Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Факультет систем управления (ФСУ)

Кафедра автоматизированных систем управления (АСУ)

Отчет по практической работе №1

«Оценка дерева целей методом анализа иерархий»

по дисциплине

«Теория систем и системный анализ»

Студент гр. 431-3

\_\_\_\_\_\_\_\_\_ Е.П. Бекиш

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2023г.

Ассистент кафедры АСУ,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.М. Аверьянова

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2023г.

Томск 2023

**Цели и задачи**

**Цель**: получить практические навыки в разработке дерева целей решения проблем конкретной системы и оценке важности целей методом анализа иерархий

**Задачи:**

1. Краткое описание объекта исследования

2. Моделирование проблемосодержащей системы

2.1. Модель взаимосвязи системы с окружающей средой

2.2. Структурный анализ системы

2.3. Построение дерева причин.

3. Постановка целей и поиск решений

3.1. Построение дерева целей (количество уровней – 4).

3.2. Оценка целей методом анализа иерархий

1. **ОПИСАНИЕ ОБЪЕКТА ИССЛЕДОВАНИЯ**

В настоящий момент информационные технологии развиваются очень быстро. Данное явление можно заметить в существование большого количество IT компаний. По данным сайта РИА Новости [1], с января 2020 года по май 2023 года, число компаний увеличилось с 178 тыс. до 201 тыс. организаций. Из всего множества IT-компаний можно выделить основные их деятельности:

* Разработка программного обеспечения — это процесс создания и модификации программного обеспечения, включающий в себя планирование, проектирование, разработку, тестирование и внедрение программного обеспечения для удовлетворения определенных потребностей пользователя. Целью разработки программного обеспечения является создание качественных, эффективных, безопасных и удобных в использовании программных продуктов.
* Управление потоками баз данных — создание баз данных и информационных ресурсов, формирование и ведение баз данных, включая сбор из одного или более источников, администрирование баз данных, поиск данных, их отбор, сортировка по запросам, предоставление отобранных данных пользователям.
* Обслуживание IT-инфраструктур — удаленное управление административными правами доступа, учетными записями пользователей, аппаратной платформой ИТ-инфраструктуры, баз данных, системой мониторинга. Выполнение системных настроек, получение обновлений, их установка и сопровождение установки, обеспечение сетевой доступности, резервное копирование, выполнение плановых и внеплановых восстановлений, осуществление переноса данных.
* Создание серверов — предоставление инфраструктуры для хостинга, web-хостинг, услуги потоковой передачи данных, предоставление прикладного хостинга, выделенный хостинг.

В работе приводится системный анализ IT-компании, занимающейся разработкой программного обеспечения, используя заказную модель разработки.

Из всего следует, что целью данной системы является получение прибыли за предоставление услуг в разработке программного обеспечения.

Так как IT-компания является самостоятельной организацией, то она подвержена сильному влиянию со стороны внешней среды.

Получается, важным элементом внешней среды IT- компании является клиент, который предоставляет прямое финансирование. Далее клиент передает один из вариантов реализации проекта, это может быть как уже программный продукт, так и техническое задание, чтобы компания смогла, основываясь на данных, управлять процессами продукта. Тем самым более качественно и быстро будет выполнена работа, которая в свою очередь повлечет за собой повторное обращение клиента в IT-компанию и распространит информацию о ней, следовательно добавив клиентскую базу.

Так же не стоит забывать о еще одном влияющем компоненте на систему – конкуренты. То, какие услуги, расценки, клиентская база, известность и т.п. в других компаниях – это оказывает главное влияние на количество клиентов в данной организации [2]. Так же сама компания является конкурентом по отношению к другим, оказывая на остальных то же самое влияние.

Одним из важных компонентов, влияющим на систему, является государственные органы, которые контролируют деятельность компаний. Перечень необходимых услуг представлен на портале Министерства цифрового развития, связи и массовых коммуникацию РФ [3]. Данный орган контролирует, чтобы каждая компания была аккредитована, тем самым числилась в реестре аккредитованных организаций для осуществления деятельности в области информационных технологий. В свою очередь, организацию обеспечивает соблюдение налоговых отчислений в соответствующий государственный орган.

Поставщики тоже являются элементом, оказывающим влияние на систему. Для того, чтобы IT-компания существовала нужно помещение, техническое оборудование и т.п. Если рассматривать техническое оборудование, то как правило, фирма заключает контракт с официальным поставщиком по оптимальным для себя условиям и заказывает оборудование, оплачивая все необходимое. Поставщики в ответ отправляют необходимое оборудование, закупленное у производителя.

Для примера приведем IT-компании по разработке программного обеспечения на территорию Томска:

* IT-компания “Веб-Киборг” – разработка интернет магазинов, сайтов, техническая поддержка имеющихся проектов [4].
* IT-компания “Sibedge” – разработка программного обеспечения для промышленного производства [5].
* IT-компания “alex-group” - создании бизнес сайтов, продвижении любых интернет-проектов обслуживание и поддержку сайтов [6].
* IT-компания “АВТОСНАБ - 70” – прокладывание и настройка локальной сети, настройка облачных сервисов, кибер-безопасность и зашита данных, удаление вирусов и шпионского ПО [7].

Целью процесса разработки программного обеспечения в IT-компании является выполнение заказа клиента в поставленные сроки.

Предназначением процесса разработки программного обеспечения является результат необходимый заказчику, включая в себя требования заказчика, качество выполненной работы, как необходимо эксплуатировать и обслуживать ПО в дальнейшем, и положительный имидж и хорошая репутацию сервиса у клиентов.

Следовательно, показателями эффективности разработки проекта будут:

* Количество выполненных заказов на разработку ПО
* Время разработки ПО
* Качественно было выполнение работы
* Размер полученной выручки за год

Все это отражается на этапах разработки программного обеспечения.

Этапы разработки программного обеспечения:

1. Сбор требований и их анализ.

Самый ответственный, важный этап, от которого зависит успех программного обеспечения. Специалисты собирают первичные данные, что позволяет создать основу. Параллельно анализируются риски, связанные с проектом. Это определяет возможность использования разных технических подходов, в основе которых лежит минимизации финансовых расходов.

Специалисты создают макеты и прототипы, определяют требования к проекту. Следующим шагом данного этапа является документирование требований со стороны клиента. Это дает полную правовую защищенность обеих сторон - разработчика и заказчика. В документе прописываются требования, которым должен соответствовать софт.

Чаще всего для этого разрабатывается SRS (Software Requirement Specification) – документ, в котором содержатся основные требования, которые предъявляются к программному продукту. Разработчикам важно точно выявить желание клиента, определить сроки разработки проекта. Здесь главная проблема - многостраничный список требований. Для их решения необходимо тесное взаимодействие с заказчиком, акцент на высокоуровневых требованиях.

1. Разработка дизайна программного продукта.

Создаваемый софт должен быть не только функциональным, но также удобным, понятным для пользователя. Для этого требуется правильно разработать архитектуру, способ представления программы, его пользовательский интерфейс, графическое решение. Особое внимание нужно уделить дизайну, где ориентируются на Software Requirement Specification.

SRS – документ, в котором закрепляется перечень требований и свойств, которые предъявляются к правильной, корректной работе программы. Разработчиком и дизайнером нужно понять, в какой форме должен быть представлен продукт. Сделать это непросто. Сам заказчик зачастую не знает этого, полагаясь на опыт, квалификацию программистов.

Для этого обычно каждый из разработчиков предлагает свой подход. После все документируется в Design Document Specification. Далее информация анализируется, выявляются требования и связи архитектурного модуля продукта с внешними модулями. Чтобы добиться успеха, важно иметь в команде грамотных лидеров, способных предложить оптимальную архитектуру на основе опыта выполнения аналогичных проектов.

1. Создание прототипа модели.

На этом этапе создается реальный прототип для поддержки знаний, полученных в процессе быстрого проектирования. Это небольшая низкоуровневая рабочая модель желаемой системы.

1. Разработка кода программного обеспечения.

После того, как разобрались с архитектурой, согласовали функционал, дизайн и концепцию, приступают к разработке софта. Пишут программный код, выполняют сборку, последовательно создают необходимые модули и фичи согласно утвержденному DDS. Практика доказывает, что чем более четкими являются требования в Design Document Specification, тем лучше происходит имплементация.

1. Тестирование полученного продукта.

После разработки софта, специалисты приступают к тестированию. Этот процесс затрагивает все этапы жизненного цикла. Все баги и недочеты фиксируются, регистрируются и отслеживаются. Недочеты исправляются, программный продукт тестируется заново. Это процесс происходит до тех пор, пока готовый информационный продукт не достигнет тех стандартов качества, которые прописаны в SRS.

Здесь начинает активно действовать команда автоматизаторов, тестировщиков. Главная сложность этого этапа – время, необходимое на выявление причин багов. Поиск ошибок в коде - сложная задача. Тестирование лучше проводить параллельно с разработкой. Это позволит не возвращаться к ним после запуска ПО.

1. Внедрение и поддержка.

После устранения всех багов, ПО выходит в релиз. Начинается поэтапное внедрение программы согласно выбранной бизнес-стратегии. Изначально софт может быть выпущен в ограниченном сегменте, протестирован в конкретной бизнес-среде. Для этого выполняют тестирование User Acceptance Testing. В его основе - получение реальных отзывов со стороны.

Это позволяет проанализировать обратную связь, увидеть недочеты, произвести улучшение продукта. В дальнейшем, после выпуска продукта на рынок, к работе подключается команда специалистов поддержки.

Количество услуг будет формироваться в том случае, если клиент доверяет организации, в которую обратился и в нашем случае, что на каждом этапе разработке все будет идти в указанные сроки. Но из-за нехватки на рынке труба высоквалифицированных специалистов время разработки будет увеличиваться из чего получаем потерю клиентов, которые перестают финансировать компанию.

Из выше сказанного, можно выделить проблему несоответствия планируемых сроков и бюджета разработки программного обеспечения фактическим.

**2 МОДЕЛИРОВАНИЕ СИСТЕМЫ**

## **2.1 Модель взаимосвязи системы с окружающей средой**

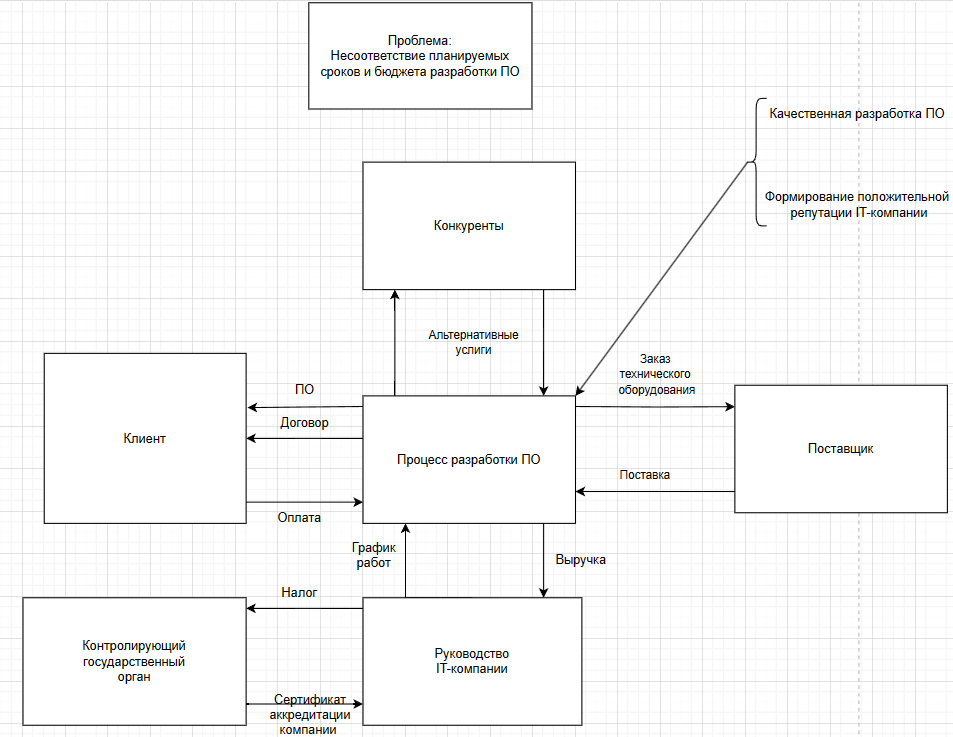
Для построения модели взаимосвязи системы с окружающей средой определим наименования элементов, входящих в неё:

* Процесс разработки программного обеспечения в IT-компании –

исследуемый процесс, представлен в данной модели как «чёрный ящик»;

* Клиенты – потребители услуг, источник прибыли фирмы;
* Поставщики – организации, которые поставляют фирме необходимое оборудованием;
* Руководство – основываясь на предпочтениях клиента, осуществляется управление компанией, устанавливает сроки выполнения работы и т.д.;
* Конкурент – различные IT-компании в том-же населенном пункте, специализирующиеся на разработке программного обеспечения;
* Контролирующий государственный орган – Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации, обеспечивающий соблюдение фирмой российского законодательства;

Построенная диаграмма взаимосвязи исследуемой системы с окружающей средой представлена на рисунке 2.1.

Рисунок 2.1 - диаграмма взаимосвязи системы с окружающей средой

Построенная модель позволяет выявить заинтересованные стороны анализируемой системы. Таковыми являются непосредственно клиенты и сотрудники IT-компании.

Клиенты являются основным потребителем услуг компании. С точки зрения поставленной проблемы, они могут влиять на качество разрабатываемого продукта несколькими способами. Большое количество клиентов, а, как следствие, большое количество поступающих заявок на разработку ПО, может увеличить срок ожидания их выполнения, т.к. каждая разработка занимает определённое время, вследствие чего может возникнуть очередь. Также на разработку влияет сложность функциональных требований продукта, иначе говоря, его масштабы решения каких-либо задач.

Сотрудники компании занимаются разработкой, непосредственно выполняют функциональные и нефункциональные требования, обеспечивая надлежащее качество ПО. Скорость выполнения работ зависит от сложности поставленных требований клиента, квалификации самих разработчиков.

Критерии, по которым каждый может оценить уровень системы в сравнении с его представлением об «идеале», а также соответствующие им системы оценивания приведены в таблице 2.1.

Таблица 2.1 – Критерии оценивания системы

|  |  |  |
| --- | --- | --- |
| Критерий | Ед. изм. и/или возможные значения | Система оценивания |
| 1. Клиент  1.1 Время затраченное на разработку ПО | Месяц | 1-2 – «отлично»;  3-4 – «хорошо»;  4-5 – «удовлетворительно»;  более 6 – «неудовлетворительно». |
| 1.2 Качество выполнения программного продукта | Балл,  Удовлетворительное – неудовлетворительное | 6 – 10 – «удовлетворительное»;  менее 6 баллов – «неудовлетворительное». |
| 2. Сотрудник компании  2.1 Размер заработной платы | Тыс. руб. | 60-80 – «неудовлетворительный»;  80-100 – «приемлемый»;  более 100 – «отличный». |
| 2.2 Уровень сложности разработки ПО | Балл | 1-5 – «высокий»  6-8 – «средний»  9-10 – «низкий» |

**2.2 Модели состава и структуры системы**

Для построения модели состава системы выполним декомпозицию системы по двум основаниям.

Первым основанием является жизненный цикл разработки ПО. В данном случае рассматриваются все её этапы – от приёма заявки клиента на разработку ПО до его передачи клиенту. Далее каждый этап жизненного цикла декомпозируется по технологическим этапам.

Жизненный цикл разработки ПО состоит из трех последовательных этапов: приёма заявки клиента на разработку ПО, решение поставленных задач, эксплуатация ПО.

Этап приёма заявки клиента на разработку ПО состоит из процедур формирование команды и постановки задач. Основываясь на тех функциональных требованиях, которые хочет получить клиент, выбираем специалистов для их реализации. Как только мы сформировали команду необходимо распределить и поставить задачи. Данный этап может оказывать сильное влияние на поставленную проблему, т.к. контроль качества разработки ПО лежит исключительно на разработчиках и есть вероятность, что квалификация специалистов напрямую влияет на качество ПО.

На этапе решения поставленных задач пишется программный код, далее проводится тестирование всех функциональных требований, чтобы перейти к нефункциональным. Главная сложность этого этапа – время, необходимое на выявление причин багов. Поиск ошибок в коде - сложная задача. После тестирование проводим отладку полученных ошибок и багов, чтобы передать клиенту полностью работающий продукт.

Теперь, когда ПО полностью выполняет функционал его нужно запустить в работу, оценить качество проделанной работы разработчиков. То есть необходимо связаться с клиентом для передачи ПО. На данном этапе может представлять затруднение в случаях, когда не получается вовремя выйти с клиентом на связь и передать ему ПО.

Также, поскольку рассматривается IT-компания типа разработки ПО, уровень специалистов можно принять за одинаково высокий и убрать вышеописанные возможные негативные факторы на этапе решения поставленных задач.

Соответствующая графическая интерпретация модели состава представлена на рисунке 2.2.

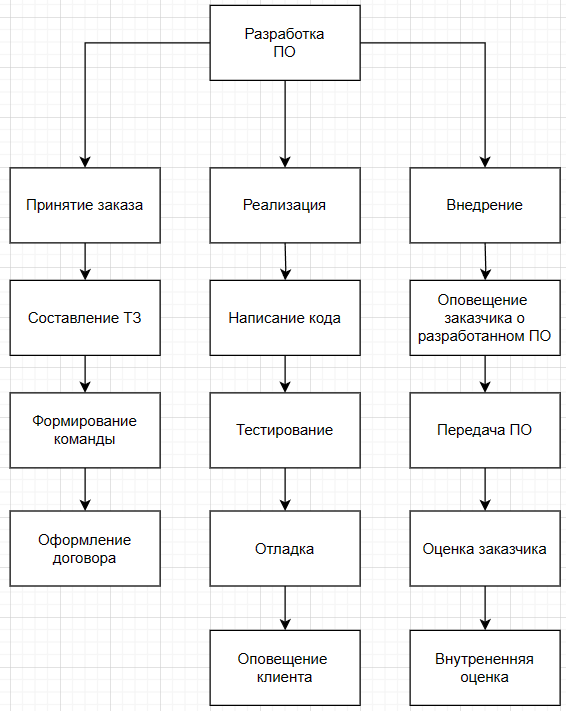


Рисунок 2.2 – Графическая интерпретация модели состава

В таблице 2.2 приведены структурные элементы деятельности рассматриваемой системы.

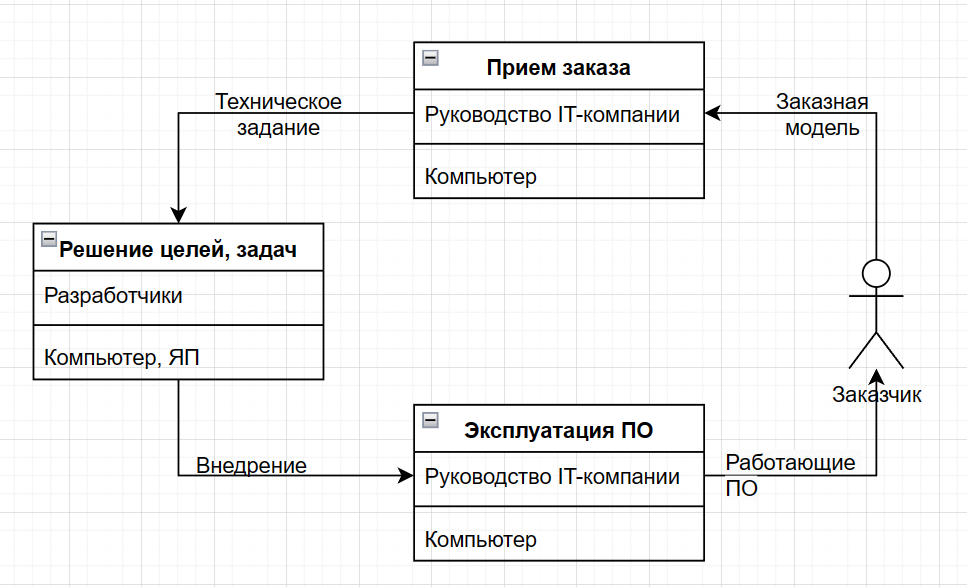
Таблица 2.2 – Структурные элементы системы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Подсистема | Предмет деятельности | Средство исполнения | Исполнитель | Регламент | Конечный продукт |
| Принятие заказа | Заказная модель | Компьютер, MS Word | Исполнитель | Предоставляемые  услуги | Договор о разработке ПО |
| Реализация | Написание программы | Компьютер, ЯП | Разработчики | Функциональные требования | Разработанное ПО |

Окончание таблицы 2.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Подсистема | Предмет деятельности | Средство исполнения | Исполнитель | Регламент | Конечный продукт |
| Внедрение | Разработанное ПО | Компьютер | Разработчики | Описание работы ПО | Внедренное ПО, результаты оценки качество |

Далее на рисунке 2.3 приведена диаграмма, отражающая связи между подсистемами, полученными в результате декомпозиции системы по жизненному циклу. Соответственно, на рисунках 2.4, 2.5, 2.6 приведены аналогичные диаграммы в результате декомпозиции каждого из этапов второго уровня декомпозиции.

Рисунок 2.3 – Диаграмма взаимосвязи подсистем

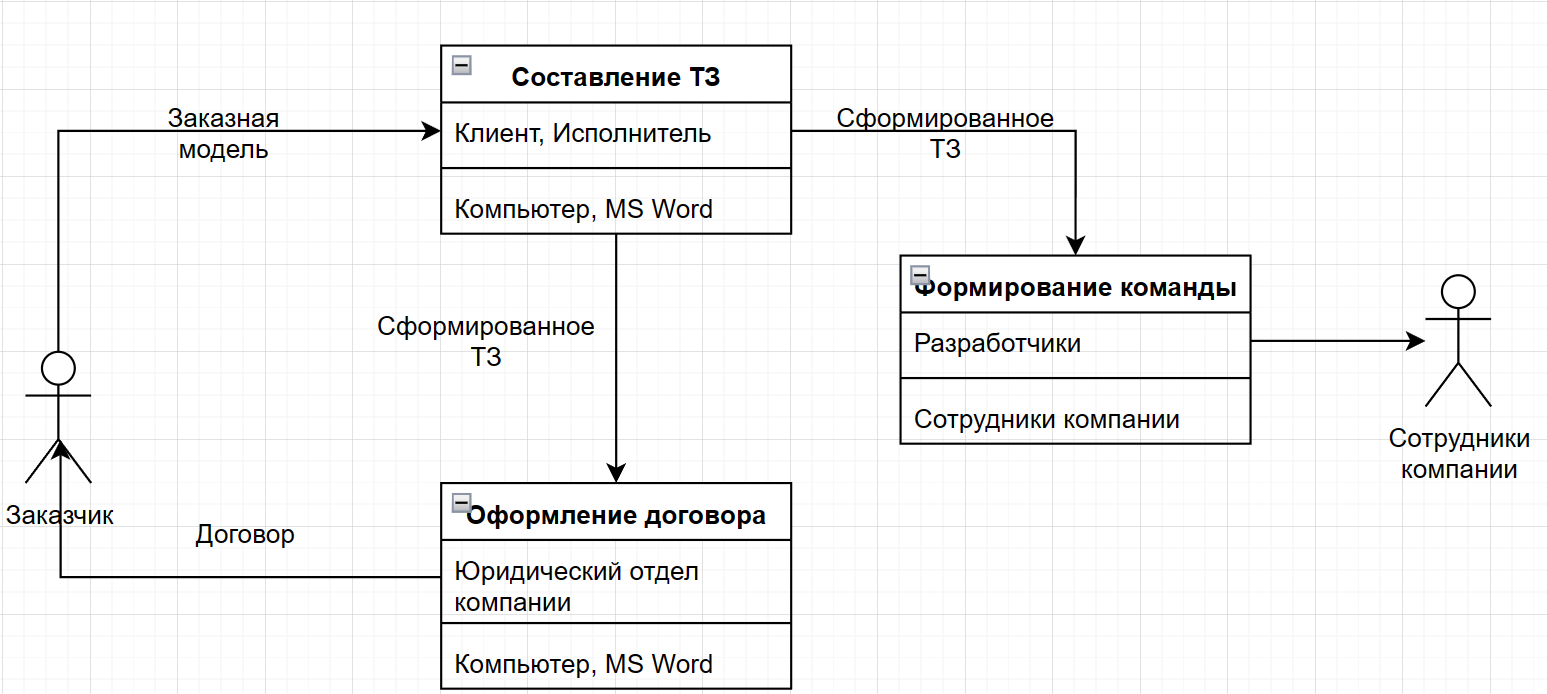


Рисунок 2.4 – Диаграмма взаимосвязи подсистемы «Принятие заказа»

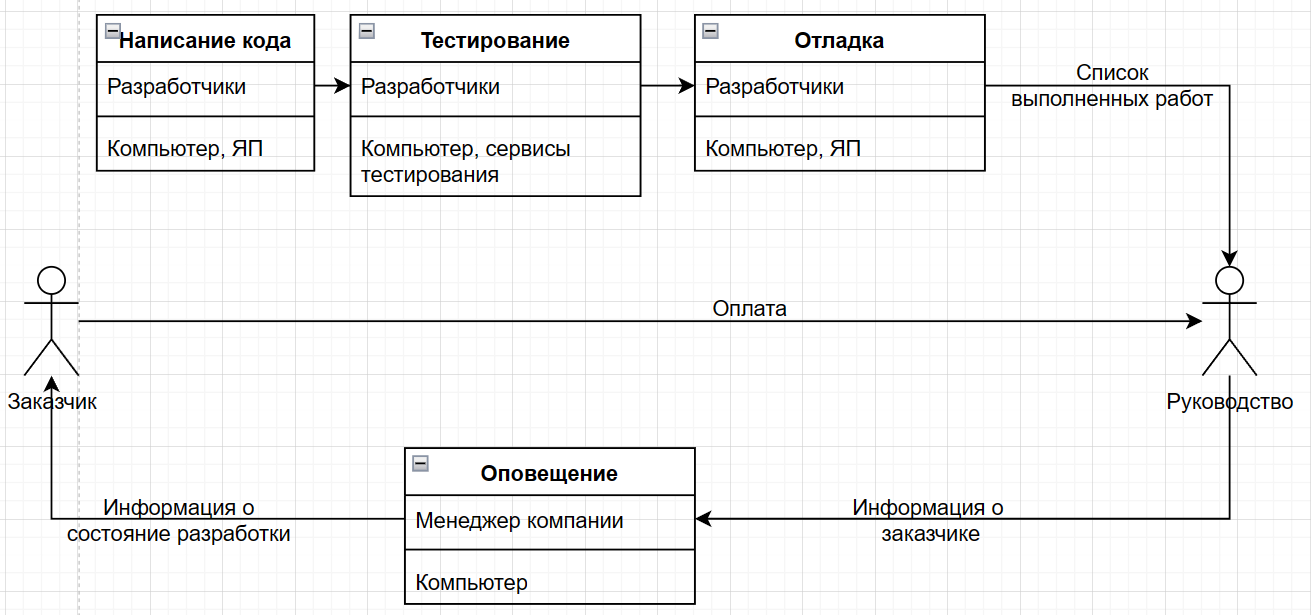


Рисунок 2.5 – Диаграмма взаимосвязи подсистемы «Решение целей, задач»

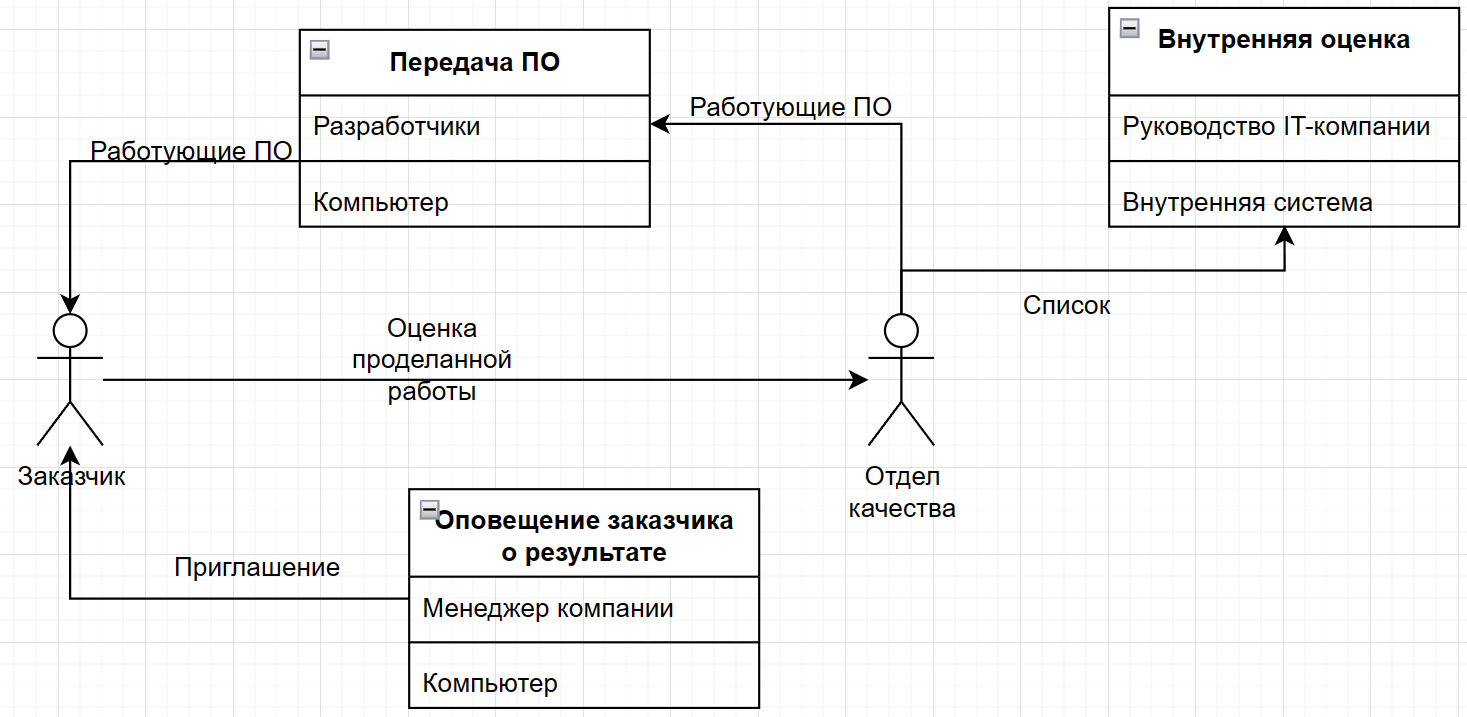


Рисунок 2.6 – Диаграмма взаимосвязи подсистемы «Внедрение ПО»

**2.3 Построение дерева причин**

Следующим этапом работы является выявление возможных причин проблемы длительного срока ожидания клиентами выполнения услуг.

В ходе анализа состояния системы и среды были выявлены следующая «подпроблема»:большой объем услуг, который задерживается из-за невыполнения поставленных в срок задач.

В свою очередь, кадровый дефицит может быть вызван в результате наплыва низкоквалифицированных специалистов. Дальнейшая проработка проблемы привела к следующим выводам:

* Некачественная коммуникация между IT-командой и клиентом. Например, клиент может долго не давать обратную связь, избыточно контролировать команду, не понимать цели проекта и постоянно менять требования.
* Неполная команда разработки на старте. Когда на проекте не хватает рук, приходится чем-то жертвовать, чтобы уложиться в сроки.
* Нехватка у IT-команды опыта на конкретном проекте. Если команда работает с чем-то новым, бывает сложно учесть все нюансы.

Тем самым влечет за собой большой кадровый дефицит специалистов, которые не в силах выполнять поставленные задачи. Анализ проблемы привел к таким результатам:

* Компании не предоставляют повышение квалификации, не проводят внутренние хакатоны, так же, заведомо, не отправляя на другие проводимые от других компаний или иных организаций.
* Компании нанимают заведомо некомпетентных людей. В целях привлечения компетентных кадров растут зарплаты.
* Требуется подготовка специалистов, обладающих современными знаниями и практическими навыками аналитических, статистических и эконометрических методов исследования, анализа социально-экономических явлений и процессов с применением цифровых технологий.

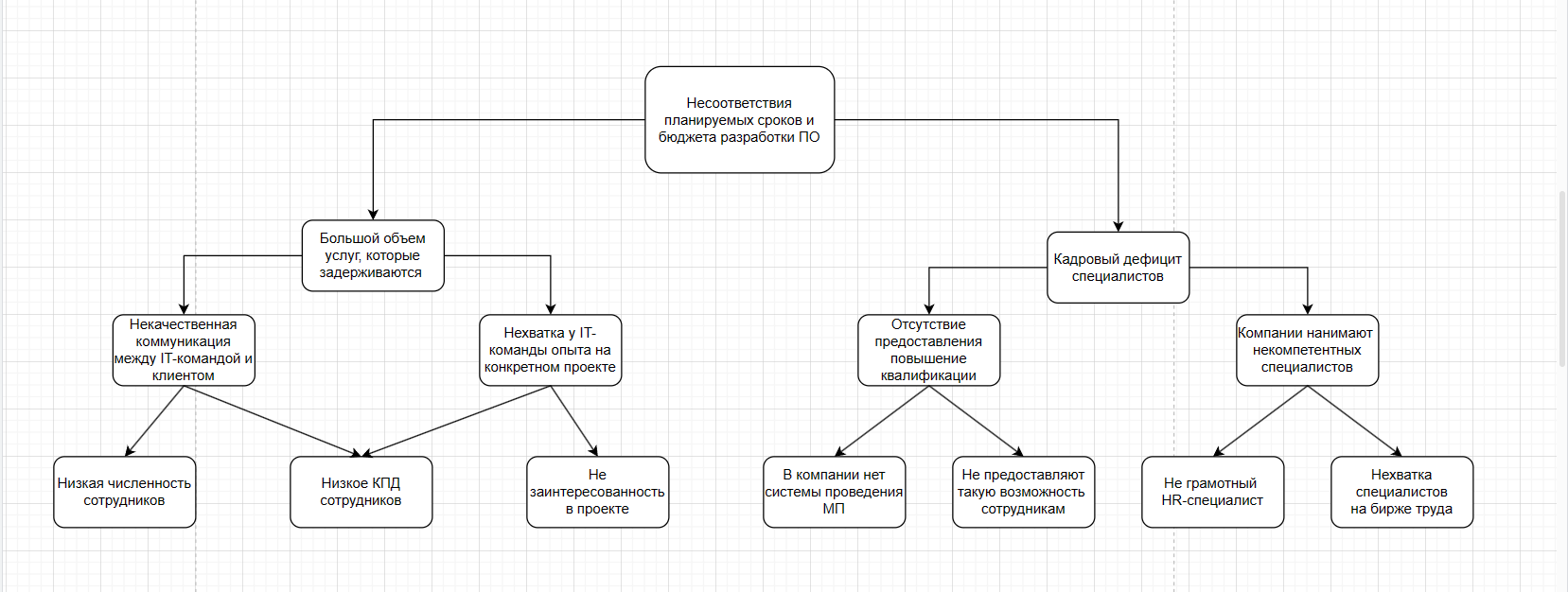
Построенное дерево причин изображено на рисунке 2.7.

Рисунок 2.7 – Дерево причин

Для оценки важности причин воспользуемся методом ранжирования. Результаты оценивания приведены в таблице 2.3.

Таблица 2.3 – Оценивание важности причин методом ранжирования

|  |  |
| --- | --- |
| Причина | Ранг |
| Низкая численность сотрудников | 5 |
| Низкое КПД сотрудников | 2 |
| Не заинтересованность в проекте | 7 |
| В компании нет системы проведения МП | 4 |
| Не предоставляют такую возможность сотрудникам | 3 |
| Не грамотный HR-специалист | 6 |
| Нехватка специалистов на бирже труда | 1 |

**3 ПОСТАНОВКА ЦЕЛЕЙ И ПОИСК РЕШЕНИЙ**

**3.1 Построение дерева целей**

Для построения дерева целей возьмём за основу дерево причин. Глобальной целью является разработать качественное программное обеспечение. Этого можно добиться двумя способами – сократить объём очереди разработки ПО и повысить квалификацию специалистов.

Сокращения большого объем услуг, который задерживается из-за невыполнения поставленных в срок задач можно добиться путём улучшения организации построения самой очереди. Или же можно повысить общую производительность сотрудников, чтобы за одно и то же время выполнялось больше задач на определённом этапе разработки.

Повышение производительности может заключаться в создании системы набора и подготовки персонала, либо в создании системы мотивации персонала, чтобы у работников появилась личная заинтересованность в повышении эффективности процесса.

Улучшение организации очереди выполняемых услуг будет возможно, если создать такую систему, которая будет заниматься учётом и отслеживанием сроков выполнения оказываемых услуг. Тем самым, эта задача не накладывает на разработчиков всю ответственность на выполненные услуги, за все будет полностью отвечать автоматизированная система, которая будет показывать, какая из выполняемых задач сейчас в приоритете по времени. Следовательно, разработка ПО будет проводиться в срок. Еще подходит система подготовки сотрудников, т.к. при условии слаженности коллектива обеспечивается лучшее распределение выполняемых задач между сотрудниками компании – знание сильных и слабых сторон коллег позволяет распределять этапы разработки так, чтобы эффективность работы каждого была максимальна.

Отметим, что имея в компании опытных и квалифицированных специалистов, время на этапах разработки сокращается, приводя к выполнению проекта в поставленные сроки.

Согласно дереву причин, длительная разработка вызвана низкой эффективностью взаимодействия с клиентом либо отсутствием опытных специалистов. Соответственно, есть вариант нахождения альтернативных специалистов, которые будут иметь больше опыта, чтобы за поставленные временные рамки разработать ПО.

Увеличение квалифицированных сотрудников. Данный вариант может повлечь определённые расходы, т.к. разработкой будут заниматься сотрудники, привлеченные их других компаний. Также здесь подходит вариант поиска альтернативных компаний, которые могут предоставлять свои услуги в более короткий период и с заранее оговорённым сроком.

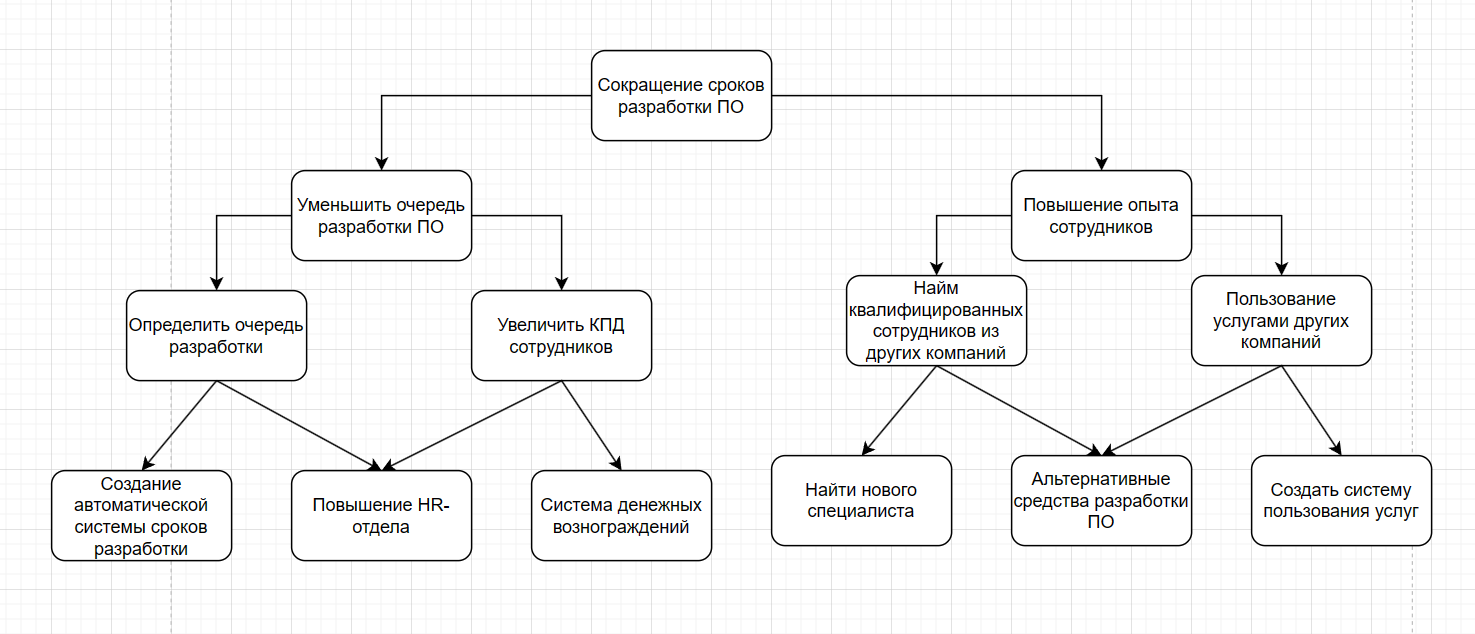
Построенное по вышеописанной логике дерево целей представлено на рисунке 3.1.

Рисунок 3.1 – Дерево целей

**3.2 Оценка целей методом анализа иерархий**

3.2.1. Описание метода анализа иерархий.

Метод анализа иерархий (МАИ), предложенный Томасом Саати, использует методологию дерева целей, т.е. также основан на формировании иерархии целей и средств по типу слоёв. Данный метод предназначен для выбора средств решения сложной многофакторной проблемы и состоит в декомпозиции цели на всё более простые составляющие [8].

Метод состоит из пяти основных этапов:

1. иерархическое представление проблемы;
2. построение множества матриц парных сравнений;
3. определение векторов локальных приоритетов;
4. проверка согласованности полученных результатов;
5. вычисление глобальных приоритетов.

Этап иерархического представления на данный момент уже пройден – дерево целей построено. Для последующих шагов кратко пронумеруем каждый элемент иерархии. Результат представлен на рисунке 3.1.



Рисунок 3.1 – Иерархия целей

Следующим этапом идёт построение матриц парных сравнений для каждой совокупности элементов, связанных с одним вышестоящим элементом. Для полученного дерева целей потребуется построить: одну матрицу, соответствующую второму уровню иерархии, для сравнения двух основных направлений возможного решения проблемы; две матрицы, соответствующие третьему уровню, для детального сравнения каждого из направлений в отдельности, и четыре матрицы, соответствующие четвёртому

уровню иерархии, для оценки влияния сценариев развития сервиса на каждое из «поднаправлений». При оценивании каждой сравниваемой паре ставится балльная оценка от 1 до 9 в зависимости от того, какой элемент пары доминирующий.

На основании каждой матрицы парных сравнений формируются наборы локальных приоритетов. Они отражают ценность, важность, силу влияния сравниваемых элементов по отношению к направляющему элементу – элементу вышестоящего уровня. Чтобы вычислить эти приоритеты, нужно для каждой матрицы вычислить множество собственных векторов и нормализовать результат к единице – так получится вектор приоритетов. Так

как задача вычисления собственных векторов довольно трудоёмкая, мы воспользуемся приближённым методом – вычислением геометрического среднего.

Затем наступает этап проверки согласованности полученных результатов. Матрицы парных сравнений проверяются на согласованность, вычисляется индекс согласованности (ИС) по формуле 3.1:

ИС = (3.1)

где n – размерность матрицы, наибольшее собственное значение матрицы. Для проверки согласованности таким путём, нужно сравнить ИС с индексом, вычисленным для абсолютно несогласованной матрице, полученной при случайном выборе суждений. Для получения конечного результата ИС нужно разделить на число, соответствующее случайной согласованности матрицы того же порядка – получится отношение согласованности (ОС).

Финальный шаг – вычисление глобальных приоритетов. Они рассчитываются, начиная со второго уровня вниз. Локальные приоритеты элементов умножаются на глобальные приоритеты элементов вышестоящего уровня - направляемых. Если направляемых несколько, то находится сумма взвешенных приоритетов по всем направляемым элементам. Процедура продолжается до нижнего уровня.

3.2.2. Построение матриц парных сравнений и расчёт локальных приоритетов.

Сравнив полученные элементы иерархии целей методом попарных сравнений, были рассчитаны локальные приоритеты для каждого из элементов иерархии относительно каждого родительского (направляющего элемента). Индекс согласованности и отношение согласованности были также рассчитаны. Результаты приведены в таблицах 3.1 – 3.7.

Таблица 3.1 – Матрица парных сравнений для цели «Сокращение сроков разработки ПО»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Уменьшить очередь  разработки ПО | Повышение опыта  сотрудников | Локальный приоритет |
| Уменьшить очередь  разработки ПО | 1,00 | 0,20 | 0,167 |

Окончание таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
|  | Уменьшить очередь  разработки ПО | Повышение опыта  сотрудников | Локальный приоритет |
| Повышение опыта  Сотрудников | 5,00 | 1,00 | 0,833 |
| Индекс согласованности | | | 0,00 |
| Отношение согласованности | | | 0,00 |

Таблица 3.2 – Матрица парных сравнений для цели «Уменьшить очередь

разработки ПО»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Определить очередь разработки | Увеличить КПД  сотрудников | Локальный приоритет |
| Определить очередь разработки | 1,00 | 0,50 | 0,333 |
| Увеличить КПД  сотрудников | 2,00 | 1,00 | 0,667 |
| Индекс согласованности | | | 0,00 |
| Отношение согласованности | | | 0,00 |

Таблица 3.3 – Матрица парных сравнений для цели «Повышение опыта

сотрудников»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Найм квалифицированных сотрудников из других компаний | Пользование услугами других компаний | Локальный приоритет |
| Найм квалифицированных сотрудников из других компаний | 1,00 | 9,00 | 0,9 |
| Пользование услугами других компаний | 0,11 | 1,00 | 0,1 |
| Индекс согласованности | | | 0,00 |
| Отношение согласованности | | | 0,00 |

Таблица 3.4 – Матрица парных сравнений для цели «Определить очередь разработки»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Создание автоматической системы сроков разработки | Повышение HR-отдела | Локальный приоритет |
| Создание автоматической системы сроков разработки | 1,00 | 0,17 | 0,143 |
| Повышение HR-отдела | 6,00 | 1,00 | 0,857 |
| Индекс согласованности | | | 0,00 |
| Отношение согласованности | | | 0,00 |

Таблица 3.5 – Матрица парных сравнений для цели «Увеличить КПД

сотрудников»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Повышение HR-отдела | Система денежных  вознаграждений | Локальный приоритет |
| Повышение HR-отдела | 1,00 | 0,14 | 0,125 |
| Система денежных  вознаграждений | 7,00 | 1,00 | 0,875 |
| Индекс согласованности | | | 0,00 |
| Отношение согласованности | | | 0,00 |

Таблица 3.6 – Матрица парных сравнений для цели «Найм квалифицированных сотрудников из других компаний»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Найти нового специалиста | Альтернативные средства разработки ПО | Локальный приоритет |
| Найти нового специалиста | 1,00 | 9,00 | 0,9 |
| Альтернативные средства разработки ПО | 0,11 | 1,00 | 0,1 |
| Индекс согласованности | | | 0,00 |
| Отношение согласованности | | | 0,00 |

Таблица 3.7 – Матрица парных сравнений для цели «Пользование услугами других компаний»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Альтернативные средства разработки ПО | Создать систему пользования услуг | Локальный приоритет |
| Альтернативные средства разработки ПО | 1,00 | 6,00 | 0,857 |
| Создать систему пользования услуг | 1,67 | 1,00 | 0,143 |
| Индекс согласованности | | | 0,00 |
| Отношение согласованности | | | 0,00 |

3.2.3. Расчет глобальных приоритетов.

Рассчитаем глобальные приоритеты всех элементов иерархии согласно

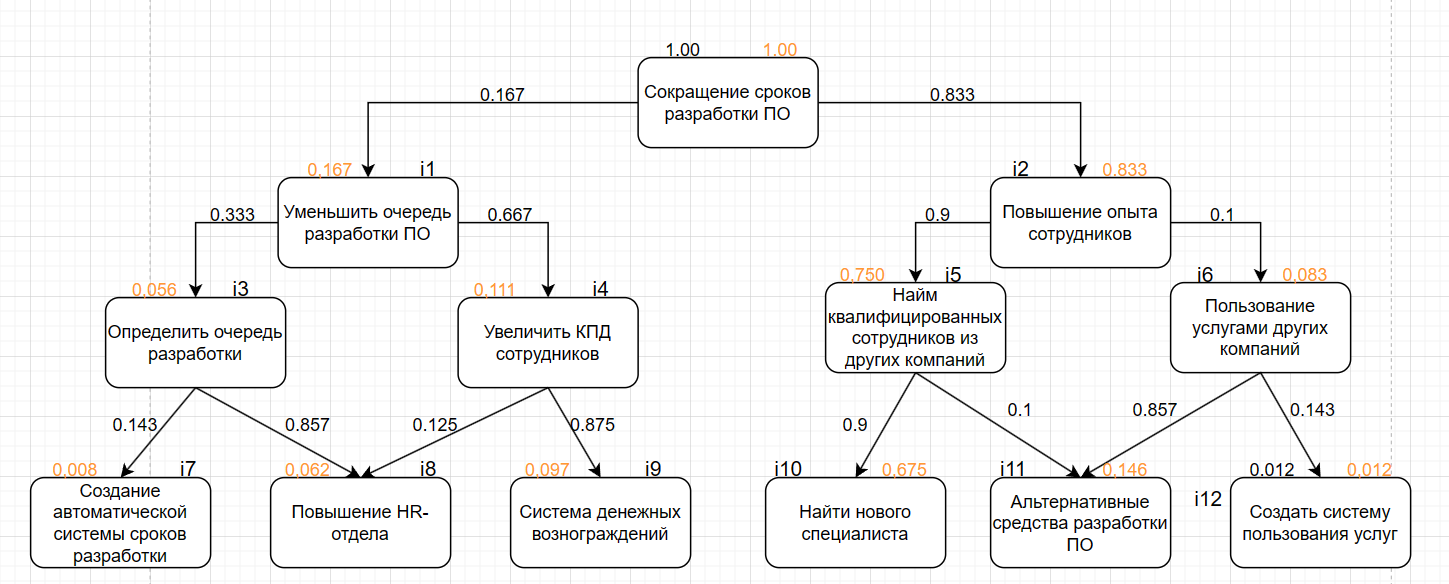
правилам, описанным в пункте 3.2.1. Результаты представлены в таблице 3.8.

Таблица 3.8 – Значения глобальных приоритетов

|  |  |  |
| --- | --- | --- |
| Уровень иерархии | Наименование элемента иерархии | Глобальный приоритет |
| 1 | Сокращение сроков разработки ПО | 1,00 |
| 2 | Уменьшить очередь разработки ПО | 0,167 |
| 2 | Повышение опыта сотрудников | 0,833 |
| 3 | Определить очередь разработки | 0,056 |
| 3 | Увеличить КПД сотрудников | 0,111 |
| 3 | Найм квалифицированных сотрудников из других компаний | 0,750 |
| 3 | Пользование услугами других компаний | 0,083 |
| 4 | Создание автоматической системы сроков разработки | 0,008 |
| 4 | Повышение HR-отдела | 0,062 |
| 4 | Система денежных вознаграждений | 0,097 |
| 4 | Найти нового специалиста | 0,675 |
| 4 | Альтернативные средства разработки ПО | 0,146 |
| 4 | Создать систему пользования услуг | 0,012 |

В результате совершённых действий можно установить, что для устранения исходной проблемной ситуации следует двигаться в направлении повышения опыта сотрудников путем найма, квалифицированных специалистом. Его и выберем в качестве приоритетного.

Для простоты понимания отобразим все полученные приоритеты на иерархии целей, как показано на рисунке 3.2. Глобальные приоритеты обозначены чёрным цветом над каждым элементом.

Рисунок 3.2 – Результат выявления приоритетов

**Заключение**

В ходе выполнения практической работы мной были получены навыки разработки дерева причин и целей, основанные на проблеме несоответствия планируемых сроков и бюджета разработки программного обеспечения фактическим, которая оценивается анализом иерархий.

**Список использованных источников**

1. Число IT-компаний в России [Электронный ресурс]: РИА Новости. URL: <https://ria.ru/20230614/kompanii-1877930925.html> (дата обращения: 01.11.2023);
2. Афендульева Д.Ю. Анализ конкурентоспособности предприятия, основанный на компьютерных технологиях искусственного интеллекта. Механико-математические науки: докл. Саратовская науч.-практ. конф. Саратов 2017. С. 4-7;
3. Аккредитация ИТ-компаний [Электронный ресурс]: Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации. URL: <https://digital.gov.ru/ru/activity/govservices/1/> (дата обращения 01.11.2023);
4. Веб-студия ВЕБ-Киборг – веб-разработка для решения задач бизнеса [Электронный ресурс]: официальный сайт IT-компании Веб-Киборг. URL: <https://web-kiborg.ru/> (дата обращения 04.11.2023);
5. Разработка ПО для цифровой трансформации бизнеса [Электронный ресурс]: официальный сайт IT-компании Sibedge. URL: <https://sibedge.team> (дата обращения 04.11.2023);
6. Создание сайтов в Томске [Электронный ресурс]: официальный сайт IT-компании Alex-group. URL: <https://ag70.ru> (дата обращения 04.11.2023);
7. ООО "АВТОСНАБ-70" [Электронный ресурс]: официальный сайт IT-компании Autosnab-70. URL: <https://autosnab70.ru> (дата обращения 04.11.2023);
8. Метод анализа иерархий Т. Саати [Электронный ресурс]: онлайн ресурс википедии. URL: [https://en.wikipedia.org/wiki/Analytic\_hierarchy  
   \_process](https://en.wikipedia.org/wiki/Analytic_hierarchy_process) (дата обращения 23.11.2023);