

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
GRADUATION THESIS**

**Исследование и прототипирование программно-визуального решения для
автоматизации бизнес процессов**

Обучающийся / Student Гусманова Римма Витальевна

Факультет/институт/клластер/ Faculty/Institute/Cluster факультет систем управления и
робототехники

Группа/Group R42445c

Направление подготовки/ Subject area 27.04.03 Системный анализ и управление

Образовательная программа / Educational program Цифровые системы управления 2020

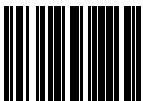
Язык реализации ОП / Language of the educational program Русский, Английский

Статус ОП / Status of educational program МОП

Квалификация/ Degree level Магистр

Руководитель ВКР/ Thesis supervisor Пенской Александр Владимирович, кандидат
технических наук, Университет ИТМО, факультет программной инженерии и
компьютерной техники, доцент (квалификационная категория "ординарный доцент")

Обучающийся/Student

Документ подписан	
Гусманова Римма Витальевна	

(эл. подпись/ signature)

Гусманова
Римма
Витальевна

(Фамилия И.О./ name
and surname)

Руководитель ВКР/
Thesis supervisor

Документ подписан	
Пенской Александр Владимирович	

(эл. подпись/ signature)

Пенской
Александр
Владимирович

(Фамилия И.О./ name
and surname)

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ /
OBJECTIVES FOR A GRADUATION THESIS**

Обучающийся / Student Гусманова Римма Витальевна

Факультет/институт/клластер/ Faculty/Institute/Cluster факультет систем управления и
робототехники

Группа/Group R42445c

Направление подготовки/ Subject area 27.04.03 Системный анализ и управление

Образовательная программа / Educational program Цифровые системы управления 2020

Язык реализации ОП / Language of the educational program Русский, Английский

Статус ОП / Status of educational program МОП

Квалификация/ Degree level Магистр

Тема ВКР/ Thesis topic Исследование и прототипирование программно-визуального
решения для автоматизации бизнес процессов

Руководитель ВКР/ Thesis supervisor Пенской Александр Владимирович, кандидат
технических наук, Университет ИТМО, факультет программной инженерии и
компьютерной техники, доцент (квалификационная категория "ординарный доцент")

Основные вопросы, подлежащие разработке / Key issues to be analyzed

Цель исследования: развитие средств формализации и представления бизнес-процессов,
предназначенных для задач автоматизации с низким объёмом программирования ("по-
code", "low-code", без программирования).

Задачи:

- 1) Исследовать рынок средств автоматизации бизнес-процессов с низким объемом
программирования.
- 2) Классифицировать существующие решения для автоматизации бизнес-процессов по
области применения и по используемым методам.
- 3) Определить/разработать пример типового бизнес-процесса предприятия, на примере
которого возможно продемонстрировать применимость различных типов средств
автоматизации. Создать испытательный стенд для выбранного фрагмента типового бизнес-
процесса.
- 4) Выполнить проектирование и прототипирование элементов системы автоматизации
бизнес-процессов с низким объемом программирования.
- 5) Провести аprobацию разработанных элементов прототипа на базе испытательного
стенда. Сравнить полученные результаты с альтернативными способами автоматизации.

Содержание работы (перечень подлежащих разработке вопросов):

- 1) Автоматизация бизнес-процессов с низким объёмом программирования.

- 2) Типовые задачи автоматизации бизнес-процессов.
- 3) Проектирование средств автоматизации бизнес-процессов.
- 4) Сравнительный анализ методов и средств автоматизации бизнес-процессов.
- 5) Прогнозирование трендов в развитии средств автоматизации бизнес-процессов с низким объёмом программирования

Рекомендуемые материалы и пособия для выполнения работы:

- 1) Oleg Lapshin Who needs low code platforms and why?, MICSECS 2021
- 2) Александр Гриценко, Погружение в RPA-технологии, MICSECS 2021
- 3) Bryan Kenneweg, Imran Kasam, Micah McMullen, Building Low-Code Applications with Mendix: Discover best practices and expert techniques to simplify enterprise web development, 2021
- 4) Enrico Murru, Hands-On Low-Code Application Development with Salesforce, 2020

Форма представления материалов ВКР / Format(s) of thesis materials:
Отчёт

Дата выдачи задания / Assignment issued on: 07.02.2022

Срок представления готовой ВКР / Deadline for final edition of the thesis 20.05.2022

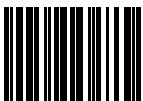
Характеристика темы ВКР / Description of thesis subject (topic)

Тема в области фундаментальных исследований / Subject of fundamental research: нет /
not

Тема в области прикладных исследований / Subject of applied research: да / yes

СОГЛАСОВАНО / AGREED:

Руководитель ВКР/
Thesis supervisor

Документ подписан	
Пенской Александр Владимирович	
28.02.2022	
(эл. подпись)	

Пенской
Александр
Владимирович

Задание принял к
исполнению/ Objectives
assumed BY

Документ подписан	
Гусманова Римма Витальевна	
28.02.2022	
(эл. подпись)	

Гусманова
Римма
Витальевна

Руководитель ОП/ Head
of educational program

Документ подписан	
Пыркин Антон Александрович	
10.05.2022	
(эл. подпись)	

Пыркин Антон
Александрович

(эл. подпись)

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University**

**АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
SUMMARY OF A GRADUATION THESIS**

Обучающийся / Student Гусманова Римма Витальевна

Факультет/институт/кластер/ Faculty/Institute/Cluster факультет систем управления и робототехники

Группа/Group R42445c

Направление подготовки/ Subject area 27.04.03 Системный анализ и управление

Образовательная программа / Educational program Цифровые системы управления 2020

Язык реализации ОП / Language of the educational program Русский, Английский

Статус ОП / Status of educational program МОП

Квалификация/ Degree level Магистр

Тема ВКР/ Thesis topic Исследование и прототипирование программно-визуального решения для автоматизации бизнес процессов

Руководитель ВКР/ Thesis supervisor Пенской Александр Владимирович, кандидат технических наук, Университет ИТМО, факультет программной инженерии и компьютерной техники, доцент (квалификационная категория "ординарный доцент")

**ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
DESCRIPTION OF THE GRADUATION THESIS**

Цель исследования / Research goal

Апробация и разработка прототипа системы автоматизации бизнес-процессов отдела по работе с персоналом с использованием подхода «без программирования».

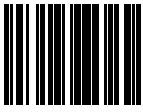
Задачи, решаемые в ВКР / Research tasks

1. Провести анализ и исследование существующих средств моделирования и автоматизации бизнес-процессов. 2. Исследовать рынок технологий платформ разработки «без программирования». 3. Произвести сравнительный анализ традиционных методов разработки и разработки «без программирования», включая прототипирование системы автоматизации бизнес-процессов. 4. Провести анализ результатов исследования и прототипирования, оценить возможный вектор развития технологии.

Краткая характеристика полученных результатов / Short summary of results/findings

В данной работе был произведен анализ бизнес-процессов отдела по работе с персоналом, рассмотрены существующие средства моделирования и произведен анализ решений на основе платформ «без программирования». Данная работа содержит разработку прототипа системы автоматизации бизнес-процессов с и без использования подхода «без программирования». Данная работа позволяет на основе полученных практических и теоретических результатов продемонстрировать достоинства и недостатки, связанные с внедрением подхода «без программирования» в сравнении с традиционной разработкой.

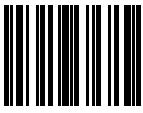
Обучающийся/Student

Документ подписан	
Гусманова Римма Витальевна	
25.05.2022	
(эл. подпись/ signature)	

Гусманова
Римма
Витальевна

(Фамилия И.О./ name
and surname)

Руководитель ВКР/
Thesis supervisor

Документ подписан	
Пенской Александр Владимирович	
25.05.2022	
(эл. подпись/ signature)	

Пенской
Александр
Владимирович

(Фамилия И.О./ name
and surname)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
Глава 1. Введение в понятие разработки «без программирования»	10
1.1. Основные принципы работы платформ «без программирования»	10
1.2. Выводы	11
Глава 2. Автоматизация бизнес-процессов на платформах «без программирования»	12
2.1. Анализ и исследование существующих средств моделирования бизнес-процессов	12
2.1.1. Понятие и сущность бизнес-процесса	12
2.1.2. Построение бизнес-процессов	15
2.1.3. Виды бизнес-процессов	16
2.2. Рассмотрение существующих средств моделирования	17
2.2.1. Нотация моделирования BPMN	18
2.2.2. ISO 15926 – стандарт представления данных	23
2.2.3. DoDaF Ideas	27
2.2.4. BORO метод	30
2.3. Существующие решения с низким объемом программирования	35
2.3.1. Comindware	36
2.3.2. Creatio	38
2.3.3. ELMA365	41
2.3.4. Типовые задачи автоматизации бизнес-процессов	45
2.3.5. Классификация существующих решений систем “без программирования”	46
2.4. Выводы	46

Глава 3. Проектирование средств автоматизации бизнес-процессов	48
3.1. Построение публичных бизнес процессов	53
3.2. Построение бизнес процессов отдела по работе с персоналом	70
3.3. Прогнозирование трендов в развитии средств автоматизации бизнес-процессов «без программирования»	80
3.4. Выводы	83
Глава 4. Отличие традиционного подхода к разработке приложений и разработке на платформе «без программирования»	84
4.1. Пример создания проекта при традиционном подходе разработки	84
4.2. Сравнительный анализ создания приложений на платформе «без программирования» и приложений созданных командой разработчиков	86
4.3. Выводы	88
ЗАКЛЮЧЕНИЕ	89
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	91

ВВЕДЕНИЕ

Актуальность темы исследования. В настоящее время невозможно представить современные предприятия и производства без информационных систем, которые значительно помогают облегчить жизнь во многих сферах работы. В наше время у любой компании устоялись две ветки развития ИТ-инфраструктуры компаний: заказать разработку необходимого программного обеспечения или же сделать самостоятельно. При достаточности ресурсов можно реализовать любую информационную систему, которая будет реализована под нужды предприятия или производства. Создать систему, которая смогла бы решить множество задач и, при необходимости, автоматизировать ту или иную деятельность. Данный продукт позволит освободить сотрудников от рутинной работы и заняться более важными задачами, с которыми пока не может справиться автоматика. Однако, для того что бы создать какую-либо систему для предприятия, при традиционном способе разработки, необходимо провести множество этапов предпроектного обследования, а также смоделировать систему и согласовать вопросы с бизнес-заказчиком. Также не стоит забывать, что уже разработанный программный продуктом необходимо быстро и гибко изменять, реагируя на изменения внутри процесса предприятия или же изменения законодательства. На все это уходит много времени и ресурсов и, соответственно, приходиться писать довольно много кода.

Сегодня появляется третья альтернатива, которая позволяет разрабатывать программные продукты быстро — это так называемые подходы к разработке с no-code решением или же, в дальнейшем, “без программирования”. Разработка, основанная на технологии “без программирования” — это логическая эволюция технологии разработки программного обеспечения, которая может привести в будущем к технологии разработки программного обеспечения без кода. Некоторые зачатки данных

технологий уже успешно используются, так как разработчики всегда стараются создавать и применять более высокий уровень абстракции для ускорения разработки программного обеспечения. Уже существуют наработки, на основе которых можно понять принцип работы с “без программирования”. Например, всем известные конструкторы веб сайтов, Webflow и Wix используют данный подход “без программирования” для создания лендингов страниц.

Пандемия заставила быстро переводить процессы компаний в цифровой формат, а также вводить в работу новое программное обеспечение как можно быстрее. Где-то приходилось проверять теории, создавая минимально рабочий продукт (MVP), а где-то быстро создавать программное обеспечение (ПО). Все это несомненно ускорило процесс эволюции в программировании, что привело к потребности создания no-code платформ.

Подход к разработке “без программирования” помогает перевести множество различных бизнес-процессов в специализированное программное обеспечение за короткий срок и с минимальными потерями, а также привлечь к участию в разработке программного продукта не только сотрудников информационных подразделений, но и бизнес подразделений. Если за годы пандемии было создано менее 25% разработанного программного обеспечения на основе “без программирования”, то по прогнозам аналитической компании Gartner уже к 2025 году данный подход займет около 75% разрабатываемых продуктов. Соответственно платформы “без программирования” будут улучшаться с каждым релизом и соответственно станут более гибкими и настраиваемыми, что позволит в разы упростить разработку и сделать создание новых программных продуктов быстрее и более настраиваемыми.

Степень теоретической разработанности темы. За последнее время было создано множество различных платформ разработки “без

программирования”, но на текущий момент нет четкой грани понимая и различия между тем, чем отличаются платформы “без программирования” от существующих средств разработки программного обеспечения, а также понимания практического применения. Это происходит потому что текущие решения главным образом созданы программистами для программистов. Это решение не дает обычным обывателям строить свои системы без привлечения разработчиков программного обеспечения.

Целью исследования работы является разработка решения для создания прототипа высокого уровня автоматизации бизнес-процессов работы отдела по работе с персоналом. Прототип, который позволит производить автоматизацию и управление функционалом разрабатываемого решения. Данное решение позволяет объединить разработчиков с бизнес-заказчиками и реализовывать цифровые решения.

Цель достигается путем решения следующих задач:

1. провести анализ и исследование существующих средств моделирования и автоматизации бизнес-процессов;
2. исследовать рынок технологий платформ разработки “без программирования”;
3. произвести качественные оценки и возможности текущих решений;
4. предоставить результаты исследования и прототипирования, а также рассмотреть возможный вектор развития технологии.

Объектами исследования являются разработка программного обеспечения с использованием современных платформ разработки “без программирования”, с помощью которого имеется возможность оперативно вносить правки в существующее решение, так как во главе стоит бизнес-процесс, а уже потом его программная составляющая.

Предметом исследования являются платформы для разработки “без программирования” их изучение и выявление их преимуществ и недостатков, что позволяет выделить основную ветвь дальнейшего развития технологии разработки “без программирования”.

Практическая значимость исследования. Выявленные закономерности, существующие подходы и недостатки позволяют произвести эволюцию в разработке программного обеспечения, что позволит создавать более гибкие и удобные продукты. Создание таких платформ поможет расширить функциональность разрабатываемых продуктов, а также дать возможность создавать новый функционал людям из бизнес-подразделений, без привлечения специалистов из информационных подразделений. Однако, данное решение не является панацеей от отказа от традиционного программирования, а позволяет закладывать задуманную бизнес-заказчиком логику.

Глава 1. Введение в понятие разработки «без программирования»

Сегодня мир работает на приложениях. И корпоративный мир, конечно же, не исключение. К сожалению, традиционные процессы разработки приложений, на которые раньше полагались компании, больше не могут быть эффективными. Создание программных решений с нуля требует времени — месяцы или даже нескольких лет планирования, проектирования, тестирования и развертывания. Кроме того, когда компании нужны быстрые решения, компании необходимо быстро найти продуктовую команду, которая будет создавать продукт, что тоже занимает не мало времени. Вот почему так много компаний переходят на разработку с использованием платформ «без программирования».

1.1. Основные принципы работы платформ «без программирования»

Разработка на платформе «без программирования» — это подход к созданию приложений, который позволяет создавать программные продукты с минимальным ручным кодированием. Платформы разработки «без программирования» полагаются на графические интерфейсы и конфигурации, обрабатывая наиболее трудоемкие аспекты и позволяя работать быстрее. При встрече простого обывателя с терминами «разработка без программирования», no-code решения и других аналогичных терминов возникает несколько вопросов:

1. Что же такое разработка «без программирования»?
2. Какие различия разработки приложений традиционным способом от разработки «без программирования»?
3. Кто может создавать решения на платформах «без программирования»?
4. Как работают платформы разработки «без программирования»?

Ответы на эти вопросы довольно просты. Данный подход к разработке не требует традиционного опыта кодирования, что дает возможность любому человеку, независимо от того, является ли он

разработчиком или нет, создавать и обновлять широкий спектр приложений в короткие сроки. Платформы разработки «без программирования» позволяют тем, кто занимается разработкой ПО, создавать приложения с минимальным программированием. Потому что они используют визуальный подход к разработке через графический интерфейс (GUI) и предварительно скомпилированные шаблоны, которые позволяют легко перетаскивать компоненты для быстрого создания своих приложений. Благодаря платформам «без программирования» и с минимальным кодом практически любой сотрудник организации может участвовать в процессе разработки.

Большое количество встроенных инструментов работают совместно, чтобы обеспечить среду, в которой разработчики нового функционала могли оставить написание большого количества кода в прошлом, лишь дорабатывая основную логику, которая уже реализована на платформе. Благодаря визуальному программированию и автоматической генерации кода эти платформы ограничивают необходимость ручного кодирования. В то же время платформы «без программирования» автоматически подключаются к серверной части, управляя хранением и использованием данных без необходимости участия пользователя.

1.2. Выводы

В этой главе были рассмотрены основные принципы работы платформ «без программирования», какие есть отличия от разработки традиционным способом, как и кто может создавать решения на данных платформах. Исходя из того, что быстрая цифровизация различных процессов компании является неотъемлемой частью развития предприятия. Любая компания заинтересована в более быстром и дешевом способе перевода их в цифровую среду появились решения, которые позволяют создавать такие решения, благодаря встроенному инструментарию.

Глава 2. Автоматизация бизнес-процессов на платформах «без программирования»

Рынок решений платформ “без программирования” с каждым днем становится все больше и больше, занимая определенную нишу, к примеру, документооборот. Системы создаются для определенных отраслей и развиваются на основе требований рынка. Но прежде чем приступить к рассмотрению уже существующих решений необходимо разобраться в понятии “бизнес-процесс” и то как он строится и моделируется, так как именно на основе бизнес-процесса разрабатываются решения на платформах «без программирования». Для этого введем основные понятия и виды бизнес-процессов и рассмотрим различные средства моделирования бизнес-процессов.

2.1. Анализ и исследование существующих средств моделирования бизнес-процессов

Данный раздел затрагивает рассмотрение основных понятий и существующих средств моделирования (формализации) бизнес-процессов, применяемых при разработке систем автоматизации. В данном разделе будет произведен анализ построения бизнес процессов от уровня “ассемблера” в прототипировании – BPMN, до уровня высших онтологий. На основе анализа можно понять, как моделирование бизнес процессов привело к появлению платформ «без программирования», а также будут рассмотрены решения, которые уже присутствуют на рынке.

2.1.1. Понятие и сущность бизнес-процесса

Прежде чем приступить к разбору и сравнению различных методологий необходимо сначала внести более четкие определения, которые в дальнейшем будут использоваться в данной работе.

Понятие бизнес-процесс содержит два элемента: бизнес и процесс, исходя из этого можно сказать, что это последовательность логически

связанных действий, которые преобразуют вход в выход. Слово «бизнес» можно трактовать в соответствии с понятием бизнес-логика. Поскольку информационная система создается для поддержки определенной области деятельности, можно сказать, что бизнес-логика — это совокупность правил, принципов, зависимостей поведения объектов предметной области.

Таким образом, можно выделить первое определение «бизнес-процесс». **Бизнес-процесс** — совокупность различных видов деятельности, в рамках которой «на входе» используется один (или более) вид ресурсов, и в результате этой деятельности «на выходе» создается продукт, представляющий ценность для потребителя. Для моделирования бизнес-процесса характерны следующие черты:

1. Для проектирования модели бизнес-процесса необходимо: выделить данные которые мы получаем на вход и что мы имеем после выполнения нашего процесса (выходные параметры), регламенты и правила выполнения процесса (управление) и ресурсы – те единицы, которые будут работать с данным процессом, которые создадут модель того, как должна функционировать система. Где:
 - вход – это описание того, что у нас имеется для передачи в систему;
 - выход – это то, что необходимо получить после обработки системы;
 - управление – по какому регламенту/правилу это должно выполняться;
 - механизм – какие ресурсы должны выполнять данные действия, пример полная автоматизация без вмешательства сотрудника или частичная автоматизация.
2. Моделирование бизнес-процесса позволяет наглядно отобразить то, как будет работать система и дать возможность согласовать это с заказчиком, так как будут представлены, как минимум, две

спроектированные модели: модель «как есть» («AS IS») и модель «должно быть» («TO BE»). С помощью различных нотаций, понятных на интуитивном уровне для многих специалистов, можно отобразить данные модели, понять все ли правильно отображено на модели «как есть» и что еще можно модернизировать в модели «должно быть».

3. Моделирование дает основу понимания того, как работает система в целом, также основную модель всегда можно декомпозировать на более мелкие части, для более детального описания. Что также позволяет описать бизнес-процесс более подробно, а значит не упустить ни одной детали.
4. У любого бизнес-процесса должно быть ответственное лицо, которое будет руководить данным бизнес-процессом, чтобы достигался намеченный результат. Такое лицо именуется владельцем бизнес-процесса. Именно владелец бизнес-процесса будет следить за состоянием процесса, для того, чтобы процесс работал в надлежащем режиме, чтобы он функционировал и совершенствовался. Т. е. именно этот человек (или группа лиц) должны четко понимать границы процессов, их входы и выходы, а также саму технологию процесса, за который они несут ответственность.
5. У каждого бизнес-процесса есть и поставщик процесса. Поставщиком является субъект, который передает входную информацию, то есть предоставляет ресурсы (входные данные) для бизнес-процесса.
6. У каждого бизнес-процесса есть свой потребитель. Потребитель – это, если говорить непосредственно об информационной системе,

пользователь системы, который получает готовый результат, т. е. выходную информацию. Потребители бизнес-процесса бывают двух типов: внутренний и внешний. Внутренний потребитель – это непосредственно должностные лица внутри данной организации, в которой шла разработка (т. е. непосредственно решение внутренних задач организации) так называемы решения B2B. B2B – это «бизнес для бизнеса» — то есть задача направлена непосредственно «для предприятия», в отличии от решений B2C – «бизнес для потребителя», где непосредственно все решения направлены на клиента и его привлечения – это и есть внешний потребитель.

7. Требования бизнес-процесса. У каждого бизнес-процесса есть определенные требования, т. е. это может быть какая-то определённая документация, в которой зафиксированные требования, сформированные заинтересованными сторонами для реализации в бизнес-процессе.

2.1.2. Построение бизнес-процессов

Для того чтобы построить любой бизнес-процесс необходимы люди – т. е. сотрудники организации, имеющие определенные компетенции. Но работа сотрудников должна сопровождаться определенным пакетом документов, которые позволяют зафиксировать итоги встреч между теми или иными подразделениями организации (или же поставщиком и заказчиком), которые позволяют «фиксировать» эти договоренности на бумаге, в виде схем, карт и других аналогичных атрибуатах.

Фиксировать бизнес-процессы можно следующим образом:

1. Схема бизнес-процесса. Схема бизнес-процесса показывает путь прохождения (алгоритм) процесса по всем необходимым

«точкам». Может быть отрисован в одной из множеств имеющихся на данный момент методологий.

2. Карта бизнес-процесса или дорожная карта. Дорожная карта представляет собой таблицу, в которой строками являются сотрудники/команды, а столбцами некий промежуток времени (прим. месяцы), в табличных данных можно записать какая задача стоит перед ними, чтобы работала та или иная часть процесса.
3. Маршрут бизнес-процесса. Например, в бизнес-процессе может участвовать несколько сторонних информационных систем с которыми нам необходимо интегрироваться, для этого нам необходимо описать какие потоки данных есть в системе и как они «перетекают» между собой (прим. с помощью DFD диаграммы мы можем отразить эти потоки данных).
4. Матрица бизнес-процесса. Матрица представляет собой таблицу взаимодействия между процессами, где для каждого процесса показано что является входной информацией для данного процесса и для какого процесса выходные данные данного процесса могут быть входными.
5. Документирование бизнес-процесса (описание его регламента). Документирование БП является фиксацией процесса и его отображения «AS IS».

2.1.3. Виды бизнес-процессов

Существует несколько видов БП:

- Основные БП – процессы, ориентированные на конечный результат, т. е. непосредственно приносящие доход или пользу для организации.

- Вспомогательные БП – процессы, которые обеспечивают ресурсами основные БП, без которых невозможно выполнить основные БП.
- Процессы управления – управляют ходом процесса, т. е. отвечают за текущее и стратегическое управление основными процессами.

2.2. Рассмотрение существующих средств моделирования

В данном разделе проведем анализ построения бизнес процессов от уровня “ассемблера” в прототипировании — BPMN до уровня высших онтологий. На основе анализа можно понять, как моделирование бизнес процессов привело к появлению решений «без программирования», а также будут рассмотрены решения, которые уже присутствуют на рынке.

Анализ и исследование существующих средств работы заключается в том, чтобы изучить некоторые онтологии и нотации моделирования. Это является обязательным звеном для прототипирования программных продуктов на основе решений «без программирования», поскольку дает возможность создавать рабочий процесс с помощью нотаций моделирования. Прежде чем перейти к моделированию процессов информационной системы необходимо рассмотреть существующие методы моделирования, которые помогут избежать проблемы при проектировании информационных систем и предложить их наилучшее решение.

Перед тем, как приступить к изучению различных методов и онтологий изобретенных для решения задач создания и реорганизации существующих информационных систем в информационные модели, введем понятие реинжиниринга в разрезе информационных систем.

Реинжиниринг — это фундаментальное переосмысление и радикальное перепроектирование бизнес-процессов для достижения существенных

улучшений в ключевых, для современного бизнеса, показателях результативности.

2.2.1. Нотация моделирования BPMN

Сегодня существуют различные инструменты управления, бизнес-процессами, которые поддерживают концепцию BPM (Business Process Management — концепция процессного управления организацией, рассматривающая бизнес-процессы как особые ресурсы предприятия, непрерывно адаптируемые к постоянным изменениям). Среди нотаций моделирования бизнес-процессов в различных решениях используются языки BPMN, EPC (англ. Event-driven Process Chain), IDEF0 и другие.

Спецификация BPMN является нотацией графического моделирования, позволяющая отобразить бизнес-процессы в виде диаграммы понятные как техническим специалистам, так и бизнес-пользователям. Так как нотация BPMN подробно отображается и описана на основе интуитивно понятных элементов, данная нотация является своего рода “ассемблером” при формализации и проектировании системы. Данная методология позволяет выстроить, понятный всем участникам проекта, бизнес-процесс, а затем автоматизировать его в соответствии с схемой.

Для того что бы понять на сколько проста в понимании данная нотация и почему она имеет такое широкое распространение, при моделировании различных процессов предприятия, рассмотрим ее основные элементы:

1. Пул и Дорожки

Каждый бизнес-процесс состоит из пулов и дорожек, который дает понимание границ бизнес-процессов и их ролей, которые выполняют этот бизнес-процесс. Таким образом, пул — определяет границы бизнес процесса, а дорожка — это описание процесса, которое выполняет определенное лицо. На рисунке 1 можно увидеть графическое представление данного элемента нотации.

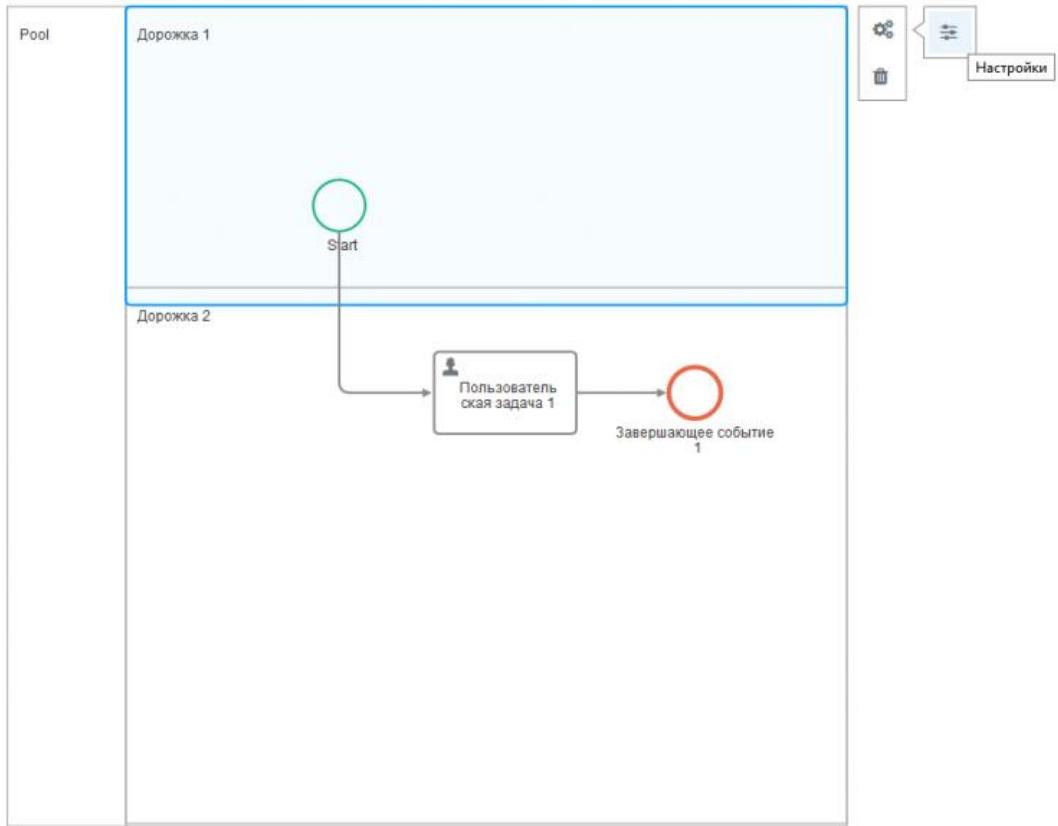


Рисунок 1 — отображение пула и дорожек в нотации BPMN

2. Действия

В нотации BPMN под действиями описывается шаг в работе над целым бизнес процессом. В нотации различаются различные виды действий, а основными являются:

- Задание — это единица работы, простейшее действие в процессе;
- Множественные экземпляры — данный блок символизирует многократное повторение действия;
- Циклическое действие — данное действие выполняется, пока условие цикла верно. Условие цикла может проверяться до или после выполнения действия.

На рисунке 2 представлено изображение данных действий в нотации.



Рисунок 2 — отображение основных видов действия в нотации BPMN

3. Логические операторы

Под логическими операторами подразумеваются шлюзы, которые разветвляют поток действий или же сливают его. Рассмотрим два основных вида разветвления/слияния потоков:

- Шлюз исключающего «или» — данный оператор позволяет создать альтернативные потоки, где каждый вариант имеет свой дальнейший путь;
- Параллельный шлюз — позволяет отобразить выполнение нескольких потоков, но при этом следующее действие не начнется, пока все потоки не будут выполнены.

Графическое отображение логических операций в нотации BPMN можно увидеть на рисунке 3.

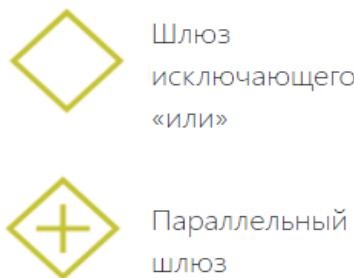


Рисунок 3 — графическое представление логических операторов в нотации BPMN

4. События

События являются основополагающими элементами, так как служат описанием того что должно случиться, так как с помощью этого элемента начинается и заканчивается бизнес-процесс. События бывают разного типа: начальное, промежуточное и завершающее. На рисунке 4 можно увидеть их графическое представление.

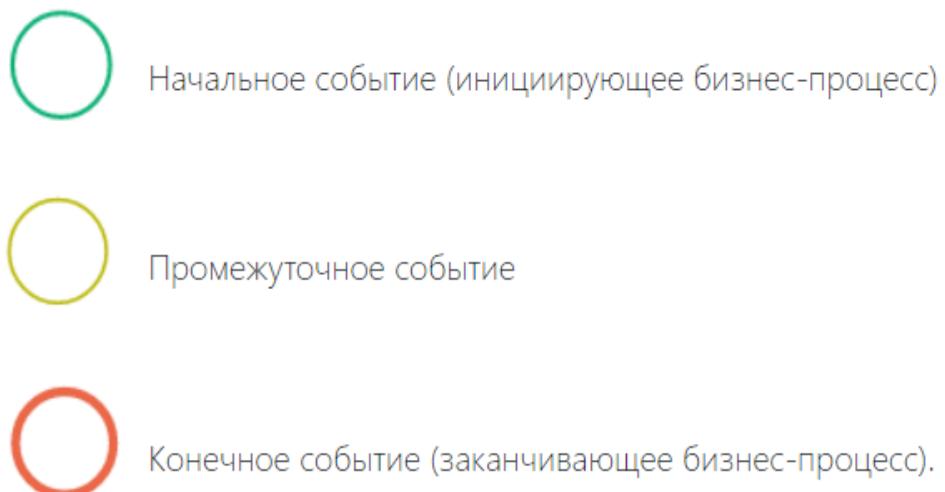


Рисунок 4 — графическое изображение представление событий в нотации BPMN

Промежуточные события могут отображаться в виде событий-сообщений, на рисунке 5 можно увидеть примеры событий сообщений.

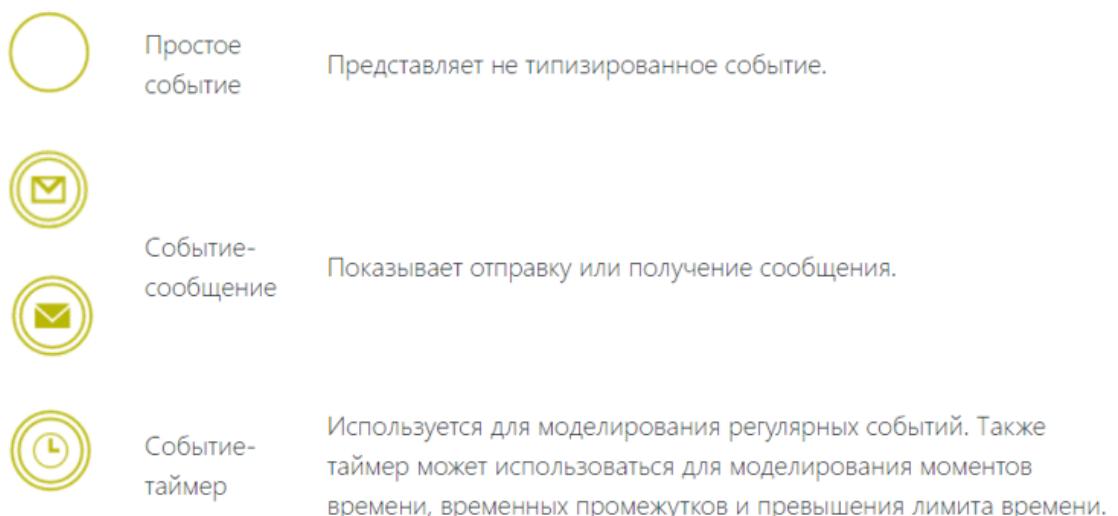


Рисунок 5 — графическое изображение событий-сообщений в нотации BPMN

5. Потоки

Поток в нотации BPMN отображается стрелкой, по этой стрелке мы можем отследить последовательность действий в бизнес-процессе. Потоки бывают разного типа:

- поток управления — поток, который указывает на дальнейшие шаги без разветвлений;
- условный поток управления — поток, который показывает, что последующее действие может выполняться только в том случае, если заданное условие будет выполнено;
- поток управления по умолчанию — показывает по какому потоку будет выполняться бизнес-процесс, если ни одно из условий не будет выполнено;
- поток сообщений — отображает общение между процессами

На рисунке 6 отображены графические изображения в нотации.

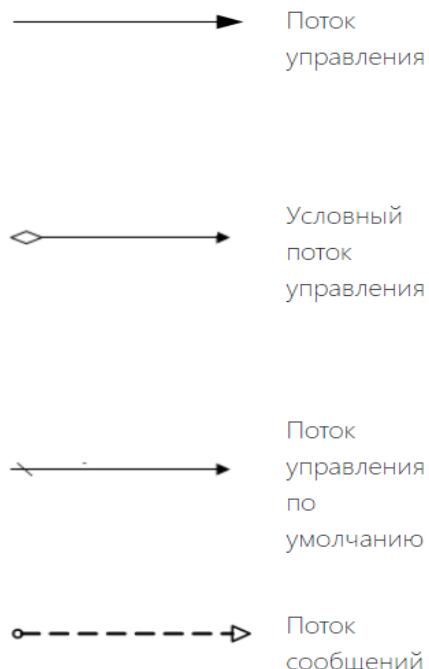


Рисунок 6 — графическое изображение потоков в нотации BPMN

6. Артефакты

Артефакты — это объекты которые не влияют на основной описываемый бизнес-процесс, а лишь являются сопровождением к описанию.

На рисунке 7 можно рассмотреть самые часто используемые артефакты в нотации BPMN.



Рисунок 7 — графическое изображение артефактов в нотации BPMN

На примере данных элементов можно убедиться, что данная нотация является легкой для восприятия и является, своего рода, усовершенствованием блок-схем, что позволяет оперировать ею не только техническими специалистами, что является большим преимуществом. Но несмотря на все свои преимущества, нотацию можно отнести к “простейшему” виду моделирования, что породило множество других различных способов моделирования в дальнейшем.

2.2.2. ISO 15926 – стандарт представления данных

Следующим примером в работе будет выделен стандарт описания ISO 15926 — единого стандарта представления данных, для обмена информации между независимых друг с другом программных продуктов.

Появление стандарта ISO 15926 вызвано тем, что была необходима систематизация данных внутри проекта или же интеграция данных между предприятиями, когда невозможно стандартизировать программное

обеспечение. Так как существует множество различных нотаций и форматов, которые вызывают или же дублирование информации или вызывают противоречия.

Стандарт ISO 15926 применяется для представления систематизированных данных при «слиянии» нескольких видов данных различных программных продуктов. ISO 15926 стандартизирует терминологию, организацию информации и то, как компьютерная система/модули связаны между собой и обмениваются информацией. Данный стандарт позволяет отобразить концептуальную модель данных, которая в свою очередь, дает представление о границах и возможностях жизненного цикла разрабатываемой системы. Рассмотрев концептуальную модель подробнее можно понять какие входящие параметры (исходные данные), понять какие требования предъявляются к тем или иным частям системы, как они (части/модули системы) описаны и как они обмениваются информацией (в виде шаблонов обмена данными). Таким образом можно сказать, что процесс строиться на основании внедрения хранилища данных, который будет интегрироваться с различными моделями и системами на всех этапах жизненного цикла системы.

Базовой моделью построения представления является модель, построенная в 4D пространстве, которая содержит в себе объект, его характеристики, состояние, состав (может меняться со временем на любой стадии жизненного цикла) и само время. Данная модель является основополагающей, на ее основе строятся классы и отношения между ними, т.е. так называемые таблицы в базе, если говорить о хранилище данных. Смоделированная модель данных по данному стандарту хранится в библиотеке справочных данных (Reference Data Library, далее RDL). Реализация стандарта подразумевает наличие как общедоступных (ссылаясь на справочные данные из них вправе каждый), так и закрытых (предназначенных исключительно для внутреннего использования) RDL. При

в этом несколько RDL могут применяться в рамках одного проекта одновременно благодаря единым процедурам доступа к ним.

Преимущество данного стандарта удобно рассмотреть на примере. Множество различных систем на предприятии, где в каждом приложении есть своя модель данных. При острой необходимости обмена информацией между различными приложениями возникает потребности в создании “информационного моста” в виде адаптеров, конверторов и тому подобного. Все это вызывает необходимость создавать интеграционные решения, на что уходит немало времени и денежных затрат. На рисунке 8 можно увидеть пример того, сколько интеграционных решений можно построить.

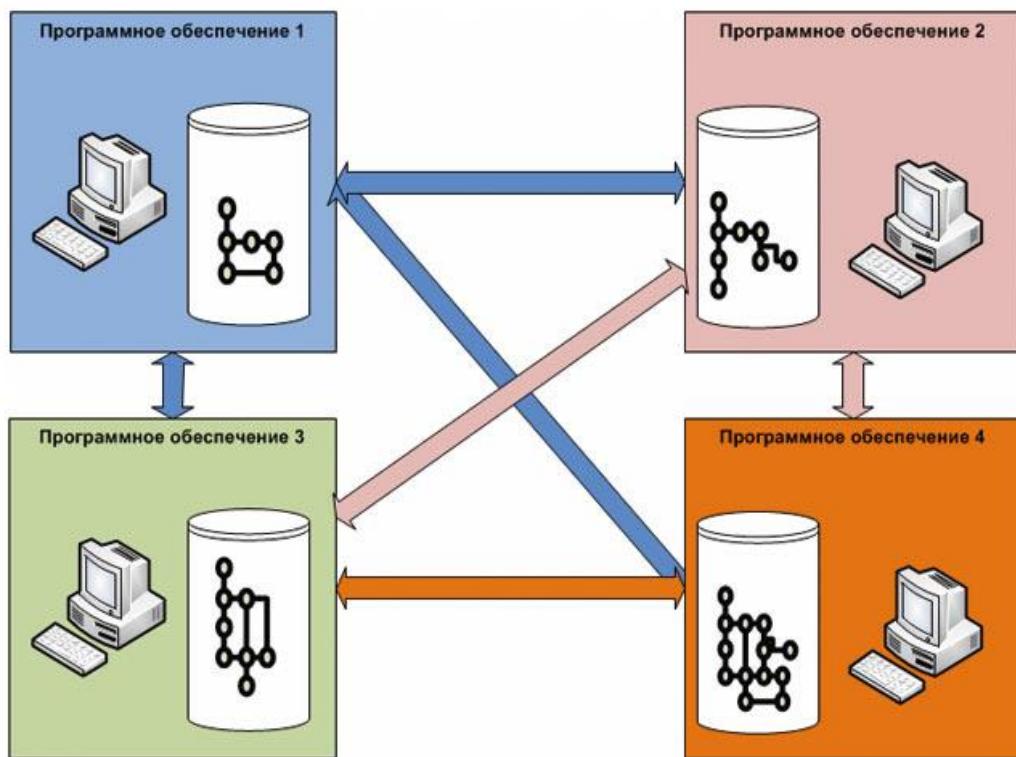


Рисунок 8 — пример построения интеграций между 4 системами

Стандарт ISO 15926 предлагает комплексное решение данной проблемы и построив интеграцию, в которой будет менее затратным и более быстрым решение. Это возможно благодаря унификации форматов обмена информацией между системами, то есть построения модели единого образца и стиля. Стандарт предлагает связать в единые целые компоненты — шаблоны, структура данных принятая в стандарте ISO 15926. Стандарт позволяет

произвести отображения данных и справочной информации в такой формат хранения и передачи данных хранящихся в RDL.

На рисунке 9 можно рассмотреть вариант реализации адаптера, построенного на стандарте ISO 15926. Данный пример показывает построение адаптера на основе модели приложения и модели стандарта ISO 15926. Основа заключается в том, что адаптер содержит инструмент для проведения соответствия между моделями данных приложения и ISO 15926 — редактор определения соответствия данных между различными семантиками систем. Этот редактор должен иметь доступ ко всем задействованным RDL и модели данных приложения. Для обеспечения программной совместимости необходимо для каждого из приложений создать соответствующие маппинг файлы, в таком случае обмен будет заключаться в передаче данных в формате ISO 15926 — едином для всех приложений информационной системы.

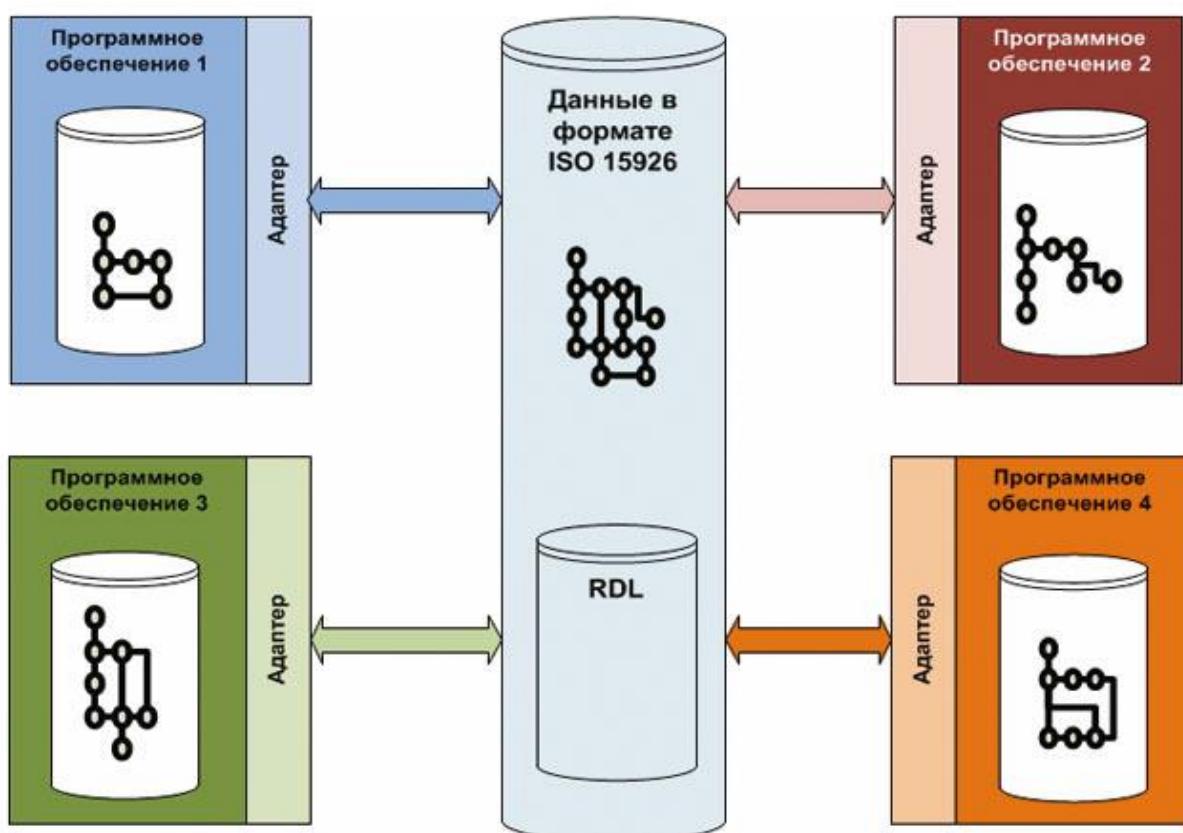


Рисунок 9 — реализации адаптера построенного на стандарте ISO 15926

Подводя итоги рассмотрения данного стандарта можно выделить как ряд преимуществ, так и ряд недостатков. Преимущества заключаются в том, что в стандарте присутствует единый стандарт обмена информацией внутри системы, возможность создания полной автоматизации между системами, что ведет к уменьшению затрат на разработку. Но есть и существенные недостатки, такие как: обучение большого числа работников и приведение всех имеющихся систем под единый стандарт трудоемко, необходимо большое количество разработчиков и тестировщиков, которые должны привести в необходимый вид все адаптеры и проверить их профпригодность в соответствии со стандартами.

2.2.3. DoDaF Ideas

DoDAF (Department of Defense Architecture Framework) — это архитектурный фреймворк Министерства обороны США. Данная методология позволяет улучшить течение и описание бизнес-процессов на всех уровнях, а также принимать более эффективные решения.

Возможность достижения наиболее понятной передачи информации достигается благодаря тому, что визуализация и отображение архитектурных данных производится с помощью моделей в виде документов, таблиц и графических изображений.

Department of Defense Architecture Framework использует метамодель данных (Data Meta-Model) или по-другому DM2. Она состоит из нескольких уровней, которые предоставляют информацию для различных групп пользователей. На рисунке 10 можно увидеть структуру верхних уровней онтологии.

Управление на всех уровнях разработки — лозунг онтологии DoDAF, их в онтологии пять штук. Пирамида уровней разработки представлена на рисунке 11. Где в основе лежит описание стандартов, а на поверхности главные цели.

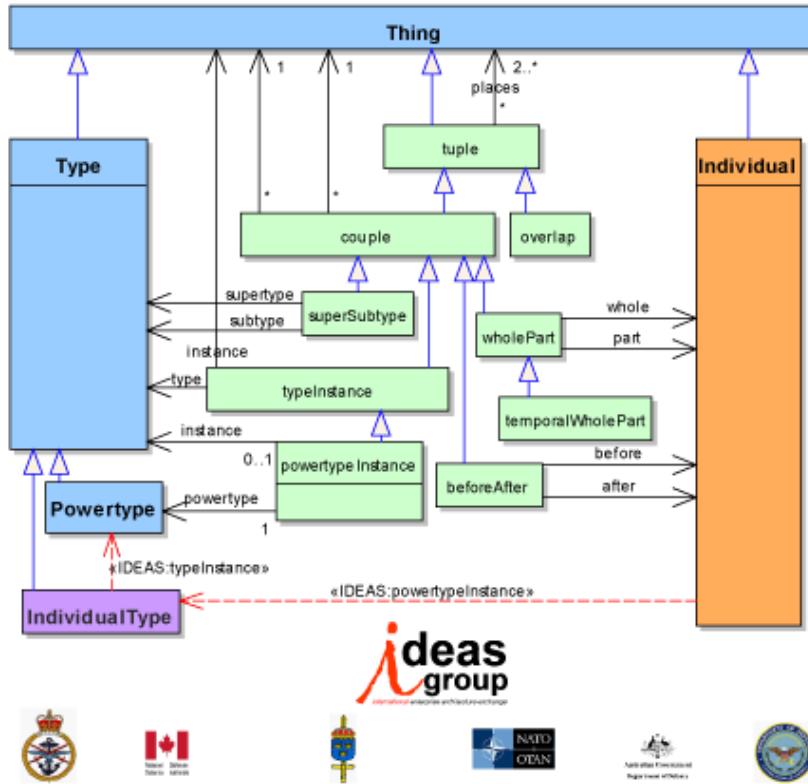


Рисунок 10 — верхние уровни онтологии DoDAF¹



На рисунке 12 представлена метамодель DoDAF описывающая взаимосвязи между основными понятиями метамодели DoDAF. Модель берет свое начало с 1980 года и содержит в себе шесть пунктов, которые объединены ключевым понятием “действие”:

1. описание данных (Data Description) — отвечает на вопрос ЧТО (включает и описание Ресурсов, кроме самих данных);

¹ DoDAF Formal Ontology: [Электронный ресурс]. URL: https://dodcio.defense.gov/Library/DoD-Architecture-Framework/dodaf20_ontology1/

2. описание функции (Function Description) — отвечает на вопрос КАК (содержит также описание исполнителя, который выполняет функции и учитывает связанные с ними Действия, Правила и Условия);
3. описание сети (Network Description) — ГДЕ;
4. описание участников (People Description) — КТО (включает и сами организации);
5. описание времени (Time Description) — ГДЕ;
6. описание мотивации (Motivation Description) — ПОЧЕМУ (с расширением, что включает описание требований к функциям).

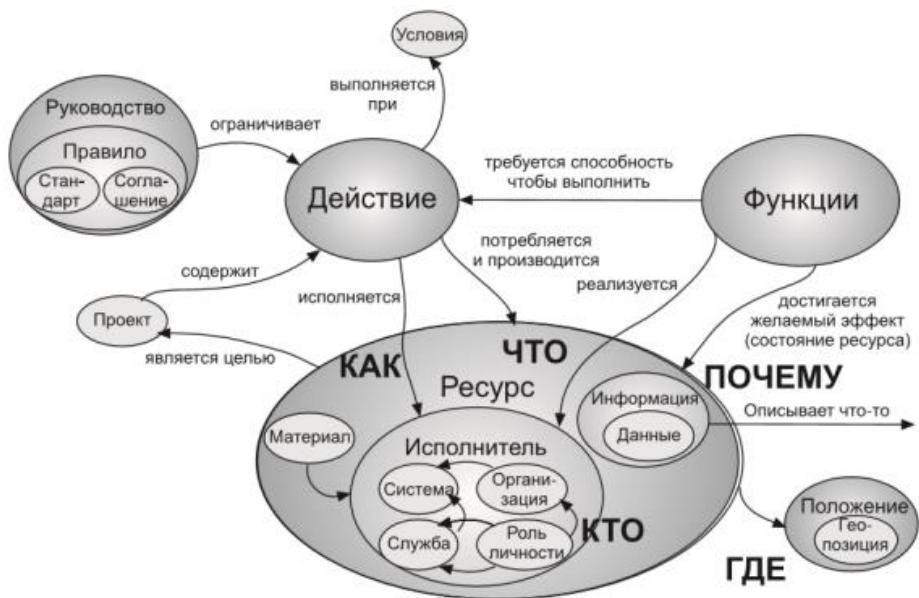


Рисунок 12 — метамодель DoDAF

Документация данной онтологии содержит в себе 52 тома, что является неоспоримым минусом. В описание этих томов входят следующие сведения:

- общее описание (All Viewpoint)
- описание сервисных компонентов (Capability Viewpoint) — описывает требования к возможностям системы; время, необходимое для развертывания системы, возможности развертки;
- описание данных и информации (Data and Information Viewpoint) — описывает взаимодействия между данными системы и согласование архитектуры данных;

- описание операций (Operational Viewpoint) — включает операционный сценарий, активности и требования по поддержке сервисных компонентов;
- описание проекта (Project Viewpoint) — описывает отношения между операционными требованиями и требованиями к возможностям системы;
- описание сервисов (Services Viewpoint) — описывает идентификацию сервисов, сервисных элементов и их взаимодействий;
- описание системы (System Viewpoint) — описывает системы и соединения, поддерживающие работу Департамента Защиты;
- описание стандартов (Standard Viewpoint) — описывает некоторые стандарты, которые используются в разработке решений.

2.2.4. BORO метод

Современные организации сталкиваются с растущими проблемами в развитии своих корпоративных информационных систем. Типичные проблемы включают системную интеграцию и переосмысление архитектуры предприятия. Основное решение всех этих проблем моделирование предприятия. В литературе было предложено много подходов к моделированию предприятий, некоторые из которых основаны на онтологии. Однако лишь немногие принимают основополагающую методологию для поддержки целого ряда моделей предприятий. Еще меньше людей считают реинжиниринг, основанный на данных, естественной отправной точкой для моделирования. Это подход, используемый справочной онтологией бизнес-объектов (BORO — Business Objects Reference Ontology). Она имеет два тесно переплетенных компонента: фундаментальную онтологию и методологию реинжиниринга. Первоначально они были разработаны для реинжиниринга корпоративных систем и впоследствии превратились в подходы к корпоративной архитектуре и системной интеграции. Вместе эти компоненты используются для систематического извлечения многоразовых и обобщенных

бизнес-моделей из существующих данных. Большинство из этих моделей были разработаны для контекста предприятия и успешно применялись в ряде коммерческих проектов в финансовой, оборонной и нефтегазовой отраслях. Методология реинжиниринга BORO основана на философии и онтологии реинжиниринга; она возникла на основе практики развертывания своей фундаментальной онтологии и была усовершенствована за последние 25 лет.

BORO включает в себя фундаментальную (или верхнюю) онтологию и тесно переплетенную методологию реинжиниринга информационных систем (ИС), поэтому термин BORO относится как к онтологии, так и к методологии. Первоначально компания BORO была задумана в конце 1980-х годов для удовлетворения особой потребности в прочном наследовании процесса реинжиниринга, а затем эволюционировала для решения более широкой потребности в разработке корпоративных систем "лучшим способом", другими словами, менее громоздким по сравнению с тяжеловесными методологиями того времен. Способом, позволяющим повысить уровень повторного использования и, как следствие, способным сократить усилия и затраты на (повторную) разработку, обслуживание и взаимодействие корпоративных систем. Таким образом, BORO – это онтология, признающая существование объективной реальности, в отличии концептуального идеализма, который рассматривает реальность как мысленно сконструированную. Реализм – это один из методологических вариантов, лежащих в основе BORO. Ключевой мотивацией для выбора реализма является то, что он увеличивает вероятность того, что различные разработчики корпоративных моделей будут представлять одни и те же бизнес-объекты.

Центральное место в BORO занимает верхний уровень – элементы, типы и кортежи, которые обеспечивают основу для остальной части онтологии. Верхний уровень обрамлен рядом явных онтологических выборов, описанных далее в этой работе. Они кристаллизуются в систему онтологических категорий, объектов верхнего уровня. Эти объекты верхнего уровня

характеризуются шаблонами верхнего уровня, включая шаблоны, описывающие типы отношений или типы кортежей.

Рассмотрим объекты верхнего уровня BORO. BORO – это категориальная фундаментальная онтология — та, в которой категории не пересекаются и образуют полное разбиение множества всех существующих вещей, и она не ограничена в том, что она может моделировать. Прагматически, как формальная онтология, она более целесообразно ориентирована на домены, которые были или должны быть формализованы, в частности автоматизированные компьютерные системы. Это связано с тем, что значительно меньше усилий требуется для построения формальной онтологии для домена, который уже формализован (например, где существуют компьютерные системы), чем тот, который никогда не был формализован (например, где все текущие процессы выполняются вручную).

Существует три категории верхнего уровня BORO: элементы, типы и кортежи. Каждый объект принадлежит к одной и только одной из трех категорий, которые, как уже упоминалось, определяются рядом метафизических выборов. Эти выборы означают, что в рамках BORO каждая категория имеет свои собственные критерии идентичности.

Элементы – это отдельные объекты, идентичность которых определяется пространственно-временной протяженностью (или протяженностью) элемента, то есть пространством и временем, которые он занимает. Боро упрощает вещи, утверждая, что материя и пространство и время тождественны.

Типы – это коллекции объектов любого типа. Идентичность типа определяется его расширением, совокупностью его состояний. Например, расширение типа лица – это совокупность всех людей. В BORO типы играют ту же роль, что и универсалии в других фундаментальных онтологиях.

Кортежи – это отношения между объектами. Идентичность кортежа определяется местом в кортеже. Примером может служить (Мария, Иван), в котором элементы Мария и Иван занимают места 1 и 2 в кортеже соответственно. Кортежи могут быть собраны в типы, называемые кортежами. Примером может служить parentOf, который представляет собой совокупность всех отношений между родителями и их детьми.

Между этими категориями существует система отношений онтологической зависимости.

Системный подход к реинжинирингу форматов сущностей становится возможным благодаря фундаментальной работе, проделанной по реинжинирингу парадигмы сущностей в парадигму объектов.

Метод BORO фокусируется на реинжиниринге данных – в частности, форматов сущностей в существующей устаревшей системе. Данные в большинстве систем содержат достаточно важных шаблоны проектирования, чтобы мы могли построить общие шаблоны проектирования, необходимые для всеобъемлющей бизнес-модели.

Рассмотрим этапы моделирования в рассматриваемом методе. Разобранные примеры имеют больше смысла, если сначала кратко рассмотреть схему метода. Метод делит реинжиниринг на два основных этапа:

- реинжиниринг формата сущностей существующей системы сущностей;
- реинжиниринг наших концептуальных шаблонов для формата сущности.

Элементы формата сущности реинжинириваются один за другим. Каждый реинжиниринг начинается со знака в формате сущности, из него идентифицируется сущность в "реальном мире".

Стоит следовать двум простым правилам при выборе порядка, в котором следует производить реинжиниринг различных элементов форматов сущностей. Это делает весь процесс реинжиниринга гораздо более простым.

Первое правило таково: реинжиниринг отдельных сущностей и знаков типов сущностей перед соответствующими им индивидуальными атрибутами и знаками типов атрибутов.

Второе правило для отдельных сущностей и типов сущностей — это реинжиниринг пары отдельных знаков сущностей и использование шаблонов для реинжиниринга их типа сущностей.

Первоначальная цель BORO состояла в том, чтобы дать возможность реинжиниринга устаревших систем, поскольку в то время (то есть в конце 1980-х и начале 1990-х годов) и до сих пор существует не так много используемых и проверенных методологий для этого. BORO эволюционировал в более эффективную и действенную общую методологию для ранних стадий моделирования развития систем по сравнению с существующими в то время методологиями. Эффективная реинжиниринг — это способность сохранить бизнес-знания, заложенные в существующую систему, для использования в новой системе и в то же время иметь возможность интерпретировать и представлять такие знания в более общем и многоразовом виде. Повторное использование всегда было в основе методологии BORO, и основополагающая онтология является ключом к семантической интерпретации и обнаружению обобщенных бизнес-моделей на основе исходных унаследованных данных.

Многократное использование существующих фундаментальных и предметных паттернов повышает эффективность, поскольку шаблоны, обнаруженные ранее, могут быть применены к задачам, которые уже были смоделированы, следовательно, сокращая время, необходимое для семантической интерпретации и моделирования унаследованного представления.

За последние 30 лет BORO был применен в других областях, связанных с предприятиями в различных отраслях промышленности, и лежит в основе стандартов взаимодействия и обмена данными, системной интеграции и архитектуры предприятия. Сегодня можно сказать, что BORO метод является одним из способов, позволяющим повысить уровень повторного использования и, как следствие, способным сократить усилия и затраты на (повторную) разработку, обслуживание и взаимодействие корпоративных систем.

2.3. Существующие решения с низким объемом программирования

На 2021 год существует около 15 известных платформ «без программирования»² и каждая из них выполняет те или иные функции. Каждую систему можно отнести к тому или иному виду, произвести классификацию — то есть распределить по тем или иным признакам. В данном разделе на основе трех самых распространенных платформ³ будут выявлены типовые задачи, которые выполняются платформами «без программирования», а также произведена их классификация по областям применения и используемым методам.

В рассмотрении будут участвовать такие платформы как:

- Comindware Business Application Platform (далее Comindware)
- ELMA365
- Creatio

Но прежде чем приступить к обзору существующих решений необходимо ввести понятие “платформа” — это набор компонентов или среда, которая позволяет решать различные задачи, которые позволяют закрыть бизнес потребности пользователей.

² MARKET.CNews [Электронный ресурс] - Режим доступа:
https://market.cnews.ru/research/lowcode_2021/table?p=review

³ tadviser [Электронный ресурс] - Режим доступа: <https://www.tadviser.ru/index.php/BPM>

2.3.1. Comindware

Начнем обзор с платформы Comindware. Данная платформа — это отличный инструмент для управления задачами, процессами и документами. Если смотреть глобально, все части данной системы являются функциями, которые связаны между собой, так платформа заточена на автоматизацию документооборота. Документооборот в автоматизированных системах представляет собой процесс связки задач и документов. На рисунке 13 можно рассмотреть все возможные решения платформы.

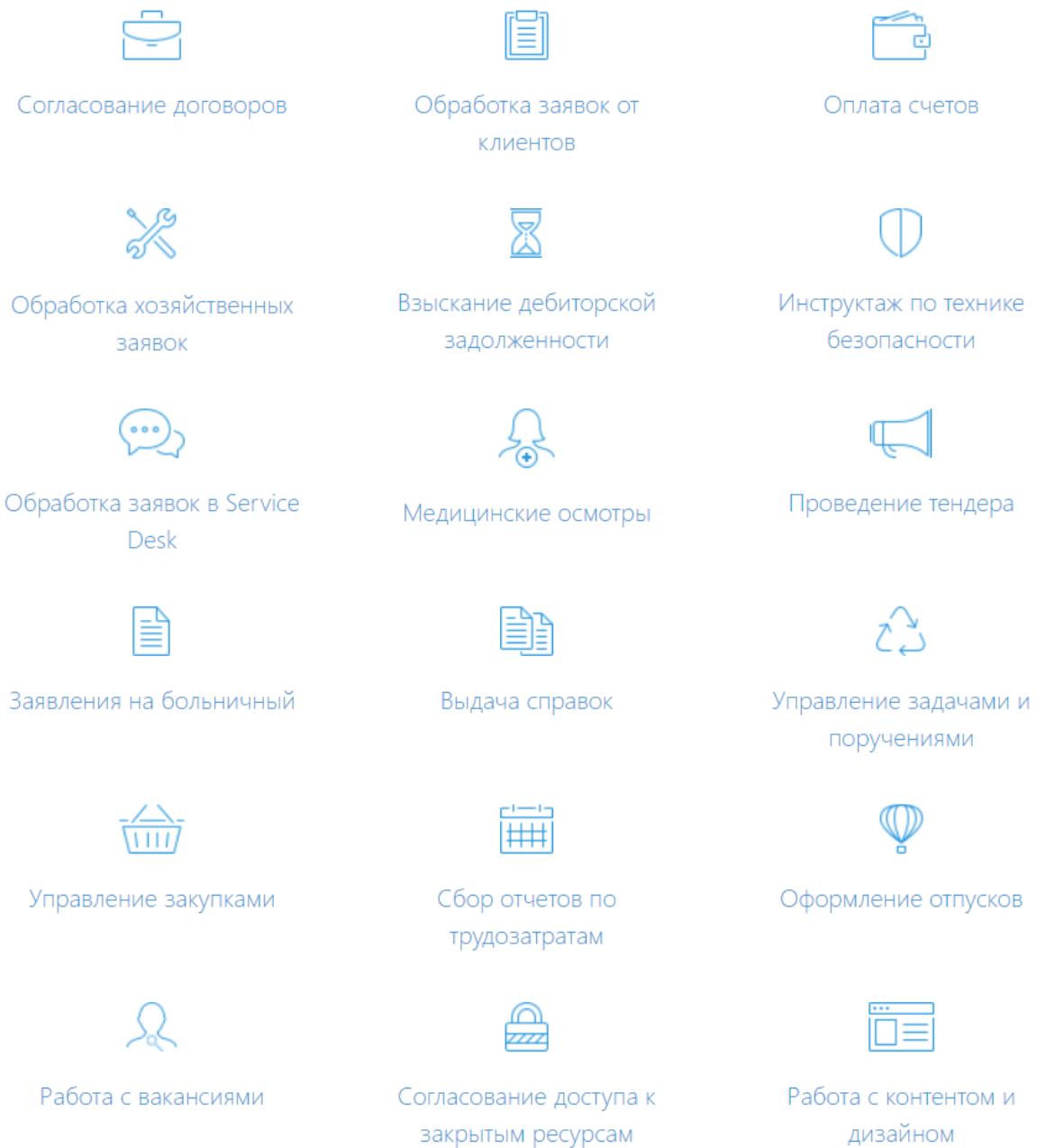


Рисунок 13 — процессы автоматизации в платформе Comindware.

Comindware является браузерным решением. Несмотря на это данная платформа не имеет своего “маркетплейса” на котором можно было бы выбрать тот или иной процесс автоматизации на основе данной платформы. Продукт предоставляется в рамках встроенных процессов, которые в основном направлены на автоматизацию следующих типовых решений:

- CRM: Клиенты и продажи
- HRM: Управление персоналом
- ITSM: Управление ИТ-услугами
- WMS: Управление закупками
- СЭД: Электронный документооборот
- FM: Управление финансами, бюджетирование
- TMS: Управление транспортной логистикой
- MDM: Управление нормативно-справочной информацией
- OMS: Управление заказами
- Сбыт: Управление отгрузками
- ERP: Гибкое управление бизнес-процессами

Также платформа имеет свои разработки в более специфичных отраслях, если ранее мы видели список решений который присутствует в любой компании и данные решения можно применить “из коробки” для любого предприятия, то есть более “узкие” решения, которые применимы для определенных отраслей:

- CRM-решение для нефтяной компании
- KPI-система для сети ресторанов
- Решение CRM для авиакомпании
- Решение CRM для брокерской компании
- СЭД-решение для научно-производственного предприятия
- Система управления проектами в строительстве
- BPM-система для центра сертификации
- Система автоматизации HR-процессов

Если с возможностями автоматизации мы разобрались, то как же строятся и настраиваются данные бизнес процессы? В платформе построение бизнес процессов происходит довольно просто: пользователь использует нотацию BPMN 2.0, где отстраивает свой бизнес процесс, где в дальнейшем, в каждой задаче можно провести настройку отображаемой формы (создание, отображение).

Comindware — это BPMS (Business Process Management Suite — класс программного обеспечения, которое заточено на управление бизнес-процессами в условиях частых изменений) система с набором процессов, где в основе лежит построение бизнес процессов в нотации BPMN. Подход BPMS в последнее время становится все более устаревшим, система по откликам пользователей, является простой и удобной в использовании, но здесь присутствует четкий “коробочный” набор, который сильно отстает от своих конкурентов, не имея свой “маркетплейс” в который постоянно добавляют новые решения, которые являются более гибкими и широко применимыми.

2.3.2. Creatio

Платформа Creatio является более современным примером решений «без программирования», по сравнению с Comindware. Она показывает более современные подходы, используемые при разработке корпоративных систем. Creatio тоже является браузерным SAAS (модель предоставления лицензии на программное обеспечение по подписке) решением, — то есть предоставлением услуги как сервис. Creatio построена на микросервисной архитектуре, что позволяет поддерживать быстро растущие потребности бизнеса к производительности, расширяемости и отказоустойчивости. На рисунке 14 представлена архитектура платформы Creatio, на которой можно рассмотреть имеющиеся микросервисы платформы.



Рисунок 14 — микросервисная архитектура Creatio.

Микросервисная архитектура позволяет поддерживать различное масштабирование компонентов, что в свою очередь, позволяет обеспечивать стабильную и надежную работу приложения круглые сутки. Также необходимо отметить, что при развертке приложения в рамках своей компании можно выбрать ту СУБД, которая уже используется на предприятии. Этой СУБД может быть одна из трех самых распространенных СУБД в мире Microsoft SQL Server, Oracle или PostgreSQL. С любой из этих СУБД платформа Creatio найдет “общий язык”.

Если про архитектуру приложения стало более понятно, то поговорим про решения «без программирования» данной платформы. Хотелось бы отметить, что Creatio имеет объемный по сегодняшним меркам маркетплейс, в нем можно найти:

- темплейты — позволяют просто и быстро начать использовать шаблоны процессов, дашборды;
- дополнения — помогают расширить функциональные возможности базовых продуктов платформы;
- коннекторы — позволяют произвести интеграцию с внешними сервисами и приложениями;
- решения — решения для конкретной области/отрасли, которые заточены на определенную специфику.

Часть из этих решений является бесплатными, так же из удобств — все решения разбиты по категориям. Если говорить о маркетплейсе, то он находится вне системы. Установка какого-либо дополнения не является трудной задачей и занимает всего пару минут, что тоже, несомненно, является преимуществом. На рисунке 15 представлен маркетплейс платформы Creatio.

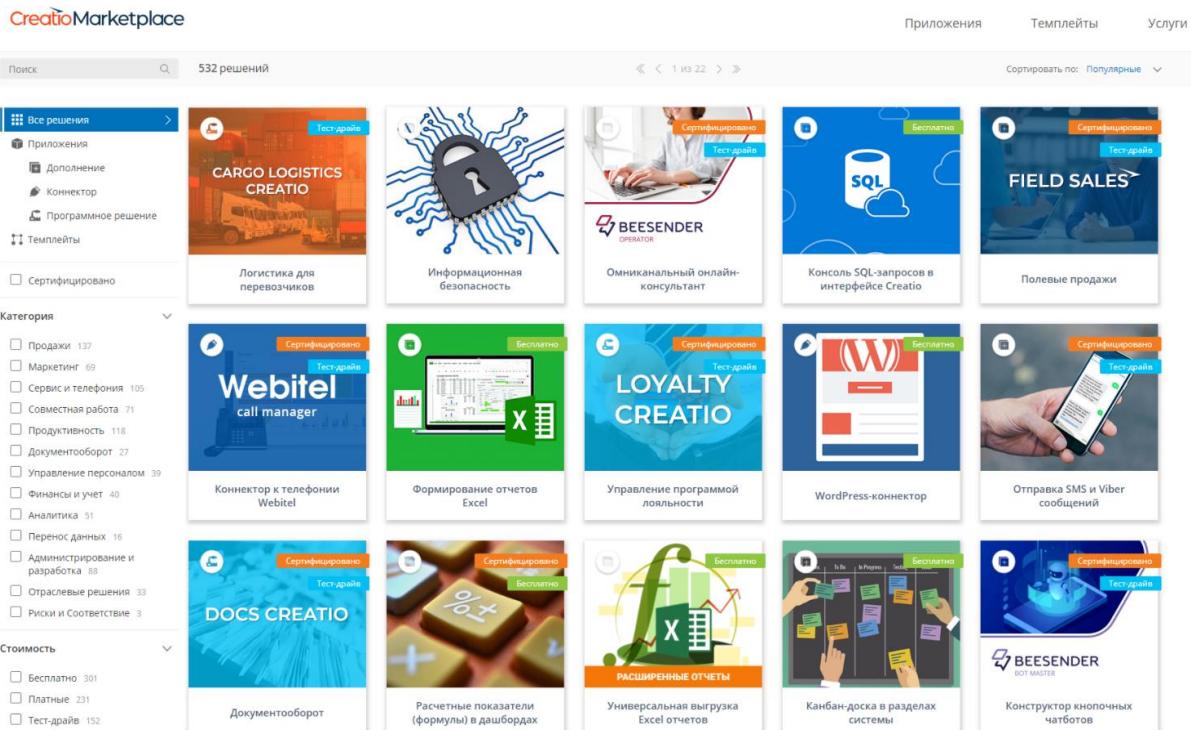


Рисунок 15 — Маркетплейс Creatio.

Платформа Creatio делится на две основные роли, как и Comindware, пользователь и разработчик. Где пользователь может произвести с процессами действия исполнения и создание (отрисовка) новых. Разработчики же занимаются более глубокой доработкой системы. Приложение использует принцип открытой платформы, где появляется возможность создавать свои разработанные модули и модифицировать имеющиеся, под свой функционал.

Creatio использует BPM-движок под капотом. Если же говорить о пользователях, которые выстраивают бизнес-процессы, то для них нет больших изменений. Пользователи, выстраивающие бизнес процессы, также используют визуальный дизайнер, который очень похож на BPMN нотацию. Но в то же время, в системах использующих BPM-движок происходит некоторое переосмысление нотации BPMN2, внешне же остается все

предельно и интуитивно понятно. На рисунке 16 можно рассмотреть построение процесса в приложении Creatio.

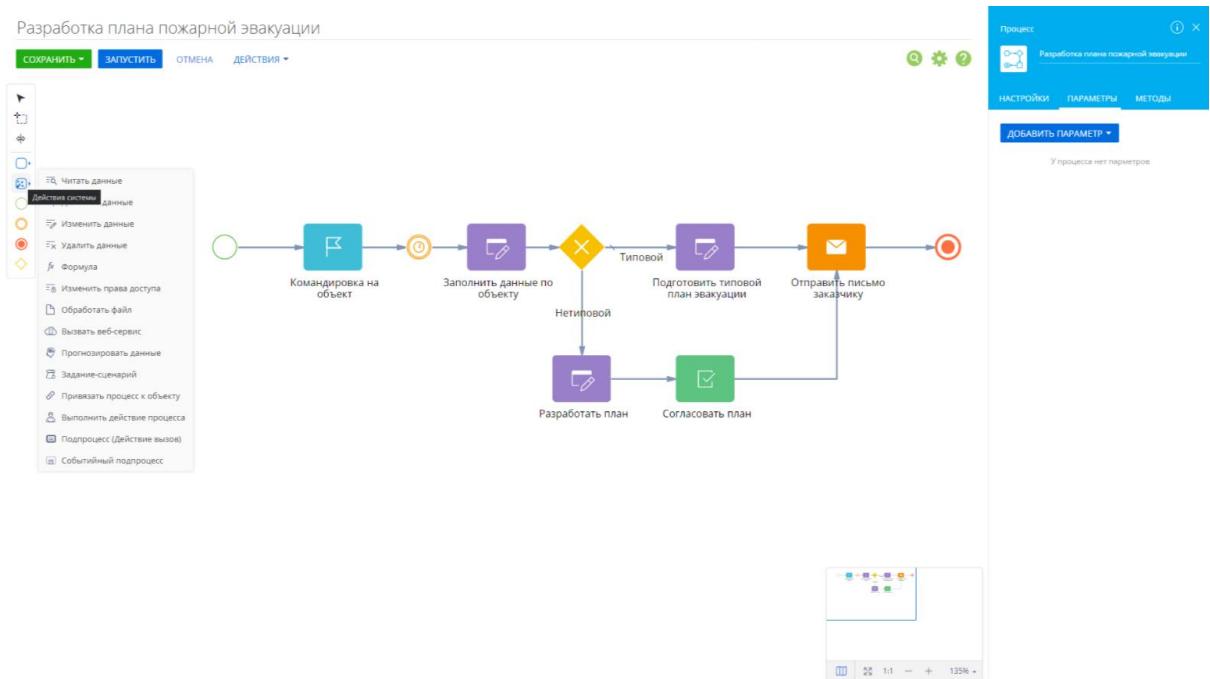


Рисунок 16 — построение бизнес-процесса в Creatio.

Creatio одна из самых первых систем «без программирования». Она имеет множество функций и возможностей, которые накоплены многолетним опытом. Ее разработка не стоит на месте и постоянно производятся доработки и внедрение нового функционала. Но она, как и все не идеальна и имеет свои минусы, которые отмечают опытные пользователи: система не совсем гибкая, как ее позиционируют и несмотря на то, что ее позиционируют как решение «без программирования», все же требует частого вмешательства разработчика. Подытожив, можно сказать, что система ведет свой путь к развитию, но пока все же является более гибкой, чем ее конкуренты, но более сложной.

2.3.3. ELMA365

Следующей платформой для обзора является ELMA365. Компания ELMA создала SaaS решение аналогично Creatio, но чуть лучше. ELMA365, как и Creatio, является BPM-решением для автоматизации бизнес-процессов компаний. Задумка системы в том, что все настройки может произвести аналитик, без привлечения программистов, где основа это отрисовка бизнес

процесса, но также можно прописать сценарии на TypeScript, чтобы усовершенствовать свой бизнес-процесс, и применить свои знания HTML.

Идея моделирования ELMA365 аналогична предыдущим решениям — используется графический интерфейс, в котором прорисовывается схема в нотации BPMN 2.0. Как схема будет готова в ней определяются данные, над которыми должны работать участники процесса. Также в схеме указывается откуда эти данные попадают в процесс и что требуется от участников данного процесса. На рисунке 17 представлено окно графического дизайнера системы.

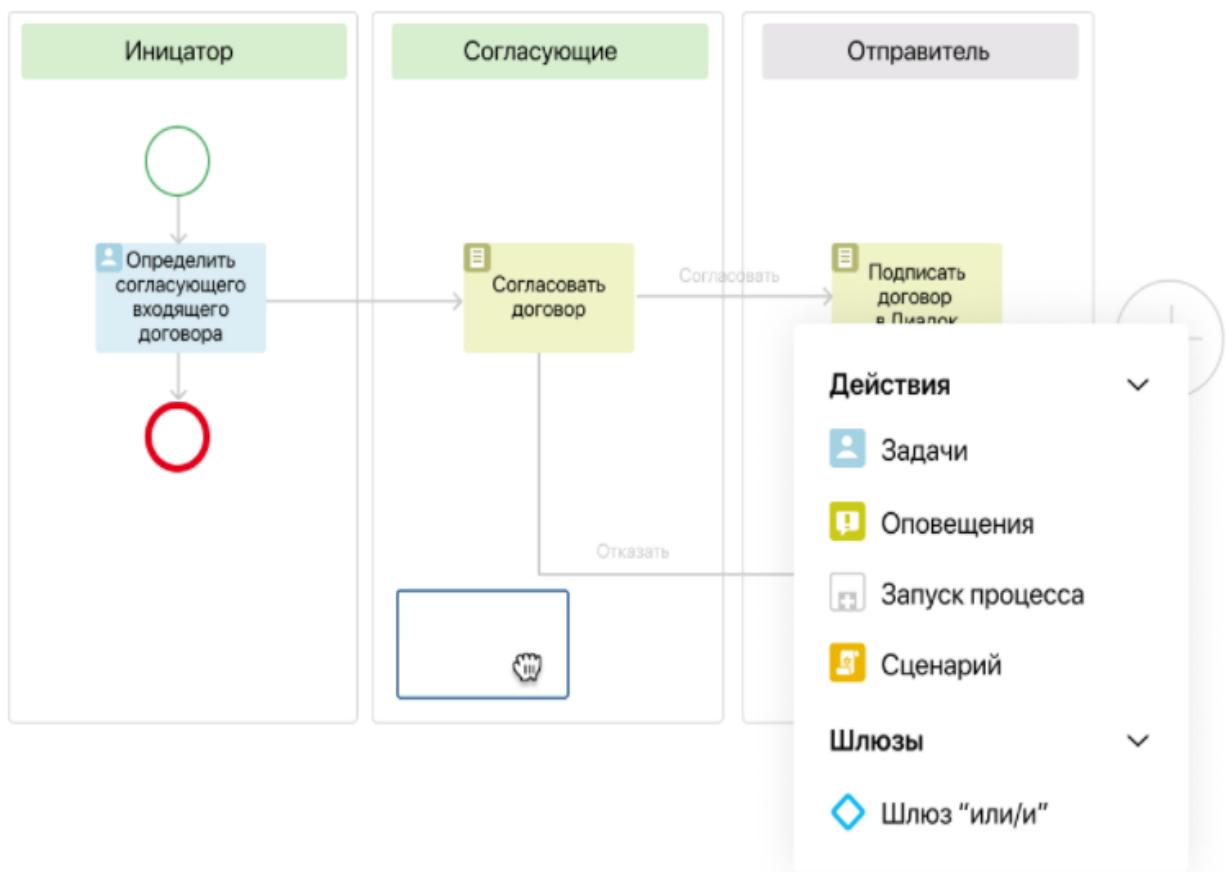


Рисунок 17 — вид графического дизайнера ELMA365

Аналогично платформе Creatio BPMN здесь тоже имеет более расширенные возможности, к примеру, есть специальные блоки, разработанные ELMA помогающие интерпретировать определенные действия, такие как подписание и согласование и т.п. На рисунке 18

представлено графическое изображение расширения нотации BPMN 2.0 специально для данной системы.

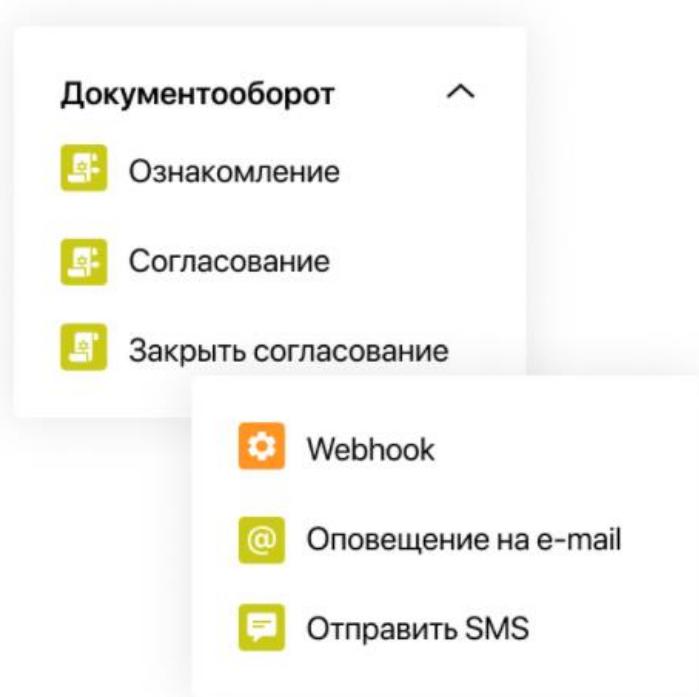


Рисунок 18 — дополнительные возможности нотации в системе

Так же система подразумевает, что если действий недостаточно, то их можно реализовать с помощью сценария — написания скрипта. На любом этапе моделирования может встать вопрос о том, что нужно автоматизировать ту или иную часть работы, к примеру, посчитать какую-то строку и вывести о ней сведения. Для этого можно использовать возможность написания сценария, который пишется на TypeScript и работает внутри форм или отдельных операций. Среда разработки адаптирована для людей, не занимающихся программированием профессионально. Существуют подсказки и подсветка синтаксиса, что делает работу более удобной. На рисунке 19 — можно увидеть пример написания скрипта в системе.

После того, как будет готов бизнес-процесс, можно произвести его отладку в режиме интерактивном режиме. В случае ошибки система укажет о ней, что позволит сразу же отладить процесс.

Сценарий

Сценарий

```

1  queryDatabase({ username: 'Arfat' })
2  .then((user) => {
3    const image_url = user.profile_img_url;
4    return getImageByURL('someServer.com/q=${image_url}')
5    .then(image => transformImage(image))
6    .then(() => sendEmail(user.email))
7  })
8  .then(() =>(C)
9  .catch(() => h

```

Context const Context: Item <Context\$\$Data>

case

catch

class

continue

Рисунок 19 — пример написания скрипта в ELMA365

Если же говорить об архитектуре приложения, то она тоже имеет микросервисную архитектуру, которую можно рассмотреть на рисунке 20. Предусмотрена возможность интеграции с различными типами СУБД.

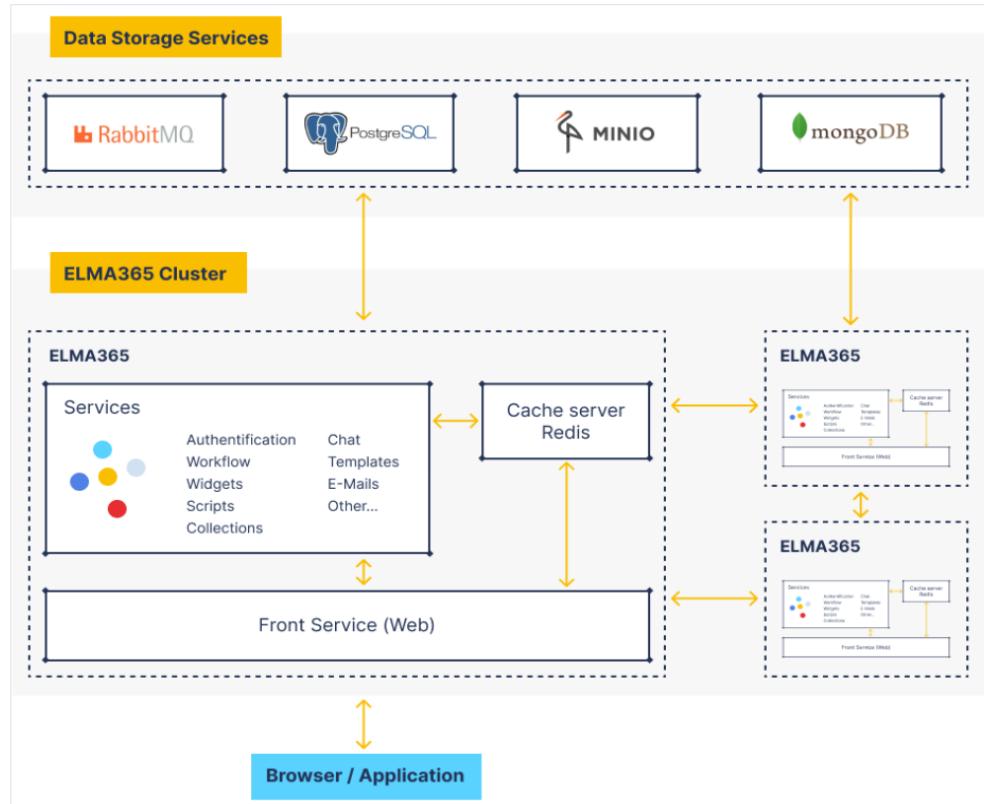


Рисунок 20 — микросервисная архитектура ELMA365

На текущий день ELMA365 является самой внедряемой платформой «без программирования» в России и это объяснимо:

- не является импортным производством, а значит является гарантом того, что поставщик по тем или иным внешним причинам не сможет покинуть рынок;
- имеет более гибкий подход по сравнению с аналогами — более легкая и дружелюбная для пользователей;
- имеется свой маркетплейс, который частично может закрыть потребность к расширению функционала;
- есть возможность внесения изменения в сценарии разработки силами системного аналитика, а не программиста.

2.3.4. Типовые задачи автоматизации бизнес-процессов

На основе анализа трех платформ, которые были рассмотрены ранее, можно выделить типовые задачи, которые решают системы «без программирования»:

- создание информации;
- передача информации;
- хранение информации;
- поиск информации;
- согласование документов/информации;
- оповещение о тех или иных событиях;
- контроль исполнения;
- настраивание автоматических действий/выгрузок;
- получение информации;
- управление закупками.

Это основной минимум, который пытаются автоматизировать в первую очередь на предприятиях. Вне зависимости от направления деятельности в компании, эти задачи будут идентичны. Системы с подходом “без

программирования” позволяют построить процесс и с минимальными затратами, как временными, так и финансовыми, автоматизировать их на основе уже устоявшегося бизнес-процесса в компании, но еще не оцифрованного.

2.3.5. Классификация существующих решений систем “без программирования”

Системы “без программирования” можно классифицировать по используемым методам, например, low-code или же no-code решения.

Low-code — метод разработки по принципу drag-and-drop (работа по принципу оперирования элементами дизайна, с предварительно встроенными в них значениями) в котором присутствуют элементы ручного кода. Здесь мы работаем с объектами, в которых уже прописан основной код, настраиваем его под наши нужды и получаем готовый функционал. Но если нам нужно индивидуальное решение, то необходимо написать код. То есть данный метод не исключает классическую разработку и программистов, но при этом помогает ускорить их работу.

No-code — метод создания приложений по тому же принципу drag-and-drop, но уже без навыков программирования или разработчиков. Отличается от low-code решений полным отсутствием создания кода или внесения в него изменений. Это позволяет свести затраты на разработку к минимуму.

Также можно классифицировать существующие решения по области применения, обычно данный подход применяют для сложных и не сильно нагруженных задач, где необходима автоматизация потоков задач, автоматизация бизнес-процессов, множественная интеграция систем и API, автоматизации документооборота и роботизации процессов.

2.4. Выводы

В данной главе были рассмотрены основные теоретические понятия «бизнес-процесса», а также его возможные вариации построения. Рассмотренные средства моделирования в данной главе позволили понять, как

необходимо моделировать бизнес-процессы для того что бы взаимодействовать с процессом на системном уровне.

С введением основных теоретических понятий был произведен анализ трех существующих решений, который позволил раскрыть основные возможности существующих платформ «без программирования», узнать, как реализованы данные платформы, по какому принципу они работают и что содержат в себе. Также в работе была произведена классификация решений без программирования по используемым методам разработки приложений.

Глава 3. Проектирование средств автоматизации бизнес-процессов

В текущих реалиях существует множество различных компаний, у которых есть своя структура и свои бизнес-процессы. Каждая в той или иной степени нуждается в автоматизации своих внутренних процессов, для того чтобы освободить или оптимизировать работу своих сотрудников, для повышения показателей работы и ведения понятного всем работникам процесса. В данной работе будет рассмотрена компания, которая имеет небольшую структуру. На рисунке 21 представлена структура компании.

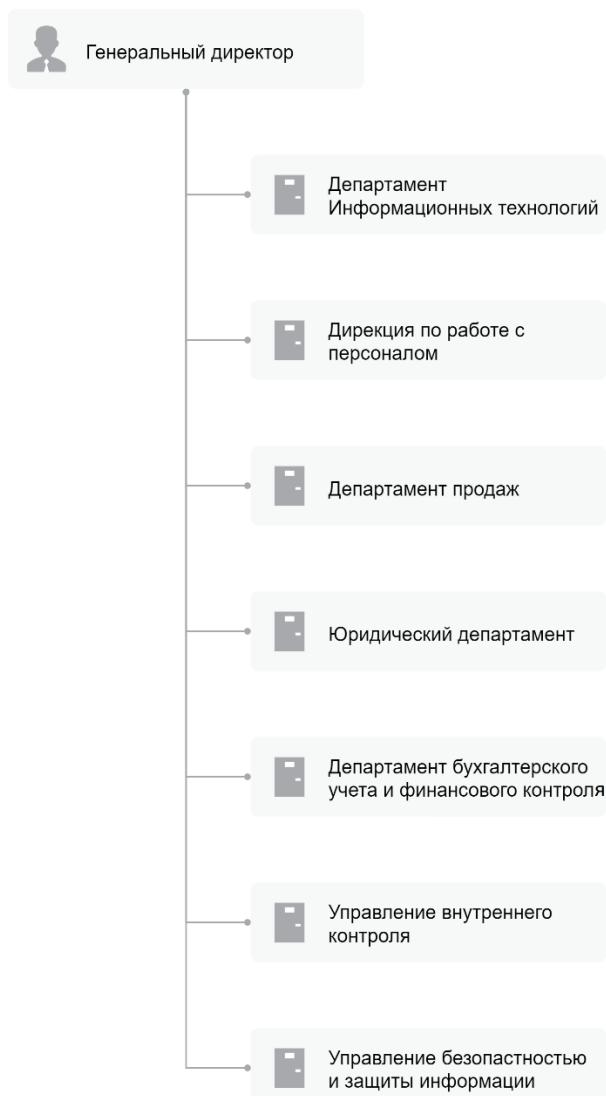


Рисунок 21 – организационная структура компании

В данной работе мы можем рассмотреть возможности работы в системе любого работника из любого подразделения, то есть рассмотреть основные функции, которые они могут выполнять в публичной части приложения. Но особое внимание мы уделим процессам, которые могут выполнять работники из подразделения по работе с персоналом и затронем некоторые аспекты, которые могут выполнять работники информационного подразделения. Для того что бы понять, как выглядеть структура подразделений, работу которых мы будем рассматривать, на рисунке 22 представлена подробная информация о составе отдела по работе с персоналом.

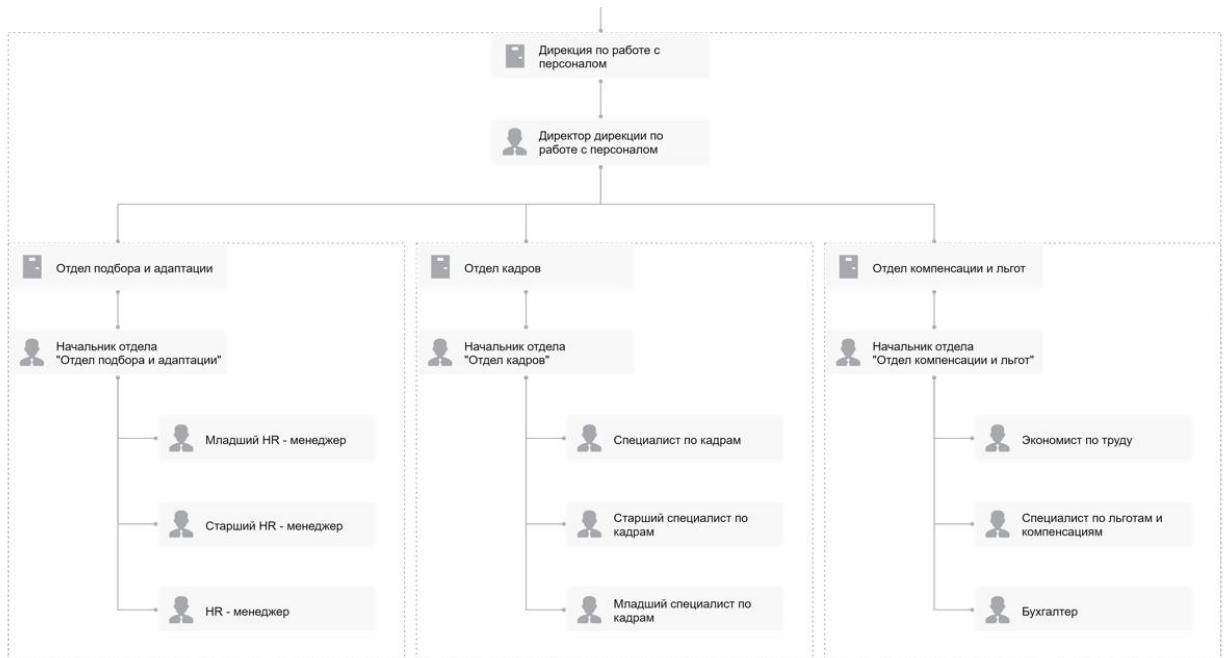


Рисунок 22 – структура дирекции по работе с персоналом

Прежде чем приступить к моделированию, рассмотрим существующие варианты. Рассмотрим самые широко используемые методологии – BPMN и UML, рассмотрим их, так как на текущем этапе целью работы является формализация требований и определение основного функционала системы, а уже потом, на основе реализуемой схемы, разбор пользовательских сценариев. Для определения основного функционала, мы должны отобразить возможность различных ролей выполнять различные функции. Для такого рода отображения схем хорошо подходит use case диаграмма (диаграмма

вариантов использования) языка UML. Именно данная диаграмма позволяет полностью, без лишних погружений в детали, понять и утвердить основной функционал системы. Именно в диаграмме вариантов использования мы можем отобразить все группы пользователей и перечислить их функционал в удобном и понятном языке для всех заинтересованных лиц.

На рисунке 23 и 24 можно увидеть возможные варианты использования пользователями проектируемой системы. На данных рисунках представлен функционал конечной системы, как можно заметить основную часть рутинной работы с персоналом предполагается перенести в цифровой формат. Это позволит существенно сократить расходы различного рода согласования внутри компании и между сотрудниками.

Если рассмотреть подробно в системе имеется 4 основные роли:

1. Сотрудник компании;
2. hr - менеджер;
3. hr – кадровик;
4. бухгалтер.

Каждая из роль подразумевает свои уровни доступа. Внедряя в систему не только сотрудников, которые работают с персоналом мы значительно можем увеличить функционал системы, а значит увеличить возможность переноса функционала. Т.е. мы автоматизируем работу не только для части сотрудников внутри отдела, а для всей компании в целом. В данной системе можно увидеть свой профиль и свои показатели и расчеты, а также загрузить и согласовать документы или заявки, если мы в систему с ролью «сотрудник компании». Если же мы являемся непосредственно сотрудником отдела управления персоналом, данная система позволит работать с уже имеющимися сотрудниками и согласовывать их заявки, что позволяет свести бумажную волокиту к минимуму, работать с вакансиями, с поиском новых сотрудников, а также формировать различного рода отчеты всего лишь настроив единожды необходимый шаблон, что в разы уменьшает рутинные

процессы. На рисунке 23 и 24 представлены use case диаграммы (диаграммы вариантов использования). Данные схемы описывают все возможности, которые может выполнять пользователь системы с той или иной ролью. В схеме описаны все возможности, которые могут быть реализованы в системе. В работе будут рассмотрены только часть вариантов бизнес-процессов. На рисунке 23 представлены возможности пользователей публичной части приложения, которая доступна всем сотрудникам компании. На рисунке 24 представлена use case схема для части отдела по работе с персоналом. У данных пользователей есть так же возможности, что и у пользователей публичной части, но также присутствуют возможности работы с функционалом, который непосредственно относится к возможностям отдела, работающего с персоналом.

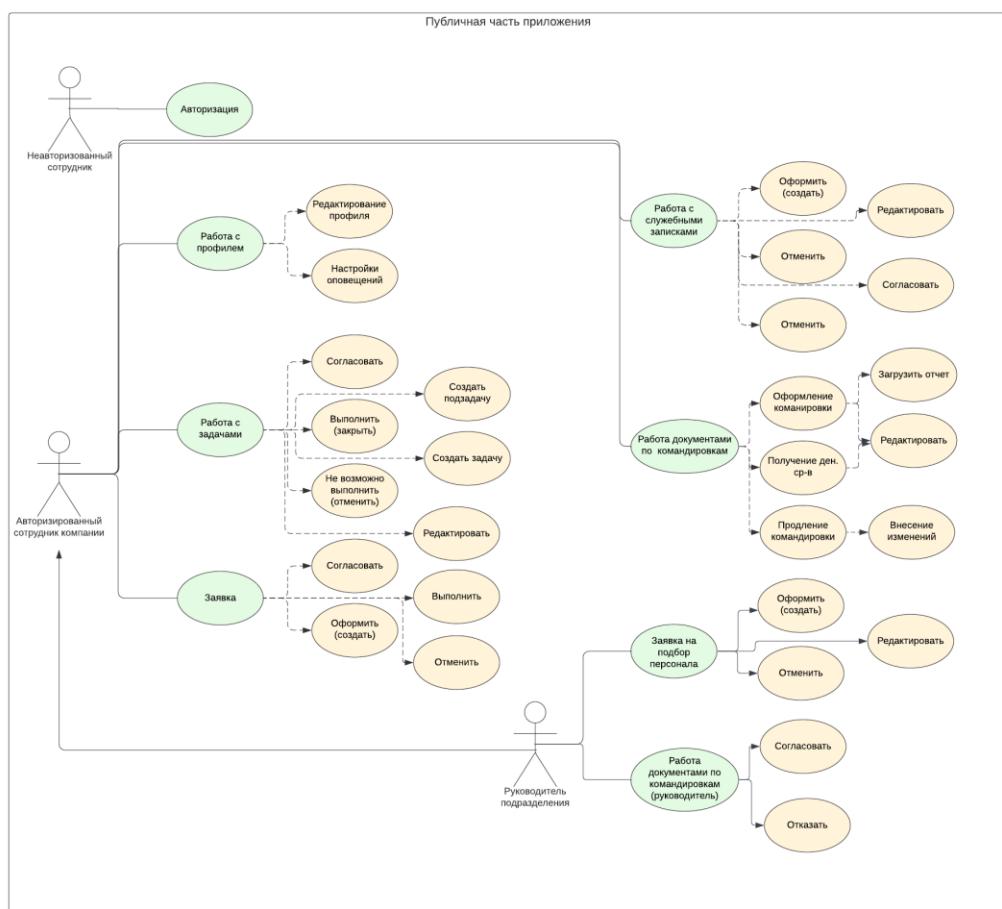


Рисунок 23 – use case публичной части проектируемой системы

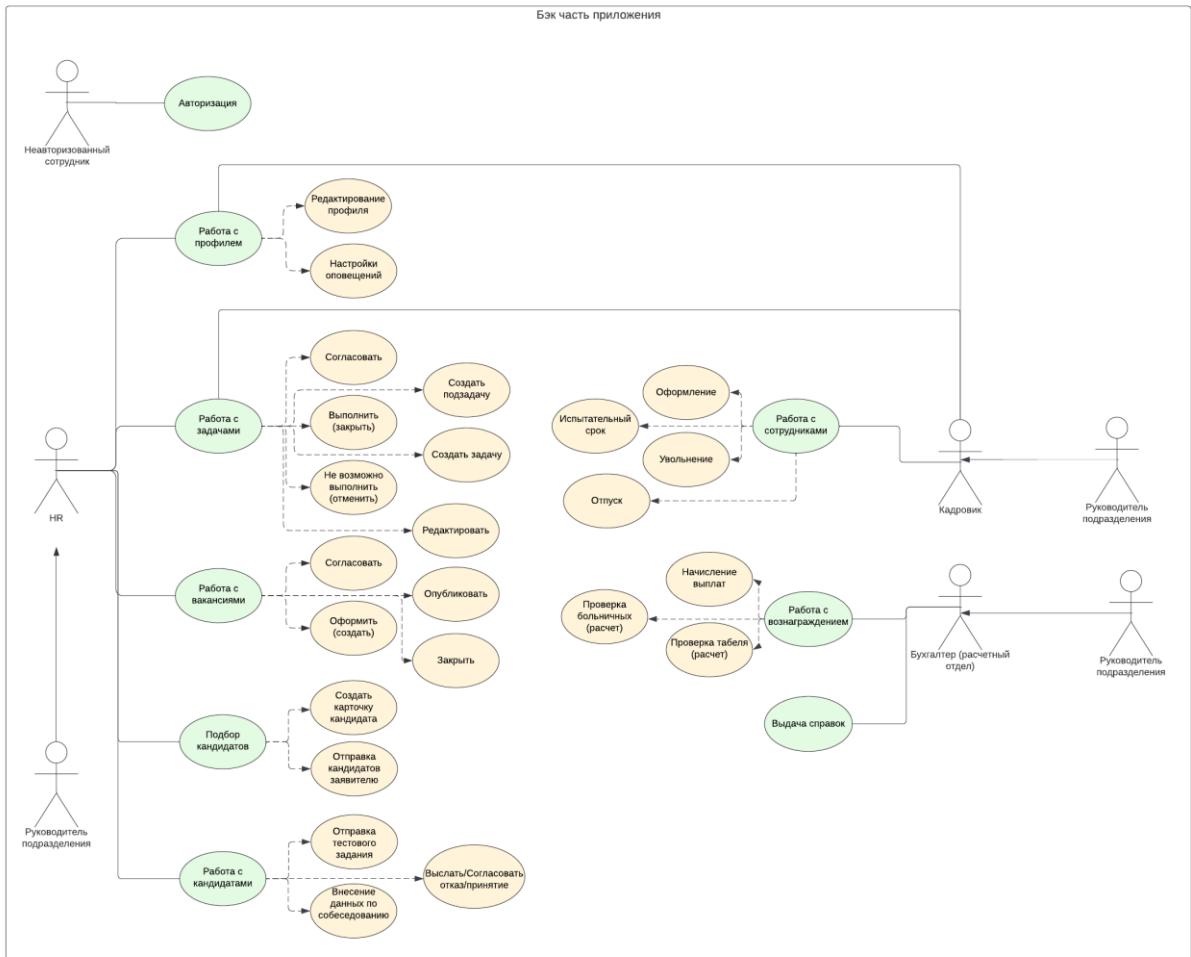


Рисунок 24 – use case диаграмма, описывающая возможности работы отдела по работе с персоналом

Все данные находящиеся в системе необходимо где-то хранить. При разработке приложения «без программирования» нам необходимо произвести подключение базы данных. Как было рассмотрено ранее, множество текущих решений позволяет подключить множество различных СУБД. В качестве примера, в данной работе будем использовать уже имеющуюся базу данных предприятия. В качестве описания базы данных будет представлена ER-диаграммы или диаграмма «сущность-связь». Именно данный тип диаграмм позволяет наглядно понять какие сущности присутствуют в базе, из каких атрибутов она состоит и как они связаны между собой. На рисунке 25 представлена часть базы данных. Данное представление, на рисунке 25, является частью целой базы данных компании, так как мы рассматриваем

работу именно с отделом по персоналу, а значит и работаем с данными персонала компании.

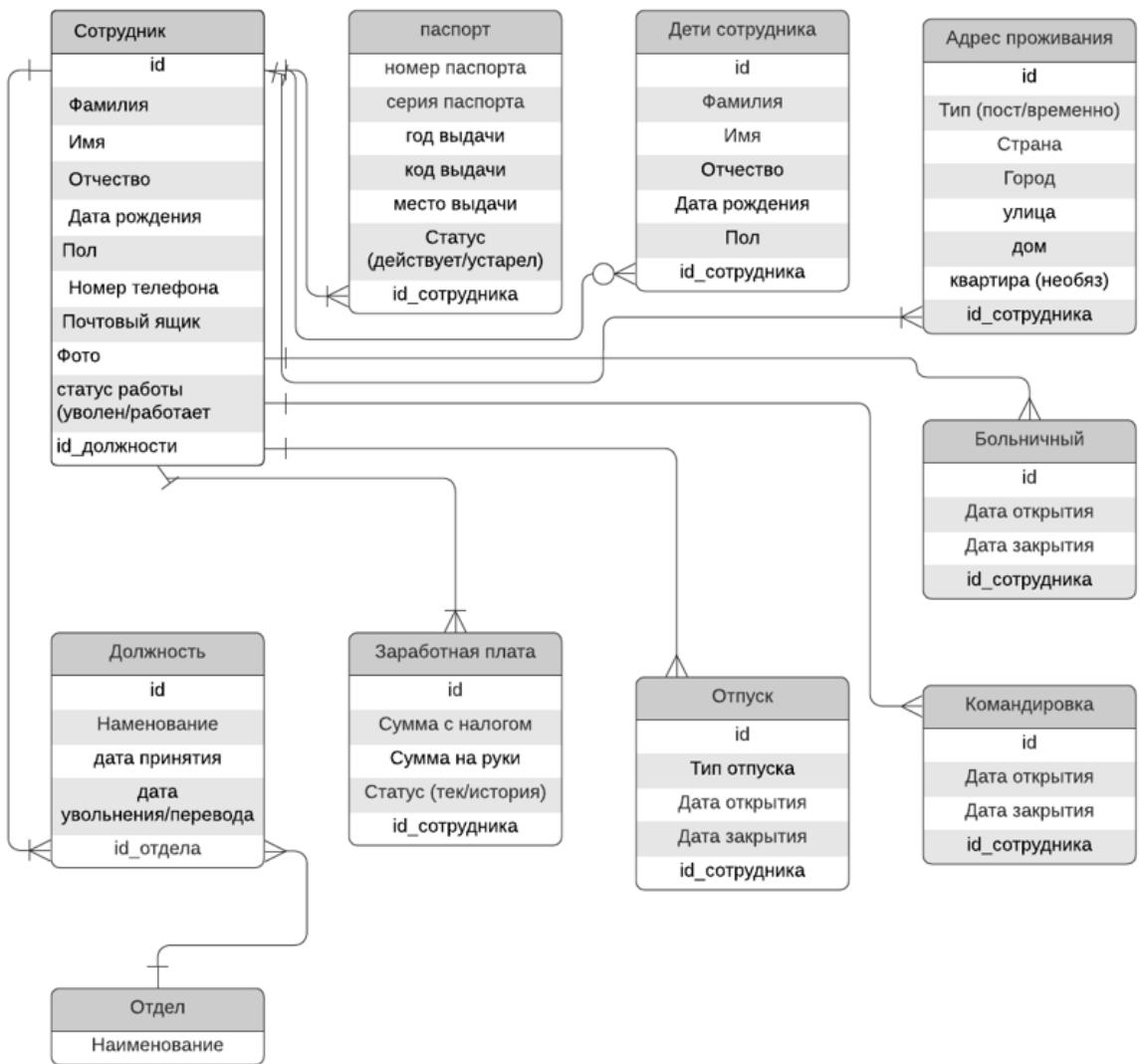


Рисунок 25 – ER-диаграмма

3.1. Построение публичных бизнес процессов

Начнем построение публичных бизнес-процессов, так мы сможем отследить как некоторые заявки и задачи попадают к менеджерам по персоналу, чтобы затем просмотреть историю того, как они работают с поступившими обращениями.

Рассмотрим блок работ по созданию задач. Как ранее говорилось, любой авторизированный пользователь системы может создать задачу для другого

пользователя или для себя. На рисунке 26 изображен бизнес-процесс работы с задачей. При разработке приложения на платформе «без программирования» построение бизнес-процесса является основой той или иной части автоматизируемого процесса компании. Изображенная на рисунке 26 схема представляет собой рабочий процесс, где каждый блок в схеме бизнес-процесса имеет свою функциональную особенность, о которой будет подробно рассказано.

Рассмотрим более подробно каждый шаг создания задачи. На старте мы должны создать задачу, в которой должны указать данные по ней, согласующих и участников (если они имеются) и исполнителя, а также оповестить их о готовности или не возможности исполнения задачи.

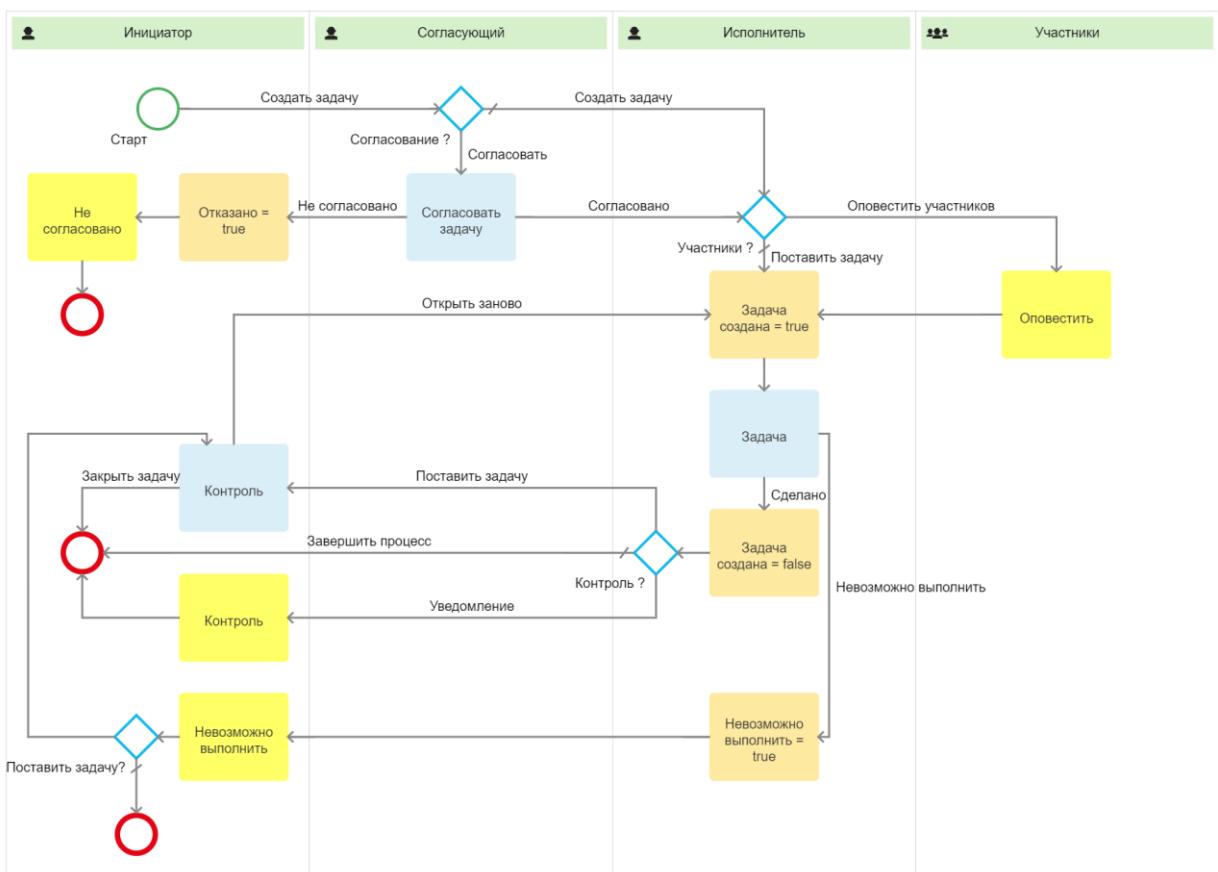


Рисунок 26 – модель бизнес процесса «Задача»

Для того что бы создавать формы процесса нам необходимо определить какие атрибуты нам необходимы, то есть нам надо определить контекстные переменные. Контекстные переменные нам необходимы, что бы построенный бизнес-процесс реагировал на тот или иной переход из одного состояния в

другое. При разработке приложений «без программирования» атрибуты создаются пользователям в специальном окне, где пользователь может задать тип поля (число, роль, текстовую строку), а также назвать атрибут определенным именем. Возможность выбора типа переменной уже заложена в платформе при разработке системы, сейчас же администратор системы создает лишь бизнес-процесс и вносит данные в форму. На рисунке 27 можно увидеть, как создается список переменных, которые необходимы для отображения в форме и перехода из одного статуса в другой. Вид формы может меняться в зависимости от того, какой тип переменной мы выбрали – могут появляться дополнительные поля или же дополнительные условия.

The screenshot shows a dialog box titled 'Создать свойство' (Create property). It contains several input fields and dropdown menus:

- Отображаемое имя***: A text input field with placeholder text 'Имя свойства, которое будет отображаться на формах'.
- Имя свойства***: A text input field with placeholder text 'Имя свойства, которое будет использоваться в скриптах и отчетах'.
- Подсказка**: An empty text input field.
- Тип***: A dropdown menu set to 'Число' (Number). Other options include 'Целое' (Integer) and 'Дробное' (Decimal). There is also a checkbox for 'По умолчанию' (Default).
- По умолчанию**: A checkbox labeled 'Поиск и сортировка по полю' (Search and sort by field).

At the bottom are three buttons: 'Создать' (Create), 'Создать и новое' (Create and new), and 'Отмена' (Cancel).

Рисунок 27 – создание полей для отображения в форме

Теперь, когда у нас присутствует готовый бизнес-процесс и контекстные переменные мы можем приступить созданию форм и настроек перехода из одного состояния в другое. На рисунке 28 представлен список контекстных переменных, с которыми мы будем в дальнейшем работать.

Как видно на бизнес-процессе работы с задачей (рис. 26) у нас есть 4 категории пользователей:

1. инициатор;
2. согласующий;
3. исполнитель;
4. участники.

Контекстные переменные			
Отображаемое имя	Имя свойства	Тип	Поиск и сортировка
			Полнотекстовый поиск
Дата начала	start_date	Дата/время	<input type="checkbox"/>
Файлы	files	Файлы	<input type="checkbox"/>
Контроль	kontrol	Категория	<input type="checkbox"/>
Отказано	rejected	Выбор «да/нет»	<input type="checkbox"/>
Комментарий согласующего	approverComment	Строка	<input type="checkbox"/>
Участники	participants	Пользователи	<input checked="" type="checkbox"/>
Задача создана	task_created	Выбор «да/нет»	<input checked="" type="checkbox"/>
Сделать до	end_date	Дата/время	<input type="checkbox"/>
Описание	description	Строка	<input type="checkbox"/>
Согласующий	approver	Пользователи	<input type="checkbox"/>
Исполнитель	executor	Пользователи	<input checked="" type="checkbox"/>
Комментарий исполнителя	executorComment	Строка	<input type="checkbox"/>
Невозможно выполнить	impossible	Выбор «да/нет»	<input type="checkbox"/>
Процент выполнения	percent	Число	<input checked="" type="checkbox"/>

Рисунок 28 – перечисление атрибутного состава заявки

В рассматриваемом БП все перечисленные роли может занимать любой пользователь системы, который занесен администратором в ту или иную группу пользователей с доступом к процессу, если же существует потребность, то администратор может сделать настройки и дать права доступа для любой из категорий пользователей особые права доступа. На рисунке 29 представлен вариант, когда мы выбираем исполнителя из структуры организации, на основе должности/департамента/отдела.

Рисунок 28 – установка групп пользователей, которые отвечают за часть бизнес-процесса

Начнем с начала бизнес-процесса, который инициирует инициатор – любой сотрудник, работающий в системе. Инициатор инициирует процесс путем создания задачи через форму. В условиях используемой системы «без программирования» форма имеет стандартный вид, но что бы отобразить поля необходимо их задать. Поскольку ранее мы перечислили список контекстных переменных, то мы можем выбрать из этого списка данные. На рисунке 29 представлен список полей, которые можно отображать в форме и какие поля являются обязательными для заполнения, а также определили, как будет называться форма.

The screenshot shows a user interface for configuring a form. At the top, there are buttons for 'Форма' (Form), 'Создание задачи' (Create Task), 'Редактировать форму' (Edit Form), and '+ Создать форму' (Create New Form). Below these are buttons for '+ Добавить' (Add) and '→ Перенести' (Move).

The main area is divided into two columns. The left column, titled 'Контекст' (Context), lists variables: Название, Комментарий, Дата начала, Файлы, Контроль, Отказано, Комментарий согласующего, Участники, Задача создана, and Сделать по. The right column, titled 'Название на форме' (Name on Form), lists the same variables. To the right of each variable name are three columns: 'Свойство' (Property), a checkbox for marking it as required (indicated by a red star), and a delete icon.

Название на форме	Свойство		
Название	Название	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Описание	Описание	<input type="checkbox"/>	<input type="checkbox"/>
Согласующие	Согласующий	<input type="checkbox"/>	<input type="checkbox"/>
Файлы	Файлы	<input type="checkbox"/>	<input type="checkbox"/>
Сделать до	Сделать до	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Исполнитель	Исполнитель	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Дата начала	Дата начала	<input type="checkbox"/>	<input type="checkbox"/>
Участники	Участники	<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 29 – перечисление контекстных переменных, которые будут отображаться в форме создания задачи

Когда форма создания задачи сформирована, то в БП идет разветвление. Если в задаче был указан согласующий – то прежде чем она уйдет на выполняющего, согласующий должен либо не согласовывать процесс (как видно на БП, рисунок 26, процесс задачи заканчивается и пользователю, заведшему задачу, приходит уведомление), либо согласовать процесс и тогда, если есть участники задачи отправить им оповещение, если же участников нет, то мы создаем задачу для исполняющего.

Задача

The screenshot shows a user interface for creating a task. At the top left, there are two input fields: 'Тема*' (Topic) containing 'Задача' and 'Исполнитель*' (Executor) with a search bar placeholder 'Начните вводить текст для поиска элемента'. Below these are buttons for '+ Согласующий' (Approver) and '+ Участники' (Participants). Under 'Сделать до*' (Due Date), the date '05.04.2022' is selected with a calendar icon, and a checkbox 'Указать время' (Specify time) is available. A button '+ Дата начала' (Start date) is also present. The 'Описание' (Description) section contains a large text area with a placeholder 'Поддерживается разметка Markdown'. The 'Файлы' (Files) section has a '+ Файл' (File) button. In the 'Контроль' (Control) section, there is a button 'Отправить на проверку' (Send for review). At the bottom left are two buttons: 'Создать задачу' (Create task) in blue and 'Отмена' (Cancel) in grey.

Рисунок 30 – внешний вид формы создания задачи

После того, как задача создана, по умолчанию идет присваивание статуса создания задачи на исполняющего равное значению TRUE, таким образом задача для выполняющего считается созданной. Далее с задачу можно или выполнить, или завершить, с условием что ее невозможно выполнить. При этом форма согласования, контроля и выполнения процесса выглядят аналогично, добавляется только информация в боковом меню о прошедших этапах, что не дает пользователю запутаться, а также понять, что было с задачей ранее, какие изменения в нее вносились и какие работы проводились. Так же пользователь имеет возможность просмотреть комментарии к задаче. На рисунке 31 можно увидеть форму, основная часть которой создана так же через контекстные переменные, а боковое меню через виджеты, которые

добавлялись непосредственно через редактор формы.

The screenshot shows a task management interface. On the left, a main panel displays task details: Author (PA, Rita Alikova), Executor (empty), Due date (26 April 2022), Creation date (26 April 2022, 20:41), Description (Change the process of raw material delivery), and Files (empty). Below this is a sub-task section with a '+Task' button. On the right, a sidebar shows approval status (Approved) and execution details: Status (In Progress), Due date (by 26.04.2022), Scheduled in Calendar (26.04.2022), Creation date (04.2022 22:56), Author (PA, Rita Alikova), Executor (empty), and Participants (empty). At the bottom, there are two buttons: 'Done' (highlighted in blue) and 'Not possible to execute'. Below the main panel, a message history is shown:

notification@task-change-field-title
PA Аликова 26 апреля 2022 г., 22:57
Описание изменено автором:
Старое описание:
Сделано Невозможно выполнить

Рисунок 31 – форма отображения задачи в статусе в работе у исполнителя

Далее рассмотрим бизнес процесс создания заявки, в примере будет рассмотрена заявка на получение новой техники в компании. В бизнес-процессе присутствует три роли:

1. инициатор – любой пользователь системы;
2. руководитель – автоматически подтягивается системой и заполняется;
3. Системный администратор – любой сотрудник, который находится в группе пользователей системных администраторов, в которой могут находиться сотрудники с ролью системный администратор и техник (отдела информационного обеспечения).

На рисунке 32 можно увидеть бизнес процесс создания заявки, при переходе от одного состояния в другое в системе меняется статус заявки и идет оповещение пользователя, который завел заявку.

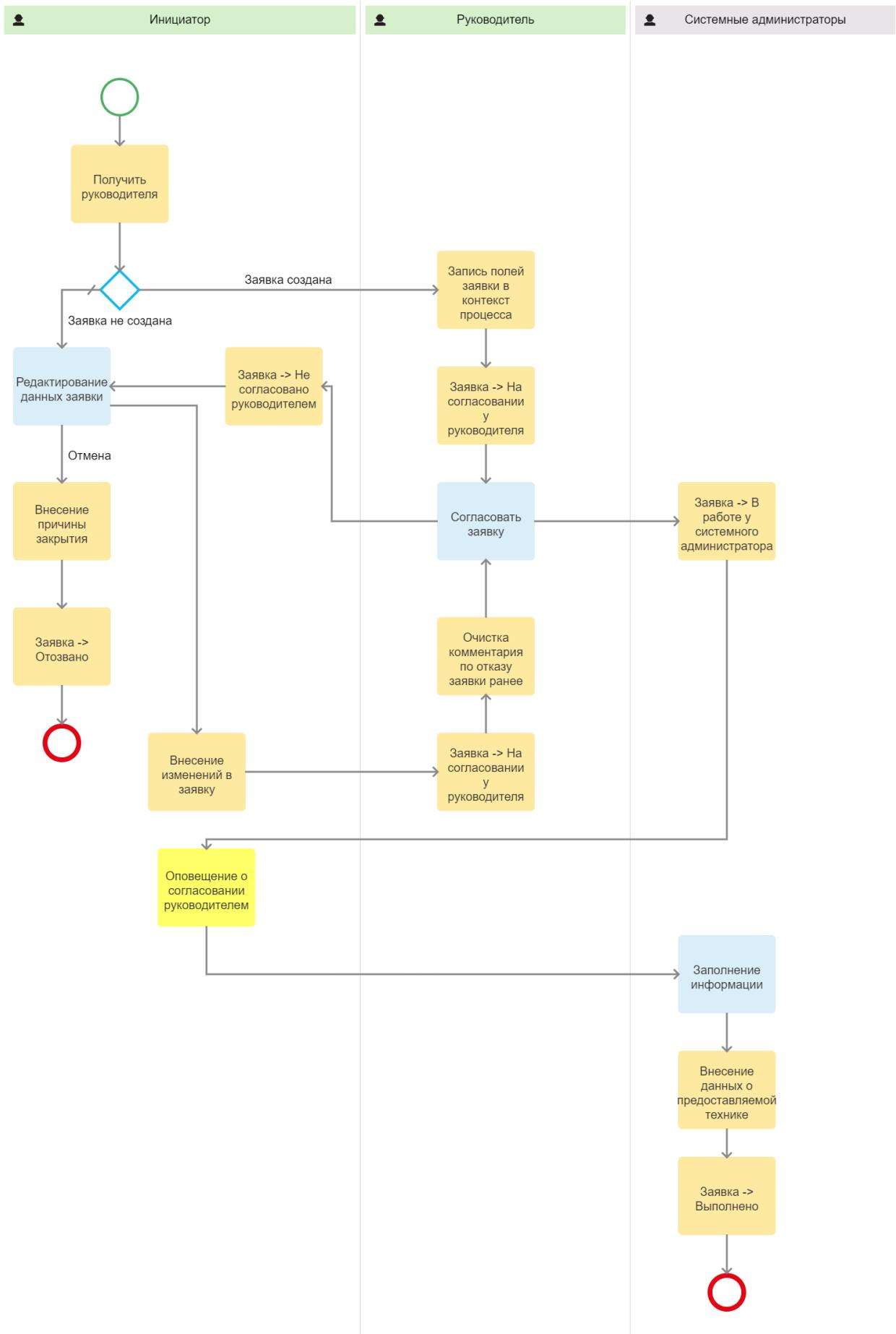


Рисунок 32 – бизнес-процесс работы заявки на получение новой техники

В отличии от предыдущего бизнес-процесса, в данном процессе согласования заявки есть отображены другие возможности системы «без программирования». На рисунке 33, можно увидеть настройку группы пользователей на эффективную работу, которые получат заявку для выполнения – группа системных администраторов, в данную группу входят все сотрудники с ролью системный администратор и техник (отдела информационного обеспечения). Таким образом задачи будут назначаться на группу, и любой пользователь группы сможет взять их на выполнение.

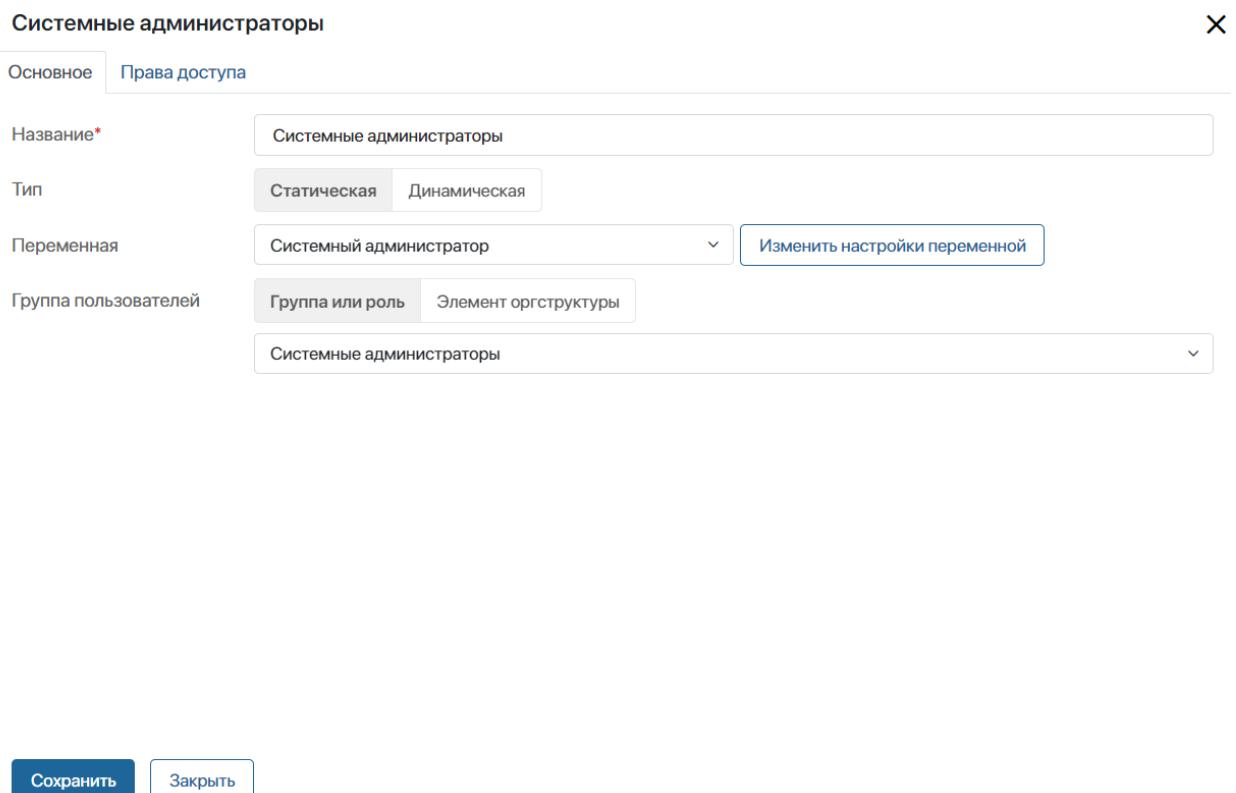


Рисунок 33 – настройка уведомления пользователей пулом «Системный администратор»

Так же в данном процессе настроено такое значение как статус заявки. Инициатор, в любой момент времени видит в каком статусе заявка и понимает, что процесс завершился или остановился на том или ином сотруднике. На рисунке 34 можно увидеть список всех статусов, которые были обозначены в данном процессе. На данном примере можно понять, что при использовании платформы «без программирования» для создания приложений на основе бизнес-процессов компании, что можно изменять свою статусную модель и

иметь возможность быстро ее изменять, ввиду изменения бизнес потребностей процесса. При необходимости, мы можем добавить еще один пул дорожки с дополнительным согласующим и быстро внести изменения в текущий бизнес процесс.

Статусы

Добавить поле "Статус"

Сохранять историю изменения

Статус можно менять вручную

Статус	Код	Финальный	Скрывать
= На согласовании у руководителя	na-soglasovaniu-rukoveditelya	<input type="checkbox"/>	<input type="checkbox"/>
= В работе у системного администратора	v-rabote-u-sistemnogo-administratora	<input type="checkbox"/>	<input type="checkbox"/>
= Выполнено	vypolnenno	<input checked="" type="checkbox"/>	<input type="checkbox"/>
= Отозвано	otklyoneno	<input checked="" type="checkbox"/>	<input type="checkbox"/>
= Не согласовано руководителем	ne-soglasovano-rukoveditelem	<input type="checkbox"/>	<input type="checkbox"/>

Добавить

Сохранить

Рисунок 34 – раздел составления статусной модели

Далее рассмотрим еще один бизнес-процесс по заявке, но данную заявку может подать только руководитель отдела. На рисунке 37 изображен бизнес-процесс заявки на подбор персонала. Первоначально мы имеем такой же обычный процесс, что рассматривали ранее, где есть несколько разных ролей – различные структуры компании и некая статусная модель заявки. Но здесь есть особые настройки, которые мы заложили в процесс. К примеру роли, если раньше мы рассматривали, что задачи, которые могут выполняться теми или иными отделами или же доступны любым сотрудникам. В бизнес-процессе заявки на подбор персонала все иначе – у нас присутствует три роли, которые доступны сотрудникам компании с определенными ролями:

1. Руководитель отдела – на рисунке 35 представлено перечисление пользователей с определёнными ролями, которым доступна данная заявка для создания. В основном представлены директора департаментов и начальники отделов.

Руководители отделов	
Описание	Сотрудники, которые могут создавать заявки в разделе "Вакансии"
Тип	Группа
Список участников группы	, Генеральный директор, Директор юридической дирекции, Директор дирекции по работе с персоналом, Начальник отдела "Департамент продаж", Директор дирекции информационных технологий, Начальник отдела "Департамент финансового", Начальник отдела "Управление безопасностью", Начальник отдела "Отдел компенсации и льгот", Начальник отдела "Управление внутреннего", Начальник отдела "Отдел подбора и адаптации", Начальник отдела "Отдел кадров", Начальник отдела "Инф. продуктов", Начальник отдела "Отдел разработки", Начальник отдела "Отдел дизайна", Начальник отдела "Отдел тестирования", Начальник отдела аналитики, Начальник отдела "Отдел технического обеспечения"

Рисунок 35 – перечисление участников группы, которые могут подавать заявки на подбор персонала

2. Согласующий вакансии — аналогично руководителям отдела представлено перечислений пользователей с определенными ролями — директор дирекции по работе с персоналом и руководитель отдела по подбору и адаптации.
3. Менеджер по персоналу — на рисунке 36 представлен элемент структуры компании, то есть те сотрудники, которые уполномочены на согласования поступившей заявки. Заявка будет доступна всем сотрудникам в отделе подбора и адаптации.

Менеджер по персоналу

Основное	Права доступа
Название*	Менеджер по персоналу
Тип	Статическая Динамическая
Переменная	Выберите переменную Создать новую переменную
Группа пользователей	Группа или роль Элемент оргструктур
<input checked="" type="checkbox"/> Отдел подбора и адаптации <ul style="list-style-type: none"> ▼ <input checked="" type="checkbox"/> Генеральный директор <ul style="list-style-type: none"> ► <input checked="" type="checkbox"/> Департамент Информационных технологий ▼ <input checked="" type="checkbox"/> Дирекция по работе с персоналом <ul style="list-style-type: none"> ► <input checked="" type="checkbox"/> Отдел подбора и адаптации ► <input checked="" type="checkbox"/> Отдел кадров ► <input checked="" type="checkbox"/> Отдел компенсации и льгот 	
<input type="button" value="Сохранить"/> <input type="button" value="Закрыть"/>	

Рисунок 36 – предоставление доступа к работе с заявкой на подбор персонала определенной структуре организации

Вторая особенность данного бизнес процесса, что тут присутствует подпроцесс, который работает независимо от данного процесса — «Взаимодействие с кандидатом». Процесс является независимым от процесса

подбора кандидата на вакансию, который будет рассмотрен в разделе построения бизнес-процессов отдела по работе с персоналом, так как данный функционал относится непосредственно к работе сотрудников по работе с персоналом.

В целом бизнес-процесс по подбору персонала похож на обычную заявку, для инициатора. Но для сотрудника отдела кадров это большая работа и новый бизнес процесс, который рассмотрим чуть позже.

Данный бизнес-процесс является одним из ярких примеров, как сотрудники внутри компании взаимодействуют с отделом кадров, можно сказать что одна из точек входа нового сотрудника в историю компании является его внесение карточки кандидата на вакансию.

Построение бизнес-процесса довольно похоже на предыдущие рассмотренные процессы, но если раньше мы создавали заявку и не могли ее редактировать до тех пор, пока заявку не отклонить следующий согласующий, то здесь построение процесса идет иначе. Бизнес-процесс (рис. 37) начинается с вопроса «Создана ли заявка?». Может возникнуть вопрос, а как же может быть иначе? Все очень просто, мы рассматриваем процесс работы с созданной заявкой, т.е. мы можем ее посмотреть и работать с ней, если она уже создана, но если ее еще не создали мы создаем новый экземпляр заявки (см. рисунок 37, блок «создание новой заявки»), с новыми данными (как руководитель). В целом, остальная часть не отличается от рассмотренных ранее бизнес-процессов, за исключением уже описанных нововведений.

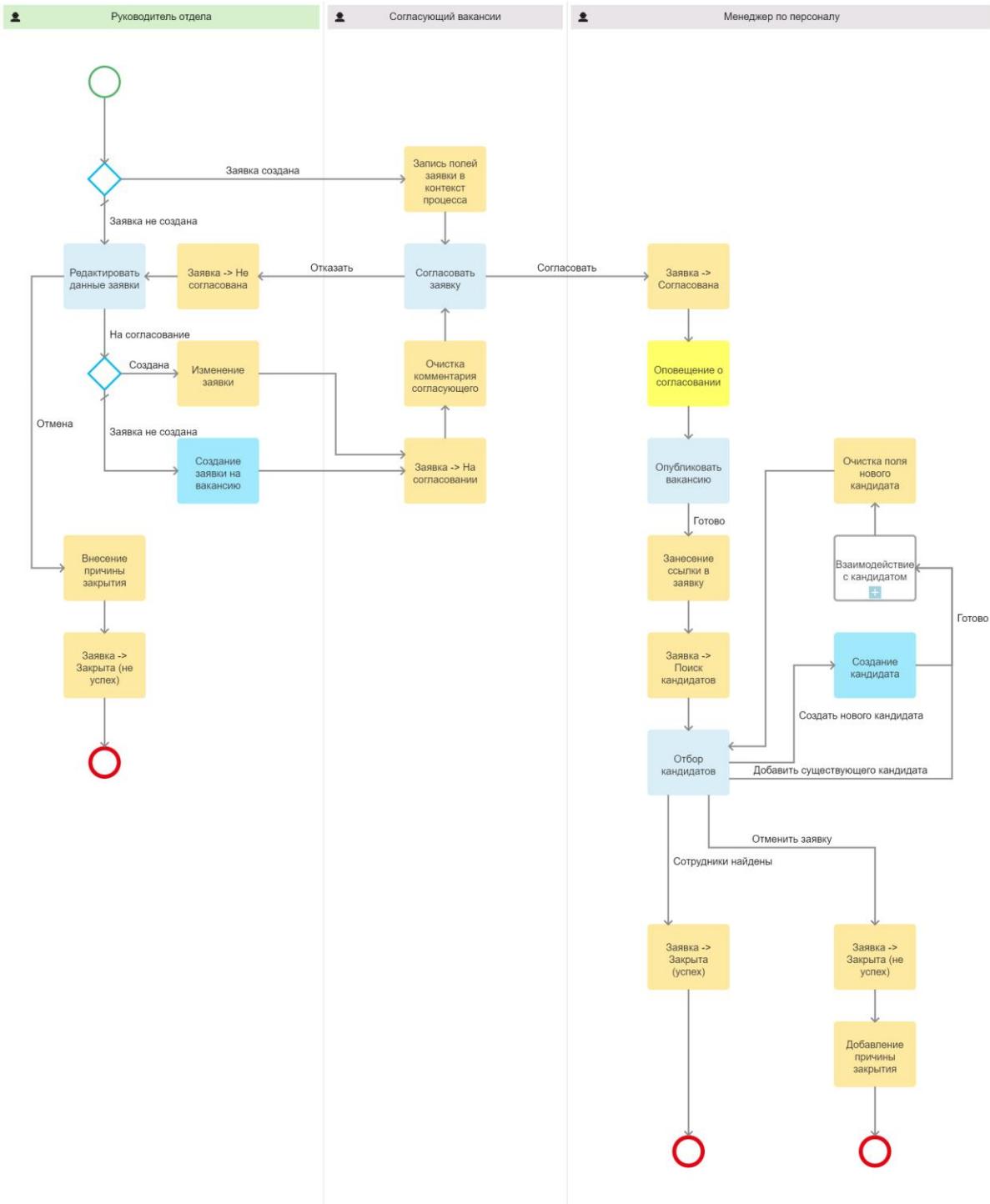


Рисунок 37 – бизнес-процесс работы с поданной заявкой на подбор

персонала

На рисунке 38 можно изображены контекстные переменные, которые используются при работе в бизнес процессе, часть из них используется при оформлении заявки и, на рисунке 39, мы можем рассмотреть, как выглядит форма заявки на поиск нового сотрудника (рис.40). Так же в процессе присутствует статусная модель. Принцип построения статусной модели не

отличается от рассмотренных ранее вариантов, но увидеть статусы можно на рисунке 41.

Заявка на подбор персонала		Схема	Контекст	Статусы
Сохранить Проверить Опубликовать Отладить				
Отображаемое имя	Имя свойства	Тип	Поиск и сортировка	
Участники дополнительного собеседования	additionalInterviewers	Пользователи	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Заявка	employee_search	Приложение	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Скрывать тестовое задание	f_exerciseHide	Выбор «да/нет»	<input type="checkbox"/>	<input type="checkbox"/>
Кандидаты на вакансии	candidatesTable	Таблица	<input type="checkbox"/>	<input type="checkbox"/>
Необязательные требования	unnecessaryRequirements	Строка	<input type="checkbox"/>	<input type="checkbox"/>
Новый кандидат	newCandidate	Приложение	<input type="checkbox"/>	<input type="checkbox"/>
Требуется согласование кандидатов	f_candidateApprovalNeeded	Выбор «да/нет»	<input type="checkbox"/>	<input type="checkbox"/>
Комментарий к вакансии	initiatorComment	Строка	<input type="checkbox"/>	<input type="checkbox"/>
Наличие тестового задания	f_testExerciseNeeded	Выбор «да/нет»	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Скрывать дополнительных собеседующих	f_addInterviewersHide	Выбор «да/нет»	<input type="checkbox"/>	<input type="checkbox"/>
Название	name	Строка	<input type="checkbox"/>	<input type="checkbox"/>
Комментарий согласующего	approverComment	Строка	<input type="checkbox"/>	<input type="checkbox"/>
Желаемый срок выхода	desiredDate	Дата/время	<input type="checkbox"/>	<input type="checkbox"/>
Обязанности	obligations	Строка	<input type="checkbox"/>	<input type="checkbox"/>
Требуется дополнительное собеседование	f_additionalInterviewNeeded	Выбор «да/нет»	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Требования	requirements	Строка	<input type="checkbox"/>	<input type="checkbox"/>
Максимум ЗП	paymentMax	Деньги	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Файл тестового задания	exerciseFile	Файлы	<input type="checkbox"/>	<input type="checkbox"/>
Минимум ЗП	payment	Деньги	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Ссылка на вакансию	vacancyLink	Ссылка	<input type="checkbox"/>	<input type="checkbox"/>
Причина отмены вакансии	cancelReason	Строка	<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 38 – контекстные переменные БП «Заявка на подбор персонала»

Название*

Обязанности*

Требования*

Необходимость согласования кандидатов с автором заявки на вакансии до собеседований

Требуется тестовое задание*

Файл тестового задания*

Наличие тестового задания для кандидатов

+ Файл

В качестве тестового задания можно выбрать один или несколько текстовых или графических файлов

Требуется дополнительное собеседование

Начните вводить текст для поиска элемента

Одни или несколько пользователей, участвующих в дополнительном собеседовании

Максимум ЗП*

Минимальное значение размера заработной платы

Минимум ЗП*

Максимальное значение размера заработной платы

Желаемый срок выхода

Комментарий

Участники дополнительного собеседования*

Дополнительные требования к навыкам кандидата

Примечание по вакансии. Сюда можно внести, например, количество искомых

Сохранить Отмена

Рисунок 39 – форма для создания вакансии на поиск нового сотрудника

Статусы

Добавить поле "Статус"

Сохранять историю изменения

Статус можно менять вручную

Статус	Код	Финальный	Скрывать
= На согласовании	approving	<input type="checkbox"/>	<input type="checkbox"/>
= Согласована	approved	<input type="checkbox"/>	<input type="checkbox"/>
= Не согласована	notApproved	<input type="checkbox"/>	<input type="checkbox"/>
= Поиск кандидатов	searching	<input type="checkbox"/>	<input type="checkbox"/>
= Закрыта (успех)	closedPositive	<input checked="" type="checkbox"/>	<input type="checkbox"/>
= Закрыта (не успех)	closedNegative	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Рисунок 41 – статусная модель заявки на подбор персонала

Далее представлены одни из возможных бизнес-процессов по согласованию служебной записке (рисунок 42) и три бизнес-процесса, связанного с командировками. Данные бизнес-процессы не будут подробно рассматриваться, а лишь приведены для примера, что их тоже возможно перевести в цифровой вид.

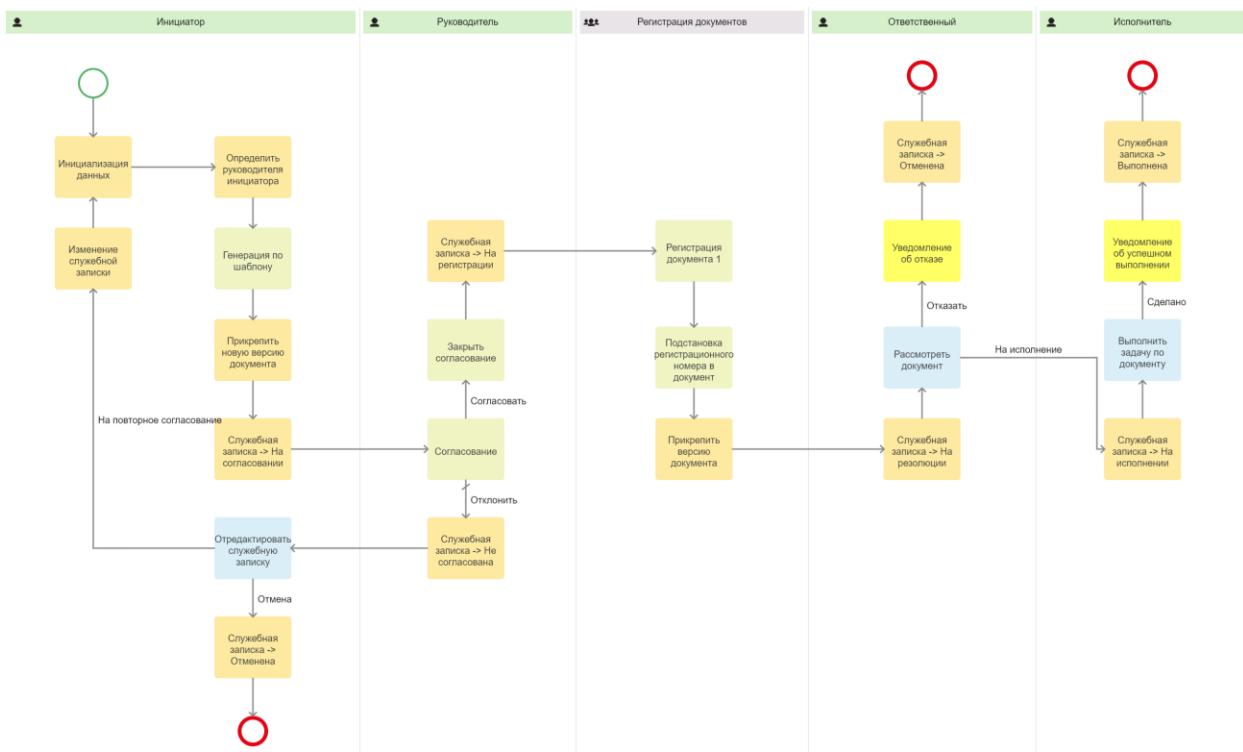


Рисунок 42 – пример бизнес-процесса работы с служебными заявками

Так же приведены примеры работы с согласованием командировок на рисунках 43-45 представлены процессы оформления командировок, возвращения денежных средств и продление командировок. Единственным нововведением является появление таймера, который дает определенный срок на выполнения задания в процессе.

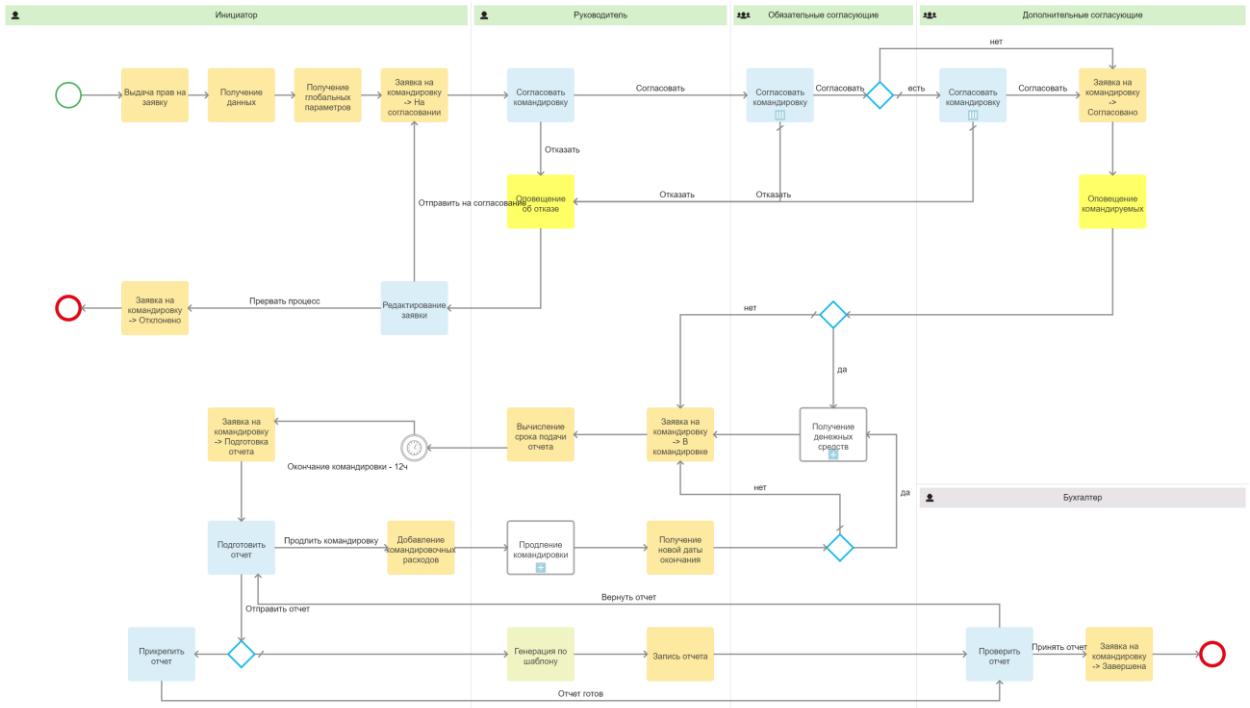


Рисунок 43 – бизнес-процесс оформления командировок

Бизнес-процесс оформления командировок является основным процессом, сопроводительные процессы по продлению и возврату денежных средств за командировку являются лишь подпроцессами большого процесса оформления командировок.

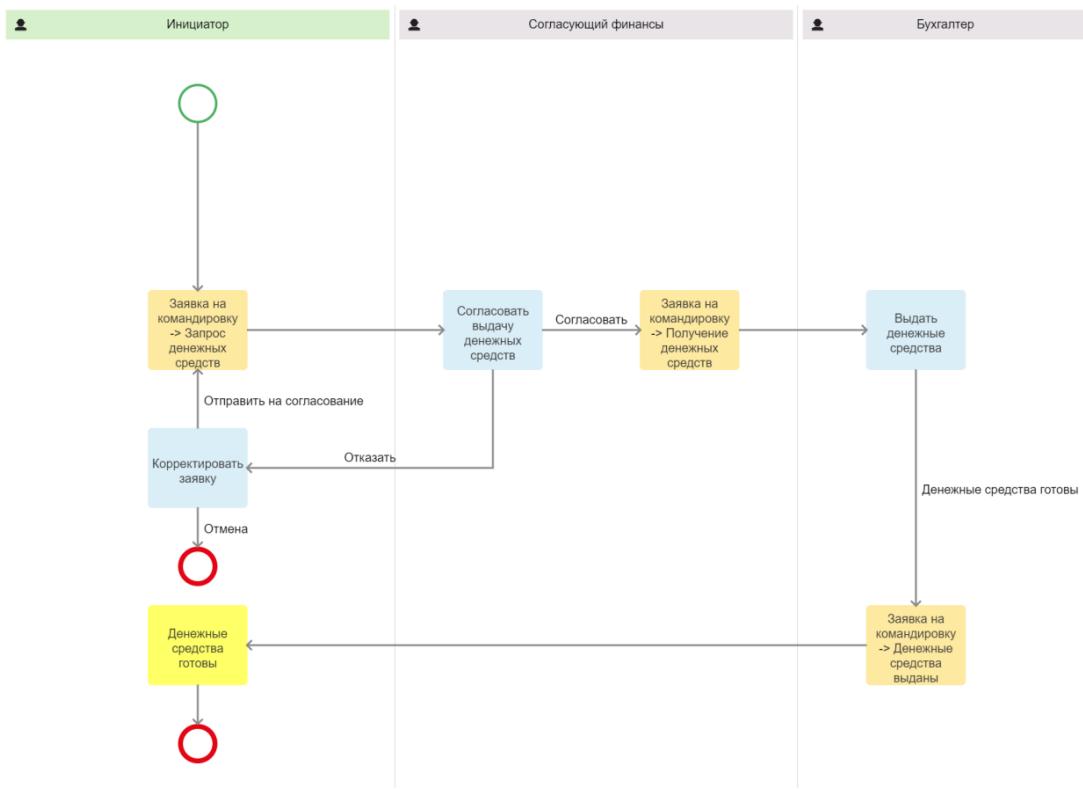


Рисунок 44 – бизнес-процесс получения денежных средств за командировку

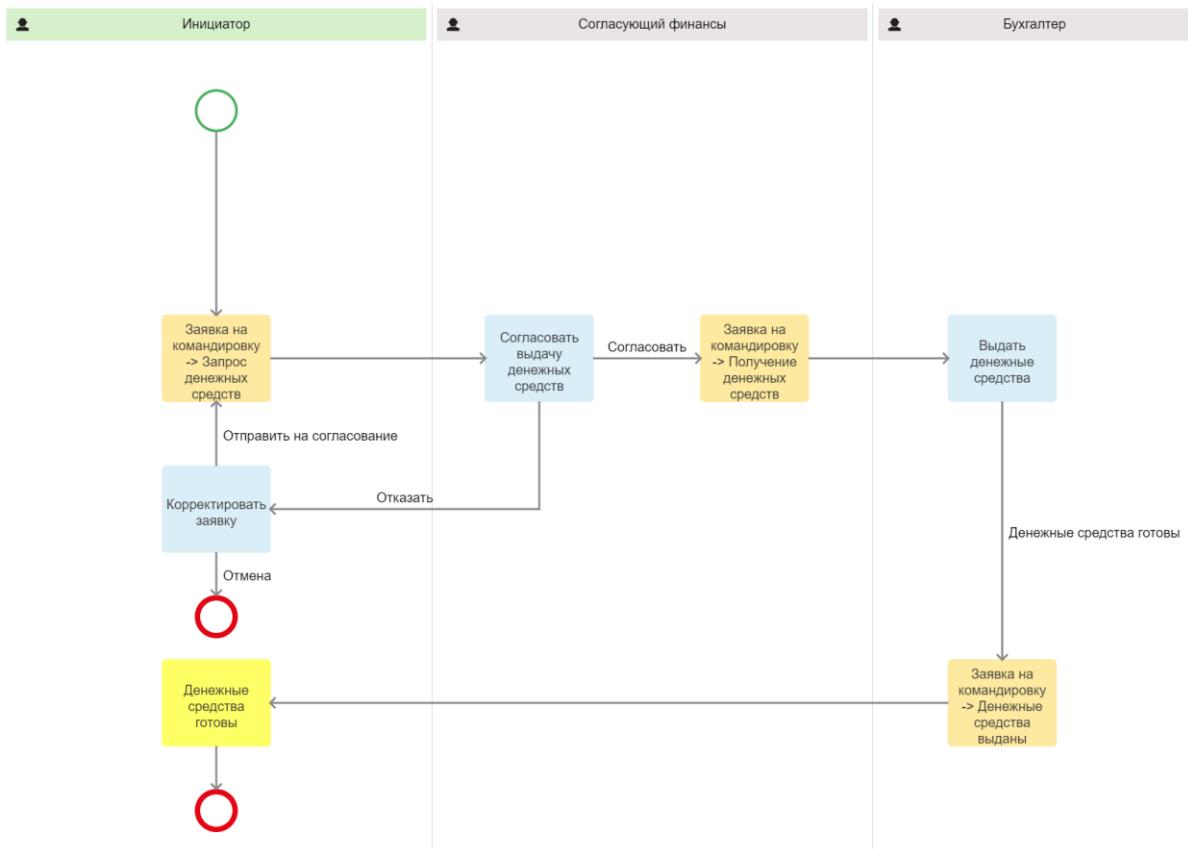


Рисунок 45 – бизнес-процесс продления командировки

3.2. Построение бизнес процессов отдела по работе с персоналом

Рассмотрим построение бизнес-процессов, с которыми работают сотрудники отдела по работе с персоналом. В отличии от ранее рассмотренных процессов, которые доступны для создания любому сотруднику компании, рассматриваемые процессы будут построены для одного отдела, но не каждый сотрудник отдела по работе с персоналом имеет полный доступ к всем процессам, а затрагивает их частично.

Начнем разбор работы с ранее знакомого процесса «взаимодействие с кандидатом», данный процесс ранее упоминался на рисунке 37, где указывалось, что это другой большой подпроцесс внутри процесса заведения заявки на поиск кандидата.

Рассмотрим подробно процесс работы взаимодействия с кандидатом со стороны любого сотрудника отдела подбора и адаптации персонала (можно рассмотреть структуру компании на рисунке 21). На рисунке 46 представлена полная схема взаимодействия менеджера по персоналу с кандидатом. Основной процесс взаимодействия как правило происходит за рамками системы, так как менеджер по подбору персонала (на схеме его действия отнесены к роли инициатор, так как именно он создает все шаги) созывается и договаривается о встрече с кандидатом и уже потом фиксирует, создание события в календаре. На рисунке 47 можно увидеть форму создания события в календаре.

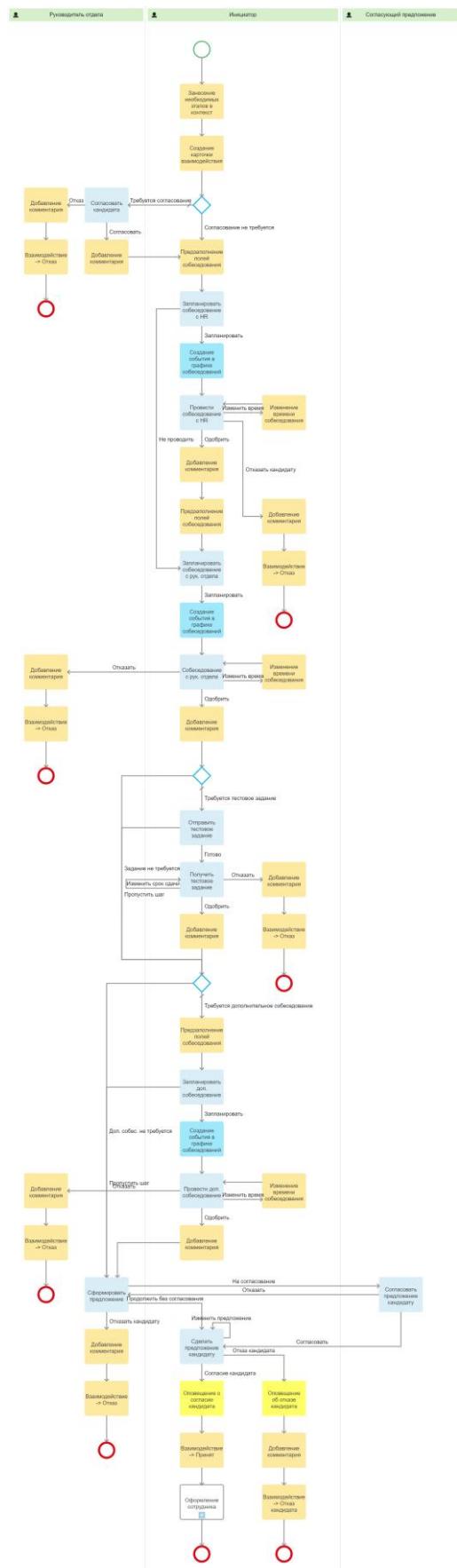


Рисунок 46 – бизнес-процесс взаимодействия менеджера по подбору персонала с кандидатом на должность

Процесс создания встречи в календаре аналогичен как для создания встречи только с менеджером по работе с персоналом, так и с руководителем. При необходимости шаги возможно пропустить.

Взаимодействие с кандидатом

Запланировать собеседование с HR: Иванов Иван Иванович (Тест)

Кандидат на должность	Иванов Иван Иванович	Название собеседования*	Собеседование с HR: Иванов Иван Иванович
Вакансия	Тестировщик	Начало собеседования*	дд.мм.гггг <input type="button" value="CALENDAR"/> h:мм <input type="button" value="TIME"/>
		Конец собеседования*	дд.мм.гггг <input type="button" value="CALENDAR"/> h:мм <input type="button" value="TIME"/>
		Участники собеседования*	Начните вводить текст для поиска элемента: <input type="text"/> <input type="button" value="SEARCH"/> РА Рита Аликова <input type="button" value="X"/>

Запланировать **Не проводить**

Рисунок 47 – процесс внесения собеседования с кандидатом в календарь

После того как руководитель проведет собеседование он может оставить комментарий кандидату и согласовать/отклонить. Если кандидат устроит всех сотрудников, которые производят найм, его карточку отправляют на оформление в штат. Если же кандидат не одобрен процесс работы с ним заканчивается.

На рисунке 48 можно увидеть, как внутри компании делают предложение в пользу выбранного кандидата. У согласующего пользователя есть возможность отказать кандидату, согласовать его или отправить далее по процессу без согласования. Поле комментария является обязательным, это реализовано для того что бы все участвующие в процессе согласования понимали, по каким причинам кандидат принят на должность или ему отказано по той или иной причине.

Взаимодействие с кандидатом

Сформировать предложение: Иванов Иван Иванович (Тестирующий)

Кандидат на должность	Иванов Иван Иванович
Вакансия	Тестирующий
Предложение*	<input type="text"/>

На согласование Продолжить без согласования Отказать кандидату

Рисунок 48 – процесс передачи согласования выбранного кандидата

Далее рассмотрим процесс оформления сотрудника, если он согласился на данную должность. Первоначально менеджеру необходимо заполнить карточку сотрудника (на рисунке 49 менеджер изображен как инициатор), которая должна быть согласована с руководителем отдела или департамента. В свою очередь, после согласования задачи с руководителем система сама, по заданному шаблону, создаст задачи на группы пользователей, которые отвечают за выдачу техники, наличие всех документов и подготовки рабочего места, все задачи назначаются параллельно. Это позволяет работать каждой группе сотрудников с задачей сразу после ее создания, а не дожидаться выполнения этой задачи сотрудниками другого отдела. Так же после выполнения всех подзадач, система дожидается выполнения всех подзадач, начинается режим отсчета времени до выхода сотрудника. В конце пользователю инициатору назначается задача, в день выхода сотрудника, о его встрече в офисе. При необходимости, сотрудник инициатор может создать подзадачу, связанную с данной задачей, на другого сотрудника, если его по каким-либо причинам не будет на рабочем месте в день выхода нового сотрудника (рис. 50).

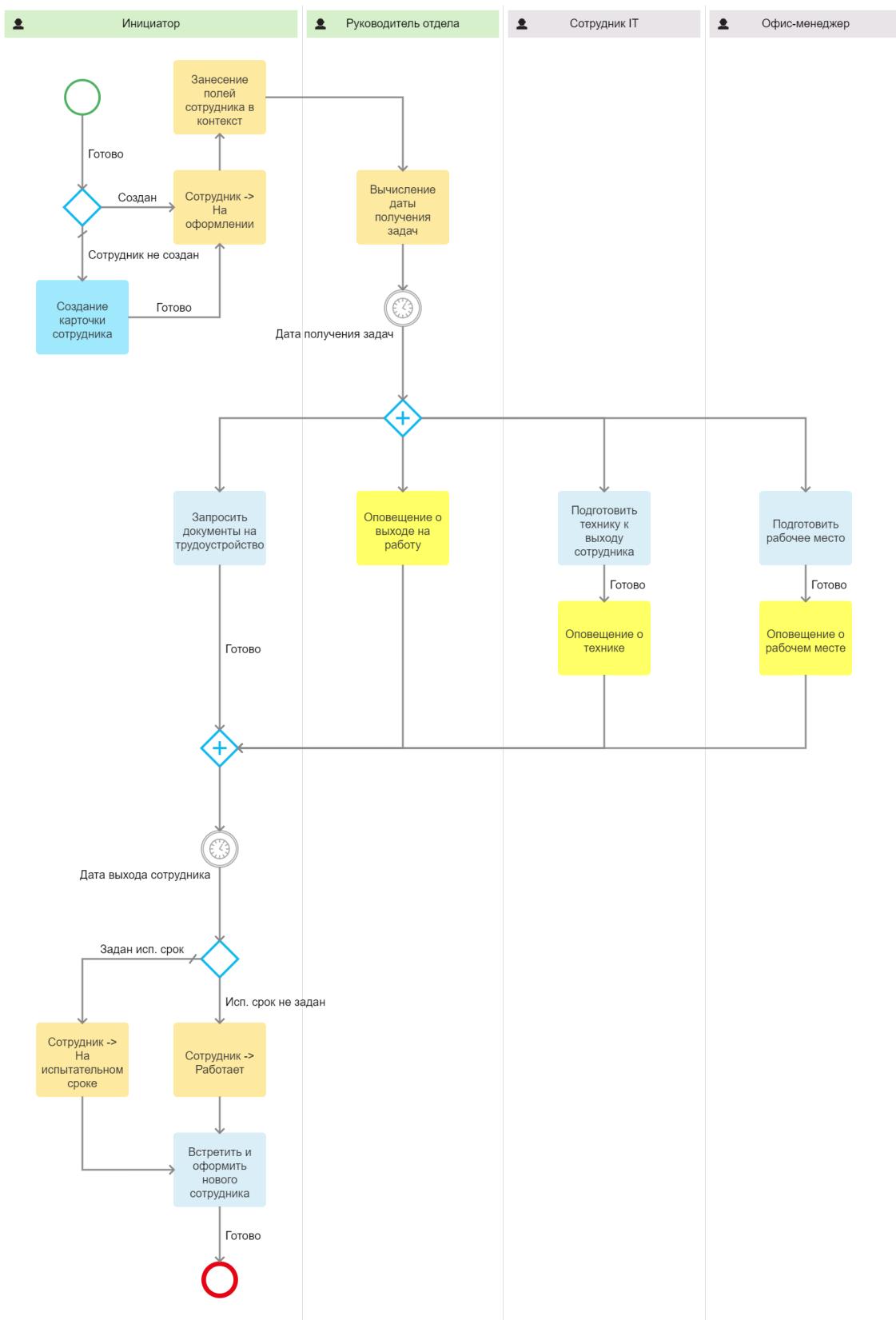


Рисунок 49 – бизнес процесс оформления сотрудника

Встретить и оформить нового сотрудника: Иванов Иван Иванович, Тестировщик**Сотрудник**

Иванов Иван Иванович (Тестировщик)

ФИО Иванов Иван Иванович

Должность Тестировщик

Дата выхода 29 апреля 2022 г.

Телефон +79182736484 (осн.)

Email t@mail.ru (осн.)

Готово

Рисунок 50 – задача, созданная системой на день выхода нового сотрудника
для менеджера по персоналу

В современных реалиях, после выхода на работу, у сотрудника начинается испытательный срок. Испытательный срок может быть на определенное время или же может быть завершен непосредственно указом руководства. В некоторых компаниях могут изменить заработную плату после испытательного срока, что тоже необходимо предусмотреть в нашем бизнес процессе. На рисунке 51 представлен бизнес-процесс перевода сотрудника с испытательного срока в постоянный штат. Процесс построен таким образом, что при открытии карточки сотрудника инициатор (в нашем случае это менеджер по работе с персоналом) подтверждает успешное прохождение испытательного срока или же может оформить увольнение, при необходимости (даный процесс будет рассмотрен чуть позднее). На рисунке 52 можно увидеть, как выглядит карточка сотрудника

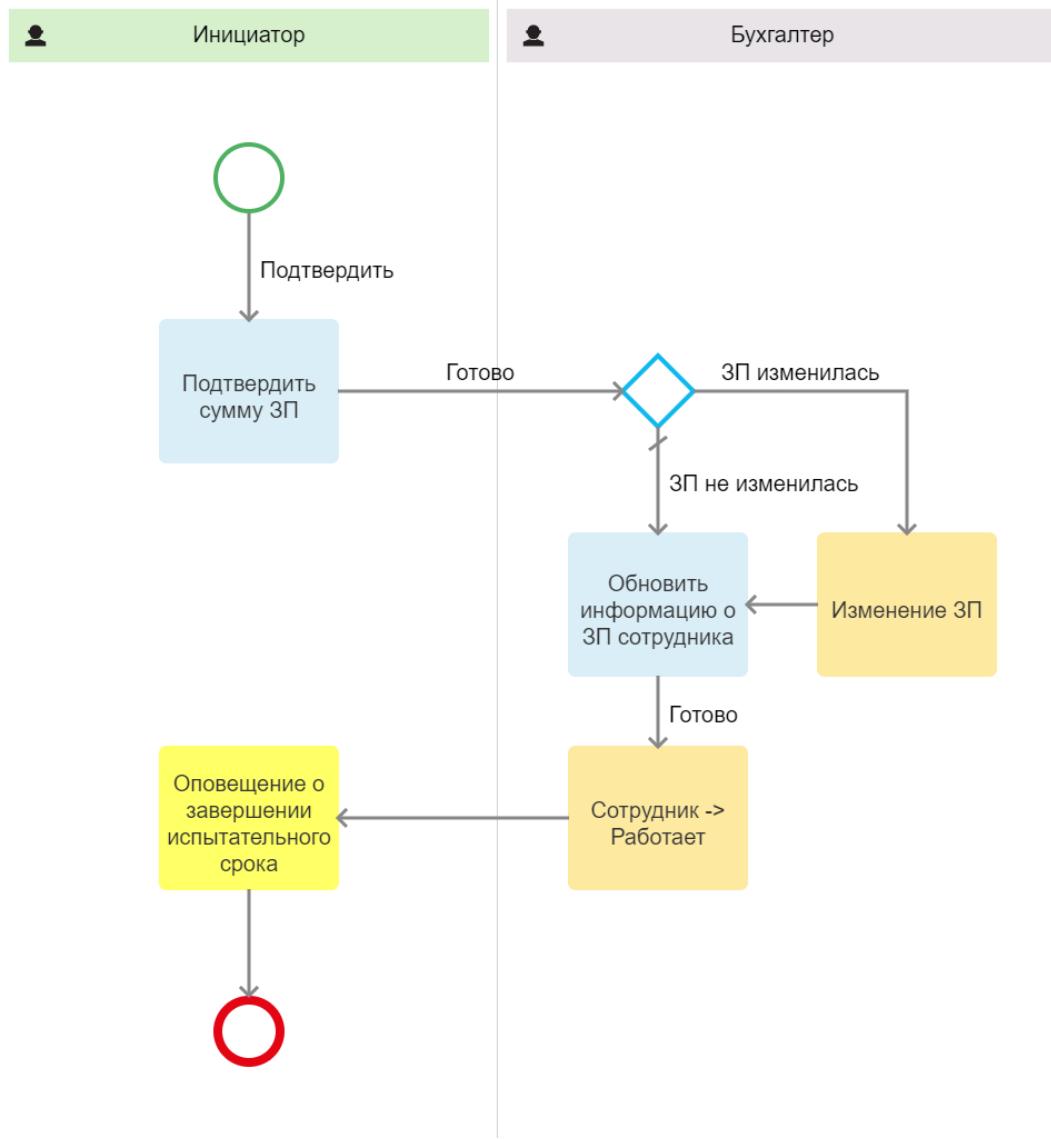


Рисунок 51 – бизнес-процесс перевода сотрудника с испытательного срока

если сотрудник находится на испытательном сроке. Как видно на рисунке, мы можем подтвердить перевод сотрудника в штат как постоянную единицу. При необходимости, можем изменить данные о заработной плате по кнопке редактирования. Так как задача уйдет на отдел по работе с персоналом в отдел начисления вознаграждения, сотруднику-бухгалтеру в любом случае необходимо будет проверить и подтвердить или опровергнуть изменение в заработной плате. По окончанию согласования всех изменений инициатору придет уведомление о том, что испытательный срок у данного сотрудника завершен.

Вычужанин Петр Васильевич (Специалист по продажам)

Карточка кандидата	Вычужанин Петр Васильевич
ФИО	Дата выхода
Вычужанин Петр Васильевич	21 апреля 2022 г.
Должность	Испытательный срок
Специалист по продажам	3 мес.
Руководитель отдела	Конец испытательного срока
 Рита Аликова	21 июля 2022 г.
Дата рождения	Зарплата на испытательном сроке
29 марта 1989 г.	9,00Р
Фото	Зарплата
	9,00Р
Комментарий	
Телефон	
+9173 907 477 455 (осн.)	
Email	
bgarvin8@solundcloud.com (осн.)	

Завершить испытательный срок

Подтвердить **Отмена**

Редактировать **Завершить испытательный срок** **Уволить** 

Рисунок 52 – форма перевода сотрудника в штат на постоянную основу

В каждой компании приходит момент, когда сотрудник хочет покинуть текущее место работы. Для этого тоже необходимо предусмотреть бизнес-процесс, так как необходимо подготовить документы, внести сведения, рассчитать его и забрать технику, для этого нужно провести взаимодействие менеджеру по персоналу с другими коллегами из подразделения информационных технологий (организовавших выдачу техники) и коллег из департамента работающих с вознаграждением. Первоначально, после того как менеджеру по персоналу попадёт информация об увольнении, он должен

отметить сколько дней ему еще необходимо отработать (рисунок 54), затем, в последний рабочий день

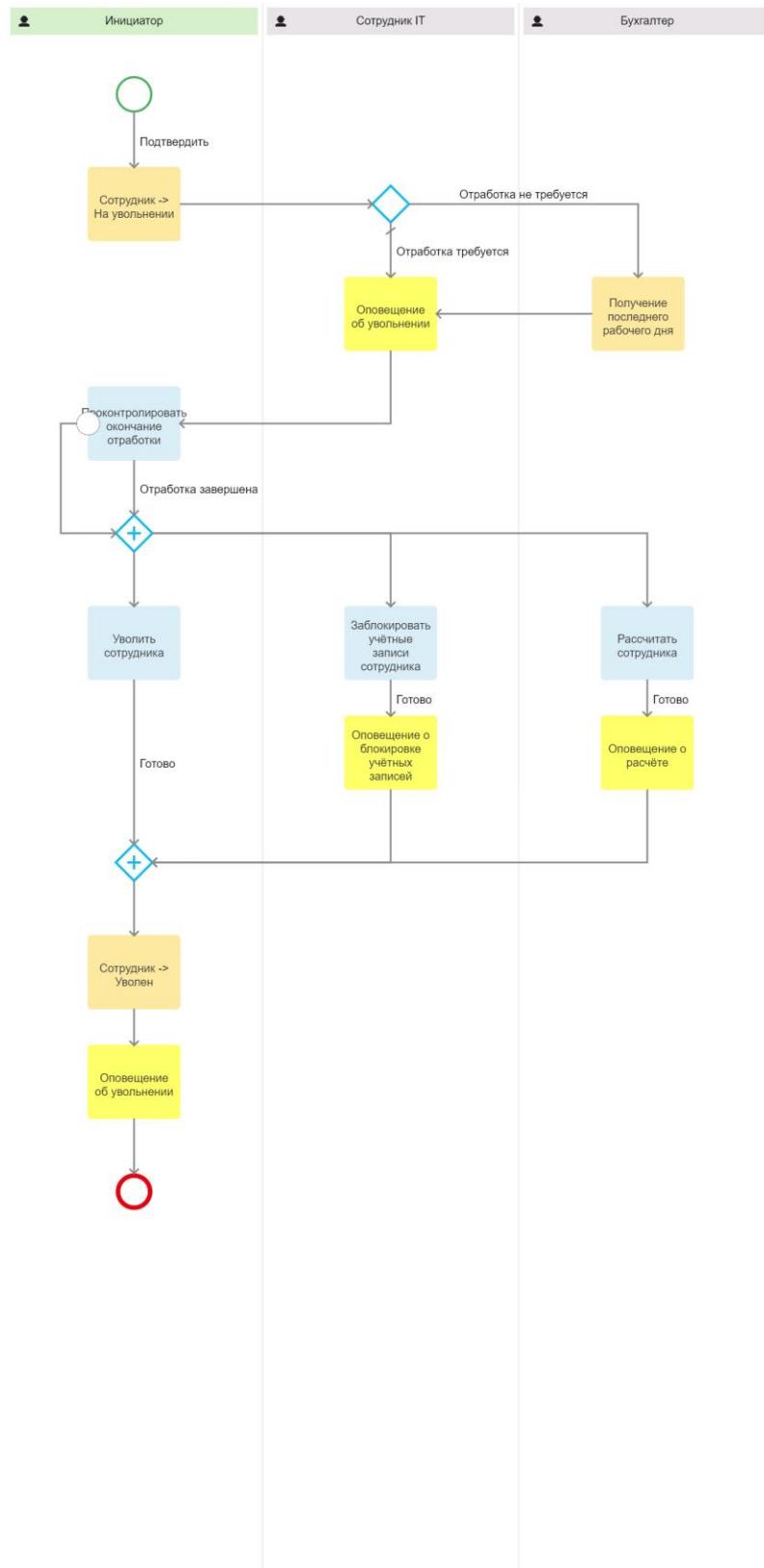


Рисунок 53 – бизнес-процесс увольнения сотрудника

увольняемого сотрудника, система автоматически создает задачи на ответственные отделы, для их параллельного (задачи создаются всем отделам сразу и считаются выполненными если все задачи закрыты) выполнения. После выполнения всех задач инициатору приходит уведомление об увольнении сотрудника и закрытии задачи, сама же задача переходит в статус сотрудник уволен.

Уволить

Увольняющийся

Вычужанин Петр Васильевич (Специалист по продажам)

ФИО

Вычужанин Петр Васильевич

Должность

Специалист по продажам

Дата увольнения*

30.04.2022



Дата приёма заявления об увольнении; со следующего дня начинается отработка

Руководитель увольняющегося*

Рита Аликова



Требуется отработка

Да

Нет

Дата окончания отработки*

дд.мм.гггг



Подтвердить

Отмена

Редактировать

Уволить



Рисунок 54 – форма с карточки сотрудника при увольнении

3.3. Прогнозирование трендов в развитии средств автоматизации бизнес-процессов «без программирования»

С помощью платформы разработки «без программирования» можно создавать полностью работающие приложения, где не требуется дополнительного программирования с нуля, а лишь доработать логику дополнительным кодом.

Платформа «без программирования» предоставляет среду разработки, используемую для создания прикладного программного обеспечения через графический интерфейс пользователя вместо традиционного компьютерного программирования. Данные платформы могут создавать полностью рабочие приложения и не требует обширных знаний в программировании. Платформы разработки с «без программирования» сокращают объем традиционного ручного кодирования, что позволяет ускорить внедрение минимально работающего продукта. Явным преимуществом является то, что более широкий круг людей может участвовать в разработке приложений, а не только те, у кого есть высокие навыки программирования. Платформы разработки «без программирования» также могут снизить первоначальные затраты на настройку, обучение персонала, развертывание и обслуживание.

Платформы разработки «без программирования» основаны на принципах проектирования на основе бизнес-моделей, автоматической генерации кода и визуального программирования. Концепция разработки для конечного пользователя также существовала ранее, хотя платформы «без программирования» привнесли некоторые новые подходы к этой разработке. Рынок платформ разработки с низким кодом начинает свое зарождение с 2011 года.

Сегодняшние тенденции заключаются в следующем:

- Более упрощенное создание бизнес процессов – по аналогии с обычным BPMN, т.е. реализация понятных интерфейсов;

- Процессы смогут строить любые сотрудники компании (сейчас же в основном этим занимается ИТ-отдел, при необходимости, программист дописывает скрипты);
- Создание любых видов приложений, сейчас же по большей части это документооборот, CRM, создание лендингов (небольших сайтов страниц);
- Внедрение искусственного интеллекта и машинного обучения в платформы - для потенциальной помощи в создании приложений, проведении тестирования и улучшении их качества;
- Роботизация бизнес-процессов (RPA) также относится к тенденциям развития. В данном случае простые роботы берут на себя множество повторяющихся операций, а платформа «без программирования» при этом выстраивают логику процессов и управляют взаимодействием;
- Process Mining, которая представляет собой систему методов, предназначенных для постоянного улучшения внутренних и внешних бизнес-процессов предприятия. Технология позволяет проводить анализ процессов на разных уровнях детализации.

Данные тенденции, которые, с одной стороны, подчеркивают необходимость для компаний упрощать процессы разработки, а с другой — важную эволюцию, происходящую в мире разработки платформ «без программирования», способных ускорить время выхода проектов на рынок. Парадигма, согласно которой разработка программного обеспечения является прерогативой высококвалифицированных инженеров, устаревает, и даже менее опытные сотрудники могут с помощью правильных инструментов создавать мощные приложения на основе использования предопределенных шаблонов и интуитивно понятных интерфейсов.

Это первое из преимуществ разработки с помощью платформ «без программирования». Данная платформа, по сути, позволяет генерировать прикладное ПО через конфигурационные модули и графические интерфейсы, вместо того, чтобы использовать исходный код. Таким образом, можно создавать базы данных, интерфейсы, веб и мобильные приложения, а также бизнес-процессы. Это уменьшает объем традиционного ручного написания кода и ускоряет разработку приложений.

Конечно, одним из преимуществ является то, что люди, даже не обладающие особыми навыками высокого уровня программирования, могут участвовать в разработке приложения. Это важные преимущества, особенно если рассматривать тот вариант, когда на рынке мало высококвалифицированных программистов. Таким образом, платформа простого и интуитивно понятной разработки, столь востребована компаниями.

Конечно, разработку «без программирования» нельзя рассматривать как панацею, способную исправить все неэффективности, например, она по-прежнему требует анализа изначально неправильных рабочих процессов, она требует минимального управления проектами, чтобы предотвратить эту проблему. Так как могут реализовываться избыточные или вредные решения, но правильная организация построения бизнес-процессов помогут избежать этих неприятных ситуаций. Несмотря на то, что сейчас идет бум развития платформ «без программирования» «традиционные» разработчики остаются ценным и незаменимым ресурсом.

Сейчас множество компаний, внедривших платформу «без программирования», просто воспользовались возможностью использовать профессиональных разработчиков для специализированных проектов, требующих высокой квалификации. Таким образом, можно представить сценарий, в котором профессиональные программисты решают сложные

задачи, а системные аналитики помогают командам удовлетворять все еще растущий спрос на новые приложения и рабочие процессы.

Платформа с низким кодом должна служить точкой интеграции всего предприятия, соединяя все системы, приложения, пользователей и данные для стимулирования роста бизнеса. Она также должна обеспечивать предоставляет такие преимущества как: надежность, безопасность, отказоустойчивость и соответствие требованиям.

Стабильность, выбранная в качестве основы для будущего роста, должна быть ключевой функцией для повышения пользовательского опыта в создании своих систем на основе бизнес-процессов. Нам нужно сосредоточиться на высокоавтоматизированной и масштабируемой платформе с низким кодом, чтобы быстро создавать продукты, ставя высокие цели: например, ускорить выход нового функционала без увеличения штата или обслуживать больше клиентов за меньшее время, благодаря быстрой автоматизации и перехода к цифровым процессам. Таким образом, это будет платформа, которая способствует сотрудничеству между командами, позволяя разрабатывать приложения на основе компонентов, созданных таким образом, что решения можно переиспользовать по-новому и в других отделах.

3.4. Выводы

В данной главе был создан прототип приложения по работе с персоналом на платформе «без программирования», в качестве примеров были приведены бизнес-процессы и публичный интерфейс приложения, доступный пользователям системы. При использовании платформы «без программирования» был отмечен вектор развития технологии разработки данных платформ. В разделе были рассмотрены тренды развития данной технологии, которые в будущем, помогут улучшить платформы «без программирования».

Глава 4. Отличие традиционного подхода к разработке приложений и разработке на платформе «без программирования»

По иронии судьбы отдел кадров переполнен избыточными и повторяющимися задачами, которые не оставляют отделу времени, для того чтобы сосредоточиться на людях в организации. По различным источникам⁴, HR-специалисты тратят до 40% своего времени на утомительные административные задачи. Платформа «без программирования» на основе которой будет создано HR-приложение может автоматизировать повторяющиеся административные задачи, тем самым повышая организационную точность и позволяя HR-менеджерам сосредоточиться на важной стратегической работе.

4.1. Пример создания проекта при традиционном подходе разработки

Ранее в работе были построены и реализованы бизнес-процессы, описаны варианты использования системы пользователями с различными ролями, а также представлена часть базы данных. При традиционном подходе после этапа анализа происходит этап разработки. Но прежде чем приступить к разработке необходимо выбрать язык программирования.

Для проектирования описанных требований был выбран фреймворк – Django, на основе языке программирования Python. Выбор языка python был связан с тем, что он является быстроразвивающимся языком, на нем можно быстро создать реляционную базу, а также написать api-функции, которые позволяют быстро получать и сохранять данные в базу данных. К тому же, python имеет множество различных библиотек, которые можно применить. Данный фреймворк был выбран за простоту использования и множество встроенных функций.

⁴ 40% рабочего времени HR занимает рутинна. Есть способы это исправить [Электронный ресурс] - Режим доступа:
<https://rb.ru/opinion/digitalization-of-hr/>

Первоначально, после выбора языка программирования нам необходимо развернуть среду для проекта и создать или подключить базу данных к реализуемой системе. После создания базы и схем, необходимо реализовать передачу и обработку данных, для того что бы мы могли хранить, и при необходимости изменять данные в базе. Для этого создали файл crud.py (ознакомиться с кодом можно в репозитории по [ссылке](#)) именно тут описаны многоразовые API-функции для взаимодействия с данными в базе данных. Итогом работы является создание нескольких API функций, которые можно рассмотреть на рисунке 55.

NIR_Employee Пoиск, получение, изменение, удаление и добавление данных по сотруднику	
GET	/employee/ Read Emp
PUT	/employee/ Emp List Post
POST	/employee/ Emp List Post
GET	/employee/{id} Read Emp By Id
NIR_Costs Пoиск, получение, изменение, удаление и добавление данных по счетам сотрудника	
GET	/costs/{id} Read Cost By Id
POST	/costs/ Create Costs
DELETE	/cost/{id} Delete Cost By Id
NIR_Cities Пoиск, получение, изменение, удаление и добавление данных по городам	
GET	/cities/ Read Cities
POST	/cities/ City List Post
GET	/cities/{id} Read City By Id
DELETE	/city/{id} Delete City By Id
NIR_Offices Пoиск, получение, изменение, удаление и добавление данных по офисам	
PUT	/office/ Put Office
POST	/office/ Create Office
GET	/offices/ Read Offices
DELETE	/office/{id} Delete Office By Id
NIR_Childs Пoиск, получение, изменение, удаление и добавление данных по детям сотрудников	
POST	/child/ Create Child
GET	/childs/ Get Childs
PUT	/childs/{id} Put Child

Рисунок 55 – модели API-функций созданные в процессе работы

При разработке данных API-функций было написано не мало кода, с которым можно ознакомиться в репозитории по [ссылке](#). На основе

развернутой среды было написан не один десяток строк кода, для того что бы мы могли обмениваться информацией с системой. Но для того что бы пользователь мог работать в привычном и понятном режиме, для него необходимо реализовать понятный интерфейс – пользовательскую часть приложения, которую можно увидеть на рисунке 56.

The screenshot shows a web-based application interface titled 'Рабочий Стол' (Workplace). On the left, there is a sidebar with various icons and links: 'Рабочий Стол' (selected), 'Последние' (Recent), 'Kanban Доска' (Kanban Board), 'Сотрудники' (Employees), 'Задачи' (Tasks), 'Календарь' (Calendar), 'Открытые Вакансии' (Open Vacancies), and 'База Резюме' (Resume Database). At the top right, there is a user profile for 'Калионова Ирина АДМИН' and a 'BDT' logo. The main content area is titled 'Сотрудники Компании' (Company Employees) and contains a table with the following data:

Фамилия Имя	Должность	Город	Возраст	Почта	Номер телефона
Александров Леонид	Старший разработчик	Москва	22	grnna@cuba.ru	891734514374
Веренцев Андрей	Старший разработчик	Москва	33	grnna@cuba.ru	891724564375
Громова Анна	Системный аналитик	Астрахань	24	grnna@cuba.ru	891734564371
Лобанова Любовь	Тестировщик	Самара	21	grnna@cuba.ru	891734564372
Михайлов Андрей	Начальник отдела	Москва	48	grnna@cuba.ru	891432147543

Рисунок 56 – панель списка сотрудников, отображаемая пользователю, работающему с системой

Текущий вариант разработки может «закрыть» все потребности заказчика и сделать функционал более адаптированным под нужды компании. Но при этом разработка будет занимать не мало времени, чем может озадачить заказчика.

4.2. Сравнительный анализ создания приложений на платформе «без программирования» и приложений созданных командой разработчиков

Ранее в работе были построены бизнес процессы и описаны различные способы разработки приложений. Но что же лучше использовать? Для этого проведем анализ и поймем, в каких случаях, в текущих реалиях, лучше использовать тот или иной вид разработки приложений.

Для того что бы написать любую платформу в объеме минимально работающего приложения для ввода в эксплуатацию компании необходимы ресурсы. В первую очередь необходимо понять, компания хочет получить? Быстрый работающий продукт, без долгой разработки для которого нет необходимости привлекать большое количество ИТ-специалистов или же компании необходимо индивидуальное решение, которое должно быть максимально проработано для компании с расчетом на будущие изменения и для нет необходимости экономить на ИТ-ресурсах?

Если компания выбирает вариант индивидуальной разработки, она нуждается в команде разработки, состоящей из аналитиков, тестировщиков, разработчиков. При разработке число участников команды может меняться, в зависимости от времени внедрения разрабатываемого продукта. При этом создаваемое решение будет иметь внешний вид приложения, который необходим заказчику, разрабатываемое решение будет иметь архитектуру наиболее подходящую под текущее состояние компании. Можно вводить в эксплуатацию любые пожелания заказчика, а не быть ограниченным платформой, при варианте выбора платформы «без программирования», тестировать готовое решение и указывать на необходимые доработки и ошибки, ожидать исправление и разбор ошибок. Так же при традиционном подходе создания приложений необходимо понимать, что бюджет может меняться, так как может изменяться количество работников над проектом, необходимо оплачивать сопроводительные расходы, связанные с индивидуальной разработкой, пример – оплата лицензий используемых продуктов при разработке, облачные хранилища и т.д.

При разработке приложения на платформе «без программирования» команда разработки будет значительно меньше, но возможность «гибкости» создаваемого приложения будет ограничена возможностями платформы. При этом заказчик всегда знает какая сумма будет затрачена на подписку на платформу, примерно сможет рассчитать затраты на персонал, который будет

обслуживать разрабатываемую систему. В то же время создавать новый функционал можно быстрее, чем при традиционном подходе, что позволит, в некоторых случаях, сразу вносить правки в логику бизнес-процесса при обнаружении ошибки.

4.3. Выводы

Сегодня существуют различные решения создания продуктов, можно выбрать любой подход, но при этом лицо, которое хочет получить разработку, должно четко понимать, каких целей он хочет достичь. Несмотря на то, что сегодня мир завоевывают платформы «без программирования» отказ от традиционной разработки в ближайшие годы не предвидится.

ЗАКЛЮЧЕНИЕ

В данной работе был произведен анализ бизнес-процессов работы отдела по работе с персоналом, рассмотрены существующие средства моделирования и произведен анализ решений на основе платформ «без программирования». Также данная работа содержит построение бизнес – процесса, который реализовывается в практической части, что позволяет на основе теоретической части рассказать все плюсы и минусы внедрения решений low-code в различные отделы компании, но и спроектировать процесс работы будущей системы, которая позволит автоматизировать весь HR отдел. Т.е. в данной работе представлен прототип системы, которая позволяет покрыть все основные нужды по работе с персоналом.

Можно сделать вывод, что, освобождая людей от повторяющейся и монотонной работы, технологии обещают предложить более привлекательную занятость и определенное конкурентное преимущество для организаций. Роботизация в HR приводит к уменьшению количества ошибок, точному анализу и отчетности, а также к оптимизации общих операций HR.

Мировая практика показывает, что большинство организаций признают роль платформ «без программирования» для снижения затрат, повышения эффективности и обеспечения бесперебойного выполнения. Эти компании готовы вкладывать свои деньги, чтобы обеспечить себе современную ИТ-составляющую, которая может работать в любых условиях, при быстро меняющемся мире.

Платформы «без программирования» не являются новой концепцией в разработке программного обеспечения, но в последнее время процветает и привлекает большое внимание в отрасли, которая может помочь как разработчикам, так и пользователям быстро получать рабочие приложения.

Как было рассмотрено в работе данные платформы успешно развиваются и позволяют создавать продукты различного типа и уровня. Конечно, как и все информационные продукты, платформы «без

программирования» не являются идеальными, но на текущий момент мы присутствуем при развитии данной технологии, возможно, через пару лет, такой процесс разработки будет повсеместный и более «продвинутый».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лекции MICSECS 2021 — Oleg Lapshin (Masterdata) Who needs low code platforms and why?
2. Лекции MICSECS 2021 — Oleg Lapshin, Александр Гриценко (Masterdata) Masterclass: Погружение в RPA-технологии
3. Building Low-Code Applications with Mendix: Discover best practices and expert techniques to simplify enterprise web development — Bryan Kenneweg, Imran Kasam, Micah McMullen, 2021
4. Hands-On Low-Code Application Development with Salesforce, Enrico Murru, 2020
5. Нотация BPMN Система управления бизнес-процессами ELMA: [Электронный ресурс]. URL: <https://www.elma-bpm.ru/bpmn2/>
6. Chris Partridge Business Objects: Re-Engineering for Re-Use / Chris Partridge, NY, — 2016
7. Замятина О.М. Метод моделирования и комплексного анализа бизнес-процессов / — Краснодар, Кубанский государственный аграрный университет имени И.Т. Трубилина, -2016 г.
8. tadviser [Электронный ресурс] — Режим доступа: <https://www.tadviser.ru/index.php/BPM>
9. MARKET.CNews [Электронный ресурс] — Режим доступа: https://market.cnews.ru/research/lowcode_2021/table?p=review
10. Gartner [Электронный ресурс] — Режим доступа: <https://www.gartner.com/en/newsroom/>
11. Топалович, Никола. Применение клиентского опыта при внедрении BPM-систем / Никола Топалович. — Текст : непосредственный // Молодой ученый. — 2020. — № 19 (309). — С. 144-145.
12. Мусатов И.С., Яхонтова И.М. Случайные процессы в моделировании бизнес-процессов / — Мусатов И.С., Яхонтова И.М, — Краснодар,

Кубанский государственный аграрный университет имени И.Т. Трубилина, -2016 г.

- 13.Нотация BPMN Система управления бизнес-процессами ELMA: [Электронный ресурс]. URL: <https://www.elma-bpm.ru/bpmn2/>
- 14.Панфилкин А.А. Основные средства разработки приложений; Издательство: ИП Шелистов Денис Александрович (издательский центр "quantum"), 2018. – 816 с.
- 15.Репин В. В. Бизнес-процессы. Моделирование, внедрение, управление / Репин В. В.; М.: МИФ, 2012. – 512 с.
- 16.Chris Partridge Business Objects: Re-Engineering for Re-Use / Chris Partridge, NY, — 2016
- 17.Raquel Sanchis, Óscar García-Perales, Francisco Fraile, and Raul Poler. 2020. Low-Code as Enabler of Digital Transformation in Manufacturing Industry. Applied Sciences 10, 1 (2020).
- 18.Meg Fryling. 2019. Low Code App Development. Journal of Computing Sciences in Colleges 34, 6 (2019).
- 19.Коваленко, А. А. Необходимость документирования и анализа бизнес-процессов для предприятия малого бизнеса / А. А. Коваленко. — Молодой ученый. — 2019. — № 1 (239). — С. 87-90.
- 20.Sergio de Cesare and Chris Partridge BORO as a Foundation to Enterprise Ontology / Sergio de Cesare — London, Brunel University London, — 2015
- 21.Долганова, О. И. Моделирование бизнес-процессов : учебник и практикум для академического бакалавриата / О. И. Долганова, Е. В. Виноградова, А. М. Лобанова ; под редакцией О. И. Долгановой. — Москва: Издательство Юрайт, 2019. — 289 с.
- 22.Deepak Kumar Sharma Hitesh Varun Rao Individualization of process model from configurable process model constructed in C-BPMN, London, 2017
- 23.Jurate Petrikina, Paul Drews, Ingrid Schirmer Integrating Business Models and Enterprise Achitecture / Jurate Petrikina, IEEE – 2015 г.

24. Robert Waszkowski. 2019. Low-code platform for automating business processes in manufacturing. IFAC-PapersOnLine 52, 10 (2019), 376–381.
25. Лоншакова, А. И. Обзор существующих инструментов для моделирования бизнес-процессов, использующих основные методологии / А. И. Лоншакова, В. В. Балицкий. — Молодой ученый. — 2020. — № 25 (315). — С. 112-114.