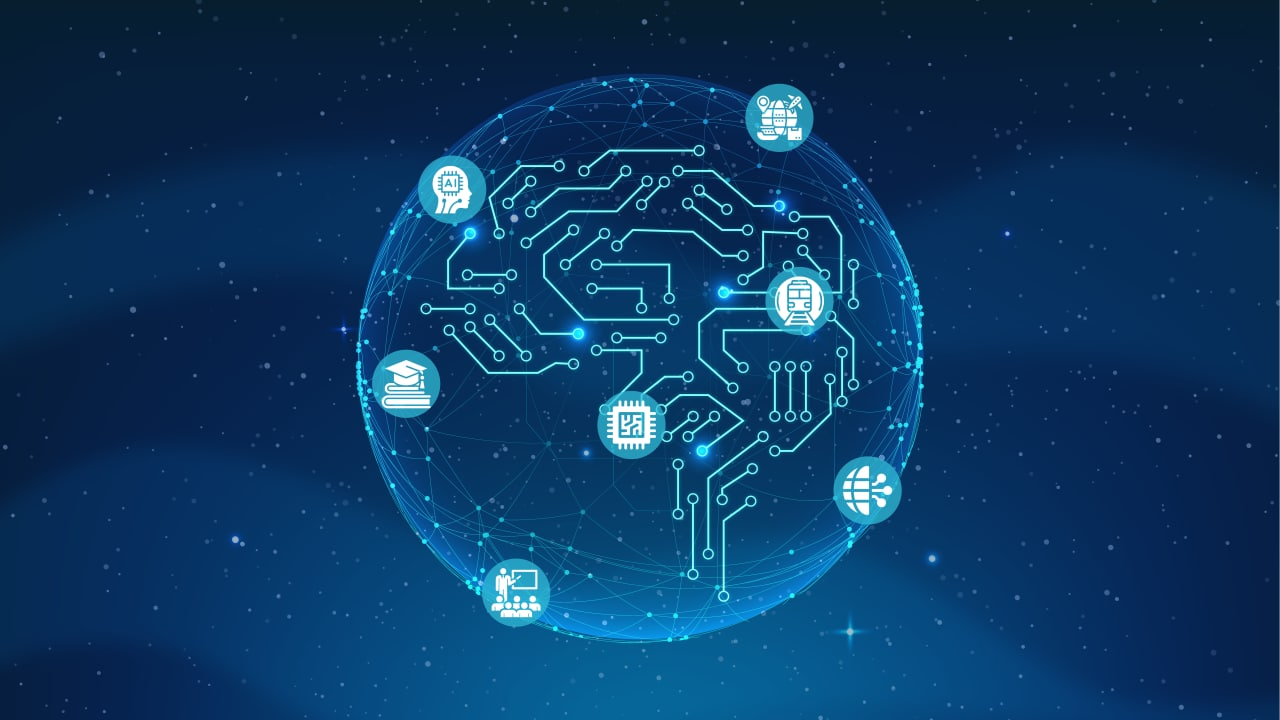
**ПРАКТИЧЕСКИЕ ЗАНЯТИЯ**

**«Жизненный цикл ИТ проекта»**

****

**Содержание**

[Практическое задание № 1. Моделирование бизнес процессов «как есть» и «как будет». 5](#_m4cqyean1pix)

[Практическое задание № 2. Моделирование бизнес процессов «как есть» и «как будет». 9](#_u5v1i6wxndyw)

[Практическое задание № 3. Оценка затрат на реализацию проекта. 13](#_kpafeeys9w7e)

[Практическое задание № 4. Оценка экономической эффективности ИТ проекта. 19](#_vyifp0wzbfgk)

[Практическое задание № 5. Разработка дорожной карты проекта. 23](#_g59x5qz5svl)

[Практическое задание № 6. Постановка задач членам команды. 29](#_mmzcngk05bs8)

[Практическое задание № 7. Проектирование ИТ проекта. 34](#_g0oszuxw1ns)

[Практическое задание № 8. Проектирование ИТ проекта. 40](#_wtxgodog1vhk)

[Практическое задание № 9. Проектирование ИТ проекта. 50](#_mpqlwkr4dcre)

[Практическое задание № 10. Разработка Технического задания. 58](#_ds2hw5otzdij)

[Практическое задание № 11. Разработка Технического задания. 66](#_fmp7kpzcordj)

[Практическое задание № 12. Разработка Технического задания. 68](#_47lw1ptw2fdp)

[Практическое задание № 13. Разработка Рабочих документов. 72](#_t37yxg6bbgnl)

[Практическое задание № 14. Разработка Рабочих документов. 76](#_kbcalvrdhns6)

[Практическое задание № 15. Разработка Программы и методики испытаний. 83](#_z9yc27t4xa7x)

[Практическое задание № 16. Презентация ИТ проекта. 90](#_66n941towpvy)

[Приложение 1. Исходные данные для выполнения практических заданий 91](#_vazc3wij6kn)

[Приложение 2. Исходные данные для выполнения практического задания №4 92](#_jyfnzwei1r1b)

**ВВЕДЕНИЕ**

Целью изучения данной дисциплины является получение знаний о модели жизненного цикла ИТ-проекта, основном содержании каждого этапа и увязки их с дисциплинами учебного плана.

Задачами данной дисциплины являются приобретение практических навыков планирования реализации ИТ проекта, его презентации и разработки проектной и эксплуатационной документации ИТ-проекта.

Что такое проект и чем отличается от продукта?

**Продукт** - произведенный артефакт, который можно выразить количественно и который может являться как конечным объектом, так и компонентом.

**Проект** - временное предприятие, направленное на создание уникального продукта, услуги или результата. Временный характер проектов определяет существование начала и конца работы проекта или ее фазы. Проекты могут существовать самостоятельно или в составе программы или портфеля.

Возьмем приложение Камера на iPhone. Это продукт, который является неотъемлемой частью другого продукта - самого iPhone. При этом разработка конкретной версии этого приложения является проектом, имеющим даты начала и завершения. Будет выходить следующая iOS, и в рамках продукта iPhone инициируют новый проект по улучшению приложения Камера.

Рассмотрим определения:

**Проект** - это комплекс взаимосвязанных мероприятий, направленный на создание уникального продукта или услуги в условиях временных и ресурсных ограничений

* ограничен по времени
* команда собирается под проект и распускается по завершению
* цель – сделать проект в срок, в бюджет и в необходимом объеме требований
* оптимизация удовлетворенности заказчика

**ИТ-проект** – это краткосрочное усилие по созданию уникального продукта, сервиса или среды, например, замещение старых сервисов новыми, разработка коммерческого сайта, создание новых видов настольных компьютеров или слияние баз данных. Это проект, в рамки которого входят работы, связанные с информационными технологиями.

**Информационные технологии** - это технологии, направленные на создание, развитие и поддержку информационных систем.

В широком смысле под информационными системами понимают следующее: «информационной системой называется комплекс, включающий вычислительное и коммуникационное оборудование, программное обеспечение, лингвистические средства и информационные ресурсы, а также системный персонал и обеспечивающий поддержку динамической информационной модели некоторой части реального мира для удовлетворения информационных **потребностей пользователей".**

**Классификация ИТ-проектов:**

* проекты разработки и развития программного обеспечения;
* проекты внедрения информационных систем;
* инфраструктурные и организационные проекты.

**Жизненный цикл проекта** – это  последовательность фаз [проекта](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82), задаваемая исходя из потребностей управления проектом.

Большинство проектов по разработке проходят этапы:

* Инициализация;
* Планирование;
* Выполнение;
* Контроль и мониторинг;
* Завершение.



Рисунок 1 – Жизненный цикл проекта

**Задание.**

Определить вариант ИТ проекта и его характеристик в соответствии с Приложением 1. Все практические задания в рамках курса «Жизненный цикл ИТ проекта» выполняются в рамках одного варианта.

Сформулруйте цели, задачи и ожидаемые конечные результаты проекта.

Допускается объединение слушателей в группы от 2 до 5 человек.

**Дополнительно:**

Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

* Операционная система Microsoft Windows;
* Интернет-браузер, например, Яндекс браузер или его аналоги  
  Пакет офисных приложений, например, Microsoft Office (Word, Excel, Power Point) или его аналоги;
* Инструмент управления проектом, например, Jira или его аналоги;
* Инструмент проектирования пользовательских интерфейсов, например, Figma или его аналоги;
* Инструмент командной работы, например, Miro или его аналоги;
* Инструменты для бизнес-моделирования процессов, например, Miro или его аналоги.

# **Практическое задание № 1. Моделирование бизнес процессов «как есть» и «как будет».**

Модель любого бизнес-процесса должна содержать информацию о том:

* кто выполняет бизнес-процесс;
* из каких действий состоит данный процесс;
* в какой последовательности они выполняются;
* какая информация используется при этом;
* какие ресурсы необходимы;
* и какой будет достигнут результат.

Вся информация о бизнес-процессе, собранная в ходе его описания, документируется, то есть записывается в так называемую карточку бизнес-процесса.

В карточке бизнес-процесса приводится подробная характеристика всех практических аспектов осуществления каждого действия бизнес-процесса, в частности:

* тип взаимодействия;
* ответственные за выполнение действий;
* входные ресурсы и критерии начала бизнес-процесса;
* результат и критерии завершения бизнес-процесса;
* время выполнения бизнес-процесса;
* финансовые затраты;
* используемые информационные системы и программные средства;
* показатели и индикаторы, характеризующие бизнес-процесс.

Данные системы поддерживает полный жизненный цикл процессов – от их описания, моделирования, проектирования до внедрения, мониторинга, анализа и совершенствования.

Совершенствование бизнес-процесса осуществляется в целях повышения его эффективности.

Состав основных мероприятий программы по совершенствованию бизнес-процесса должен предусматривать:

* Проведение анализа бизнес-процессов
* Выработку рекомендаций по улучшению бизнес-процессов
* Обсуждение и согласование проектных решений с заинтересованными лицами;
* Внедрение проектных решений.

Целью этапа анализа является определение соответствия основных характеристик процесса установленным требованиям, выявление источников отклонений и определение приоритетности возможных действий по улучшению процесса.

Процедура анализа бизнес-процесса проводится на всех стадиях его жизненного цикла в целях:

а) описания существующего бизнес-процесса для разработки его модели;

б) определения степени и уровня автоматизации бизнес-процесса;

в) определения «узких мест»;

г) подготовки предложений по оптимизации процесса с учетом лучших практик для сокращения временных и финансовых затрат.

В отчете по результатам анализа действующего бизнес-процесса приводятся списки выявленных «узких мест», перечисляются ошибки в модели бизнес-процесса и т.п.

Главный вывод проведенного анализа должен содержать обоснованное утверждение о наличии или отсутствии оснований для изменения бизнес-процесса, и в случае наличия оснований для изменения - формирование бизнес-процесса « как будет»

**Задание №1.**

Ознакомьтесь с требованиями ГОСТ Р ИСО 21500— 2014 к организации и управлению проектной деятельности.

Изучите в ГОСТ раздел 4 «Процессы проектного менеджмента».

Опишите «как есть» и «как будет» бизнес-процесс, автоматизируемый в рамках ИТ проекта, соответсвующий вашему варианту (Приложение 1). Определите состав участников процесса, перечень и последовательность выполняемых ими действий и другие необходимые условия. Данные сведите в таблицу.

Выполните следующие шаги:

1. Составить список действий.
2. Перевести действия в задачи.
3. Назначить действия исполнителям.
4. Вычислить финалы процесса.
5. Описать условия (шлюзы).
6. Описать внешние по отношению к процессу сущности.
7. Переложить описания в нотации.

**Пример решения №1.**

*Закупка товара*.

«При необходимости в товаре, которого нет в наличии, продавец создает документ “Заявка на закупку” и направляет его на согласование закупщику. Закупщик проверяет необходимость в закупке данного товара и если закупщик разрешает закупить товар согласно документу “Заявка на закупку”, то продавец информируется о разрешении закупить товар, и закупщик создает документ “Заказ поставщику”. Иначе заявка аннулируется с комментарием, содержащем причину отказа в закупке товара. Продавец информируется об отказе в закупке товара.»

*Шаг 1. Составить список действий*

1. Продавец создает документ “Заявка на закупку”
2. Продавец направляет документ на согласование закупщику
3. Закупщик проверяет необходимость в закупке данного товара
4. Если закупщик разрешает закупить товар
5. Продавец информируется о разрешении закупить товар
6. Закупщик создает документ “Заказ поставщику”
7. Иначе заявка аннулируется с комментарием
8. Продавец информируется об отказе в закупке

*Шаг 2. Перевести действия в задачи.*

| **№п/п** | **Действие** |  | **Задача** |
| --- | --- | --- | --- |
| 1 | Продавец создает документ “Заявка на закупку” | → | Создать документ «Заявка на закупку» |
| 2 | Продавец направляет документ на согласование закупщику | → | Отправить заявку на согласование |
| 3 | Закупщик проверяет необходимость в закупке данного товара | → | Проверить необходимость в закупке данного товара |
| 4 | Если закупщик разрешает закупить товар | → | Одобрить заявку на закупку |
| 5 | Продавец информируется о разрешении закупить товар | → | Информировать о разрешении закупить товар |
| 6 | Закупщик создает документ “Заказ поставщику” | → | Создать документ «Заказ поставщику» |
| 7 | Иначе заявка аннулируется с комментарием | → | Аннулировать заявку, в случае неодобрения |
| 8 | Продавец информируется об отказе в закупке | → | Информировать об отказе в закупке |

*Шаг 3. Назначить действия исполнителям*

| **№п/п** | **Действие** |  | **Задача** |  | **Исполнитель** |
| --- | --- | --- | --- | --- | --- |
| 1 | Продавец создает документ “Заявка на закупку” | → | Создать документ «Заявка на закупку» | → | Продавец |
| 2 | Продавец направляет документ на согласование закупщику | → | Отправить заявку на согласование | → | Продавец |
| 3 | Закупщик проверяет необходимость в закупке данного товара | → | Проверить необходимость в закупке данного товара | → | Закупщик |
| 4 | Если закупщик разрешает закупить товар | → | Одобрить заявку на закупку | → | Закупщик |
| 5 | Продавец информируется о разрешении закупить товар | → | Информировать о разрешении закупить товар | → | Система |
| 6 | Закупщик создает документ “Заказ поставщику” | → | Создать документ «Заказ поставщику» | → | Закупщик |
| 7 | Иначе заявка аннулируется с комментарием | → | Аннулировать заявку, в случае неодобрения | → | Закупщик |
| 8 | Продавец информируется об отказе в закупке | → | Информировать об отказе в закупке | → | Продавец |

*Щаг 4. Вычислить финалы процесса*

«При необходимости в товаре, которого нет в наличии, продавец создает документ “Заявка на закупку” и направляет его на согласование закупщику. Закупщик проверяет необходимость в закупке данного товара и если закупщик разрешает закупить товар согласно документу “Заявка на закупку”, то продавец информируется о разрешении закупить товар, и закупщик **создает документ “Заказ поставщику”**. Иначе заявка аннулируется с комментарием, содержащем причину отказа в закупке товара. Продавец **информируется об отказе в закупке товара**.»

*Шаг 5. Описать условия (шлюзы)*

«При необходимости в товаре, которого нет в наличии, продавец создает документ “Заявка на закупку” и направляет его на согласование закупщику. Закупщик проверяет необходимость в закупке данного товара и **если** **закупщик разрешает** закупить товар согласно документу “Заявка на закупку”, **то** продавец информируется о разрешении закупить товар, и закупщик создает документ “Заказ поставщику”. Иначе заявка аннулируется с комментарием, содержащем причину отказа в закупке товара. Продавец информируется об отказе в закупке товара.»

*Шаг 6. Описать внешние по отношению к процессу сущности*

**Внутренние** - те, которыми мы управляем внутри процесса. Это взаимоотношения внутри компании или подразделения, если бизнес-процесс описывает только какую-то часть работы компании. Ими вы можете управлять, т.е. изменить в случае необходимости последовательность задач, ответственных за выполнение и т.д.

**Внешние** - такие, которыми мы не можем управлять. Это сущности, которые необходимы для достижения результата, но сами они находятся за рамками вашего бизнес-процесса. Самый простой пример – поставщики, сотрудники транспортных компаний, клиенты. В некоторых случаях в число внешних сущностей также входят другие подразделения компании, например, бухгалтерия или производство. Например, вы описываете работу отдела продаж, а другие подразделения работают хорошо, и руководитель компании не желает там ничего менять. В любом случае от их действий зависит получение результата, но управлять ими вы не можете.

# **Практическое задание № 2. Моделирование бизнес процессов «как есть» и «как будет».**

На основе собранной информации, проведенного анализа может быть создана модель процесса «как есть», «как будет». Для визуального изображения модели бизнес-процесса используются различные профессиональные нотации.

На сегодняшний день существует несколько десятков нотаций моделирования бизнес-процессов.

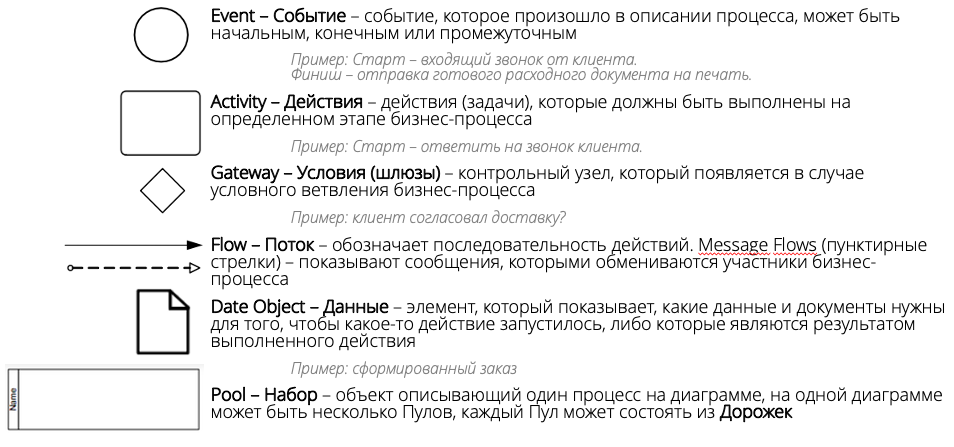
В отечественной практике применяются, как правило, следующие нотации:

* UML;
* EPC;
* IDEF0;
* BPMN 2.0.

Популярность каждой нотации зависит от её применимости в информационных системах моделирования бизнес-процессов BPMS.

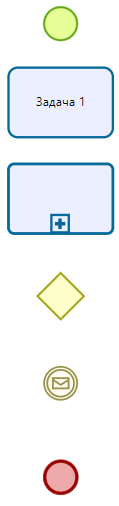
Несмотря на наличие множества нотаций, можно выделить несколько универсальных блоков, с использованием которых можно практиковать построение бизнес-процессов.

Используемые универсальные блоки.



Для построение бизнес-процессов можно использовать такие средства, как bizagi (<https://www.bizagi.com/>), miro (https://miro.com/), MS Visio и др.

Используемые блоки в программе bizagi (<https://www.bizagi.com/>)%

начальное действие

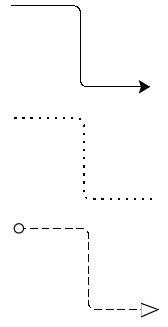
выполняемое действие

составное действие (подпроцесс), задача, имеющая детализацию (декомпозицию)

условие (шлюз) для разделения потоков управления на альтернативные маршруты

сообщение

завершающее действие



последовательность выполнения действий

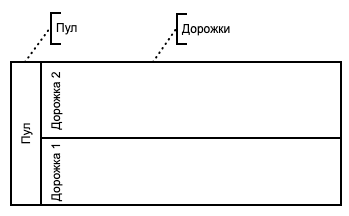
передача объектов данных между действиями бизнес-процесса внутри одного пула, а также для обозначения входов и выходов действий

передачи сообщение и объектов данных между пулами бизнес-процесса



объект данных, артефакт

хранилище данных



**Дорожка** объединяет действия, выполняемые любым субъектом процесса: подразделением, сотрудником или ролью;

**Пул** показывает участника взаимодействия в процессе.

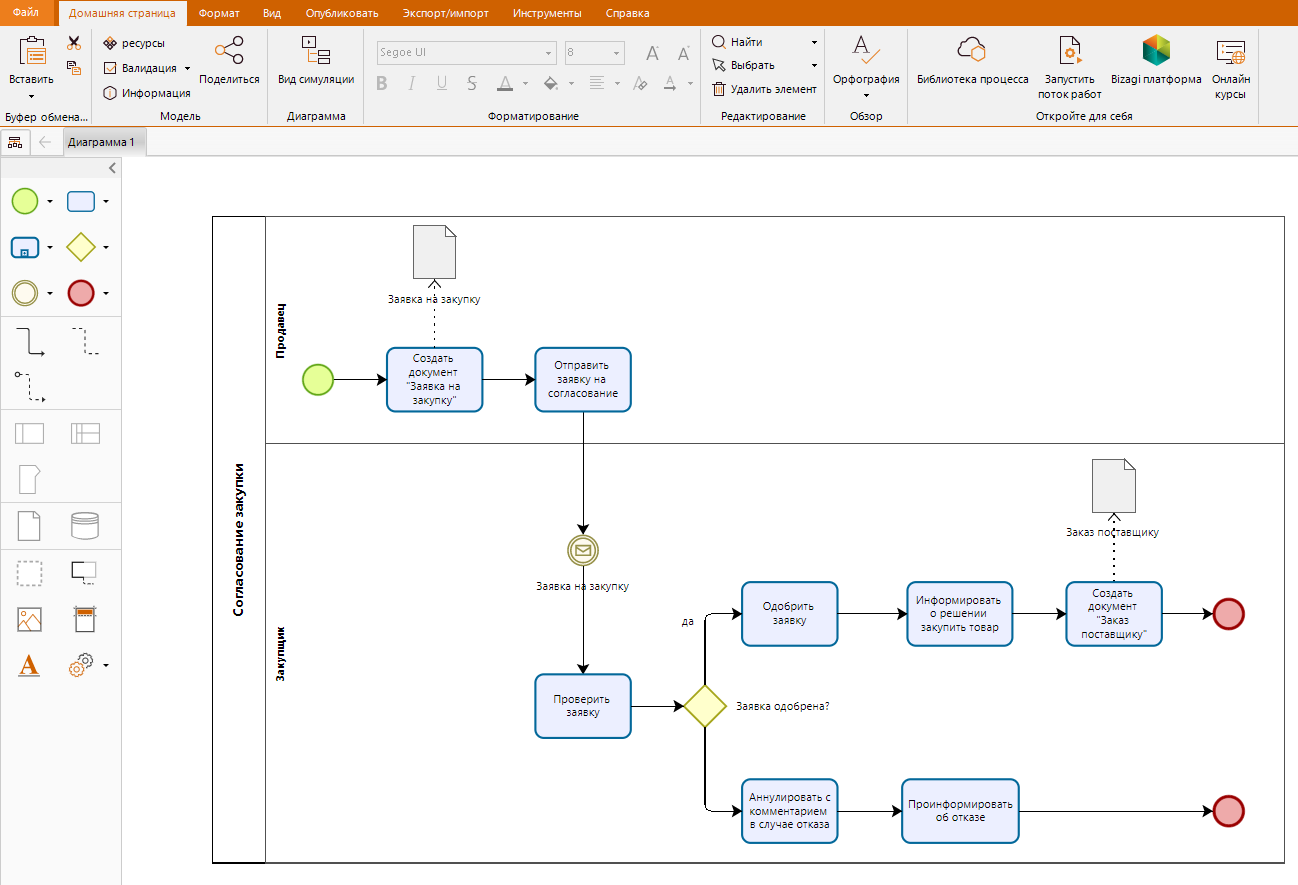


Рисунок 2.1 – Пример сформированной модели бизнес-процесса

**Задание №2.**

Разработайте модель процесса на основании подготовленного описания на Практическом занятии №1, в любой нотации.

**Пример решения №2.**



Рисунок 2.2 - Бизнес-процесс без разделения на исполнителей и условий (шлюзов)

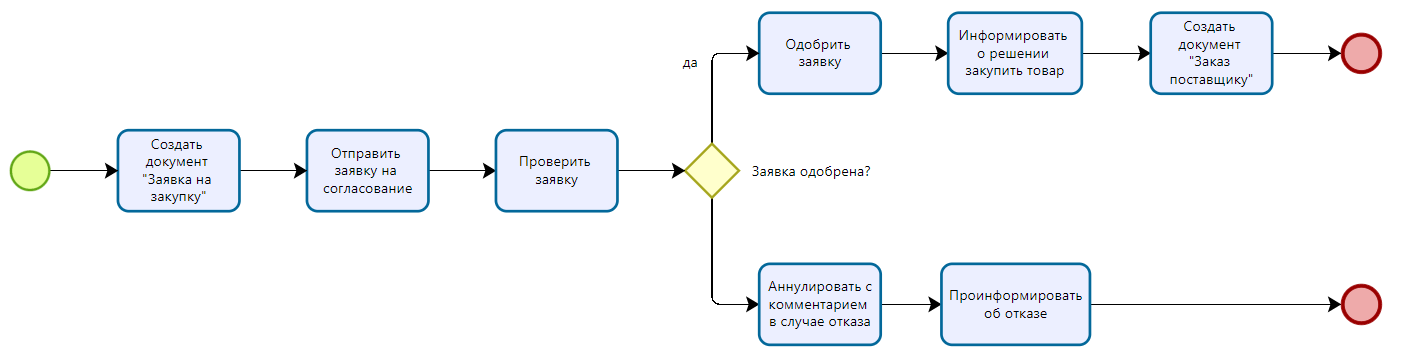


Рисунок 2.3 - Бизнес-процесс с условием (шлюзом)

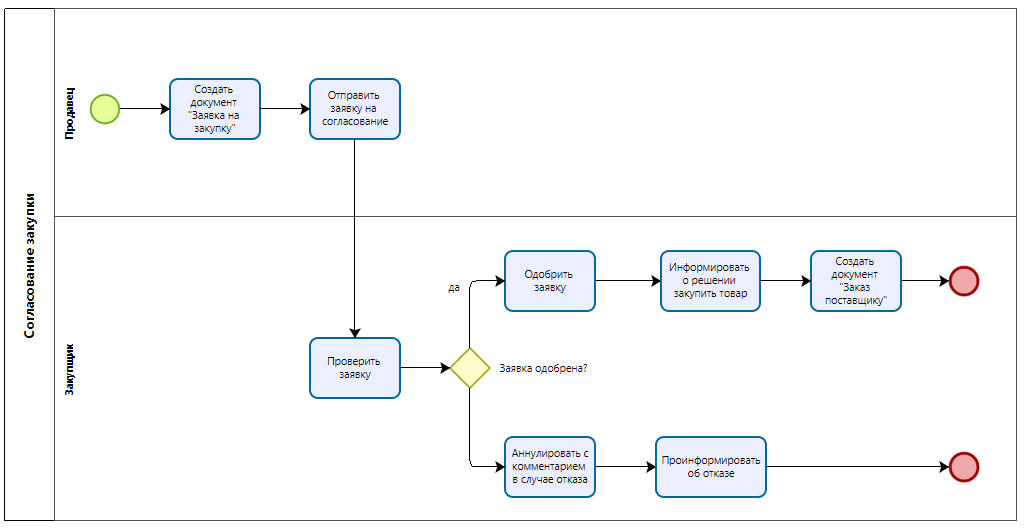


Рисунок 2.4 - Бизнес-процесс с разделением по исполнителем и условием (шлюзом)

# **Практическое задание № 3. Оценка затрат на реализацию проекта.**

Для **расчета оценки инвестиционного проекта и** принятия решения о необходимости старта и реализации проекта проводятся многоэтапные расчеты потенциальных затрат на разрабатываемый продукт, эффекты, которые будут получены и риски, которые могут произойти и которые необходимо предусмотреть для успешного ввода в эксплуатацию и дальнейшего использования информационного ресурса.

Для выполнения указанных расчетов должен быть проведен сбор и верификация исходных данных – эффекты, операционные расходы, капитальные инвестиции, сроки, этапы, пр.

Таблица 3.1 – Исходные данные для оценки инвестиционного проекта

| **№п/п** | **Группа данных** | **Описание данных** |
| --- | --- | --- |
| 1 | Общая информация о проекте | Заказчики  Основные участники процесса  Сроки  Цели  Задачи  Ожидаемые конечные результаты проекта  Идея |
| 2 | Организация и планирование работ по разработке проекта | Оценка оптимального объема работ по проекту и распределение работ по этапам. |
| 3 | Обоснование | Обоснование принятых в проекте решений (выбор технологии, оборудования, способа разработки) |
| **4** | **Расчет затрат на разработку проекта** | **Требуемое количество средств (финансовых, материальных, трудовых), требующихся на разработку данного проекта. Определение объемов и источников финансирования** |
| **5** | **Расчет эксплуатационных затрат** | **Требуемое количество средств (финансовых, материальных, трудовых), требующихся на поддержание данного проекта. Определение объемов и источников финансирования (как правило, собственные средства)** |
| **6** | **Расчет эффектов** | **Экономические и технические результаты, к которым должен привести проект. Обоснование расчетов** |

Основные статьи расходов по проектам:

* **Прямые затраты**. Явно относимые на отдельный проект.

Такие затраты непосредственно относятся и полностью сопоставляются конкретному заказчику или проекту.

* **Косвенные** — разделяемые затраты: Выполняемые от имени всех или нескольких проектов.

Эти затраты разделяются среди нескольких заказчиков или проектов и сопоставляются пропорционально некоторому параметру, например, количеству пользователей или процентному соотношению.

* **Накладные расходы** - затраты, которые не могут быть прямо отнесены к какому-либо проекту.

Эти затраты не могут быть отнесены на заказчика или сервис. Примерами накладных расходов являются зарплата руководителей, общая административная деятельность и так далее.

Аналогия с молоком:

Приведем пример действия этого принципа не из ИТ области: вычисление общей стоимости стакана молока. Если вы будете учитывать только затраты на обслуживание и питание коров, то единица затрат на один стакан молока может составить всего 25 рублей. Однако, когда вы учтете процент стоимости страховки фермы, платежи по закладным и лизингу оборудования фермы, то общая стоимость стакана молока может стать 50 рублей. В конце концов, все это придется платить.



Рисунок 3.1 – Объединение прямых, косвенных и накладных затрат для определения общей стоимости проекта

Таблица 3.2 – Затраты на реализацию ИТ проекта

| **№ п/п** | **Вид затрат** | **Из чего состоят** | **Примечание** |
| --- | --- | --- | --- |
| **1** | **2** | **3** | **5** |
| 1 | **Прямые затраты** (создание и модернизация ОС/НМА) | * ФОТ внутреннего персонала (IT, производственного); * Платежи внешним вендорам; * Материалы и комплектующие как компоненты актива; * ПИР, СМР; | Включаются в бюджет проекта (CapEx) |
| 2 | **Прямые затраты** (прочие затраты по проекту) | * Обучение персонала; * Предпроектные исследования; * Разработка документации по проекту; * Аренда каналов связи, серверов, оборудования только под проект; | Включаются в бюджет проекта |
| * Закупка ПО и серверов под конкретный проект, необходимые для создания ОС/НМА |
| 3 | **Косвенные затраты** (затраты на поддержку ОС/НМА) | * ФОТ внутреннего персонала по тех.поддержке и обслуживанию (IT, производственный); * Платежи внешним вендорам по тех. поддержке и обслуживанию; * Материалы и комплектующие для ремонта; | Включаются в бюджет проекта (*OpEx*) |
| 4 | **Накладные расходы**  (прочие затраты) | * Затраты по общей деятельности, распределяемые по видам деятельности; * ФОТ административного и управленческого персонала; * Общехозяйственные затраты, коммунальные услуги; | Могут включаться пропорционально измерителям |

Рассмотрим определения:

**Основные средства (ОС)** – это основные средства в бухгалтерском и налоговом учете. К основным средствам относятся здания, сооружения, передаточные устройства, рабочие и силовые машины и оборудование, измерительные и регулирующие приборы и устройства, вычислительная техника, транспортные средства, инструмент, производственный и хозяйственный инвентарь и принадлежности, рабочий и продуктивный скот, многолетние насаждения, внутрихозяйственные дороги и прочие средства, а также земельные участки, находящиеся в собственности предприятия, учреждения.

**Нематериальный актив (НМА)** – это идентифицируемые немонетарные активы, не имеющие физической формы; входят в состав внеоборотных активов. Это объекты интеллектуальной собственности, которые используются более 1 года и приносят доход. При этом они не имеют материальной формы и отделимы от других активов.

**Фонд оплаты труда (ФОТ)** – это совокупность средств фирмы, которые были направлены на выплаты коллективу. В ФОТ входят фонд заработной платы (ФЗП, сумма выплат зарплаты всем сотрудникам) и премиальный фонд (ПФ, сумма выплат премий всем сотрудникам), а также компенсационные выплаты.

**Проектно-изыскательские работы (ПИР)** - это комплекс работ по разработке проектных документов, на основе которых будет производиться строительство или реконструкция зданий, реализации инженерных изысканий обеспечивающих изучение условий строительства и выявлению негативных факторов влияющих на него, подготовка документации и схем, составлению сметы на строительство.

**Строительно-монтажные работы (СМР)** – это комплекс работ направлены на строительство зданий и сооружений, ремонт и реконструкцию, монтаж и установка оборудования.

**Капитальные расходы** (*CapEx)* — капитал, использующийся компаниями для приобретения или модернизации физических активов. *CapEx* — это инвестиционные затраты на покупку основных фондов, а также затраты по обслуживанию кредитов на их приобретение

**Операционные затраты или операционные расходы** (Opex) — повседневные затраты компании для ведения бизнеса, производства продуктов и услуг. Сумма операционных расходов и капитальных расходов составляют расходы компании, которые не включаются в прямую себестоимость продуктов или услуг, которые предлагает рынку данная компания.

Для принятия решения Заказчиком о стоимости проекта и выбора Исполнителя работ, осуществляется запрос коммерческих предложений.

Оценка затрат производится Исполнителем на основе данных по срокам, целям, ТЗ.

Возможно заложения фиксированного % от капитальных вложений при отсутствии точной оценки на будущие периоды.

Предварительная стоимость ИТ проекта осуществляется медианным значением по всем полученным предложениям.

**Задание № 3.1**.

Сформировать перечень затрат, необходимых для реализации ИТ проекта в зависимости от характеристик проекта Вашего варианта (Приложение 1). В случае отсутствия необходимых данных – сформировать требования самостоятельно.

Провести исследование уровня заработной платы нужных специалистов.

Провести расчет затрат на реализацию проекта, исходя из продолжительности реализации ИТ проекта.

**Пример решения №3.1.**





**Задание № 3.2**.

Оценить стоимость работ по проекту по запросу предложений. К реализации проекта предполагается привлечение внешнего исполнителя. Предварительно был проведен запрос предложений на открытом рынке.

Результаты представлены в таблице 3.3.

Какие могут предположения при проведения расчета требуемого бюджета по проекту?

На какое ценовое предложение стоит ориентироваться при формировании бюджета?

Таблица 3.3 – Исходные данные для примера решения задания 3.2

| **Порядковый номер заявки** | **Ценовое предложение** |
| --- | --- |
| 1 | 1 196 280 000 |
| 2 | 65 000 000 |
| 3 | 1 338 409 779 |
| 4 | 540 000 000 |
| 5 | 706 621 000 |
| 6 | 980 406 000 |
| 7 | 7 894 000 000 |
| 8 | 899 143 000 |

**Пример решения №3.2.**

1) Проанализируем ценовые предложения:

| **Порядковый номер заявки** | **Ценовое предложение** | **Комментарий** |
| --- | --- | --- |
| 1 | 1 196 280 000 |  |
| 2 | 65 000 000 | Значительное отклонение от средней цены в меньшую сторону – предпочтительно исключить из рассмотрения |
| 3 | 1 338 409 779 |  |
| 4 | 540 000 000 |  |
| 5 | 706 621 000 |  |
| 6 | 980 406 000 |  |
| 7 | 7 894 000 000 | Значительное отклонение от средней цены в большую сторону – предпочтительно исключить из рассмотрения |
| 8 | 899 143 000 |  |

2) Сформируем вывод:

| **Решение:** | | |
| --- | --- | --- |
| Среднее по всем предложениям: | 1 702 482 472 | *Искаженная оценка* |
| Медиана по всем предложениям | 939 774 500 | *Возможная оценка бюджета проекта* |
| Среднее по предложениям без учета исключенных | 943 476 630 | *Возможная оценка бюджета проекта* |
| Медиана по предложениям без учета исключенных | 939 774 500 | *Возможная оценка бюджета проекта* |

# **Практическое задание № 4. Оценка экономической эффективности ИТ проекта.**

Типы эффектов по проектам и примеры расчетов:

* **Сокращение расходов.**

Например:

*Экономия прямых производственных и непроизводственных расходов:*

* снижение ФОТ персонала (высвобождение людей, сокращение ч./ч.) (количество ШЕ \* зарплата с учетом ЕСН);
* сокращение расходуемых материалов (количество\*цена);
* снижение числа ремонтов (количество\*средняя цена);
* сокращение затрат на сервисное обслуживание;
* сокращение общехозяйственных затрат (количество\*средняя цена);
* сокращение затрат на продвижение, маркетинг, аутсорсинг (% от затрат к сокращению\*средние затраты на основе данных предыдущих лет);

*Снижение рисков роста расходов по производственной и управленческой деятельности:*

* сокращение затрат на контроль текущего состояния ОС
* сокращение числа инцидентов (прогноз сокращения\*средняя стоимость устранения)
* исключение налоговых рисков (прогнозный % сокращения\*средние риски на основе данных предыдущих лет)
* снижение количества штрафов(прогнозный % сокращения\*средний объем штрафов за период на основе данных предыдущих лет)
* снижение числа отрицательных экспертиз

*Повышение эффективности бизнес-процессов и рост производительности труда:*

* сокращение простоев (прогноз сокращения простоев\*стоимость простоя (оплата занятости пути);
* снижение транзакционных издержек;
* **Рост доходов.**

Например:

*Увеличение прибыли от перевозок и оказания услуг*

* прирост клиентской базы (рост кол-ва клиентов\*средний чек);
* новые виды услуг (средний чек по новым услугам\*прогноз продаж);
* рост среднего чека (прирост стоимости чека\*среднее количество клиентов);

*Снижение потери доходов от перевозок и оказания услуг:*

* снижение доли простоев ОС (увеличение времени полезного использования ОС\*выработка в период\*стоимость вырабатываемой продукции в период);
* сокращение времени на выполнение операций (прогноз сокращения рабочего времени в часах\*выработка в период\*стоимость вырабатываемой продукции в период);
* **Прочие.**

Например:

*Социально-экономические и имиджевые эффекты*

Основные показатели эффективности проектов:

**Простой срок окупаемости проекта** – момент времени в которой NVв первый раз достигает 0.

**NV** - график чистого денежного потока (поступления – платежи накопленным итогом).

**DPP** (discounted payback period, срок окупаемости с учетом дисконта) – число периодов с момента начала инвестиций до момента, когда NPV становится равным 0. Расчет производится графическим методом, метод переребора.

**NPV** (internal rate of return, ЧПС , чистая приведенная стоимость проекта) – денежный поток, приведенный к текущему моменту (моменту начала инвестиций) при помощи нормированной ставки дисконтирования R.

,

где t0 – текущий интервал времени

**IRR** (internal rate of return, **внутренняя норма прибыли**) - величина ставки дисконтирования (***r***), при которой NPV проекта (на весь срок проекта) становится равной нулю. Расчет производится графическим методом или функция ВСД в MS Excel.

Иными словами, стоимость денег, полученных завтра, не эквивалентна стоимости денег, полученных сегодня. Для учета этих отличий вводится коэффициент дисконтирования, условно равный эффективному проценту по внутреннему кредиту.

Критерии принятия положительного решения по инвестиционному проекту на основе показателей чистой приведенной стоимости (NPV руб), внутренней ставки доходности (IRR%) и срока окупаемости проекта в днях, годах:

* чистая приведенная стоимость проекта (NPV) больше 0;
* Внутренняя норма прибыли (IRR) больше ставки дисконтрования (r);
* DPP (срок окупаемости) меньше срока жизни проекта;

**А**лгоритм принятия решений по инвестиционному проекту (основные шаги)

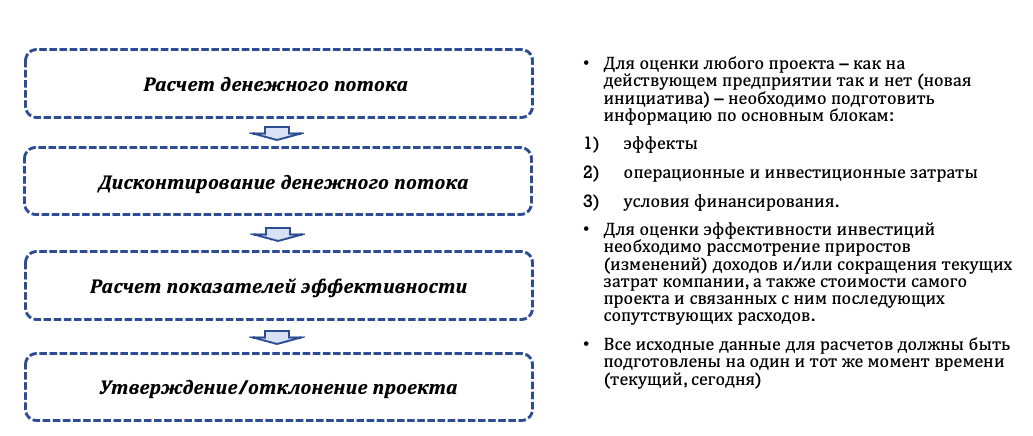


Рисунок 4 – Последовательность принятия решений по инвестиционному проекту

Для оценки любого проекта – как на действующем предприятии так и нет (новая инициатива) – необходимо подготовить информацию по основным блокам:

1. эффекты;
2. операционные и инвестиционные затраты;
3. условия финансирования.

Для оценки эффективности инвестиций необходимо рассмотрение приростов (изменений) доходов и/или сокращения текущих затрат компании, а также стоимости самого проекта и связанных с ним последующих сопутствующих расходов.

Все исходные данные для расчетов должны быть подготовлены на один и тот же момент времени (текущий, сегодня**).**

**Задание №4.1.**

Требуется оценить экономическую эффективность проекта. Исходные данные в соответсвии с Приложением 2.

**Пример решения №4.1:**

Исходные параметры:

* Единовременные инвестиции – 15 млн руб.;
* Срок эксплуатации – 5 лет;
* Выручка от реализации проекта по годам, млн руб.: 6, 7, 8,1, 9,4, 10,9;
* Операционные расходы по годам, млн руб.: 5, 5, 4, 3,1;
* Норма амортизации – 20%;
* Ставка налога на прибыль – 20%;
* Ставка дисконтирования – 15%.

Для расчетов допускается использование шаблона файла Excel с подготовленными формулами.

1. Расчет недостающих данных

| **Показатель** | **год 1** | **год 2** | **год 3** | **год 4** | **год 5** |
| --- | --- | --- | --- | --- | --- |
| Инвестиции | 15 |  |  |  |  |
| Операционные расходы | 5 | 5 | 4 | 3 | 1 |
| Выручка | 6 | 7 | 8,1 | 9,4 | 10,9 |
| *Амортизация* | *15\*20%* | *15\*20%* | *15\*20%* | *15\*20%* | *15\*20%* |
| Амортизация | 3 | 3 | 3 | 3 | 3 |
|  |  |  |  |  |  |
| *Прибыль от продаж* | *=6-5-3* | *=7-5-3* | *=8,1-4-3* | *=9,4-3-3* | *=10,9-1-3* |
| Прибыль от продаж | -2 | -1 | 1,1 | 3,4 | 6,9 |
| Налог на прибыль (20%) | 0 | 0 | 0,22 | 0,68 | 1,38 |
| Чистая прибыль | -2 | -1 | 0,88 | 2,72 | 5,52 |
| **FCF (ЧП + амортизация -инвестиции)** | **-14** | **2** | **3,88** | **5,72** | **8,52** |

1. Расчет показателей эффективности

| **Показатель** | **год 1** | **год 2** | **год 3** | **год 4** | **год 5** |
| --- | --- | --- | --- | --- | --- |
| Период | 0 | 1 | 2 | 3 | 4 |
| Ставка дисконта | 15% | 15% | 15% | 15% | 15% |
| Коэффициент дисконтирования | 100% | 87% | 76% | 66% | 57% |
| FCF | -14 | 2 | 3,88 | 5,72 | 8,52 |
| Накопленный FCF | -14 | -12 | -8,12 | -2,4 | 6,12 |
| DCF | -14 | 2 | 3 | 4 | 5 |
| Накопленный DCF | -14,0 | -12,3 | -9,3 | -5,6 | -0,7 |
| **NPV** |  |  |  |  | **-0,7** |
| **IRR** |  |  |  |  | **13%** |

1. Принятие решения

* NPV < 0 – проект отклоняется;
* IRR < ставки дисконтирования (13% < 15%) - проект отклоняется;
* Проект окупается на 5-й год (накопленный FCF становится положительным) – проект принимается;
* Дисконтированный срок окупаемости < срока жизни проекта – проект отклоняется.

**Задание №4.2.**

Компания планирует реализовать инвестиционный проект. Поступило четыре инвестиционных проекта, по которым уже были посчитаны все критерии. Ставка дисконтирования, принятая в компании = 12%. Дополнительное условие – окупаемость проектов не должна превышать 2-х лет.

Требуется определить наиболее оптимальный проект/проекты.



**Пример решения №4.2:**

1) Срок окупаемости ограничен 2-мя годами (24 мес.) – *исключается* проект «Дельта».

2) Ставка дисконтирования, принятая в компании = 12%.

Сравниваем IRR по проектам со ставкой дисконтирования

Проект «Альфа» : IRR = 15%, 15%>12%, проект *принимается*

Проект «Бета» : IRR = 17%, 17%>12%, проект *принимается*

Проект «Гамма» : IRR = 11%, 11%<12%, проект *исключается*

# **Практическое задание № 5. Разработка дорожной карты проекта.**

**Дорожная карта проекта (roadmap)** — документ, в котором перечислены цели проекта, его ключевые этапы, контрольные даты и ответственные исполнители. Обычно дорожную карту создают в самом начале проекта. На основе карт формируют подробные планы и графики.

Чаще всего дорожные карты применяют в проектах, в которых участвует несколько рабочих групп. Команды таких проектов могут быть полностью автономными, но для них важно придерживаться пунктов дорожной карты, чтобы не сойти с общего маршрута. Например, при разработке новой Программы для ЭВМ участвует несколько рабочих групп — различные подразделения со стороны Заказчика, команда Исполнителя, которые в свою очередь также разбиваются на команды. У каждой команды — свои задачи и свой план, но они соответствуют общей стратегии, описанной в дорожной карте.

Главное отличие дорожной карты от плана в том, что карта представляет собой только общий обзор проекта, без детализации всех этапов, в то время как в плане подробнее расписаны все задачи проекта.

**Дорожная карта** — это визуализация стратегического развития компании или продукта. В общем случае дорожная карта отвечает на вопросы «**кто**?», «**что**?» и «**когда**?». Ключевое преимущество дорожной карты — указание взаимосвязей между различными подразделениями на длительном временном промежутке. Поскольку этот инструмент является понятным для любого специалиста или инвестора, то использовать дорожную карту для объяснения бизнес-стратегии можно буквально на любых переговорах.

На какие вопросы стоит оветить перед началом формирования дорожной карты:

* **Задачи: чему посвящена дорожная карта?** Это может быть, например, небольшой новый проект или пятилетняя стратегия развития компании. От особенностей задачи зависит структура и формат дорожной карты.
* **Структура: в каком виде будет представлена дорожная карта?**Это может быть рисованная линия с контрольными точками, таблица со столбиками и ячейками или [канбан-доска](https://skillbox.ru/media/management/vse_chto_nuzhno_znat_o_kanban/). Выбор структуры зависит от масштаба проекта и от того, насколько его команды зависят друг от друга: можно ли запараллелить некоторые задачи или все они должны идти линейно, друг за другом.
* **Приоритеты: на каких деталях проекта нужно сделать акцент?**Здесь определяют, какие задачи в проекте самые важные, и какие контрольные точки нужно пройти, чтобы успешно их завершить.
* **Ответственные: с кого спрашивать результат каждого этапа?**Обычно ответственными назначают руководителей отделов или самых опытных сотрудников.

Дорожные карты проектов предполагают командную работу, поэтому создавать их важно тоже в команде.

Дорожные карты чаще встречается в методологиях, которые требуют чёткой фиксации задач и порядка работ. Например, в Waterfall — это каскадная, или водопадная, разработка продуктов. Дорожные карты также используют в гибридной модели разработки. Эта модель предполагает, что проектная команда берёт лучшие практики из водопадного подхода и гибких методологий, известных как Agile.

Инструменты для построение дорожных карт:

* **Электронные таблицы Excel или «Google Таблицы».**

В таблицах можно отразить все необходимые элементы дорожных карт — цели, этапы выполнения, сроки, ответственных — и визуализировать их так, чтобы было удобно отслеживать изменения.

Один из распространённых методов создания дорожных карт в таблицах — диаграммы Ганта.

Обычно диаграмма Ганта состоит из двух панелей: в левой дан список этапов работы, в правой — шкала времени (дни, недели, месяцы). Диаграмма включает контрольные точки внутри этапов проекта.

В «Google Таблицах» есть специальный [шаблон](https://docs.google.com/spreadsheets/d/1aVXQ53WBQ2YSaa6JXaaUuMP1k6iDweD4LYyR-S70knQ/edit#gid=1115838130), с которым можно легко создать свою дорожную карту, заполнив необходимые поля. Шаблон доступен для всех, у кого есть аккаунт Gmail.

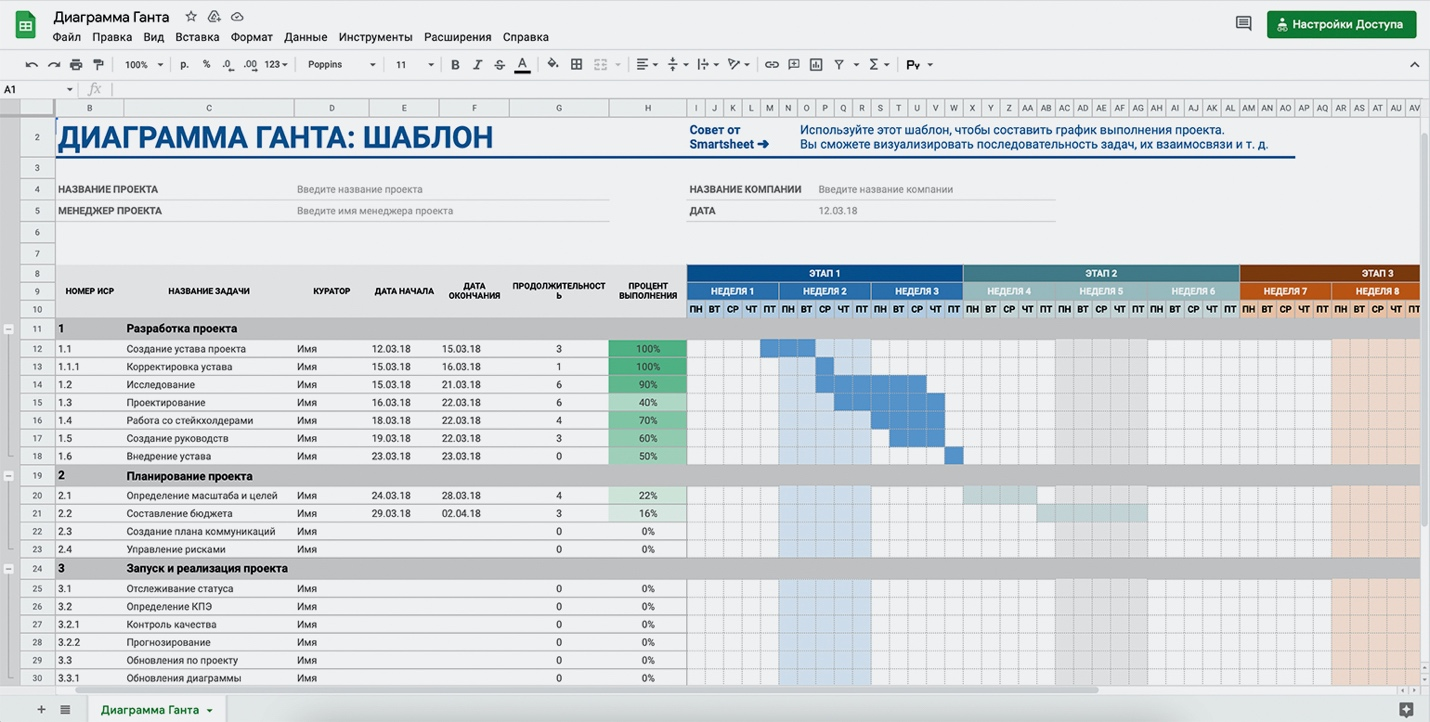


Рисунок 5.1 - Пример дорожной карты проекта в виде диаграммы Ганта в «Google Таблицах»

* **Специальные сервисы.**

Рассмотрим на примере платформы Miro.

[Miro](https://miro.com/app/dashboard/) (https://miro.com/)— сервис для совместной работы команд. В нём можно составить дорожную карту на собственной онлайн-доске или воспользоваться уже готовыми шаблонами. Вот [пример шаблона](https://miro.com/app/board/uXjVO294MnQ=/) дорожной карты в виде временной линии с контрольными точками, где прописаны основные процессы проекта и ответственные исполнители.

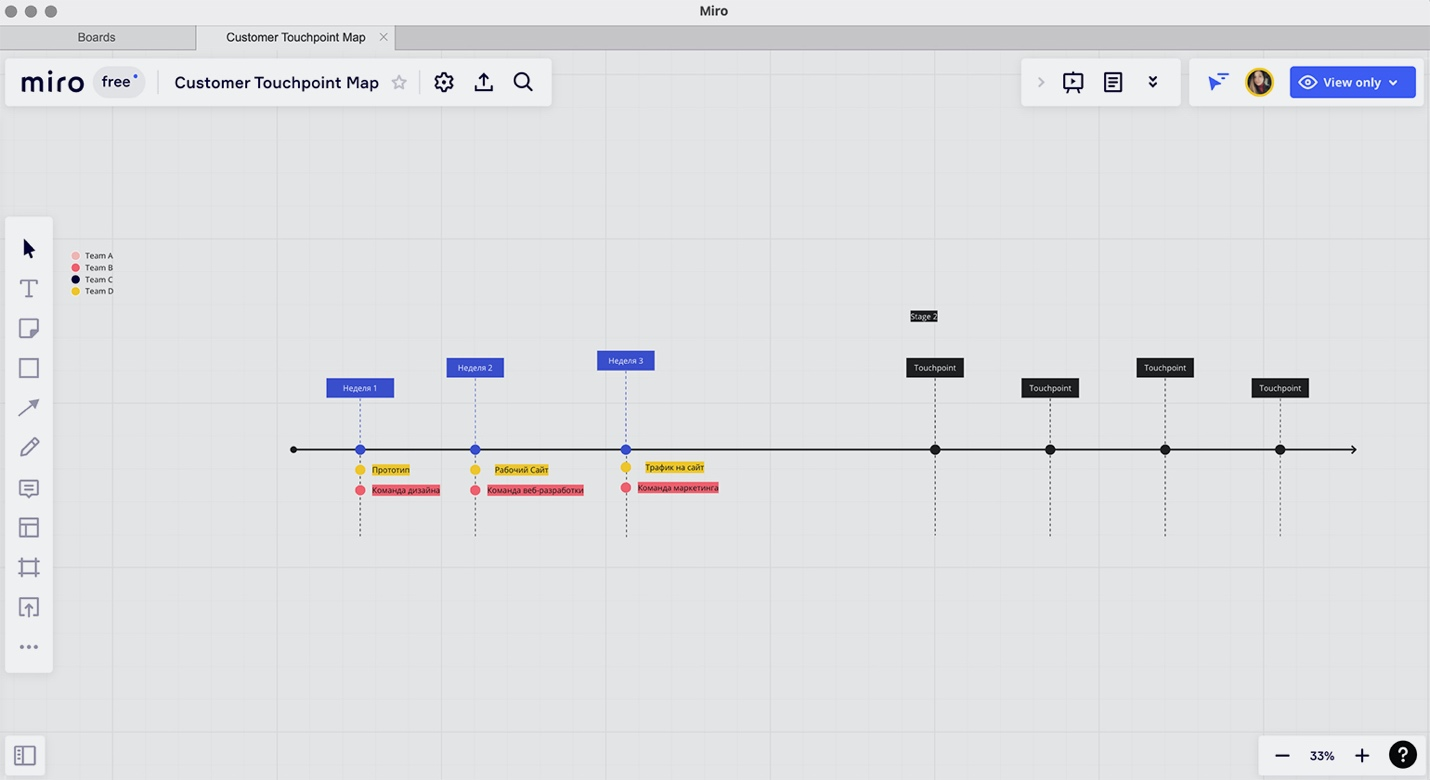


Рисунок 5.2 - Пример дорожной карты проекта в виде временной линии на онлайн-доске Miro



Рисунок 5.3 - Пример дорожной карты проекта в виде дорожной карты на онлайн-доске Miro

**Задание №5.**

Исходя из темы проекта, его характеристик и сроков реализации сформировать дорожную карту.

**Пример решения №5.**

Формирование дорожной карты.

Проект – разработка заказного программного обеспечения для согласования заявок на переадресовку грузов. Выполняется с разработкой интеграционного решения со смежной системой. Срок реализации проекта – 9 месяцев. Программное обеспечение устанавливается на ПТК Заказчика.

|  | **Дорожная карта по развитию АС №1 в части реализации процесса согласования переадресовки** | | | | |  | |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | |
| **№ п/п** | **Наименование работ** | **Сроки** | **Ответ-ный** | **Результат** | **Статус** | |
| 1. | Внесение изменений в Порядок взаимодействия филиалов ОАО «РЖД» при передаче разрешений на переадресовку грузов и порожних вагонов, не принадлежащих перевозчику, на железнодорожную станцию переадресовки | 10.05.22 | Заказчик №1 | Изменения в распоряжение № 2630р от 22 декабря 2016 г |  | |
| 2. | Разработка технического задания на систему | 10.05.22 | Исполнитель №1 | техническое задание |  | |
| 3. | Согласование и утверждение технического задания на систему | 10.06.22 | Исполнитель №1, Заказчик №1, Причастные организации | Утвержденное техническое задание |  | |
| 4. | Формирование заявки на модификацию АС №2 | 10.06.22 | Заказчик №1 | Заявка на модификацию в АСУ ПИ |  | |
| 5. | Разработка проектного решения интеграции АС №1 и АС №2 | 10.06.22 | Исполнитель №1, Исполнитель №2 | проектное решение |  | |
| 6. | Согласование и утверждение проектного решения интеграции АС №1 и АС №2 | 28.06.22 | Исполнитель №1, Исполнитель №2, Заказчик №1, Причастные организации | Утвержденное проектное решение |  | |
| 7. | Модификация программного обеспечения АС №2 в соответствии с проектным решением, разработанным по пункту 5 передача ПО на установку при условии выполнения п.4 и 5 в срок | 17.08.22 | Исполнитель №2 | ПО, наряд на установку ПО в ГВЦ |  | |
| 8. | Разработка программного обеспечения АС №2 передача ПО на установку | 17.08.22 | Исполнитель №1 | ПО, наряд на установку ПО в ГВЦ |  | |
| 9. | Разработка и передача в ГВЦ рабочей и эксплуатационной документации ЕАСАПР СД. Очередь 2021 | 10.08.22 | Исполнитель №1 | Рабочая и эксплуатационная документация |  | |
| 10. | Согласование рабочей и эксплуатационной документации АС №1 | 30.08.22 | Исполнитель №1, Заказчик №1, Причастные организации | Согласованная рабочая и эксплуатационная документация |  | |
| 11. | Разработка комплексной программы и методики испытаний программного обеспечения | 10.08.22 | Исполнитель №1, Исполнитель №2 | ПМИ |  | |
| 12. | Согласование и утверждение комплексной программы и методики испытаний программного обеспечения | 30.08.22 | Исполнитель №1, Исполнитель №2, Заказчик №1, Причастные организации | Утвержденная ПМИ |  | |
| 13. | Направление письма в эксплуатирующую организацию о создании приемочной комиссии | 30.08.22 | Исполнитель №1 | Официальное письмо |  | |
| 14. | Установка модифицированного ПО АС№2 на тестовый полигон | 25.08. 22 | Эксплуатирующая организация | Наряд на установку ПО в ГВЦ |  | |
| 15. | Установка разработанного программного обеспечения АС №1 | 25.08.22 | Эксплуатирующая организация | Наряд на установку ПО в ГВЦ |  | |
| 16. | Проведение комплексного тестирования программного обеспечения по комплексной ПМИ | 07.09.22 | Исполнитель №1, Исполнитель №2, Заказчик №1, Причастные организации | Протокол тестирования ПО |  | |
| 17. | Внесение изменений в программное обеспечение, рабочую и эксплуатационную документацию по результатам тестирования (при необходимости) | 13.09.22 | Исполнитель №1, Исполнитель №2 | Рабочая и эксплуатационная документация |  | |
| 18. | Установка модифицированного ПО АС №2 на промышленный полигон | 21.09.22 | Эксплуатирующая организация | Наряд на установку ПО в ГВЦ |  | |
| 19. | Установка разработанного программного обеспечения АС №1 на промышленный полигон | 21.09.22 | Эксплуатирующая организация | Наряд на установку ПО в ГВЦ |  | |
| 20. | Ввод в опытную эксплуатацию программного обеспечения АС №1 | 23.09.22 | Заказчик №1, Причастные организации, Исполнитель №1 | Акт ввода в ОЭ |  | |
| 21. | Проведение обучающего вебинара для технологов и пользователей | 23.09.22 | Заказчик №1, Причастные организации, Исполнитель №1, Исполнитель №2 | Вебинар |  | |
| 22. | Ввод в постоянную эксплуатацию программного обеспечения АС №1 | 20.10.22 | Заказчик №1, Причастные организации, Исполнитель №1 | Акт ввода в ПЭ |  | |

# **Практическое задание № 6. Постановка задач членам команды.**

Разработка ИТ-продукта – это длительный и трудоемкий процесс. В каждом проекте участвует большое количество специалистов. выполняя несколько ролей. В крупных корпорациях такое не приветствуется.

Эффективность команды во многом зависит от того, насколько хорошо налажена коммуникация внутри нее, соблюдаются ли дедлайны и насколько четко распределены роли, обязанности и задачи.

**Команда проекта** – совокупность отдельных лиц (участников проекта), привлеченных к выполнению работ проекта и ответственных перед руководителем проекта за их выполнение.

Команда проекта — это группа специалистов разного профиля, которые вовлечены в работу над ним.

Основная задача команды — наметить и выполнить все цели. Участники проекта могут существовать на разных уровнях организации.

Рассмотрим стандартный список ролей, которые используются в каждом проекте.

**Менеджер продукта** в команде отвечает за продукт, поэтому может быть не особо осведомлен в том, как развивается проект и на каком он сейчас этапе. Зато о конечном продукте он должен знать все.

В его задачи входит:

* продумывание стратегии развития продукта;
* определение ЦА продукта, понимание ее болей;
* анализ рынка и стратегия продвижения продукта;
* знание особенностей продукта, его плюсов и особенностей;
* презентация продукта.

**Менеджер проекта** следит за соблюдением дедлайнов и требований по проекту, ведет документацию. Работает напрямую с заказчиком и представляет его сторону, представитель со стороны команды разработчиков и работает именно от ее лица.

В его задачи входит:

* бюджет проекта;
* сбор требований по проекту и постановка целей по итогам анализа этих требований;
* делегирование задач;
* контроль за дедлайнами;
* общение с заказчиком по конкретным техническим задачам и нюансам ТЗ;
* распределение зон ответственности между ключевыми специалистами проекта;
* сбор и контроль метрических данных проекта.

**Архитектор** принимает на проекте решения о том, каким будет внутреннее строение программной системы и ее интерфейсов. Среди других задач архитектора в ИТ-проекте можно выделить:

* составление конструкции программного обеспечения (ПО), элементов и их взаимосвязи;
* знание мировых практик разработки ПО и построение системы в зависимости от задач бизнеса, для которого она разрабатывается;
* проектирование архитектуры ПО;
* выполнение требований заказчика к системе: быстрое восстановление, быстрая работа, возможность дальнейшего развития.

**Бизнес Аналитик** нужен на любом проекте по разработке ИТ-продукта, ведь именно он составляет, так называемый, Vision, или конечную концепцию продукта, который и предстоит разрабатывать команде. Он должен хорошо представлять себе конечный вид и функционирование системы.

Задачи бизнес-аналитика на проекте:

* общение с заказчиком и выявление его желаний;
* выявление целей, для которых разрабатывается продукт, какие задачи он должен решать;
* предложение собственных идей по улучшению конечного продукта;
* формирование совместно с заказчиком документации по проекту, в которой подробно описывается, какой именно продукт разрабатывается, каковы его цели и задачи, для какой ЦА предназначен и основные возможности будущей системы;
* определение ключевых заинтересованных лиц проекта ([Stakeholders](https://mellarius.ru/stakeholders)), ожидания которых должны быть удовлетворены;
* составление технического задания (ТЗ);
* (совместно с Заказчиком) определение Бизнес-требований;
* определение Бизнес-ограничений, Требований государственных и международных регуляторов
* (совместно с Ключевым пользователем) выявление Пользовательских требований;
* (совместно с менеджером проекта) разработка концепции Проекта
* планирование работ с требованиями в рамках проекта (анализ, верификация, проверка качества)
* документирование Бизнес и Пользовательских требований
* (при необходимости) реверс-инжиниринг [бизнес-процессов](https://mellarius.ru/analysis).

**Системный аналитик** работает в паре с бизнес-аналитиком, но в отличие от него обращает внимание не на то, как должна выглядеть и какие задачи решать система. Главное, что волнует системного аналитика – это то, как система должна работать в различных вариантах использования и какие у нее функции. Именно это детально описывает системный аналитик проекта.

Задачи системного аналитика включают в себя:

* анализ данных и метрик;
* принятие решений о том, какие именно методы использовать;
* составление технического задания (ТЗ);
* разработка и написание спецификации;
* составление списка требований к системе;
* функциональный анализ системы;
* (совместно с Системным архитектором) определение границ Системы (разрабатываемого ПО) и контекста её работы
* анализ Бизнес-требований и Пользовательских требований, определение Ограничений
* участие в разработке [Проектного решения](https://mellarius.ru/solution-design);
* декомпозиция Бизнес-сценариев на Системные сценарии;
* создание сценариев использования ([use case](https://mellarius.ru/analysis));
* разработка Функциональных и неФункциональных требований, реализующих Системные сценарии
* разработка Пользовательских требований в границах Системы
* документирование Требований
* Ведение репозитория требований

**Технический писатель** рассказывает о продукте и ведет его в массы. Основными задачами такого специалиста становятся:

* написание инструкции по эксплуатации системы;
* составление документации для сотрудников фирмы-заказчика, обеспечивающих поддержку программы в дальнейшем.

**Проектировщик, UX/UIДизайнер** нужен в проекте для построения макетов разрабатываемой системы. При этом он обязательно должен учитывать удобство ее использования клиентом. Отвечает за оформление и внешний вид продукта. Именно он «рисует» все элементы продукта, которые видит заказчик в конечном варианте, подбирает цвета и формы.

Основные задачи проектировщика:

* создание панелей инструментов, меню и кнопок, которые были описаны в ТЗ;
* создание макета расположения графических элементов;
* демонстрирует команде/заказчику черновик продукта, например, как будут осуществляться переходы между элементами и страницами (экранами) продукта.
* определение формы и цвета каждого элемента продукта, чтобы вместе они составляли единую картину;
* прорисовка графических элементов;
* отрисовка баннеров и логотипов для продукта;
* конечное оформление продукта.

**Frontend разработчик** «оживляет» макеты, созданные проектировщиками и дизайнерами. Эти специалист, который воплощает дизайн сайта или приложения в коде.

Задачи верстальщика:

* постановка правил, как браузер должен отобразить тот или иной элемент на web-странице;
* создание эффектов переходов и кликабельности;
* выравнивание текста.

**Backend разработчик** в команде занимается реализацией требований, которые ранее были прописаны аналитиками. Это специалист, который разрабатывает и поддерживает серверную часть веб-приложений. Простыми словами, они разрабатывают программы, выполняемые на серверах, и делают вычисления, нужные для корректной работы веб-ресурсов и серверов. Они создают программы, которые выполняются на серверах и осуществляют вычисления, необходимые для работы сайтов и интернет-серверов.

Задачи разработчика включают в себя:

* воплощение в жизнь функций, которые должна иметь система;
* создание логики, которая отвечает за то, чтобы все функции системы выполнялись именно так, как это и было задумано изначально.

**Тимлид** - это управленец с техническими навыками. Такой специалист, как правило, требуется, если проект масштабный и техническое управление просто необходимо.

Задачи тимлида, обычно включают в себя:

* набор сотрудников в команду, подбор необходимых для выполнения поставленной задачи специалистов. Важно, чтобы в команде была сплоченность и высокий уровень корпоративной культуры.
* принятие решений по вопросам стратегии разработки.
* делегирование задач между специалистами. Контроль за дедлайнами.
* выстраивание коммуникации с другими отделами, а не только внутри команды. Именно тимлид общается с тестировщиками, дизайнерами и прочими специалистами, решая вопросы, не прибегая к помощи заказчика.

**Тестировщик** получает возможность запустить продукт и пользоваться им. Его основные задачи:

* выявление ошибок и недочетов в работе системы;
* давать фидбек по продукту.

В разработке продукта принимают участие большое количество человек, при этом у каждого из них свои задачи. Чтобы добиться эффективного взаимодействия между ними, важно правильно подобрать команду и правильно расставить задачи, а также осуществлять их мониторинг.

Инструменты для постановки задач членам команды:

Программное обеспечение для управления проектами — это набор программных продуктов или комплексное программное обеспечение, включающее в себя приложения для следующего набора задач:

* планирования задач;
* составления расписания;
* контроля цены и управления бюджетом;
* распределения ресурсов;
* совместной работы;
* общения;
* быстрого управления;
* документирования и администрирования системы, которая используется совместно для управления крупными проектами.

Рассмотрим на примере платформы Miro и доски Kanban.

**Задание №6.1.**

В качестве руководителя проекта (прджект менеджер) сформулировать цели, которые должны быть достигнуты в ходе реализации и внедрения разработанной системы и сформулировать назначение проекта.

**Пример решения №6.1.**

1. Цель создания автомобиля: повышение качества жизни человека.

Назначение работы: обеспечение перемещения из пункта А в пункт Б.

1. Цель создания системы АСКМ: Целью создания системы является повышение качества работы хозяйства коммерческой работы в сфере грузовых перевозок, а также уровня обеспечения безопасности грузовых перевозок.

Назначение системы АСКМ:

* ведение ревизорской деятельности в сфере грузовой и коммерческой работы;
* мониторинг состояния технических средств коммерческого осмотра;
* ведение единого информационного ресурса, позволяющего хранить и обрабатывать сведения по типовым, местным и непредусмотренным техническим условиям, эскизам размещения и крепления грузов.

**Задание №6.2.**

Сформировать перечень задач команде проекта с использованием доски доски Kanban платформы Miro или аналогах.

**Пример решения №6.2.**

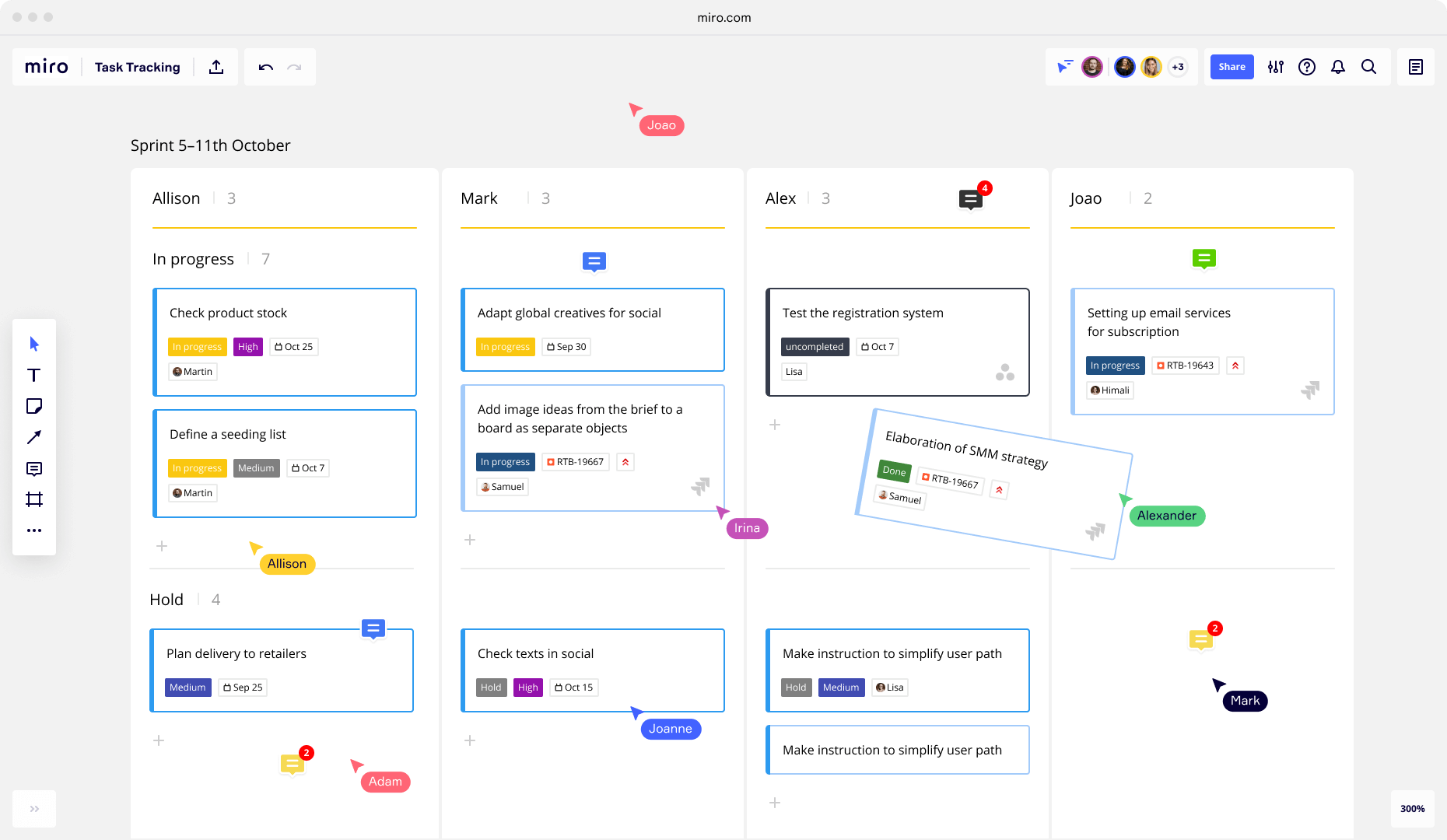


Рисунок 6.1- Пример оформления задач с использованием платформы Miro

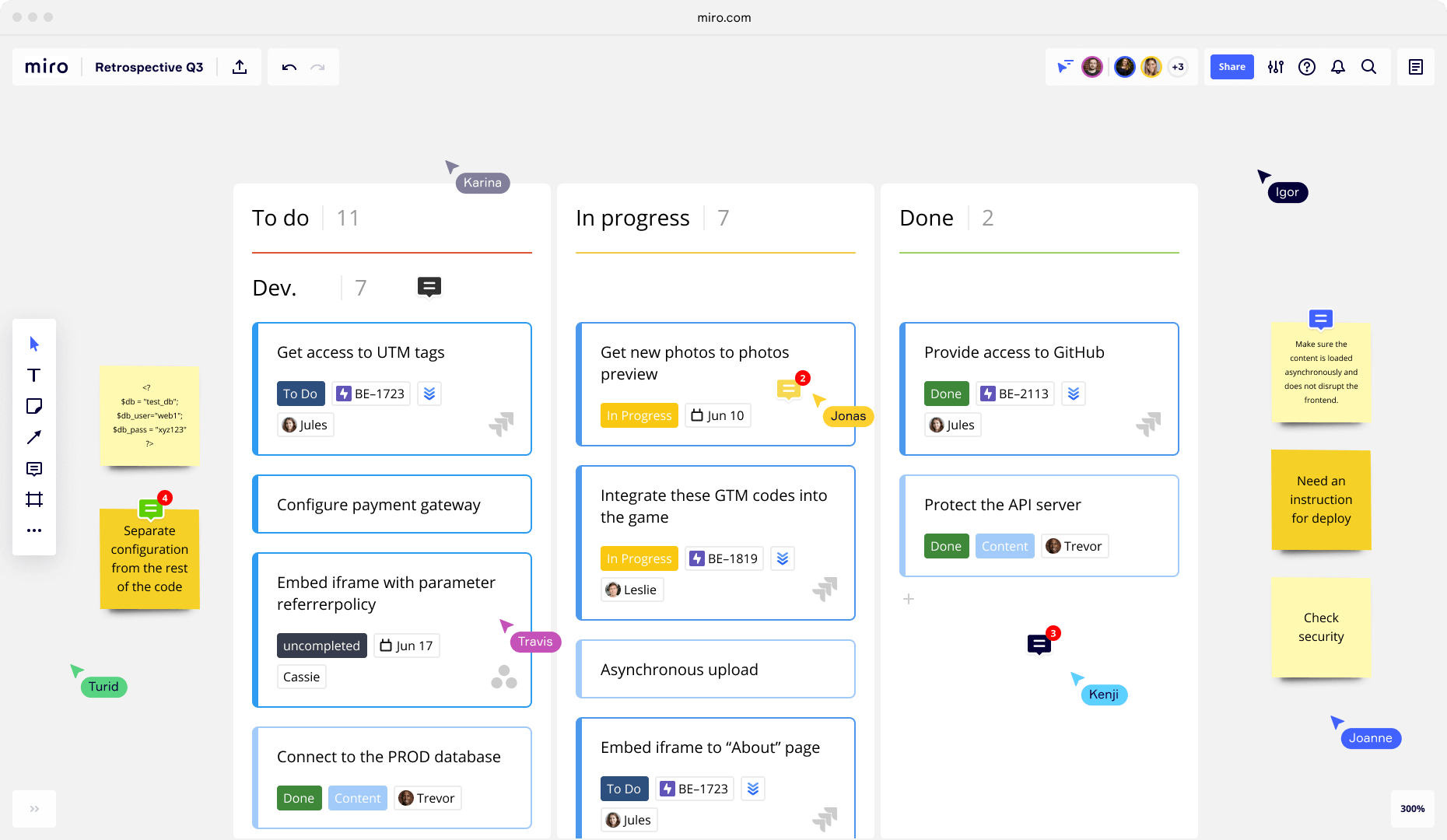


Рисунок 6.2- Пример оформления задач с использованием платформы Miro

# **Практическое задание № 7. Проектирование ИТ проекта.**

При разработке же программного обеспечения важно понимать, что начинать её следует с проектирования — т.е. с полного планирования того, что непосредственно нам придётся разрабатывать, в какие сроки, с какими исходными данными и ожидаемым результатом.

В качестве артефакта, который будет получен в результате – это Техническое задание на разрабатываемую систему. Оно применимо при создании крупных проектов, имеющих большой срок реализации и большое количество заинтересованных сторон.

В рамках курса практических заданий будет посвящено несколько занятий непосредственно разработке технического задания. В практических работах №7, 8, 9 рассмотрим основные блоки выполяемых работ, которые потом послужат наполнением и исходной информацией для формирования окончательного варианта ТЗ для дальнейшего согласования и утверждения с заказчиком.

Проектируя ПО заранее, разработчик получает возможность:

* оценить стоимость и время разработки программного продукта,
* исключить потери времени и денег на ненужные действия, вынужденные доработки, длительное согласование,
* избежать разногласий и неудовлетворённости клиента и исполнителя.

При подготовке к проектированию решаются организационные вопросы:

* что заказчик может предоставить (ТЗ, макеты, дизайн), насколько достаточны исходники и какие этапы закрывают — таким образом определяется состав работ,
* бюджет и сроки: на основе имеющихся материалов утверждается примерная стоимость, срок всего проекта, а также срок и точная стоимость ближайшего этапа.

Этапы и результаты проектирования:

1. Описание: совместная работа Заказчика (говорит о пользе продукта, требованиях к работоспособности и внешнему виду) и Исполнителя (предлагает технические и алгоритмические решения).
2. Архитектура: утверждается язык программирования, база данных, серверы и фреймворки.
3. Техническое задание: составляется архитектором на основании описания и ответов заказчика на вопросы, согласовывается с менеджером проекта, затем передается клиенту, производятся правки.
4. Макеты (добавляются к техзаданию): интерфейсов, принципиальные схемы устройства, диаграммы структуры базы данных, схемы взаимодействия компонентов.
5. Контроль: исполнители устраняют замечания менеджера проектов, заказчика, архитектора, тимлида.
6. Утверждение: заказчик проверяет и меняет ТЗ самостоятельно или сообщает список правок проект-менеджеру, замечания устраняются, ТЗ утверждается и прилагается к контракту.

Как результат проектирования, мы получаем техническое задание с понятной и однозначной для заказчика и исполнителя (руководителя проекта, программистов, тестировщиков, дизайнеров и других участников процесса разработки) иллюстрацией ответов на вопросы:

1. Что делаем (описание продукта, функционала, пользователей)?
2. Как делаем (архитектура)?
3. Как проверить, что цель достигнута (тестирование, критерии оценки)?

Техническое задание должно содержать:

* общие данные о проекте (название продукта, кем и для чего будет использоваться);
* общие требования к ПО (к структуре, функциям, в частности приложить схему архитектуры и описать связь подсистем, виды интерфейсов всех составляющих для каждой из ролей пользователей — готовый дизайн или его концепцию);
* подробный план работ (перечень этапов, сроки по ним);
* порядок тестирования и приемки (виды и состав испытаний продукта в целом и отдельных частей);
* перечень действий для запуска продукта;
* требования к документированию процесса и результата разработки.

На этапе проектирования необходимо уделить внимание описанию следующих вопросов:

1. деталей:

* пользователи программного продукта: роли, права и функции;
* описание алгоритмов обработки данных;
* перечень открытых и закрытых протоколов;
* требования к безопасности данных на всем жизненном цикле;
* список компонентов (платных, свободных), которые будут использоваться в разработке;

1. примеров:
   * при наличии аналогов, интегрируемых систем указываются ссылки на них;
   * в описании работы системы приводится описание типичных сценариев взаимодействия с ней пользователей;
   * примеры входящих данных и формат данных взаимодействия подсистем (таблицы, базы, страницы и др.);
   * примеры исходящих данных (виды отчетов и экспортируемых файлов);
2. производительности и надежности:
   * указание уровней нагрузки системы (день, месяц, максимальный);
   * требования к производительности, сохранности;
   * обоснование выбора оборудования запуска программного обеспечения;
   * указание хостинга серверной части.

В качестве формирования бэклога разрабатываемого продукта, проекта рассмотрим процесс создания User Story Mapping – инструмента, который помогут сделать лучше ваши продукты и жизнь клиентов.

**User Story Mapping** (USM, карта пользовательских историй) – инструмент целостного проектирования продукта на основе пользовательского пути.

Для чего применяется USM?

* + для проектирования пользовательского опыта в продукте;
  + для определения границ MVP (минимальной работоспособной версии продукта) и планирования релизов на базе пользовательского сценария;
  + для формирования единого понимания пользователя у команды разработки и заинтересованных лиц.

Для построения USM вам потребуется:

* + инструмент визуализации: стикеры или электронный инструмент вроде Miro или Mural;
  + владелец продукта и команда разработки;
  + представление о пользовательском сценарии.

**Задание №7.1.**

Спроектировать по шагам действия пользователя в продукте на основе его реального сценария.

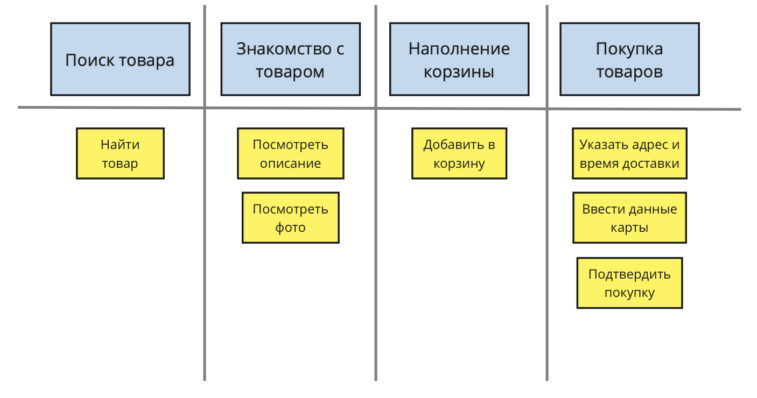
**Пример решения №7.1.**

Магазин цветов решил запустить сайт. Визуализируем опыт клиентов с помощью техники USM.

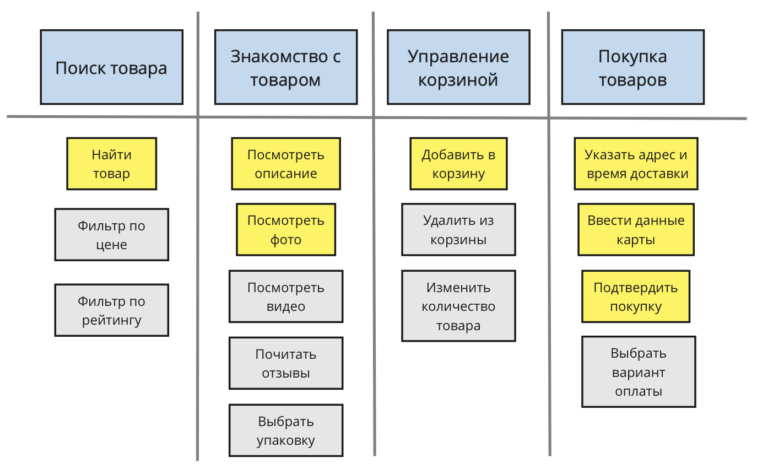
1. Расскажите историю клиента по шагам



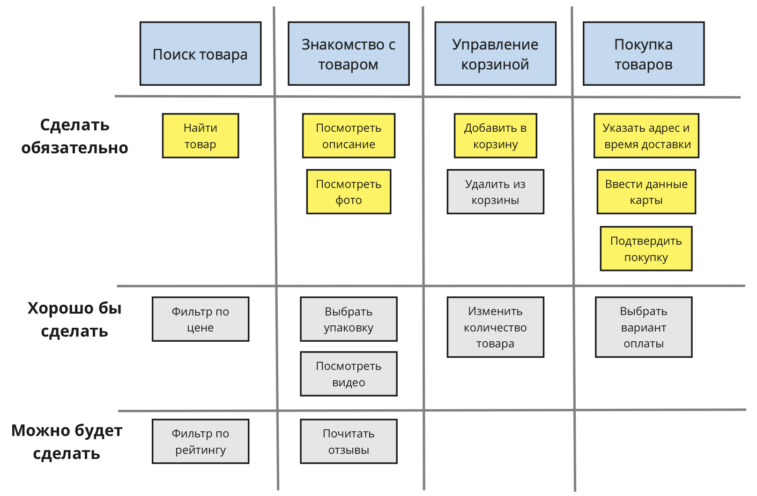
1. Сгруппируйте действия клиента в этапы



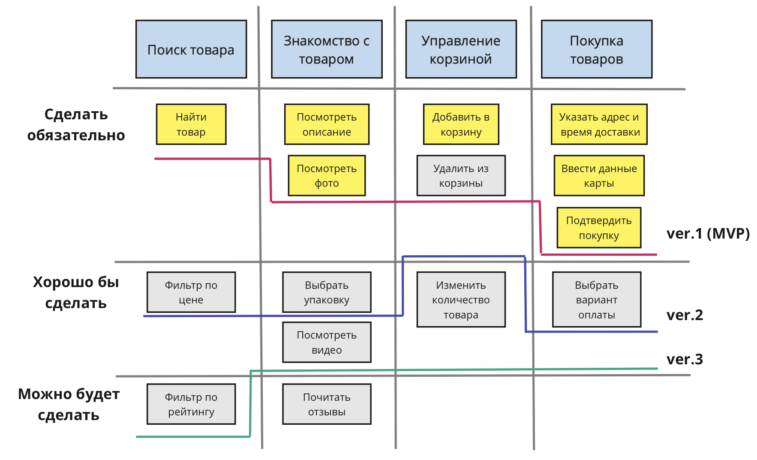
1. Заполнение пробелов в истории ([User Story](https://scrumtrek.ru/blog/product-management/3364/user-story-instruktsiya-po-primeneniyu/))



1. Приоритезируйте истории внутри каждого этапа пути



1. Выделите релизы



1. Рассматриваем приоритизированный бэклог



Обратите внимание, что итоговый приоритет определяется сценарием пользователя. Мы идем слева направо, и только затем по версиям – ведь мы хотим принести больше ценности пользователю. А для пользователя, как правило, ценность тем выше, чем более полно наш продукт закрывает его сценарий.

**Задача №7.2.**

Описать одно действие из USM подробно и детально с точки зрения реализации его в разрабатываемом программном обеспечении.

# **Практическое задание № 8. Проектирование ИТ проекта.**

На основании сформированной UMS и детального описания каждой функции происходит разработка графических пользовательских интерфесов. Для выполнения данной задачи привлекаются:

**UX**-**дизайнер** (user experience designer):

* + проводит исследование пользовательского опыта;
  + проектирует интерфейсы и сервисы, которые вникают в потребности пользователей;
  + строит логические схемы работы интерфейса;
  + тестирует прототипы на целевой аудитории;
  + составляет техническое задание для **UI**-**дизайнера**.

**UI-дизайнер** (user interface designer):

* + разработка графического, тактильного, голосового, звукового дизайна (кнопки, меню, иконки, анимация, иллюстрации, коллажи, шрифты, интерактивные компоненты);
  + создание фирменного стиля;
  + воплощение в визуальных деталях пользовательского опыта.

**Графический интерфейс пользователя (ГИП)** (graphical user interface, GUI) — система средств для взаимодействия пользователя с электронными устройствами, основанная на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов экрана (окон, значков, меню, кнопок, списков и т. п.).

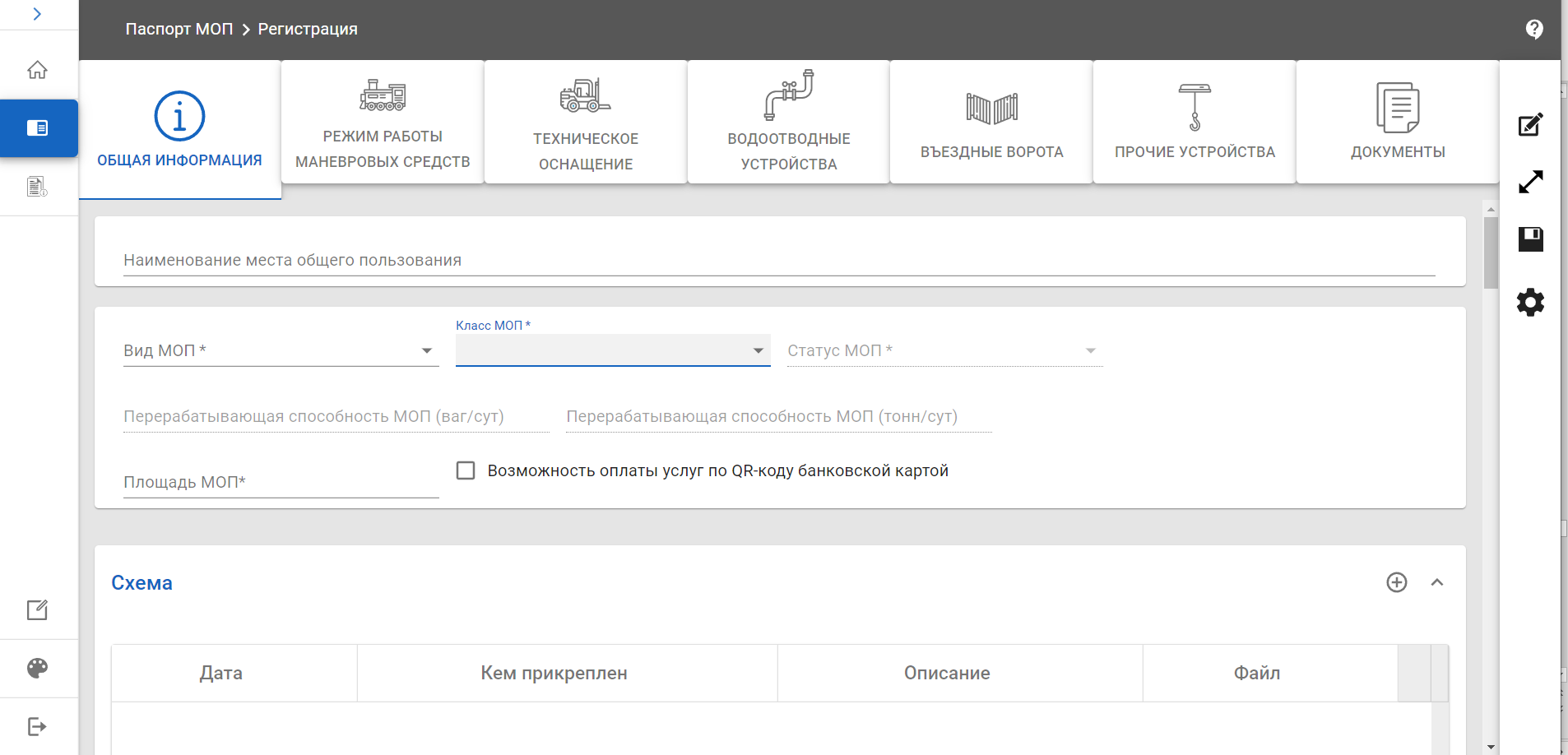


Рисунок 8.1 – Пример графического интерфейса пользователя

ГИП должен отвечать следующим требованиям:

* + Простота — понимание интерфейса на уровне интуиции;
  + Очевидность — предсказуемые элементы, соответствующие ожиданиям пользователя;
  + Уменьшение работы — применение автоматических решений для пользователя (например, автозаполнение данных);
  + Единообразие — узнаваемые элементы дизайна, принятые в большинстве проектов
  + Предотвращение ошибок — предусмотреть поэтапность заполнения данных, статусы и др способы исключения неверных действий;

Для разработки ГИП используются следующие элементы:

* 1. **«Хлебные крошки» —** это цепочка ссылок на «родительские» разделы сайта, через которые пользователь попадает на конкретную страницу. Крошки не только показывают, где пользователь находится сейчас, но и дают ему возможность вернуться в предыдущий раздел.

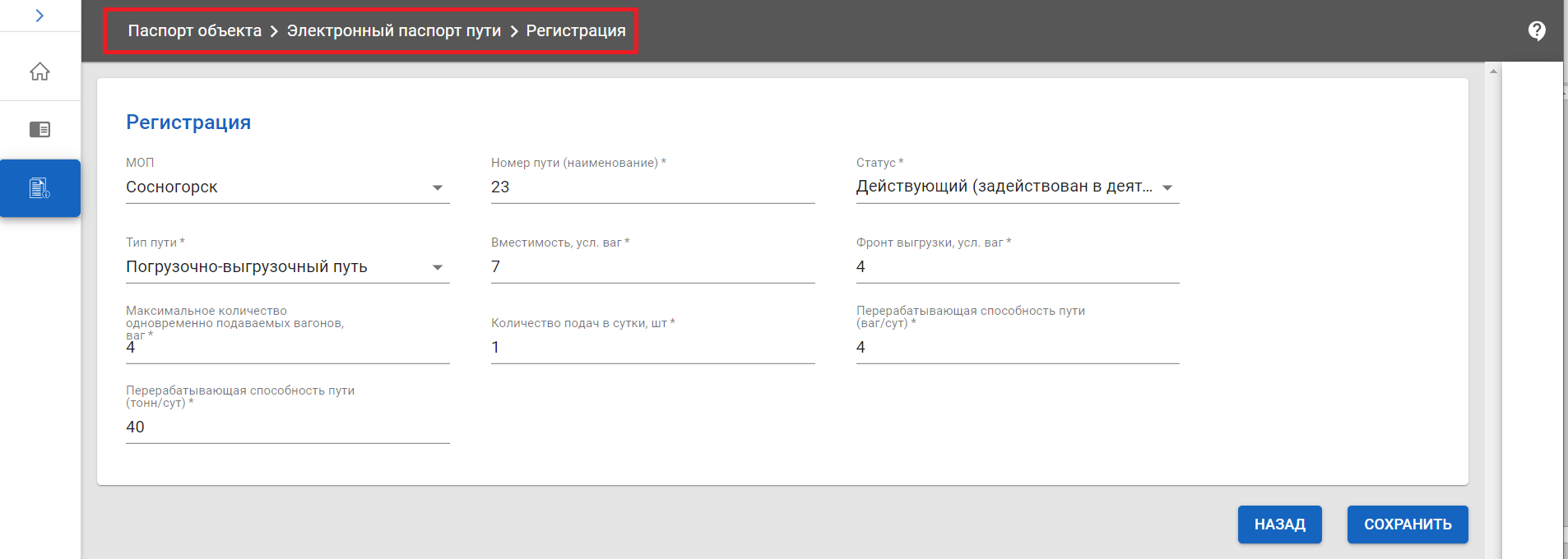
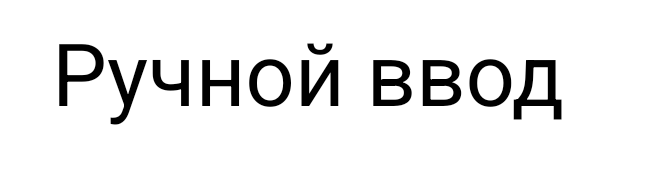
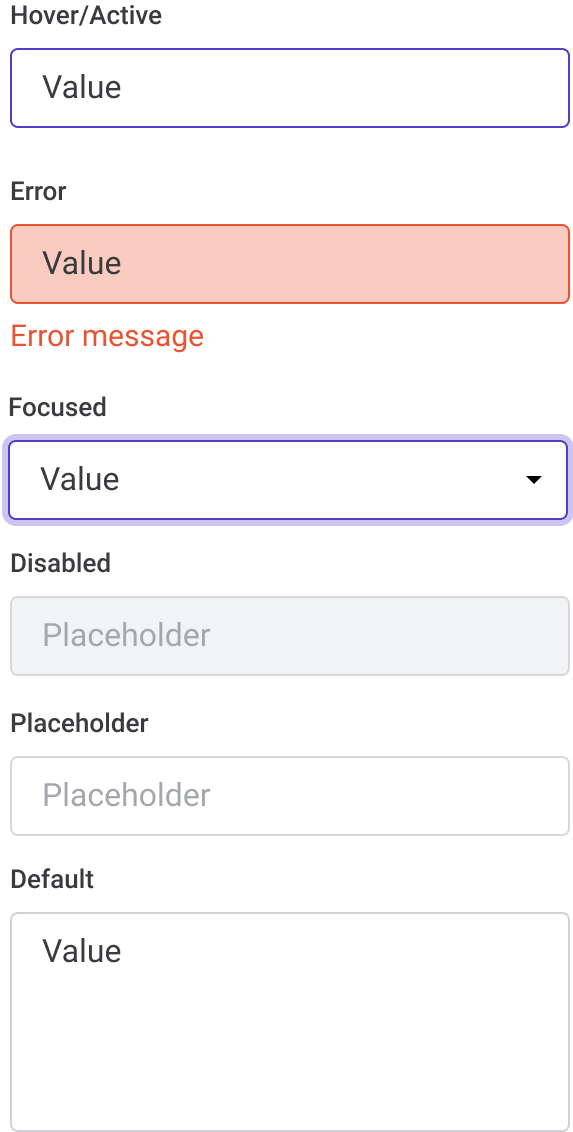
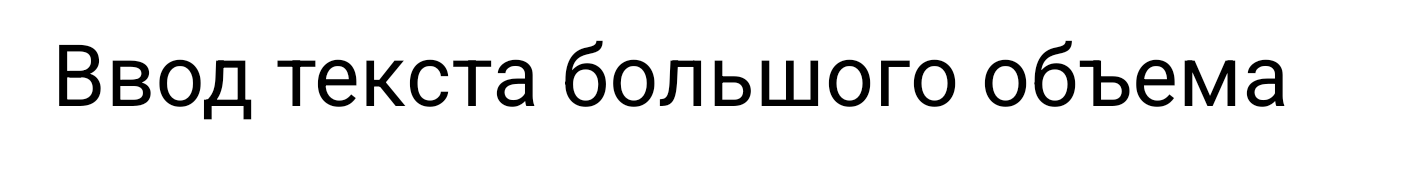
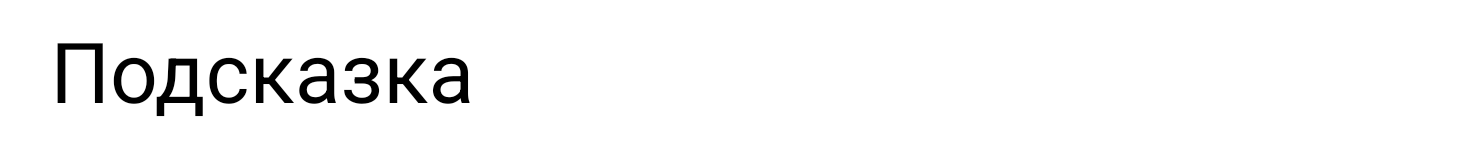
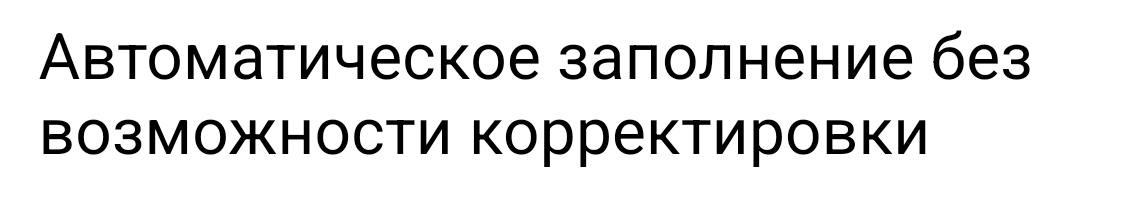
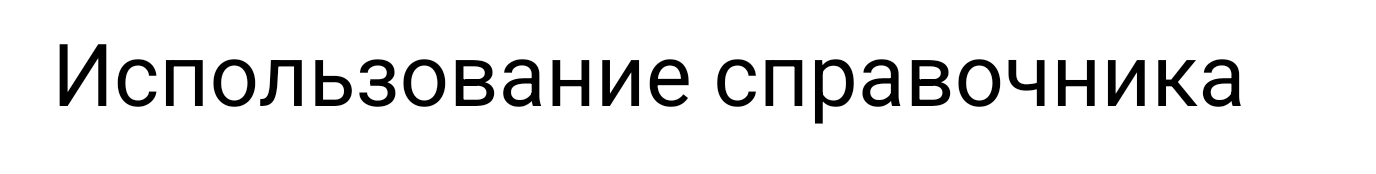


Рисунок 8.2 – Пример элемента «Хлебные крошки»

* 1. **Поля для ввода информации**, основные принципы:
  + Оставлять на экранной форме только те поля, без которых не обойтись;
  + Оставлять подсказки (длина поля, пример заполнения);
  + Отмечать обязательные поля для исключения ошибок при заполнении;
  + Сообщай об ошибках и помогай их исправить.





* 1. **Шрифт:**
  + Размер для каждого вида содержимого экранной формы;
  + Цвет для каждого вида содержимого экранной формы.

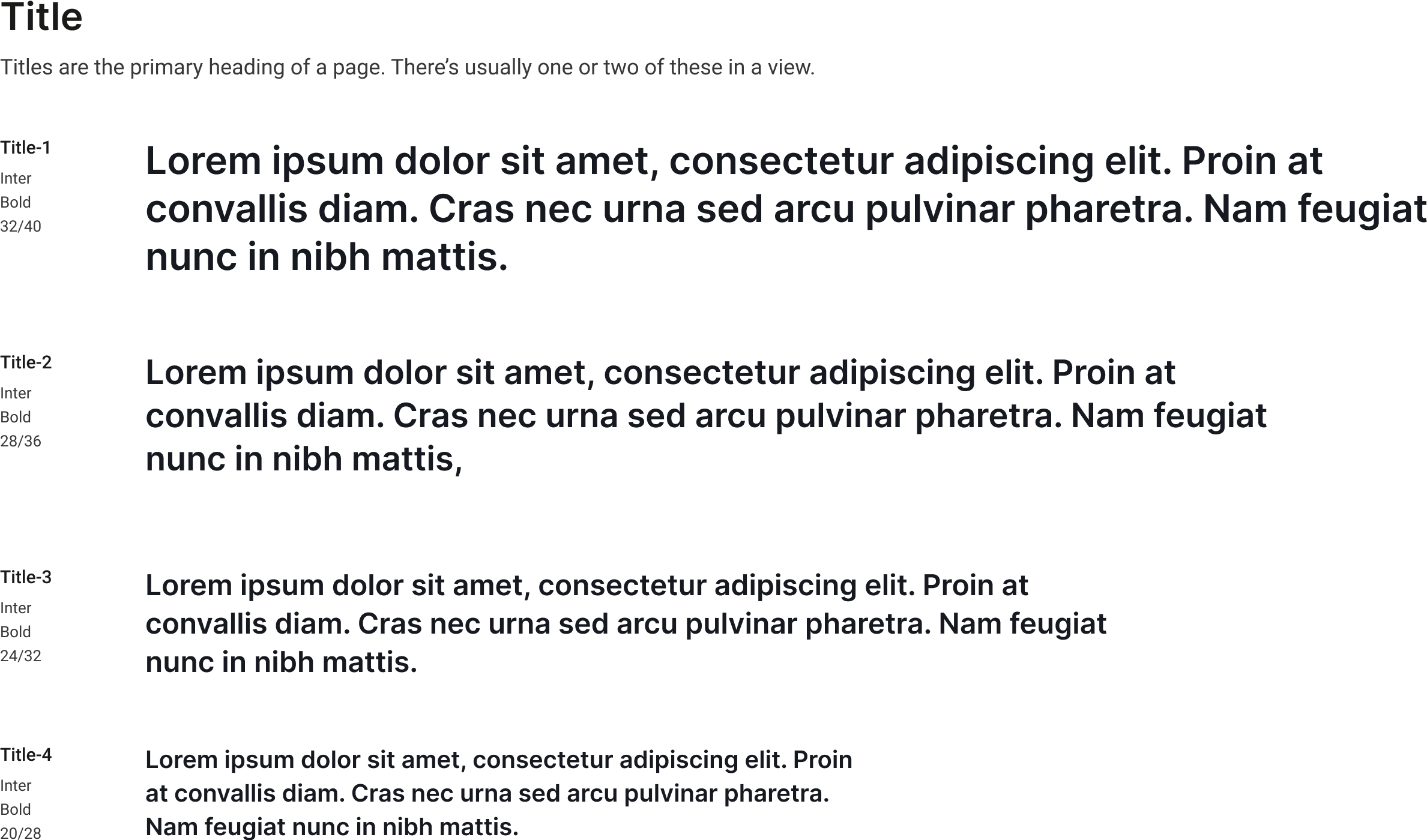


Рисунок 8.3 – Пример элемента «Шрифты»

* 1. **Иконки:**
  + копки «Создать», «Удалить», «Добавить», «Подтвердить» и др.;
  + возвращение на главную станицу;
  + окно профиля пользователя;
  + просмотр детальной информации;
  + и мн. др.

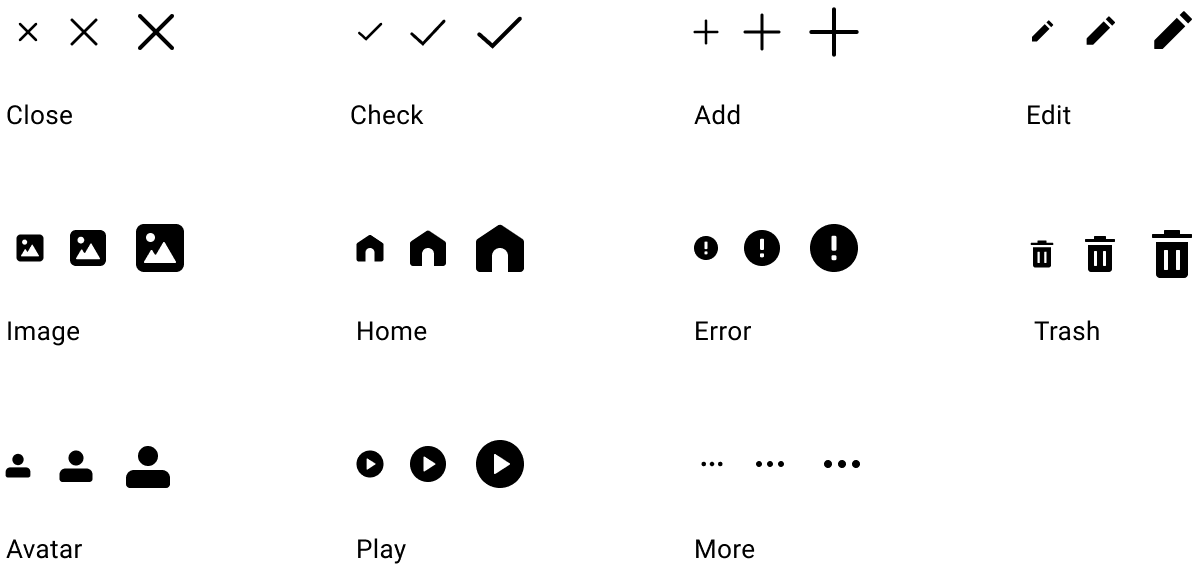


Рисунок 8.4 – Пример элемента «Иконки»

* 1. **Элементы управления выбором:**
  + Радиокнопки
  + Флажки
  + Переключатели
  + Тумблеры
  + и др.



Рисунок 8.5 – Пример элемента «Кнопки»

* 1. **Подсказки**

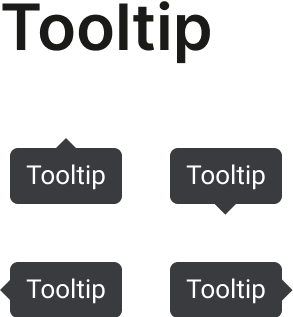


Рисунок 8.6 – Пример элемента «Подсказки»

* 1. **Всплывающие окна**

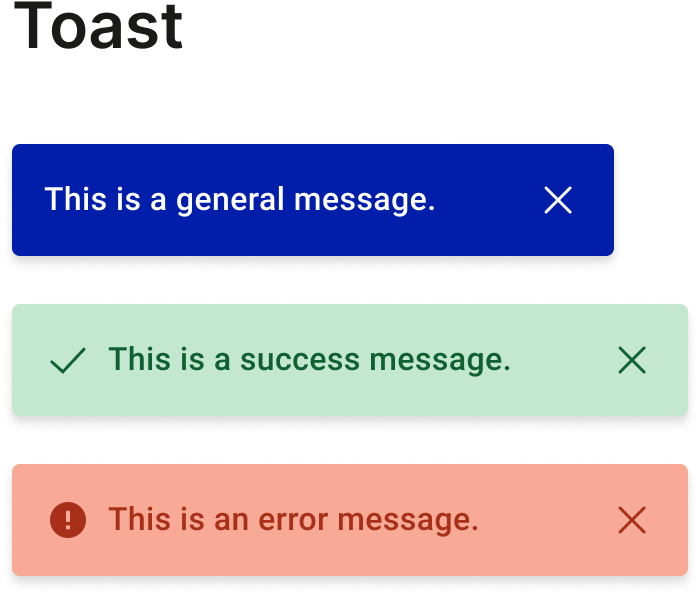


Рисунок 8.7 – Пример элемента «Всплывающие окна»

* 1. **Модальные окна**

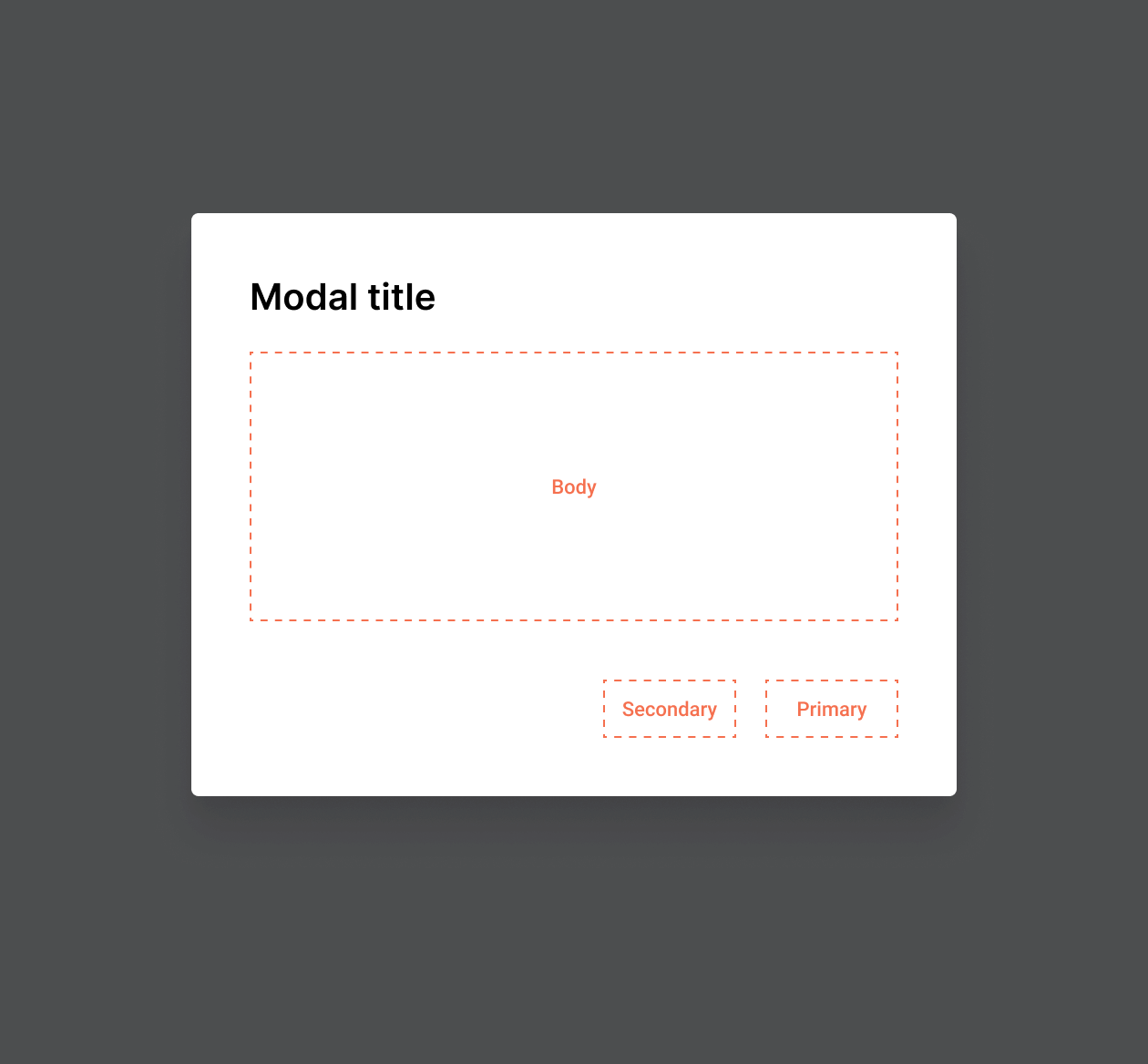


Рисунок 8.8 – Пример элемента «Модальные окна»

Рассмотрим примеры реализованных пользовательских интерфейсов для решения ряда задач:

**Режим Уведомления** (получение системных уведомлений)

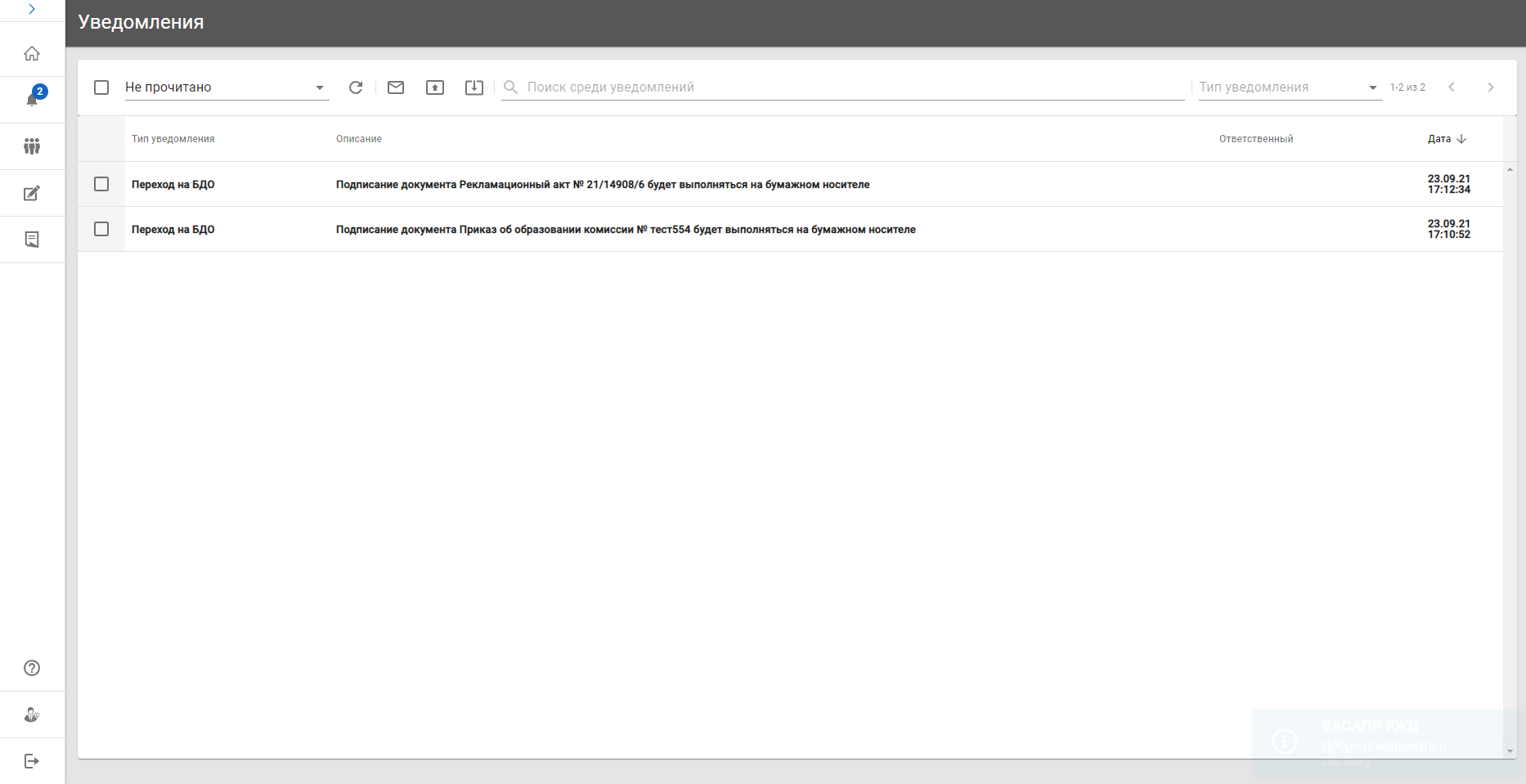


Рисунок 8.9 – Пример реализации режима уведомлений пользователей системы

**Режим Реестр документов** (для подписания ЭП, согласования электронных документов)

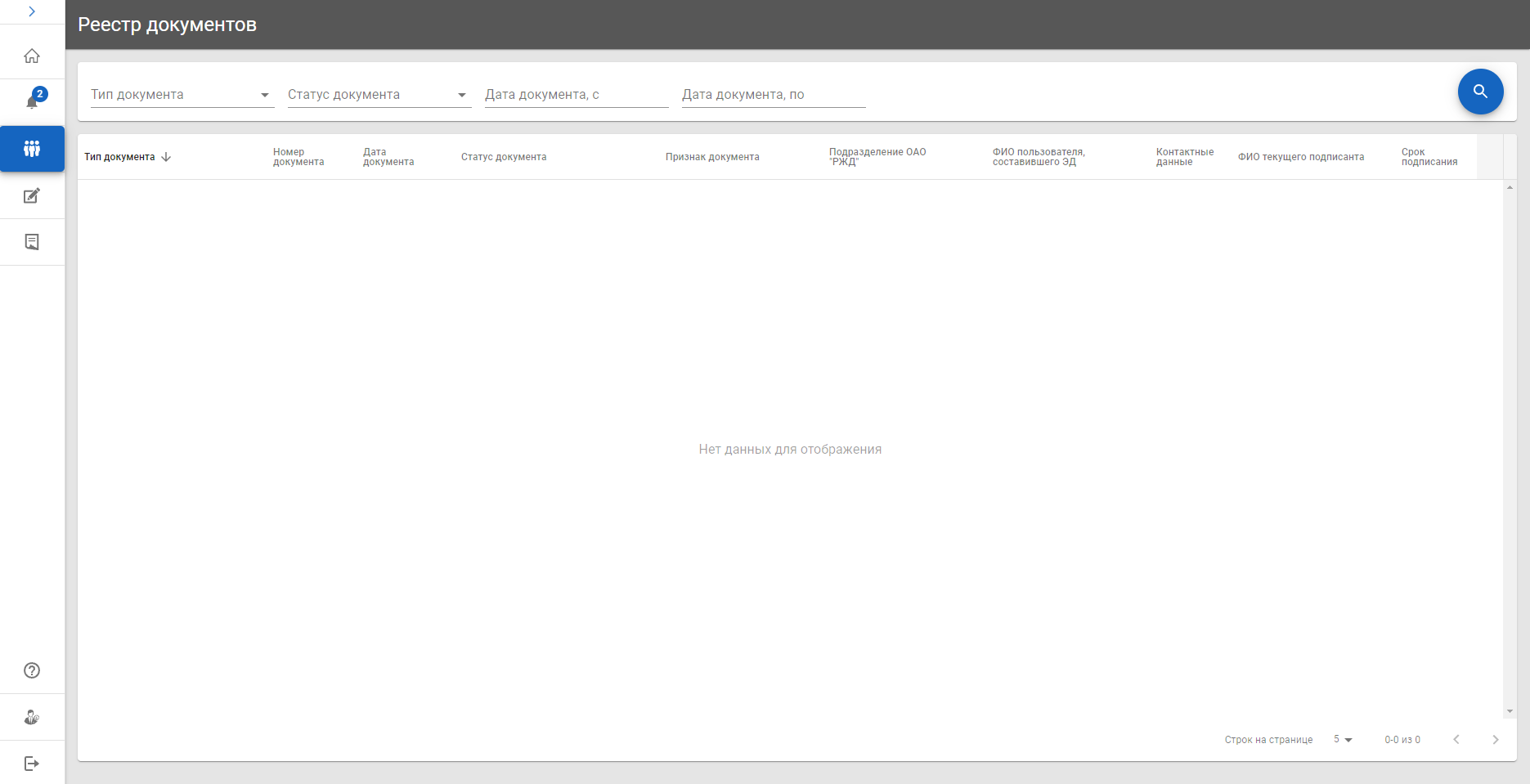


Рисунок 8.10 – Пример реализации режима реестра документов

**Режим** для поэтапной регистрации информации в рамках одного события (документа)

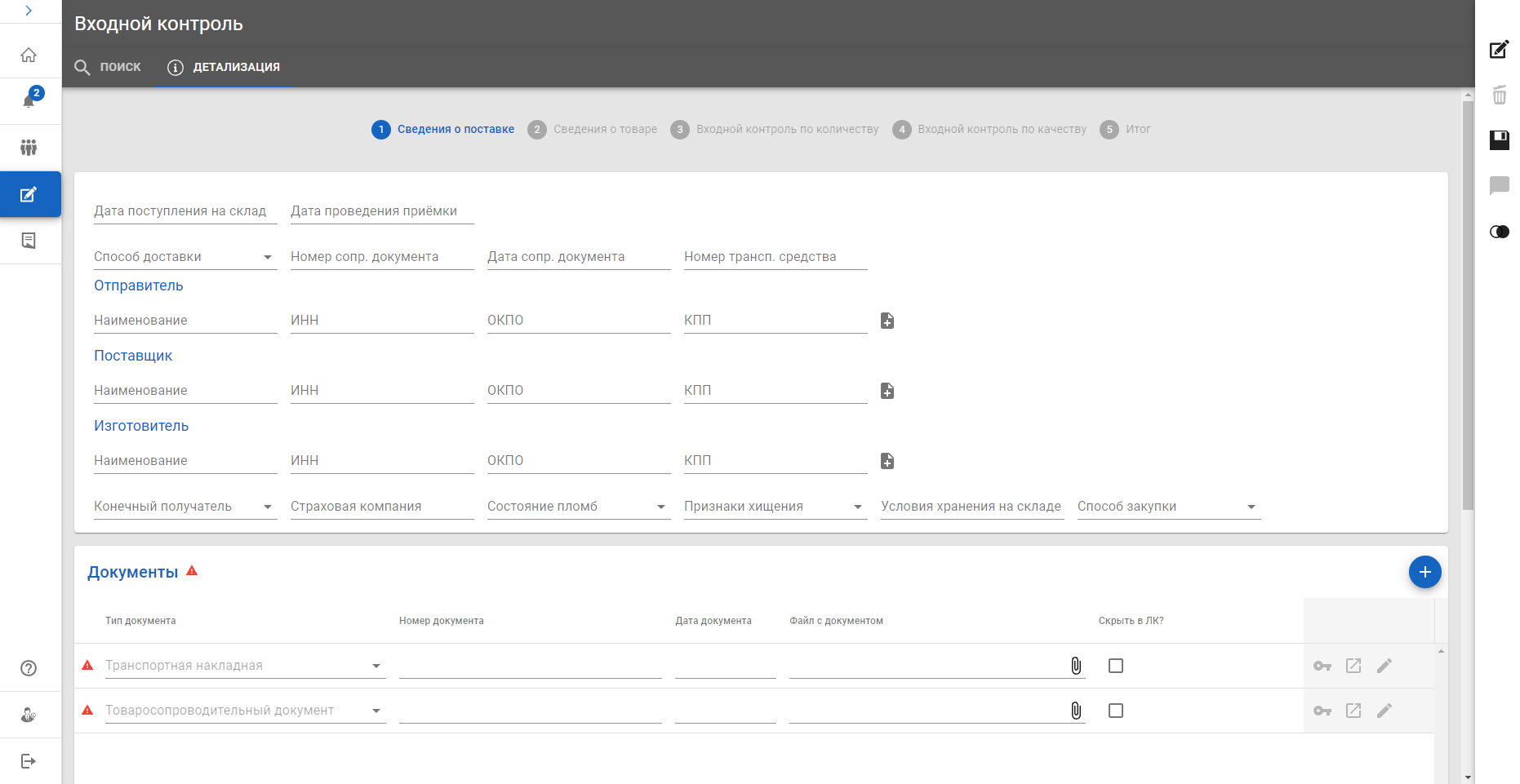


Рисунок 8.11 – Пример реализации режима оформления события

**Режим** для регистрации информации для формирования электронного документа

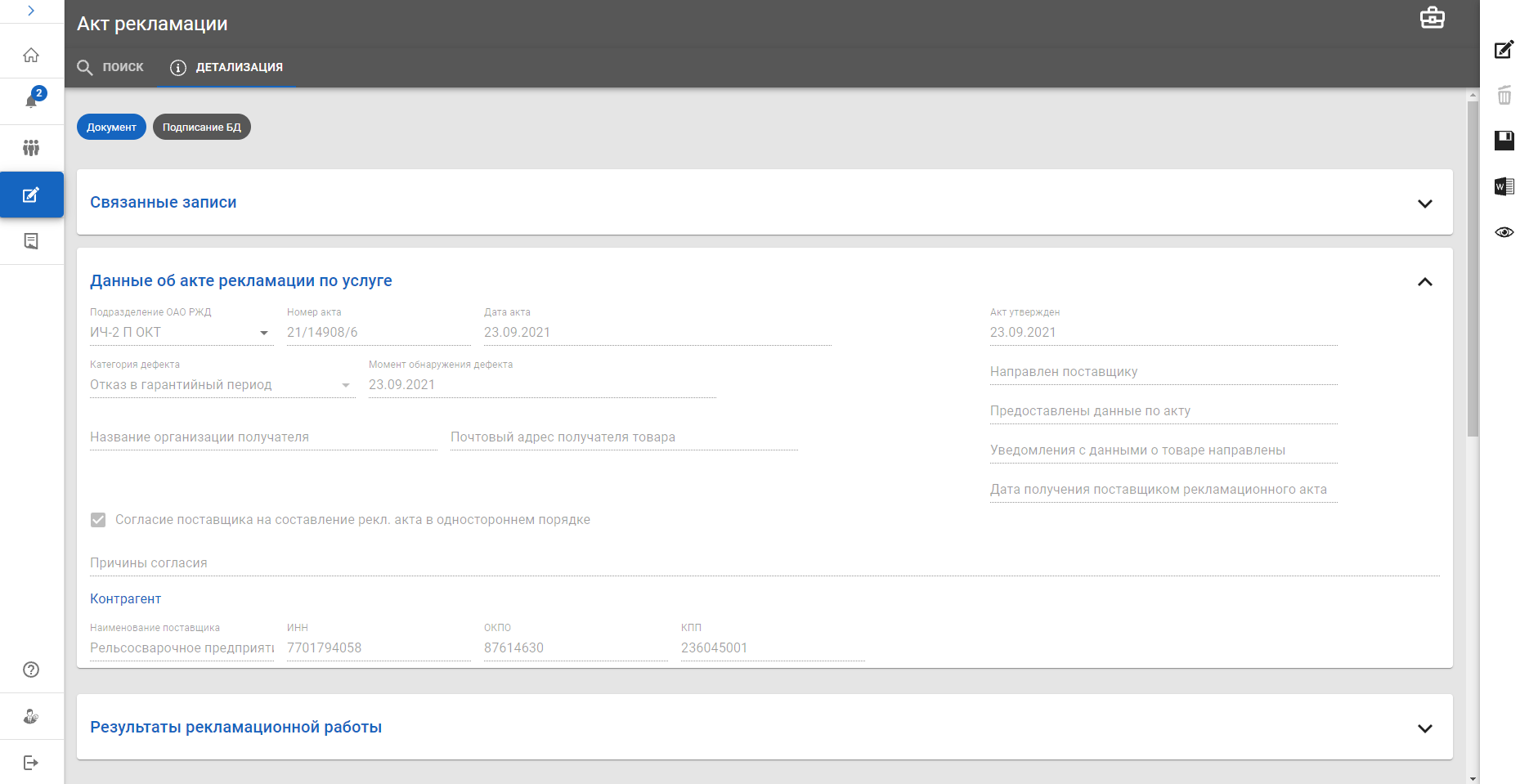


Рисунок 8.12 – Пример реализации режима оформления документа

**Меню** системы с перечнем режимов разных уровней

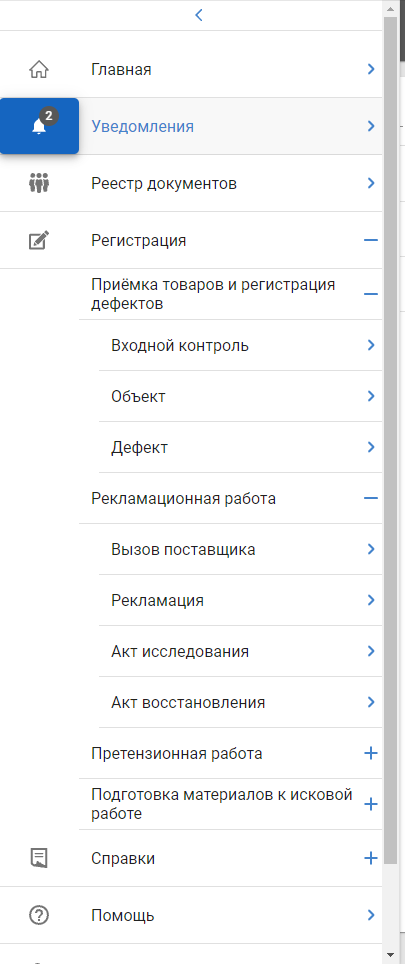


Рисунок 8.13 – Пример реализации меню системы

**Модальное окно Профиль пользователя** для просмотра информации из учетной записи.

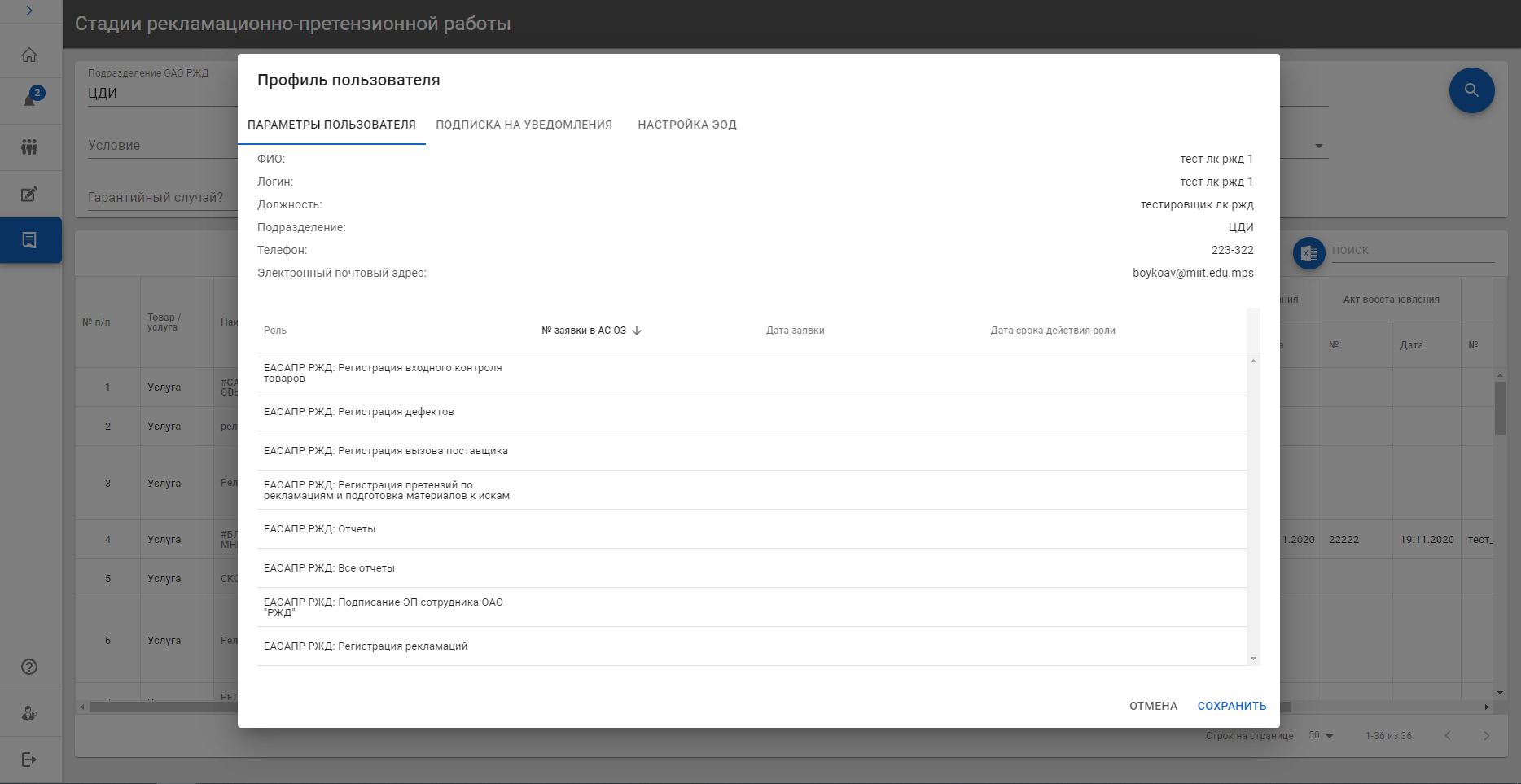


Рисунок 8.13 – Пример реализации информации о пользователе системы

Для разработки ГИП используется инструмент Figma – сервис для разработки интерфейсов и прототипирования.

После регистрации пользователю доступны две рабочие области — графический редактор и менеджер файлов, в котором можно создавать проекты и изменять настройки профиля.

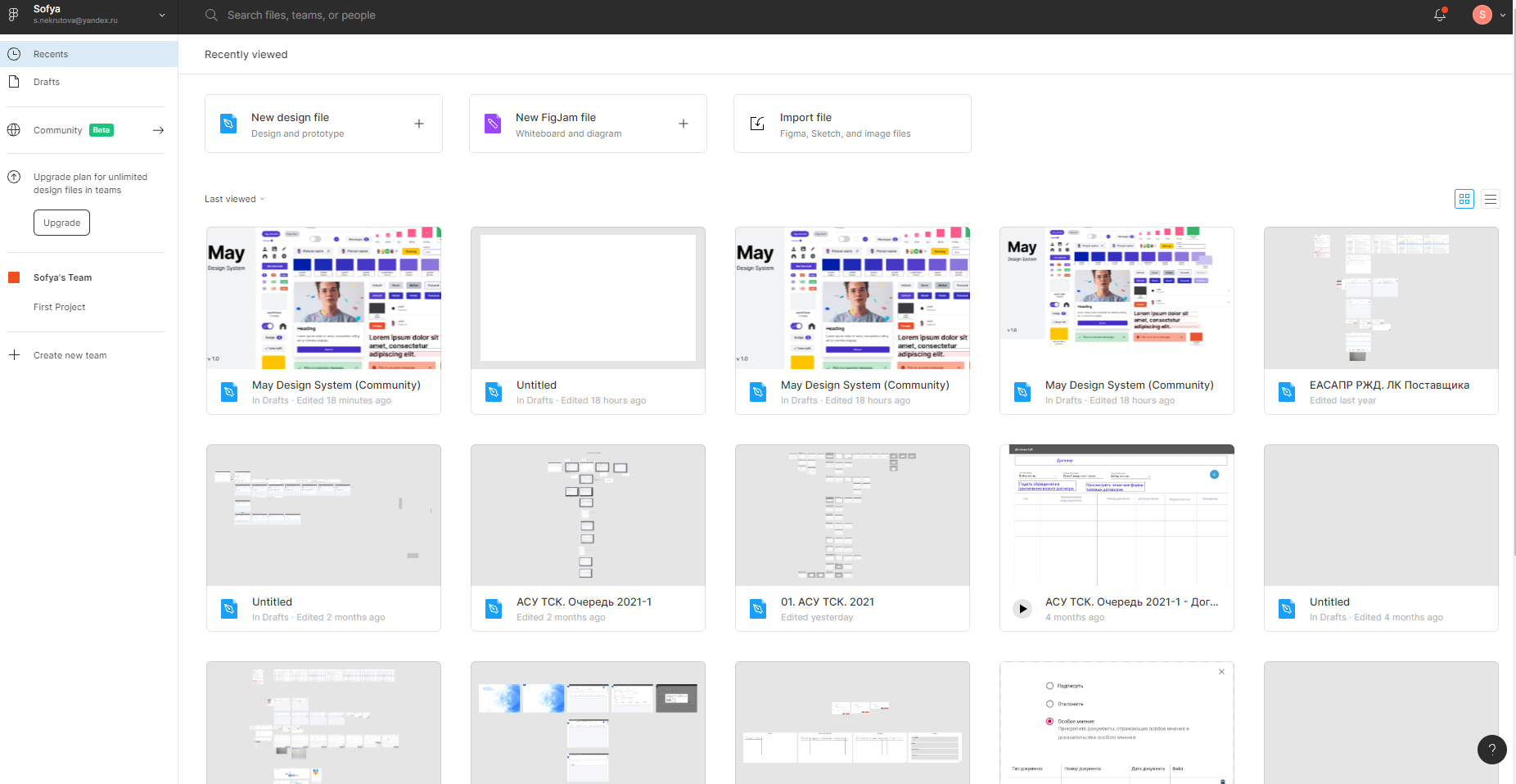


Рисунок 8.14 – Инструмент Figma

Создайте новый файл по нажатии на кнопку «Drafts – Design file».



Рисунок 8.15 – Создание нового файла в Figma

Для выполнения практической работы предлагается воспользоваться подготовленной дизайн системой Мay Design System (<https://www.figma.com/community/file/849353769466663442>).

Для начала использования нажмите кнопку «Duplicare».

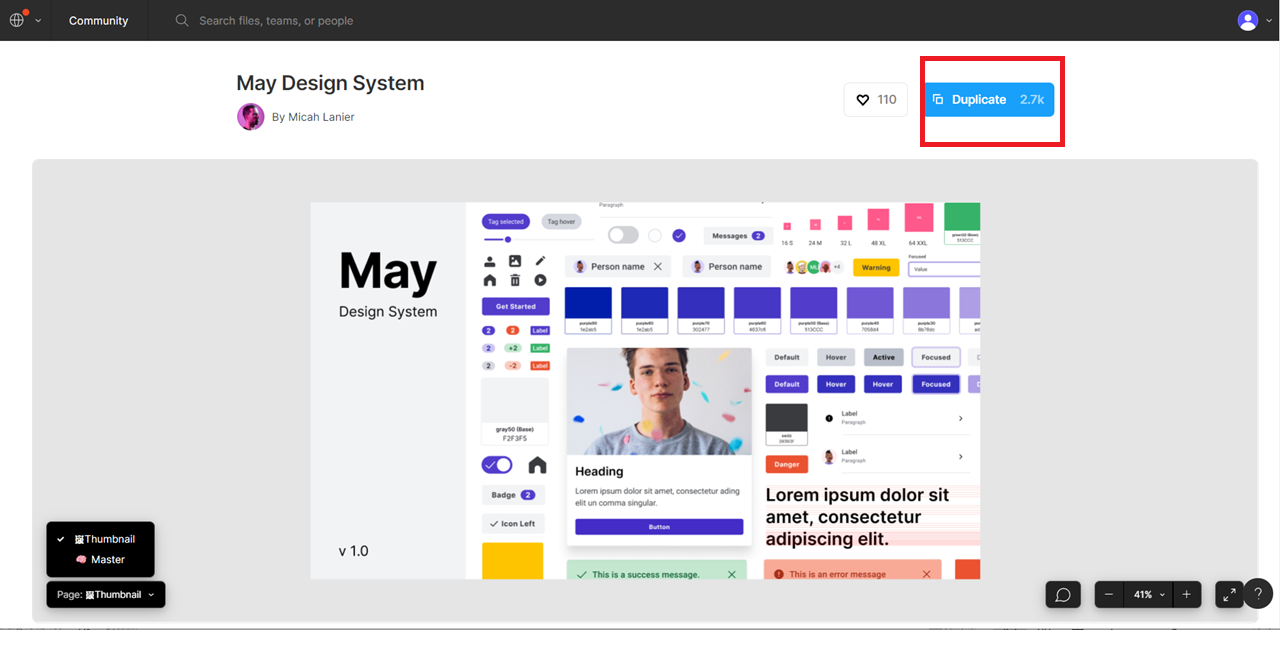


Рисунок 8.16 – Использование дизайн системы в Figma

Для возможности редактирования, копирования компонентов перейти в режим «Master».

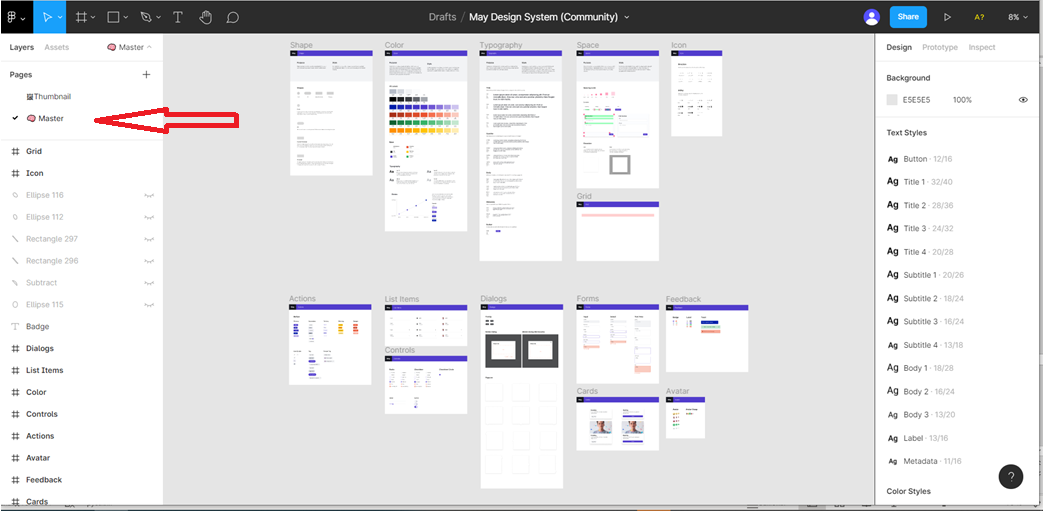


Рисунок 8.17 – Использование дизайн системы в Figma

**Задание №8**

1. Разработать Альбом графических интерфейсов системы

# **Практическое задание № 9. Проектирование ИТ проекта.**

На этапе проектирования архитектуры системы, должен быть произведен обоснованный расчет оборудования (число процессоров, память, диски, производительность дисковой подсистемы, сетевые интерфейсы, производительность каналов передачи данных, сформулированы требования к качеству сети передачи данных) и возможный рост потребности в ресурсах в соответствии с показателями назначения. Расчет оборудования должен производиться с учетом дальнейшего возможного использования на серверах, следующего дополнительного программного обеспечения:

* + агент антивирусной системы;
  + агент системы мониторинга;
  + агент системы резервного копирования.

Требования должны быть указаны раздельно к клиентской части («толстый клиент», мобильное приложение) и серверной в разрезе компонентного состава (серверы приложений, серверы интеграций, SSO, подсистемы мониторинга, серверы БД, серверы отчетности, серверы обновлений, серверы очередей, серверы балансировки, ftp-серверы и т.д.) и включать в себя информацию о:

* + количестве серверов/контейнеров для каждого компонента системы (серверы приложений, серверы интеграций, SSO, подсистемы мониторинга, серверы БД, серверы отчетности, серверы очередей, серверы балансировки, ftp-серверы, серверы обновлений и т.д.);
  + объемах ресурсов под каждый сервер/контейнер (если система размещается в контейнерной конфигурации, то дополнительно указывается общее количество контейнеров при старте системы).

Требования рекомендуется указывать в отдельных таблицах для промышленного (продуктивного), резервного и тестового ПТК. Пример оформления представлен в таблице 9.1.

В ходе проектирования архитектором или опытным программистом создается проектная документация, включающая текстовые описания, диаграммы, модели будущей программы. Для решения данной задачи используется поможет язык UML.

**UML** — является графическим языком для визуализации, описания параметров, конструирования и документирования различных систем (программ в частности). Диаграммы создаются с помощью специальных CASE средств, например Rational Rose (http://www-01.ibm.com/software/rational/) и Enterprise Architect (http://www.sparxsystems.com.au/). На основе технологии UML строится единая информационная модель. Приведенные выше CASE средства способны генерировать код на различных объектно-ориентированных языках, а так же обладают очень полезной функцией реверсивного инжиниринга. (Реверсивный инжиниринг позволяет создать графическую модель из имеющегося программного кода и комментариев к нему.)

Рассмотрим основные типы диаграмм для визуализации модели:

* + Диаграмма вариантов использования (use case diagram);
  + Диаграмма классов (class diagram);
  + Диаграмма состояний (statechart diagram);
  + Диаграмма последовательности (sequence diagram);
  + Диаграмма кооперации (collaboration diagram);
  + Диаграмма компонентов (component diagram);
  + Диаграмма развертывания (deployment diagram).

**Диаграмма вариантов использования (use case diagram)**

Проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых прецедентов. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой.

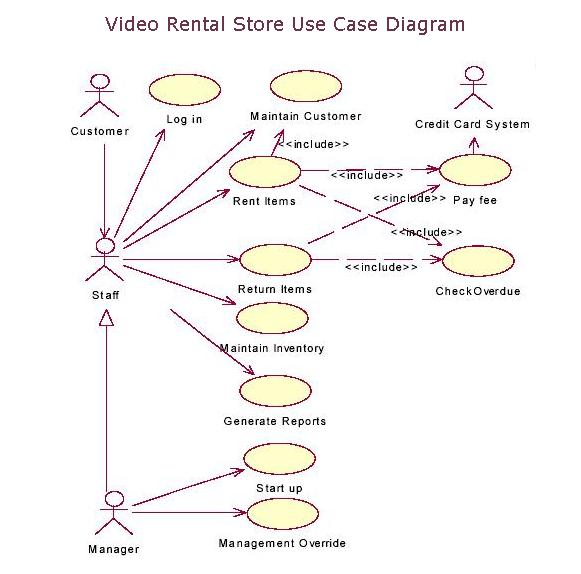


Рисунок 9.1 - Диаграмма вариантов использования

**Диаграмма классов (class diagram)**

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру (поля, методы) и типы отношений (наследование, реализация интерфейсов). На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы. На этом этапе принципиально знание ООП подхода и паттернов проектирования.

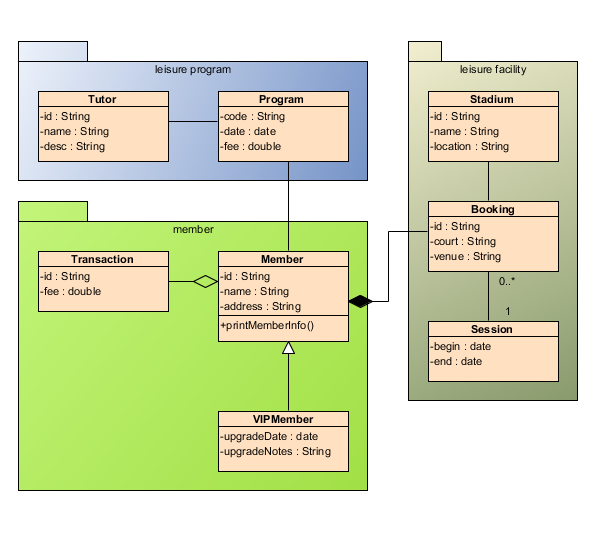


Рисунок 9.2 - Диаграмма классов

**Диаграмма состояний (statechart diagram)**

Главное предназначение этой диаграммы — описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла. Диаграмма состояний представляет динамическое поведение сущностей, на основе спецификации их реакции на восприятие некоторых конкретных событий.

**Диаграмма последовательности (sequence diagram)**

Для моделирования взаимодействия объектов в языке UML используются соответствующие диаграммы взаимодействия. Взаимодействия объектов можно рассматривать во времени, и тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности. Взаимодействующие объекты обмениваются между собой некоторой информацией. При этом информация принимает форму законченных сообщений. Другими словами, хотя сообщение и имеет информационное содержание, оно приобретает дополнительное свойство оказывать направленное влияние на своего получателя.

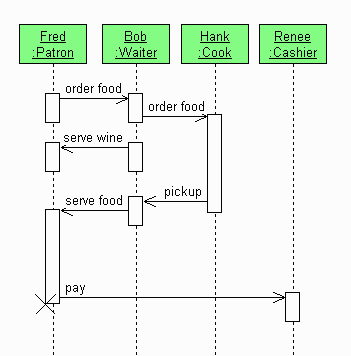


Рисунок 9.3 – Диаграмма последовательности

**Диаграмма кооперации (collaboration diagram)**

На диаграмме кооперации в виде прямоугольников изображаются участвующие во взаимодействии объекты, содержащие имя объекта, его класс и, возможно, значения атрибутов. Как и на диаграмме классов, указываются ассоциации между объектами в виде различных соединительных линий. При этом можно явно указать имена ассоциации и ролей, которые играют объекты в данной ассоциации.

В отличие от диаграммы последовательности, на диаграмме кооперации изображаются только отношения между объектами, играющими определенные роли во взаимодействии.

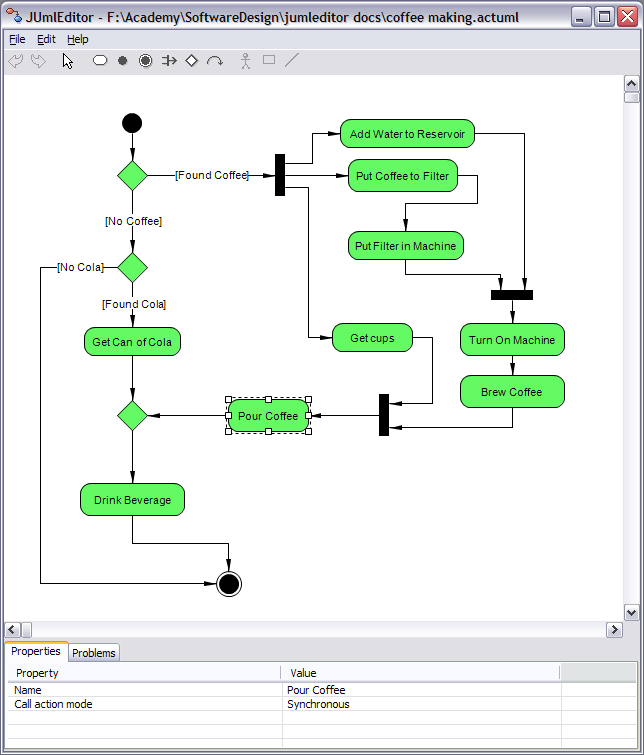


Рисунок 9.4 – Диаграмма кооперации

**Диаграмма компонентов (component diagram)**

Диаграмма компонентов, в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Во многих средах разработки модуль или компонент соответствует файлу. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

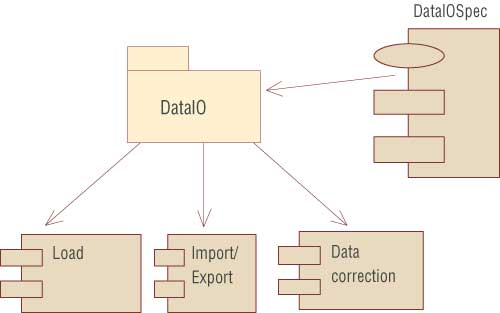


Рисунок 9.5 – Диаграмма компонентов

**Диаграмма развертывания (deployment diagram)**

Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения (runtime). При этом представляются только компоненты-экземпляры программы, являющиеся исполнимыми файлами или динамическими библиотеками. Те компоненты, которые не используются на этапе исполнения, на диаграмме развертывания не показываются.

Диаграмма развертывания содержит графические изображения процессоров, устройств, процессов и связей между ними. В отличие от диаграмм логического представления, диаграмма развертывания является единой для системы в целом, поскольку должна всецело отражать особенности ее реализации. Эта диаграмма, по сути, завершает процесс ООАП для конкретной программной системы и ее разработка, как правило, является последним этапом спецификации модели.

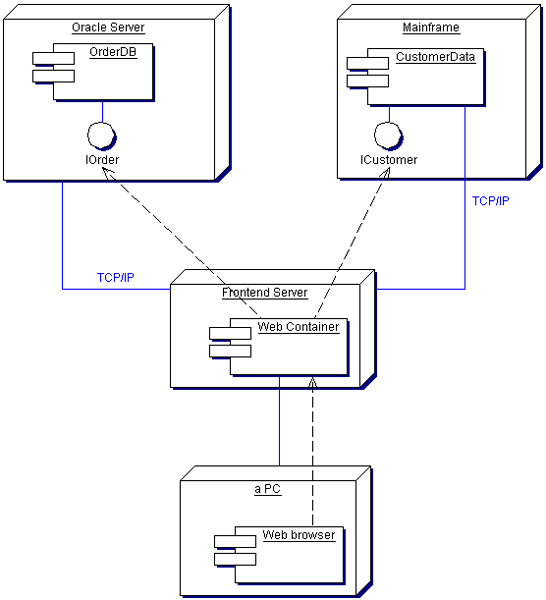


Рисунок 9.6 – Диаграмма развертывания

**Задание №9.1.**

Сформировать таблицу промышленного и тестового ПТК проекта в рамках выполнения практических работ.

**Пример решение №9.1.**

Таблица 9.1 – Требования к техническому обеспечению

| **Сервер** | **Количество** | **Серверные характеристики**  **(на 1 сервер)** | | | **Программная среда** | **Дополнительно** |
| --- | --- | --- | --- | --- | --- | --- |
| **Процессор** | **Оперативная память** | **Дисковое пространство** |
| **Продуктивный контур** | | | | | | |
| Сервер СУБД | 4 | Не менее 24 ядер | Не менее 32 ГБ | Не менее 50ГБ + 1ТБ+ 1ТБ  (системный, диск с данными, диск под резервные копии) | * CentOS 7.8 и выше (x64); * PostgreSQL 12 и выше (x64); * PCSD; * Corosync. | * 1 адаптер локальной сети 1 ГБ/c |
| Сервер приложений | 10 | Не менее 14 ядер | Не менее 20 ГБ | Не менее 50ГБ + 100ГБ | * CentOS 7.8 и выше (x64); * keepalived, docker-ce 19.03, docker-compose 1.25   . | * 1 адаптер локальной сети 1 ГБ/c |
| Шлюз файлового хранилища | 3 | 4 ядра | 4 ГБ | 50ГБ + 100ГБ | * CentOS 7.8 и выше (x64); * keepalived, docker-ce 19.03, docker-compose 1.25 | * 1 адаптер локальной сети 1 ГБ/c |
| Узел файлового хранилища | 5 | 4 ядра | 8 ГБ | 50ГБ + 6ТБ | * CentOS 7.8 и выше (x64); * keepalived, docker-ce 19.03, docker-compose 1.25 | * 1 адаптер локальной сети 1 ГБ/c |
| Узел хранилища конфигураций | 3 | 4 ядра | 8 ГБ | 50ГБ + 50ГБ + 30ГБ | * CentOS 7.8 и выше (x64); * keepalived, docker-ce 19.03, docker-compose 1.25 | * 1 адаптер локальной сети 1 ГБ/c |
| **Тестовый контур** | | | | | | |
| Сервер СУБД | 2 | Не менее 4 ядер | Не менее 8 ГБ | Не менее 50ГБ + 100ГБ + 100ГБ  (системный, диск с данными, диск под резервные копии) | * CentOS 7.8 и выше (x64); * PostgreSQL 12 и выше (x64); * PCSD; * Corosync. | * 1 адаптер локальной сети 1 ГБ/c |
| Сервер приложений | 5 | Не менее  4 ядер | Не менее 8 ГБ | Не менее 50ГБ + 100ГБ | * CentOS 7.8 и выше (x64). * keepalived, docker-ce 19.03, docker-compose 1.25 | * 1 адаптер локальной сети 1 ГБ/c |
| Шлюз файлового хранилища | 1 | 4 ядра | 4 ГБ | 50ГБ + 100ГБ | * CentOS 7.8 и выше (x64); * keepalived, docker-ce 19.03, docker-compose 1.25 | * 1 адаптер локальной сети 1 ГБ/c |
| Узел файлового хранилища | 2 | 4 ядра | 8 ГБ | 50ГБ + 100ГБ | * CentOS 7.8 и выше (x64); * keepalived, docker-ce 19.03, docker-compose 1.25 | * 1 адаптер локальной сети 1 ГБ/c |
| Узел хранилища конфигураций | 3 | 4 ядра | 8 ГБ | 50ГБ + 50ГБ + 25ГБ | * CentOS 7.8 и выше (x64); * keepalived, docker-ce 19.03, docker-compose 1.25 | * 1 адаптер локальной сети 1 ГБ/c |

**Задание №9.2.**

Разработать одну из рассмотренных диаграмм для проекта в рамках выполнения практических работ.

# **Практическое задание № 10. Разработка Технического задания.**

При реализации ИТ проекта - проектировании и создании программного обеспечения важным этапом является разработка Технического задания на Программу для ЭВМ.

Основная задача этого этапа – описать подробно функции и возможности будущего программного обеспечения и согласовать их между Заказчиком и Исполнителем. Техническое задание необходимо для снижения возможных рисков при выполнении работ.

Разработка программного обеспечения должна осуществляться в соответствии с Техническим заданием и в сроки, указанные в Техническом задании.

Техническое задание на программное обеспечение – это основной документ, предъявляющий требования к создаваемому программному обеспечению и устанавливающий порядок, в соответствии с которым будет проводиться разработка и ее прием при вводе в эксплуатацию.

На этапе разработки, обсуждения и согласования технического задания решается 99% всех возможных разногласий, в ходе реализации, могут возникать дополнительные нюансы, но как правило, они не имеют принципиального характера и быстро решаются.

Техническое задание дает четкие критерии оценки готового продукта (сайта, сервиса, приложения). Правильно сформулированное техническое задание дает исполнителю понимание того, что должно быть на выходе, а заказчику — как оценивать результат. Чем меньше в техническом задании абстрактных формулировок, которые можно понять по-разному, тем лучше.

Техническое задание поможет точно оценить затраты и возможности.

Техническое задание поможет организовать работу с несколькими подрядчиками или безболезненно сменить исполнителя.

Итак, Техническое задание должно быть в любом ИТ проекте и должно содержать формализованное отражение общей концепции, а также детализацию (как можно более точную) всех требований и пожеланий заказчика.

Кроме того, проработанное техническое задание — это еще и минимизация трудозатрат со стороны разработчика. Имея четкое задание, исполнитель пойдет к цели кратчайшим путем.

Наличие проработанного технического задания является критически важным фактором для большинства проектов.

Техническое задание на Программу для ЭВМ, ПО, АС и т.д. разрабатывается в соответствии со следующими нормативными документами:

* + ГОСТ 34.602-89. Комплекс стандартов на автоматизированные системы «Техническое задание на создание автоматизированной системы»;
  + Стандарт ОАО «РЖД» СТО РЖД 04.001.1–2021 «Автоматизированные системы и программные средства. Требования к составу, содержанию и оформлению документов при создании автоматизированных систем и программных средств ОАО «РЖД»;

Оформление Технического задания, рабочей документации должно осуществляться в соответствии со следующими нормативными документами:

* + ГОСТ 7.32-2017. Межгосударственный стандарт. «Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления».
  + Основными требованиями по оформлению являются:
  + Межстрочный интервал – 1,1 пт;
  + Тип шрифта – Times New Roman;
  + Применение полужирного шрифта для заголовков;
  + Размеры полей: левое — 30 мм, правое — 15 мм, верхнее и нижнее — 20 мм;
  + Абзацный отступ - 1,25 см;
  + Заголовки располагаются в середине строки без точки в конце, прописными буквами, не подчеркивая;
  + Каждый раздел основной части начинается с новой страницы;
  + и другие требования к построению отчета, нумерации страниц, нумерации разделов и т.д.

Техническое задание на разработку программного обеспечения должны содержаться следующие разделы:

* + общие сведения;
  + цели и назначение создания Программы для ЭВМ;
  + характеристика объектов автоматизации;
  + требования к Программе для ЭВМ;
  + состав и содержание работ по созданию Программы для ЭВМ;
  + порядок контроля и приемки Программы для ЭВМ;
  + требования к составу и содержанию работ по подготовке объекта автоматизации к вводу Программы для ЭВМ в действие;
  + требования к документированию;
  + источники разработки.

Разделы «**Общие сведения», «Цели и назначение создания Программы для ЭВМ», «Характеристика объектов автоматизации»** должны предоставлять информацию общего характера о проекте, из которой должна быть понятна концепция разрабатываемого программного обеспечения.

Примеры заполнения разделов приведены в таблицах 10.1 - 10.3.

Таблица 10.1 - Раздел 1 «Общие сведения»

| **Наименование подраздела** | **Пример заполнения** |
| --- | --- |
| 1.1 Полное наименование системы и ее условное обозначение | Полное наименование системы: Единая автоматизированная система рекламационно-претензионной работы ОАО «РЖД»  Сокращенное наименование системы: ЕАСАПР РЖД |
| 1.2. Шифр темы или шифр (номер) договора | Основанием для проведения работ по проектированию автоматизированной системы являются практические работы по дисциплине «Цифровые технологии» |
| 1.3 Наименование предприятий разработчика и заказчика системы | Заказчик – Предполагаемый заказчик или РУТ (МИИТ)  Разработчик – Название команды |
| 1.4 Перечень документов, на основании которых создается система, кем и когда утверждены эти документы | Проектирование Автоматизированной системы «ЕАСАПР РЖД» осуществляется на основании следующих нормативных документов. |
| 1.5 Плановые сроки начала и окончания работы по созданию системы | Начало работы: 31.03.2020 г.  Окончание работы: 30.10.2020 г. |
| 1.6 Сведения об источниках и порядке финансирования работ | Не указываются |
| 1.7 Порядок оформления и предъявления заказчику результатов работ по созданию системы | Результатом работ является утвержденный проект автоматизированной системы ЕАСАПР РЖД |

Таблица 10.2 - Раздел 2 «Цели и назначение создания Программы для ЭВМ»

| **Наименование подраздела** | **Пример заполнения** |
| --- | --- |
| 2.1 Назначение системы | ЕАСАПР РЖД. ЛК Поставщик предназначена для использования внешними поставщиками товаров и услуг через корпоративный веб-портал ОАО «РЖД» (при наличии соглашения об ЭОД), а также структурными подразделениями ОАО «РЖД», осуществляющими рекламационную и претензионную работу. |
| 2.2 Цели создания системы | Целью создания ЕАСАПР РЖД. ЛК Поставщик является повышение эффективности рекламационно-претензионной работы за счет перехода на взаимодействие ОАО «РЖД» с поставщиками товаров и услуг в электронном виде. |

Таблица 10.3 - Раздел 3 «Характеристика объектов автоматизации»

| **Наименование подраздела** | **Пример заполнения** |
| --- | --- |
| 3 Характеристика объектов автоматизации | В настоящее время взаимодействие между поставщиками товаров и услуг и ОАО «РЖД» осуществляется посредством обмена документами на бумажном носителе. Внутренний документооборот процесса рекламацонно-претензионной работы структурных подразделений ОАО «РЖД» также осуществляется с использованием бумажного документооборота.  В рамках работ по разработке программы для ЭВМ ЕАСАПР РЖД. ЛК Поставщик в целях перехода на электронный документооборот с поставщиками товаров и услуг реализуется возможность обмена информацией между сотрудниками ОАО «РЖД» и внешними поставщиками товаров и услуг в рамках ведения рекламационно-претензионной работы в электронном виде с применением простой электронной подписи при подписании рекламационно-претензионных документов. |

Раздел «**Требования к программе для ЭВМ»** является основным по содержанию и описанию функций будущего программного обеспечения. В нем должны быть описаны все требования и функциональные возможности. На основании указанной информации будут выполняться работы как со стороны Исполнителя, так и со стороны сопровождающей организации, в ведении которой находится обеспечение работоспособности ПТК, подключение пользователей, мониторинг состояния Программы для ЭВМ.

Первым подразделом является **«4.1** **Требования к системе в целом».** Он состоит из множества блоков. Однако, в рамках выполнения практических заданий, будут рассмотрены следующие:

* + 4.1.1.1 Перечень модулей, их назначение и основные характеристики;
  + 4.1.1.3 Требования к характеристикам взаимосвязей создаваемой системы со смежными системами;
  + 4.1.13 Требования по стандартизации и унификации.

В пункте «**4.1.1.1 Перечень модулей, их назначение и основные характеристики**» указывается перечень функциональных подсистем, модулей, а также приводится соответствующее перечисление функциональности системы в соответствии с договором. Если система является многоуровневой (имеет линейный, региональный, сетевой уровни и т.п.) следует описывать её иерархию.

Для обмена и синхронизации данных системы с внешними и смежными системами должен быть реализован программный интерфейс (API). Программный интерфейс должен предоставлять механизмы запуска открытых методов программных систем. Общие требования к API включают:

* + возможность расширения;
  + описание возвращаемых кодов выполнения запросов;
  + стандартизированный формат запросов и ответов;
  + аутентификацию участников взаимодействия и ограничение доступа на пользование интерфейсами и запуск методов.

Параметры подключения к смежным системам должны быть вынесены в отдельные конфигурационные файлы.

Входящие в состав системы сервисы в процессе функционирования должны обмениваться информацией на основе открытых форматов обмена данными, используя для этого входящие в их состав модули информационного взаимодействия.

В пункте «**4.1.1.3 Требования к характеристикам взаимосвязей создаваемой системы со смежными системами**» необходимо привести перечень смежных и внешних систем, с которыми должна взаимодействовать система и отразить на схеме информационных потоков.

Требования к содержательной части информационного обмена следует излагать на основании архитектурной схемы системы. Рекомендуемый формат схемы информационных потоков представлен на рисунке 10.1, 10.2.

Все взаимодействия на схеме рекомендуется пронумеровать и описать их после схемы в табличном виде со следующими столбцами: «№ связи на схеме», «Источник», «Получатель», «Состав данных», «АС-инициатор взаимодействия», «Способ взаимодействия и протокол», «Периодичность (временной регламент)», «Ориентировочный объем передаваемых данных за итерацию».

При необходимости (наличие внешних систем или пользователей вне СПД ОАО «РЖД»), отображаются СЗИ (средства защиты информации).

В пункте **«4.1.13 Требования по стандартизации и унификации»** определяются требования стандартизованных методов, моделей, алгоритмов, проектных решений, программных средств, форм документов, указывается, какая нормативно-справочная информация и из каких источников должна использоваться в системе. Приводятся требования по использованию общеотраслевых классификаторов и унифицированных документов, с которыми должна работать система. Указывается перечень используемых справочников АС ЦНСИ. В случае отсутствия детализированной информации по справочникам на данном этапе, указывается ссылка на документ, в котором это должно быть отражено (например, Описании информационной технологии и/или РОС).

Примеры заполнения разделов приведены в таблице 10.4.

Примеры формирования схемы информационных потоков приведены на рисунках 10.1, 10.2.

Таблица 10.4 - Раздел 4 «Требования к Программе для ЭВМ»

Подраздел 4.1 «Требования к системе в целом»

| **Наименование подраздела** | **Пример заполнения** |
| --- | --- |
| 4.1.1 Требования к структуре и функционированию системы | |
| 4.1.1.1 Перечень модулей, их назначение и основные характеристики | В составе ЕАСАПР РЖД. ЛК Поставщик должны быть разработаны следующие программные модули:   1. Модуль применения электронной подписи (ЭП) для согласования и утверждения рекламационно-претензионных документов:  * подписание документа ЭП с учетом маршрутизации * информирование пользователей о необходимости подписания документа ЭП * фиксация отказа от подписания документа с указанием причины |
| 4.1.1.2 Требования к способам и средствам связи для информационного обмена между компонентами системы | Требования не предъявляются |
| 4.1.1.3 Требования к характеристикам взаимосвязей создаваемой системы со смежными системами | В рамках работ должно быть реализовано взаимодействие системы ЕАСАПР РЖД. ЛК Поставщик с системой КВП в части трансляции интерфейсов ЕАСАПР РЖД. ЛК Поставщик в рамках Модуля взаимодействия с внешними поставщиками.  Общая схема взаимодействия ЕАСАПР РЖД с внешними системами представлена на рисунке.  Схема взаимодействия (см. слайд 14) |
| 4.1.1.4 Требования к режимам функционирования программного обеспечения | Требования не предъявляются |
| 4.1.1.5 Требования по диагностированию системы | Требования не предъявляются |
| 4.1.1.6 Перспективы развития, модернизации системы | Описание следующего этапа развития системы |
| 4.1.2 Требования к численности и квалификации персонала системы и режиму его работы | См. ГОСТ 34.602-89 и СТО РЖД №951р от 29.04.2021  В рамках проекта данные требования не предъявляются. |
| 4.1.3 Показатели назначения |
| 4.1.4 Требования к надежности |
| 4.1.5 Требования безопасности |
| 4.1.6 Требования к эргономике и технической эстетике |
| 4.1.7 Требования к транспортабельности для подвижных АС |
| 4.1.8 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы |
| 4.1.9 Требования к защите информации от несанкционированного доступа |
| 4.1.10 Требования по сохранности информации при авариях |
| 4.1.11 Требования к защите от влияния внешних воздействий |
| 4.1.12 Требования к патентной чистоте |
| 4.1.13 Требования по стандартизации и унификации | Программа для ЭВМ «ЕАСАПР РЖД. ЛК Поставщик» должна использовать следующую справочную информацию:  Статусы подписи:   * Направлено на подписание; * Подписано; * Отклонено;   Подписано с особым мнением. |

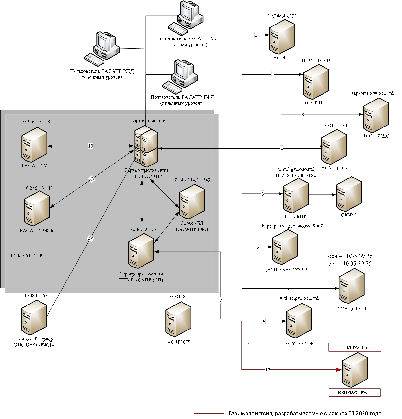


Рисунок 10.1 – Пример изображения схемы информационных потоков



Рисунок 10.2 – Пример изображения схемы информационных потоков

**Задание № 10**

Сформировать Техническое задание оформленное в соответствии с требованиями ГОСТ, содержащее следующие разделы:

* + Титульный лист (см. ГОСТ 34.602-89 или СТО РЖД 04.001.1–2021);
  + Содержание;
  + 1 Общие сведения;
  + 2 Назначение и цели создания системы;
  + 3 Характеристика объектов автоматизации;
  + 4.1 Требования к системе в целом;
  + Схема взаимодействий со смежными АС в разделе 4.1.1.3;

# **Практическое задание № 11. Разработка Технического задания.**

Продолжаем рассматривать формирование Раздела 4 «Требования к системе в целом» Технического задания на программу для ЭВМ.

Вторым подразделом является **«4.2 Требования к функциям (задачам), выполняемым системой».** Это главный подраздел технического задания, в котором приводится описание функций системы в соответствии с договором.

В данный раздел включают пункты, подробно описывающие функциональные возможности разрабатываемого программного обеспечения в целях согласования с заказчиком и передачи требований в разработку.

В качестве сопровождения описываемых функций к Техническому заданию делаются приложения к Техническому заданию, в которых приводятся примеры визуального отображения режимов, шаблонов формируемых документов, алгоритмы выполнения тех или иных функций, выходные формы, формируемые системой, шаблоны справочной, отчетной и аналитической информации. В тексте описываемых функций указывается ссылка на номер Приложения к Техническому заданию с указанием информации, которая в нем будет отражена.

Пример заполнения подраздела «4.2 Требования к функциям (задачам), выполняемым системой» приведен в таблице 11.1.

Таблица 11.1 - Раздел 4 «Требования к Программе для ЭВМ»

Подраздел 4.2 «Требования к Требования к функциям (задачам), выполняемым системой»

| **Наименование подраздела** | **Пример заполнения** |
| --- | --- |
| 4.2 Требования к функциям (задачам), выполняемым системой | В подразделе указывается перечень функций и описание их реализации.  **Подписание документа ЭП с учетом маршрутизации**  Должен быть реализован механизм подписания простой ЭП следующих документов:   * акт об отсутствии документов при приемке товара; * акт отбора проб; * акт о несоответствии.   Для регистрации маршрута подписания ЭП документа в составе интерфейсов ЕАСАПР РЖД должен быть предусмотрен блок «Подписания ЭП».  Блок «Подписание ЭП» должен включать в себя таблицу для указания подписантов документа в составе следующих граф:   * + ФИО;   + Структурное подразделение;   + Должность;   + Статус подписи;   + Дата и время подписи;   + Действие.   Графа «ФИО» должна заполняться с использованием справочника системы администрирования ЕАСАПР РЖД по всем зарегистрированным в системе пользователям с активной заявкой. |

**Задание №11**

Разработать разделы технического задания оформленного в соответствии с требованиями ГОСТ:

* 4.2 «Требования к функциям (задачам), выполняемым системой»;
* Приложения к ТЗ (альбом интерфейсов системы, альбом выходных форм, альбом справочных, отчетных и аналитических форм);

# **Практическое задание № 12. Разработка Технического задания.**

Завершаем рассматривать формирование Раздела 4 «Требования к системе в целом» Технического задания на программу для ЭВМ.

Третьим подразделом является **«4.3 Требования к видам обеспечения».** Данный подраздел содержит восемь пунктов, которые отражают техническую сторону реализуемого программного обеспечения и описывает требования для его создания и дальнейшего успешного функционирования.

Пунктами подраздела 4.3, на которые стоит обратить внимание, являются:

* 4.3.4 Требования к системному программному обеспечению;
* 4.3.5 Требования к техническому обеспечению;
* 4.3.8 Требования к методическому обеспечению

В пункте «**4.3.4 Требования к системному программному обеспечению**» для каждого программно-технического комплекса приводится перечень системных программных средств, которые необходимы для обеспечения эксплуатации системы: операционные системы, СУБД, программные платформы серверов приложений, сервисная шина предприятия (ESB), иные сервисные программы.

В случае если система будет размещена на существующих технических средствах, указывают, что серверное оборудование не требует (или требует) дополнительного лицензионного ПО. Указывают ссылку на документ, где описаны текущие характеристики сервера (серверов).

Требования должны быть указаны раздельно к клиентской части («толстый клиент», мобильное приложение) и серверной в разрезе компонентного состава (например, серверы приложений, серверы интеграций, SSO, подсистемы мониторинга, серверы БД, серверы отчетности, серверы обновлений, серверы очередей, серверы балансировки, ftp-серверы и т.д.).

Требования к тестовому и промышленному полигону указываются отдельно.

В пункте «**4.3.5 Требования к техническому обеспечению**» для каждого программно-технического комплекса приводится перечень и технические характеристики оборудования, необходимость его резервирования, включая горячее резервирование, а также иные специальные требования по развертыванию системы. Требования к серверному техническому обеспечению указываются в формате «не более», к клиентскому в формате «не менее».

В случае если система будет размещена на существующих технических средствах, то указывают, что серверное оборудование не требует (или требует) модернизации.

В пункте «**4.3.8 Требования к методическому обеспечению**» приводят требования к составу нормативно-технической документации системы (перечень применяемых при ее функционировании стандартов, нормативов, методик и т. п.). Обратите внимание на то, что в данном разделе приводится перечень методических материалов, используемый при функционировании системы, а не при её создании. В случае отсутствия таковых указывается: требования не предъявляются.

Для перечисленных документов необходимо привести их подробные реквизиты: децимальный номер, кем и когда утверждено, место и год издания, издательство (для типографских изданий), URL (при размещении документа на Интернет-ресурсе) и т.д.

Таблица 12.1 - Раздел 4 «Требования к Программе для ЭВМ»

Подраздел 4.3 «Требования к видам обеспечения»

| **Наименование подраздела** | **Пример заполнения** |
| --- | --- |
| 4.3.1 Требования к математическому обеспечению | См. ГОСТ 34.602-89 и СТО РЖД №951р от 29.04.2021  В рамках проекта данные требования не предъявляются. |
| 4.3.2 Требования к информационному обеспечению | См. ГОСТ 34.602-89 и СТО РЖД №951р от 29.04.2021  В рамках проекта данные требования не предъявляются. |
| 4.3.3 Требования к лингвистическому обеспечению | См. ГОСТ 34.602-89 и СТО РЖД №951р от 29.04.2021  В рамках проекта данные требования не предъявляются. |
| 4.3.4 Требования к программному обеспечению | Сервер БД: \_\_\_\_\_.  Операционная система \_\_\_\_ или более новая версия.  СУБД \_\_\_\_ или более новая версия и др.  Для разработки мобильного приложения (МП):. Система должна обеспечивать работу на следующих ОС:  Android версии 4 и выше.  Windows 7 и выше. |
| 4.3.5 Требования к техническому обеспечению | Система должна обеспечивать возможность организации серверов в среде виртуализации Заказчика. В состав технической архитектуры системы должны входить:  выделенный сервер или набор серверов на стороне Заказчика для обеспечения работы промышленной версии АС;  выделенный сервер или набор серверов Заказчика для обеспечения работы тестовой версии АС***.***  См. таблицу 9.1 |
| 4.3.6 Требования к метрологическому обеспечению | См. ГОСТ 34.602-89 и СТО РЖД №951р от 29.04.2021  В рамках проекта данные требования не предъявляются. |
| 4.3.7 Требования к организационному обеспечению | См. ГОСТ 34.602-89 и СТО РЖД №951р от 29.04.2021  В рамках проекта данные требования не предъявляются. |
| 4.3.8 Требования к методическому обеспечению |  |

Пример заполнения раздела «4.3.5. Требования к техническому обеспечению приведен в таблице 9.1, в рамках выполнения практической работы №9.

Раздел 5 **«Состав и содержание работ»** должен включать таблицу (пример приведен в таблице 12.2), содержащую перечень необходимых к выполнению работ в соответствии с договором, сроки их выполнения, перечень организаций - исполнителей работ, перечень документов, предъявляемых по окончании соответствующих стадий и этапов работ. В тексте раздела указывают полигон внедрения.

Таблица 122 - Пример заполнения таблицы «Состав и содержание работ»

| **Стадии** | **Этапы работ** | **Сроки выполнения** | **Организация-исполнитель** | **Документы, предъявляемые по окончании стадии** |
| --- | --- | --- | --- | --- |
| Техническое задание и проектирование | Разработка, согласование и утверждение технического задания | 02.2022 | Исполнитель | * Техническое задание (ТЗ); |
| Разработка, согласование и утверждение описания информационной технологии | 10.2022 | Исполнитель | * Описание информационной технологии (ПТ) |
| Рабочая документация | Разработка рабочей документации | 10.2022 | Исполнитель | * Руководство администратора базы данных (И5); * Руководство администратора системы (И6); * Руководство по инсталляции (И7); * Руководство пользователя (И3); * Руководство по организации сопровождения (92); * Программа и методика испытаний (ПМИ); * Ведомость РД (РД). |
| Ввод в действие | Подготовка и ввод в опытную эксплуатацию | 10.2022 | Исполнитель | * Протокол испытаний; * Акт приемки в опытную эксплуатацию. |
| Проведение опытной эксплуатации | 11.2022 | Исполнитель, Подразделения полигона внедрения | * Отчет о проведении опытной эксплуатации; |
| Подготовка и ввод в постоянную эксплуатацию | 11.2022 | Исполнитель | * Протокол испытаний; * Акт приемки в постоянную эксплуатацию. |

Раздел **6 «Порядок контроля и приемки системы»** должен содержать определенные договором требования к порядку организации контроля и приемки системы, а также соответствовать положениям СТО РЖД 04.001.2.

Приемка системы производится в соответствии с требованиями, изложенными в программе и методике испытаний. Перечень организаций и структурных подразделений ОАО «РЖД», участвующих в предварительных и приёмочных испытаниях, место и сроки их проведения должны быть отражены в Программе и методике испытаний.

Раздел **7 «Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие»** должен содержать требования к перечню основных мероприятий и их исполнителей, которые следует выполнить при подготовке объекта автоматизации к вводу системы в действие.

В разделе **8 «Требования к документированию»** приводятся отдельные требования, к содержанию и оформлению технических и организационных документов или ссылки на стандартизирующие сборники, распоряжения ОАО «РЖД», национальные и государственные ГОСТы.

В разделе **9 «Источники разработки»** должны быть перечислены документы и информационные материалы (технико-экономическое обоснование, отчеты о законченных научно-исследовательских работах, информационные материалы на отечественные, зарубежные системы-аналоги и др.), на основании которых разрабатывалось ТЗ на систему и которые должны быть использованы при разработке ПО. Для перечисленных документов необходимо привести их подробные реквизиты: децимальный номер, кем и когда утверждено, место и год издания, издательство (для типографских изданий), URL (при размещении документа на Интернет-ресурсе) и т.д.

Раздел **«Обозначения и сокращения»** должен содержать все примененные в тексте сокращения, за исключением тех, что предусмотрены ГОСТ или являются общеупотребительными. Рекомендуется оформлять перечень сокращений в табличной форме со скрытой сеткой, как показано в таблице 12.3. Сокращения должны располагаться по алфавиту.

Таблица 12.3 - Пример представления перечня сокращений

| **CPU** | – | (Англ. central processing unit) Центральный процессор (ЦП; также центральное процессорное устройство — ЦПУ) |
| --- | --- | --- |
| **JSON** | – | (Англ. JavaScript Object Notation) - текстовый формат обмена данными, основанный на JavaScript. |

**Задание №12.**

Разработать разделы технического задания оформленного в соответствии с требованиями ГОСТ:

* 4.3 «Требования к видам обеспечения»;
* Перечень сокращений;
* Источники разработки;

# **Практическое задание № 13. Разработка Рабочих документов.**

Следующим этапом при проектировании и создании программного обеспечения является разработка Рабочей документации.

Основная задача рабочей документации – дать полное представление о том, как устроено программное обеспечение, из чего оно состоит и как функционирует.

Документация входит в состав проекта по созданию, внедрению, сопровождению, модернизации и ликвидации ИС на протяжении полного жизненного цикла этой̆ ИС.

Пользовательская документация ПО объясняет пользователям, как они должны действовать, чтобы применить разрабатываемое ПО. Она необходима, если ПО предполагает какое-либо взаимодействие с пользователями. К такой̆ документации относятся документы, которыми должен руководствоваться пользователь при инсталляции ПО, при применении ПО для решения своих задач и при управлении ПО (например, когда разрабатываемое ПО будет взаимодействовать с другими системами).

Эксплуатационные документы содержат сведения необходимые для обеспечения функционирования и эксплуатации системы.

В рабочую документацию входят:

* **Руководство пользователя**. Руководство непосредственно для пользователей системы.
* **Описание комплекса программ**. Содержание текущего документа должно быть изложено с учетом системы в целом. Указываемые в описании комплекса программ решения, должны соответствовать фактической реализации задач с учетом новой функциональности системы по договору
* **Руководство администратора базы данных**. Руководство для лица, отвечающего за выработку требований к базе данных, её проектирование, реализацию, эффективное использование и сопровождение, включая управление учётными записями пользователей базы данных и защиту от несанкционированного доступа.
* **Руководство администратора системы**. Руководство для сотрудников, в обязанности которых входит создание оптимальной работоспособности компьютеров и программного обеспечения для пользователей, часто связанных между собой общей работой на определённый результат.
* **Руководство по инсталляции**. Документ содержит информацию по установке программного обеспечения, начальной загрузке и конфигурированию программного обеспечения.
* **Руководство по организации сопровождения**. Руководство, в котором содержатся сведения, необходимые для обеспечения работоспособности программного обеспечения – назначение программного обеспечения, конфигурация серверного комплекса.
* **Программа и методика испытаний**. Это документ, который формализует этап тестирования программного обеспечения.
* **Ведомость рабочей документации**. Ведомость содержит последовательный перечень документации, входящей в состав полного комплекта рабочей документации по техническому заданию.

Из перечисленных документов для использования конечными пользователями системы предназначено Руководство пользователя. Функциональность документа подлежит изменению в соответствии с обязательствами разработчика по договору. Информация должна быть предоставлена на каждую передаваемую по договору в эксплуатацию пользователям программу (АРМ) с учетом разработанной ранее в программе функциональности. На усмотрение разработчика руководства пользователя отдельных программ (АРМ) могут быть объединены титульным листом в один документ «руководство пользователя» с указанием ссылок на программы в оглавлении.

**Руководство пользователя** — это, собственно, основной документ, описывающий, как пользоваться системой. В хорошем руководстве сначала описывается идея системы, основные функции и как ими пользоваться, а уже потом идет описание всех клавиш и меню. Многие современные книги по программам Microsoft представляют собой яркие примеры никуда не годных руководств – можно прочесть 100 страниц и не понять основных функций.

Цель создания документа заключается в том, чтобы предоставить пользователю возможность самостоятельно решать свои прикладные задачи с помощью системы. Этой цели может служить и введение в предметную область, и ознакомление со всеми возможностями программы, и описание конкретных процедур решения задач, и приведение различных инструкций. Иногда Руководство пользователя больше похоже на справочник, к которому можно обращаться в процессе работы, а иногда – на учебник, который позволяет изучить принципы работы с программой и ее возможности, а затем применять их на практике.

Принимая во внимание все различия и особенности, сложно привести структуру любого Руководства пользователя к одному виду. Тем не менее, РД 50-34.698 предлагает нам такой список разделов:

**Введение**, где указывают область применения ПО, краткое описывают его возможности, требуемый уровень знаний пользователя и список документов, которые необходимо изучить помимо настоящего руководства;

**Назначение и условия применения**, где описывают виды деятельности и функции, которые автоматизированы и условия, при соблюдении которых автоматизация используется;

**Подготовка к работе**, где описывают комплектность дистрибутива, порядок установки и загрузки программы, а также способ проверки ее работоспособности;

**Описание операций**, представляет собой основной раздел, где описывают функции программы, процессы работы с данными, выполнение конкретных задач пользователя;

В разделе «Описание операций» указывают:

* описание всех выполняемых функций, задач (комплексов задач), процедур;
* описание операций технологического процесса обработки данных для выполнения функций, задач (комплексов задач), процедур.

В разделе «Описание операций» должна быть предусмотрена идентификация ошибочных действий и стандартизована форма сообщений об ошибках.

Кроме этого, должны быть описаны:

* соответствие между элементами интерфейсов пользователя (экранными формами) и типовыми процедурами;
* последовательность допустимых операций и переходы между экранными формами (навигация экранов);
* форма идентификации ошибочных действий и/или ситуаций;
* формы входных и выходных документов.

**Аварийные ситуации**, где описывают действия в нештатных ситуациях – сбоях в программе, ошибок в данных и т.д.;

Р**екомендации по освоению**, где приводят методические рекомендации по изучению программы и примеры использования.

Данная структура может меняться и дополняться – например, основной раздел часто разбивают на несколько значимых разделов по группам функций или задач, также в современных системах нередко добавляют раздел Интерфейс пользователя, где описывают взаимодействие пользователя с программой с примерами и снимками экрана.

Руководство пользователя разрабатывается в соответствии со следующими нормативными документами:

* ГОСТ 34.201;
* Стандарт ОАО «РЖД» СТО РЖД 04.001.1–2021 «Автоматизированные системы и программные средства. Требования к составу, содержанию и оформлению документов при создании автоматизированных систем и программных средств ОАО «РЖД»;
* РД 50-34.698.

Однако, в зависимости от сложности, назначения и области применения ПО, различные Руководства пользователя могут отличаться друг от друга по способу, методике и стилю изложения.

Грамотно написанное Руководство пользователя может сэкономить значительное количество времени на обучение и адаптацию пользователя к программе, а также снизить количество ошибок в работе что, в свою очередь, повышает экономическую эффективность системы.

**Задание №13.**

Опишите порядок работы пользователя в реализуемом ИТ проекте в соответствии с требованиями ГОСТ и СТО.

**Пример решения №13.**

Описание функциональности «Просмотр перечня зарегистрированных заявок»:

При входе в режим «АСКМ. Негабарит» на экране отображается перечень заявок, поступивших из ЛК, а также заявок, зарегистрированных пользователем – работником ОАО «РЖД» ранее (рис. 4.2.1.1). Помимо перечня, доступны фильтры для навигации по записям.

Набор фильтров представлен следующими полями:

* Дата регистрации заявки с;
* Дата регистрации заявки по;
* Наименование условия размещения и крепления;
* Номер;
* Статус (подробнее в п.4.2.1.1):
  + Черновик;
  + На рассмотрении;
  + На согласовании;
  + Возвращено на доработку;
  + Согласовано.
* Индекс негабаритности;
* Род подвижного состава:
  + Крытые;
  + Платформы;
  + Полувагоны;
  + Цистерны;
  + Изотермические;
  + Прочие.
* Наименование груза по ЕТСНГ;
* Наименование груза;
* Дорога отправления;
* Станция отправления;
* Дорога назначения;
* Станция назначения;
* Грузоотправитель.

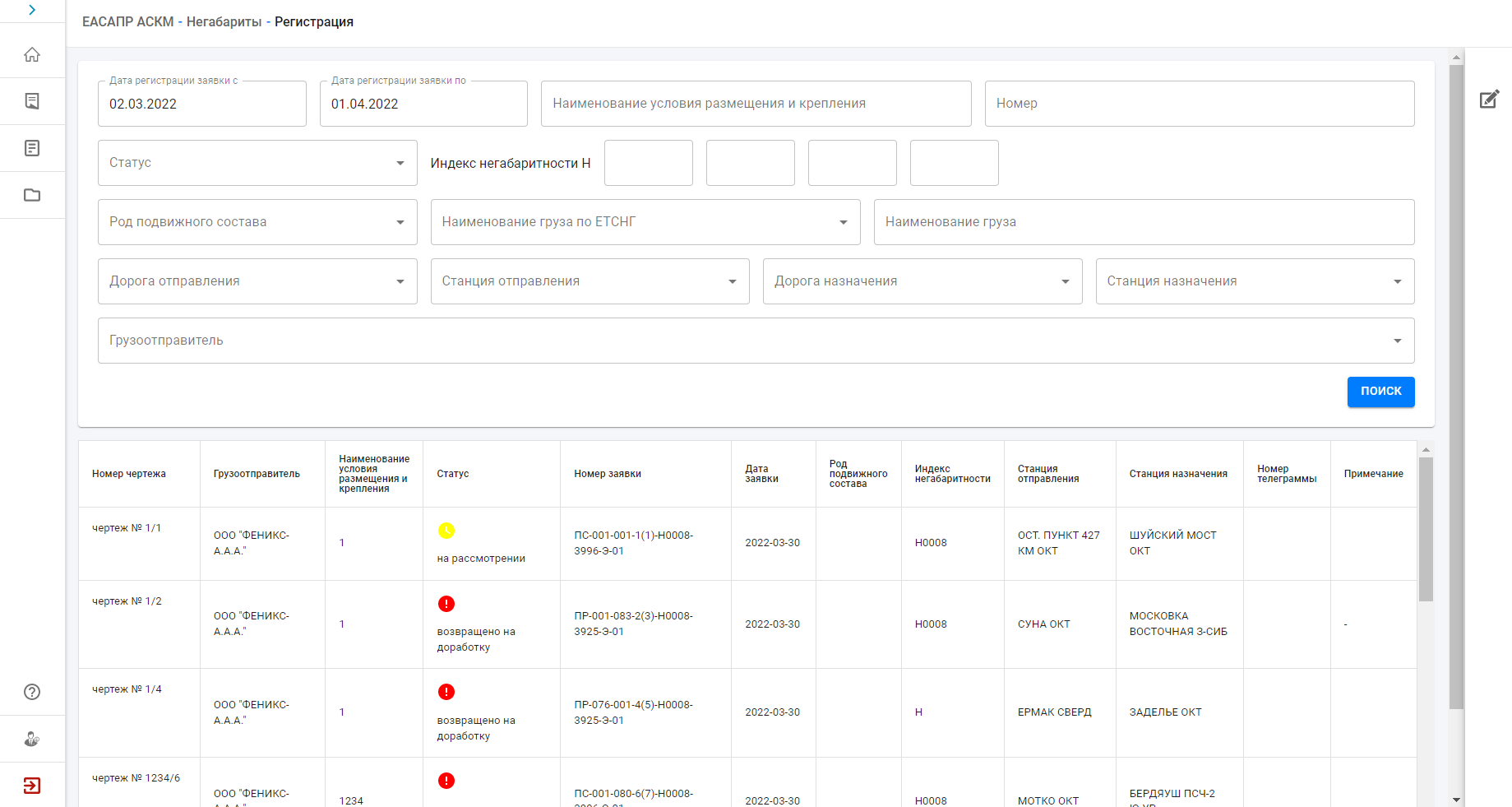


Рисунок 4.2.1.1 – АСКМ Негабарит. Главный экран. Перечень заявок

Для загрузки перечня заявок необходимо применить нужные фильтры и нажать на кнопку . На экране загрузится перечень заявок. Для просмотра заявки и работы с ней необходимо нажать на нее.

Если у Клиента нет учетной записи в ЛК РЖД ГРУЗ, он не сможет создать заявку. В таком случае работник ОАО «РЖД» может зарегистрировать заявку в АСКМ вручную. Для создания новой заявки необходимо нажать на кнопку .

# **Практическое задание № 14. Разработка Рабочих документов.**

**Руководство администратора системы**

Системный̆ администратор – сотрудник компании, задача которого заключается в обеспечении штатного функционирования компьютерной̆ техники, используемого в ее среде ПО и сети. Именно системный̆ администратор занимается установкой̆ и настройкой̆ любого необходимого программного обеспечения, следит за бесперебойной̆ работой̆ сети, а также проводит различные профилактические мероприятия, осуществляя свою деятельность на основании руководства системного администратора.

Руководством системного администратора (также встречается название «Руководство системного программиста») называется документ, предоставляющий̆ информацию, которая необходима системному администратору для выполнения его должностных обязанностей̆. Документ составляется профессиональным системным администратором, имеющим опыт работы в описываемой̆ программной̆ среде.

Содержание типового руководства системного администратора.

Разработка руководства системного администратора осуществляется в соответствие с требованиями ГОСТ 19.503-79, который̆ предписывает обязательное наличие в документе таких разделов, как:

* Общие сведения о ПО – его назначение и функционал, а также данные о программных и технических средствах, обеспечивающих работоспособность программы.
* Структура программы составные части программного обеспечения, их иерархия, а также связи между ними и прочими программами.
* Настройка программы – подробные инструкции по настройке ПО для решения конкретных задач (оптимизация для работы на используемых технических средствах, выбор функций, настройка интерфейса и т.д.).
* Проверка ПО – описание способов проверки настроенной̆ программы, позволяющих сделать заключение о ее работоспособности (указываются конкретные примеры и методы тестирования, а также результаты, которые должны быть получены).
* Дополнительные возможности – описание дополнительного функционала программ, способов вызова и настройки функций.
* Сообщения системному администратору – тексты сообщений, формирующихся в процессе настройки, проверки и выполнения программы, их содержание и действия, которые должен предпринять системный̆ администратор по ним.

Имеющиеся нормативы допускают объединение и введение дополнительных разделов. Это зависит от особенностей̆ используемого программного обеспечения.

Руководство системного администратора призвано максимально подробно раскрыть информацию о комплексе технических средств, необходимых для работы программного обеспечения, а также описать процесс и особенности установки, настройки и поддержки работоспособности вашего ПО.

Документация администрирования при эксплуатации информационной̆ системы должна обеспечивать поддержку первичной̆ инсталляции, штатного функционирования и восстановления программ и данных после сбоев и отказов. Управляющая деятельность администратора состоит в манипулировании управляемыми объектами и должна описываться, анализироваться и регламентироваться совокупностью требований и документов. Для этого необходима полная документация о компонентах информационной̆ системы (компьютерах, сетевых устройствах), которые имеют свои особенности в управлении с помощью специальных программных компонентов, поддерживающих администрирование и управление системой̆.

К основным функциям системы администрирования, документы для которых подлежат разработке и оцениванию, относятся:

* планирование использования памяти и производительности вычислительной̆ системы в рабочем режиме применения ПС, оперативное управление и распределение ресурсов информационной̆ системы;
* инсталляция и генерация рабочих версий ПС для оперативных пользователей̆;
* управление и учет внешней̆ среды при выполнении адаптации и реконфигурации конкретного ПС;
* выявление, регистрация и накопление данных о сбоях и дефектах функционирования программ и данных;
* управление средствами защиты информации и санкционированного доступа пользователей̆, анализ попыток взлома системы защиты, восстановление программ и информации баз данных при искажениях;
* сбор и обобщение статистики о качестве функционирования ПС в составе системы обработки информации.

**Руководство администратора баз данных**

Документ содержит информацию, необходимую для освоения работы с автоматизированной системой в качестве администратора базы данных системы.

Руководство администратора баз данных состоит из следующих разделов:

* **Описание комплекса**. В разделе указывают технические характеристики ПТК, на котором размещена СУБД, Перечень установленных приложений и их версии, Схему взаимодействия программных компонентов в системе с БД, Методы резервирования, порядок репликации данных и иные механизмы, Описание применяемого кластерного ПО.
* **Конфигурация СУБД**. В разделе указывают описание всех настроек в используемой СУБД, отличных от установленных по умолчанию.
* **Конфигурация ОС.** В разделе указываются все параметры ОС, которые должны быть изменены при развертывании системы для корректной работы СУБД.
* **Размещение СУБД.** В разделе должны быть описаны пути ко всем объектам СУБД на сервере:
* Пути размещения рабочего каталога СУБД.
* Пути размещения табличных пространств БД.
* Пути размещения конфигурационных файлов СУБД.
* Пути размещения логов транзакций СУБД.
* Пути размещения архива логов транзакций СУБД.
* Пути размещения локальных копий СУБД.
* Пути размещения диагностических журналов СУБД.
* Пути к исполняемым файлам СУБД.
* **Описание баз данных.** Раздел должен содержать описание всех созданных прикладных объектов, необходимых для работы приложения. В частности, описываются:
* БД (системные БД указывать не нужно);
* Схемы (наименование, функциональное назначение в системе, владелец, БД в которой размещены).
* Таблицы (наименование, функциональное назначение в системе, структура таблицы, владелец, в какой схеме размещены, глубина хранения данных в таблице, если она отлична от глубины хранения БД).
* Триггеры (наименование, назначение).
* Хранимые процедуры (наименование, назначение, порядок конфигурирования, размещение в БД).
* Функции (наименование, назначение, порядок конфигурирования, размещение в БД).
* Пакетные задания, включая задания чистки (наименование, назначение, расписание, порядок конфигурирования).
* Ограничения (наименование, назначение, затрагиваемые объекты).
* Индексы первичной инсталляции (наименование, назначение, к каким таблицам относятся).
* Прочие объекты.
* **Руководство к СУБД.** Раздел должен содержать следующие инструкции:
* Процедура остановки/запуска.
* Инструкция по переходу на резерв.
* Обновление СУБД.
* Ввод вывод резервного узла из кластера.
* и др.
* **Ролевая модель.** Раздел должен содержать состав прикладных и технических пользователей БД с перечнем ролей и привилегий, необходимых для эксплуатации приложения, так же должна быть добавлена инструкция по созданию указанных пользователей (допускается отображение в виде команд) и расширению/уменьшению их полномочий.
* **Политика резервного копирования СУБД.** Раздел должен содержать следующую информацию в табличном виде:

1. Требования к локальному хранению резервных копий – сколько дней, возможность сбора архива транзакционных логов и т.д.
2. Требования к хранению резервных копий на СРК – сколько дней, возможность сбора архива транзакционных логов и т.д.
3. Возможность резервного копирования средствами виртуализации, информация по регламенту хранения.
4. Способ реализации резервного копирования БД:

* Локальная резервная копия переносится на СРК файловым агентом.
* Резервное копирование осуществляется на СРК агентом СУБД напрямую.
* Для систем выдается отдельное файловое хранилище NFS раздел, которые монтируется напрямую к серверу СУБД. Резервное копирование осуществляется на примонтированный раздел, далее резервная копия переносится на СРК файловым агентом.

Также в данном разделе или в приложениях к документу следует привести следующие инструкции:

* Инструкция по созданию полной резервной копии БД.
* Инструкция по созданию резервной копии отдельных объектов БД.
* Инструкция по хранению транзакционных логов.
* Инструкция по очистке устаревших транзакционных логов и бэкапов.
* **Требования к восстановлению СУБД.** В разделе указывается:
* Методики и механизмы восстановления данных в случае различных сценариев потери/порчи данных (например, восстановление БД из резервной копии штатными средствами СУБД, специальными утилитами или скриптами).
* RPO.
* RTO.
* **Методика восстановления СУБД.** Раздел должен содержать подробные инструкции восстановления СУБД в случае порчи или потери данных:
* Инструкция по восстановлению БД из полного локально бекапа и с СРК (если резервная копия пишется сразу на СРК).
* Инструкция по частичному восстановлению данных.
* Инструкция по восстановлению БД на точку во времени и до выполнения определенной транзакции.
* Инструкция по восстановлению табличного пространства.
* Инструкция по восстановлению ролевой модели.
* Команды пересоздания индексов (если возможно пересоздать в режиме онлайн – указать).
* Инструкция по перестройке или перемещению как отдельного так и всех индексов для конкретной таблицы с указанием факта возможности проведения данной операции без блокировки объектов БД.
* Инструкция по дефрагментации/перемещению таблиц/БД с указанием факта возможности проведения данной операции без блокировки объектов БД.
* **Требования к информационной безопасности.** Раздел должен содержать описание требований к информационной безопасности и защиты данных и примененных механизмов обеспечения этих требований. Для систем, где отдельная СЗИ не создается, такими механизмами может быть ограничение по СПД, шифрование и т.п. Например, указывается таблица правил доступа для локальных firewall серверов, где инсталлированы СУБД.
* **Дополнительное ПО, необходимое для обеспечения работы.** Раздел должен содержать описание используемых вспомогательных механизмов СУБД (все возможные надстройки СУБД, форки и утилиты, не входящие в стандартный пакет СУБД, примененные исключительно для обеспечения работоспособности конкретного архитектурного и прикладного решения в системе, драйвер JDBC/ODBC и его версия для работы серверов приложений и смежных систем с СУБД (включая клиентские приложения). Также приводятся параметры строки подключения серверов приложений к БД для автоматического переключения на резервный сервер.
* **Мониторинг работоспособности БД.** Указываются механизмы сбора метрик работоспособности СУБД и перечень собираемых метрик. Под механизмами подразумеваются системные таблицы, агенты мониторинга, сторонние утилиты.
* **Средства и методы диагностики сбоев.** Указываются утилиты и клиентские приложения диагностики работоспособности СУБД, например, технические страницы опроса работоспособности (JSON). Указывается место размещения дистрибутива, версия, порядок инсталляции.

Данный раздел должен содержать подробные инструкции по операциям диагностики сбоев, например:

* Поиск и устранение блокировок.
* Выбор оптимально плана выполнения запроса и его фиксация.
* Проверка и изменение интервалов обслуживания БД.
* Проверка и создание индексов для партиционированных таблиц.
* и т.д.

**Руководство по инсталляции**

Документ содержит информацию, необходимую для инсталляции нового ПО на серверы и рабочие места пользователей.

В документе описывается порядок Порядок установки ПО системы и состоит из разделов:

* **Требования к установке ПО системы**. В подразделе приводятся требования к системному программному обеспечению, к составу и характеристикам технических средств, необходимых для функционирования разрабатываемого ПО.
* **Установка ПО**. В подразделе описывается процедура инсталляции программного обеспечения ПО на серверы и рабочие места пользователей, а также процедура проверки работоспособности функциональности ПО
* **Настройка ПО**. В подразделе описывается процедура настройки серверной и клиентской частей ПО, а также настройка интеграций.

**Описание комплекса программ**

Документ содержит описание комплекса программ, входящих в программное обеспечение, их наименования, логическую структуру, структуру входных и выходных данных, описание вызова и загрузки программ.

Основные разделы документа:

* **Программное обеспечение, необходимое для функционирования программы.** Указывается системное программное обеспечение (операционные системы, СУБД, библиотеки, вспомогательные программы и пр.), необходимое для функционирования программы. Если для всех программ комплекса нужно одинаковое системное программное обеспечение, допускается вынести данный подпункт в качестве подраздела раздела 1 настоящего документа.
* **Языки программирования.** Приводится перечень языков программирования, используемых при разработке программы.
* **Функциональное назначение программы.** В разделе «Функциональное назначение программы» указываются классы решаемых задач и (или) назначение программы (ее функции) и сведения о функциональных ограничениях на применение.
* **Описание логической структуры программы. Алгоритм программы.** Способ описания алгоритма - произвольный. Рекомендуется его оформлять в виде блок-схемы в соответствии с UML или ГОСТ 19.701.
* **Описание логической структуры программы. Структура программы с описанием функций составных частей и связи между ними.** Главное назначение описания структуры - уточнение того, из каких частей состоит программа и как они связаны между собой. Составные части программы следует увязать с текстом программы на исходном языке и указанием их смысловой нагрузки.
* **Описание логической структуры программы. Связи программы с другими программами.** Указываются программы, с которыми должна взаимодействовать данная программа. Описывается характер взаимодействия.
* **Вызов и загрузка программы.** В разделе «Вызов и загрузка программы» указываются:
* способ вызова с соответствующего носителя (конкретные операторы, директории);
* входные точки в программу;
* сведения об использовании памяти, объеме загрузочных модулей программы в килобайтах.
* **Входные данные программы.** В разделе «Входные данные программы» указываются:
* тип данных (числовые, буквенные, целые и пр.);
* организация данных;
* предварительная подготовка данных;
* формат, описание данных, способ кодировки.
* **Выходные данные программы.** В разделе «Выходные данные программы» указывают тип данных (числовые, буквенные, целые и пр.), организацию выходных данных, формат данных, способ кодирования данных.

**Руководство по организации сопровождения (РОС)**

Содержание документа должно быть изложено с учетом системы в целом, с учетом предыдущих версий системы. Указываемые в РОС решения должны соответствовать фактической реализации задач с учетом новой функциональности системы по договору.

Формирование некоторых разделов руководства по организации сопровождения выполняется эксплуатирующей организацией.

Руководство по организации сопровождения разрабатывается с целью описания механизмов сопровождения системы на серверах ГВЦ. Документ содержит информацию по организации сопровождения пользователей на сетевом/дорожном уровне. В документе приведены основные сведения, необходимые для обеспечения работоспособности системы, назначение системы, конфигурация серверного комплекса, распределение работ между разработчиком, заказчиком и эксплуатирующей организацией.

В процессе эксплуатации Системы возможно внесение изменений в настоящий Документ. Изменения должны быть согласованы со всеми заинтересованными сторонами.

Все изменения настоящего документа подлежат регистрации в Листе регистрации изменений.

Процедура переподписания РОС (титульный лист и таблица Согласования) осуществляется при модификации АС (при сдаче/приемке новой очереди).

**Руководство программиста**

Руководство программиста относится к эксплуатационно-технической документации. Разрабатывается такой документ для программных продуктов. Предназначен для ознакомления программистом, который будет решать те или иные задачи, связанные с эксплуатацией данной программы.

Руководство программиста необходимо в нескольких случаях:

* Программа, на которую составляется документация, представляет собой среду разработки или библиотеку.
* Данный программный продукт предоставляет платформу для написания типовых программ или систем.
* Распространение продукта проходит совместно с программным кодом или же происходит его постоянная модификация разработчиком.

При помощи такого документа программисту должна быть представлена вся необходимая информация, которая может быть использована для создания собственных программных продуктов на базе данной системы. Информация должна быть предоставлена в достаточном количестве. Разработчики же при помощи руководства программиста имеют возможность зафиксировать текущее состояние выпускаемого продукта, чтобы избежать путаницы при выпуске новых продуктов.

К типичным задачам такого документа относится:

* Уточнение и пояснение специалисту текущего состояния объектов, их местонахождения и методов взаимодействия. Также руководство программиста должно четко разграничить объекты, которые изначально внесены в систему, и объекты, которые программист создает самостоятельно.
* Перечисление дополнительных средств разработки, которые потребуются при работе, кроме текущего продукта.
* Уточнение требований к системе, программной среды, а также средств, необходимых для запуска.

**Руководство программиста** должно составляться грамотным профессионалом, знакомым со спецификой работы конкретного программного продукта и правилами составления подобных документов, регламентируемых по [ГОСТ 19.504-79](http://www.swrit.ru/doc/espd/19.504-79.pdf).

Требования к заполнению руководств программиста установлены соответствующим государственным стандартом. Структура такого документа должна включать в себя:

* Предназначение и условия эксплуатации программного продукта.
* Основные характеристики программы.
* Методы обращения к программному продукту.
* Основная входная и выходная информация.
* Сообщения.

На данный момент руководство программистов включает также различные схемы - схемы баз данных, диаграммы классов, графы вызова функций.

**Задание №14.**

Ознакомиться с описанием документов из состава рабочей документации. Сформировать требования к реализуемому ИТ проекту в рамках курса практических занятий, по любому из разделов указанных документов.

# **Практическое задание № 15. Разработка Программы и методики испытаний.**

**Программа и методика проведения испытаний системы** (подсистемы) на этапе опытного функционирования предназначена для установления данных, обеспечивающих получение и проверку проектных решений, выявление причин сбоев, определение качества работ, показателей качества функционирования системы (подсистемы), проверку соответствия системы требованиям техники безопасности, продолжительность и режим испытаний.

Программа и методика испытаний – это документ, входящий в комплект конструкторской документации, составляемой на автоматизированную систему или программу на стадии разработки. Программа методики испытаний призвана установить технические данные, которые подлежат проверке во время испытаний всей системы в целом или ее отдельных компонентов. Этим документом устанавливается порядок опытов и способы контроля их результатов.

Согласно руководящему документу [РД 50-34.698-90](https://www.swrit.ru/doc/gost34/50_34_698_90.pdf), в этом документе перечисляются все проверки, призванные установить эффективность проектных решений, выявить причины отказов или сбоев, определить качество проведенных работ, проверить соответствие АСУ технике безопасности, а также устанавливается продолжительность всех опытов, их режим и прочее.

Программа и методика испытаний содержит перечень необходимых проверок, проводимых во время испытаний. К таким проверкам обычно относится:

* Проверка соответствия техзаданию;
* Проверка комплектности системы;
* Проверка комплектности и качества документации;
* Установление достаточности, комплектности и качества программной поддержки и документации на нее;
* Установление квалификации обслуживающих работников;
* Проверка соответствия системы требованиям функционального назначения;
* Установление пригодности системы к контролю;
* Выявление нарушений техники безопасности, санитарии, эргономики;
* Проверка взаимодействия системы с другими программными средствами.

Согласно государственному стандарту [ГОСТ 19.301-79](https://www.swrit.ru/doc/espd/19.301-79.pdf), документ «программа и методика испытаний» оформляется в соответствии с [ГОСТ 19.105-78](https://www.swrit.ru/doc/espd/19.105-78.pdf) (общими требованиями к оформлению программных документов) и должен содержать следующие разделы:

* Объект испытаний, с указанием наименования, области применения, обозначения АСУ.
* Цель испытаний, с указанием конкретных данных, которые должны быть получены по результатам опытов.
* Требования к программе. Содержит перечень требований, которые заданы тех заданием и должны быть проверены опытным путем.
* Требования к программным документам, с перечислением состава документов и особых требований на испытания системы, установленных ТЗ.
* Средства и порядок испытаний, с указанием технических и программных средств, использующихся для проведения испытаний, и порядка опытов.
* Методы испытаний. Описание применяемой методологии, с перечнем данных, которые должны быть получены во время проведения испытаний.
* Приложения. Программы и методики испытаний могут содержать дополнительные материалы – графики, таблицы, тестовые примеры и их контрольные распечатки.

Таблица 15.1 - Раздел 1 «Объект испытаний»

| **Наименование раздела** | **Пример заполнения** |
| --- | --- |
| Полное наименование автоматизированной информационной системы, обозначение | Полное наименование – корпоративное хранилище данных.  Обозначение – КХД или Система. |
| Комплектность испытательной автоматизированной информационной системы | Испытаний проводятся для всех подсистем и функций Системы. Так же испытания включают проверку необходимого сетевого оборудования и каналов связи |

Таблица 15.2 - Раздел 2 «Цель испытаний»

| **Наименование раздела** | **Пример заполнения** |
| --- | --- |
| Цель испытаний | Целью проведения испытаний является: - проверка взаимодействия подсистем Системы; - проверка работоспособности Системы; - проверка соответствия Системы требованиям приведенным в документе «[Техническое задание](https://www.prj-exp.ru/patterns/pattern_tech_task.php)»; - проверка готовности Системы к проведению опытной эксплуатации или приемочных испытаний на территории Заказчика. |

Таблица 15.3 - Раздел 3 «Общие положения»

| **Наименование раздела** | **Пример заполнения** |
| --- | --- |
| Общие положения | Настоящая Программа и Методика Испытаний разработана в соответствии со следующими документами: - Процедура управления внутренней нормативной документацией. - Процедура управления записями в системе менеджмента качества. - Типовая методика измерений и оценки эксплуатационных параметров автоматизированных информационных систем. - Перечень эксплуатационных параметров автоматизированных информационных систем. - ГОСТ 34.603-92 Виды испытаний автоматизированных систем. - РД 50-34.698-90 Автоматизированные системы требования к содержанию документов. - ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. - РД 50-34.698-90 Методические указания информационная технология комплекс стандартов и руководящих документов на автоматизированные системы автоматизированные системы требования к содержанию документов. - [Техническое задание](https://www.prj-exp.ru/patterns/pattern_tech_task.php) |
| Место и продолжительность испытаний | Испытательный стенд находится на территории Заказчика по адресу: г. Москва, ул. Ленина, д.111. Испытания проводятся в течении X дней. |
| Участники испытаний | В испытаниях принимают участие Заказчик и Исполнитель. Допускается привлечение экспертов из сторонних организаций |
| Перечень предъявляемых на испытания документов | Паспорт. - Общее описание системы. - Технологическая инструкция. - Руководство пользователя. - Описание технологического процесса обработки данных (включая телеобработку). - Инструкция по формированию и ведению базы данных (набора данных). - Состав выходных данных (сообщений). - Каталог базы данных. - Описание программ. - Текст программ. |

Таблица 15.4 – Раздел 4 «Объем испытаний»

| **Наименование раздела** | **Пример заполнения** |
| --- | --- |
| Перечень этапов испытаний и проверок, а также количественные и качественные характеристики, подлежащие оценке | Перечень проверок требований изложенных в техническом задании на создание Системы |
| Последовательность проведения и режима испытаний |
| Требования по испытаниям программных средств |
| Перечень работ, проводимых после завершения испытаний, требования к ним, объем и порядок проведения | По результатам испытаний делают заключение о работоспособности АС и возможности её ввода в эксплуатацию.  По результатам предварительных испытаний рабочая комиссия может принять одно из следующих решений:   1. система работоспособна и рекомендована к вводу в опытную эксплуатацию; 2. опытную эксплуатацию системы начать после устранения некоторых (существенных) замечаний; 3. система не готова к опытной эксплуатации. Рекомендуется полностью устранить замечания комиссии, после чего провести повторную проверку готовности к опытной эксплуатации.   На основании решения комиссии составляются протокол и акт приемки в опытную эксплуатацию (акт о невозможности ввода в опытную эксплуатацию).  При необходимости производится доработка программного обеспечения и документов, предъявляемых к испытаниям.  По завершении корректировки программного обеспечения и рабочей (эксплуатационной) документации разработчик передает рабочую документацию и программные средства в ГВЦ ОАО «РЖД» для ввода в постоянную эксплуатацию. |

Таблица 15.5 – Раздел 5 «Условия проведения испытаний»

| **Наименование раздела** | **Пример заполнения** |
| --- | --- |
| Условия проведения испытаний | Перечень проверок требований изложенных в техническом задании на создание Системы |
| Условия начала и завершения отдельных этапов испытаний | Этапы испытания системы проводятся последовательно. Условием начала каждого этапа испытаний является успешное завершение предыдущего этапа.  Условием завершения каждого этапа является завершение всех проверок, предусмотренных на данном этапе. |
| Требования к техническому обслуживанию системы | Техническое обслуживание ПО системы во время испытаний должно проводиться в соответствии с требованиями рабочей документации. |
| Меры, обеспечивающие безопасность и безаварийность проведения испытаний | При проведении испытаний должно быть обеспечено соблюдение требований безопасности, установленных ГОСТ 12.2.007.0, «Правилами техники безопасности при эксплуатации электроустановок потребителей» и «Правилами технической эксплуатации электроустановок потребителей». |
| Порядок взаимодействия организаций, участвующих в испытаниях | Разработчик письменно извещает Заказчика о готовности к проведению приемо-сдаточных испытаний не позднее, чем за 5 дней до намеченного срока проведения испытаний.  Заказчик Приказом назначает срок проведения испытаний и приемочную комиссию, которая должна включать в свой состав представителей Заказчика и Разработчика.  Заказчик письменно извещает организации, которые должны принять участие в приемо-сдаточных испытаниях.  Заказчик совместно с Разработчиком проводит все подготовительные мероприятия для проведения испытаний на объекте испытаний, а также проводят испытания в соответствии с настоящей Программой и методиками.  Заказчик осуществляет контроль проведения испытаний. |
| Порядок привлечения экспертов для исследования возможных повреждений в процессе проведения испытаний | Сторонние эксперты к испытаниям не привлекаются. Возможные повреждения в процессе проведения испытаний выявляются разработчиком. |
| Требования к персоналу, проводящему испытания и порядок его допуска к испытаниям | Персонал, проводящий испытания, должен обладать навыками работы с персональным компьютером, операционной системой Windows, а также должен ознакомиться с рабочей документацией.  Допуск сотрудников к проведению испытаний производится на основании представления руководителем подразделения, в котором работает сотрудник. |

Таблица 15.6 – «Описание контрольного примера»

| **Наименование раздела** | **Пример заполнения** |
| --- | --- |
| Описание контрольного примера | Контрольный пример позволяет осуществить проверку работоспособности системы. Для выполнения контрольного примера необходимо последовательно выполнить следующие действия:  Перейти по ссылке http://\_\_\_\_\_/  Убедиться в доступности системы по ссылке;  Убедиться в возможности авторизации в системе;  Ввести логин пользователя центрального уровня «Специалист», пароль: New123\*;  Убедиться, что осуществлен вход в систему.  Контрольный пример считается выполненным успешно, если успешно закончились все указанные проверки. |

Таблица 15.7 – «Описание методик проверок»

**П.5.2. Проверка модуля согласования условий на перевозку негабаритного, тяжеловесного и длинномерного груза**

| **Пункт ТЗ** | **Группа функций** | **Контрольное действие** | **Результат испытания работоспособности функции** |
| --- | --- | --- | --- |
| **4.2.2** | **Модуль согласования условий на перевозку негабаритного, тяжеловесного и длинномерного груза** | | |
| 4.2.2.1 | Регистрация пользователем Программы для ЭВМ заявок на предварительное согласование возможности, способа и условий перевозки и окончательное согласование технической документации на перевозку негабаритных, тяжеловесных и длинномерных грузов (Чертежей и расчетов размещения и крепления грузов на подвижном составе), поступивших на согласование от Клиента на бумажном носителе с присвоением номера | 1. Войти в систему АСКМ по адресу [http://easapr-askm-test.msk.oao.rzd/](http://easapr-askm-test.msk.oao.rzd/admin)   Пользователь логин: Специалист, пароль: 123.  Режим «НГБ – Согласование».   1. Нажать на кнопку создания новой записи. 2. Ввести информацию, сохранить данные | 1. Открыт интерфейс системы. 2. Открыт режим создания заявки. 3. Отсутствие сообщения об ошибке. |

**Задание №15.**

Сформировать методику проверки реализуемого ИТ проекта в рамках курса практических занятий по аналогии с таблицей 15.7.

# **Практическое задание № 16. Презентация ИТ проекта.**

После разработки программного обеспечения походит этап сдачи проекта заказчику и презентация нового реализованного продукта. Для этих целей проводятся приемки в опытную и постоянную эксплуатацию, на которых происходит демонстрация разработанного программного обеспечения комиссии со стороны заказчика в соответствии с Программой и методикой испытаний.

Помимо выполнения операция по контрольному примеру и методики проверки для более успешной презентации проекта и вовлечение членов комиссии в результат проделанной работы рекомендуется формировать презентационный материал.

В составе презентации должна быть отражена следующая информация:

* Название проекта;
* Основание выполнения работа (номер договора, нормативные документы, методологии, на основании которых было разработано ПО);
* Цель и назначение проекта;
* Состав выполненных работ (структура реализованного проекта – модули, сервисы, функциональные возможности);
* Схематичное изображение бизнес-процесса (приветствуется сравнение моделей «как было» и «как стало»);
* Схема информационных потоков (в случае наличия интеграционных решений в реализованном проекте);
* Ролевая модель проекта и потенциальные пользователи;
* Эффекты от внедрения проекта.

Рекомендации по визуальному оформлению презентационных материалов:

* Дизайн слайдов должен быть простым. На диаграммах не должно быть сносок и ссылок; в текстовых слайдах количество слов не должно превышать 30-ти;
* Используйте цвет осмысленно, а не для красоты. Цвет применяется для акцентирования, привлечения внимания к лейтмотиву презентации, выделения или символического выражения;
* Специальные эффекты (например, анимация) должны быть сведены к минимуму. Их использование должно диктоваться содержанием;
* Хорошие, читаемые предустановленные шрифты - Calibri, Helvetica, Open Sans и Roboto;

Полезные ссылки:

* Где можно подобрать цветовую гамму для презентации:

color.adobe.com

flatuicolorpicker.com

* Где можно найти иконок в презентации:

flaticon.com

icons8.com

thenounproject.com

* Бесплатные браузерные аналоги PowerPoint:

Google презентации

visme.co

piktochart.com

**Задание №16.**

Сформировать презентацию реализуемого ИТ проекта в рамках курса практических занятий

# **Приложение 1. Исходные данные для выполнения практических заданий**

| **№ вар.** | **Сроки, мес** | **Бюджет, млн. руб** | **Команда** | **Бэкраунд**  **проекта** | **Заказчик** | **Наличие интеграционных решений** | **Принадлежность ПТК** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|  | 9 | 2, 00 | собственная | с нуля | внешний | да | разработчика |
|  | 12 | 1, 00 | частично аутсорсинг | развитие | внутренний | нет | заказчика |
|  | 3 | 5, 00 | полностью аутсорсинг | модификация | гос.заказ | да | аренда |
|  | 9 | 2, 00 | собственная | с нуля | стартап | нет | разработчика |
|  | 12 | 1, 00 | частично аутсорсинг | развитие | лицензия | да | заказчика |
|  | 3 | 5, 00 | полностью аутсорсинг | модификация | внешний | нет | аренда |
|  | 9 | 2, 00 | собственная | с нуля | внутренний | да | разработчика |
|  | 12 | 1, 00 | частично аутсорсинг | развитие | гос.заказ | нет | заказчика |
|  | 3 | 5, 00 | полностью аутсорсинг | модификация | внутренний | да | аренда |
|  | 9 | 2, 00 | собственная | с нуля | лицензия | нет | разработчика |
|  | 12 | 1, 00 | частично аутсорсинг | развитие | внешний | да | заказчика |
|  | 3 | 5, 00 | полностью аутсорсинг | модификация | внутренний | нет | аренда |
|  | 9 | 2, 00 | собственная | с нуля | гос.заказ | да | разработчика |
|  | 12 | 1, 00 | частично аутсорсинг | развитие | стартап | нет | заказчика |
|  | 3 | 5, 00 | полностью аутсорсинг | модификация | внешний | да | аренда |

# **Приложение 2. Исходные данные для выполнения практического задания №4**

П2.1. Исходные данные для задания 4.1

| **№ вар.** | **Единовременные инвестиции,**  **млн. руб** | **Срок эксплуатации, лет** | **Выручка от реализации проекта по годам,**  **млн руб** | **Операционные расходы по годам,**  **млн руб** | **Норма амортизации, %** | **Ставка дисконтирования, %** |
| --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** |
|  | 12, 00 | 5 | 6, 7, 8.1, 9.4, 10.9 | 5, 5, 4, 3,1 | 20 | 15 |
|  | 11, 00 | 3 | 3, 5, 7, 9, 11 | 3, 4, 2, 2, 1 | 10 | 10 |
|  | 15, 00 | 4 | 2, 4.5, 6, 7.5, 10 | 10, 8, 3, 3, 3 | 7 | 20 |
|  | 12, 00 | 2 | 6, 7, 8.1, 9.4, 10.9 | 5, 5, 4, 3,1 | 12 | 15 |
|  | 11, 00 | 1 | 3, 5, 7, 9, 11 | 3, 4, 2, 2, 1 | 4 | 10 |
|  | 15, 00 | 5 | 2, 4.5, 6, 7.5, 10 | 10, 8, 3, 3, 3 | 20 | 20 |
|  | 12, 00 | 3 | 6, 7, 8.1, 9.4, 10.9 | 5, 5, 4, 3,1 | 10 | 15 |
|  | 11, 00 | 4 | 3, 5, 7, 9, 11 | 3, 4, 2, 2, 1 | 7 | 10 |
|  | 15, 00 | 2 | 2, 4.5, 6, 7.5, 10 | 10, 8, 3, 3, 3 | 12 | 20 |
|  | 12, 00 | 1 | 6, 7, 8.1, 9.4, 10.9 | 5, 5, 4, 3,1 | 4 | 15 |
|  | 11, 00 | 5 | 3, 5, 7, 9, 11 | 3, 4, 2, 2, 1 | 20 | 10 |
|  | 15, 00 | 3 | 2, 4.5, 6, 7.5, 10 | 10, 8, 3, 3, 3 | 10 | 20 |
|  | 12, 00 | 4 | 6, 7, 8.1, 9.4, 10.9 | 5, 5, 4, 3,1 | 7 | 15 |
|  | 11, 00 | 2 | 3, 5, 7, 9, 11 | 3, 4, 2, 2, 1 | 12 | 10 |
|  | 15, 00 | 1 | 2, 4.5, 6, 7.5, 10 | 10, 8, 3, 3, 3 | 4 | 20 |

П2.2. Исходные данные для задания 4.1

| **Критерий** | **Проект А** | **Проект Б** | **Проект В** | **Проект Г** | **Проект Д** | **Проект Е** | **Проект Ж** | **Проект З** | **Проект И** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Срок жизни, мес | 36 | 23 | 29 | 34 | 25 | 30 | 33 | 35 | 21 |
| DPBP | 20 | 22 | 24 | 26 | 20 | 22 | 24 | 26 | 20 |
| IRR,% | 15 | 17 | 11 | 14 | 13 | 16 | 12 | 18 | 10 |