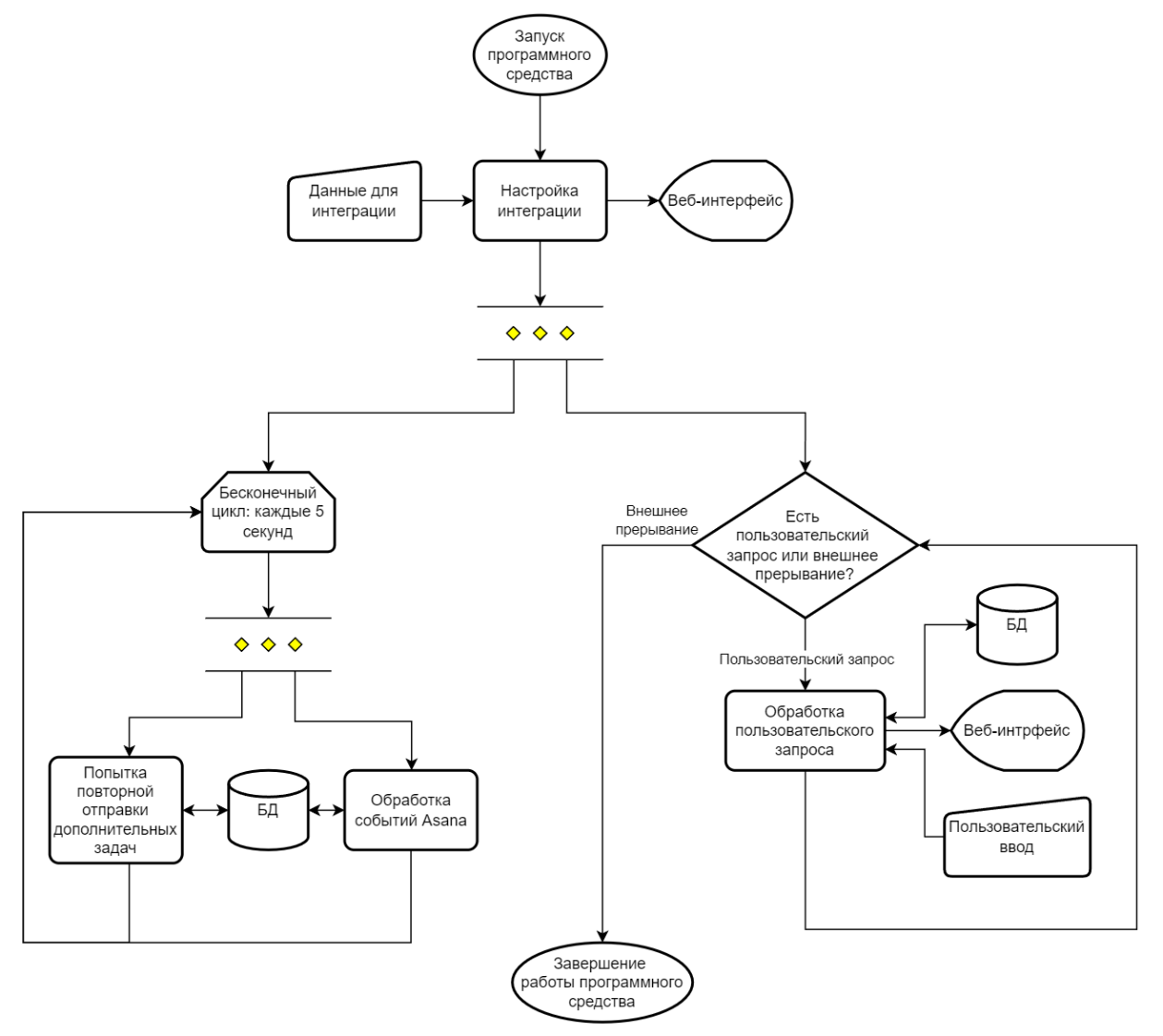


Рисунок 22 – Технологическая схема обработка событий Asana

* 1. Обоснование экономической эффективности проекта
     1. Выбор метода расчета экономической эффективности

Оценка экономической эффективности разрабатываемого аналитического программного средства представляет собой задачу, отличающуюся от классического финансового анализа, так как его внедрение не приводит к прямым финансовым результатам, таким как рост выручки или снижение себестоимости. Ценность такого программного средства заключается в улучшении управленческих решений за счет анализа распределения задач, выявления проблемных клиентов и оптимизации загрузки сотрудников.

Классические подходы к оценке экономической эффективности, такие как финансовые расчеты и методы инвестиционного не позволяют в полной мере отразить косвенные выгоды от внедрения инструмента учета. Такие методы ориентированы на явные денежные потоки, но данном случае ключевые преимущества лежат в области повышения прозрачности управления, снижения операционных рисков и улучшения стратегического планирования.

В связи с этим для оценки экономической эффективности был выбран качественный метод оценки – Система сбалансированных показателей (ССП), которая позволяет учесть не только финансовые, но и клиентские и процессные аспекты, а также аспекты развития. Метод ССП основан на причинно-следственных связях между ключевыми показателями эффективности (KPI) и стратегическими целями компании.

* + 1. Расчет экономической эффективности

Основные проекции системы сбалансированных показателей ООО «Восток ИТ»:

* финансы: снижение издержек и повышение прибыли;
* клиенты: повышение скорости обслуживания и выявление сложных случаев;
* внутренние процессы: оптимизация эффективности процессов;
* развитие: автоматизация процессов.

Основные цели по проекциям:

* финансы:
  + повысить прибыль компании;
  + повысить рентабельность сложных проектов;
  + снизить издержки на выполнение проектов;
* клиенты:
  + снизить время выполнения проектов;
  + выявить сложных клиентов;
* внутренние процессы:
  + повысить эффективность распределения задач;
  + повысить качество анализа задач;
  + сократить время на анализ задач;
* развитие:
  + автоматизировать процессы анализа.

Карта системы сбалансированных показателей представлена на рисунке Рисунок 23.

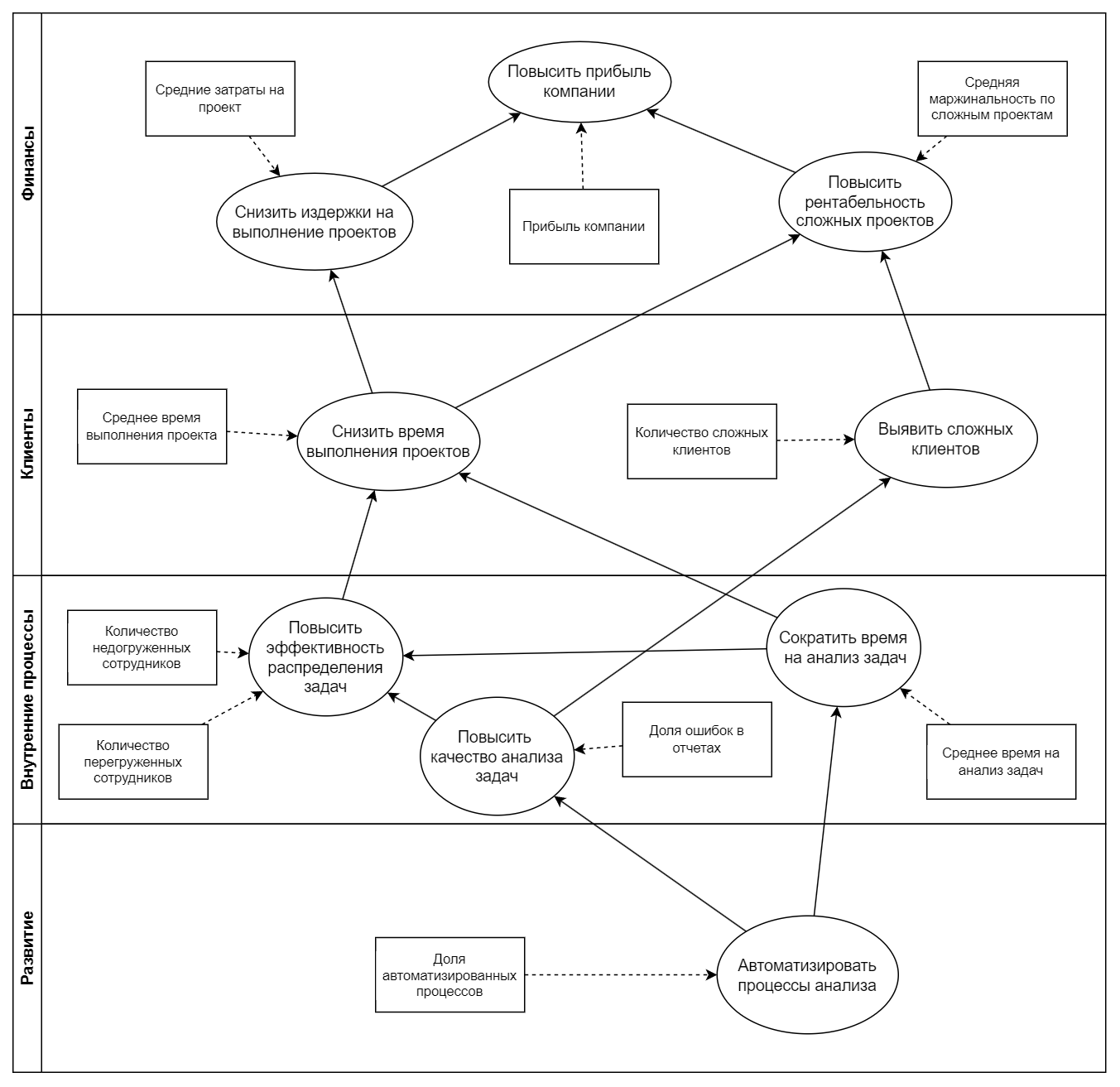


Рисунок 23 – Карта системы сбалансированных показателей

Автоматизация процессов анализа позволит сократить время анализа и повысить его качество, что позволит более эффективно распределять задачи, что, в свою очередь, снизит издержки на проекты и повысит прибыль компании.

Процедуры измерения и расчета параметров системы сбалансированных показателей представлены в таблице Таблица 18.

Таблица 18 – Процедуры измерения и расчета параметров

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название показателя | Цель | Единицы измерения | Период оценки | Формула для расчета |
| Прибыль компании | Повысить прибыль компании | Рубли | Месяц / квартал | Прибыль – расходы - налоги |
| Средние затраты на проект | Снизить издержки на выполнение проектов | Рубли | Месяц / квартал | Сумма всех прямых затрат по проектам / количество проектов |
| Средняя маржинальность по сложным проектам | Повысить рентабельность сложных проектов | Проценты (%) | Месяц / квартал | Сумма((Доход за проект − Прямые затраты на проект) / Доход за проект × 100%) / количество проектов) |
| Среднее время выполнения проекта | Снизить время выполнения проектов | Часы | Месяц / квартал | Сумма(Время выполнения проекта) / количество выполненных проектов |
| Количество сложных клиентов | Выявить сложных клиентов | Штуки | Месяц / квартал | Количество клиентов, по проектам для которых дополнительных задач больше 2 |
| Количество недогруженных сотрудников | Повысить эффективность распределения задач | Штуки | Месяц / квартал | Количество сотрудников, у которых количество задач меньше среднее − σ (стандартное отклонение) |
| Количество перегруженных сотрудников | Повысить эффективность распределения задач | Штуки | Месяц / квартал | Количество сотрудников, у которых количество задач больше среднее + σ (стандартное отклонение) |
| Доля ошибок в отчетах | Повысить качество анализа задач | Проценты (%) | Месяц / квартал | Количество корректных показателей в отчете / общее количество показателей в отчете |
| Среднее время на анализ задач | Сократить время на анализ задач | Часы | Месяц / квартал | Сумма(Время анализа задач) / Количество проведенных анализов задач |
| Доля автоматизированных процессов | Автоматизировать процессы анализа | Проценты (%) | Месяц / квартал | (Количество автоматизированных процессов / общее количество процессов) \* 100% |

* 1. Описание контрольного примера реализации проекта

Компания имеет проект в сервисе Asana. Проект без задач представлен на рисунке Д.1.

При обнаружении дополнительной задачи сотрудник отела проектов заполняет данные о ней. Ввод дополнительной задачи представлен на рисунке Д.2.

При сохранении задачи она автоматически передается сервису Asana. Проект с новой задачей представлен на рисунке Д.3.

При работе с задачей сотрудник перемещает ее в колонках канбан-доски, изменяя её статус (Рисунок Д.4). Программное средство получает информацию об изменении статуса и при формировании отчета за соответствующий период отображаются актуальные данные о задаче. Отчет о нагрузке на сотрудника представлен на рисунке Д.5.

При завершении работы над задачей сотрудник удаляет её в сервисе Asana (Рисунок Д.6). Программное средство получает информацию об этом событии и устанавливает на соответствующую задачу отметку о завершении. Благодаря этому в отчете отображается информация о завершенной задаче, а также остается возможность отслеживания её изменений при формировании отчета за прошедшие периоды. Отчет о нагрузке на сотрудника с завершенной задачей представлен на рисунке Д.7.

Данный пример показывает, как программное средство позволяет отслеживать актуальное состояние задач.

Заключение

В ходе выполнения выпускной квалификационной работы была решена задача автоматизации процесса учета проектных задач в ООО «Восток ИТ». Были проанализированы текущие бизнес-процессы предприятия, выявлены основные проблемы, связанные с высокой долей ручной обработки данных, риском возникновения ошибок и дублированием информации при использовании сервисов Яндекс Формы и Excel в процессах анализа задач проектов. На основании проведенного анализа была спроектировано и реализовано веб-приложение, обеспечивающее автоматизированную интеграцию с сервисом Asana, сбор и обработку данных, а также формирование сводных отчетов.

В рамках реализации использовался функционально-ориентированный подход и построены модели бизнес-процессов в нотации IDEF0, а также информационные модели в нотации DFD и IDEF1X.

Предложенное решение представляет собой программное средство, адаптированное под внутренние процессы компании, без необходимости замены существующих сервисов управления проектами. Программное средство способно не только отображать текущую загрузку сотрудников, но и сохранять историю задач, что делает возможным проведение анализа истории задач и выявление тенденций в распределении нагрузки.

Экономический эффект от внедрения выражается в снижении трудозатрат на формирование отчетности, сокращении числа ошибок, повышении качества и скорости принятия управленческих решений и, как следствие, повышения скорости выполнения проектов и увеличения прибыли компании.

Разработанное программное средство было внедрено в компании ООО «Восток ИТ». Потенциальные направления совершенствования включают расширение существующих отчетов и добавления новых, добавления визуализации в виде графиков и интеграции с другими системами.

Таким образом, в ходе выполнения выпускной квалификационной работы решены поставленные задачи и достигнута поставленная цель.

Список литературы

**Нормативно-справочные документы**

1. ГОСТ Р ИСО/МЭК 12207–2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. — Введ. 2012–03–01. — М. : Стандартинформ, 2011. — 105 с.
2. ГОСТ 7.1–2003. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – Введ. 2004–07–01. – М.: Стандартинформ, 2004. – 23 с.
3. ГОСТ 2.105–2019. Единая система конструкторской документации. Общие требования к текстовым документам. — Введ. 2019–07–01. — М. : Стандартинформ, 2019. — 40 с.
4. ГОСТ 7.32–2017. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. — Введ. 2018–01–01. — М. : Стандартинформ, 2017. — 28 с.

**Учебная и научная литература**

1. Аниче М. Простое объектно-ориентированное проектирование: чистый и гибкий код – Издательский дом «Питер», 2025 – 224 с.
2. Шестаков А.В., Орехова Е.Л. UML, IDEF, BPMN: методы моделирования бизнес-процессов и проектирования ИС. – М.: Инфра-М, 2018. – 326 с.
3. Хэлл К., Датта А. Проектирование информационных систем. – СПб.: Питер, 2019. – 416 с.
4. Котельникова А.В., Гончарова Н.И. Управление проектами. Учебник и практикум. – М.: Юрайт, 2022. – 364 с.
5. Локшин Б.С., Дементьев В.И. Управление ИТ-проектами: методология, практика, контроль. – М.: КНОРУС, 2021. – 312 с.
6. Бурда А.Ю., Капустина О.А. Управление ИТ-проектами в гибридной среде: Agile, Waterfall, MS Project. – М.: Юрайт, 2022. – 310 с.
7. Назаров А.И., Шевченко Л.Н. Проектирование информационных систем. Учебник для вузов. – М.: Форум, 2020. – 368 с.
8. Гвоздева Т. В., Баллод Б. А. Проектирование информационных систем: методы и средства структурно-функционального проектирования: практикум: учебное пособие для СПО. — 3-е изд., стер. — Санкт-Петербург: Лань, 2024. — 148 с.
9. Хаммер М., Чампи Дж. Реинжиниринг корпорации: Манифест революции в бизнесе. – М.: Альпина Паблишер, 2018. – 332 с.
10. Титов В.А., Костров А.Н. Методология и практика оценки эффективности ИТ-проектов. — СПб.: Питер, 2019. — 240 с.
11. Лапин В.В. Экономика информационных систем: оценка эффективности и стоимости проектов. — М.: КНОРУС, 2020. — 272 с.
12. Поллард Б., Бомбакова П. HTTP/2 в действии. – ДМК-Пресс, 2021 – 424 с.

**Электронные ресурсы**

1. Официальный сайт Первый Бит [Электронный ресурс]. – Режим доступа: <https://vladivostok.1cbit.ru/>
2. Официальный сайт Asana [Электронный ресурс]. – Режим доступа: [https://asana.com/ru](https://asana.com/rua)
3. Официальный сайт Jira [Электронный ресурс]. – Режим доступа: <https://www.atlassian.com/software/jira>
4. Официальный сайт ПланФикс [Электронный ресурс]. – Режим доступа: <https://planfix.ru/main/>
5. Лучшие канбан-доски: онлайн-инструменты для ведения проектов [Электронный ресурс]. – Режим доступа: <https://www.leadertask.ru/blog/kanban-doska>
6. RUP методология разработки [Электронный ресурс]. – Режим доступа: <https://qaevolution.ru/metodologiya-menedzhment/rup/>
7. RUP (Rational Unified Process) [Электронный ресурс]. – Режим доступа: <https://appmaster.io/ru/glossary/rup-rational-unified-process-ru>
8. Asana API Reference [Электронный ресурс]. – Режим доступа: <https://developers.asana.com/reference/rest-api-reference>
9. Документация по Project. Документация по Project для администраторов и ИТ-специалистов [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/project/>

Приложение А

**Модель бизнес-процессов «AS-IS»**

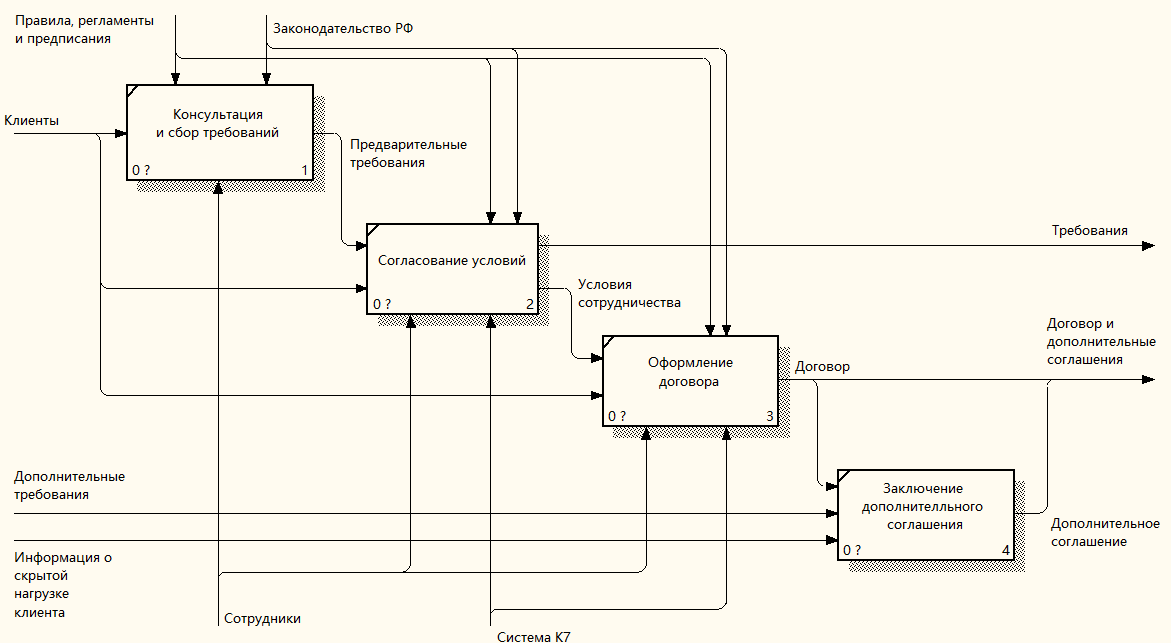


Рисунок А.1 – Декомпозиция процесса «Работа с клиентами»

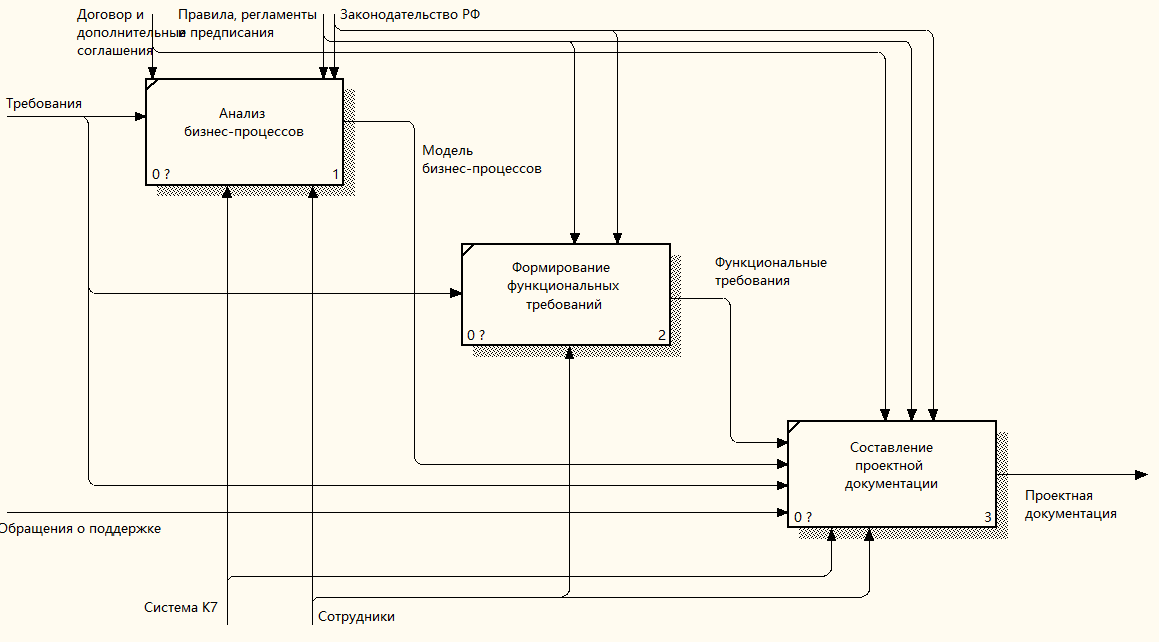


Рисунок А.2 – Декомпозиция процесса «Проектирование решений»

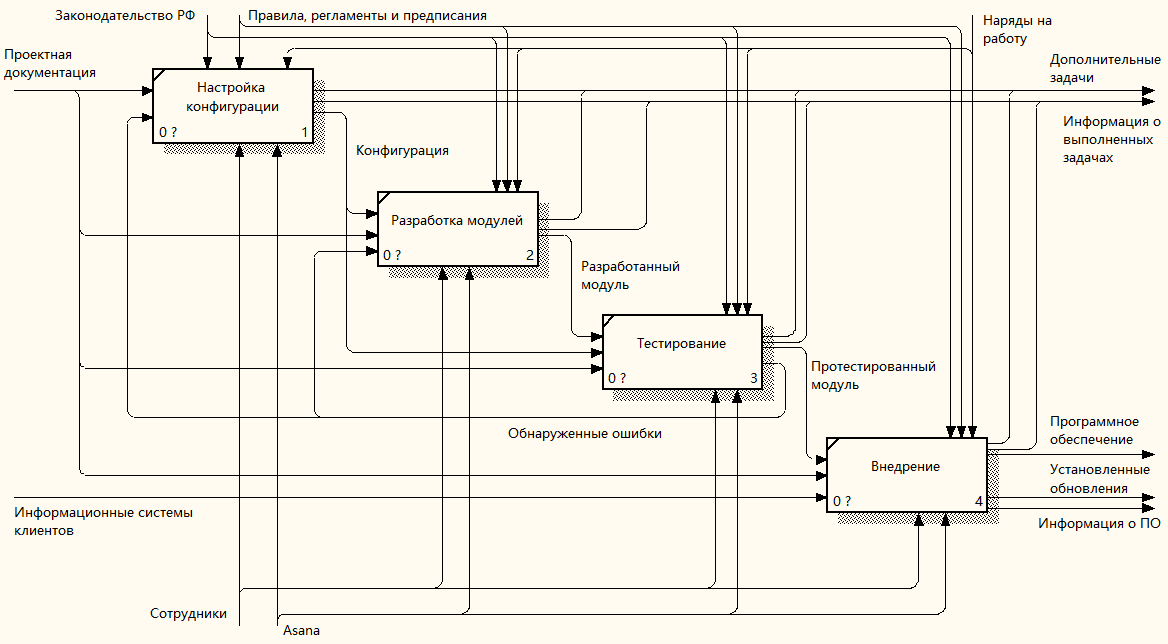


Рисунок А.3 – Декомпозиция процесса «Разработка и внедрение»

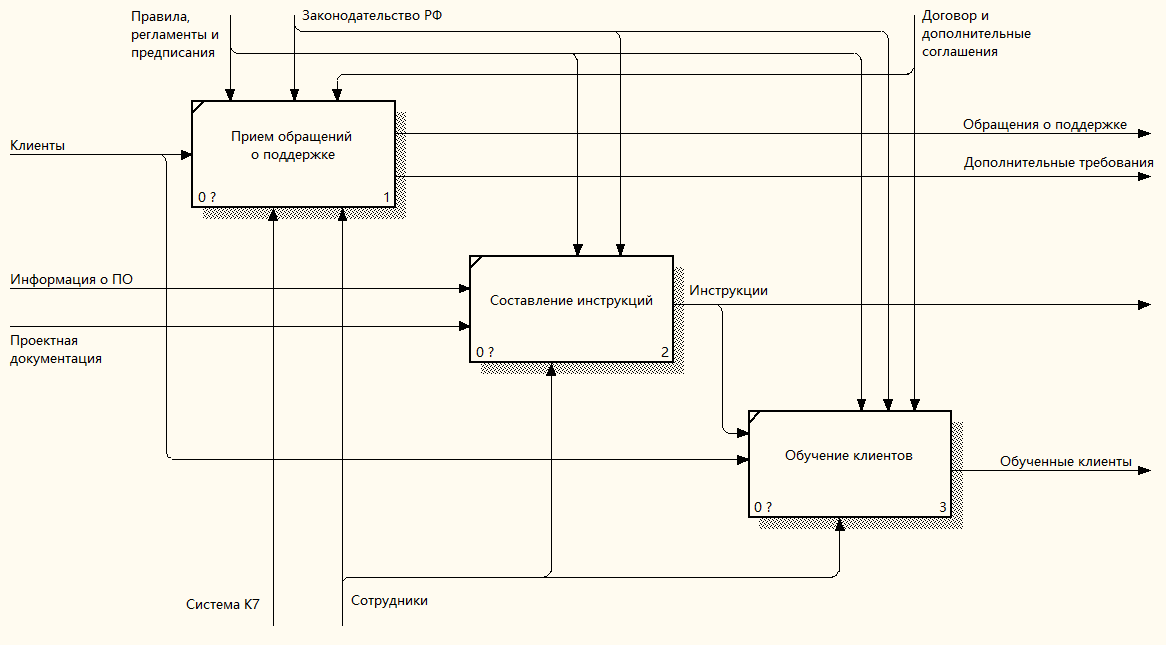


Рисунок А.4 – Декомпозиция процесса «Сопровождение и поддержка»

**Приложение Б**

**Календарный план**

Таблица Б.1 – Состав проекта

|  |  |
| --- | --- |
| Наименование | Описание |
| Составление плана проекта | Этап, включающий в себя планирование проекта |
| Определение содержания | Формирование списка задач проекта |
| Определение длительности работ | Определение длительности каждой задачи по методу PERT |
| Планирование ресурсов и затрат | Формирование списка ресурсов, их ставок и назначение на задачи |
| Идентификация рисков и разработка стратегии их смягчения | Анализ плана проекта, выделение рисков и их параметров, составление плана реакции на риски |
| Завершение составления плана | Веха проекта – завершение этапа планирования |
| Анализ требований и проектирование | Этап, включающий в себя определение требований и проектирование системы |
| Интервьюирование заказчика | Проведение интервью с заказчиком, анализ целей проекта, формирование требований |
| Бизнес-моделирование | Бизнес-анализ предметной области: деятельность, объекты |
| Разработка системных требований | Определение функции проектируемой системы |
| Анализ системы | Анализ системных прецедентов: ключевые абстракции, последовательности |
| Проектирование архитектуры | Проектирование архитектуры ИС: подсистемы, модули, физическая реализация |
| Проектирование компонентов | Определение характеристик классов и моделирование баз данных |
| Проектирование интерфейса | Проектирование интерфейсных компонентов и системы навигации по этим компонентам |
| Завершение проектирования | Веха проекта – завершение этапа проектирования |
| Разработка | Этап, включающий в себя разработку системы |
| Backend программирование | Разработка логики серверной части ИС |
| Разработка базы данных | Выбор и настройка системы управления базами данных |

*Окончание таблицы Б.1*

|  |  |
| --- | --- |
| Наименование | Описание |
| Верстка страниц | Разработка web-страниц с использованием гипертекстовой разметки и таблиц стилей |
| Frontend программирование | Разработка логики клиентской части ИС |
| Интеграция frontend и backend | Разработка элементов логики, обеспечивающих взаимодействие клиентской и серверной части |
| Завершение разработки | Веха проекта – завершение этапа разработки |
| Тестирование и исправление ошибок | Этап, включающий в себя тестирование и исправление ошибок системы |
| Разработка модульных тестов | Разработка модульных тестов для автоматической проверки функциональности (unit-тестирование) |
| Ручное тестирование | Ручная проверка работоспособности функциональности системы |
| Исправление ошибок | Анализ причин и исправление ошибок, выявленных при тестировании |
| Завершение тестирования и исправления | Веха проекта – завершение этапа тестирования |
| Внедрение | Этап, включающий в себя подготовку и запуск системы |
| Подготовка пакета системы | Подготовка ИС к запуску с помощью выбранного сервиса |
| Подготовка инструкций для заказчика | Формирование инструкций для заказчика по использованию разработанной ИС |
| Запуск системы и проверка качества | Запуск ИС на удаленном сервере, проверка работоспособности всех элементов системы |
| Завершение внедрения | Веха проекта – завершение этапа внедрения |
| Сдача проекта | Представление результатов заказчику, передача инструкций |

Таблица Б.2 – Оценка продолжительности работ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Виды работ | Оптимистичная оценка, Tmin, дней | Реалистичная оценка, Tнв, дней | Пессимистичная оценка, Tmax, дней | Ожидаемая продолжительность, Tож, дней |
| Определение содержания | 1 | 2 | 3 | 2 |
| Определение длительности работ | 1 | 2 | 3 | 2 |
| Планирование ресурсов и затрат | 1 | 2 | 3 | 2 |
| Идентификация рисков и разработка стратегии их смягчения | 1 | 2 | 3 | 2 |
| Интервьюирование заказчика | 1 | 2 | 3 | 2 |
| Бизнес-моделирование | 2 | 6 | 10 | 6 |
| Разработка системных требований | 3 | 6 | 9 | 6 |
| Анализ системы | 5 | 12 | 19 | 12 |
| Проектирование архитектуры | 3 | 6 | 9 | 6 |
| Проектирование компонентов | 3 | 6 | 9 | 6 |
| Проектирование интерфейса | 3 | 6 | 9 | 6 |
| Backend программирование | 20 | 30 | 40 | 30 |
| Разработка базы данных | 6 | 12 | 18 | 12 |
| Верстка страниц | 2 | 8 | 14 | 8 |
| Frontend программирование | 6 | 16 | 26 | 16 |

*Окончание таблицы Б.2*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Виды работ | Оптимистичная оценка, Tmin, дней | Реалистичная оценка, Tнв, дней | Пессимистичная оценка, Tmax, дней | Ожидаемая продолжительность, Tож, дней |
| Интеграция frontend и backend | 2 | 6 | 10 | 6 |
| Разработка модульных тестов | 6 | 12 | 18 | 12 |
| Ручное тестирование | 10 | 18 | 26 | 18 |
| Исправление ошибок | 4 | 18 | 32 | 18 |
| Подготовка пакета системы | 1 | 3 | 5 | 3 |
| Подготовка инструкций для заказчика | 2 | 3 | 4 | 3 |
| Запуск системы и проверка качества | 2 | 6 | 10 | 6 |
| Сдача проекта | 0,5 | 1 | 1,5 | 1 |

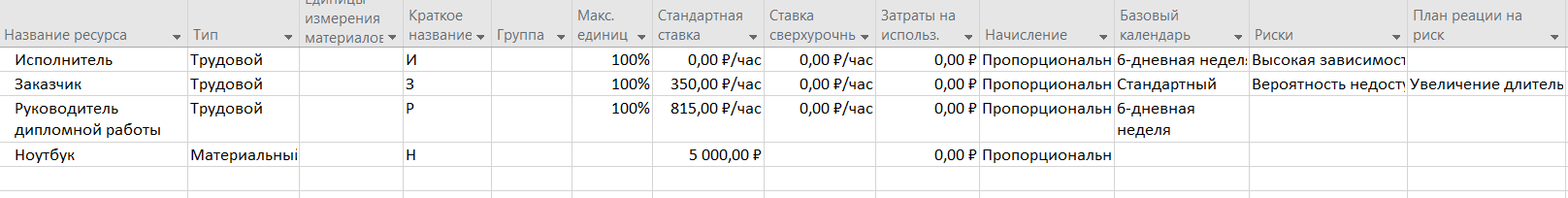


Рисунок Б.1 – Ресурсы проекта

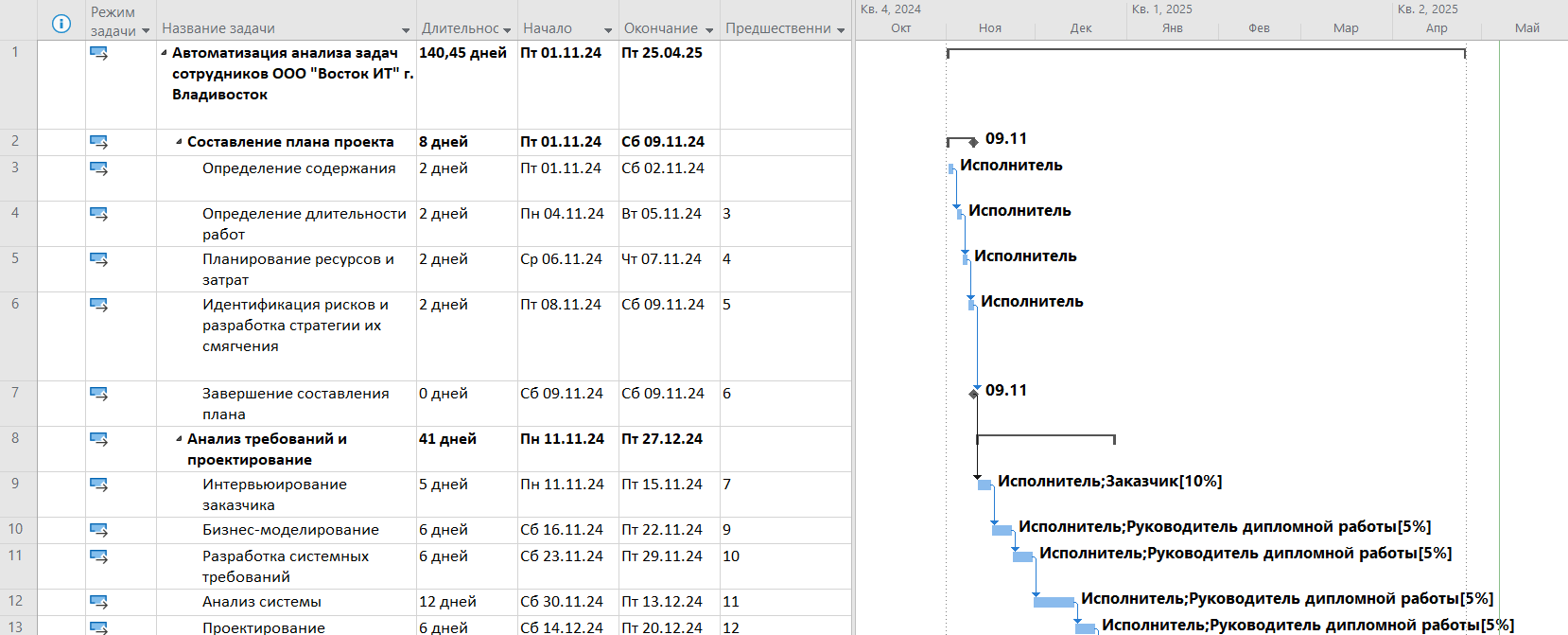


Рисунок Б.2 – Содержание проекта

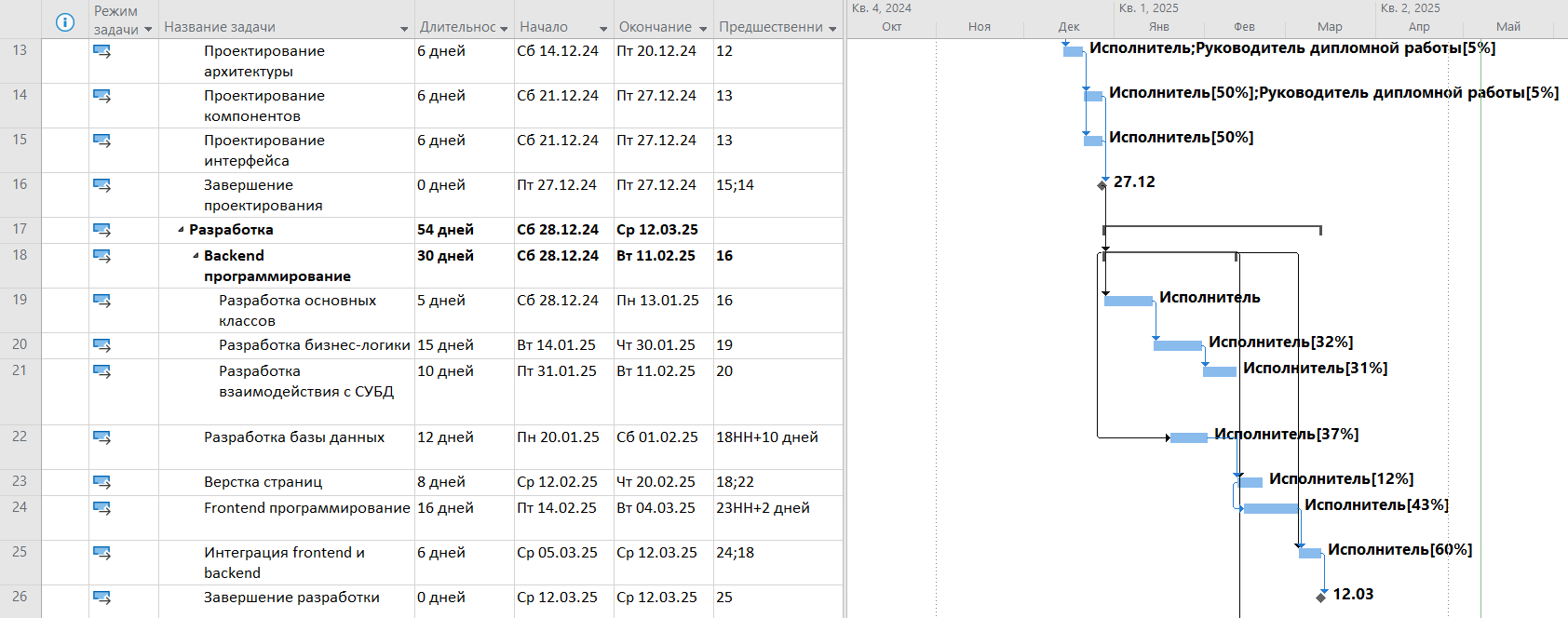


Рисунок Б.3 – Содержание проекта

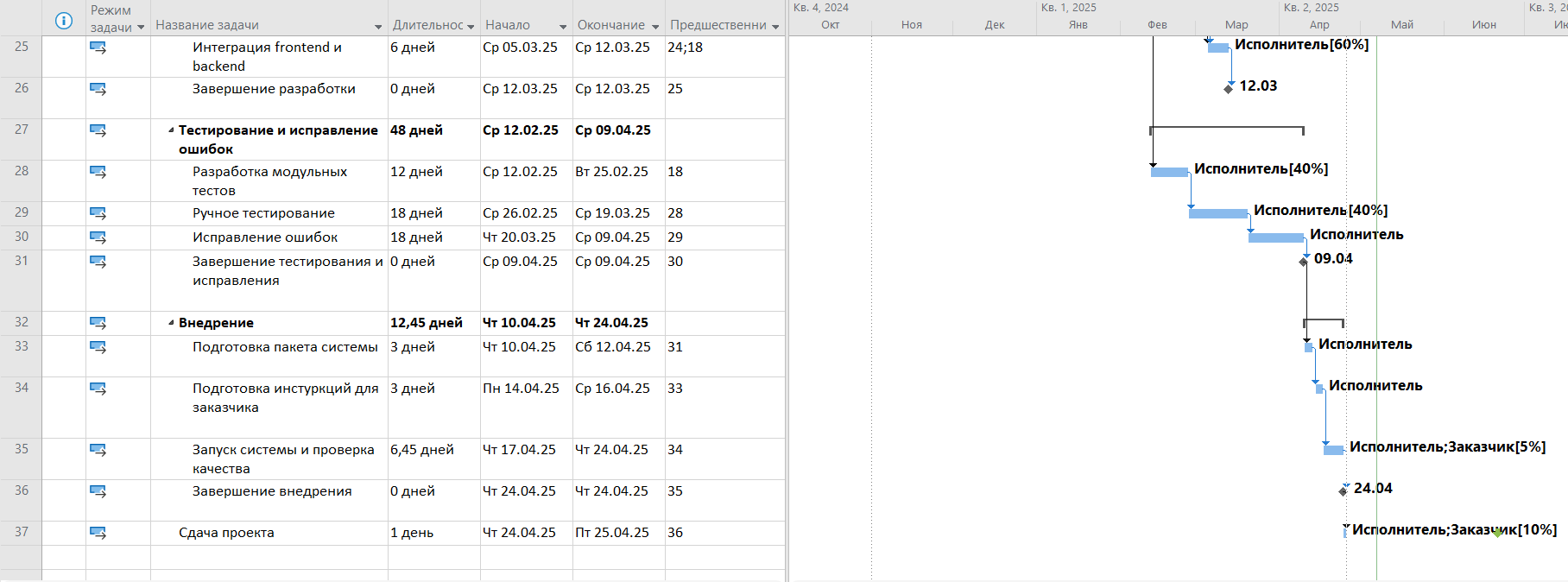


Рисунок Б.4 – Содержание проекта

**Приложение В**

**Дерево функций**

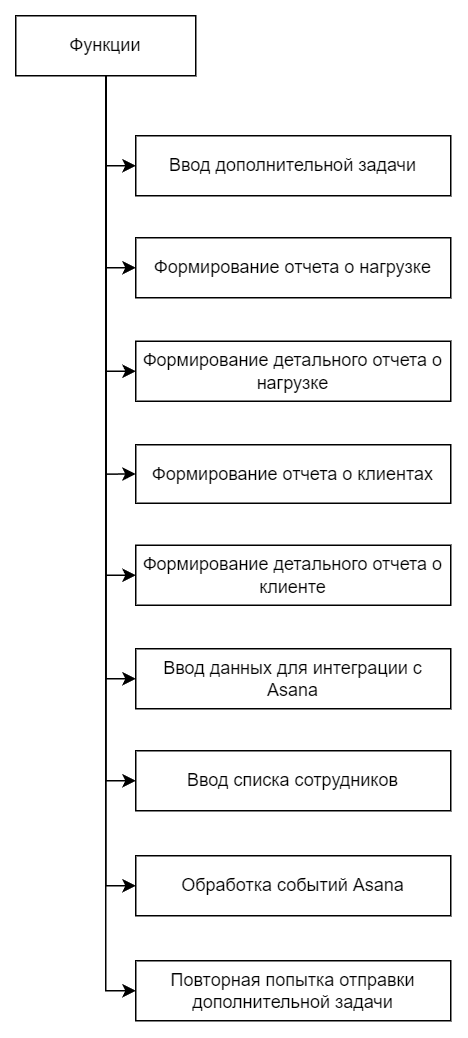


Рисунок В.1 – Дерево функций

**Приложение Г**

**Интерфейс**

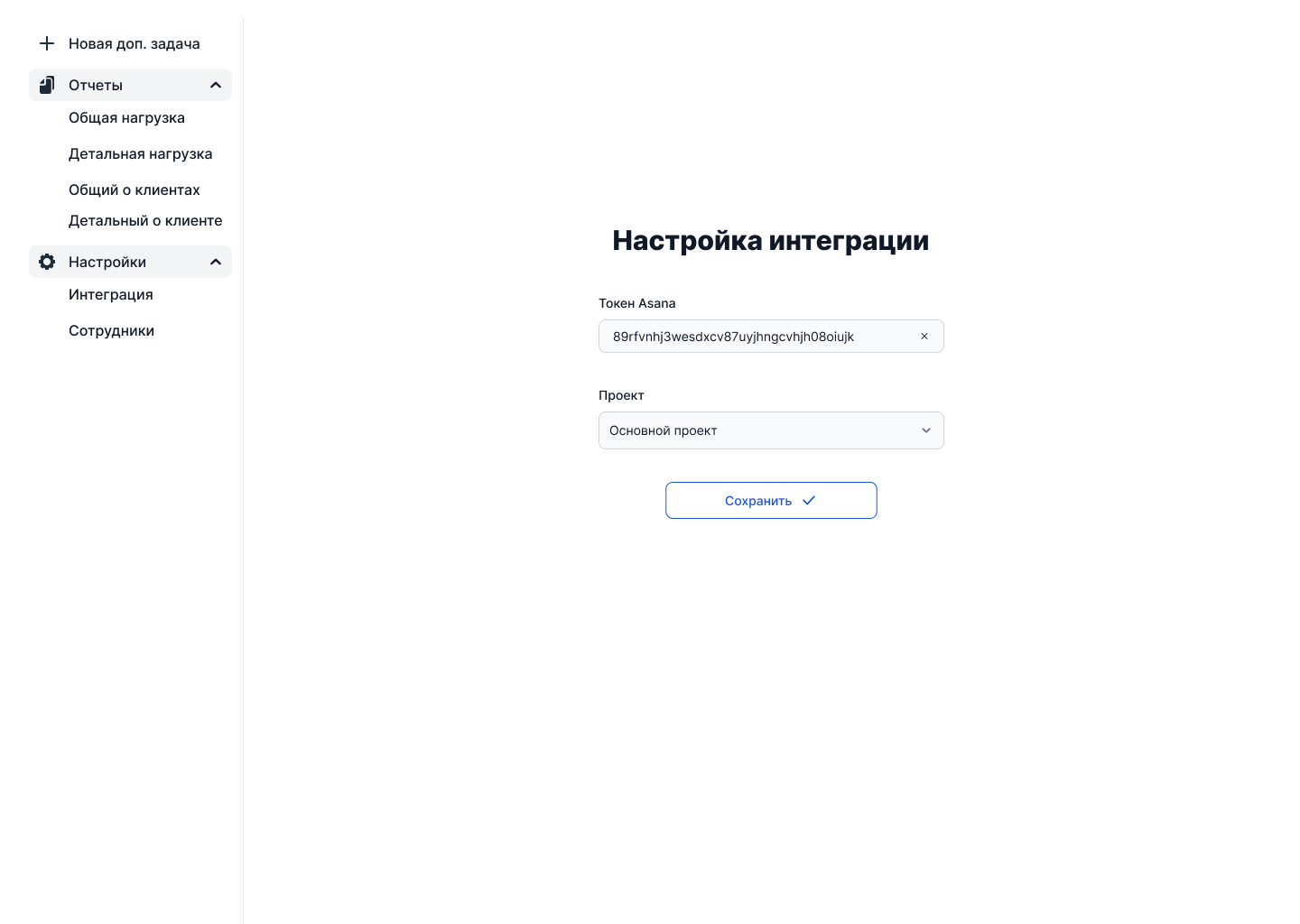


Рисунок Г.1 – Форма ввода данных для интеграции с сервисом Asana

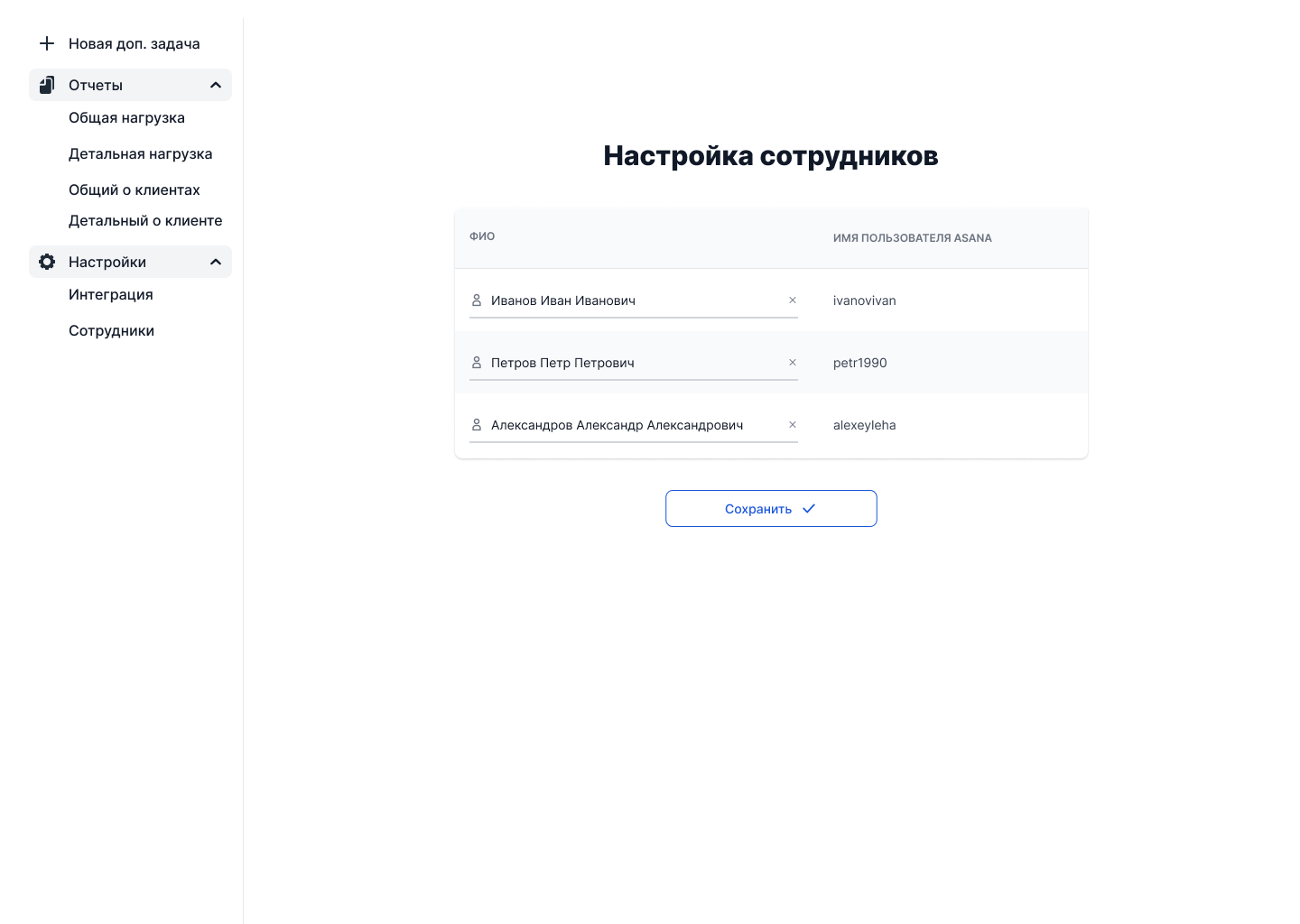


Рисунок Г.2 – Форма ввода ФИО сотрудников

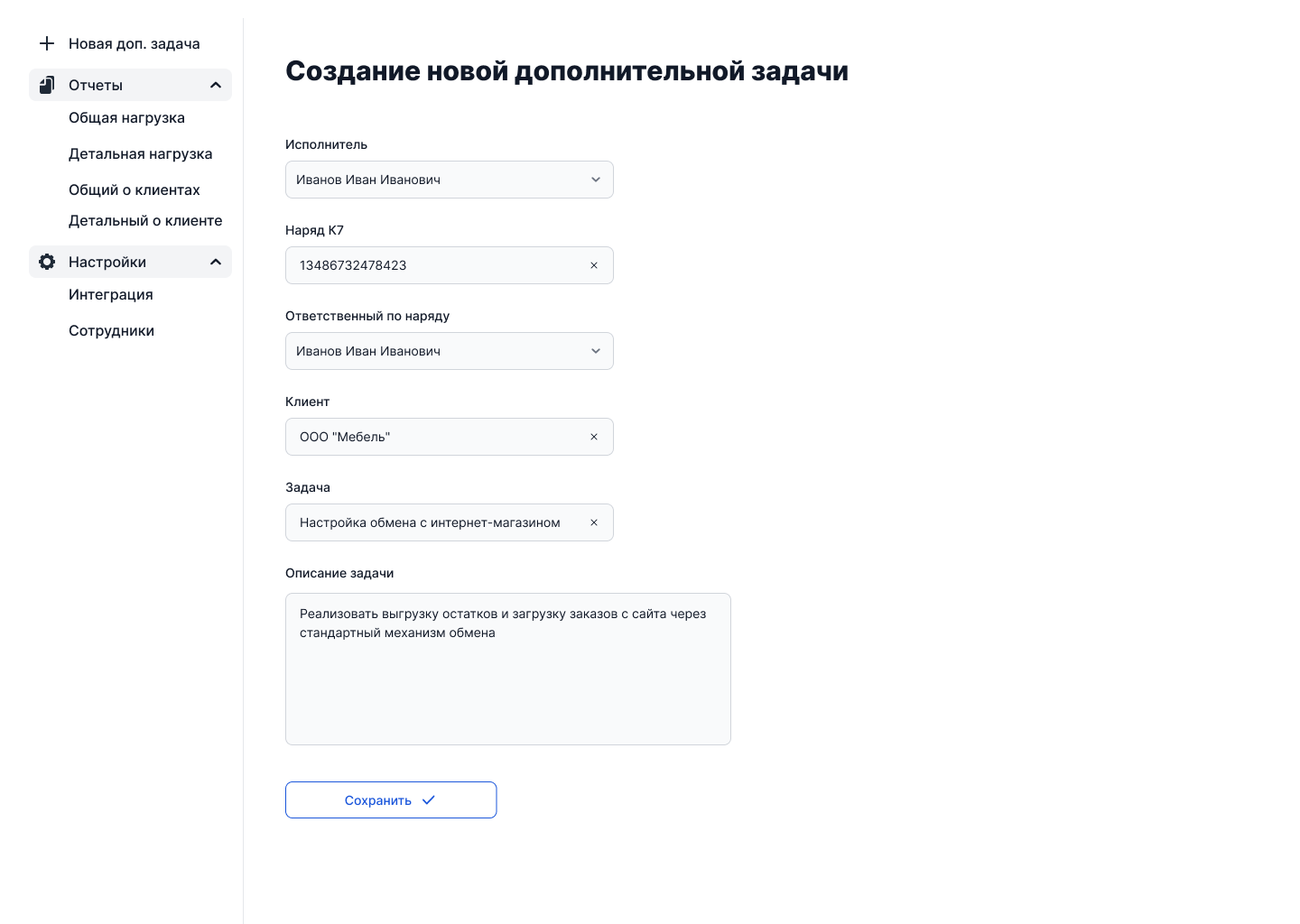


Рисунок Г.3 – Форма ввода дополнительной задачи

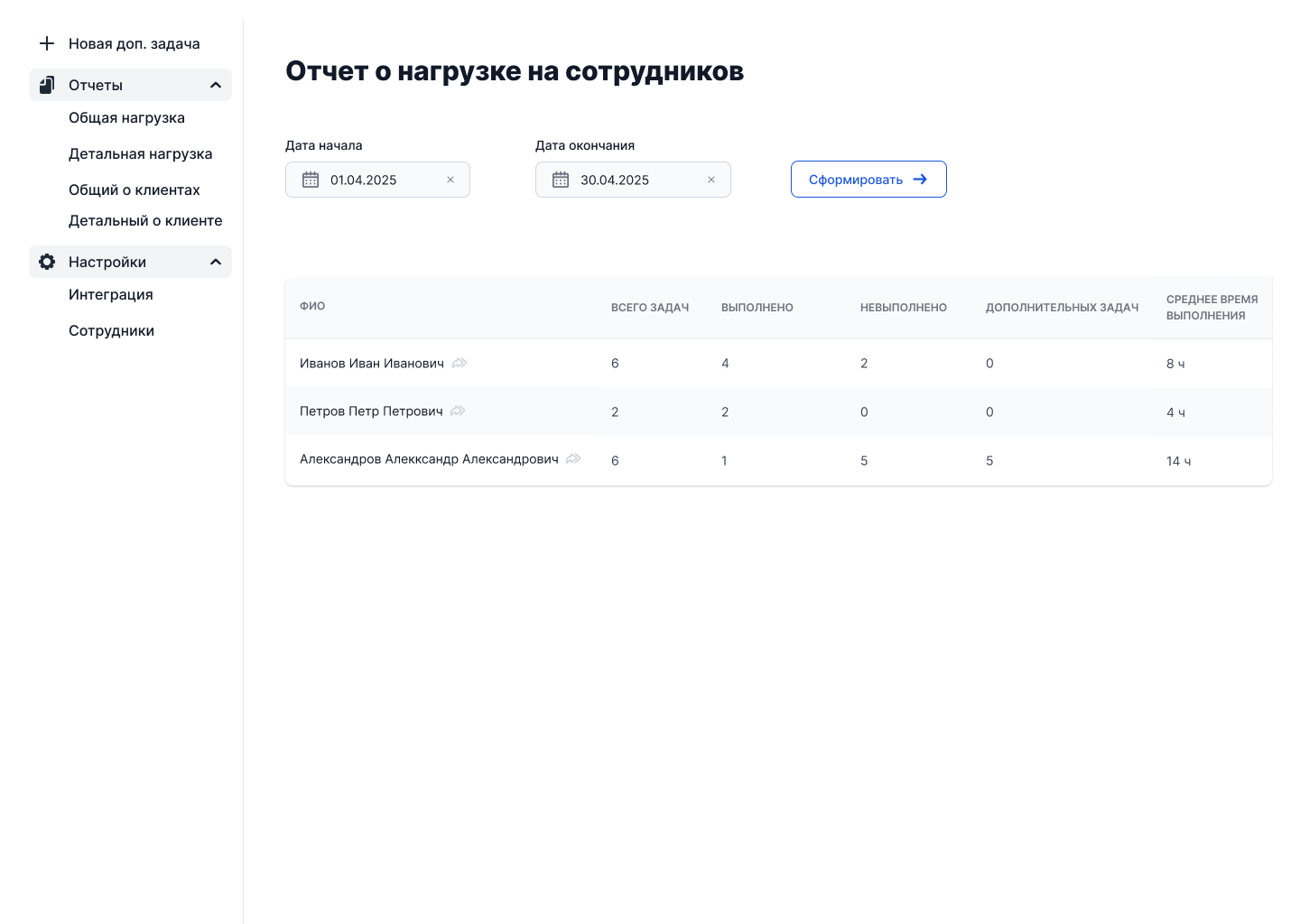


Рисунок Г.4 – Отчет о нагрузке на сотрудников

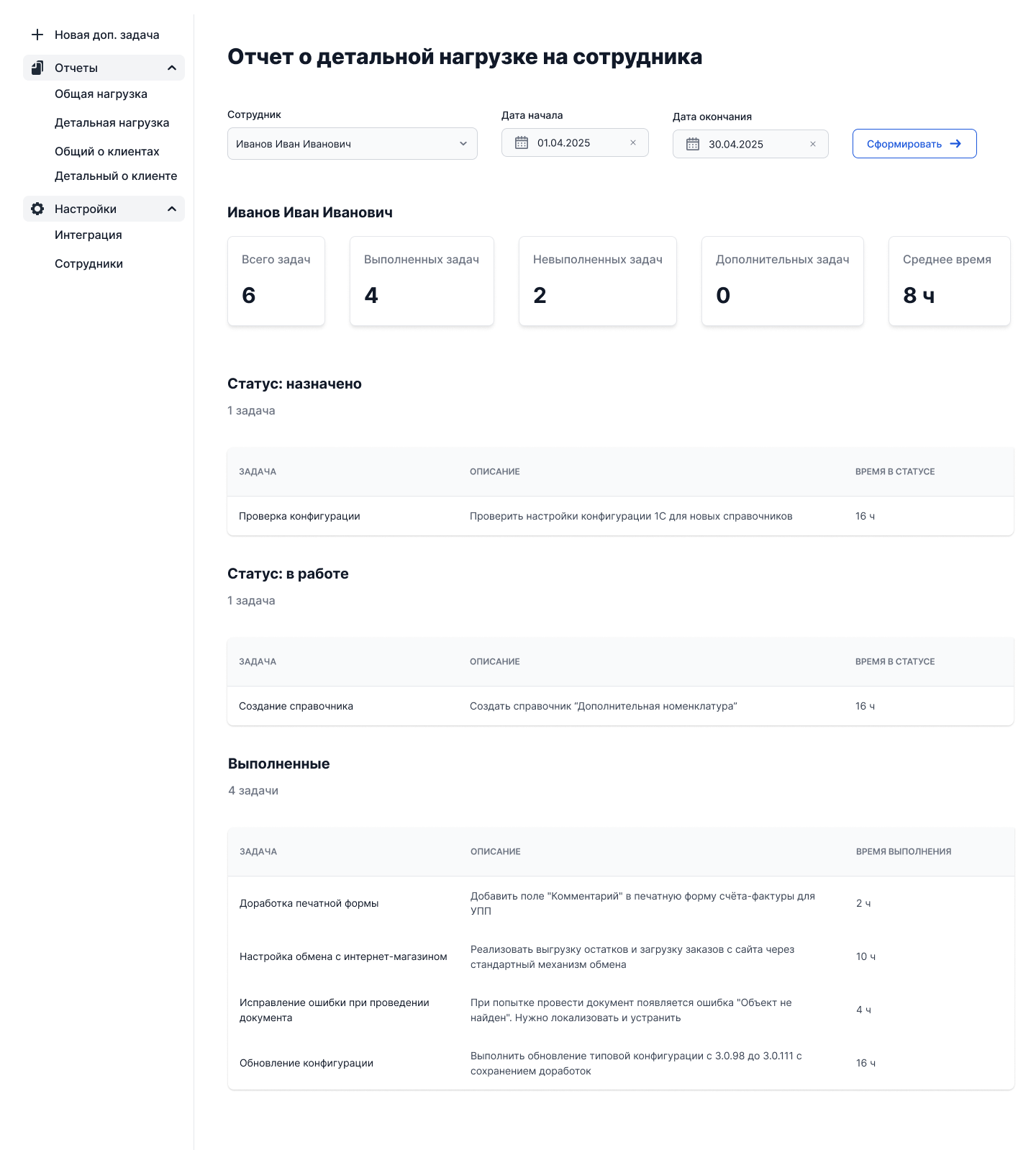


Рисунок Г.5 – Отчет о детальной нагрузке на сотрудника

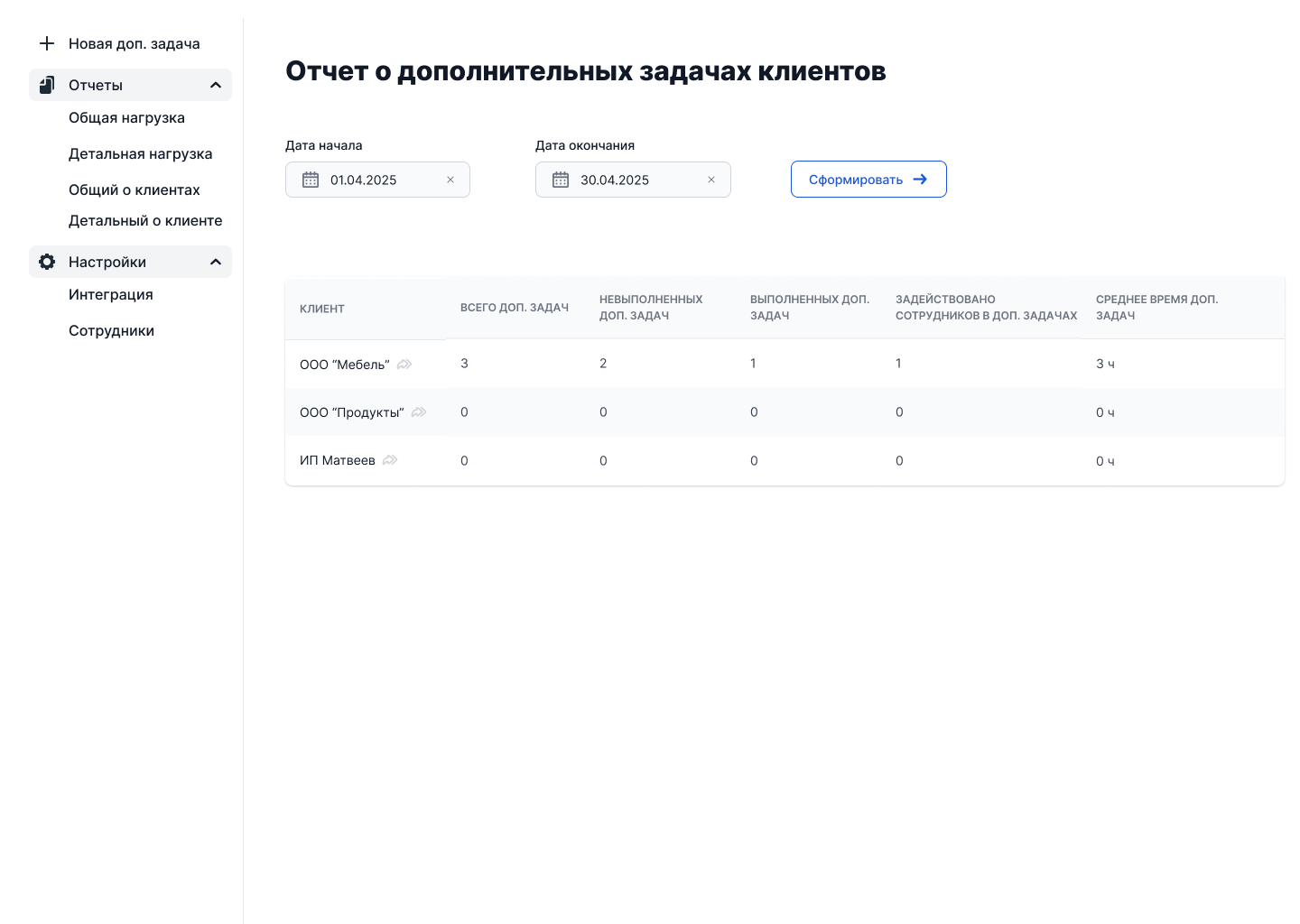


Рисунок Г.6 – Отчет о дополнительных задачах по клиентам

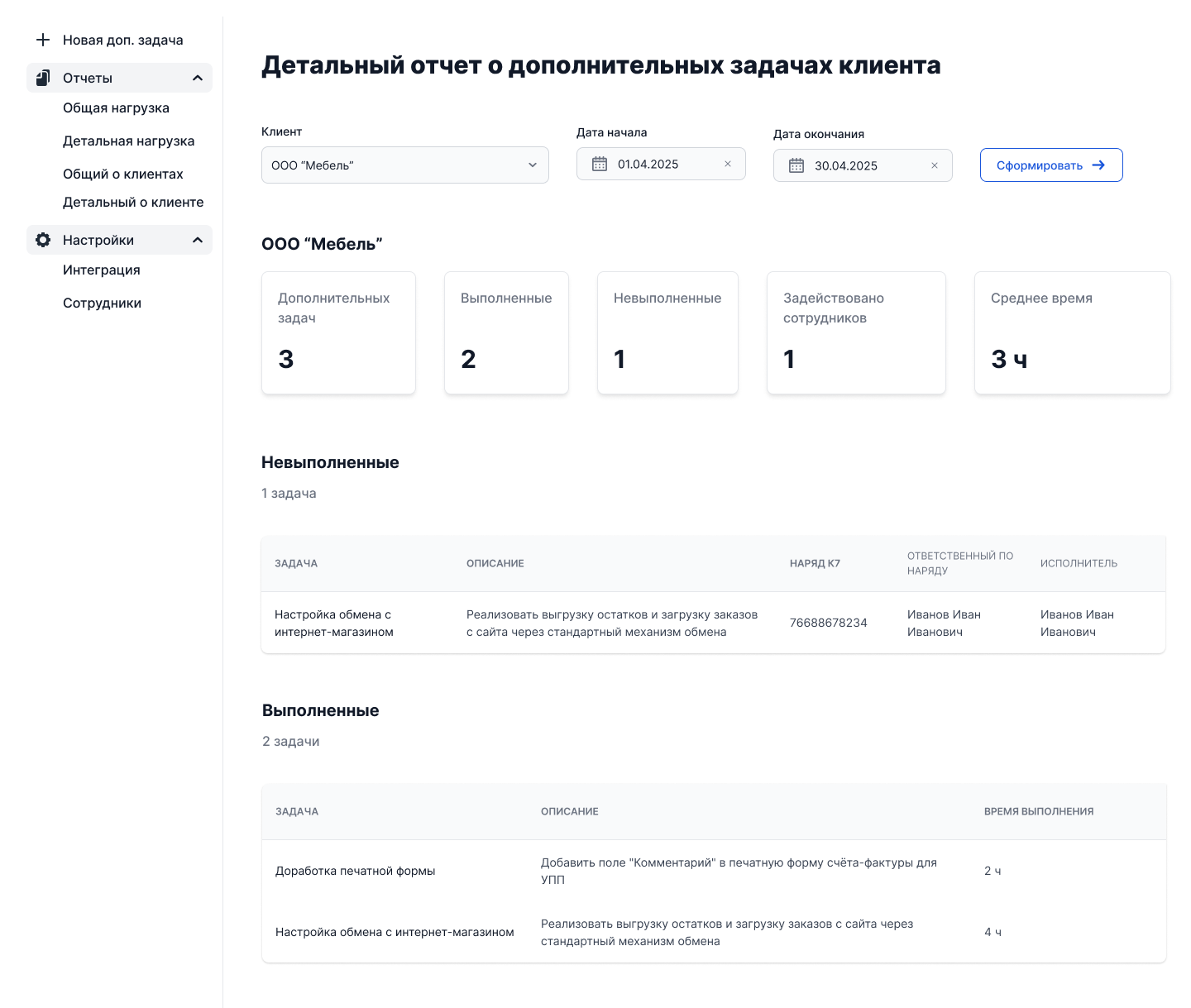


Рисунок Г.7 – Детальный отчет о дополнительной нагрузке по клиенту

**Приложение Д**

**Контрольный пример**

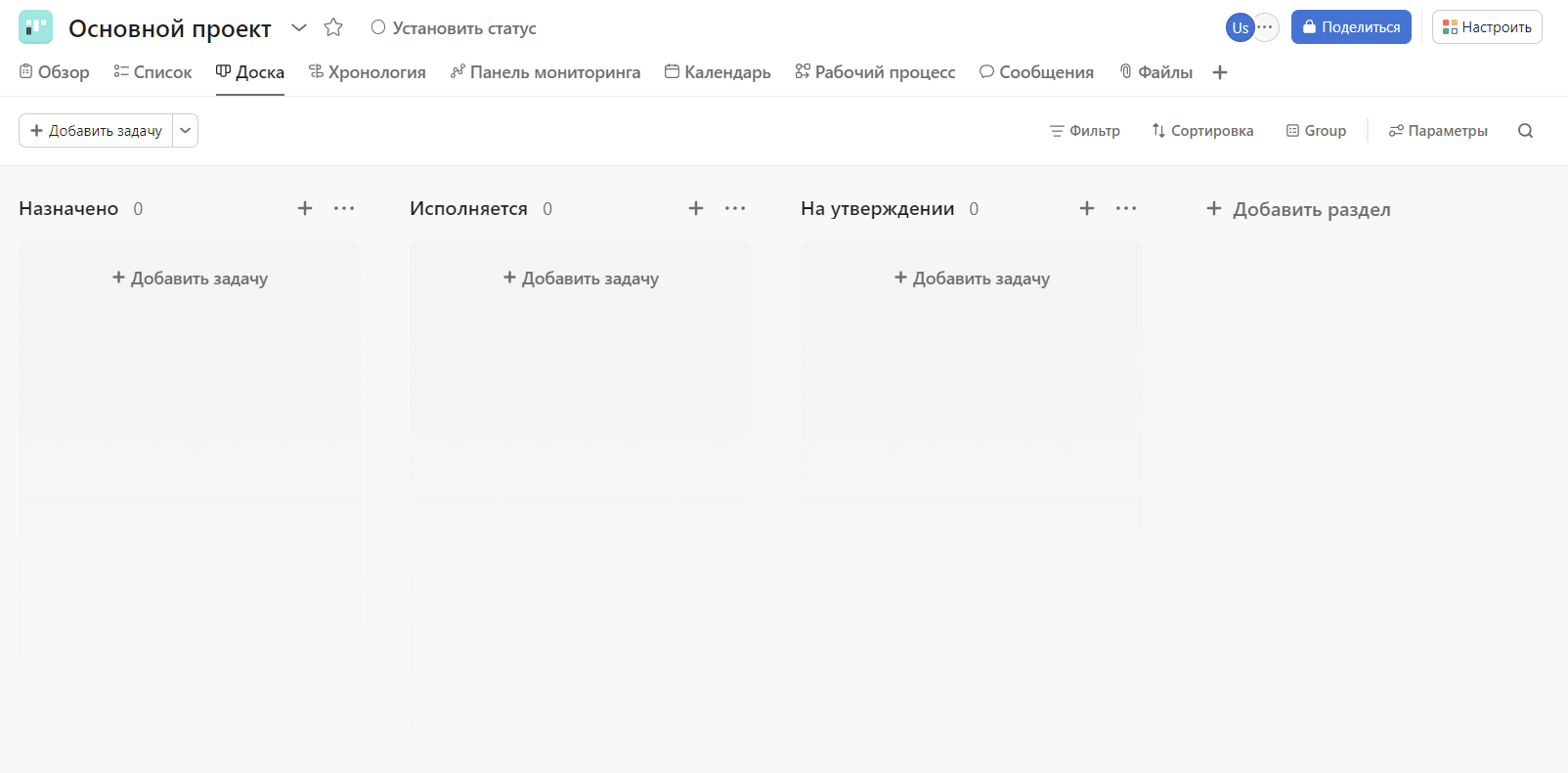


Рисунок Д.1 – Пустой проект в сервисе Asana

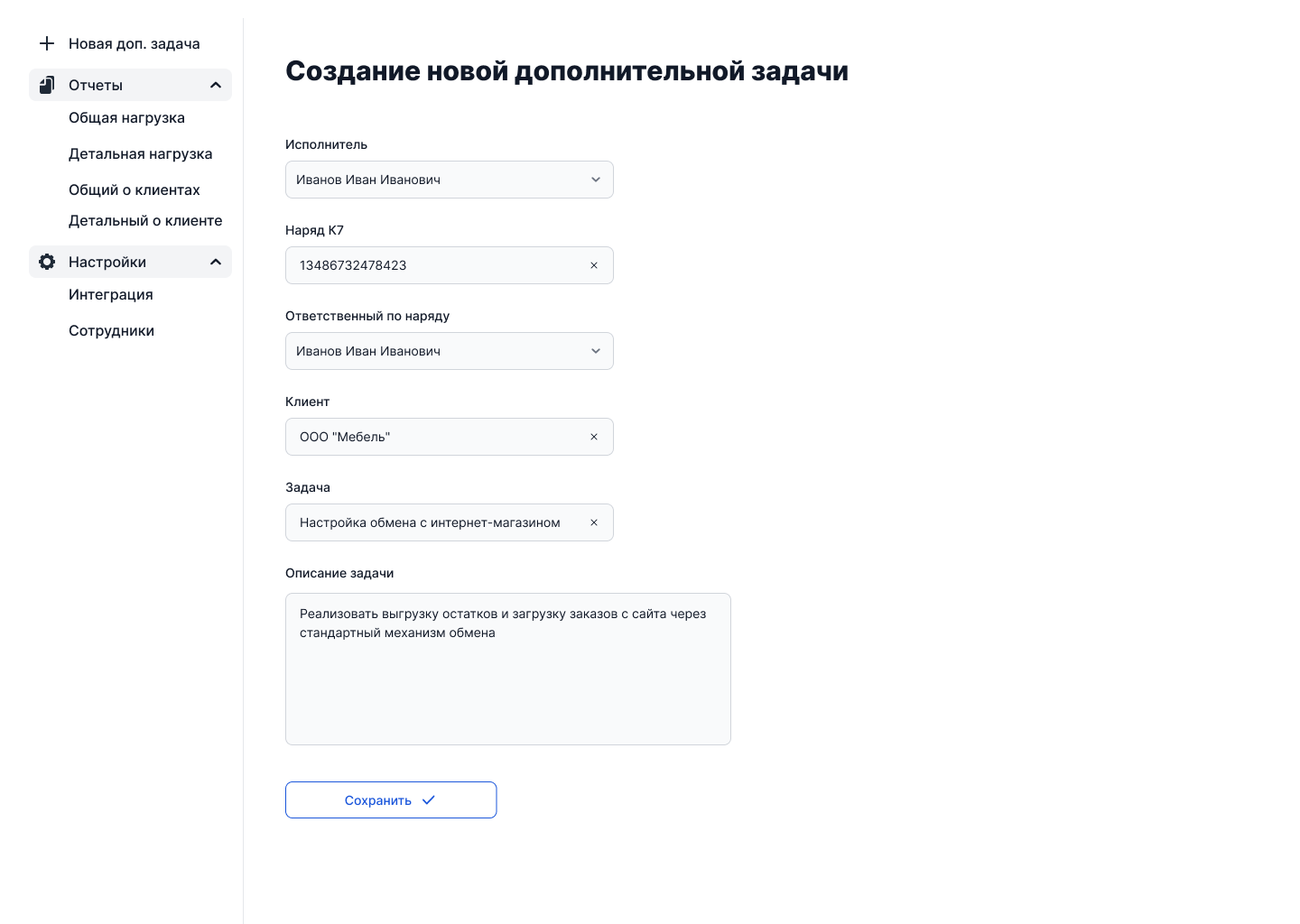


Рисунок Д.2 – Ввод дополнительной задачи

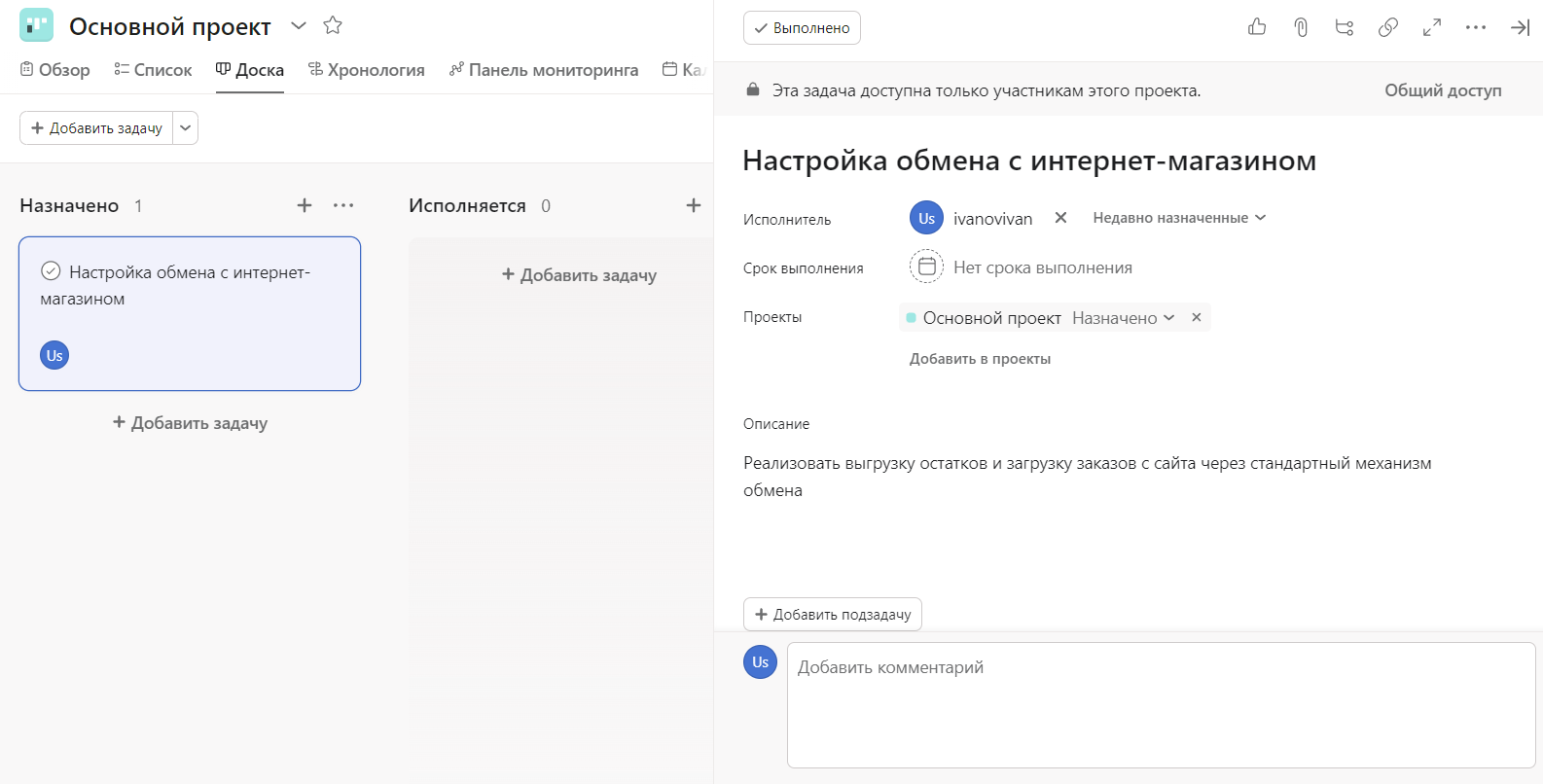


Рисунок Д.3 – Созданная задача в сервисе Asana

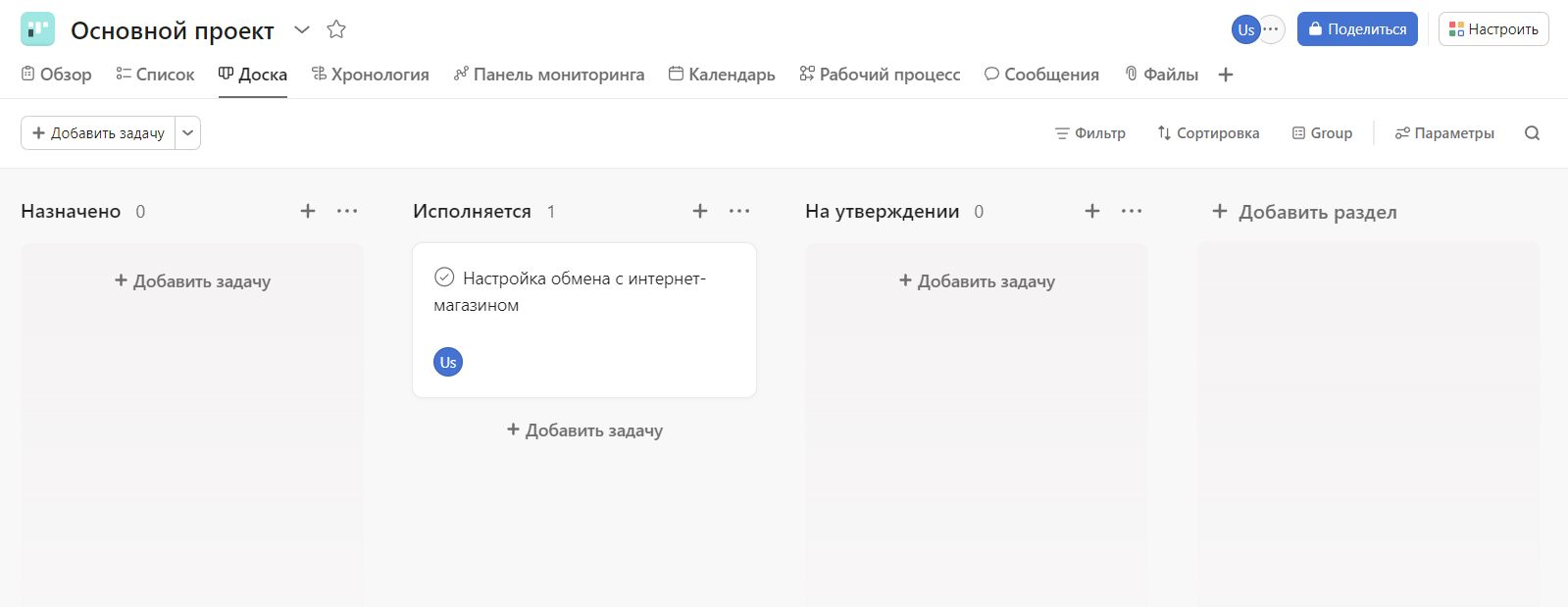


Рисунок Д.4 – Изменение статуса задачи в сервисе Asana

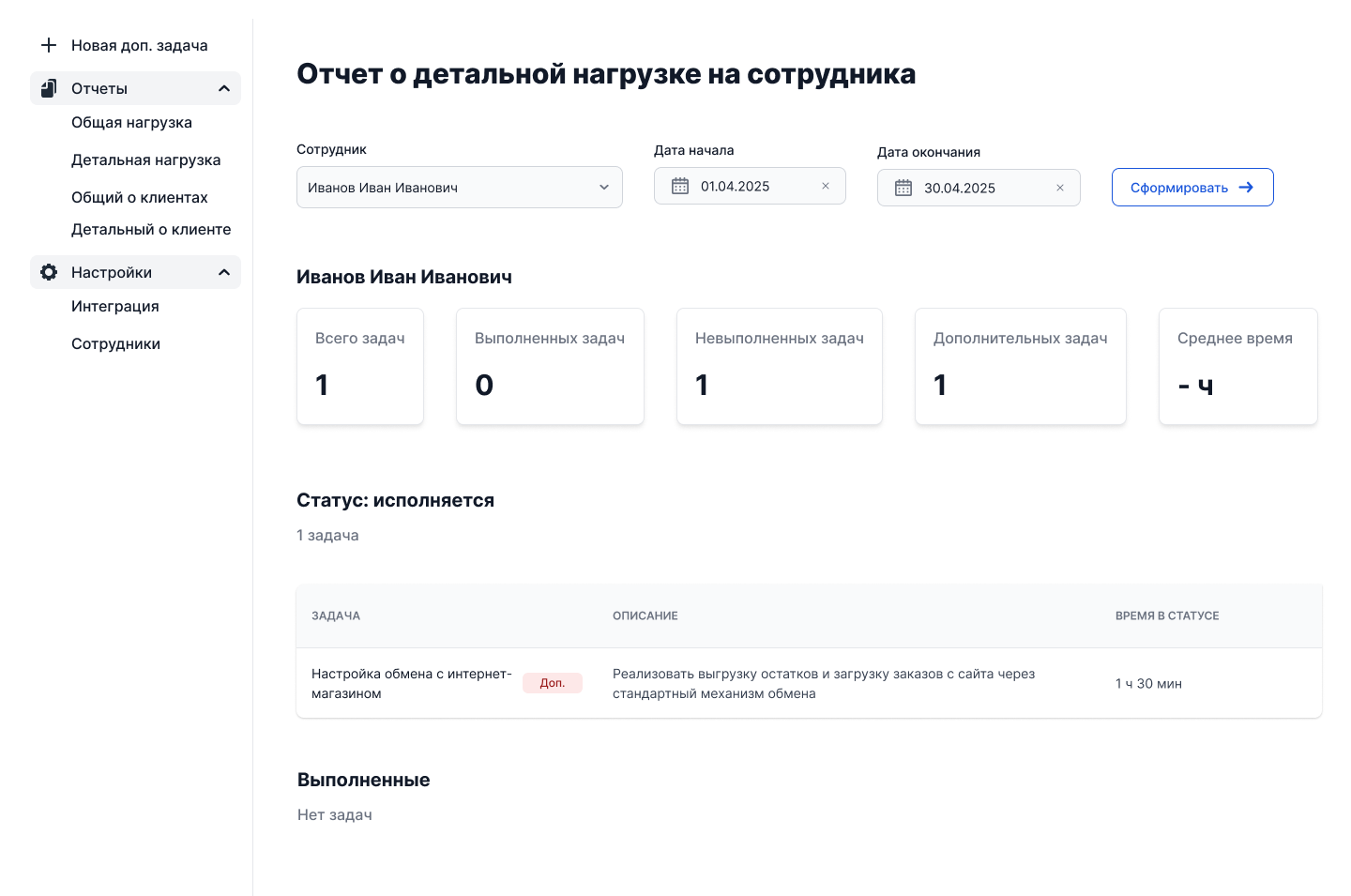


Рисунок Д.5 – Отчет о нагрузке: задача в статусе «Исполняется»

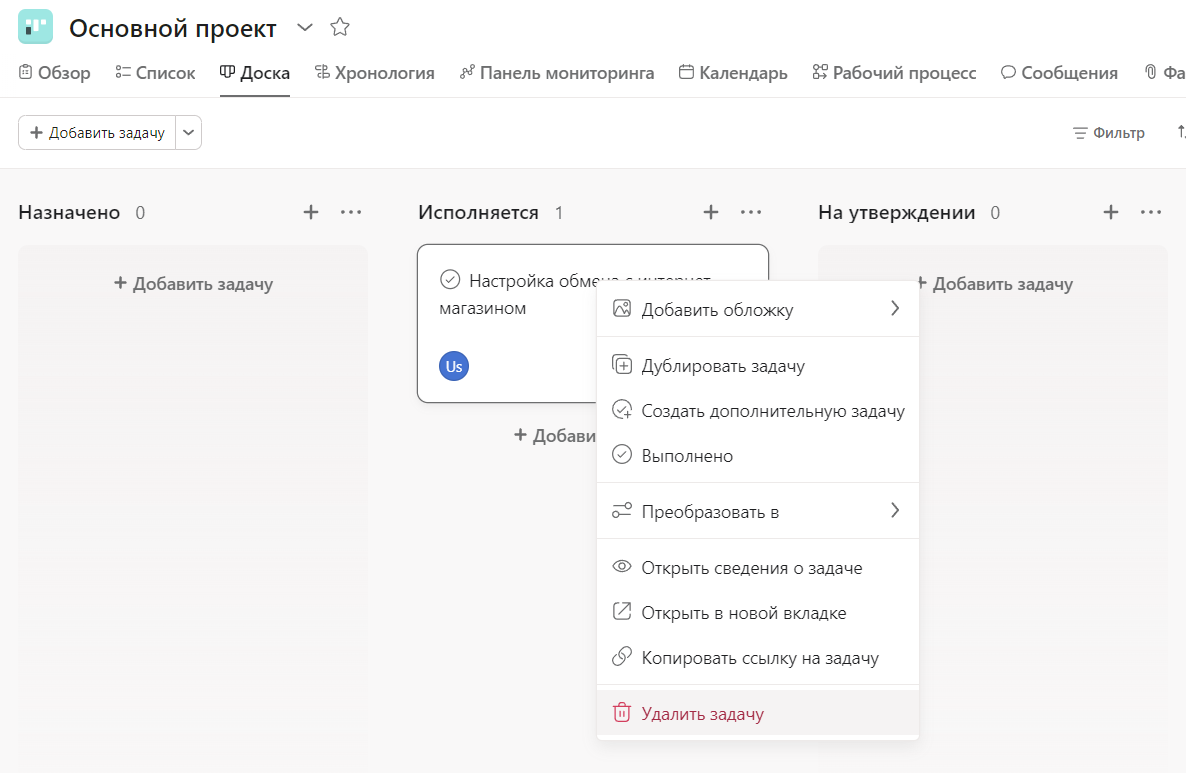


Рисунок Д.6 – Удаление задачи в сервисе Asana

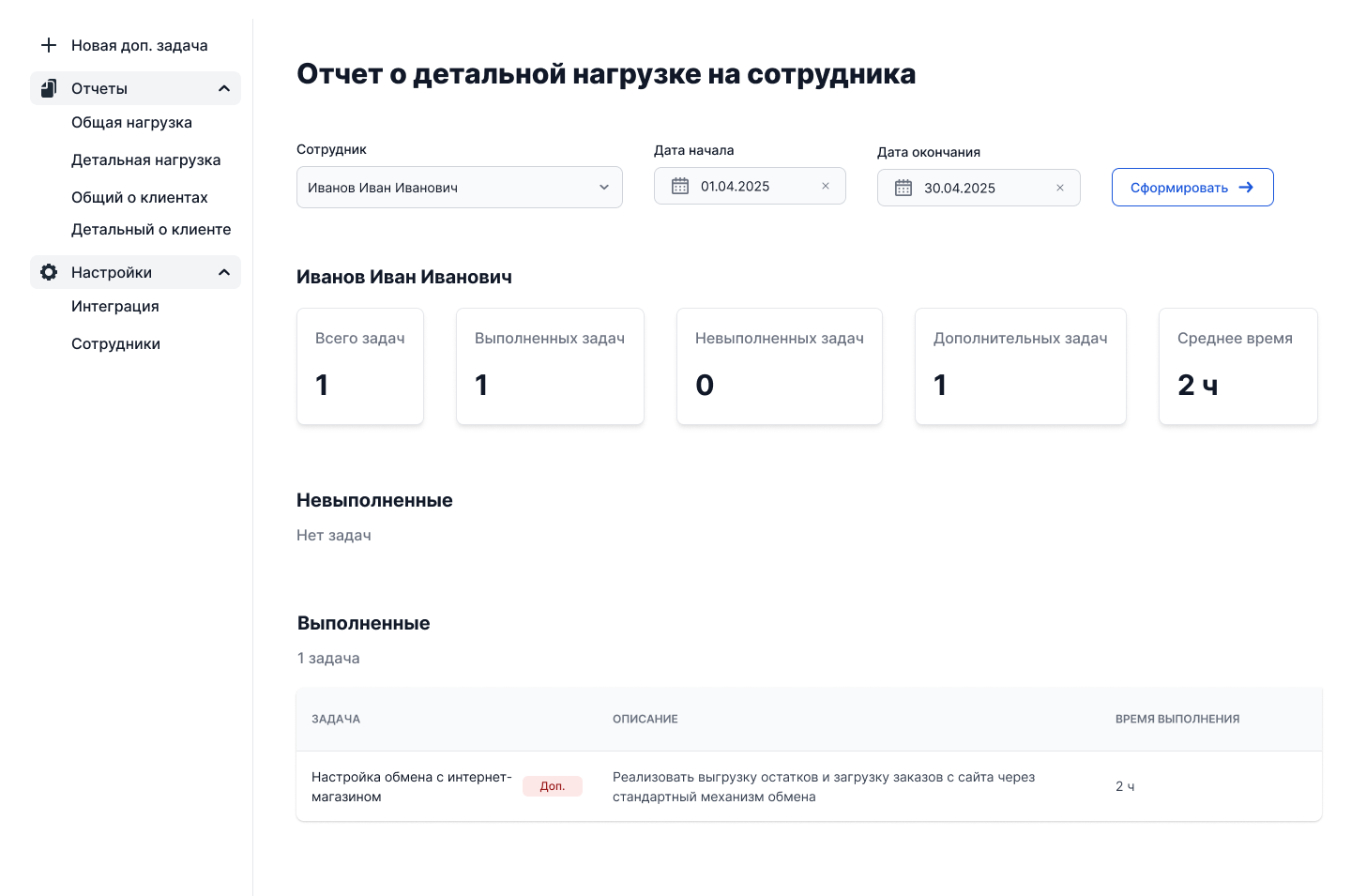


Рисунок Д.7 – Отчет о нагрузке: задача завершена