



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский технологический университет»
МИРЭА

Институт информационных технологий
Кафедра корпоративных информационных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРА НА ТЕМУ:

**«РАЗРАБОТКА ПРИЛОЖЕНИЯ ПО УЧЕТУ ДОГОВОРОВ
С ИСПОЛЬЗОВАНИЕМ НОН-КОД ПЛАТФОРМ НА ОСНОВЕ
МЕТОДА AGILE SCRUM»**

Студент:

Стаценко М.Н.

Научный руководитель:

к.т.н., доц. МИРЭА Степанов Д.Ю.

Москва – 2022

Оглавление

Введение.....	4
Цель и задачи.....	5
Раздел 1. Описание метода Agile Scrum для управления проектами по разработке программного обеспечения.....	7
1.1. Описание метода Agile Scrum.....	7
1.2. Анализ деятельности по учёту договоров.....	10
1.3. Анализ нон-код платформ.....	11
1.4. Анализ применения Agile Scrum для реализации корпоративных информационных систем.....	13
Раздел 2. Идентификация требований.....	15
2.1. Специфика процесса управления договорами.....	15
2.2. Формирование бэклога продукта.....	17
2.3. Формирование спринтов.....	20
Раздел 3. Первый спринт: проектирование ключевых бизнес-процессов.....	25
3.1. Проектирование процессов AS-IS в нотациях ARIS VACD и BPMN SLD.....	27
3.2. Проектирование процесса «TO-BE» в нотациях ARIS VACD и BPMN SLD.....	31
3.3. Проектирование данных.....	36
3.4. Проектирование интерфейсов.....	38
Раздел 4. Реализация основных функций приложения.....	42
4.1. Второй спринт: разработка каркаса приложения.....	42
4.2. Третий спринт: разработка базы данных и интерфейса для оставшихся экранов..	45
4.3. Четвёртый спринт: добавление функционала для фильтрации и поиска.....	48
4.4. Нагрузочное тестирование.....	51
Заключение.....	54
Список литературных источников.....	55
Приложение А.....	58
Приложение Б.....	59
Приложение В.....	60
Приложение Г.....	61



Приложение Д..... 62

Введение

В настоящее время как в рамках мирового опыта, так и в рамках России наблюдается рост заинтересованности в каждой из сфер экономики в применении ERP-систем для управления и, в частности, автоматизации бизнес-процессов предприятием [1]. Однако в большинстве случаев компании малого и среднего бизнеса не могут позволить полномасштабное внедрение таких систем по причине продолжительности процесса внедрения и как следствие его дороговизны [2].

Однако одним из вариантов решения данного вопроса является приобретение или разработка собственного программного обеспечения для решения критически важных, профильных процессов силами штатных сотрудников или наёмных разработчиков. Очевидно, что все варианты требуют временных и денежных затрат в различном соотношении. Для снижения данных видов затрат и решения задачи автоматизации можно использовать платформы по созданию приложений «без кода», ориентированные на пользователей без знаний языков программирования и опыта разработки, и привлечь к работе штатных сотрудников.

С момента появления первых нон-код платформ, датирующихся серединой 90-х, многое изменилось. За последние десятилетия их возможности сильно увеличились. Многие представители ИТ-сообщества считают, что это связано с изменениями рынка ИТ и его спецификой, а также высокой стоимостью их работы. Сфере необходимо достаточно большое количество высококвалифицированных специалистов, которых всегда не хватает, поэтому разработчики начали развивать и платформы для автоматизации работы самих программистов.

Ещё одной веской причиной распространения применения нон-код является большее преимущество в части стоимости владения программой,

изменения и ошибки создания, чем классический подход к программированию. Стоит отметить, что рынок всегда видит потребности своих клиентов, и в период с 2019 года был запущен ряд сервисов для создания собственных приложений, позволяющих решить потребность в автоматизации бизнес-задач без использования кода. Приложение, созданное на базе этого сервиса, имеет достаточно мощный функционал и предоставляет пользователю следующие возможности:

- передачи доступа и управления уровнем доступа для списка лиц компании с целью делегирования задач уполномоченным сотрудникам;
- разработки на уровне программы, определенной регламентом последовательности действий, при необходимости таковой;
- простые настройки интерфейса программы с учётом пожеланий будущих пользователей;
- организации доступа всех заинтересованных лиц компании, в частности в случае территориального распределения компании;
- контроля над исполнением договоров в целом и финансовых обязательств по ним в частности.

Цель и задачи

Цель данной работы состоит в разработке приложения по учёту договоров с использованием нон-код платформ на основе метода Agile Scrum. Таким образом, будет рассмотрена возможность применения гибкого метода управления проектами по разработке программного продукта Agile Scrum в определённой нон-код платформе. Научная новизна настоящей работы заключается в проведении оценки возможности разработки приложения для учёта договоров с применением нон-код платформ, а также в исследовании наиболее подходящих платформ от различных вендоров с выделением их преимуществ и недостатков.

Разработанное приложение должно оптимизировать процессы и способствовать улучшению пользовательского опыта, в свою очередь гибкий метод управления проектами по разработке программного обеспечения обеспечит более качественное его воплощение. Исходя из цели работы, необходимо выполнить следующие задачи:

- проанализировать применение Agile Scrum для реализации корпоративных информационных систем;
- идентифицировать требования к процессам управления договорами;
- спроектировать процессы в нотациях ARIS VACD и BPMN SLD для моделей AS-IS и TO-BE до 3-4 уровней детализации;
- построить карту бизнес-процессов;
- смоделировать данные в нотации UML Class Diagram, включая нормализацию таблиц баз данных до 3-НФ;
- предложить структуру приложения;
- реализовать приложение с использованием non-код платформы;
- выполнить количественную оценку качества работы программы путем нагрузочного тестирования.

Раздел 1. Описание метода Agile Scrum для управления проектами по разработке программного обеспечения

1.1. Описание метода Agile Scrum

В настоящее время Scrum является одним из самых популярных методов управления проектами по созданию программного обеспечения (далее – ПО). Scrum входит в семейство методов гибкой разработки Agile. Он представляет собой метод реализации и внедрения программного обеспечения на основе итерационной модели. Рассматриваемая методология и сам метод базируется на соблюдении основных ценностей и принципов, отображённых на рисунке 1.1 и описанных в манифесте Agile [3-4].

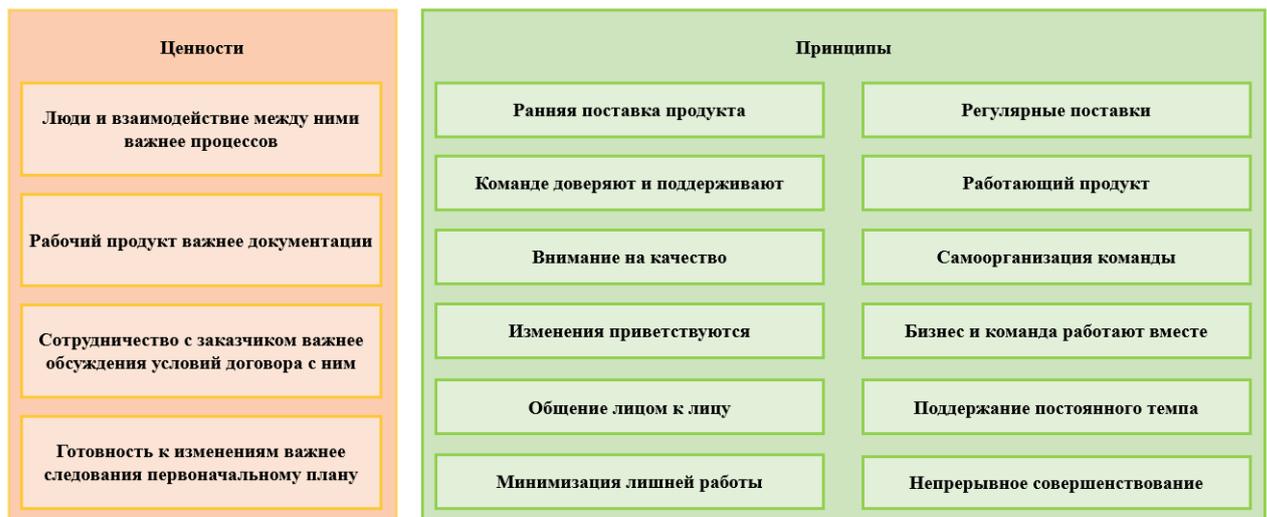


Рисунок 1.1 – Ценности и принципы методологии Agile

Действующими лицами в Scrum-проекте являются профильные технические специалисты, участвующие в нём от первой встречи и до полного завершения проекта, согласно принципам гибкой методологии. Scrum-команда, в центре которой находится владелец продукта, в большинстве

случаев состоит из аналитиков, программистов, тестировщиков, администратора и скрам-мастера. Владелец продукта ведёт коммуникацию с заказчиком и аккумулирует бизнес-требования, соединяет команду исполнителей с заказчиком и следит за развитием проекта.

Одним из наиболее важных элементов Scrum является бэклог продукта, состоящий из собранных владельцем продукта пользовательских историй на основе информации, полученной от заинтересованных лиц со стороны заказчика, а также ранжированной командой с точки зрения приоритета и сложности реализации. Для определения приоритета, сложности задач и анализа необходимости реализации функций производится покер планирования.

В рамках первой встречи по завершении наполнения бэклога продукта производится мероприятие по планированию спринтов, а именно: выбор задач, в последствии включаемых в бэклог спринта на основе оценки размера (в пунктах или идеальных днях), в некоторых случаях группировка в темы, определение длительности и скорости выполнения работ. Выбор приоритета выполнения задач складывается из комментариев заказчиков, бизнес-логики и преимущества реализации базовых функций.

В процессе определения длительности спринта участники чаще всего руководствуются наличием опыта на аналогичных проектах и в целом сложностью проекта. При выборе спринта с минимальной продолжительностью в одну неделю отрицательным фактом может являться обязательство частого представления инкрементов заказчику, которое подразумевает большие временные затраты на подготовку к презентации части готового продукта и самой презентации. Однако положительной стороной короткого спринта является гибкость разработки и, как следствие, более качественное управление проектной деятельностью. В свою очередь

месячный спринт, наоборот, позволяет сконцентрироваться на работе и избежать часть обязательных отчётных мероприятий Scrum. Данные плюсы и минусы взвешиваются и принимается решение в зависимости от ситуации.

Стоит отметить, что принципы методологии позволяют вносить изменения в бэклог продукта непосредственно в процессе реализации на любом из этапов. Такие изменения и ввод новых требований должны быть своевременно отражены в очереди бэклога продукта. Важной особенностью метода является время создания бэклогов. Бэклог продукта создается на самом первом планировании спринта, а бэклог спринта должен создаваться командой на каждом планировании нового спринта. Таким образом, первый бэклог существует на протяжении всей разработки продукта, а бэклог спринта – в рамках текущего спринта с заданной продолжительностью.

На протяжении первого спринта команда реализует поставленные на первую встречу задачи, скрам-мастер консультирует участников в части ведения проекта по Scrum, а владелец продукта контролирует выполнение работ и консультирует в технической части. По окончании спринта производится презентация работающего продукта заказчику, а также обзор и ретроспектива спринта. Задача двух последних мероприятий заключается в оценке эффективности команды за время итерации, прогнозировании ожидаемой скорости и как следствие производительности участников в следующей, оценке или переоценке бэклога предстоящего спринта, выявлении имеющихся проблем и проведении оценки вероятности завершения всех необходимых работ вовремя. По завершении всех описанных процедур начинается новый спринт. Исходя из вышеописанного концептуальное представление ведения Scrum-проекта отображено на рисунке 1.2.

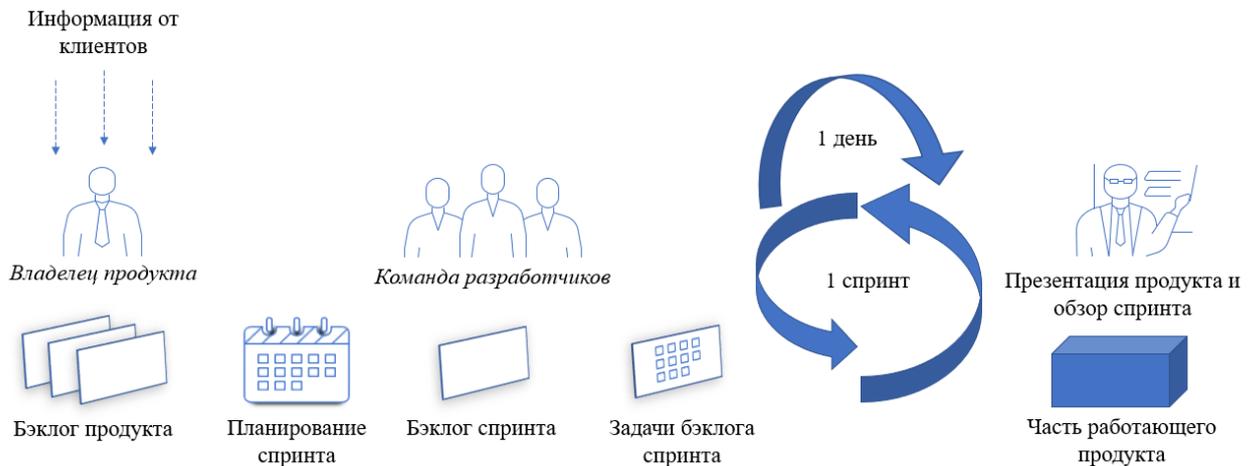


Рисунок 1.2 – Процесс реализации Scrum-проекта

Scrum и его принципы позволяют достичь максимальной гибкости, прозрачности, простоты и частоты отчётов о проделанной работе в работе над проектом в целом, то есть произвести переоценку приоритетов и внедрить новые функции или наоборот отказаться от них.

1.2. Анализ деятельности по учёту договоров

Вернемся к предметной области, под договором понимается сделка, выражающая согласованную волю двух или более лиц и их действия, направленные на установление, изменение или прекращение гражданских прав и обязанностей, то есть обязательств [5]. В ГОСТ Р 7.0.8-2013 о делопроизводстве и архивном деле дано следующее определение учёта архивных документов. Учёта архивных документов – определение количества и состава архивных документов в единицах учета и отражение этого количества и состава в учетных документах для контроля за их наличием и состоянием [6].

Под учётом поднимается отражение хозяйственной или иной деятельности предприятия на основании документов в различных измерителях (количественных и (или) качественных) [7]. Таким образом, под учётом

договоров можно понимать деятельность по поддержанию актуальной информации о статусе договора в целом и информации о контрагенте в частности в рамках договорных обязательств. Иными словами, учёт договоров – это фиксация всей информации, связанной с заключением договоров. С правовой точки зрения, данный вид деятельности не регулируется, однако во многих компаниях разрабатываются и применяются внутренние нормативные документы (договорные регламенты), которые отражают порядок договорной работы.

1.3. Анализ нон-код платформ

Теперь рассмотрим более подробно нон-код платформы. С момента появления первых нон-код платформ их возможности значительно продвинулись вперёд. В последние пять лет особенно явно заметно повышение интереса к нон-код разработке. Существует мнение, что этот факт связан с самим IT-рынком и его спецификой: сфере всегда требуется большое количество высококвалифицированных специалистов, которых не хватает, а также высокой стоимостью их работы. Поэтому разработчики начали развивать платформы для автоматизации работы самих программистов. Нон-код не панацея и использование технологии имеет ряд недостатков, связанных с изменением культуры организации, продолжительностью изучения для набора необходимого уровня знаний, стоимостью и ограниченностью поддержки и ресурсов сообщества, особенно для наиболее «молодых» платформ. [8].

Однако нон-код платформы имеют ряд преимуществ. Они позволяют пользователям без опыта программирования создавать веб-сайты, приложения и организовать взаимодействие, в том числе обмен информацией, под собственные нужды. Несмотря на то, что нон-код-инструменты не способны решить абсолютно все задачи разработки программных проектов, подбор и

использование адекватного стека подобных инструментов, без сомнения, упростит и ускорит решение некоторых задач, сделает доступнее то, от чего раньше, по разным причинам, отказывались, а в совокупности с применением гибкой методологии Scrum позволит получить максимально подходящий под запросы непосредственных пользователей работающий продукт. Проведем анализ существующих платформ для разработки «без кода». В таблице 1.1 представлено сравнение доступных наиболее популярных нон-код платформ.

Таблица 1.1 – Сравнение популярных нон-код платформ

Наименование платформы	Возможность применения платформы интеграции	Количество бесплатных записей	Стоимость корпоративного тарифа (долл./мес.)	Функциональность платформы для реализации функционала будущего приложения
Bubble	нет	до 2500	88	достаточна
Glide	нет	до 1000	90	недостаточно
Webflow	нет	до 2500	73	достаточна
Adalo	нет	до 3000	70	достаточна
Bravo Studio	есть	до 3500	100	достаточна
HoneyCode	есть	до 4500	70	достаточно
Thunkable	нет	до 3000	85	достаточна

Исходя из данных таблицы, можно сделать вывод о целесообразности разработки на двух платформах: Bravo Studio и HoneyCode. При этом минимальный тариф при потенциальном масштабировании и как следствии увеличении записей, предлагается компанией Amazon в продукте HoneyCode.

1.4. Анализ применения Agile Scrum для реализации корпоративных информационных систем

В настоящее время под реализацией корпоративных информационных систем (далее – КИС) понимается настройка и кастомизация готового пакетного решения «из коробки» [9]. В большинстве проектов по внедрению КИС, в частности ERP-систем от компаний SAP, Oracle и Microsoft и отечественного вендора 1С, используется, ставшая классической каскадная модель внедрения информационных систем (далее – ИС). Однако, несмотря на ориентированность на управление проектами по разработке программного обеспечения, также большой популярностью пользуется метод Scrum. Scrum в соответствии со статьей [10] подходит не для всех видов проектов. Приемлемых результатов можно достичь только в проектах по развитию ERP отдельно для настройки и разработки, но не внедрения «с нуля» и тиражирования.

В части доработки (кастомизации) положительный эффект применения Scrum возможен за счёт выбора лишь одного способа организации и программирования, который удовлетворяет конечному пользователю. Стоит отметить, что кастомизация решения весьма ограничена, в свою очередь варианты настройки либо носят единичный характер, либо очень близки, из чего следует, что при настройке ИС ситуация противоположна. Демонстрация промежуточного продукта пользователю кардинально не может изменить выбранного решения ввиду его детерминированности. В итоге, конечный пользователь мало на что может повлиять, а сам метод Scrum обеспечивает лишь быструю доставку решения без какой-либо существенной обратной связи [9].

Как и любая система, по определению, КИС является совокупностью программных продуктов. Допускается проектирование и реализация её при

помощи нескольких программных средств, решающих основные задачи данного класса систем, а именно по автоматизации процессов и принятию решений на основе накопленных данных. В отличие от вышеописанных доводов о преимуществах и недостатках применения Agile Scrum в процессе реализации КИС, при разработке предполагаемого приложения по учёту договоров, разработка будет вестись «с нуля», из чего следует возможность применения метода в полной мере со всеми его признанными плюсами.

Таким образом, концептуально последовательность действий для выполнения Scrum-проекта будет происходить в следующей последовательности:

- собрать информацию от заказчика;
- сформировать бэклог продукта;
- выполнить приоритизацию бэклога;
- разбить бэклог на спринты;
- реализовать первый спринт;
- презентовать работающую часть приложения заказчику
- провести встречу с целью оценки скорости выполнения работ (при необходимости выполнить переоценку спринтов);
- возвращаться на пятый этап до тех пор, пока не будет разработан продукт в полном объёме.

Раздел 2. Идентификация требований

Для рассмотрения на условно-реальном примере в последующей работе предлагается взять компанию с кодом ОКВЭД 85.42.9 (деятельность по дополнительному профессиональному образованию прочая, не включенная в другие группировки). Эта группировка включает разного рода курсы, а также обучение работодателей и работников по охране труда, подготовку, переподготовку, повышение квалификации и дополнительное профессиональное образование специалистов [11]. Организация принадлежит к компании малого бизнеса, 70 % сотрудников – преподавательский состав, остальные – сотрудники финансового, правового отделов, отдела маркетинга, административного отдела, а также директор компании.

2.1. Специфика процесса управления договорами

В рамках работы будет рассматриваться наиболее критичный для компании процесс – процесс учёта договоров, входящий в блок управления договорами. В организации могут быть установлены два подхода работы с договорами: централизованный и децентрализованный. Первый вариант подразумевает наличие специального структурного подразделения или установленной рабочей группы, которая занимается договорами. При этом задача исполнителя заключается лишь в доставке проекта договора уполномоченным сотрудникам, а согласованием, учётом и хранением занимаются специалисты. В свою очередь децентрализованный подход предполагает самостоятельное ведение конкретным исполнителем своего договора. В рассматриваемой компании применяется второй подход, зафиксированный в договорном регламенте, где описаны зоны ответственности и зависимости мероприятий (таблица 2.1).

Таблица 2.1 – Зоны ответственности и зависимости мероприятий в процессе работы с договорами

Шаг	Структурное подразделение	Ответственный	Мероприятие	Зависимость мероприятий
1. Заключение договора				
1.1	Маркетинговый отдел	Менеджер по работе с клиентами	Создание нового/ продление/ расторжение договора	
1.2	Правовой отдел	Юрист	Согласование	1.1
1.3		Генеральный директор	Визирование	1.2
1.4	Маркетинговый отдел	Менеджер по работе с клиентами	Направление договора контрагенту	1.3
1.5		Контрагент	Подписание второй стороной	1.4
2. Исполнение договорных обязательств				
2.1	Маркетинговый отдел	Менеджер по работе с клиентами	Уведомление ответственных, участвующих в исполнении	1.5
3. Учёт и хранение				
3.1	Маркетинговый отдел	Менеджер по работе с клиентами	Подготовка и передача договора в архив	1.5
3.2	Маркетинговый отдел	Менеджер по работе с клиентами	Запись в книге учёта договоров	3.1

Рассматриваемый процесс делится на три этапа: подготовительный, на котором заключается (или продлевается, или расторгается) и согласуется контракт, этап исполнения договорных обязательств, на котором происходит уведомление ответственных сотрудников, а также этап учёта и хранения. Как итог анализа процесса, можно сделать вывод, что его особенностями являются: чёткая, зафиксированная регламентом, последовательность действий между четырьмя видами исполнителей, юридический статус договора, требование к оперативности уведомления и передачи информации ответственным за конкретное мероприятие сотрудникам.

2.2. Формирование бэклога продукта

Для лучшего понимания бизнес-процессов владельцы продукта стараются максимально погрузиться в деятельность компании, используя информацию, полученную от непосредственных пользователей, открытых источников и собственного опыта. Для идентификации требований к разрабатываемому продукту владельцы продукта применяют определённые механизмы сбора этих требований. Наиболее часто использующийся – предоставление в письменной форме стороной заказчика описания того, чего они хотят получить от продукта в виде: «Я как **должность/роль** хочу **требование** потому что **причина**». Подобный подход позволяет определить примерные характеристики продукта, поскольку в большинстве случаев при устном опросе конечные пользователи не могут описать конкретные функции продукта.

Вышеописанные мероприятия являются частью подготовительного этапа, от которого зависит успешность на пути реализации проекта. В ходе рассмотрения специфики процесса управления договорами было выявлено четыре роли потенциальных пользователей будущего приложения: ге-

неральный директор, менеджер по работе с клиентами, юрист и главный бухгалтер. В качестве ключевого процесса выбрана операция учёта договоров.

В рамках специфики ОКВЭД также были выявлены общие требования к приложению, такие как: точность внесённой информации, возможность работы на разных устройствах одновременно (мультиплатформенность). По окончании сбора пользовательских историй владелец продукта представил сформулированные пользовательские истории от заинтересованных отделов, представленных в таблице 2.2.

Скрам-команда дала оценку каждой из историй с использованием чисел Фибоначчи (1, 2, 3, 5, 8), где 8 – история, реализация, которая подразумевает наиболее массивный объём работ, 1 – история, не требующая больших затрат для реализации. Также истории были расставлены в приоритетном порядке в соответствии с бизнес-требованиями со стороны заказчика.

Таблица 2.2 – Оценка пользовательских историй

№	Пользовательская история	Оценка сложности
1	Как генеральный директор я хочу иметь доступ к приложению с мобильных устройств (планшет, телефон) для ускорения процесса согласования и первичного ознакомления с данными договора	1
2	Как менеджер по работе с клиентами я хочу иметь доступ к приложению с мобильного устройства (планшета, телефона) с целью внесения данных в любых условиях	1
3	Как менеджер по работе с клиентами я хочу иметь возможность вводить информацию, соответствующую пунктам утвержденного стандартного печатного бланка договора организации для учёта договоров	8
4	Как менеджер по работе с клиентами я хочу иметь возможность удаления данных о договоре и о контрагенте для удаления ошибочно внесённой информации	8
5	Как менеджер по работе с клиентами я хочу внести информацию об ответственном за исполнение договорных обязательств с целью его оперативного оповещения	5

№	Пользовательская история	Оценка сложности
6	Как генеральный директор я хочу иметь возможность получать уведомления о статусе договоров для оперативного принятия решений и общей осведомленности	5
7	Как менеджер по работе с клиентами я хочу иметь возможность получать автоматические уведомления за неделю до окончания договора для проработки вопроса о продлении или конечном завершении контракта	5
8	Как генеральный директор я хочу, чтобы непосредственный исполнитель по договору получали уведомления о поступившем договоре или статусе уже имеющегося	5
9	Как юрист, я хочу оперативно получать уведомления о поступивших на меня договорах и о их статусе для понимания приоритета выполнения своих задач	5
10	Как менеджер по работе с клиентами я хочу иметь возможность применения фильтров и сортировки для ориентации среди имеющихся договоров	5
11	Как менеджер по работе с клиентами я хочу иметь возможность поиска по уникальному значению для поиска конкретного договора среди всех остальных	5
12	Как генеральный директор я хочу знать какой сотрудник был определён ответственным за договор с целью осведомленности о занятости сотрудников	3
13	Как генеральный директор я хочу иметь возможность определять уровень доступа для сотрудников из соображения информационной безопасности	1
14	Как генеральный директор я хочу импортировать имеющиеся записи в электронном формате (.CSV) для учёта прошлых договоров	1
15	Как менеджер по работе с клиентами я хочу иметь возможность прямого доступа к скану физического договора для использования в работе	3

Как можно видеть из таблицы, наиболее приоритетными историями для заинтересованного лица являются пользовательские истории, связанные с доступом с разных устройств для всех представленных ролей пользователей, а также внесение информации и изменение внесённой записей. С точки зрения

разработки, данные истории являются базовыми для будущего приложения и требуют первостепенной проработки. Вторая группа историй связана с отслеживанием и получением статуса договора, требованиями реализации возможности ориентации в списках, определением уровня доступа и с выгрузкой данных. После окончания формирования бэклога продукта начинается этап планирования ближайших итераций (спринтов), их наполнение и определение их продолжительности.

2.3. Формирование спринтов

Как было сказано ранее в таблице 2.2, можно заметить несколько логических групп пользовательских историй. Такие группы называют темами. Они позволяют легко ориентироваться в требованиях к будущему продукту. В таблице 2.3 представлены сгруппированные истории в темы.

Таблица 2.3 – Группировка пользовательских историй по темам

История	Тема
Как генеральный директор я хочу иметь доступ к приложению с мобильных устройств (планшет, телефон) для ускорения процесса согласования и первичного ознакомления с данными договора	Вход с любого устройства, у которого есть доступ к интернету
Как менеджер по работе с клиентами я хочу иметь доступ к приложению с мобильного устройства (планшета, телефона) с целью внесения данных в любых условиях	
Как менеджер по работе с клиентами я хочу иметь возможность вводить информацию, соответствующую пунктам утвержденного стандартного печатного бланка договора организации для учёта договоров	Внесение и удаление информации
Как менеджер по работе с клиентами я хочу иметь возможность удаления данных о договоре и о контрагенте для удаления ошибочно внесённой информации	

История	Тема
Как менеджер по работе с клиентами я хочу внести информацию об ответственном за исполнение договорных обязательств с целью его оперативного оповещения	
Как менеджер по работе с клиентами я хочу иметь возможность прямого доступа к скану физического договора для использования в работе	
Как генеральный директор я хочу иметь возможность получать уведомления о статусе договоров для оперативного принятия решений и общей осведомленности	Отслеживание и получение статуса договора и об ответственном
Как генеральный директор я хочу, чтобы непосредственный исполнители по договору получали уведомления о поступившем договоре или статусе уже имеющегося	
Как юрист, я хочу оперативно получать уведомления о поступивших на меня договорах и о их статусе для понимания приоритета выполнения своих задач	
Как менеджер по работе с клиентами я хочу иметь возможность получать автоматические уведомления за неделю до окончания договора для проработки вопроса о продлении или конечном завершении контракта	
Как генеральный директор я хочу знать какой сотрудник был определён ответственным за договор с целью осведомленности о занятости сотрудников	
Как менеджер по работе с клиентами я хочу иметь возможность применения фильтров и сортировки для ориентации среди имеющихся договоров	Ориентации в списках
Как менеджер по работе с клиентами я хочу иметь возможность поиска по уникальному значению для поиска конкретного договора среди всех остальных	
Как генеральный директор я хочу иметь возможность определять уровень доступа для сотрудников из соображения информационной безопасности	Определение уровня доступа для сотрудников
Как генеральный директор я хочу импортировать имеющиеся записи в электронном формате (.CSV) для учёта прошлых договоров	Импорт данных

Бэклоги спринтов требуют конкретизации, в следствии чего, нужно указать конкретные технические задачи по их реализации. Кроме того, скрам-команде необходимо определить временные затрат на выполнение задач, измеряемые в часах, и разбивку задач на спринты (таблице 2.4).

Таблица 2.4 – Определение задач и времени реализации задач в рамках тем

Тема	Задачи	Часы на задачи
Вход с любого устройства, у которого есть доступ к интернету	Нахождение наиболее подходящего решения (выбор нон-код платформы)	5
	Изучение документации по использованию платформы	7
Внесение и удаление информации	Реализация базы данных, соответствующим экранам	20
	Реализация интерфейса экранов	13
Отслеживание и получение статуса договора и об ответственном	Настройка всех требующихся видов уведомлений	25
	Проведение функционального тестирования	7
Ориентации в списках	Создание и настройка полей для фильтрации	5
Определение уровня доступа для сотрудников	Сбор информации с целью заведение учётных записей пользователей	5
	Заведение учётных записей пользователей в приложении	3
	Настройка уровня доступа	3
Импорт данных	Приведение накопленных данных к единой форме и их импорт	5

На этапе формирования бэклога продукта было принято решение о выставлении наибольшей приоритетности для тех историй, которые

основывают скелет продукта и несут основной функционал приложения. После всех преобразований данная последовательность отражена в таблице 2.4. По итогу проработки вопроса с трудозатратами на реализацию поставленных задач и с учётом сложности проекта, было принято решение об определении продолжительности спринта в одну неделю при пятидневной рабочей неделе и продолжительности рабочего дня в восемь часов. Для достижения наилучшего результата спринты необходимо делить на приблизительно равные части, с точки зрения временных затрат. Поэтому задачи были сформированы в спринты таким образом, как они представлены в таблице 2.5.

Таблица 2.5 – Разбивка задач по спринтам

Спринт	Тема	Дополнительные задачи
I	Вход с любого устройства, у которого есть доступ к интернету	Проектирование процессов, данных и приложения в AS-IS и TO-BE моделях, функциональное тестирование
II	Внесение и удаление информации	Функциональное тестирование
III	Отслеживание и получение статуса договора и об ответственном	Функциональное тестирование
IV	Ориентации в списках, определение уровня доступа для сотрудников, импорт данных	Функциональное и нагрузочное тестирования

Как можно увидеть, в таблице 2.5 в отличие от 2.4, на первой итерации были дополнены такие задачи как проектирование моделей AS-IS и TO-BE, архитектуры данных и интерфейса, а также активности по проведению функционального тестирования по окончании каждого спринта. Задачи, связанные с проектированием процессов, базы данных и интерфейса

необходимы для документирования существующего бизнес-процесса, понимания требований к будущему результату и структуры базы данных, приблизительного представления о будущем приложении. В свою очередь функциональное тестирование необходимо для проверки быстродействия и проверки работоспособности продукта.

Метод Scrum позволяет декомпозировать задачи и переносить на следующие спринты те задачи, которые не помещаются в рамках одного спринта. Было предпринято решение о разделении задачи реализации базы данных и пользовательского интерфейса по соответствующим экранам главного меню и заключение договоров в рамках первого спринта; и создание базы данных и интерфейса для карточки контрагента, адреса организации и удаления сведений в рамках второго спринта.

Из таблицы следует, что в первых трёх спринтах на выполнение задач в рамках одного спринта планируется тратить по 32 часа. Данный показатель определён из учёта, что оставшиеся рабочие часы будут потрачены на обязательные скрам-мероприятия. Стоит отметить, что согласно принципам Scrum не стоит прикладывать большое количество усилий по планированию всех итераций за раз, необходимо обращать внимание на ближайшие и наиболее важные задачи в рамках предстоящего спринта. Также Scrum предполагает, что после каждой итерации на ретроспективе команда оценивает свои показатели выполнения работ и при необходимости проводит корректировку бэклогов спринтов.

Таким образом, перед началом завершающего спринта и при отсутствии условия обязательной реализации проекта в определённый срок, четвёртый спринт может быть скорректирован и выполнен за 32 часа.

Раздел 3. Первый спринт: проектирование ключевых бизнес-процессов

В рамках задач первого спринта, были выделены требования по проектированию базы данных будущего приложения, бизнес-процессов и интерфейсов. Для проектирования бизнес-процессов был проведён сбор информации о деятельности организации, определён набор операций, которые выполняют критическую организационную цель. Для более детального понимания командой разработчиков, какого результата должны добиться при использовании приложения по учёту договоров, требуется графическая фиксация процессов организации в моделях «как есть» и «как должно быть».

Для проектирования и отображения в графическом виде концептуального представления «AS-IS» и «TO-BE» существует множество нотаций проектирования бизнес-процессов. В данной работе была выбрана нотация ARIS VACD для описания верхних уровней и для детализации нижних уровней – BPMN SLD.

ARIS является одной из многих методологий моделирования бизнес-процессов, одной из графических нотаций ARIS является схема VACD. Основным объектом нотации VACD – процесс или группа функций организации, необходимые для получения добавленной стоимости. Элементы, используемые для моделирования в VACD, представлены в таблице 3.1.

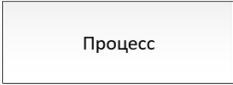
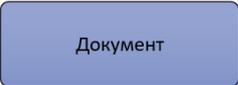
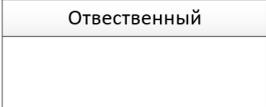
Таблица 3.1 – Элементы нотации ARIS VACD

Графический элемент	Описание
	Информация или материал

Графический элемент	Описание
	Процесс
	Ответственный

Для описания процессов нижних уровней организации будет применена графическая нотация для отображения бизнес-процессов в виде диаграмм бизнес-процессов BPMN SLD, элементы которой даны в таблице 3.2.

Таблица 3.2 – Графические элементы нотации BPMN SLD

Графический элемент	Описание
	Процесс
	Документ
	Условие
	Ответственный/ организационный уровень

Стоит учесть, что на момент разработки приложения, никаких программных средств для учёта договоров в компании не имеется, вся документация ведётся в бумажном виде.

3.1. Проектирование процессов AS-IS в нотациях ARIS VACD и BPMN SLD

На рисунке 3.1 представлена диаграмма функционирования организации «как есть» в нотации ARIS VACD.

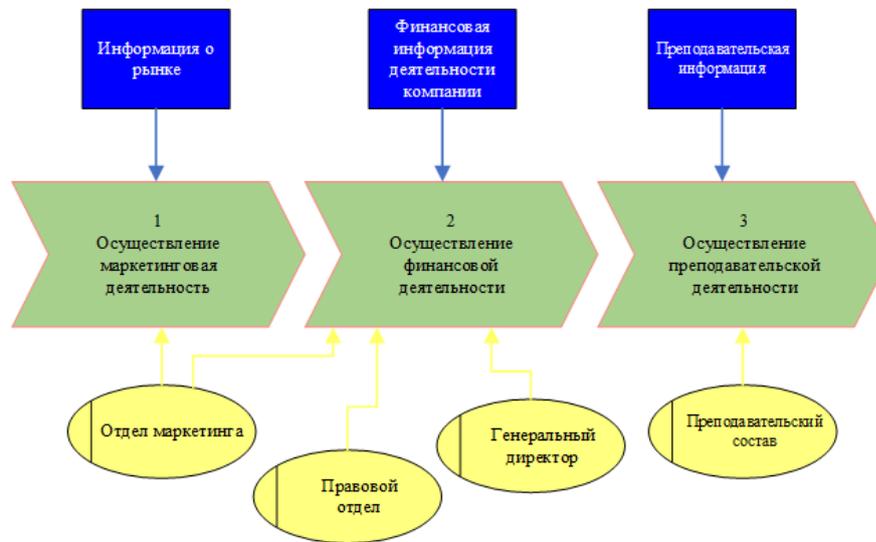


Рисунок 3.1 – Модель уровня управления предприятием

Данные процессы являются основными процессами в деятельности организации. Диаграмма отображает верхнеуровневое представление о процессах, зоне ответственности и данных, с которыми они работают. Процесс «Осуществление финансовой деятельности» необходимо детализировать и рассмотреть подробнее, поскольку в нем заложены подпроцессы следующих уровней декомпозиции, требующие более пристального анализа. Декомпозиция бизнес-процесса представлена на рисунке 3.2.



Рисунок 3.2 – Декомпозиция процесса «Осуществление финансовой деятельности»

Процессы финансовой деятельности разделяются на три основные подпроцесса: осуществление деятельности по заключению, продлению или расторжению договоров. Дальнейшая декомпозиция будет осуществлена в нотации BPMN SLD. Рассмотрим результат декомпозиции «Осуществление деятельности по заключению договоров», представленного в Приложении А, как наиболее встречающегося в рамках финансовой деятельности компании. Можно видеть четыре пула ответственных, имеющих отношение к деятельности по заключению договоров: «менеджер по работе с клиентами», «юрист», «генеральный директор» и пул ответственного «контрагент». В процессе создаются документы «проект договора» и «договор». Проект договора не имеет юридической ценности, тем не менее является промежуточным документом, с которым происходит работа и поэтому отражён в рассматриваемой модели. Договором он начинает являться после подписания двумя сторонами. Бизнес-процесс создания проекта договора детально показан на рисунке 3.3.

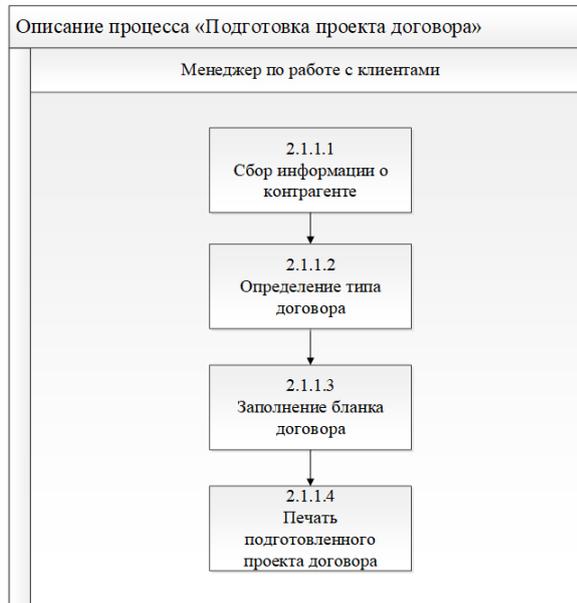


Рисунок 3.3 – Декомпозиция процесса «Подготовка проекта договора»

Деятельность менеджера по работе с клиентами в рамках рассматриваемого процесса заключается в получении информации, такой как: наименование организации, юридическое местонахождение, банковские реквизиты, информации о требуемой услуге, заполнении полученной информацией бланк и как результат печати проекта договора. Для учёта коммерческих договоров ведётся журнал, содержащий информацию о статусе исполнения договора и информацию нахождения в архиве. На рисунке 3.4. представлена декомпозиция процесса «Подготовка договора для хранения».

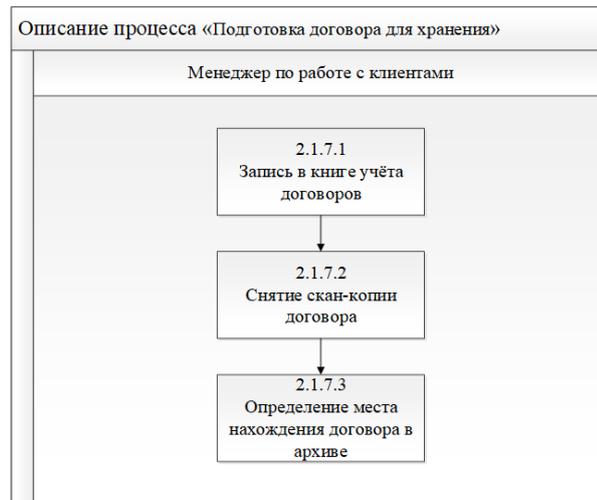


Рисунок 3.4 – Декомпозиция процесса «Подготовка договора для хранения»

В контексте этого процесса менеджер по работе с клиентами как было сказано ранее, вносит информацию в книгу учёта договоров, после чего снимает копию для передачи на ознакомление непосредственным исполнителям и определяет место нахождения договора в архиве. Данный процесс имеет несколько значительных недостатков:

- невозможность оперативного доступа к документам в виду удалённости и обособленности помещения архива;
- вероятность упущенной выгоды, связанной с отсутствием своевременной работы с контрагентом по вопросу продления договора по причине отсутствия оповещений о скором окончании договора.

В рамках всего процесса по осуществлению деятельности по заключению договоров, также можно выделить один большой недостаток, такой как отсутствие системы оперативных оповещений необходимых сотрудников, задействованных в процессе. Перечисленные недостатки возможно решить путем разработки приложения по учёту договоров.

3.2. Проектирование процесса «ТО-БЕ» в нотациях ARIS VACD и BPMN SLD

В Приложении Б представлена контекстная модель функционирования организации «как должно быть» в нотации ARIS VACD. Как можно видеть на рисунке, верхнеуровнево изменений не произошло. Также нет никаких изменений в модели на втором уровне декомпозиции, представленном в Приложении В. При дальнейшей декомпозиции процесса (Приложении Г), наблюдаются изменения, связанные с использованием разработанного приложения. Можно заметить увеличение числа операций за счёт появления приложения, отвечающего за рассылку уведомлений.

В рамках подпроцесса по подготовке проекта договора при декомпозиции, представленной на рисунке в Приложении Д, в отличие от аналогичной операции в модели AS-IS, были определены дополнительные активности, связанные с внесением информации в разрабатываемое приложение и определением статуса нахождения договора. При получении уведомления о рассмотрении проекта договора, юрист компании заходит в программу и открывает ссылку на файл, указанную в соответствующем поле договора, далее проверяет его на корректность. При наличии незначительных замечаний он может внести правки в электронный документ и выставить статус согласования. При наличии каких-либо серьёзных недочётов, допускается проставить статус о направлении на доработку с соответствующим комментарием, как показано на рисунке 3.5.

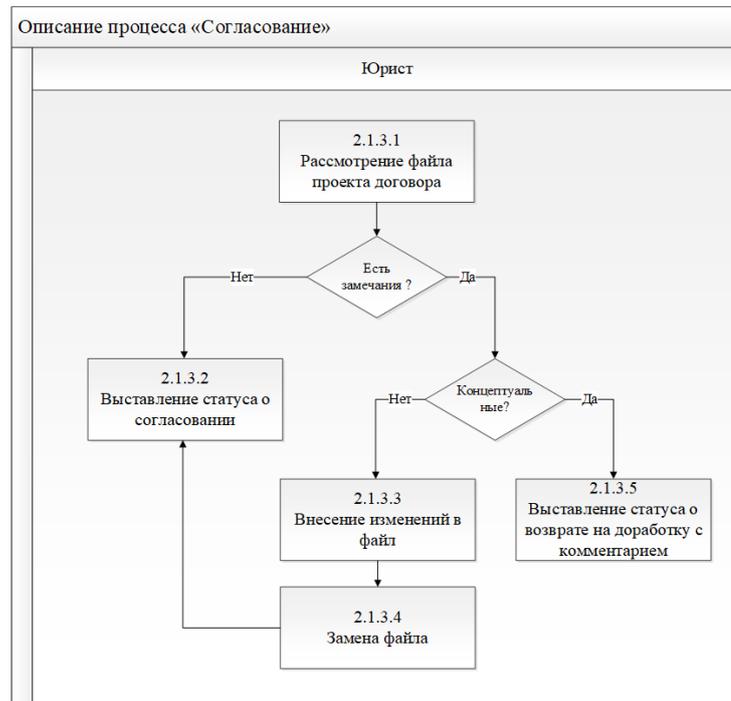


Рисунок 3.5 – Декомпозиция процесса «Согласование»

В ходе передачи проекта договора на визирование директору, отображённому на рисунке 3.6, осуществляется печать согласованного юристом проекта договора, подготовка документа для визирования и отметка статуса в программе по учёту договоров.



Рисунок 3.6 – Декомпозиция процесса «Передача проекта договора на визирование»

По окончании визирования директором компании документ отправляется контрагенту. Декомпозиция данного процесса отражена на рисунке 3.7.



Рисунок 3.7 – Декомпозиция процесса «Отправка документа контрагенту»

После получения визы директора компании менеджер по работе с клиентами готовит документ к хранению в архиве. Подпроцессы «Подготовка договора для хранения» даны на рисунке 3.8.

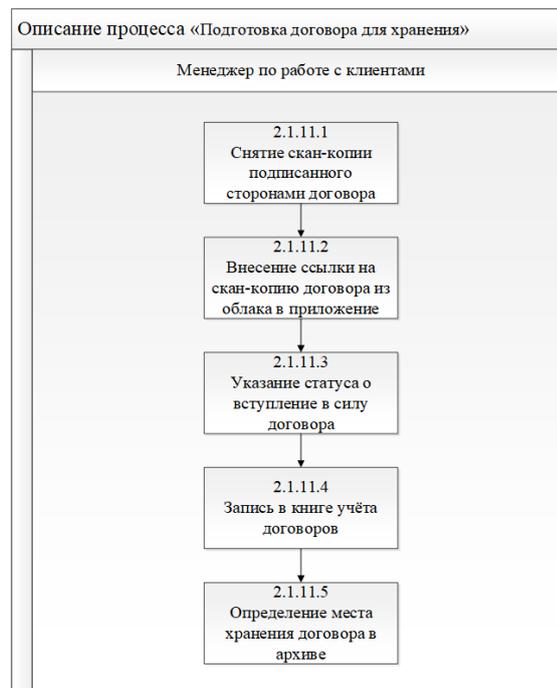


Рисунок 3.8 – Декомпозиция процесса «Подготовка договора для хранения»

Для лучшей наглядности представления и сравнения моделей «как есть» и «как будет» были созданы карты бизнес-процессов, представленные на рисунках 3.9 и 3.10.

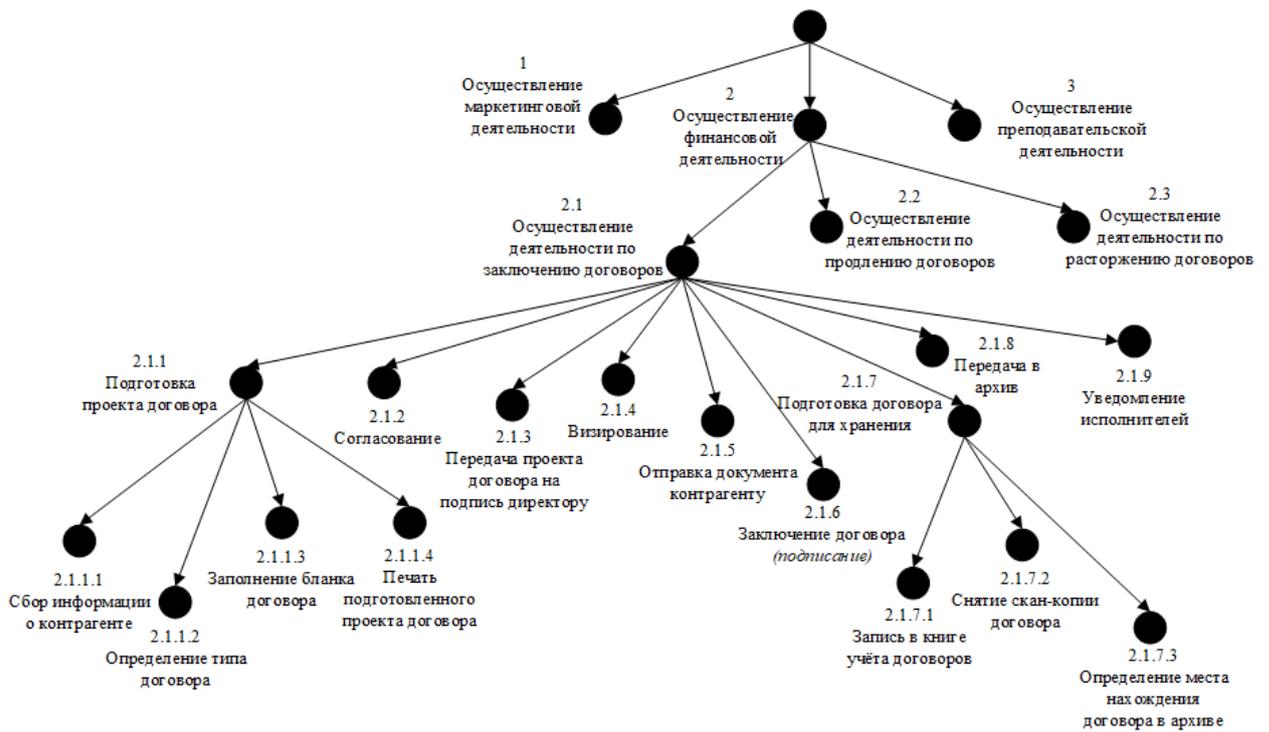


Рисунок 3.9 – Карта процессов в представлении модели AS-IS

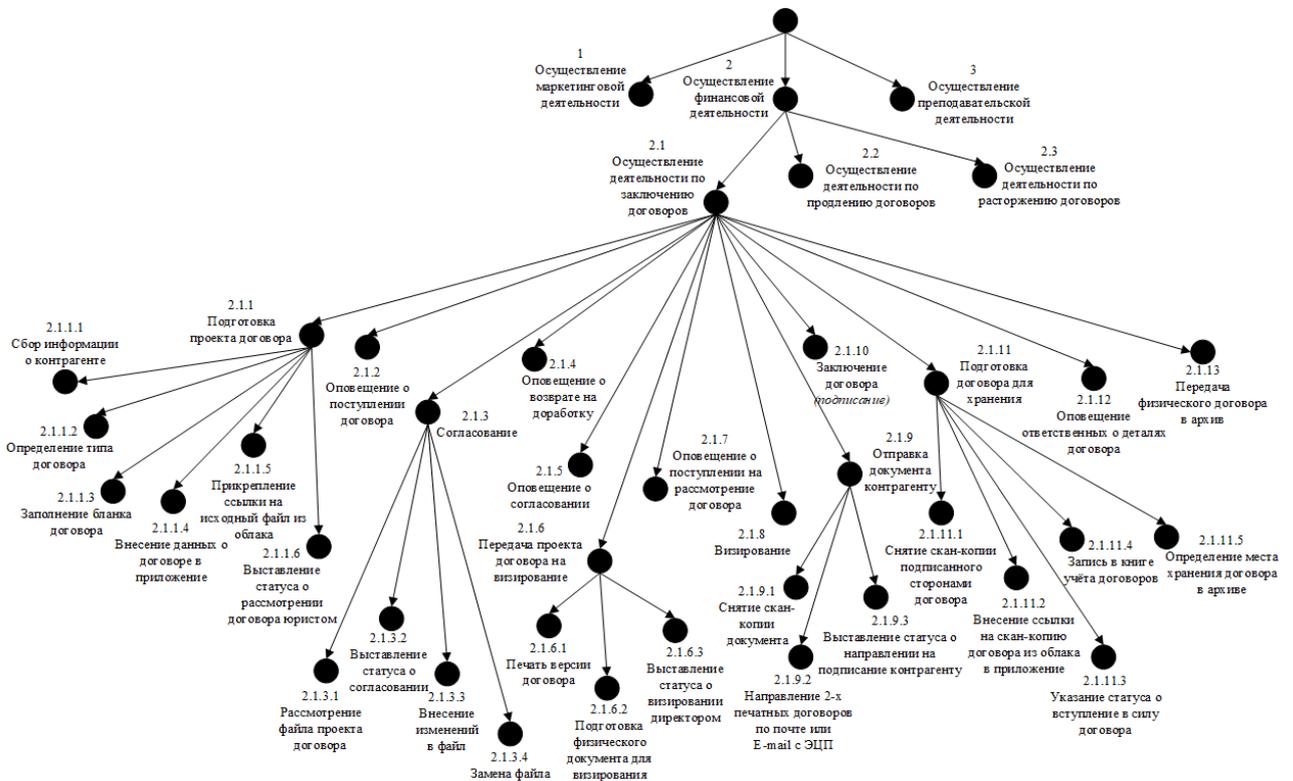


Рисунок 3.10 – Карта процессов в представлении модели TO-BE

При сравнении карт можно наблюдать значительные изменения в количестве операций из-за включения в работу разрабатываемого приложения. Несмотря на увеличение операций, часть из них происходят автоматически, общее время исполнения работ сокращается из-за оперативности исполнения задач в связи с автоматической отправкой уведомлений при изменении статуса договора. В процессе моделирования бизнес-процессов ТО-ВЕ были отмечены некоторые особенности. При окончании выполнения почти каждого подпроцесса на третьем уровне, ответственному сотруднику необходимо отмечать статус нахождения договора и осуществление функции уведомления в приложении.

3.3. Проектирование данных

При выполнении первого спринта была спроектирована база для наглядности представленная в таблице 3.3. Она отображает таблицы данных и входящие в них поля.

Таблица 3.3 – Данные для создания баз данных

Таблица	Наименование поля	Тип данных	Макс. длина	Ключ
Контрагент	Наименование	text	50	+
	Организационная форма	text	5	
	ИНН	text	12	
	КПП	text	12	
	Юридический адрес	text	60	
Адрес	Юридический адрес	text	60	+
	Страна	text	60	
	Город	text	60	
	Улица	text	60	
	Дом	text	15	
	Строение	int	5	

Таблица	Наименование поля	Тип данных	Макс. длина	Ключ
	Корпус	varchar	5	
	Кабинет	varchar	5	
Тип организационной формы	Организационная форма	text	5	+
Договор	Номер контракта	int	5	+
	ID Контрагента	int	5	
	ID Исполнителя	int	5	
	Тип договора	text	15	
	Дата начала	date	10	
	Дата окончания	date	10	
	Сумма	varchar	15	
	Валюта	text	3	
	Ссылка на файл проекта договора	text	255	
	Комментарий	longtext	500	
	Ссылка на скан-копию	text	255	
Тип валюты	Валюта	text	25	+
Тип договора	Тип договора	text	25	+
Исполнитель	Фамилия	text	20	+
	Имя	text	20	+
	Отчество	text	20	+
	Должность	text	25	+

В результате конструирования таблиц были выбраны размеры и типы данных для каждого поля, которое входит в определенный класс. Представленные данные были приведены к третьей нормальной форме. В данном случае будет использоваться база данных, содержащая семь таблиц: «Карточка контрагента», «Адрес», «Тип организационной формы», «Договор», «Тип договора», «Тип валюты» и «Исполнитель». Исходя из

таблиц нормализованных данных была спроектирована архитектура данных, представленная в Приложении Е.

3.4. Проектирование интерфейсов

В соответствии с потребностями, описанными сотрудниками компании заказчика и на основе сформулированных командой разработчиков требований, были спроектированы следующие экранные формы, данные на рисунках 3.11-3.16.

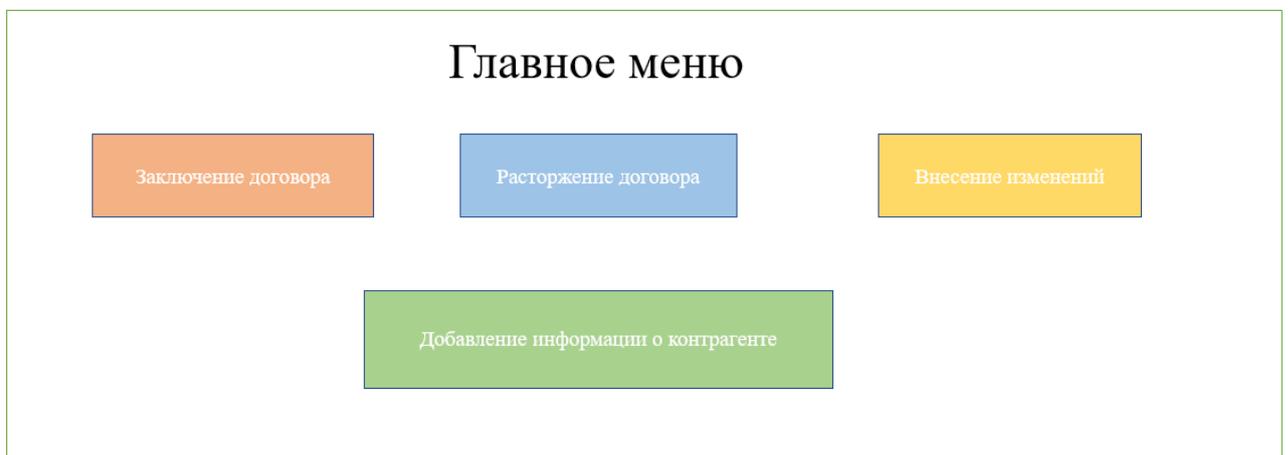


Рисунок 3.11 – Структура экрана главного меню

Заключение договора

[Добавить данные о контрагента](#)

Номер	<input type="text"/>
Тип	<input type="text"/>
Организация	<input type="text"/>
Дата начала	<input type="text"/>
Дата окончания	<input type="text"/>
Сумма	<input type="text"/>
Валюта	<input type="text"/>
Ответственный	<input type="text"/>

[Отмена](#)[Вернуться в Главное меню](#)

Рисунок 3.12 – Структура экрана заключения договора

Карточка контрагента

[Добавить адрес организации](#)

Организация	<input type="text"/>
Тип партнёра	<input type="text"/>
ИНН	<input type="text"/>
КПП	<input type="text"/>

[Отмена](#)[Вернуться в заключение договора](#)

Рисунок 3.13 – Структура экрана карточки контрагента

Адрес организации

Организация	<input type="text"/>
Страна	<input type="text"/>
Город	<input type="text"/>
Улица	<input type="text"/>
Дом	<input type="text"/>
Корпус	<input type="text"/>
Индекс	<input type="text"/>

Рисунок 3.14 – Структура экрана с информацией с адресом организации

Расторжение договора

Номер	Тип	Организация	Дата начала	Дата окончания	Сумма	Валюта	Ответственный
<input type="text"/>							
<input type="text"/>							
<input type="text"/>							
<input type="text"/>							

Рисунок 3.15 – Структура экрана расторжения договора

Для лучшего понимания о переходах между экранами и в принципе объёме приложения на рисунке 3.16 предложена структура приложения.

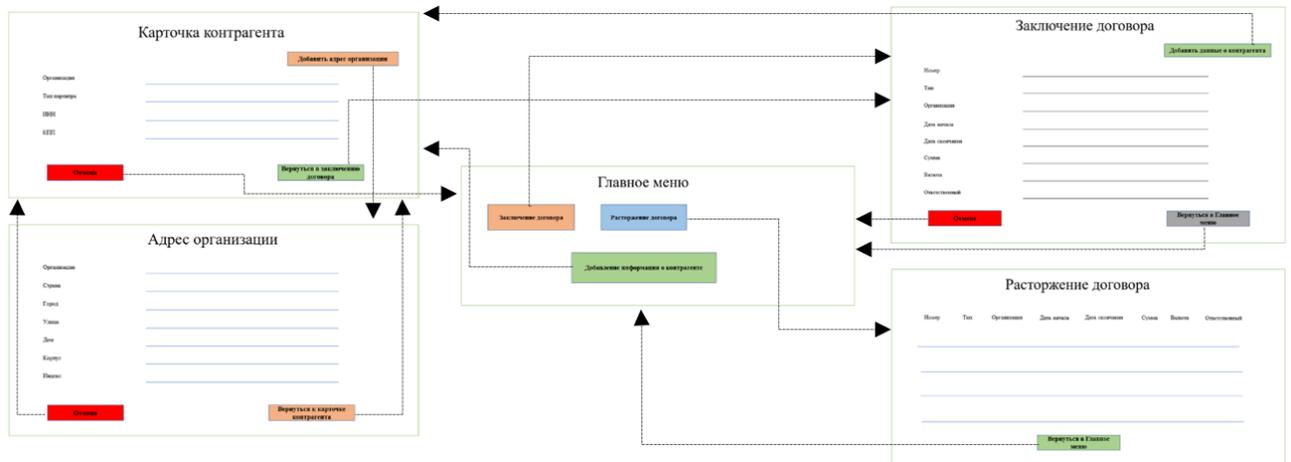


Рисунок 3.16 – Структура приложения

Раздел 4. Реализация основных функций приложения

4.1. Второй спринт: разработка каркаса приложения

На втором спринте была запланирована реализация одной темы, связанной с требованиями мультиплатформенности, которая реализуется автоматически при использовании выбранной по результатам сравнительного анализа в разделе первом non-code платформы HoneyCode. В рамках этой темы также необходимо ознакомиться с документацией и принципами её работы, реализовать часть базы данных и интерфейса соответствующим экранам главного меню и заключению договора. Фрагменты реализованных экранов приложения, открытого на ноутбуке и работающего через web представлены на рисунках 4.1 и 4.2.

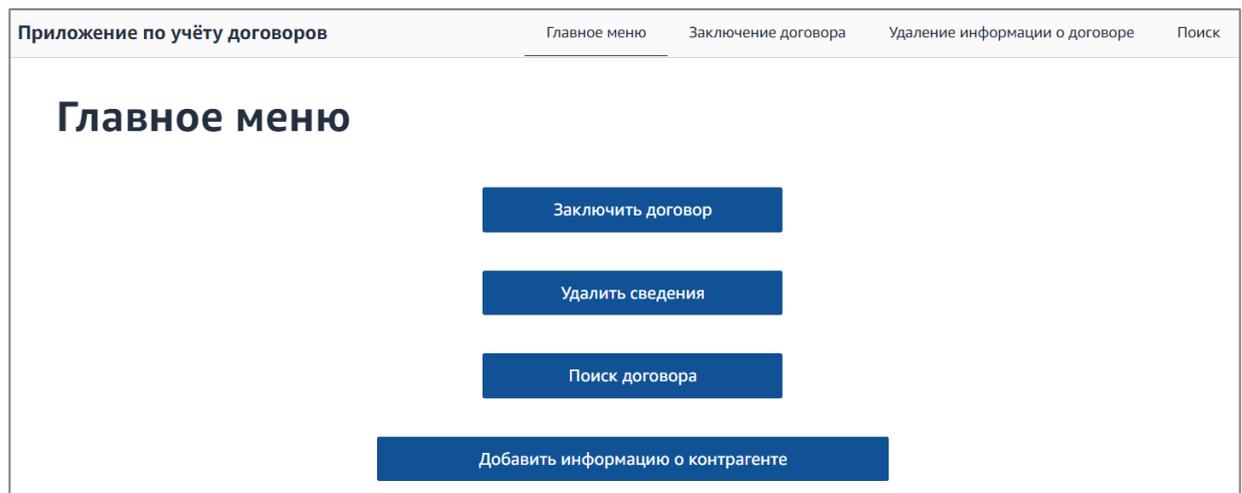


Рисунок 4.1 – Экран главного меню

При нажатии на кнопку «Заключить договор» осуществляется переход на экран создания нового договора, показанного на рисунке 4.2.

Заклучение договора

[Добавить данные о контрагенте](#)

Введите данные о договоре

Номер договора	<input type="text" value="Edit Value"/>
Тип договора	<input type="text" value="Select an option"/>
Контрагент	<input type="text" value="Select an option"/>
Дата начала	<input type="text" value="M/d/yyyy"/>
Дата окончания	<input type="text" value="M/d/yy"/>
Сумма	<input type="text" value="Edit Value"/>
Валюта	<input type="text" value="Select an option"/>
Ссылка на файл проекта договора	<input type="text" value="Edit Value"/>
Статус проект договора	<input type="text" value="Select an option"/>
Комментарий юриста	<input type="text" value="Edit Value"/>
Ответственный	<input type="text" value="Select a contact"/>
Ссылка на скан-копию	<input type="text" value="Edit Value"/>

[Вернуться в главное меню](#) [Сохранить](#)

Рисунок 4.2 – Экран приложения «Заклучение договора»

Создание нового договора подразумевает внесение информации о контрагенте, дате фактического заключения договора, сумме договора, денежных единицах и определение ответственного лица его заключившего. При отсутствии информации (в случае первичного заключения договора) в базе о контрагенте необходимо нажать кнопку «Добавить данные о контрагенте» с этого же экрана откроется новый подэкран «Карточка контрагента». Поле «Дата начала договора» имеет выпадающий календарь. Также есть возможность ввести дату числами. Вид заполнения представлен на рисунке 4.3.

Приложение по учёту договоров

Главное меню Заключение договора Удаление информации о договоре Поиск SHAR

Добавить данные о контрагенте

Введите данные о договоре

Номер договора: 425342

Тип договора: Select an option

Контрагент: Ростелеком

Дата начала: M/d/yyyy

Дата окончания: ДД.ММ.ГГГГ

Сумма: Edit Value

Валюта: Select an option

Calendar: June 2022

Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Рисунок 4.3 – Фрагмент экрана «Заключение договора», календарь для выбора даты начала договора

Область с суммой договора заполняется вручную. Допустимы значения с запятой обозначающие копейки, центы и евроценты. Поле обозначающее денежные единицы имеет три заданных значения обозначающие российские рубли, евро и доллары. Статус договора позволяет определить, на какой стадии договор сейчас находится, в последствии (при реализации задач следующих спринтов) при выборе актуального статуса будет отправляться уведомление нужному сотруднику о необходимости рассмотрения договора. В поле «Ответственный» можно назначить ответственного сотрудника путём определения из выпадающего меню ФИО нужного специалиста, как показано на рисунке 4.4.

Приложение по учёту договоров		Главное меню	Заключение договора	Удаление информации о договоре	Поиск	SHARE
Тип договора	Select an option					
Контрагент	Select an option					
Дата начала	M/d/yyyy					
Дата окончания	M/d/yy					
Сумма	Edit Value					
Валюта	Select an option					
Ссылка на файл проекта договора	Edit Value					
Статус проект договора	No Selection Самойлов Хмельёва Алборов					
Комментарий юриста						
Ответственный	Select an option					
Ссылка на скан-копию	Edit Value					

[Вернуться в главное меню](#) [Сохранить](#)

Рисунок 4.4 – Выбор ответственного лица для создания нового договора

4.2. Третий спринт: разработка базы данных и интерфейса для оставшихся экранов

На третьем спринте планируется доработка оставшихся экранных форм и соответствующие им части базы данных. Результат реализованных на данном спринте работ представлен на рисунках 4.5 – 4.7. На рисунке 4.5 – отображён экран «Карточка контрагента». Поля «Организация», «Тип партнёра», «ИНН» и «КПП» соответствуют полям, которые заполняются в юридически значимых договорах.

Карточка контрагента

Добавьте информацию о юридическом адресе организации

Добавить адрес организации

Организация	Select an option
Тип партнёра	Select an option
ИНН	Edit Value
КПП	Edit Value

Вернуться к заключению договора

Сохранить

Рисунок 4.5 – Экран приложения «Карточка контрагента»

При нажатии на поле «Организация» приложение открывает выпадающий список с имеющимися заведёнными в базе данных наименования организаций, как показано на рисунке 4.6 на основе карточек контрагентов. Также для удобства, имеется возможность подбора нужного названия контрагента при частичном заполнении данного поля. Аналогичный выбор даётся с полем «Тип партнёра».

Карточка контрагента

Добавьте информацию о юридическом адресе организации

Добавить адрес организации

Организация	ПАО "Ростелеком"
Тип партнёра	No Selection
ИНН	ПАО "Ростелеком"
КПП	ПАО "МОЭК"

Вернуться к заключению договора

Сохранить

Рисунок 4.6 – Предзаполненные поля контрагентов экрана создания нового договора

После заполнения всех полей карточки контрагента можно нажать на кнопку «Вернуться к заключению договора» и перейти на предыдущий экран. Для добавления информации о месте расположения компании необходимо нажать на кнопку «Добавить адрес организации», как показано на рисунке 4.6, далее откроется новая страница, отображённая на рисунке 4.7.

The screenshot shows a web application interface titled "Приложение по учёту договоров" (Contract Accounting Application). The main heading is "Адрес организации" (Organization Address). Below the heading, there are seven input fields, each labeled with a field name and "Edit Value":

Организация	Edit Value
Страна	Edit Value
Город	Edit Value
Улица	Edit Value
Дом	Edit Value
Корпус	Edit Value
Индекс	Edit Value

At the bottom of the form, there are two buttons: a red button labeled "Вернуться к карточке контрагента" (Return to counterparty card) and a blue button labeled "Сохранить" (Save).

Рисунок 4.7 – Экран приложения «Адрес организации»

Для связки данных карточки контрагента с его адресом используется поле «Организация». Приложение также имеет возможность удаления данных о договоре. Для этого необходимо в главном меню нажать на кнопку «Удаление данных», при этом происходит переход на соответствующий экран, показанный на рисунке 4.8.

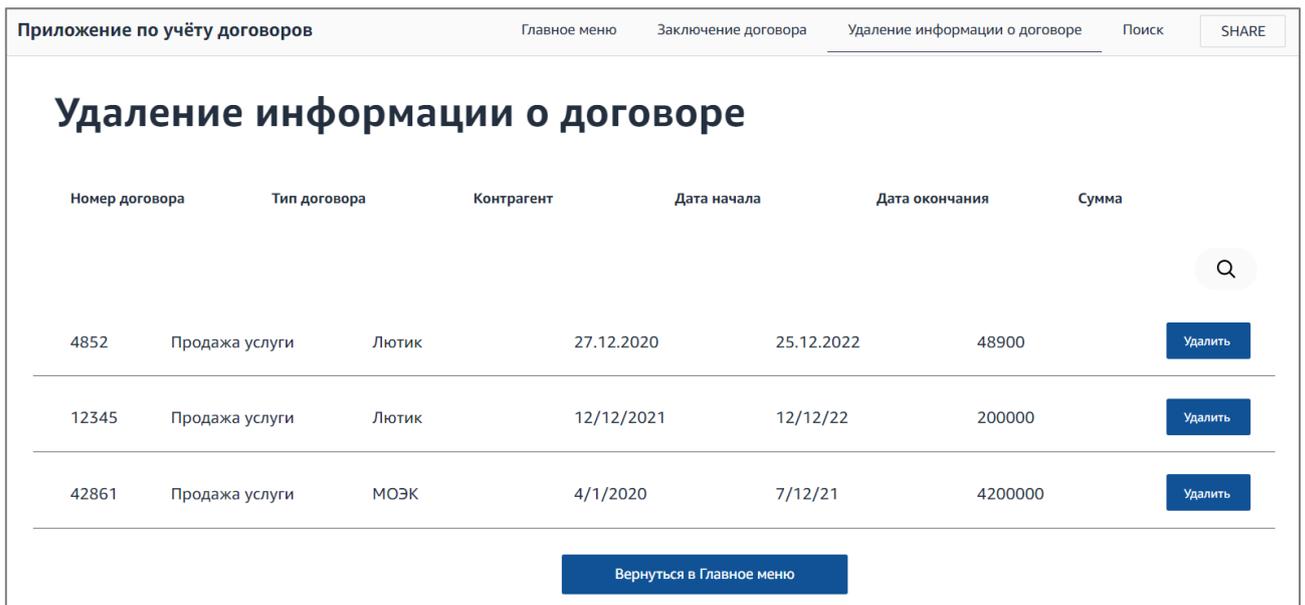


Рисунок 4.8 – Экран приложения «Удаление информации о договоре»

Для удаления договора необходимо нажать на кнопку «Удалить» напротив необходимой строки. Подобный функционал может потребоваться в случае некорректного заполнения информации, устаревших договорах или иной причине.

4.3. Четвёртый спринт: добавление функционала для фильтрации и поиска

При проведении четвертого спринта, должны быть реализованы функции фильтрации и импорт имеющихся на момент внедрения приложения данных. Данные работы отражены на рисунках 4.9 – 4.13. Для просмотра и фильтрации договоров была создана отдельная кнопка «Поиск договора» на экране «Главное меню», предназначенная исключительно для поиска нужного договора при помощи поисковой строки, которую можно заполнить значениями «Номер договора» и «Контрагент». Данный способ нахождения договора представлен на рисунке 4.9. На рисунке 4.10 показан способ фильтрации (упорядочивания) договоров.

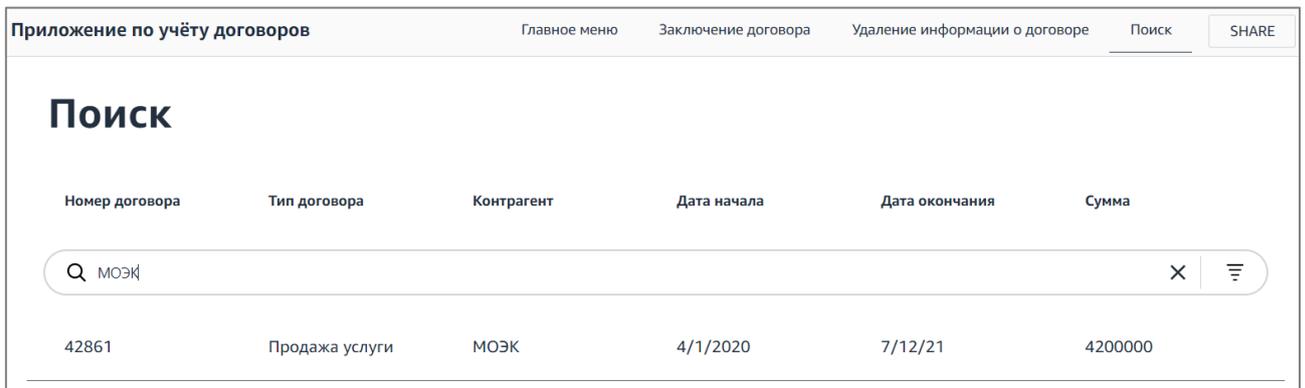


Рисунок 4.9 – Экран приложения «Поиск»

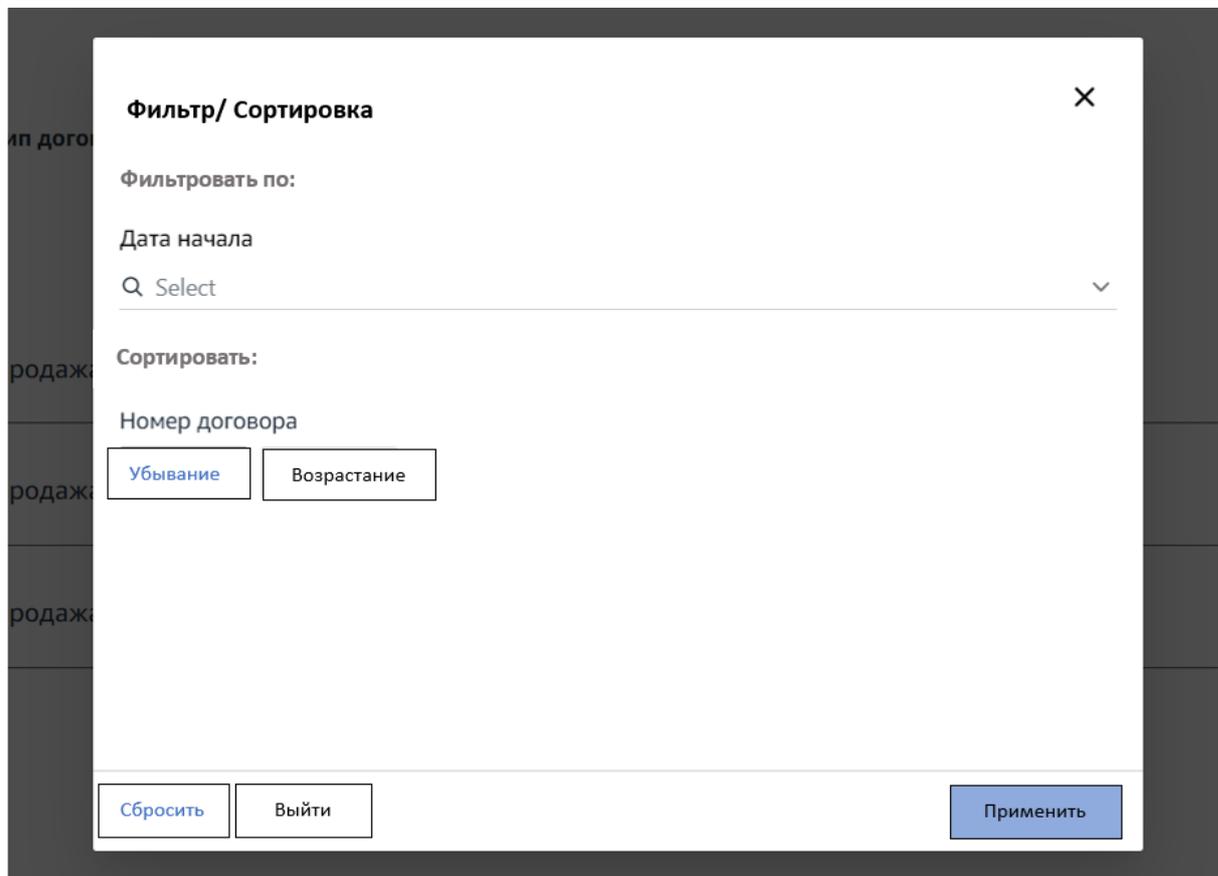


Рисунок 4.10 – Экран приложения «Удаление информации о договоре»

По умолчанию договоры можно отфильтровать по дате от самого раннего, как показано на рисунке, к самому позднему. Для этого нужно

выбрать одну из кнопок отвечающих за упорядочивание результатов поиска по убыванию или по возрастанию и нажать на кнопку «Применить», после чего все договора отсортируются в нужном пользователю порядке. Для сброса или выхода нужно нажать на соответствующие кнопки в нижнем левом углу всплывающего экрана. Определение уровня доступа к приложению осуществляется в момент авторизации. Экран авторизации представлен на рисунке 4.11 и имеет два поля: адрес электронной почты и пароль.

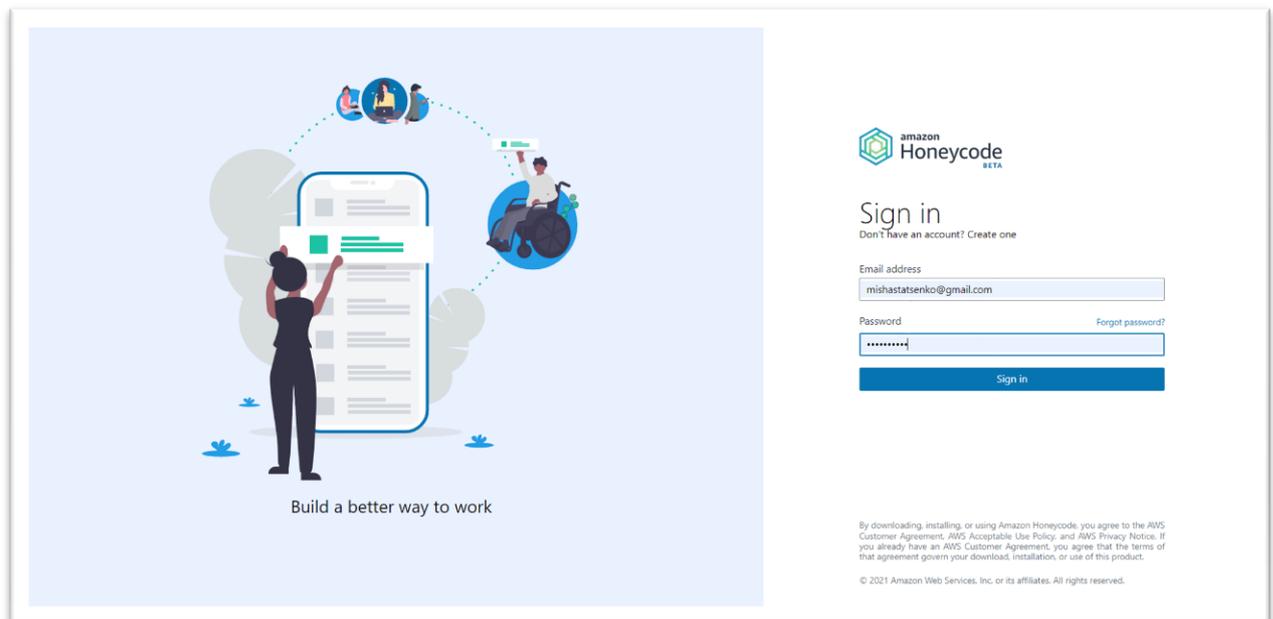


Рисунок 4.11 – Авторизация в приложении

Для настройки функций разграничения полномочий по уровню доступа были заведены учётные записи пользователей и проведены настройки на сайте платформы, представленные на рисунке 4.12.

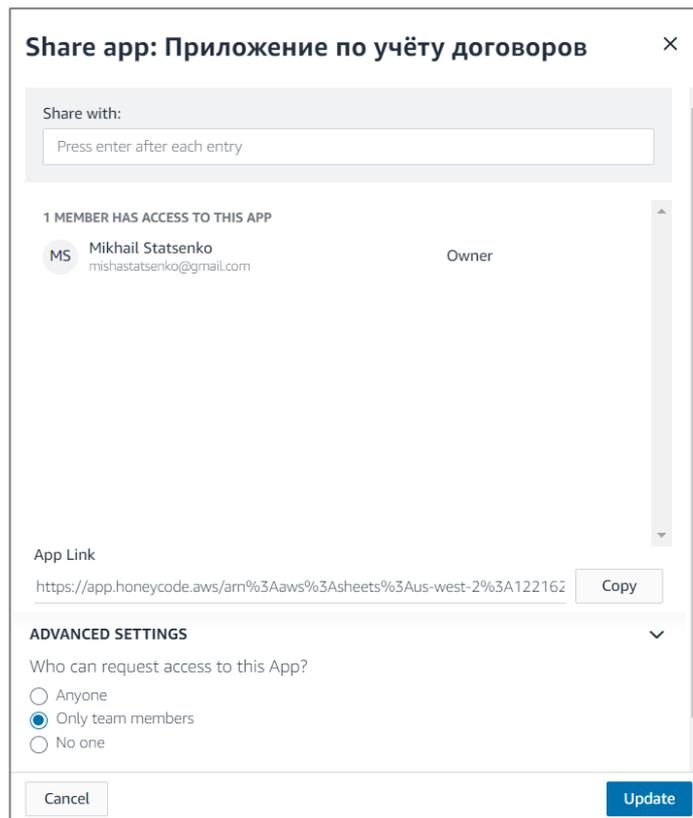


Рисунок 4.12 – Настройка доступа к приложению

4.4. Нагрузочное тестирование

Тестирование в основном принято разделять на функциональное и нефункциональное. К первому относят тестирование на соответствие заложенных функций, а ко второму все те виды тестирования, отличные от функционального. К последнему виду испытаний относят: нагрузочные и стрессовые тестирования, тестирование на возможность проникновения, тестирование удобства использования и т.д. В рамках текущей работы рассмотрено нагрузочное тестирование.

Согласно ГОСТ Р 56920-2016/ISO/IEC/IEEE 29119-1:2013 под нагрузочным тестированием понимается тип тестирования уровня производительности, проводимого для оценки поведения элемента тестирования при ожидаемых условиях переменной загрузки, обычно для

ожидаемых условий низкого, типичного и пикового использования [10]. Стоит отметить, что для нормальной работы приложения среднее время отклика составляет 0,5 сек.

Главным показателем производительности будет являться время отклика в трёх вариантах заполненности строк экрана «Заключение договора». Таким образом в рамках теста необходимо определить три показателя для разных видов нагрузки. В качестве таковых предлагается использовать разное количество заполненных строк с записями в базе данных приложения: 10, 50 и 100.

Для расчёта показателя необходимо определить хронометраж нескольких измерений с помощью электронного секундомера для разного количества записей в базе данных. Было произведено пять измерений для определённого ранее количества записей. Определив среднее время и отклонения времени для различного количества пациентов, определим время отклика. Для получения результатов необходимо рассчитать среднее арифметическое для всех измерений. Данный показатель рассчитывается по формуле:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}, \quad (4.1)$$

где \bar{x} – среднее арифметическое значение результатов n измерений при тестировании [24].

Для расчёта среднего квадратичного значения, была применена формула:

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}. \quad (4.2)$$

Результаты проведенных расчетов отражены в таблице 4.1.

Таблица 4.1. – Результаты нагрузочного тестирования

Количество заполненных строк полей договоров	Действие	t1 с.	t2 с.	t3 с.	t4 с.	t5 с.	Время отклика
10	Запись	0.22	0.25	0.21	0.23	0.21	0,2 сек.
	Поиск	0,2	0,19	0,2	0,21	0,2	0,2 сек.
50	Запись	2,51	2,24	2,57	2,55	2,5	0,25 сек
	Поиск	2,5	2,5	2,52	2,6	2,4	0,25 сек
100	Запись	0,4	0,35	0,3	0,38	0,38	0,38 сек.
	Поиск	0,36	0,37	0,38	0,35	0,35	0,35 сек.

Таким образом, с условием погрешности измерения, данные показатели удовлетворяют требование о нормальной работе с приложением.

Заключение

В работе были рассмотрены популярные методики разработки программных средств и выявлено, что метод Scrum подходит для задачи реализации программного продукта на non-код платформе. Принципы Agile Scrum позволяют облегчить работу команды разработчиков. Минимизация процессов, направленных на сокращение рутинных операций (и как правило, наименее приятных для разработчиков), таких как: написание документации и технического задания, сведены к минимуму. Акцент направлен на продукт, скорость его выхода в релиз и изменения в процессе периодических представлений рабочих версий заказчику.

В ситуации постоянных колебаний в бизнесе и в мире, практически невозможно предсказать (особенно продолжительного проекта), насколько актуальным получится продукт. Тем не менее использование гибких методологий разработки позволяют на любом этапе работ внести новые корректировки и переформулировать требования. Также часто случаются ситуации, когда на основе презентованной части продукта заказчик решает, что необходимо что-то поменять, улучшить или вообще убрать.

Тесное сотрудничество между заказчиком и ответственным со стороны исполнителя позволяет погрузиться в тонкости работы компании-заказчика, тем самым улучшить понимание требований. Частые очные встречи и обсуждения проделанной работы позволяют команде делиться результатом и улучшать результаты работы. Тесное общение даёт возможность задавать интересующие вопросы, тем самым позволяет обучаться в процессе.

Список литературных источников

1. Scrum.org. Scrum Glossary. – URL: <https://www.scrum.org/resources/scrum-glossary> (дата обращения 10.04.2021).
2. Свистунова Ю.В. Современная структура корпоративной информационной системы // Компьютерные технологии в моделировании, управлении и экономике. – 2021. – № 13. – С. 67-70.
3. Larman C., Basili V.R. Iterative and incremental development: a brief history // Computer. – 2003. – Vol. 36, №6. – P. 47-56.
4. Agile Alliance. Основопологающие принципы Agile-манифеста. – URL: <http://agilemanifesto.org/iso/ru/principles.html> (дата обращения 10.12.2021).
5. Гражданский кодекс Российской Федерации (часть первая) от 30.11.1994 г. № 51-ФЗ (с изм. от 23 июля 2013 г.).
6. ГОСТ Р 7.0.8-2013. Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу. Делопроизводство и архивное дело. Термины и определения (утв. Приказом Росстандарта от 17.10.2013 №1185-ст).
7. Кураков Л.П., Кураков В.Л., Кураков А.Л. Экономика и право: словарь-справочник. – М.: Вуз и школа, 2004. – 1070 с.
8. Магомадов В.С. Платформы low-code и no-code как способ сделать программирование более доступным для широкой общественности // Международный научно-исследовательский журнал. – 2021. – № 6-1(108). – С. 100-103.
9. Степанов Д.Ю. Бизнес и технологическая неопределенности в проектах имплементации корпоративных информационных систем на основе каскадной и многопроходных моделей внедрения // Высокопроизводительные вычислительные системы и технологии. – 2021. – Т.5, №1. – С. 304-312.

10. Степанов Д.Ю., Вельсовский А.В. Применение Agile Scrum в проектах SAP // Корпоративные информационные системы. – 2018. – №1. – С. 9-17.
11. Общероссийский классификатор видов экономической деятельности (утв. Приказом Росстандарта от 31.01.2014 №14-ст) (ред. от 23.12.2021).
12. ГОСТ Р 56920-2016/ISO/IEC/IEEE 29119-1:2013. Национальный стандарт Российской Федерации. Системная и программная инженерия. Тестирование программного обеспечения. Ч.1. Понятия и определения.
13. Куликов С.С. Тестирование программного обеспечения. Базовый курс. – Минск: Четыре четверти, 2017. – 312 с.
14. ГОСТ 19.102-77. Межгосударственный стандарт. Единая система программной документации. Стадии разработки.
15. Хабр. Amazon для тех, кто не умеет программировать. – URL: <https://clck.ru/ZZMsz> (дата обращения: 15.03.2022).
16. Amazon Honeycode. Help and Community. – URL: <https://clck.ru/ZZMyr> (дата обращения: 25.01.2022).
17. Sutherland J. Scrum. The Art of doing twice the work in half the time, 2014.
18. Wikipedia. Kanban. – URL: <https://clck.ru/9VDKf> (дата обращения: 24.03.2022).
19. Wikipedia. Scrum. – URL: <https://ru.wikipedia.org/wiki/SCRUM> (дата обращения: 21.02.2022).
20. Хабр. Scrum, революционный метод управления проектами. – URL: <https://clck.ru/U93rS> (дата обращения: 20.11.2021).
21. Кон М. Agile: оценка и планирование проектов. – М.: Альпина Паблишер, 2019.

22. Кон М. Пользовательские истории: гибкая разработка программного обеспечения. – М.: Вильямс, 2019.

23. User Guide. Пользовательское руководство Amazon Honeycode. – URL: <https://clck.ru/ZZNJv> (дата обращения: 01.05.2022).

24. Habr. Проектирование программного обеспечения. – URL: <https://clck.ru/ZZNMu> (дата обращения: 12.12.2021).

Приложение А

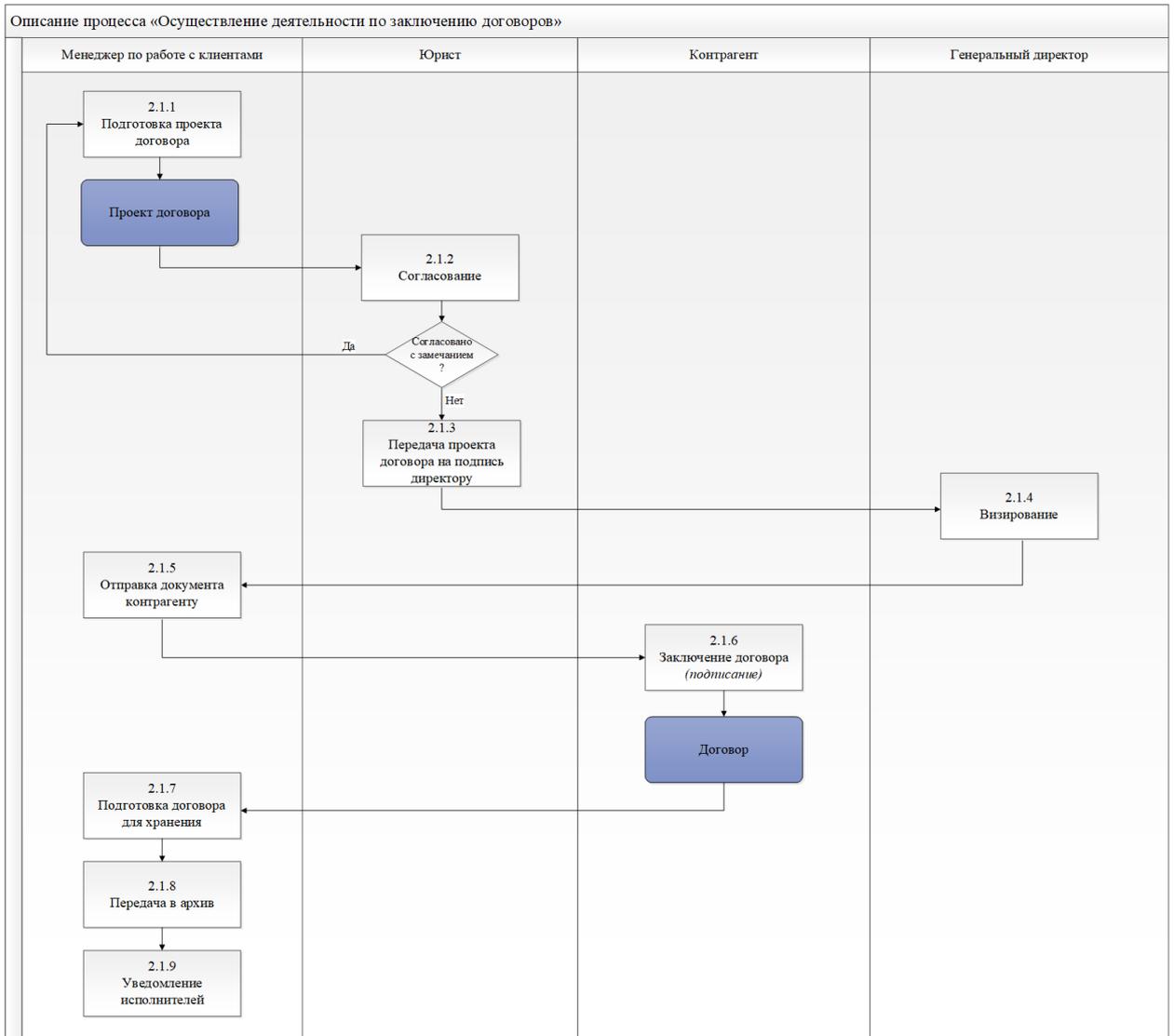


Рис. А1. Декомпозиция процесса «Осуществление деятельности по заключению договоров» в модели AS-IS

Приложение Б

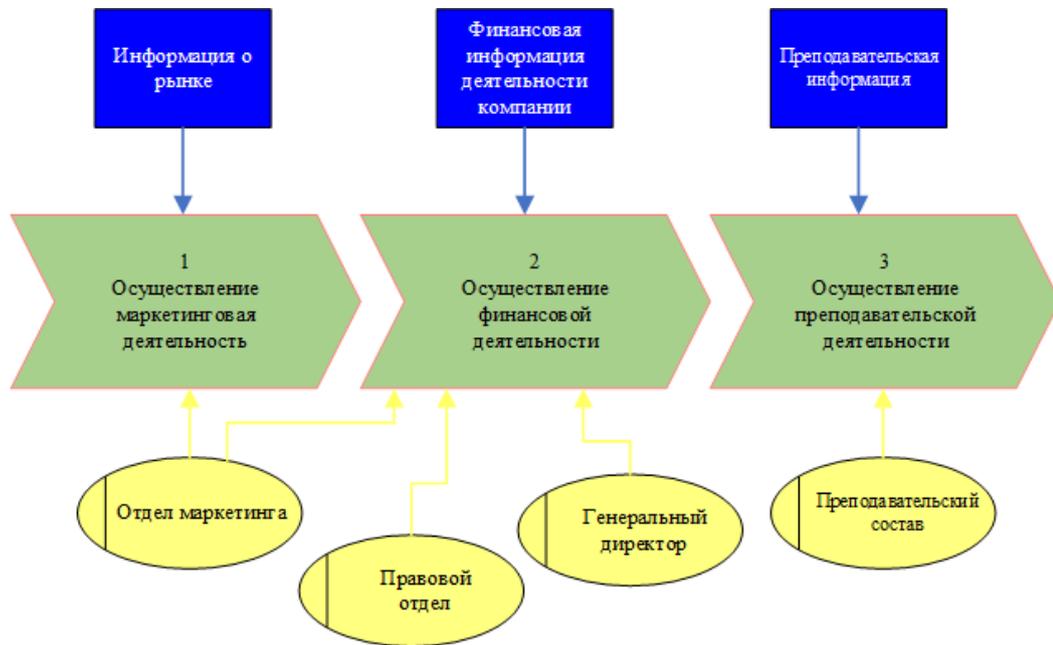


Рис. Б1. Модель первого уровня с ключевыми процессами ТО-ВЕ

Приложение В

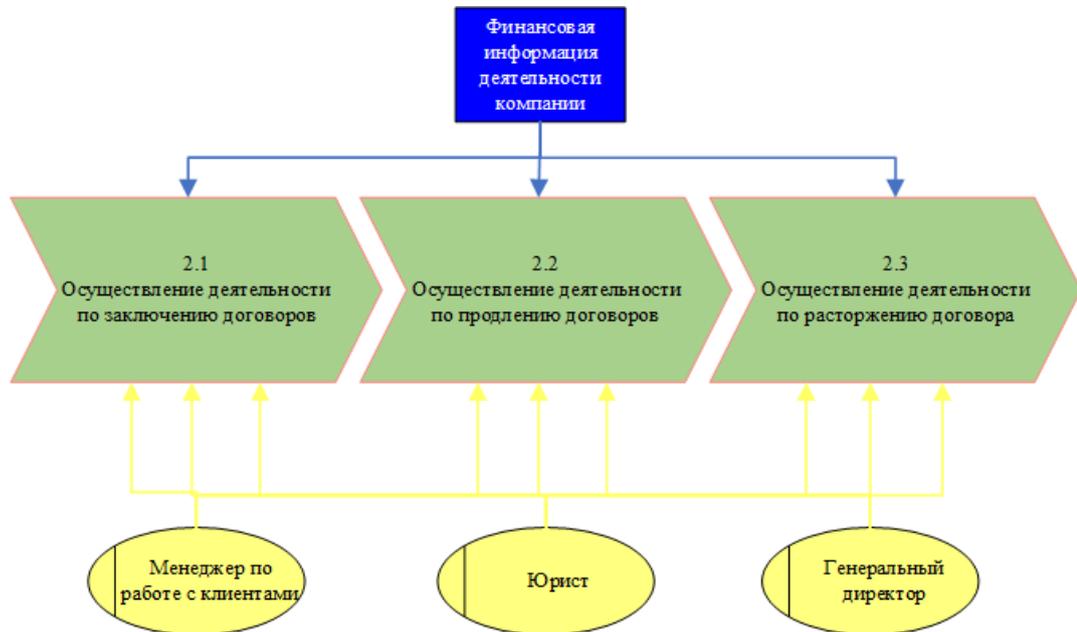


Рис. В1. Модель второго уровня, декомпозиция процесса «Осуществление финансовой деятельности»

Приложение Г

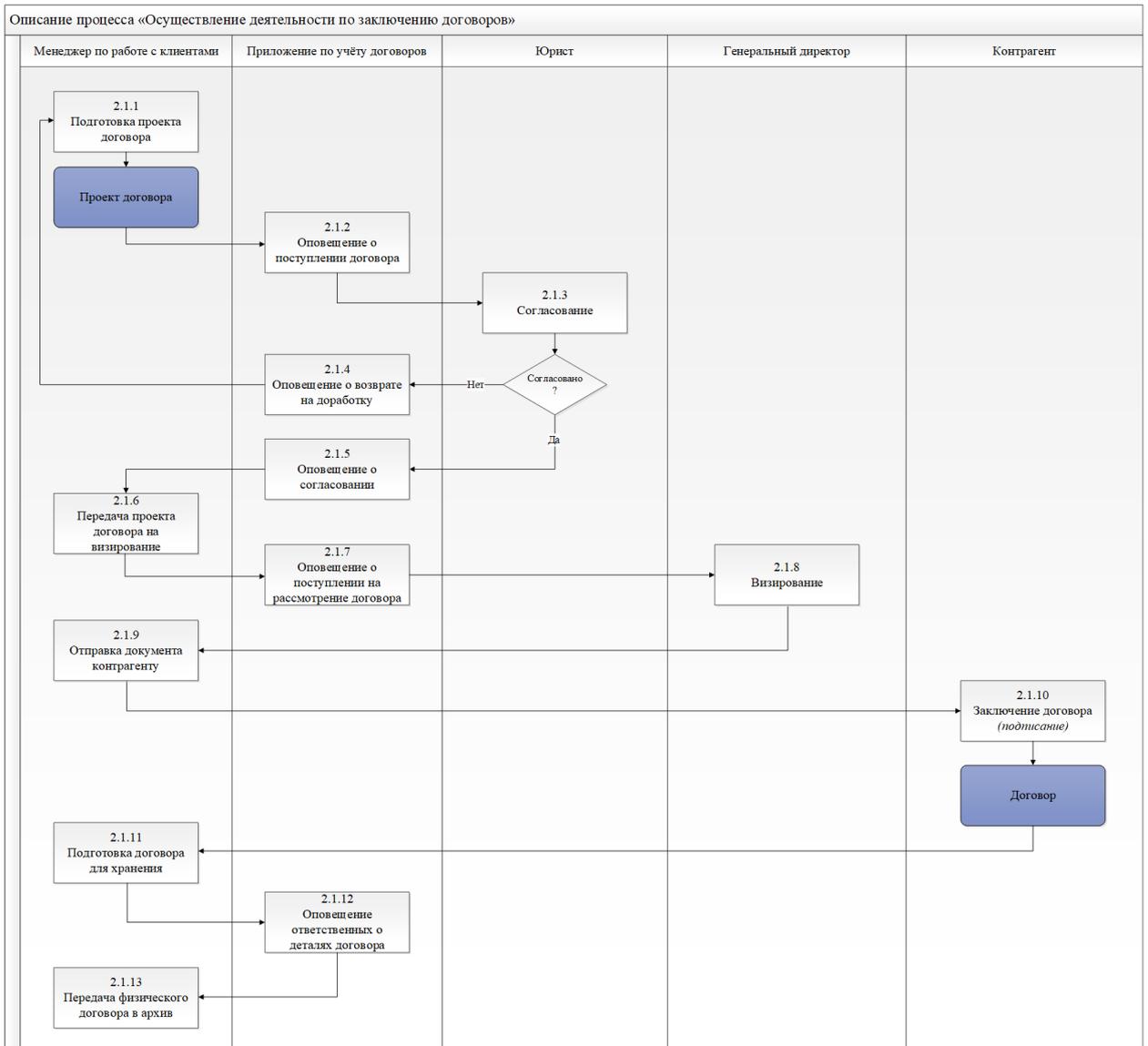


Рис. Г1. Декомпозиция процесса «Осуществление деятельности по заключению договоров» в модели ТО-ВЕ

Приложение Д

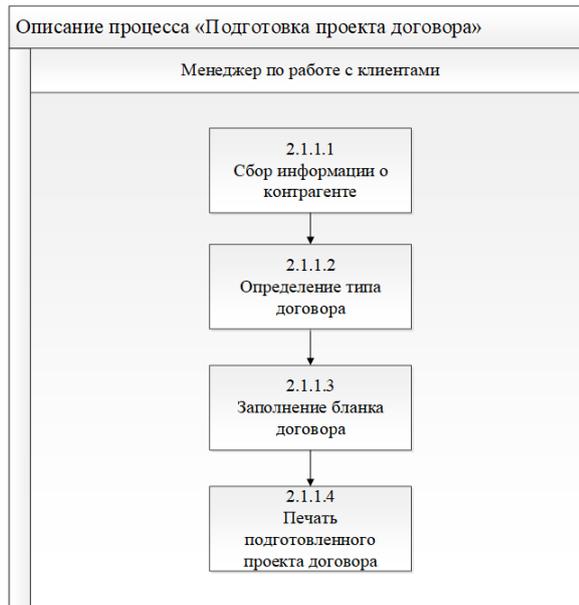


Рис. Д1. Декомпозиция процесса «Подготовка проекта договора»

Приложение Е

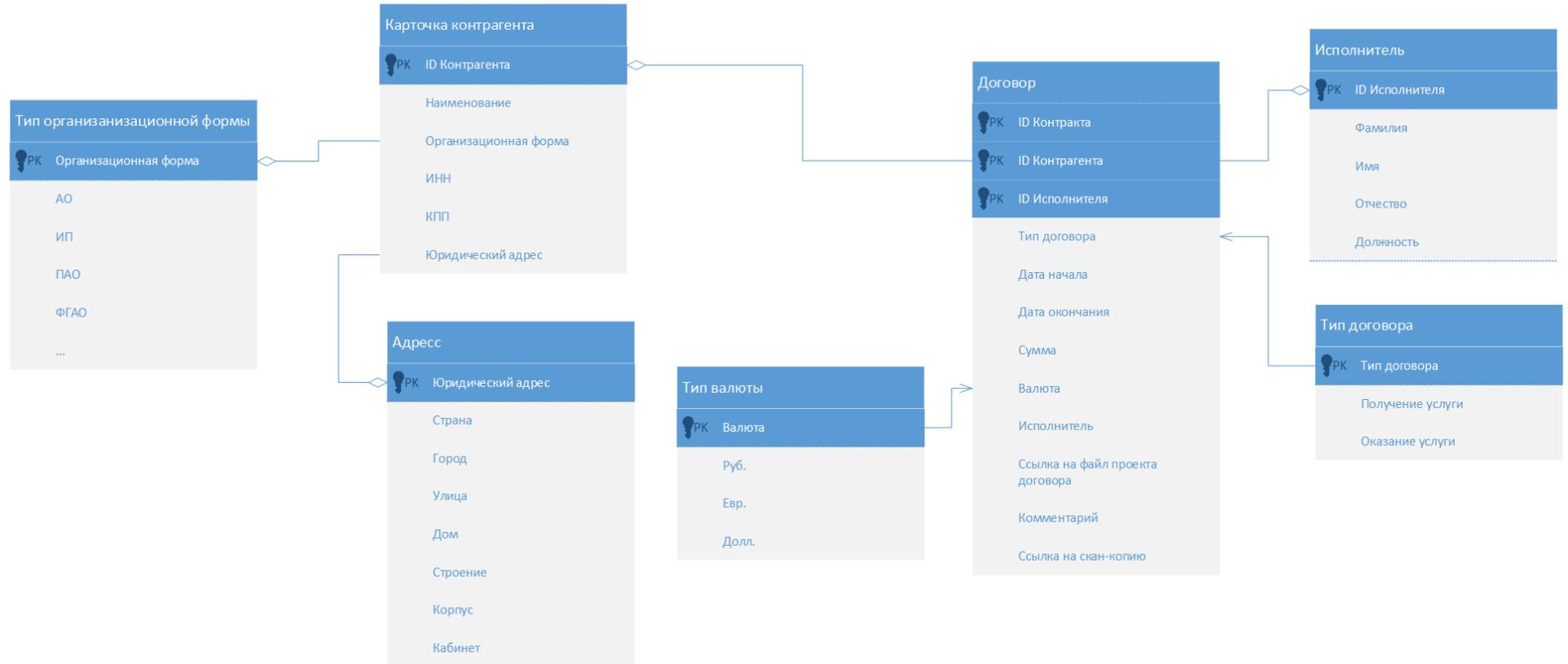


Рис. Е1. Архитектура данных