



## ЭКОНОМИКА

УДК 004.9; 004.422

DOI: 10.18324/2224-1833-2024-3-9-21

### Управленческие аспекты поддержки автозаправочной деятельности с учетом особенности разработки требований к программной системе на примере АЗС «Тран-Ойл»

Х.Г. Ахмедова<sup>1а</sup>, Л.А. Геращенко<sup>1б</sup>, Е.И. Луковникова<sup>2с</sup>

<sup>1</sup> МИРЭА – Российский технологический университет, пр. Вернадского, 78, Москва, Россия

<sup>2</sup> Братский государственный университет, ул. Макаренко, 40, Братск, Россия

<sup>а</sup> h.ahmedova@mail.ru, <sup>б</sup> gerashsenko@mail.ru, <sup>с</sup> lena\_lukovnikova@mail.ru

<sup>а</sup> <https://orcid.org/0000-0003-2442-9955>, <sup>б</sup> <https://orcid.org/0000-0002-1766-2735>,

<sup>с</sup> <https://orcid.org/0009-0009-7294-053X>

Статья поступила 01.11.2024, принята 21.11.2024

*Известно, что любая коммерческая организации стремится получить как можно больше прибыли, следовательно, сокращение времени обслуживания одного клиента приведет к увеличению потока клиентов, а это, в свою очередь, – к увеличению прибыли компании и, как следствие, повышению конкурентоспособности. В связи с этим целью исследования является разработка требований к программной системе поддержки автозаправочной деятельности АЗС «Тран-Ойл», способной сократить время обслуживания одного клиента. Одной из важнейших стадий разработки программных систем является разработка требований. Допущенные на этой стадии ошибки очень дорого обходятся после разработки и внедрения системы. Несмотря на огромный опыт, накопленный в области разработки программных продуктов, проблемы, связанные с разработкой требований, остаются все еще нерешенными. Целью исследования является разработка требований к программной системе поддержки автозаправочной деятельности АЗС «Тран-Ойл», способной сократить время обслуживания одного клиента. Требования разрабатываются поэтапно, включая все мероприятия, необходимые для создания и утверждения документа, содержащего спецификацию системных требований. Различают четыре основных этапа процесса разработки требований: выявление требований; формирование и анализ требований; документирование требований; утверждение (аттестация) требований. В исследовании применялись разные теории и методологии – теория систем, теория информации, системный подход, системный анализ, методы структурно-функционального моделирования, методы объектно-ориентированного моделирования, эмпирические методы (интервью, анкетирование, совещания, анализ документов, прототипирование) и др. На основе проведенного исследования созданы спецификации требований к программной системе на разработку мобильного приложения. Исследование проводилось с привлечением конечных пользователей продукта. Построенные модели, определенные требования и реализованные в работе идеи могут быть использованы бизнес-аналитиками и разработчиками при создании собственных программных систем.*

**Ключевые слова:** бизнес-цели; бизнес-требования; пользовательские требования; функциональные требования; спецификации требований; мобильное приложение; модели анализа.

### Management aspects of supporting gas station activities taking into account the specifics of developing requirements for a software system using the example of TRAN-OIL gas station

Kh.G. Akhmedova<sup>1а</sup>, L.A. Gerashchenko<sup>1б</sup>, E.I. Lukovnikova<sup>2с</sup>

<sup>1</sup> MIREA – Russian Technological University; 78, Vernadsky Ave., Moscow, Russia

<sup>2</sup> Bratsk State University; 40, Makarenko St., Bratsk, Russia

<sup>a</sup> h.ahmedova@mail.ru, <sup>b</sup> gerashsenko@mail.ru, <sup>c</sup> lena\_lukovnikova@mail.ru

<sup>a</sup> <https://orcid.org/0000-0003-2442-9955>, <sup>b</sup> <https://orcid.org/0000-0002-1766-2735>,

<sup>c</sup> <https://orcid.org/0009-0009-7294-053X>

Received 01.11.2024, accepted 21.11.2024

*It is known that any commercial organization strives to get as much profit as possible, therefore, reducing the time spent servicing one client will lead to an increase in the flow of clients, and this, in turn, will increase the company's profit and, as a result, increase competitiveness. In this regard, the purpose of the study is to develop requirements for a software system to support the refueling activities of TRAN-OIL gas stations, capable of reducing the time of servicing one client. One of the most important stages in the development of software systems is the development of requirements. Mistakes made at this stage are very costly after the system is developed and implemented. Despite the vast experience accumulated in the field of software product development, problems associated with requirements development still remain unresolved. The purpose of the study is to develop requirements for a software system to support the refueling activities of TRAN-OIL gas stations, capable of reducing the time of servicing one client. Requirements are developed in stages, including all activities necessary to create and approve a document containing a system requirements specification. There are four main stages in the requirements development process: requirements elicitation; formation and analysis of requirements; documenting requirements; approval (certification) of requirements. The study uses different theories and methodologies - systems theory, information theory, systems approach, system analysis, structural-functional modeling methods, object-oriented modeling methods, empirical methods (interviews, questionnaires, meetings, document analysis, prototyping), etc. Based on the research, specifications of requirements for a software system for the development of a mobile application are created. The study is conducted with the participation of end users of the product. The constructed models, certain requirements and ideas implemented in the work can be used by business analysts and developers when creating their own software systems.*

**Keywords:** business goals; business requirements; user requirements; functional requirements; requirements specifications; mobile application; analysis models.

**Введение.** Важнейшим этапом разработки программных систем является обоснование и разработка требований. Допущенные в требованиях ошибки очень дорого обходятся после разработки системы. Можно великолепно написать программу, но если требования изначально были неполные, неоднозначные или неправильные, результат такой работы сильно разочарует пользователя. Требования к программной системе — это совокупность утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации. Требования нуждаются в тщательном анализе, поэтому в практике сложился и достаточно успешно используется системный подход [1].

Имеются три иерархических уровня требований: бизнес-требования; пользовательские требования; функциональные и нефункциональные требования [2].

Бизнес-требования включают высокоуровневую бизнес-цель организации, которую можно достичь с помощью разрабатываемой системы. На этом этапе определяется концепция продукта с заказчиком для выработки единого видения будущей программной системы.

Кроме текстового или табличного описания, для однозначного понимания, конкретизации (детализации) бизнес-требований, как правило, создают структурно-функциональные модели, используя разные методологии и нотации [3].

Пользовательские требования описывают задачи, которые могут выполнять разные классы пользователей в данной системе для достижения бизнес-цели. Пользовательские требования описываются с помощью пользовательских историй, а для конкретизации (детализации) и однозначности их понимания создают диаграммы вариантов ис-

пользования (*Use Case*) на языке UML [4].

Функциональные требования включают описание требуемого поведения системы в определенных условиях для решения задач пользователей. Для определения функциональных требований детализированные варианты использования (основной и альтернативные сценарии) подробно описываются в статических и динамических диаграммах UML. Это позволяет избежать неоднозначности требования [5].

Нефункциональные требования включают описание свойства или особенности, никак не влияющих на функции приложения, которыми должна обладать система или ограничение, которое должна соблюдать система. В сложных системах для описания ограничений, конкретизации атрибутов качества и описания внешних интерфейсов требуется статическое описание состава и структуры программных и аппаратных компонентов системы. Для этого используются диаграммы компонентов и размещения (развертывания) UML.

Качественно определенные требования гарантируют, что разрабатываемая система будет соответствовать ожиданиям клиента, позволит сэкономить бюджет и сократить сроки разработки.

Требования разрабатываются поэтапно, включая все мероприятия, необходимые для создания и утверждения документа, содержащего спецификацию системных требований. Различают четыре основных этапа процесса разработки требований: выявление требований; формирование и анализ требований; документирование требований; утверждение (аттестация) требований.

Для выявления требований используются разные методы — интервью, совещания, анализ документов, создание прототипов и др.

На этапе анализа требований создается окончательный список требований и производится их группировка. Здесь можно выявить и избавиться от дублирующихся, неоднозначных требований, полученных из пользовательских историй и вариантов использования, разложенных по сценариям. Для этого как нельзя кстати подходят диаграммы *Use Case* и классы анализа [6].

На этапе документирования требований, по сути, создается проект будущей системы. На этом этапе разработчики определяют, какую функциональность будущей системы необходимо реализовать, какие качественные характеристики, внешние интерфейсы и ограничения учесть, чтобы удовлетворить потребности пользователей. Итогом этого этапа становится разработанное техническое задание [7].

Крайне важно на основе технического задания создать модели работы системы (прототипы) с конечными пользователями [8]. Можно создать, например, макет пользовательского интерфейса и выполнить таким образом тестирование требований. Это позволит найти упущенные или повторяющиеся требования, увеличить качество продукта и снизить его стоимость. Легче и дешевле внести изменения в требования, чем в готовый продукт.

**Объекты и методы исследования.** Объектом исследования является автозаправочная станция «Тран-Ойл», предметом исследования — автозаправочная деятельность.

В работе используются следующие методы исследования:

- теория систем;
- системный анализ;
- системный подход;
- структурный анализ;
- методы структурно-функционального моделирования;
- методы объектно-ориентированного моделирования;
- эмпирические методы (интервью, анкетирование, совещания);
- анализ документов;
- прототипирование и др.

**Актуальность.** Приезжая на автозаправочную станцию (АЗС), клиенты нередко сталкиваются с огромными очередями. Это невыгодно не только клиентам АЗС, но и ее владельцам. Клиенты вынуждены либо ожидать свою очередь, либо, несмотря на все преимущества этой АЗС, уезжать, не заправившись, и искать относительно свободную АЗС. Владельцы из-за этого теряют большие деньги. Решение этой проблемы на сегодняшний день является очень актуальным. Причем решением, несомненно, является разработка и внедрение программной системы автоматизации работы заправочных станций с помощью мобильного приложения или веб-версии. В результате автоматизации

будет возможность удаленно просматривать количество машин в очереди, занимать очередь на определенное время, не будет необходимости выходить из машины, чтобы оплатить услугу заправки машины или заправить ее. Следовательно, клиенты будут меньше задерживаться на заправочной станции и смогут быстрее освободить заправочное место для другого клиента.

Имеются два подхода к разработке программной системы и соответственно к определению требований [9]:

- 1) заказная (нетиповая) программная система;
- 2) типовая (массово тиражируемая) программная система на продажу.

В случае заказной (нетиповой) продукции требования предъявляет заказчик, а исполнитель только фиксирует их и формирует из представленных бизнес-требований и пользовательских требований — функциональные, а после создает проект (техническое задание). Обычно содержание проекта, сроки и бюджет указываются в контракте (договоре). Для успешного их взаимодействия в проектах *Agile* всегда есть выделенный представитель пользователей [10].

В случае же типового программного продукта все требования формирует сторона разработчика на основе конкурентного анализа рынка. Важная роль в продвижении приложения принадлежит отделу маркетинга компании [11]. Причем бизнес-требования и пользовательские требования представляет отдел маркетинга компании-разработчика вместе с аналитиком, а из них командой разработчиков создаются функциональные требования. Как правило, содержание проекта, сроки и бюджет указываются в уставе, и все вопросы, связанные с требованиями, отдел маркетинга и отдел разработки в компании решают вместе. Отдел маркетинга внимательно изучает рынок программных продуктов, анализирует, какие программные системы на сегодняшний день могут быть актуальны или какие функции в программных системах могут показаться интересными покупателям и представляет очень приближенные (примерные), но не самые точные данные о том, для кого именно разрабатывается система.

**Целью исследования** является разработка требований к программной системе поддержки автозаправочной деятельности АЗС «Тран-Ойл», способной сократить время обслуживания одного клиента.

Известно, что любая коммерческая организации стремится получить как можно больше прибыли, следовательно, сокращение времени обслуживания одного клиента означает увеличение потока клиентов, а это, в свою очередь, приведет к увеличению прибыли компании и, как следствие, повышению конкурентоспособности.

В исследовании, в первую очередь, нам нужно убедиться в необходимости разработки именно

того программного средства, о котором говорит заказчик. Создадим концепцию (видение) продукта и представим границы проекта – образ решения и ограничения для текущей версии, а также ориентировочные сроки и бюджет проекта.

Видение продукта у всех участников должно быть единым. Многие авторы считают его отдельным проектом, поскольку на его основании принимается решение, «быть или не быть» продукту [12].

*Возможности бизнеса.* Планируется разработка мобильного приложения, которое позволило бы сократить время заправки одного транспортного средства. Сэкономленное время позволит увеличить поток клиентов и повысить доходность автозаправочной станции.

*Бизнес-цели:*

- увеличить прибыль за счет сокращения времени обслуживания одного клиента на заправочной станции на 45 % в течение 3 мес. после первого выпуска системы;

- сократить расходы автозаправочной станции на 20 % в течение 6 мес. после первого выпуска системы;

- достигнуть лидерства автозаправочной станции в течение 2-х лет после первого выпуска системы.

*Критерии успеха:*

- увеличение количества обслуженных клиентов за день на 30 % в течение 3 мес. и на 45 % – в течение 6 мес. после первого выпуска системы;

- повышение доходности автозаправочной станции до 40 %.

*Бизнес-риски:*

- новая система (мобильное приложение) может оказаться неудобной для некоторых пользователей, в связи с чем поток клиентов может уменьшиться (вероятность = 0,2; ущерб = 2).

*Основные функций приложения:*

- предварительное резервирование топлива;
- выбор колонки и вида топлива через мобильное/веб-приложение;

- удаленное приглашение заправщика к автомобилю;

- определение количества машин в очереди к определенной колонке;

- заправка транспортного средства сотрудником;

- онлайн оплата услуг АЗС;

- отправка клиенту электронного чека.

В разработке планируется использование итерационной модели, поэтому очень важно распределить функции пользователя по приоритетам. Это позволит реализовать, в первую очередь, наиболее востребованные и исключить второстепенные или невостребованные функции (которые не подчиняются цели разработки).

*Особенности развертывания.* ПО веб-сервера нужно обновить до последней версии. В рамках 1-го выпуска нужно разработать веб-версию, а в рамках 2-го выпуска – мобильное приложение для смартфонов и планшетов под управлением iOS и Android. К моменту готовности 2-го выпуска все соответствующие изменения должны быть выполнены. Планируется разработать видеоролики длительностью не более 5 мин, обучающие пользователей работе с веб-версией и приложением системы.

Для определения требований к системе, в первую очередь, выполним анализ предметной области. АЗС «Тран-Ойл» является небольшой организацией, занимающейся продажей топлива, сопутствующих товаров и обслуживанием автомобилей в Астрахани. Основными видами топлива являются бензин и дизельное топливо. Создадим организационную модель АЗС «Тран-Ойл» и изучим ее бизнес-процессы (рис. 1). Главным лицом, отвечающим за ее работу, является директор, у которого в подчинении находятся четыре отдела: технический, отдел логистики, бухгалтерия, отдел продаж. Технический отдел занимается ремонтом и обслуживанием автомобилей и другой техники на АЗС. Отдел логистики занимается контролем за складскими запасами топлива, его доставкой и хранением. Бухгалтерия занимается финансовыми операциями и бухгалтерским учетом. Отдел продаж занимается работой с клиентами и продажей топлива, а также других товаров, связанных с автомобильным транспортом. Кроме этого, АЗС «Тран-Ойл» также своим клиентам оказывает следующие услуги: мойка автомобилей, шиномонтаж, автомойка. Имеется небольшой гастронорм.

Итеративно разработаны будут три выпуска (табл. 1).

**Таблица 1.** Состав первого и последующих выходов программной системы

Функция	Выпуск 1	Выпуск 2	Выпуск 3
Выбор топлива и заправка транспортного средства	Реализация основных функций: вход в приложение, выбор колонки и топлива, оплата услуг, формирование чека	Реализована полностью	–
Отправка электронного чека на электронную почту, программа лояльности	Отправка чека на электронную почту. Формирования накопительной, балльной системы, формирование купонов	Добавление в чек QR-кода	–
Резервирование топлива и вызов заправщика к машине	Предварительное резервирование топлива по предоплате	Формирование функции вызова заправщика к машине	Реализована полностью

Нас интересуют отдел продаж, поскольку для него и будет разработана программа, и бухгалтерия, поскольку программа будет иметь внешний интерфейс с бухгалтерской программой для проведения оплаты. В отделе продаж работают диспетчер, заправщик, кассир.

Подробно разберем бизнес-процессы автозаправочной деятельности на структурно-функциональной модели. Многие исследователи подчеркивают преимущества структурно-функционального подхода к моделированию бизнес-процессов компании [13–16]. Для этого лучше всего использовать диаграммы в нотации IDEF0, поскольку она считается наиболее удобной для их подробного изучения и анализа в контексте системного подхода (рис. 2–4). Цель моделирования – провести анализ бизнес-процесса «как есть», выявить недостатки и уязвимости в функционировании системы, выработать контрмеры для устранения недостатков и построить модель «как должно быть» с учетом контрмер.

Клиент заезжает на заправку. ориентировочно, к колонке с нужным топливом, ожидает своей очереди и оплачивает за топливо (при необходимости его консультирует диспетчер), возвращается к машине и заправляет ее (если есть свободный заправщик, то эту работу делает он).

В контекстной диаграмме описывается бизнес-

процесс в общем. На входе у нас:

- заказ клиента;
- информация о топливе;
- само топливо в резервуаре;
- незаправленный автомобиль.

На выходе:

- заправленный автомобиль;
- отчетность по заправке для руководства;
- чек оплаты для клиента.

В механизмах: клиент, который подгоняет машину для заправки, диспетчер, консультирующий клиентов и регулирующий все вопросы, заправщик автомобиля, кассир, который принимает оплату за топливо.

В управлении:

- нормативно-правовые документы;
- ГОСТ 58404;
- регламент работы АЗС.

Контекстная диаграмма была декомпозирована на три подпроцесса (рис. 3):

- определение нужной колонки с топливом;
- оплата выбранного топлива;
- заправка автомобиля.

– Основные недостатки существующих бизнес-процессов и предлагаемые контрмеры отражены в табл. 1.

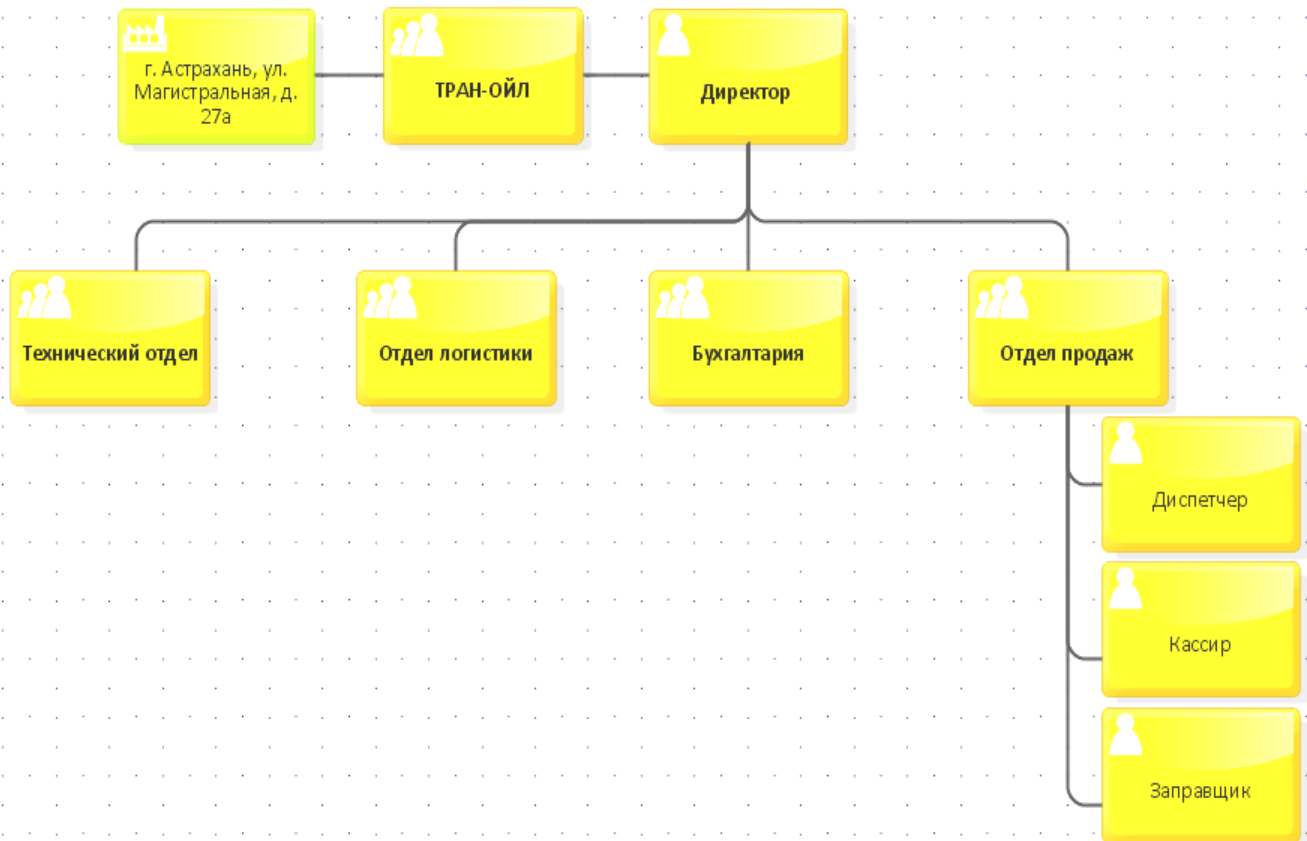


Рис. 1. Организационная диаграмма АЗС

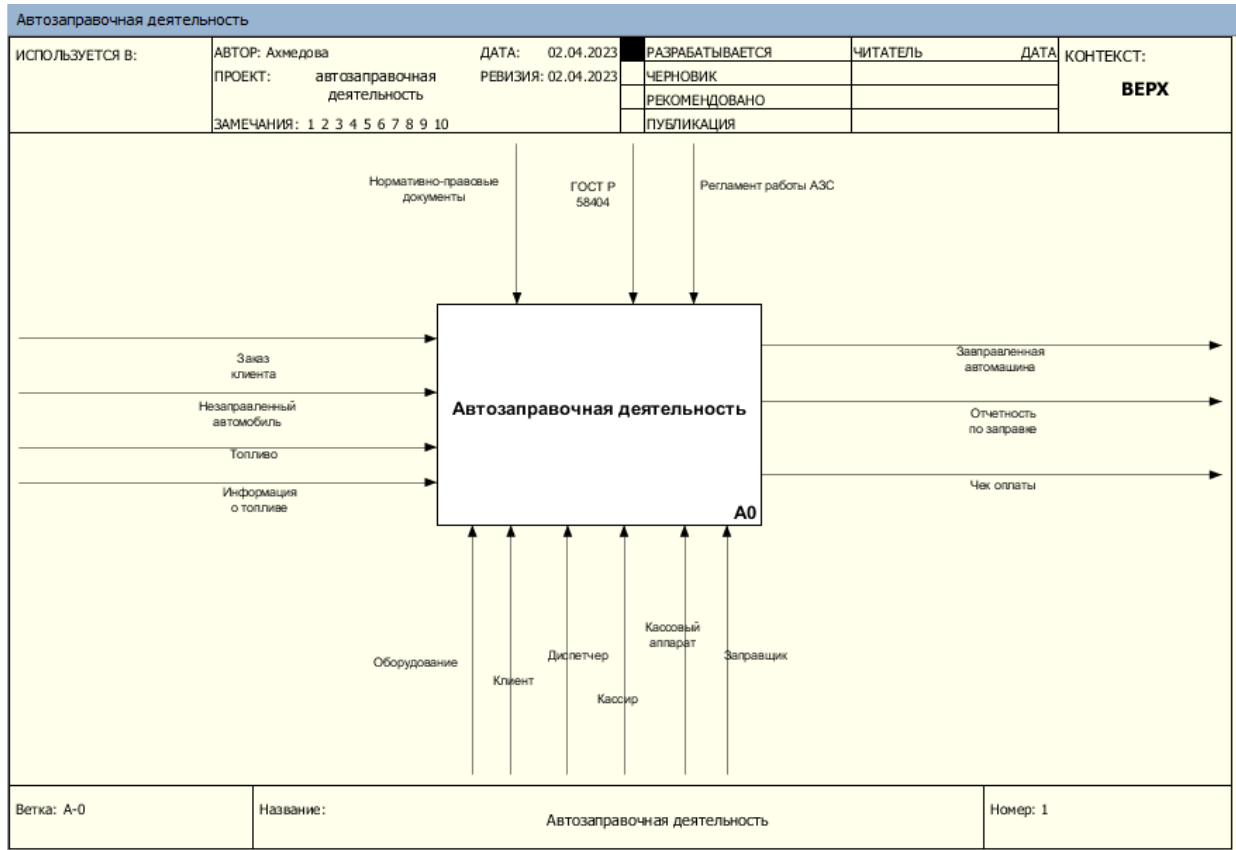


Рис. 2. Контекстная диаграмма автозаправочной деятельности в состоянии «as is»

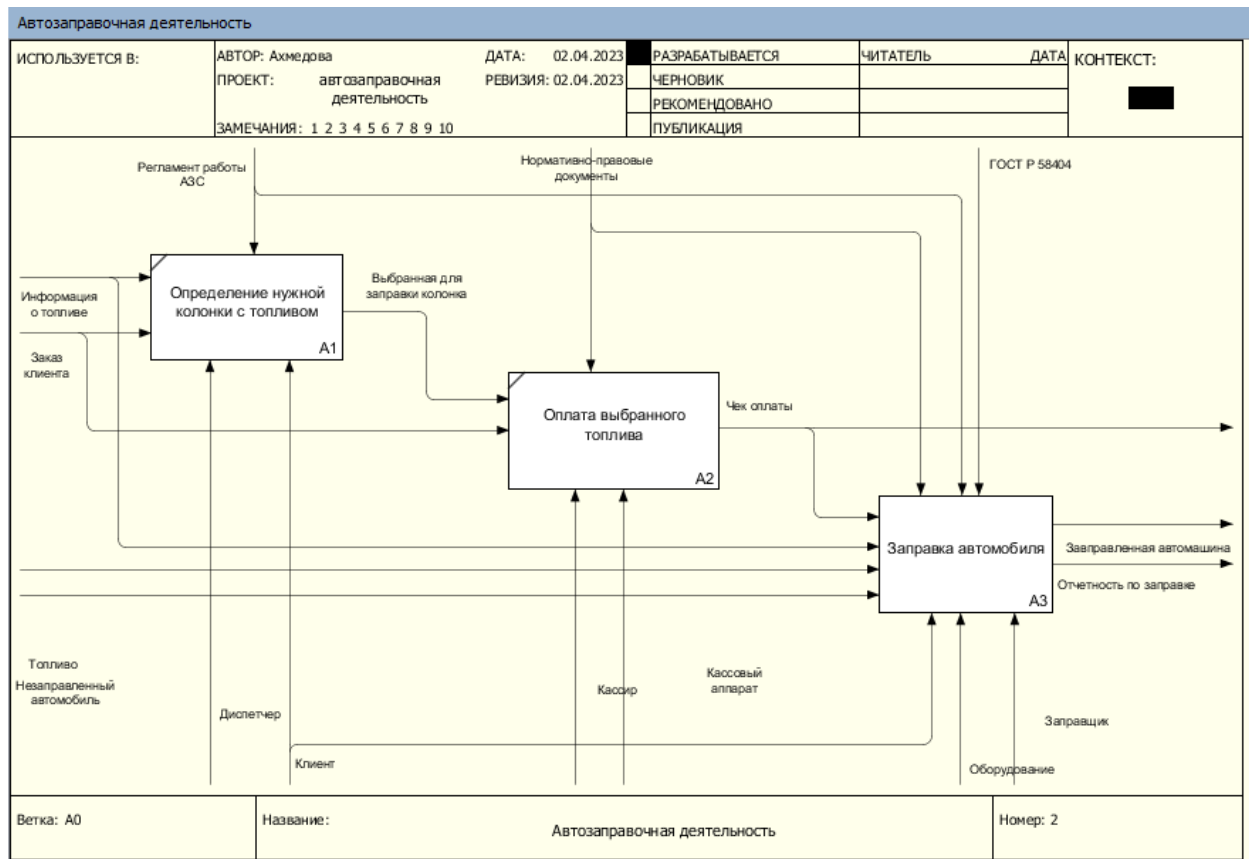


Рис. 3. Диаграмма декомпозиции автозаправочной деятельности в состоянии «as is»

**Таблица 1.** Выявленные недостатки в существующих бизнес-процессах и предлагаемые контрмеры

Недостатки существующих бизнес-процессов	Предлагаемые контрмеры
Продолжительный по времени период (увеличенное время) обслуживания одного автомобиля	Предварительное бронирование нужного количества имеющегося в наличии вида топлива, предварительное бронирование определенной колонки с нужным видом топлива, приглашение заправщика к колонке к определенному времени, онлайн оплата через приложение;
Образование очередей и утечка клиентов из-за длинных очередей	Определение количества машин в очереди к определенной колонке
Отсутствие возможности предварительного резервирования топлива на случай его продажи и исчерпания запасов к моменту подачи машины к колонке	Предварительное резервирование топлива через приложение
Отсутствие возможности учета постоянных покупателей и скидок для клиентов	Создание программы лояльности и купонов для клиентов через приложение
Отсутствие возможности узнать о качестве топлива	Обеспечение возможности написания отзывов о топливе и просмотра чужих отзывов
Большие расходы руководства АЗС на оплату зарплаты кассирам	Сокращение кассиров за счет онлайн оплаты

После того как проанализировали текущие бизнес-процессы, выявили недостатки и определили контрмеры, проведем сравнительный анализ программных систем (сервисов) автоматизации работы автозаправочных станций, которые на сегодняшний день существуют на рынке информационно-коммуникационных технологий. На данном этапе очень важно проанализировать идеи, реализованные в конкурентных программах, для того, чтобы можно было использовать наиболее удачные решения и исключить неудачные [17].

Имеется ряд систем автоматизации работы автозаправочной станции, среди которых лидирующие позиции занимают «Яндекс заправки», «Benzuber», «Turbo», «АЗС Лукойл», «АЗС Газпромнефть», «АЗС.Go», «Тинькофф АЗС» и др. Как оказалось, сервисы имеют много возможностей, начиная от страхования клиентов от скачков цен на бензин, поломки из-за некачественного топлива и заказа еды на АЗС с доставкой к машине до оплаты бензина бонусными баллами картой лояльности, вызова техпомощи и оформления ОСАГО. У всех этих приложений есть мобильные версии и веб-версии, причем все они бесплатны. Однако, несмотря на обилие пользовательских возможностей, ни один из сервисов перечисленных программ не имеет возможности предварительного резервирования топлива на случай продажи и исчерпания запасов к моменту обслуживания определенного клиента, ни у одной программы нет возможности приглашения заправщика к машине к моменту подачи машины к колонке для заправки, нет возможности определения количества машин в очереди к определенной колонке, а вот возможность онлайн оплаты за

топливо имеется только у «Яндекс заправки», «Лукойл» и «Тинькофф АЗС» [18].

Отличным решением для достижения поставленной главной цели, а именно сокращения времени обслуживания одного клиента, послужит выпуск мобильного приложения и веб-приложения. Это позволит клиентам предварительно резервировать топливо, приглашать заправщика к машине, определять количество машин в очереди к колонке, производить оплату за топливо, не выходя из машины, копить бонусные баллы для частичной оплаты последующих покупок, пользоваться купонами, узнать о качестве топлива по отзывам, а владельцам АЗС, кроме выгоды от привлечения клиентов, — иметь возможность учета клиентов и их хранение в базе данных для последующего взаимодействия, минимизировать количество кассиров за счет увольнения, соответственно, сократить расходы на оплату зарплаты кассирам. Это позволит, с одной стороны, ускорить обслуживание одного автомобиля, избавиться от очередей, удержать старых и привлечь новых клиентов, с другой стороны — сократить лишние кадры, увеличить прибыль компании.

Далее построим дерево целей, которое показывает по иерархическому принципу совокупность целей программной системы, в которой выделены генеральная цель, подчиненные ей подцели первого, второго и последующих уровней (рис. 4).

Для достижения поставленных целей, устранения выявленных недостатков и уязвимостей с учетом проектируемого приложения нам необходимо внести изменения в подпроцессы и отразить эти изменения в диаграмме «to be» (рис. 5, 6).



Рис. 4. Дерево целей построения программной системы для АЗС

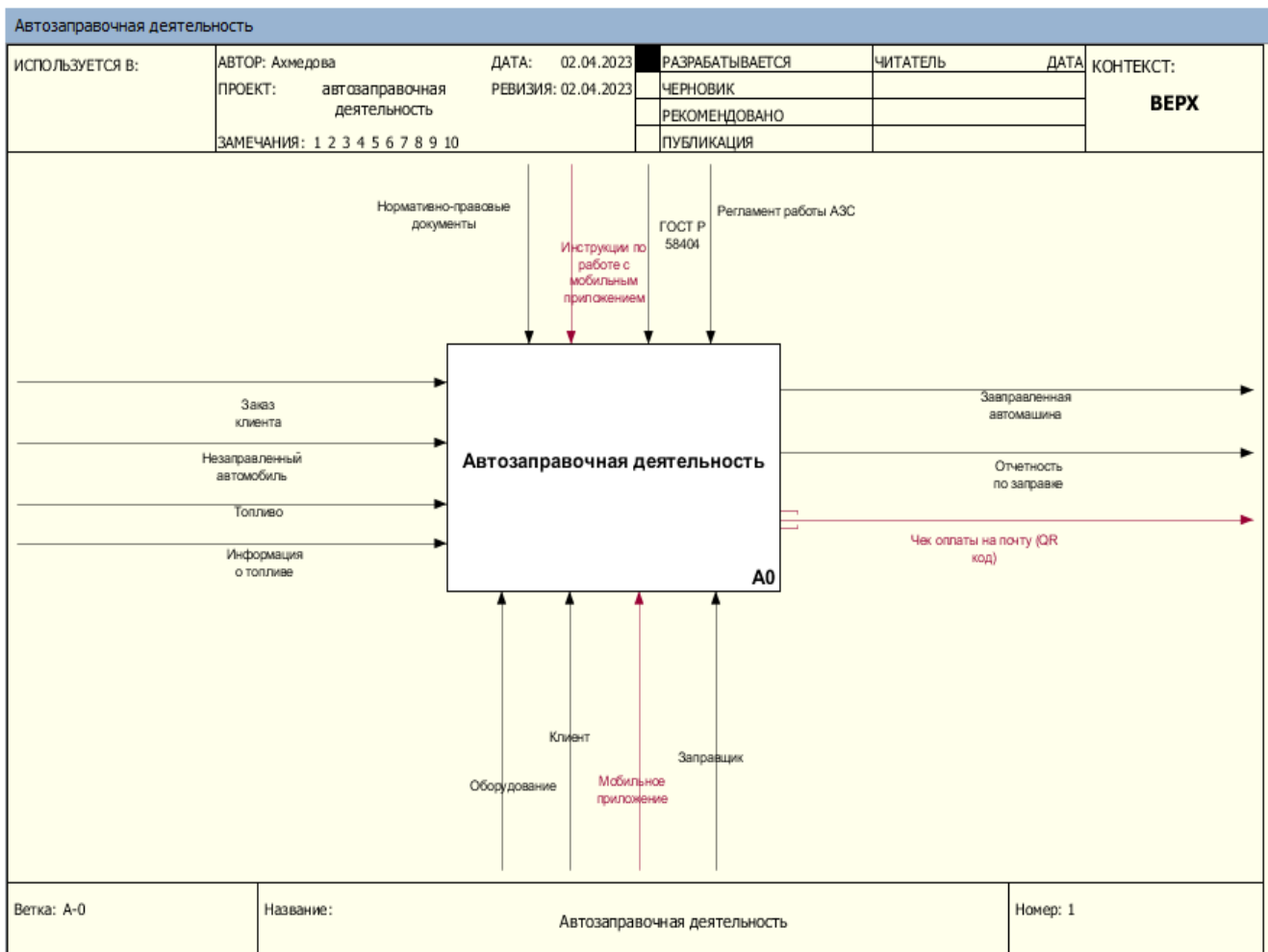


Рис. 5. Процесс «автозаправочная деятельность» в состоянии «to be»

Измененные подпроцессы отражены на декомпозированной диаграмме (рис. 6). Под процесс

«заправка автомобиля» выполняется в следующей последовательности действий (рис. 7).



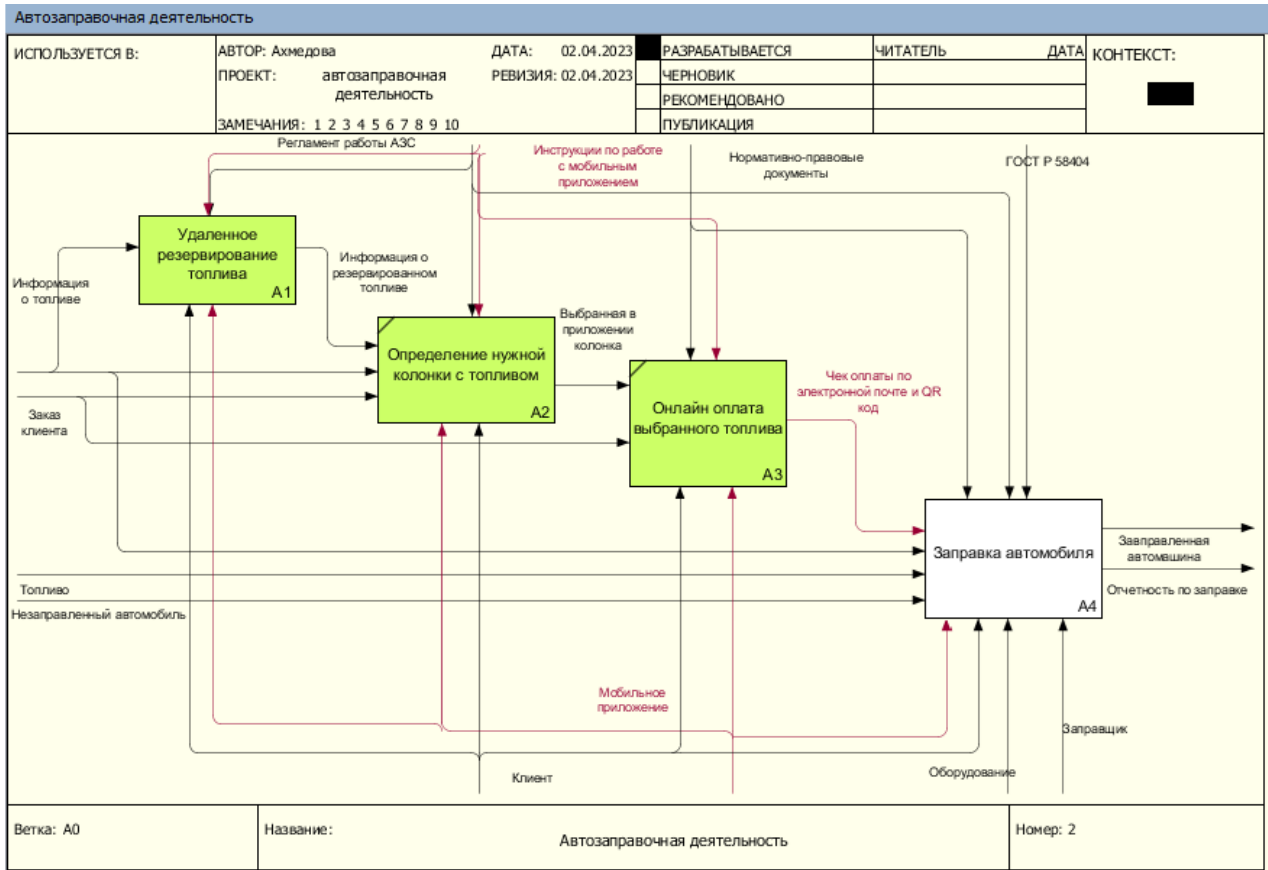


Рис. 6. Подпроцессы процесса «автозаправочная деятельность» в состоянии «to be»

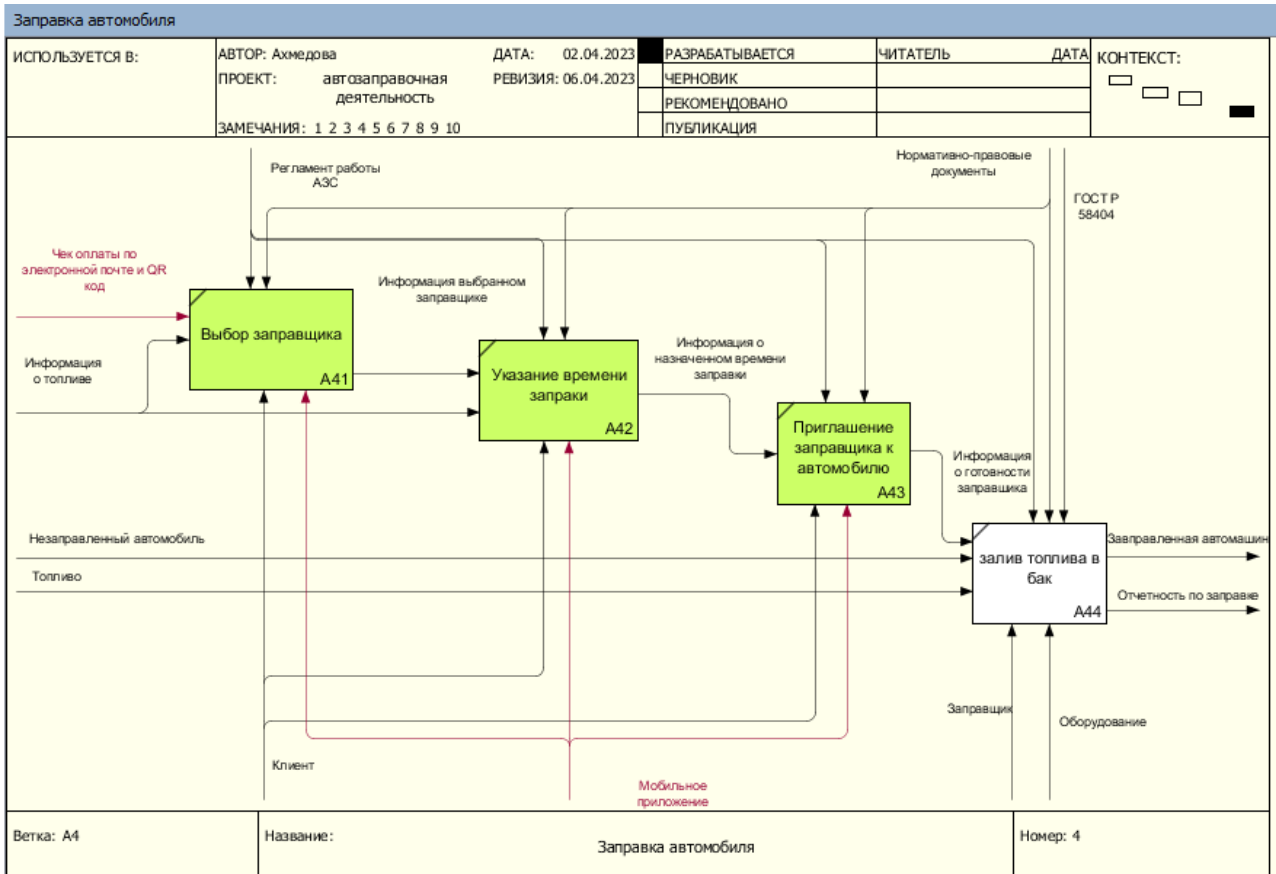


Рис. 7. Декомпозиция подпроцесса «заправка автомобиля» в состоянии «to be»

На рис. 6 видно, что в управлении появилась инструкция по работе с программной системой, а в механизмах — сама программная система. Роли кассира и диспетчера выпали из механизмов. Все подпроцессы, кроме физической заправки автомобиля заправщиком, полностью автоматизированы. Добавился подпроцесс «удаленное резервирование топлива»; определение колонки и оплата происходят с помощью программной системы. Подпроцесс «заправка автомобиля» выполняется в следующей последовательности действий (рис. 7): по чеку оплаты клиент выбирает в программе из списка заправщиков свободного заправщика и указывает время заправки автомобиля, далее приглашает заправщика к автомобилю также в программе (по номеру колонки и номеру автомобиля), заправщик подходит к машине и заливает оплаченное топливо в автомобильный бак.

Как известно, пользовательские требования определяют набор пользовательских задач, которые должна решать программа, а также способы (сценарии) их решения в системе. Ранее мы определили роли работников отдела продаж, которые занимались автозаправочной деятельностью — заправщик, диспетчер, кассир. Но, как видим на рис. 6, 7, уже отсутствуют профили кассира и диспетчера. Это значит, что диспетчер может заниматься текущими проблемами автозаправочной станции, а работа кассира вовсе может подлежать сокращению.

Пользовательские требования могут выражаться в виде фраз утверждений, т. е. пользовательских историй, а также в виде сценариев использования [19]. Пользовательские истории для нашей системы имеют следующий вид:

- как клиент АЗС, я хочу иметь возможность *предварительно бронировать* нужное количество имеющегося в наличии топлива»;
- как клиент АЗС, я хочу иметь возможность просмотреть виды топлива;
- как клиент, я хочу иметь возможность заранее выбрать вид топлива для заправки автомобиля»;
- как клиент, я хочу иметь возможность *предварительно резервировать* необходимое количество определенного вида топлива»
- как клиент АЗС, я хочу иметь возможность выбрать для заправки колонку с нужным мне видом топлива;
- как клиент АЗС, я хочу иметь возможность пригласить заправщика к колонке к определенному времени для заправки машины;

- как клиент АЗС, я хочу иметь возможность произвести онлайн оплату;
- как клиент АЗС, я хочу иметь возможность определять количество машин в очереди к определенной колонке;
- как клиент АЗС, я хочу иметь возможность написать отзыв и прочитать чужие отзывы;
- как заправщик, я хочу иметь возможность назначить время прибытия машины для ее заправки;
- как заправщик, я хочу иметь возможность принять заказ для подхода к машине для ее заправки»;
- как заправщик, я хочу иметь возможность видеть в системе определенную колонку и машину для заправки;
- как диспетчер, я хочу видеть запросы клиентов на резервирование топлива;
- как диспетчер, я хочу бронировать по запросу клиента определенное топливо на определенное время;
- как диспетчер, я хочу иметь возможность добавлять клиентов в базу данных клиентов;
- как администратор приложения, я хочу иметь возможность управлять отзывами о топливе;
- как администратор приложения, я хочу иметь возможность управлять отзывами о топливе;
- как администратор приложения, я хочу иметь возможность управлять внутренними профилями пользователей;
- как бухгалтер, я хочу иметь возможность управлять онлайн оплатами клиентов за бензин;
- как бухгалтер, я хочу иметь возможность предоставлять скидки постоянным клиентам;
- как бухгалтер, я хочу иметь возможность формировать купоны и др.

Описать, какой функционал разрабатываемой программной системы доступен каждой группе пользователей, а также детализировать и дробить пользовательские истории помогает также диаграмма вариантов использования. Она позволяет далее сформулировать общие требования к функциональному поведению проектируемой системы, а также подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями [20]. Основной фрагмент диаграммы вариантов использования показан на рис. 8.



Рис. 8. Фрагмент диаграммы вариантов использования

Далее опишем некоторые функциональные требования к программной системе. Система должна предоставлять следующие возможности:

- регистрация пользователей для упрощения взаимодействия с системой;
- выбор автозаправочной станции на карте;
- выбор колонки и вида топлива через всплывающие экранные формы;
- оплата услуг, предоставленных АЗС, с помощью банковской карты;
- зарегистрированным пользователям – возможность просматривать скидки и акции на конкретных АЗС;
- зарегистрированным пользователям – возможность написания отзыва на конкретной АЗС;
- зарегистрированным пользователям – возможность редактировать и удалять написанные отзывы;
- администраторам АЗС – возможность следить за всеми процессами, происходящими на АЗС, через специальное окно логов;
- автозаправщикам – предоставлять информацию о том, на какой колонке и какой вид топлива выбрал клиент, после чего разрешать подачу

топлива на выбранную колонку;

- передавать информацию автозаправщикам через специальную форму и др.

Далее определим нефункциональные требования к системе. Нефункциональные требования относятся к техническим аспектам, которым должна соответствовать система, таким как характеристики качества (проблемы, связанные с производительностью, надежностью, доступностью и др.), ограничения, внешние интерфейсы.

Основные требования к качеству разрабатываемой программной системы следующие:

– *Безопасность.* Не допускается утечка конфиденциальных данных сотрудников и пользователей (клиентов). Системе необходимо обладать защитой от различных видов взлома и других атак. Система должна предупреждать пользователя о попытках взлома или кражи данных третьей стороной.

– *Доступность.* Система должны быть доступна и успешно поддерживаться 90 % существующих на данный момент мобильных устройств.

– *Надежность.* Допускается незначительная потеря пакетов при отправке или получении данных от сервера. Потерянные пакеты не должны превышать

0,8 % от общего числа отправленных/полученных пакетов.

– *Устойчивость*. Система должна быть устойчива к множественным запросам со стороны клиента. При возникновении сбоя какая-либо потеря данных не допускается.

– *Эффективность*. Система должна потреблять не более 15 % ресурсов процессора, доступных для мобильного приложения. Система обязана предупреждать пользователя в случае, если нагрузка на устройство достигнет 80 % максимальной плановой загрузки.

– *Простота использования*. Интерфейс приложения должен быть удобен для пользователя. Использование приложения не должно быть сложным, пользователю необходимо предоставить возможность быстро сориентироваться в интерфейсе приложения

*Описание ограничений к программной системе:*

– при проектировании приложения рекомендуется использовать паттерн *mvvm*;

– программная система во время эксплуатации не должна нарушать какие-либо нормативно-правовые акты, предусмотренные законодательством;

– приложение должно быть реализовано на достаточном теоретическом уровне.

*Внешние интерфейсы:*

– интерфейс пользователя должен быть удобным и интуитивно понятным;

– программа должна иметь возможность взаимодействия с 1С-CRM, 1С, а также экспортировать данные в программы *Microsoft Word*, *Microsoft Excel* и импортировать данные из них.

**Заключение.** На основе проведенного исследования спецификации требований к программной системе (мобильному и веб-приложению) для формирования технического задания можно сделать следующие выводы. Постоянное взаимодействие с заказчиком и конечными пользователями позволило избежать проблем, связанных с разработкой требований. На всех этапах разработки требований для однозначного понимания пользовательских требований активно использовались модели анализа и проектирования. Это позволило выявить некорректные, избыточные и недостающие требования. Собранные требования успешно утверждены, соглашение о требованиях достигнуто. Менеджер проекта со стороны заказчика подтвердил, что все требования починены бизнес-цели. Разработанные требования позволят создать систему, которая способна сократить время заправки одного транспортного средства на 40 %, увеличить количество обслуженных клиентов за день на 45 % в течение 6 мес. после 1-го выпуска системы, увеличить прибыль за счет этого на 45 % в течение 3 мес. после 1-го выпуска системы, сократить расходы автозаправочной станции на 20 % в течение 6 мес. после 1-го выпуска системы, достигнуть лидерства автозаправочной станции в течение 2-х лет после 1-го выпуска системы.

#### Литература

1. Казиев В.М. Введение в анализ, синтез и моделирование систем. М.: Интернет ун-т информационных технологий, 2018. 248 с.
2. Вигерс Карл, Битти Джой. Разработка требований к программному обеспечению. 3-е изд., доп. / пер. с англ. М.: Изд-во «Рус. редакция», 2014. 736 с.
3. Советов Б.Я., Яковлев С.А. Моделирование систем. М.: Юрайт, 2018. 638 с.
4. Буч Грейди, Рамбо Джеймс, Джекобсон Айвар. Язык UML. Руководство пользователя. М.: ДМК, 2015. 432 с.
5. Леоненков А.В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose. М.: Интернет ун-т информационных технологий, 2020. 320 с.
6. Химонин Ю. Сбор и анализ требований к программному продукту. Версия 1.03. 2019. URL: [https://wiki.dmdevelopment.ru/wiki/Download/Books/Test/Software\\_Requirements\\_Khimonin.pdf](https://wiki.dmdevelopment.ru/wiki/Download/Books/Test/Software_Requirements_Khimonin.pdf) (дата обращения: 05.02.2023).
7. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы [Электронный ресурс]. URL: <https://docs.cntd.ru/document/1200006924> (дата обращения: 05.02.2023).
8. Ким В.Ю. Особенности разработки дизайна пользовательского интерфейса для мобильного приложения // Новые информационные технологии в автоматизированных системах. 2015. № 18. URL: <https://cyberleninka.ru/article/n/osobennosti-razrabotki-dizayna-polzovatelskogo-interfeysa-dlya-mobilnogo-prilozheniya> (дата обращения: 05.02.2023).
9. Гаврилов Л.П. Инновационные технологии в коммерции и бизнесе. М.: Юрайт, 2019. 372 с.
10. Дейв Томас. Гибкая веб-разработка с Rails (прагматичные программисты). М.: Pragmatic Bookshelf, 2018. 720 с.
11. Гаджиев Н.Г., Карпунин А.Ю., Карпунина Е.В., Коноваленко С.А. Развитие маркетинга во внешнеэкономической деятельности // Экономика. Информатика. 2022. Т. 49, № 1. С. 79-91.
12. Орлов С.А. Технологии разработки программного обеспечения. М.: Питер, 2017. 464 с.
13. Алпатов Ю.Н. Моделирование процессов и систем управления. СПб.: Лань, 2018. 140 с.
14. Власов В.А., Курочкин С.П., Коновалов С.В. Задачи по функциональному анализу. М.: ИНФРА-М, 2020. 106 с.
15. Дворецкий С.И. Моделирование систем. М.: Академия, 2019. 304 с.
16. Елиферов В.Г., Репин В.В. Бизнес-процессы. Регламентация и управление. М.: ИНФРА-М, 2022. 320 с.

17. Васильева К.Н., Хусаинова Г.Я. Обзор программных средств для разработки мобильных приложений // Colloquium journal. 2020. № 2 (54). URL: <https://cyberleninka.ru/article/n/obzor-programmnyh-sredstv-dlya-razrabotki-mobilnyh-prilozheniy> (дата обращения: 05.02.2023).
18. Тинькофф журнал. 7 приложений, которые помогут заправиться, не выходя из машины. 2021. № 8. URL: <https://journal.tinkoff.ru/short/azs-online/> (дата обращения: 05.01.2023).
19. Паттон Джефф. Пользовательские истории. Искусство гибкой разработки ПО. СПб.: Питер, 2017. 288 с.
20. Фаулер Мартин. UML. Основы. Краткое руководство по стандартному языку объектного моделирования. М.: Символ-плюс, 2016. 192 с.
8. Kim V.Yu. Features of user interface design development for mobile applications // Novye informacionnye tekhnologii v avtomatizirovannyh sistemah. 2015. № 18. URL: <https://cyberleninka.ru/article/n/osobennosti-razrabotki-dizayna-polzovatelskogo-interfeysa-dlya-mobilnogo-prilozheniya> (дата обращения: 05.02.2023).
9. Gavrilov L.P. Innovative technologies in commerce and business. М.: Yurajt, 2019. 372 p.
10. Dejev Tomas. Flexible web development with Rails (pragmatic programmers). М.: Pragmatic Bookshelf, 2018. 720 p.
11. Gadzhiev N.G., Karpunin A.Yu., Karpunina E.V., Konovalenko S.A. Development of marketing in foreign economic activity // Economics. Information technologies. 2022. V. 49, № 1. P. 79-91.
12. Orlov S.A. Software development technologies. М.: Piter, 2017. 464 p.
13. Alpatov Yu.N. Modeling of processes and control systems. SPb.: Lan', 2018. 140 p.
14. Vlasov V.A., Kurochkin S.P., Konovalov S.V. Tasks on functional analysis. М.: INFRA-M, 2020. 106 p.
15. Dvoreckij S.I. Modeling of systems. М.: Akademiya, 2019. 304 p.
16. Eliferov V.G., Repin V.V. Business processes. Regulation and management. М.: INFRA-M, 2022. 320 p.
17. Vasil'eva K.N., Husainova G.Ya. Review of software tools for the development of mobile applications // Colloquium journal. 2020. № 2 (54). URL: <https://cyberleninka.ru/article/n/obzor-programmnyh-sredstv-dlya-razrabotki-mobilnyh-prilozheniy> (дата обращения: 05.02.2023).
18. Tinkoff magazine. 7 apps that will help you refuel without leaving the car. 2021. № 8. URL: <https://journal.tinkoff.ru/short/azs-online/> (дата обращения: 05.01.2023).
19. Patton Dzheff. User stories. The art of agile software development. SPb.: Piter, 2017. 288 p.
20. Fauler Martin. UML. The basics. A brief guide to the standard object modeling language UML. М.: Simvolplyus, 2016. 192 p.

#### *References*

1. Kaziev V.M. Introduction to analysis, synthesis and modeling of systems. М.: Internet un-t informacionnyh tekhnologij, 2018. 248 p.
2. Vigers Karl, Bitti Dzhoj. Development of software requirements. 3-e izd., dop. / per. s angl. М.: Izd-vo «Rus. redakciya», 2014. 736 p.
3. Sovetov B.Ya., Yakovlev S.A. Modeling of systems. М.: Yurajt, 2018. 638 p.
4. Buch Grejdi, Rambo Dzhejms, Dzhekobson Ajvar. UML Language. User's Guide. М.: DMK, 2015. 432 p.
5. Leonenkov A.V. Object-oriented analysis and design using UML and IBM Rational Rose. М.: Internet un-t informacionnyh tekhnologij, 2020. 320 p.
6. Himonin Yu. Collection and analysis of software product requirements. Version 1.03. 2019. URL: [https://wiki.dmdevelopment.ru/wiki/Download/Books/Test/Software\\_Requirements\\_Khimonin.pdf](https://wiki.dmdevelopment.ru/wiki/Download/Books/Test/Software_Requirements_Khimonin.pdf) (дата обращения: 05.02.2023).
7. GOST 34.602-89. Information technology. A set of standards for automated systems. Technical specification for the creation of an automated system [Elektronnyj resurs]. URL: <https://docs.cntd.ru/document/1200006924> (дата обращения: 05.02.2023).