

1) Функциональные требования к ИУС

1.1. Подсистема «Учет членов организации»

- 1. Регистрация/изменение данных члена (ФЛ/ЮЛ):** ФИО/название, контакты, реквизиты, дата вступления, статус (активен/приостановлен/выбыл).
- 2. Категории членства:** ведение справочника категорий (например: “Физ. лицо”, “Юр. лицо”, “Партнер”), размер взноса, дата начала действия тарифа, хранение истории изменений тарифа.
- 3. Учет членских взносов:** начисление (план) и факт оплаты, привязка к периоду, способ оплаты.
- 4. Учет дополнительных пожертвований (не членские):** сумма, назначение, проект/мероприятие, источник, документ-основание.
- 5. Поиск/фильтрация/сортировка:** по категории, статусу, периоду оплат, задолженности, типу лица, региону и т. д.

1.2. Подсистема «Учет контрагентов»

Контрагенты: поставщики, жертвователи (не члены), получатели помощи.

- 1. Карточка контрагента** + история изменений (кто/когда/что поменял).
- 2. Взаиморасчеты и документы:**
 - ежемесячные выплаты (массовое формирование пакета документов);
 - платежные поручения (оплата с р/с);
 - расходный кассовый ордер (выплаты из кассы);
 - приходный кассовый ордер (поступление в кассу);
 - акт приемо-передачи/списания товара (помощь в натуральной форме).
- 3. Журнал расчетов:** начислено/оплачено, задолженность, назначения платежа.

4. Поиск/сортировка: по типу, периоду, статусу, суммам, договору/проекту.

1.3. Подсистема «Учет и планирование доходов»

- 1. Ввод планируемых доходов** по периодам/проектам/источникам.
- 2. Автопланирование членских взносов** (по категориям и списку активных членов).
- 3. Журнал приходных документов:** ПКО, взносы, платежи контрагентов, “не прошедшие платежи”.
- 4. Контроль поступлений на р/с:**
 - ручной ввод выписок;
 - импорт по протоколу “клиент-банк” (файл/интеграция);
 - разбор выписки + уведомления о поступлениях/несопоставленных платежах.

1.4. Подсистема «Учет и планирование расходов»

- 1. Ввод планируемых расходов** по периодам/направлениям (помощь, зарплата, мероприятия, закупки).
- 2. Автопланирование** регулярных выплат (мат. помощь, зарплата/премии).
- 3. Журнал расходных документов:** платежки, РКО, ведомости з/п, премии.
- 4. Контроль произведенных выплат:**
 - ручной ввод/импорт банковских выписок;
 - анализ неудачных платежей + уведомления.

1.5. Подсистема «Отчетность и аналитика»

Минимально по заданию (и логично для темы):

- отчет **о составе членов** за период (категории/статусы);
- отчет **о суммарных взносах каждого члена** за период;
- **количество контрагентов** за период в разрезе типов;
Дополнительно полезно:
- задолженность по членским взносам;
- поступления/расходы по проектам;
- доля “несопоставленных” платежей из выписки.

2) КПИ (ключевые показатели эффективности)

1. Доля членов с **просроченными взносами**, %
2. Среднее время подготовки **пакета документов** для выплат, час/месяц
3. Доля платежей из выписки, **автоматически сопоставленных** с планом/контрагентом, %
4. Время формирования стандартных отчетов (минуты)
5. Количество ошибок/исправлений в документах и реквизитах, ед./месяц

3. Нефункциональные требования к системе

3.1. Требования к пользовательскому интерфейсу и удобству работы

1. Пользовательский интерфейс системы должен быть **русскоязычным**, с использованием общепринятой терминологии бухгалтерского и управлеченческого учета.
2. Интерфейс должен быть выполнен в **едином визуальном стиле**, обеспечивающем логичную навигацию, единообразие форм ввода данных, таблиц и отчетов.
3. Система должна поддерживать **ролевую модель доступа**, предусматривающую следующие типы пользователей:
 - бухгалтер;
 - оператор (специалист по работе с членами и контрагентами);
 - менеджер проектов;

- руководитель организации;
 - администратор системы.
4. Для каждого типа пользователя должен предоставляться доступ **только к тем функциям и данным**, которые необходимы для выполнения его должностных обязанностей.
 5. Интерфейс должен обеспечивать:
 - удобные формы ввода и редактирования данных;
 - функции поиска, фильтрации и сортировки;
 - предотвращение ошибок ввода (проверка форматов, обязательных полей).

3.2. Требования к производительности и времени отклика

1. Время отклика основных пользовательских операций (открытие справочников, форм ввода, журналов документов, отчетов) **не должно превышать 3 секунд** при штатной нагрузке.
2. Штатной нагрузкой считается:
 - одновременная работа до 10 пользователей;
 - объем базы данных до 500 000 записей по финансовым операциям.
3. Формирование стандартных отчетов за месяц и квартал должно выполняться **не более чем за 5 секунд**.
4. Система должна обеспечивать стабильную работу при увеличении объема данных без существенного снижения производительности.

3.3. Требования к надежности и отказоустойчивости

1. Система должна обеспечивать **сохранность данных** при сбоях программного или аппаратного характера.
2. Должно быть реализовано **ежедневное автоматическое резервное копирование базы данных** с хранением резервных копий не менее 30 календарных дней.
3. Должна быть предусмотрена возможность:

- восстановления данных из резервной копии;
 - минимизации потерь данных при аварийных ситуациях.
4. Обновления системы не должны приводить к потере или искажению данных.

3.4. Требования к безопасности и защите информации

1. Доступ пользователей к системе должен осуществляться **только после прохождения процедуры аутентификации** (логин и пароль).
2. Система должна обеспечивать **разграничение прав доступа** в соответствии с ролью пользователя.
3. Должен вестись **журнал действий пользователей**, фиксирующий:
 - дату и время действия;
 - пользователя;
 - тип выполненной операции (создание, изменение, удаление, просмотр).
4. Должен быть реализован **аудит изменений данных**, позволяющий определить:
 - какие данные были изменены;
 - кем и когда произведены изменения;
 - предыдущее и новое значение.
5. Хранение и передача данных должны осуществляться с применением современных средств защиты информации (защищённые соединения, хеширование паролей).

3.5. Требования к интеграции и обмену данными

1. Система должна обеспечивать **импорт банковских выписок**:
 - из файлов стандартных форматов, предоставляемых банками;
 - посредством интеграции с системами «клиент-банк» (при наличии технической возможности).
2. При импорте банковских выписок должна выполняться:

- автоматическая загрузка данных;
 - анализ и сопоставление платежей с членами, контрагентами и документами системы;
 - уведомление пользователя о несопоставленных операциях.
3. Система должна поддерживать **экспорт отчетности** в следующие форматы:
- электронные таблицы (XLSX);
 - документы для печати (PDF).
4. Экспортируемые отчеты должны сохранять структуру, форматирование и корректность данных.

3.6. Требования к сопровождаемости и масштабируемости

1. Архитектура системы должна обеспечивать возможность:
 - добавления новых отчетов;
 - расширения перечня учитываемых документов;
 - подключения дополнительных интеграций без полной переработки системы.
2. Система должна быть реализована с использованием широко распространённых и поддерживаемых программных средств, что упрощает сопровождение и развитие.
3. Документация по системе должна включать:
 - описание архитектуры;
 - инструкцию пользователя;
 - описание процедур резервного копирования и восстановления.

4. Информационная архитектура системы

4.1. Описание таблиц базы данных

4.1.1. Таблица system_user

Назначение:

Таблица предназначена для хранения информации о пользователях системы и их ролях.

Основные поля:

- `user_id` — уникальный идентификатор пользователя;
- `username` — имя пользователя;
- `role` — роль пользователя в системе.

4.1.2. Таблица `member_category`

Назначение:

Таблица предназначена для хранения справочной информации о категориях членов благотворительной организации.

Основные поля:

- `category_id` — уникальный идентификатор категории;
- `name` — наименование категории;
- `description` — описание категории.

4.1.3. Таблица `member`

Назначение:

Таблица предназначена для хранения сведений о членах благотворительной организации — физических и юридических лицах.

Основные поля:

- `member_id` — уникальный идентификатор члена организации;
- `member_type` — тип члена (физическое или юридическое лицо);
- `full_name` — ФИО или наименование организации;
- `phone` — контактный телефон;

- email — адрес электронной почты;
- join_date — дата вступления в организацию;
- status — статус члена организации;
- category_id — ссылка на категорию членства.

4.1.4. Таблица counterparty

Назначение:

Таблица предназначена для хранения информации о контрагентах благотворительной организации.

Основные поля:

- counterparty_id — уникальный идентификатор контрагента;
- cp_type — тип контрагента;
- name — наименование или ФИО контрагента;
- inn — идентификационный номер;
- bank_account — расчетный счет;
- phone — контактный телефон;
- email — адрес электронной почты;
- status — статус контрагента.

4.1.5. Таблица party

Назначение:

Таблица предназначена для представления универсальной сущности «Участник», объединяющей членов организации и контрагентов и обеспечивающей единый механизм ссылок в финансовых и операционных таблицах.

Основные поля:

- party_id — уникальный идентификатор участника;

- party_type — тип участника (член организации или контрагент);
- member_id — ссылка на таблицу member (для участников типа «член»);
- counterparty_id — ссылка на таблицу counterparty (для участников типа «контрагент»).

4.1.6. Таблица counterparty_changelog

Назначение:

Таблица предназначена для хранения журнала изменений сведений о контрагентах.

Основные поля:

- log_id — уникальный идентификатор записи;
- counterparty_id — ссылка на контрагента;
- changed_at — дата и время изменения;
- changed_by — пользователь, внесший изменения;
- field_name — наименование измененного поля;
- old_value — предыдущее значение;
- new_value — новое значение.

4.1.7. Таблица category_fee_history

Назначение:

Таблица предназначена для хранения истории изменения размеров членских взносов по категориям членов организаций.

Основные поля:

- fee_id — уникальный идентификатор записи;
- category_id — ссылка на категорию члена;
- fee_amount — размер членского взноса;
- date_from — дата начала действия взноса;

- `date_to` — дата окончания действия взноса.

4.1.8. Таблица membership_fee_plan

Назначение:

Таблица предназначена для учета начисленных, оплаченных и просроченных членских взносов членов организации.

Основные поля:

- `plan_id` — уникальный идентификатор начисления;
- `member_id` — ссылка на члена организации;
- `period_start` — начало отчетного периода;
- `period_end` — окончание отчетного периода;
- `amount` — сумма взноса;
- `status` — статус взноса.

4.1.9. Таблица finance_document

Назначение:

Таблица предназначена для хранения финансовых документов, отражающих доходные и расходные операции благотворительной организации.

Основные поля:

- `document_id` — уникальный идентификатор документа;
- `document_type` — тип финансового документа;
- `document_date` — дата документа;
- `amount` — сумма операции;
- `direction` — направление операции (доход или расход);
- `party_id` — ссылка на универсальную сущность «Участник»;
- `project_id` — ссылка на проект или мероприятие;
- `status` — статус документа;

- comment — примечание.

4.1.10. Таблица donation

Назначение:

Таблица предназначена для учета добровольных пожертвований членов организации и контрагентов.

Основные поля:

- donation_id — уникальный идентификатор пожертвования;
- party_id — ссылка на универсальную сущность «Участник»;
- donation_date — дата пожертвования;
- amount — сумма пожертвования;
- purpose — назначение пожертвования

4.1.11. Таблица bank_statement

Назначение:

Таблица предназначена для хранения информации о банковских выписках по расчетному счету организации.

Основные поля:

- statement_id — уникальный идентификатор выписки;
- period_start — начало периода выписки;
- period_end — окончание периода выписки;
- import_date — дата загрузки выписки.

4.1.12. Таблица bank_transaction

Назначение:

Таблица предназначена для хранения отдельных банковских операций, входящих в состав банковской выписки.

Основные поля:

- transaction_id — уникальный идентификатор операции;
- statement_id — ссылка на банковскую выписку;
- transaction_date — дата операции;
- amount — сумма операции;
- description — назначение платежа;
- party_id — ссылка на универсальную сущность «Участник»;
- match_status — статус сопоставления операции.

4.1.13. Таблица project

Назначение:

Таблица предназначена для хранения информации о проектах и мероприятиях благотворительной организации.

Основные поля:

- project_id — уникальный идентификатор проекта;
- name — наименование проекта;
- start_date — дата начала проекта;
- end_date — дата окончания проекта;
- budget_plan — плановый бюджет проекта.

4.2. Аналитическая модель для анализа и отчетности

Для формирования аналитических отчетов и оценки эффективности деятельности благотворительной организации в системе предусмотрена **аналитическая модель данных**, реализованная в виде **звездообразной схемы (star schema)**.

Центральным элементом аналитической модели является таблица фактов, содержащая агрегируемые показатели деятельности организации, а также набор таблиц измерений, обеспечивающих многомерный анализ данных.

Таблица фактов: FACT_FINANCE

Назначение:

Таблица предназначена для хранения фактических показателей финансово-хозяйственной деятельности благотворительной организации за определенные периоды времени.

Таблицы измерений

Измерение времени: DIM_TIME

Назначение:

Позволяет анализировать данные в разрезе дат, месяцев, кварталов и лет.

Поля:

- time_id — первичный ключ;
- date — календарная дата;
- day — день месяца;
- month — месяц;
- quarter — квартал;
- year — год.

Измерение участников: DIM_PARTY

Назначение:

Используется для анализа финансовых операций по членам организации и контрагентам.

Поля:

- party_id — первичный ключ;
- party_type — тип участника (член организации / контрагент).

Измерение категорий членов: DIM_MEMBER_CATEGORY

Назначение:

Используется для анализа взносов и пожертвований в разрезе категорий членов организации.

Поля:

- member_category_id — первичный ключ;
- category_name — наименование категории.

Измерение проектов: DIM_PROJECT

Назначение:

Позволяет анализировать доходы и расходы по проектам и мероприятиям благотворительной организации.

Поля:

- project_id — первичный ключ;
- project_name — наименование проекта.

Измерение типов документов: DIM_DOCUMENT_TYPE

Назначение:

Используется для анализа операций по типам финансовых документов.

Поля:

- document_type_id — первичный ключ;
- document_type_name — наименование типа документа.

Измерение типов операций: DIM_OPERATION_TYPE

Назначение:

Позволяет разделять доходные и расходные операции.

Поля:

- operation_type_id — первичный ключ;
- operation_type_name — тип операции (доход / расход).

Измерение статусов платежей: DIM_PAYMENT_STATUS

Назначение:

Используется для анализа выполненных, просроченных и неудачных платежей.

Поля:

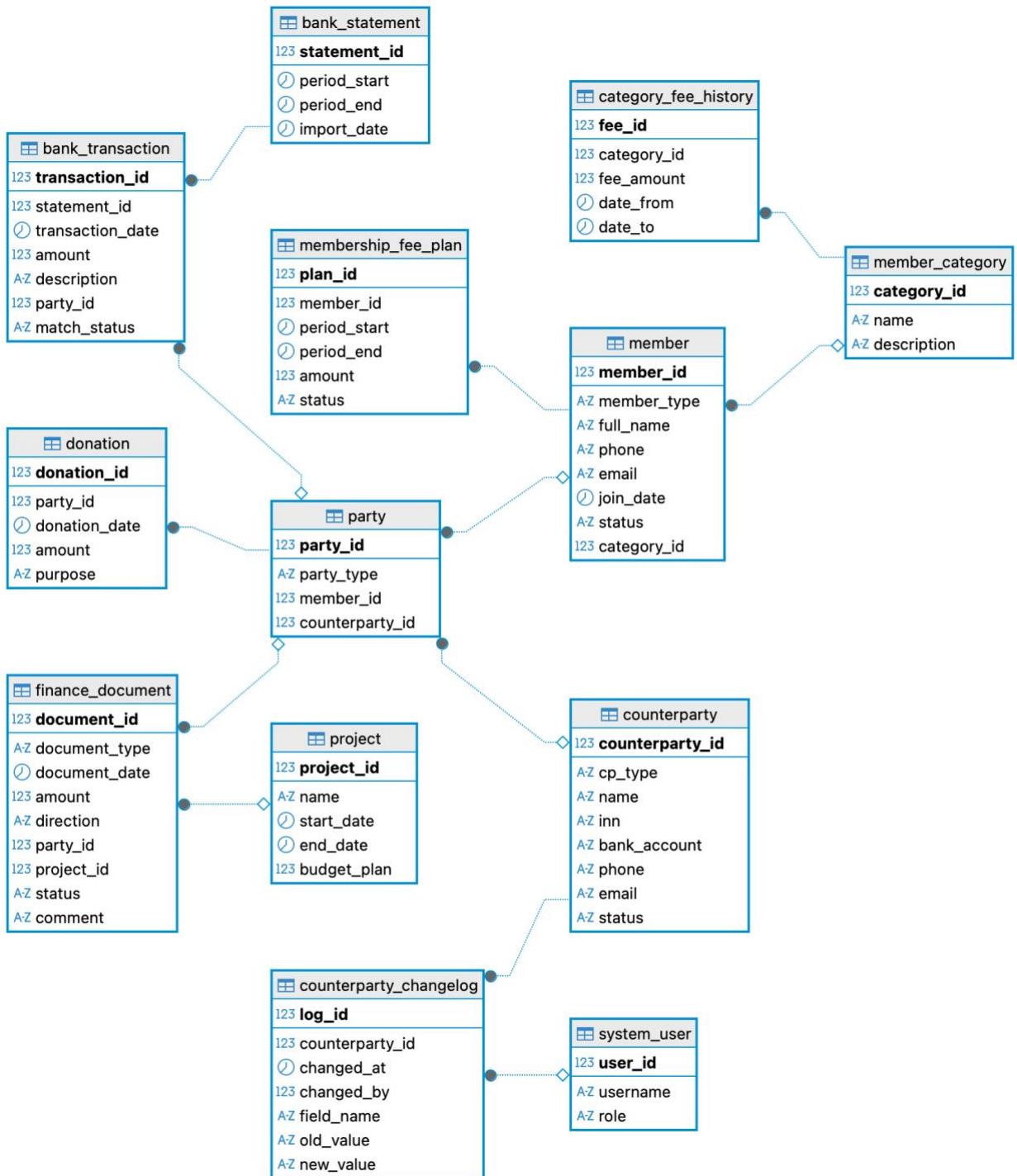
- payment_status_id — первичный ключ;
- payment_status_name — наименование статуса.

Показатели эффективности

Для контроля эффективности деятельности в таблице фактов используются следующие аналитические признаки:

- deviation — отклонение фактических показателей от плановых значений;
- is_overdue — признак просрочки поступлений или выплат;
- amount — фактический объем поступлений или расходов;
- planned_amount — запланированный объем средств.

4.3 ERP диаграмма архитектуры



5. Функциональная и программно-техническая архитектура системы

Функциональная и программно-техническая архитектура определяет состав подсистем проектируемой системы автоматизации деятельности

благотворительной организации, а также способы их реализации и взаимодействия между собой.

Архитектура системы построена по модульному принципу, что обеспечивает расширяемость, удобство сопровождения и возможность поэтапного развития функциональности.

5.1. Функциональная архитектура системы

Функциональная архитектура представляет собой совокупность взаимосвязанных подсистем, реализующих основные бизнес-процессы благотворительной организации.

5.1.1. Подсистема учета членов организации

Подсистема предназначена для:

- ведения информации о членах организации (физических и юридических лицах);
- управления категориями членства и размерами членских взносов;
- учета начисленных, оплаченных и просроченных членских взносов;
- учета дополнительных пожертвований членов организации;
- поиска, фильтрации и сортировки информации о членах.

5.1.2. Подсистема учета контрагентов

Подсистема обеспечивает:

- ведение карточек контрагентов (поставщиков, жертвователей, получателей помощи);
- учет взаиморасчетов с контрагентами;
- формирование финансовых документов для расчетов;
- ведение журнала изменений данных о контрагентах;
- поиск и анализ информации о контрагентах.

5.1.3. Подсистема учета доходов организации

Подсистема предназначена для:

- учета членских взносов и добровольных пожертвований;
- ведения журнала приходных финансовых документов;
- планирования доходов по периодам;
- контроля фактических поступлений денежных средств;
- анализа поступлений на расчетный счет организации.

5.1.4. Подсистема учета расходов организации

Подсистема обеспечивает:

- учет планируемых и фактических расходов;
- формирование расходных финансовых документов;
- учет выплат получателям благотворительной помощи;
- контроль произведенных платежей;
- выявление неудачных и просроченных выплат.

5.1.5. Подсистема работы с банковскими выписками

Подсистема предназначена для:

- загрузки банковских выписок;
- автоматического анализа банковских операций;
- сопоставления поступлений и выплат с членами организации и контрагентами;
- информирования пользователя о несопоставленных операциях.

5.1.6. Подсистема отчетности и аналитики

Подсистема обеспечивает:

- формирование регламентной и аналитической отчетности;

- анализ данных в разрезе периодов, участников и проектов;
- использование аналитической витрины данных;
- экспорт отчетов в форматы XLSX и PDF.

5.2. Программно-техническая архитектура системы

Программно-техническая архитектура определяет набор программных средств, используемых для реализации функциональных подсистем системы.

5.2.1. Общая архитектура

Система реализуется по клиент-серверной архитектуре и включает следующие основные компоненты:

- сервер базы данных;
- сервер приложений;
- пользовательский интерфейс.

5.2.2. Сервер базы данных

Для хранения данных используется реляционная система управления базами данных, обеспечивающая:

- хранение операционных данных;
- поддержку ссылочной целостности;
- хранение аналитической витрины данных;
- резервное копирование и восстановление данных.

5.2.3. Сервер приложений

Сервер приложений реализует:

- бизнес-логику системы;
- контроль прав доступа пользователей;
- обработку данных из базы данных;

- взаимодействие между функциональными подсистемами.

5.2.4. Пользовательский интерфейс

Пользовательский интерфейс представляет собой графический интерфейс, обеспечивающий:

- доступ к функциональности системы через веб-браузер;
- работу с формами ввода и редактирования данных;
- просмотр отчетов и аналитической информации;
- разграничение доступа в зависимости от роли пользователя.

5.3. Взаимодействие компонентов системы

Взаимодействие компонентов системы осуществляется следующим образом:

- пользователь взаимодействует с системой через пользовательский интерфейс;
- сервер приложений обрабатывает запросы пользователя и выполняет бизнес-логику;
- сервер базы данных обеспечивает хранение и предоставление данных;
- подсистема аналитики использует агрегированные данные для формирования отчетов.

5.4. Вывод по архитектуре системы

Разработанная функциональная и программно-техническая архитектура обеспечивает:

- автоматизацию основных процессов деятельности благотворительной организации;
- целостность и надежность хранения данных;
- удобство работы пользователей различных ролей;
- возможность масштабирования и дальнейшего развития системы.

6. Технико-экономическое обоснование проекта

6.1. Метод определения эффективности системы автоматизации

Эффективность внедрения системы автоматизации деятельности благотворительной организации определяется на основе **сравнения затрат на выполнение ключевых бизнес-процессов до и после внедрения системы**, а также **косвенных эффектов**, связанных с повышением прозрачности и управляемости деятельности.

В качестве основного метода оценки применяется:

- **метод расчёта годового экономического эффекта от автоматизации;**
- **метод срока окупаемости инвестиций (Payback Period).**

Годовой экономический эффект определяется как сумма:

- снижения трудозатрат персонала;
- сокращения потерь от ошибок, несвоевременных выплат и несопоставленных платежей;
- роста объёма поступлений за счёт повышения собираемости членских взносов и пожертвований.

6.2. Метрики оценки эффективности внедрения системы

Для оценки эффективности внедрения системы используются следующие метрики:

1. **Сокращение трудозатрат на ведение учёта членов, контрагентов и финансовых документов (чел-час/месяц).**
2. **Сокращение времени подготовки финансовых документов** (платёжные поручения, кассовые ордера, акты).
3. **Доля автоматически сопоставленных банковских операций от общего числа операций (%)**.

4. Снижение количества ошибок в реквизитах и документах (ед./месяц).
5. Рост собираемости членских взносов (% к общему объёму).
6. Скорость формирования отчётности (время формирования стандартного отчёта).

6.3. Оценка ресурсов на разработку и внедрение системы

6.3.1. Трудовые ресурсы

Этап работ	Трудозатраты, чел-мес
Анализ деятельности организации и формирование требований	1,0
Проектирование архитектуры и базы данных	1,0
Разработка подсистем учета и планирования	2,0
Реализация модуля отчетности и аналитики	1,0
Тестирование и опытная эксплуатация	0,5
Внедрение и обучение пользователей	0,5
Итого	6,0

При средней стоимости 1 чел-мес = 110 000 руб:

Затраты на разработку = 6,0 × 110 000 = 660 000 руб

6.3.2. Технические ресурсы

Статья затрат	Сумма, руб
Серверное оборудование / облачный хостинг (1 год)	120 000
Настройка резервного копирования и безопасности	70 000
Сопровождение и поддержка (1 год)	150 000
Итого	340 000

6.3.3. Общие инвестиционные затраты

$$Z = 660\ 000 + 340\ 000 = 1\ 000\ 000 \text{ руб}$$

6.4. Расчёт годового экономического эффекта

6.4.1. Экономия трудозатрат персонала

До внедрения системы:

- бухгалтер — 40 ч/мес на ручной учёт, сверки и отчёты;
- специалист по работе с членами — 30 ч/мес;
- администратор мероприятий — 20 ч/мес.

Итого: 90 ч/мес.

После внедрения системы:

- автоматизация начислений, документов и отчетности снижает затраты до **35 ч/мес.**

Экономия:

- $(90 - 35) \times 12 = 660 \text{ ч/год.}$

При средней стоимости **1 часа = 700 руб**:

$$E_1 = 660 \times 700 = 462\ 000 \text{ руб/год}$$

6.4.2. Снижение потерь от ошибок и несвоевременных операций

Ручное ведение учета приводит к:

- ошибкам в платежных документах;
- задержкам выплат получателям помощи;
- несопоставленным банковским операциям.

Оценим снижение потерь консервативно:

$$E_2 = 120\ 000 \text{ руб/год}$$

6.4.3. Рост поступлений от членских взносов и пожертвований

Внедрение системы:

- повышает прозрачность начислений;
- обеспечивает контроль задолженностей;
- автоматизирует уведомления и аналитику.

Предположим:

- годовой объём взносов и пожертвований — **6 000 000 руб**;
- рост собираемости за счёт автоматизации — **3%**.

$$E_3 = 6\ 000\ 000 \times 0,03 = 180\ 000 \text{ руб/год}$$

6.4.4. Совокупный годовой экономический эффект

$$E_{\text{год}} = E_1 + E_2 + E_3$$

$$E_{\text{год}} = 462\ 000 + 120\ 000 + 180\ 000 = 762\ 000 \text{ руб/год}$$

6.5. Расчёт срока окупаемости проекта

Срок окупаемости определяется по формуле:

$$T = Z / E_{\text{год}}$$

$$T = 1\ 000\ 000 / 762\ 000 \approx 1,31 \text{ года}$$

6.6. Вывод по технико-экономическому обоснованию

Внедрение системы автоматизации деятельности благотворительной организации требует первоначальных инвестиций в размере около **1 млн руб.** При этом ожидаемый годовой экономический эффект составляет около **762 тыс. руб.**, что обеспечивает **срок окупаемости порядка 1,3 года.**

Кроме прямого экономического эффекта, внедрение системы обеспечивает:

- повышение прозрачности деятельности организации;
- улучшение контроля целевого использования средств;
- рост доверия со стороны членов и жертвователей;
- повышение управляемости и устойчивости деятельности организации.