

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.04.03 Прикладная информатика
(код и наименование направления подготовки)

Информационные системы и технологии корпоративного управления
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему «Исследование методов и средств моделирования систем управления проектами на предприятии»

Студент

М.В. Марченко

(И.О. Фамилия)

(личная подпись)

Научный
руководитель

канд. пед. наук, доцент Е.В. Панюкова

(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
Глава 1 АНАЛИЗ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ	9
1.1 Основные понятия, используемые в управлении проектами	9
1.2 Модели жизненных циклов проектов и методы управления проектами ..	16
1.3 Анализ инструментов управления проектами	28
Выводы к главе 1	41
Глава 2 МЕТОДЫ И АЛГОРИТМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ В КОМПАНИИ ООО «МАСТЕР МАИНД ИНК».....	42
2.1 Алгоритмы процесса разработки программных продуктов и методы управления проектами на предприятии ООО «Мастер Маинд Инк».....	42
2.2 Организация процесса исследовательской деятельности на предприятии ООО «Мастер Маинд Инк».....	47
2.3 Модель системы управления проектами «As Is».....	59
Выводы к главе 2	63
Глава 3 ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ В КОМПАНИИ ООО «МАСТЕР МАИНД ИНК».....	65
3.1 Модель автоматизированной системы управления проектами на предприятии ООО «МАСТЕР МАИНД ИНК»	65
3.2 Качественная оценка модели автоматизированной системы управления проектами на предприятии ООО «МАСТЕР МАИНД ИНК»	75
Выводы к главе 3	81
ЗАКЛЮЧЕНИЕ	83
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	87
Приложение А Модель бизнес-процессов на этапе R&D.....	91
Приложение Б Модель “AS IS” управления проектами на этапе R&D.....	92
Приложение В Модель АСУ управления проектами на предприятии.....	93

ВВЕДЕНИЕ

Общественные и технологические условия современного мирового развития характеризуются постоянным возрастанием объема цифровых технологий, основывающихся на использовании искусственного интеллекта (ИИ), инновационными процессами в обществе в целом, а также в сферах производства, науки, бизнеса. Изменения, которые происходят в социуме и в IT-области, порождают новые требования к качеству управления проектами, воплощающими в жизнь «умные» программы: распознавание лиц, постановка диагнозов, беспилотное управление транспортом и т.д.

Работа по разработке нейросетей (как математической основы искусственного интеллекта) по своему содержанию является научно-исследовательской, то есть носит творческий характер, и значит, структура производственных отношений в процессе изготовления программного продукта не такая жесткая, менее иерархическая, причем иерархия в течение времени создания программы может нарушаться. Это обуславливает необходимость выбора более гибких и устойчивых к изменениям методологий управления проектами в компаниях, специализирующихся на разработке ИИ-продуктов. На сегодняшний день наиболее распространенными и популярными методологиям является Agile-семейство, включающее в себя Scrum, Kanban, XP и другие методы. Но они не обеспечивают в полной мере запросы управления ИИ-проектами, как наукоемкими и оперирующими большими массивами данных. Поэтому проектировщики предпочитают такой метод, как Crisp-Dm, доказавший свою эффективность при работе с Big Data.

Современный рынок информационных технологий предлагает большой выбор платформ для автоматизированного управления проектами. Ассортимент таких систем очень велик. Насчитывается около ста автоматизированных систем управления процессом создания программных продуктов. Самыми популярными из них являются Trello, Jira и Asana. Каждый из предлагаемых продуктов имеет свои преимущества и недостатки. Так, Trello ориентируется на малые и краткосрочные проекты, хорошо зарекомендовала себя в работе с

легко формализуемыми проектами; Jira удобна в использовании в больших командах, но при условии предварительной настройки и расширения базового набора функций с помощью плагинов, и надстроек. Данный процесс является достаточно трудоемким, требует много усилий. Сама Jira имеет сложную структуру и трудна в понимании для новых пользователей.

Большое разнообразие в клиентских запросах на рынке IT-услуг, постоянно расширяющийся спектр применения ИИ обуславливает существование различных видов компаний, различающихся как по количеству сотрудников, так и производимому продукту. Поэтому становятся актуальными вопросы адаптации существующих моделей управления проектами к конкретной компании.

С точки зрения менеджмента, исследовательская работа является одной из самых сложных объектов управления [6,12-14]. Это обусловлено особенностями творческого процесса мышления, неопределенностью и трудной предсказуемостью конечных результатов, невозможностью полной алгоритмизации процесса поиска новых решений.

Таким образом, выявляется *противоречие*: с одной стороны, объективно возрастает доля исследовательской работы в сфере IT-бизнеса, усиливается необходимость постоянного совершенствования системы управления инновационными и исследовательскими проектами в области ИИ; с другой стороны, методы системы управления являются недостаточно разработанными и зачастую не адекватными объекту управления. Такое несоответствие приводит к тому, что затраты, вложенные в проект, не окупаются, и работа оказывается нерентабельной.

Поиск путей разрешения указанного противоречия обусловил выбор **темы** диссертационного исследования: «Исследование методов и средств моделирования систем управления проектами на предприятии».

Таким образом, следующие факторы: постоянные и достаточно быстрые изменения в информационных технологиях; увеличение количества точек приложения искусственного интеллекта во всех сферах социума;

возникновение большого количества компаний, конкурирующих на рынке программных продуктов; усложнение разработок, содержащих ИИ; уникальность каждого проекта, который использует ИИ; увеличение доли исследовательского, новаторского, творческого труда в работе IT-специалиста; создание R&D-отделов в компаниях; ужесточение требований инвесторов; увеличение степени неопределенности и риска повышают ответственность менеджмента всех уровней компании, обуславливают необходимость постоянного совершенствования методов управления проектами и **актуализируют** данное исследование.

Объект исследования: система управления проектами, которые основаны на использовании искусственного интеллекта и машинного обучения.

Предмет: методы и средства моделирования автоматизированной системы управления проектами в компании, ориентированной на выпуск ИИ-содержащих продуктов.

Цель работы: теоретическое обоснование и построение модели автоматизированной системы управления проектами на предприятии ООО «Мастер Маинд Инк».

В соответствии с объектом, предметом и целью работы была сформулирована **гипотеза:** для разработки модели автоматизированной системы управления проектами на предприятии ООО «Мастер Маинд Инк» могут быть использованы существующие методы и средства управления проектами.

Цель и гипотеза исследования обусловили решение таких **задач:**

1) изучить современные методы и инструменты управления проектами, выяснить их преимущества и недостатки, особенности применения в различных IT-компаниях;

2) проанализировать существующие информационные системы по управлению проектами;

3) проанализировать содержание деятельности проектной команды предприятия ООО «Мастер Маинд Инк» в процессе создания программного продукта, использующего искусственный интеллект;

4) выяснить особенности управления проектами в компании ООО «Мастер Маинд Инк», применяемые методологии и инструменты управления проектами;

5) проанализировать организационную структуру управления проектной командой в компании ООО «Мастер Маинд Инк», оценить возможности усовершенствования процесса управления проектами;

6) выполнить проектирование и разработать модель АСУ проектами на предприятии ООО «Мастер Маинд Инк»;

7) выполнить качественную оценку разработанной модели с применением таких методик: метод экспертных оценок; SWOT-анализ; сравнительный анализ уровня автоматизации бизнес-процессов предприятия при использовании моделей управления «As Is» и «To Be».

Проблемы управления проектами рассмотрены во многих научных работах – статьях, учебниках, докладах на конференциях. На сегодняшний день существует большое количество исследований, посвященных методам управления проектами; создан институт управления проектами – Project Management Institute с дочерними организациями по всему миру, Международная ассоциация управления проектами – International Project Management Association; разработаны международные, национальные, отраслевые стандарты.

Исследования вопросов управления проектами отличаются разнообразием направлений. Так, рассматриваются общие вопросы управления проектами (Преображенская Т.В., Сербская О.В.), различные предметные области (Озерова Т., Мазур И.И.); методы управления проектами (Голубев С.А., Вареникова О.В., Чуланова О.Л., Александрова Т.В.); вопросы автоматизации систем управления проектами в различных отраслях промышленности, энергетике, транспорте (Шайхулова А.Ф., Дерябкин В.П.) и др.

Проблемы моделирования автоматизированной системы управления проектами рассматривались в работах Норенкова И.П., Николаенко В.С., Макашова П.Л., Романенко Н.А. Большое количество статей на тему автоматизированного управления проектами в IT-сфере содержит Интернет.

Однако вопросы моделирования АСУ проектами в IT-компаниях являются востребованными. Интерес к этой теме остается постоянным в силу динамичного развития информационно-компьютерной сферы.

В процессе выполнения исследования по теме диссертации были использованы следующие **подходы и методы**:

- общенаучные приемы и способы логического познания: анализ и синтез, абстрагирование;
- системно-структурный, функциональный и формально-логический подходы;
- анализ литературы по вопросам теории управления проектами, моделирования АСУ проектами, существующих стандартов в управлении проектами;
- эмпирические методы (индивидуальные и групповые интервью с сотрудниками компании, наблюдение за их деятельностью на протяжении процесса создания программного продукта, мозговой штурм, бэнчмаркинг);
- методы структурного и объектно-ориентированного анализа моделирования информационной системы (процессное моделирование, ER-моделирование, описание прецедентов использования, контекстная диаграмма) в нотациях IDEF1x, BPMN, UML;
- экспертная оценка, SWOT-анализ, оценка уровня автоматизации бизнес-процессов.

Научная новизна исследования заключается в том, что создана модель автоматизированной системы управления проектами в компании, специализирующейся на производстве продуктов, содержащих ИИ; был предложен способ представления жизненного цикла проекта и фаз проектной

деятельности в виде дерева решений; разработан алгоритм процесса проектной деятельности на этапе R&D.

Теоретическую значимость исследования для дальнейшего развития представляют: модель АСУ управления ИИ-проектами; способ визуализации жизненного цикла ИИ-проекта в виде дерева решений; алгоритм процесса проектной деятельности на этапе R&D.

Практическую значимость выполненное исследование представляет для небольших стартап-компаний, специализирующихся на создании нейронных сетей, где процессы автоматизированного управления пока не налажены; для устоявшихся компаний такой же специализации для повышения производительности работы R&D-отдела.

Положения, выносимые на защиту:

- алгоритм процесса проектной деятельности на этапе R&D;
- способ представления жизненного цикла ИИ-проекта и фаз проектной деятельности в виде дерева решений;
- модель информационной системы управления проектами на этапе R&D.

В структуру работы входят: введение, 3 главы, заключение, список используемой литературы и приложения.

Работа изложена на 90 страницах и включает 29 рисунков, 10 таблиц, 45 источников, 3 приложения.

Глава 1 АНАЛИЗ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ

1.1 Основные понятия, используемые в управлении проектами

Понятие «проект» широко распространено в разных сферах науки, производства, даже в быту. Рассмотрим существующие в научной литературе *определения* понятия «проект»:

- организованная деятельность, направленная на создание уникальных продуктов, услуг или результатов (согласно РМВоК);
- отдельное предприятие с определенными целями, часто включающими требования по времени, стоимости и качеству достигаемых результатов (согласно английской Ассоциации проект-менеджеров);
- предприятие (намерение), которое в значительной степени характеризуется неповторимостью условий в их совокупности, например: задание цели; временные, финансовые, людские и другие ограничения; разграничения от других намерений; специфическая для проекта организация его осуществления (в соответствии с немецким стандартом DIN 6990) [29].

По Бэбьюли Ф., проект - это последовательность взаимосвязанных событий, которые происходят в течение установленного ограниченного периода времени и направлены на достижение неповторимого, но в то же время определенного результата [4].

Ивасенко А.Г. дает следующее определение: «Проект – это ограниченное по времени целенаправленное изменение отдельной системы с изначально четко определенными целями, достижение которых определяет завершение проекта, с установленными требованиями к срокам, результатам, риску, рамкам расходования средств и ресурсов и к организационной структуре» [10].

По Фунтову В.Н., проектом называется «целенаправленная, ограниченная во времени деятельность, осуществляемая для удовлетворения конкретных потребностей при наличии внешних и внутренних ограничений и использовании ограниченных ресурсов» [25].

Как видно, приведенные определения отличаются достаточно общим подходом к термину «проект», но тем не менее позволяют выделить существенные особенности данного понятия. К таковым относятся:

- наличие четко сформулированных целей, а также ряда технических, экономических и других целевых показателей;
- системный характер любого проекта, то есть наличие внутренних и внешних связей между всеми элементами системы, а именно целями, задачами, операциями, ресурсами (включая людские), искомым результатом. Это дает возможность алгоритмизации проекта, то есть представления его в виде комплекса взаимосвязанных действий;
- наличие предварительно обозначенных временных интервалов (сроки начала и конца проекта);
- ограниченные ресурсы;
- определенная степень уникальности целей проекта и условий его осуществления.

Таким образом, *проект* – динамическая система действий, направленных на получение заданных результатов в многокритериальном поле в течение установленного срока и в рамках выделенных ресурсов с привлечением исполнителей, обладающих необходимыми навыками и знаниями.

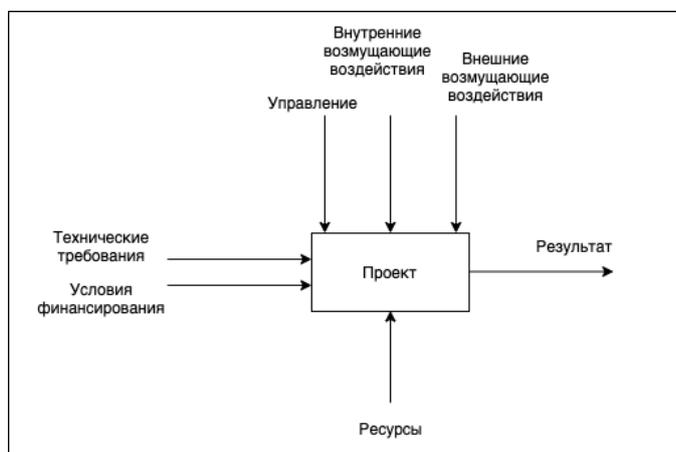


Рисунок 1.1 - Проект как система

Контекстная диаграмма (рис. 1.1) представляет проект как систему. Входными данными являются технические требования и условия финансирования; целью является достижение требуемого результата. Наличие ресурсов в виде материалов, финансов, человеческого ресурса обеспечивает выполнение работ. Эффективность определяется управлением процессом реализации проекта. В функции управления входит распределение ресурсов, координация выполняемой последовательности операций, компенсация возмущающих внутренних и внешних воздействий [21, 22]

В таблице 1.1 раскрывается содержание составных элементов проекта.

Таблица 1.1 — Описание элементов системы «проект»

Элемент	Содержание
Цель (результат)	Описываются новые продукты или услуги, которые получит заказчик в результате реализации проекта
Стоимость проекта	Финансовые затраты, требуемые для выполнения работ проекта
Технические требования	Объемы работ (количественные показатели объема работ проекта); сроки выполнения; качество (соответствие характеристик проекта и его продукции установленным заранее параметрам качества)
Ресурсы	Оборудование, материалы, персонал, программное обеспечение, информационные системы, производственные площади; специалисты и организации, привлеченные к выполнению работ проекта, их квалификация
Внутренние возмущающие воздействия	Стиль руководства проектом; организация проекта с точки зрения коммуникации между основными участниками проекта, распределения прав, ответственности и обязанностей; методы и средства взаимодействия между сотрудниками всех уровней на проекте; условия труда и техники безопасности, страхование и социальное обеспечение и т.п.
Внешние возмущающие воздействия	Взаимодействие с заказчиком и конкурентами; ситуация на рынке и связанные с этим риски; непредвиденные обстоятельства

Существуют также различные варианты определения термина «управление проектами». Рассмотрим некоторые из них.

Согласно PMBoK, управление проектами – это процесс применения знаний, навыков, методов, средств и технологий к проектной деятельности с целью воплощения замыслов участников проекта [17, 31].

Энтони Уокер определяет управление проектами как планирование, координацию и контроль проекта с позиций его завершения (и ввода в действие) от лица заказчика и с учетом его целей в единицах полезности, предназначения, качества, сроков реализации и затрат; установление взаимосвязи между ресурсами, координацию и контроль участников проекта, их персонального вклада в общий результат, а также оценку и выбор альтернатив ради наибольшего удовлетворения потребностей заказчика [45].

Гарольд Оберлендер считает управление проектами искусством и умением скоординировать материальные и нематериальные ресурсы, организовать последовательность работ по реализации проекта во времени и в рамках утвержденной стоимости [43].

По мнению Шапиро В. Д., управление проектами – это синтетическая дисциплина, объединяющая специальные и надпрофессиональные знания [24].

Таким образом, существует разнообразие в определении понятия «управление проектами».

Однако отличительной чертой любого проекта является организация взаимодействия между участниками проекта, осуществляемая через project manager'a. В процессе коммуникации обозначается круг проблем, возникающих при реализации проекта и их возможные решения.

Очевидно, что люди – наиболее значимый актив при управлении проектами, так как они – источник идей и их реализаторы.

В научной литературе присутствует определение различных типов проектов по разным основаниям.

В результате обобщения существующих в источниках [1,9,20] классификаций, построим таблицу 1.2, в левом столбце которой указано основание классификации, в правом – соответствующий вид проекта.

Таблица 1.2 — Виды проектов

Критерий	Вид проекта
по составу и структуре проекта	монопроект, мультипроект, мегапроект
по основным сферам деятельности, в которых осуществляется проект	технический, организационный, экономический, социальный, смешанный
по характеру предметной области проекта	инвестиционный, инновационный, научно-исследовательский, учебно-образовательный, смешанный
по продолжительности периода осуществления проекта	краткосрочные (до 2-х лет), среднесрочные (до 5-ти лет), долгосрочные (свыше 5-ти лет)
по масштабу (по размерам бюджета, количеству участников и степени влияния на окружающий мир)	мелкие, средние, крупные
степени охвата этапов инновационного процесса	полные инновационные проекты, включающий НИР, ОКР, освоение новшества и его коммерциализацию, неполные инновационные проекты, включающие отдельные этапы инновационного процесса
по сложности	простые, сложные, очень сложные
по отраслям экономики и социальной сферы	промышленность, строительство, транспорт, здравоохранение, туризм
по результирующему продукту	продукт (часть другого изделия, усовершенствование изделия или конечное изделие); услуга (бизнес-функция, оптимизирующая производственные процессы); улучшение (существующей линейки продуктов или услуг); нематериальный продукт (например, исследовательский проект приносит новые знания)

Классификация проектов позволяет достаточно четко ранжировать перспективные и реализуемые проекты, и как следствие, ставить выполнимые цели, задавать реальные сроки достижения целей, и привлекать оптимально необходимые ресурсы для их успешной реализации [2, 19].

Понятие *жизненного цикла* проекта подразумевает под собой определенную последовательность этапов по реализации той или иной идеи касательно производственного или управленческого процесса. Важность данного понятия обуславливается тем, что оно фиксирует продолжительность проекта, четко обозначая сроки его выполнения; позволяет детализировать процесс реализации замысла, разбивая его на конкретные фазы; дает возможность четко определить количество задействованного персонала, а также необходимые ресурсы; облегчает процедуру контроля.

Жизненный цикл проекта – это совокупность фаз, через которые реализуется первоначальный замысел. Такое разделение важно не только с теоретической, но также и с практической точки зрения, ведь оно дает возможность лучше контролировать процесс производства программного продукта. Так, принято выделять следующие этапы жизненного цикла IT-проекта: инициация, планирование, исполнение, завершение (рис. 1.2).

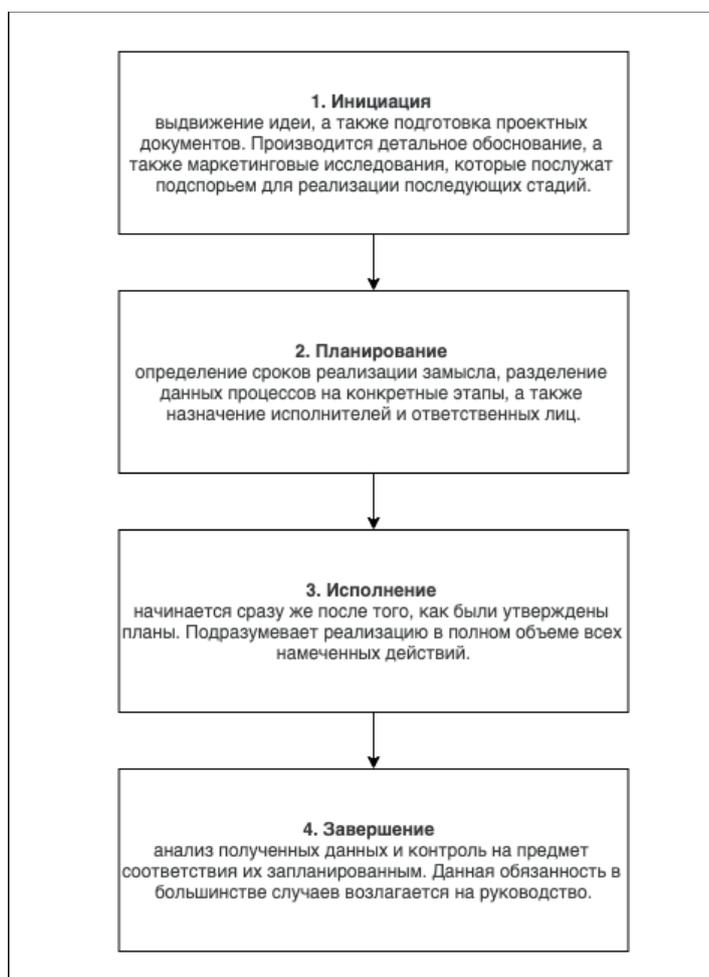


Рисунок 1.2 - Этапы жизненного цикла проекта

При изучении жизненных циклов проектов было замечено, что, несмотря на их различную направленность, всем проектам присущи следующие общие свойства:

- самые высокие показатели затрат и количество единиц персонала, присутствующего на проекте, присущи середине цикла; начало и конец данного процесса характеризуются невысокими показателями;
- уровень риска наиболее высок в начале жизненного цикла;
- именно в начале жизненного цикла проекта сотрудники имеют большую свободу по внесению изменений и совершенствования технологии реализации проекта, с течением времени это становится сделать все сложнее [8,17].

Как указывает Султанов И.А., «каждый project manager, набирая опыт, все больше понимает значимость жизненного цикла для того, чтобы проектная реализация с каждым разом проводилась все безопаснее и с более прогнозируемым результатом. В этом помогает не только система оценки рисков. Большое значение имеет планирование проекта по фазам его ЖЦ.

После каждого этапа намечаются вехи.

В эти моменты руководители обязаны остановиться, оценить достигнутый результат, осуществить прогнозный анализ и решить дальнейшую судьбу уникальной задачи.

Опыт, знания и управленческая интуиция одного из лидеров бизнеса позволяют доверить ему столь ответственные решения»[32].

Методологическая ценность модели жизненного цикла в том, что она помогает понять особенности исполнения работы, вовремя принять нужные управленческие решения, поскольку модель – это подробное описание последовательности необходимых действий, что позволяет не терять контроль над разработкой проекта ни на одной стадии жизненного цикла.

1.2 Модели жизненных циклов проектов и методы управления проектами

Управление проектами как вид профессиональной деятельности и как объект научных исследований получает существенное развитие в 1980-х годах, когда мировая экономика выходит из кризиса, растет насыщение рынка и возникает необходимость решения новых, больших задач. Именно тогда выходит первая значительная работа PMBoK (A Guide to the Project Management Body of Knowledge), выполненная в Project Management Institute (PMI). Данная организация на сегодняшний день является наиболее авторитетной профессиональной ассоциацией, разрабатывающей стандарты в области управления проектами. Среди других институтов, занимающихся стандартизацией проектного менеджмента, следует назвать IPMA (International Project Management Association), OGC (The Office of Government Commerce, стандарты PRINCE2), ISO (International Standardization Organization), APM (Association for Project Management) и др. Следует также отметить, что, кроме международных стандартов, существуют также национальные системы стандартизации, отраслевые и корпоративные.

Развитие IT-сферы, интенсивная работа в области программного обеспечения привели к накоплению (и этот процесс не останавливается) большого практического опыта («best practice»). Комплекс таких «лучших практик», реализуемых на различных стадиях жизненного цикла проекта и базирующихся на общей идеологии, стандарт SWEBOOK [44] называет «методология разработки программного обеспечения». Методологии, или методы разработки программных продуктов на сегодняшний день являются самой быстро развивающейся областью информационных технологий, так как опираются на реальные практические знания.

Методы управления проектами тесно связаны с жизненным циклом проекта. Можно сказать, что они взаимно обуславливают друг друга. Фазы жизненного цикла проекта, представленные на рис. 1.2, являются весьма обобщенными и одинаково присущими проектам, выполняемым, например, в

строительной сфере, где все этапы четко обозначены и понятна последовательность разных видов работ, или, к примеру, в авиационной или космической области, где возможно распараллеливание операций, или повторение (если есть необходимость) одних и тех же видов работ [5, 7, 15, 19].

Поэтому при разработке конкретного программного продукта необходима подробная детализация его жизненного цикла и адекватная (содержанию проекта) модель.

«В общем случае, жизненный цикл определяется моделью и описывается в форме методологии (метода). Модель или парадигма жизненного цикла определяет концептуальный взгляд на организацию жизненного цикла и, часто, основные фазы жизненного цикла и принципы перехода между ними. Методология (метод) задает комплекс работ, их детальное содержание и ролевую ответственность специалистов на всех этапах выбранной модели жизненного цикла, обычно определяет и саму модель, а также рекомендует практики (best practices), позволяющие максимально эффективно воспользоваться соответствующей методологией и ее моделью» [33].

Традиционными и хронологически первыми были разработаны каскадная (водопадная, waterfall) модель, впервые описанная в 1970 году в работе Уинстона Ройса [42] и спиральная (Spiral), предложенная Барри Бозом в 1986 г. [40]. Позже, с усложнением информационных технологий и нарастающим функционалом программных продуктов возникли так называемые гибкие (Agile) модели [23].

В настоящее время они получили наибольшее распространение.

Рассмотрим современные методы и алгоритмы управления проектами и проведем их сравнительный анализ по наиболее существенным параметрам.

I. Каскадная модель (Waterfall).

Самой старой и известной моделью построения многоуровневого процесса разработки является каскадная (или водопадная) модель: в ней каждый этап разработки, соответствующий стадии жизненного цикла ПО,

продолжает предыдущий. То есть новый этап начинается только после полного завершения текущего.

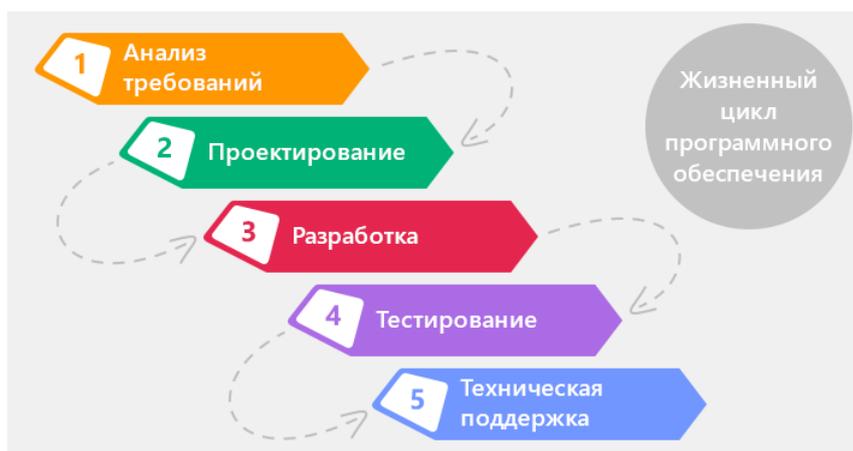


Рисунок 1.3 – Каскадная модель жизненного цикла

Каскадная модель проста и понятна, но не так практична, как раньше. В условиях динамично изменяющихся требований, строго структурированный процесс может из преимущества превратиться в помеху на пути успешного завершения разработки системы. Поэтому сегодня водопадная модель применяется преимущественно крупными компаниями для больших и сложных проектов, которые предполагают всеобъемлющий контроль рисков [27].

Преимущества каскадной модели:

- понятная и чёткая схема рабочего процесса;
- возможность просчёта точного количества затраченных на проект ресурсов;
- не требует затрат по налаживанию коммуникаций между всеми членами команды.

Недостатки каскадной модели:

- приоритет формального подхода к последовательности процесса работы;
- невозможность внесения изменений заказчиком до окончания разработки продукта;

- в случае нехватки ресурсов страдает качество проекта из-за сокращения этапа тестирования.

Несмотря на то, что каскадная модель все еще используется, она уже утратила былые позиции. Сегодня ей на смену приходят более продвинутые модели и методологии разработки программного обеспечения.

II. Спиральная модель.

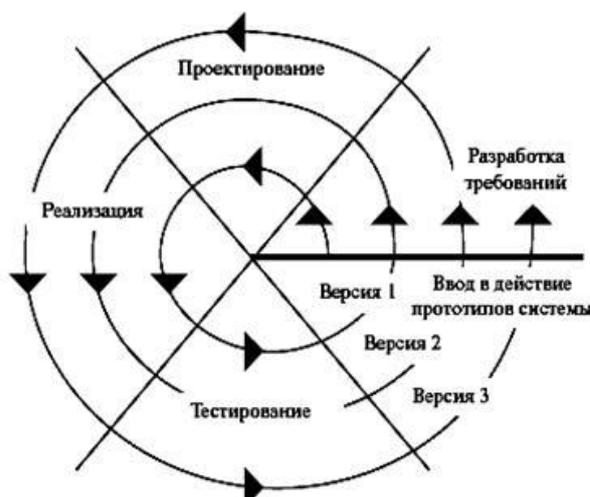


Рисунок 1.4 – Спиральная модель жизненного цикла

Спиральная модель использует разбиение проекта на итерации. Повышенное внимание уделяется начальным этапам разработки – анализу и проектированию. Основная функция начального этапа – обоснование возможности реализации программного решения; для этого создается прототип будущего продукта [18].

Свое название спиральная модель получила вследствие того, что жизненный цикл проекта в этой модели благодаря итерационному (то есть постепенно приближающему) подходу создает образ спирали – с преодолением каждой итерации, или отдельного цикла, будущее программное изделие приобретает требуемые формы и содержание.

Такой подход позволяет на каждой итерации уделять время уточнению целей и параметров проекта, выяснению возникающих затруднений, планированию работ на следующем витке.

Благодаря такому подходу к разработке время выполнения всех поставленных задач уменьшается.

Преимущества спирального подхода:

- преодоление жесткого, регламентированного процесса разработки в каскадной модели; создание дополнительных возможностей;
- облегчение процесса внесения корректив в проект, если требования заказчика изменились;
- постоянная связь пользователя будущего продукта с разработчиком, что позволяет оперативно отслеживать возникающие ошибки или слабые места проекта;
- более эффективное управление проектом благодаря наличию итераций;
- снижение уровня рисков.

Трудности использования спиральной модели:

- отсутствие ограничений времени на каждом из этапов жизненного цикла;
- то преимущество, что имеется возможность постоянно вносить изменения в требования к проекту, оборачивается тем, что количество циклов может расти, следовательно, удлиняются сроки разработки;
- довольно сложная структура модели, что создает затруднения для разработчиков и менеджеров при ее применении.

III. Гибкие (итеративные) модели (Agile).

Кроме последовательных моделей, существуют итеративные (инкрементальные). Это целое семейство методологий, объединенных общим подходом в управлении проектами:

- люди и взаимодействие важнее процессов и инструментов;
- работающий продукт важнее исчерпывающей документации;
- сотрудничество с заказчиком важнее согласования условий контракта;

- готовность к изменениям важнее следования первоначальному плану [26,28].

В таких моделях жизненный цикл проекта разбит на некоторое количество мини-циклов, каждый из которых состоит из базовых стадий модели жизненного цикла. Эти мини-циклы называются итерациями. Разработка отдельного компонента систем происходит внутри итерации, затем этот компонент добавляется к ранее разработанному функционалу.

Итеративная модель не нуждается в полном объеме требований для начала работ над продуктом. Разработка программы может начинаться с требований к части функционала, причем впоследствии требования могут претерпевать изменения и дополнения. Путем повторения процесса происходит обновление версии продукта на каждом цикле.

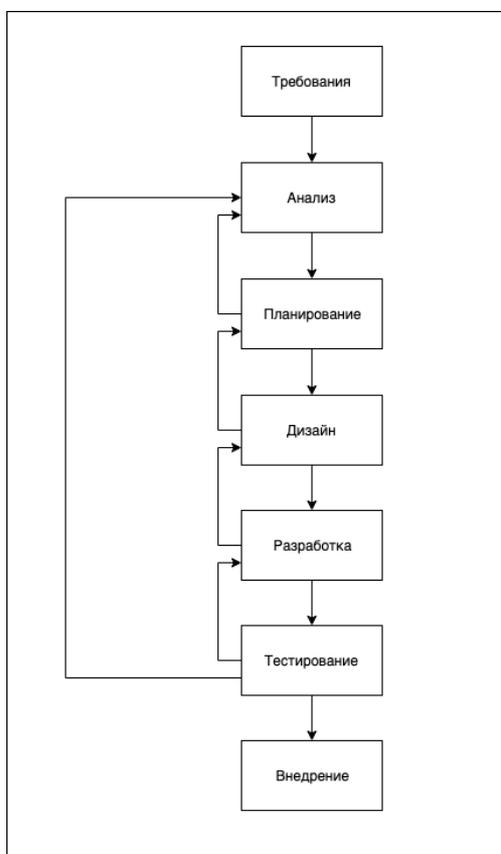


Рисунок 1.5 - Итеративная модель жизненного цикла

Решение об использовании результатов предыдущей итерации в качестве входных для следующей принимается по окончании предыдущей (т.н.

инкрементальное прототипирование). В конечном итоге, достигается точка, в которой все требования были воплощены в продукте – происходит релиз.

Используя математическую терминологию, итеративная модель реализует последовательную аппроксимацию, то есть приближение к заданным параметрам, т.е. готовому продукту.

Успех использования этой модели определяется строгой валидацией требований и верификацией разрабатываемой функциональности на каждой итерации.

Основные стадии процесса разработки в итеративной модели фактически повторяют модель водопада. В каждой итерации создается программное обеспечение, требующее тестирования на всех уровнях.

Достоинства итеративной модели:

- высокий уровень взаимодействия между членами команды проекта;
- быстрый результат (рабочий код) в итоге «спринтов»;
- стимулирование изменения и улучшений продукта во время его разработки;
- непосредственное вовлечение заказчика в рабочий процесс;
- высокая способность к адаптивности под разные процессы и условия;
- быстрый отклик на изменения внешних и внутренних условий проекта;
- приспособлен для разработки инновационных продуктов с высокой степенью неопределенности и недостаточной информативностью;

Недостатки:

- риск бесконечных изменений продукта;
- большая зависимость от уровня квалификации и опыта команды;
- практически невозможно точно подсчитать итоговую стоимость проекта;
- эффективность применения Agile-методик существенно зависит от квалификации менеджмента (для облегчения применения подхода принято использовать методы Scrum, Kanban и другие).

Наиболее востребованными представителями семейства Agile-методологии являются Scrum, Lean, Canban, Six Sigma, PRINCE2. Рассмотрим подробнее Scrum и Canban.

III.1. Scrum - наиболее структурированный в группе Agile-методов.

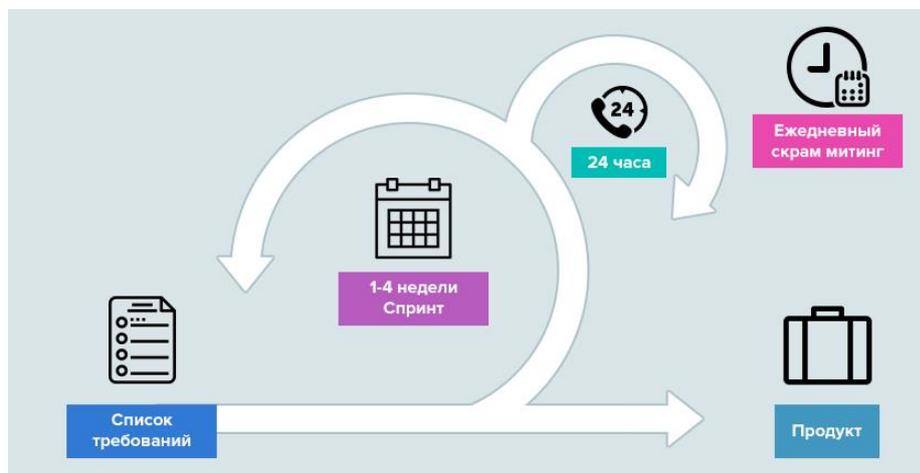


Рисунок 1.6 - Scrum модель жизненного цикла

Упрощенная схема работы по Scrum состоит в следующем. Согласно технологии Scrum, проект разбивается на части - задачи, которые будут создаваться в течение проекта. После этого полученным частям присваивается свой приоритет.

Хронологически проект разбивается на итерации - определенные промежутки времени, устанавливаемые командой и согласованные с заказчиком для текущего контроля и управления проектом.

Наиболее важные задачи первыми отбираются для выполнения в итерации.

По итогам итерации заказчик получает рабочий промежуточный вариант продукта. Чтобы убедиться в соответствии проекта требованиям заказчика, перед началом любого спринта нужно выполнять переоценку еще не выполненного содержания проекта и вносить в него изменения.

Участие в этом процессе принимает руководитель проекта, команда и инициатор. Ответственность распределяется на всех участников.

Процесс работы в среде Scrum опирается на пять производственных совещаний:

- совещание-упорядочивание; в первый день новой итерации происходит обсуждение того, что уже сделано по проекту, что еще предстоит выполнить;
- совещание-планирование; команда выбирает из списка приоритетных задач те, которые она будет выполнять на протяжении текущей итерации;
- совещания-летучки; происходят ежедневно, оперативно (до 15 минут) для обмена сведениями о текущем состоянии выполнения индивидуальных задач каждого члена команды;
- итоговые совещания; команда демонстрирует свой результат всем заинтересованным лицам; выясняется, насколько продукт соответствует целям и ожиданиям;
- совещание-ретроспектива. Проводится анализ того, насколько слаженно работала команда, обсуждаются возникшие проблемы методов и коммуникации, делаются определенные выводы.

Преимущества Scrum:

- удобен для проектов, в которых возможен быстрый, оформленный и пригодный для первоначального использования, результат;
- тесная и постоянная коммуникация сотрудников допускает присутствие в команде малоопытного разработчика;
- быстрое исправление ошибок благодаря мгновенной обратной связи между всеми участниками проекта.

Недостатки Scrum:

- подходит не каждому темпераменту; наличие постоянной и очень тесной коммуникации всех участников проекта выдвигает требования к психологическим свойствам и социальным навыкам участников проекта;
- не подходит для проектов из строго регламентированных и нормированных областей (например, юриспруденция).

III.2. Kanban-метод из группы Agile.

В отличие от Scrum, Kanban-метод не предполагает наличие четких временных итераций. Процесс создания продукта не разделяется на итерации, а предполагает ограничение по количеству одновременно выполняемых задач.

Особенности метода состоят в том, что нет ограниченных по времени итераций, регламентированных производственных совещаний, каждый сотрудник команды может решать несколько задач одновременно. Отсутствие временных рамок каждого этапа позволяет приостановить выполнение текущей задачи, если изменился ее приоритет или появились иные, более важные задачи.

Преимущества Kanban:

- отсутствие дедлайнов;
- экономически эффективен вследствие разумного распределения количества задач на каждого члена команды.

Недостатки Kanban:

- наиболее эффективен в командах, члены которых взаимозаменяемы;
- не пригоден для проектов, сроки выполнения которых строго оговорены.

IV. Crisp DM.

С развитием методов создания искусственного интеллекта встал вопрос пересмотра жизненной модели цикла проектов. Программные продукты, разрабатываемые на основе искусственного интеллекта, существенно отличаются от стандартных программных продуктов (то есть таких, которые не содержат ИИ) в организации структуры жизненного цикла. Кроме того, такие продукты основываются на использовании больших массивов данных, а это требует других подходов к управлению проектами.

Платформа CRISP-DM изначально была создана для Data Mining – глубинного анализа данных, по впоследствии обнаружилось, что она хорошо моделирует жизненный цикл ИИ-содержащих программных продуктов.

Содержание модели CRISP-DM жизненного цикла исследования данных заключается в интерпретации и обработке данных. Внутри процесса построения Data Science-модели не происходит накопления результатов как при итеративных, спиральных и других моделях. Каждый новый цикл обновляет (или даже обнуляет) те результаты, которые получились в предыдущем цикле.

Рассмотрим подробнее модель CRISP DM. В общем виде она представлена на рисунке 1.4.

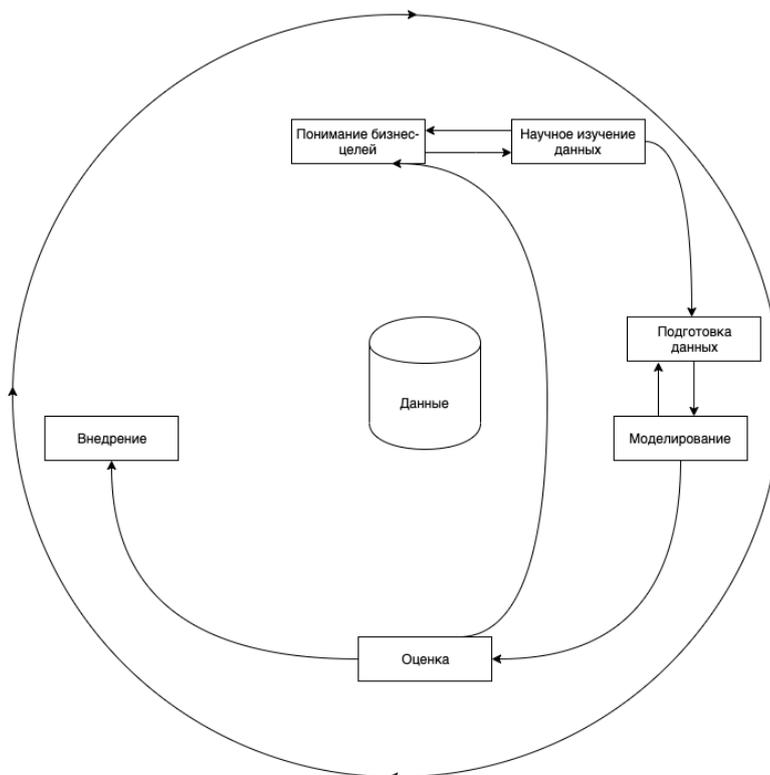


Рисунок 1.7 - Модель жизненного цикла исследования данных CRISP-DM

Модель жизненного цикла CRISP-DM содержит шесть фаз:

1) понимание бизнес-целей (Business Understanding). Содержанием данной фазы является следующее: исследование бизнес-процессов компании и выдвижение предложений по анализу данных исходя из конечных целей анализа. В обсуждении принимает участие максимальное количество заинтересованных лиц, в конечном итоге вырабатывающих план аналитического проекта. Решается также вопрос о целесообразности проекта;

2) начальное изучение данных (Data Understanding): подробное изучение имеющихся данных с целью устранения проблем на стадии подготовки данных.

В результате организуется доступ к данным, их исследование с использованием таблиц и графиков, оценка качества данных, разработка необходимой документации;

3) подготовка данных (Data Preparation): наиболее трудоемкий и ответственный этап, поглощающий 50-70% времени и ресурсов. Подготовка данных — это их консолидация; формирование выборок; агрегирование; обогащение; очистка; разделение данных на обучающие и тестовые;

4) моделирование (Modeling): построение и внедрение аналитических моделей, обычно в течение нескольких итераций;

5) оценка (Evaluation) соответствия результатов проекта критериям. Решение о соответствии принимается наиболее ответственными лицами компании, формулирующими бизнес-цели;

6) внедрение (Deployment) как процесс использования новых идей и знаний для повышения эффективности деятельности предприятия.

Business Understanding/ Бизнес-анализ	Data Understanding/ Анализ данных	Data Preparation/ Подготовка данных	Modeling/ Моделирование	Evaluation/ Оценка решения	Deployment/ Внедрение
Determine Business Objectives/ Определение бизнес-целей	Collect Initial Data/ Сбор данных	Select Data/ Выборка данных	Select Modeling Techniques/ Выбор алгоритмов	Evaluate Results/ Оценка результатов	Plan Deployment/ Внедрение
Assess Situation/ Оценка текущей ситуации	Describe Data/ Описание данных	Clean Data/ Очистка данных	Generate Test Design/ Подготовка плана тестирования	Review Process/ Оценка процесса	Plan Monitoring and Maintenance/ Планирование мониторинга и поддержки
Determine Data Mining Goals/ Определение целей аналитики	Explore Data/ Изучение данных	Construct Data/ Генерация данных	Build Model/ Обучение моделей	Determine Next Steps/ Определение следующих шагов	Produce Final Report/ Подготовка отчета
Produce Project Plan/ Подготовка плана проекта	Verify Data Quality/ Проверка качества данных	Integrate Data/ Интеграция данных	Assess Model/ Оценка качества моделей		Review Project/ Ревью проекта
		Format Data/ Форматирование данных			

Рисунок 1.8 – Задачи фаз Crisp DM

На рисунке 1.8. показаны задачи каждой фазы жизненного цикла проекта, разрабатываемого на основе нейронной сети.

К преимуществам данной модели можно отнести нейтральность в отношении предметных областей; удобство использования на проектах, использующих ИИ и МО; одна из немногих моделей, включающая

интеллектуальный анализ данных; является одним из важнейших понятий для технологий больших данных (Big Data).

Недостатки модели следующие: отсутствие инструментов управления проектами; отсутствует критерий времени для каждого из циклов; циклы могут быть разной продолжительности.

1.3 Анализ инструментов управления проектами

Инструменты, используемые менеджером проекта, могут применяться как в течение всего проекта, так и на определенном этапе (фазе) жизненного цикла.

Под *инструментами управления* мы будем понимать законченные формализованные методики, процедуры, а также шаблоны необходимых проектных документов.

Программное обеспечение **Trello** для управления Agile-проектами. Trello - облачная программа для управления проектами небольших групп, разработанная Fog Creek Software.

Это бесплатная многоплатформенная система управления проектами (рис. 1.9, 1.10).

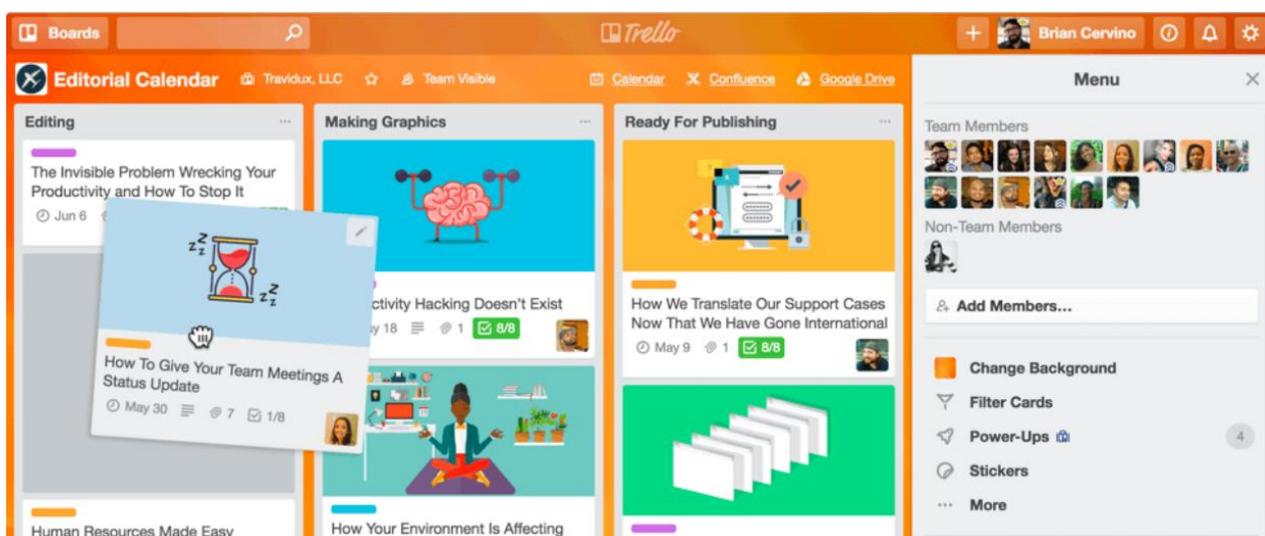


Рисунок 1.9 – Trello Board

Она использует парадигму управления проектами, известную как канбан.

Проекты изображаются досками, содержащих списки.

Списки содержат карты, которыми изображаются задачи.

Карточки должны переходить из предварительного списка к следующему (с помощью перетаскивания), таким образом изображая движение какой-то функции от идеи, до тестирования.

Карточке может быть присвоено имя ответственных за нее пользователей. Пользователи и доски могут объединяться в команды.

Trello имеет ограниченную поддержку тегов в виде шести цветных меток.

Карточки могут содержать комментарии, вложения, дату завершения и перечни (списки подзадач).

Форматируются карточки разметкой Markdown [21].

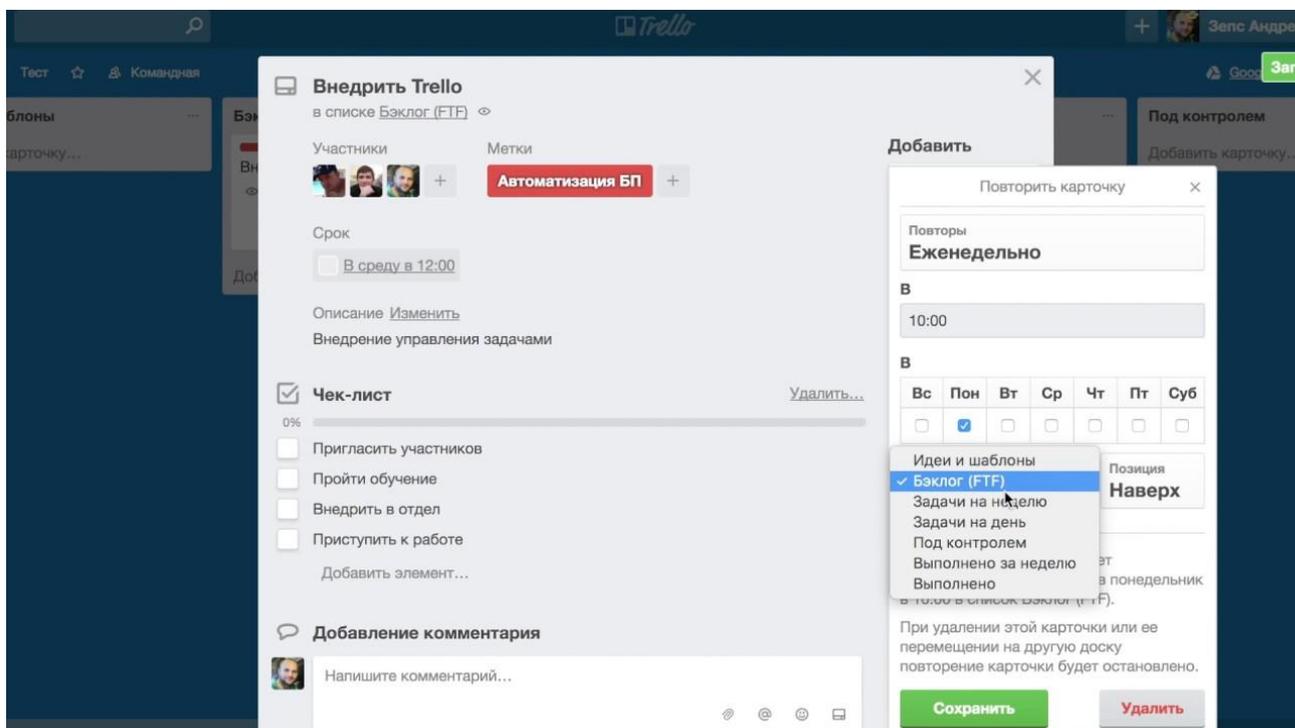


Рисунок 1.10 - Карточки задачи в Trello

Преимущества работы с Trello заключаются в следующем:

- наличие функциональных возможностей, упрощающих процесс управления проектом, а именно: обмен файлами (включая фотографии и видео)

со своими членами команды; комментарии к карте; слежение за to-do list; установление цветных меток в соответствии с приоритетом; неограниченное создание карт под проект;

- наличие функциональных возможностей, упрощающих процесс управления проектом, а именно: обмен файлами (включая фотографии и видео) со своими членами команды; комментарии к карте; слежение за to-do list; установление цветных меток в соответствии с приоритетом; неограниченное создание карт под проект;

- стоимость. По сравнению с другими инструментами управления проектами, Trello имеет менее сложную структуру ценообразования. Более простой вариант версии позволяет пригласить неограниченное количество участников, создать доски, карты и списки. Версия бизнес-класса, стоимость которой 25 долларов США в месяц, предоставляет целый ряд функций, таких, как интеграция в Службы Google, легкий объемный экспорт, а также возможность доступа и управления всеми досками [34];

- понятный мониторинг времени. Вы никогда не пропустите дату завершения проекта или важную встречу с Trello Борд. При создании Cards можно добавить соответствующие даты (deadline). Карточка станет желтой за 1-2 дня до окончания промежутка времени, отведенного на выполнение, карточка станет полностью красной в день сдачи проекта;

- mobile friendly. Trello работает на каждой платформе. Независимо от того, находится ли он на компьютере, планшете или телефоне, инструмент может адаптироваться под размер экрана;

- instant notifications. Вы всегда будете в курсе новостей, так как Trello оснащена функцией мгновенного оповещения при обновлении, комментировании или удалении задачи. Также можно получать оповещения по электронной почте.

Таблица 1.3 — Обзор Trello

Назначение продукта	Отслеживание задач проекта и их статусов. Управление небольшими проектами.
Users	Software developers Project managers SCRUM masters
Use Cases	Управление проектами Менеджмент продукта Управление задачами Разработка программного обеспечения Agile разработка программного обеспечения
Интеграция	Confluence JIRA Slack GitHub
Hosting options	Cloud

Недостатки работы с Trello следующие:

- нет offline-доступа. Trello не будет работать, если нет wifi. Если вы путешествуете или ваше электронное устройство переключено на режим «в самолете», вы не получите доступа к Trello . Доступ к сайту Trello требует подключение к Interwebs. Эти особенности, присущие Trello, создают определенный дискомфорт для пользователя, находящегося офлайн;
- данная платформа не предназначена для работы над крупными проектами. Оптимальное ее применение - стартапы, домашние проекты или некоторая идея, которая требует проверки. Trello - неудобный инструмент управления масштабными проектами с несколькими командами со всего мира;
- ограниченный объем файлохранилища. Количество вложенных файлов на карте Trello может быть любым, но для каждого вложенного файла существует лимит загрузки файла до 10 МБ. Члены бизнес-класса и Trello Gold могут пользоваться ограничением загрузки файлов размером до 250 Мб. Лимита хранения данных учетной записи не существует;

- неудобство комментирования. После опубликования и сохранения комментария на карте редактировать его в случае необходимости невозможно, придется писать новый комментарий;
- недостаточный уровень визуализации процесса работы всех членов команды одновременно;
- невозможно создать долгосрочные планы. Доска проста и удобна для разработки программного обеспечения небольших проектов, но для проектов длительностью даже в две недели создать дорожную карту оказывается невозможным;
- нет возможности просматривать итерации. Эффективность работы повышается при условии регулярного наблюдения за выполнением текущих задач, продуманного плана контроля предстоящих заданий, а также возможности анализа и корректирования уже выполненной работы. Trello не располагает встроенными инструментами для просмотра и анализа реализованных задач для успешного проведения ретроспектив;
- нет возможности вносить данные о затраченном на задачу времени. Отсутствие возможности для менеджеров проекта и его участников отслеживать количество времени, затраченное на выполнение каждой задачи. Это затрудняет составление ежемесячной отчетности, а также осложняет контроль за выполнением задачи.

Программное обеспечение **ASANA** для управления Agile-проектами. Asana (Асана) — одно из ведущих мобильных и веб-приложений для управления проектами в небольших командах.

Это универсальное решение для управления проектами и задачами, которое позволит автоматизировать некоторые из наиболее сложных задач, таких, как коммуникация и удобное сотрудничество. Многие компании хвалят Asana за точность и эффективное использование времени, особенно отмечая удобство отслеживания задач и обсуждения их в реальном времени (рис. 2.9).

Основной акцент создатели сервиса делают на том, что теперь управлять проектами возможно и без использования электронной почты. Разработчики

Asana предоставили пользователям возможность обсуждать рабочие процессы на той же платформе, где они были созданы, и усовершенствовали программу дополнительным функционалом: notes, groups, combined tasks, and followers [35].

Asana мгновенно сообщает о каждом изменении, которое может оказать влияние на проект. Программное обеспечение помогает сделать качественную оценку проекта и управлять рисками до момента подписания контракта с клиентом.

Сервис предлагает синхронизацию в режиме реального времени для различных устройств; проверку вашего почтового ящика; создание плана на грядущий день; просмотр, редактирование и создание задач и проектов с приоритетами; структурирование проектов с помощью разделов; лайки к задачам; обсуждения по проекту и задачам; настраиваемые дэшборды; календари; поиск задач, проектов, людей и тегов; прикрепление файлов из Dropbox, Box или Google Диска; канбан-доски.

Основные характеристики Asana представлены в таблице 1.4.

Таблица 1.4 — Обзор Asana

Назначение продукта	Планирование, документирование, отслеживание и выпуск программного обеспечения.
Users	Software developers Project managers SCRUM masters
Use Cases	Управление проектами Менеджмент продукта Управление задачами Разработка программного обеспечения Agile разработка программного обеспечения
Интеграция	JIRA Slack GitHub
Hosting options	Cloud

Рассмотрим *преимущества* Asana:

- расширенный список функций управления;
- существует бесплатная пробная версия. Сервис также бесплатен для команды до 15 человек;
 - обозначение цветом наиболее важных задач. Удобный функционал для making priority for task (рис. 1.11);
 - неограниченный функционал добавления более одного тега к каждому элементу (рис. 1.12);
 - поддержка мобильной версии для различных платформ;
 - одна история / элемент / задача может иметь ссылки на более чем один проект и тег;
 - есть поддержка горячих клавиш;
 - добавление новых задачи без необходимости заполнения многих полей;
 - интеграция электронной почты;
 - четкое дерево задач, которые нужно выполнить. Возможность увидеть все задачи в одном месте.
- платформа «Премиум» стоит 10 долларов США за каждого члена/в месяц (в случае, если вы хотите иметь дополнительный и расширенный функционал) [37].

Недостатки Asana следующие:

- можно назначить историю/задачи только одному из членов вашей команды;
- графические детали перегружают интерфейс, делая его медленным при перемещении;
- трудно работать с задачами в задании (sub-tasks). Есть возможность потерять связь при перемещении элементов;
- чрезмерное количество функций, которые отображаются на экране, затрудняют и замедляют процесс обучения.

- сообщения через программу зашифрованы, но, к сожалению, не существует двухэтапной проверки. Отсутствие two-step verification делает программное обеспечение небезопасным для ваших личных данных [37].

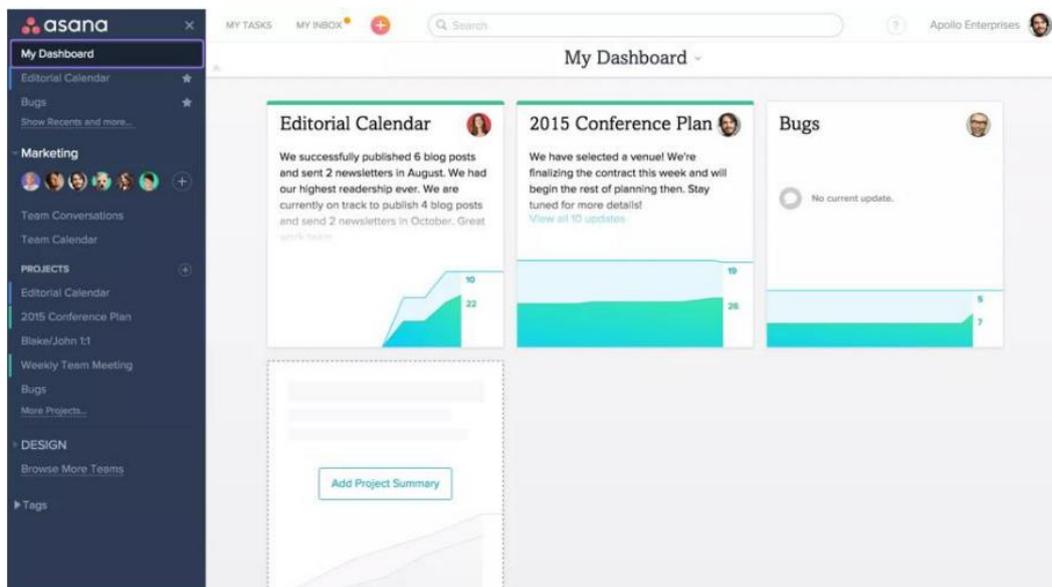


Рисунок 1.11 - Asana Dashboard

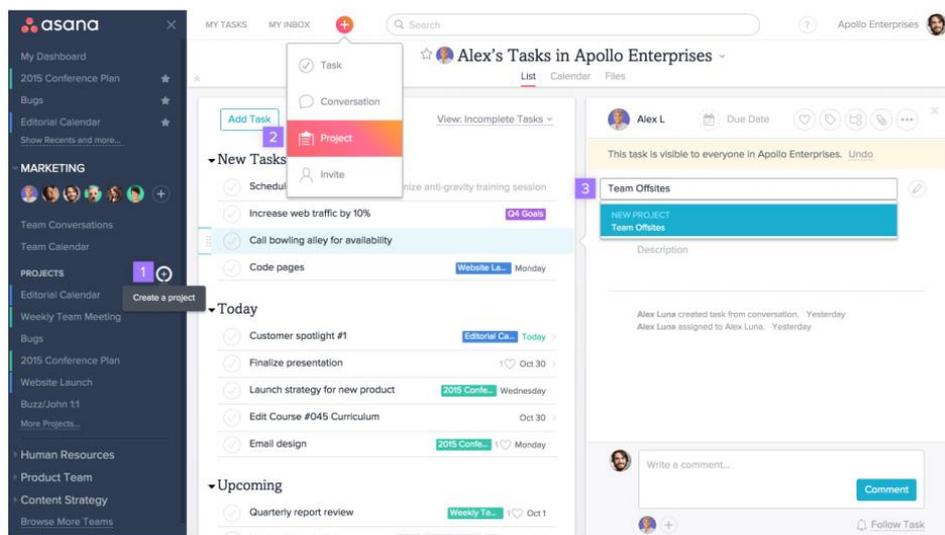


Рисунок 1.12 - Открытая задача в Asana

Программное обеспечение **JIRA** для управления Agile-проектами. JIRA - система отслеживания ошибок. Данный продукт разработан для организации взаимодействия с пользователями, также ее можно использовать и для управления проектами. Разработчик указанной системы - компания Atlassian.

Jira является одним из двух её основных продуктов (наряду с вики-системой Confluence). Одним из достоинств Jira является наличие веб-интерфейса [16].

JIRA — это комплекс систем, позволяющий координировать работу всех команд, занятых разработкой нового продукта. JIRA предлагает несколько продуктов и вариантов развертывания, предназначенных специально для программного обеспечения, ИТ, бизнеса, команд операторов и тому подобное.

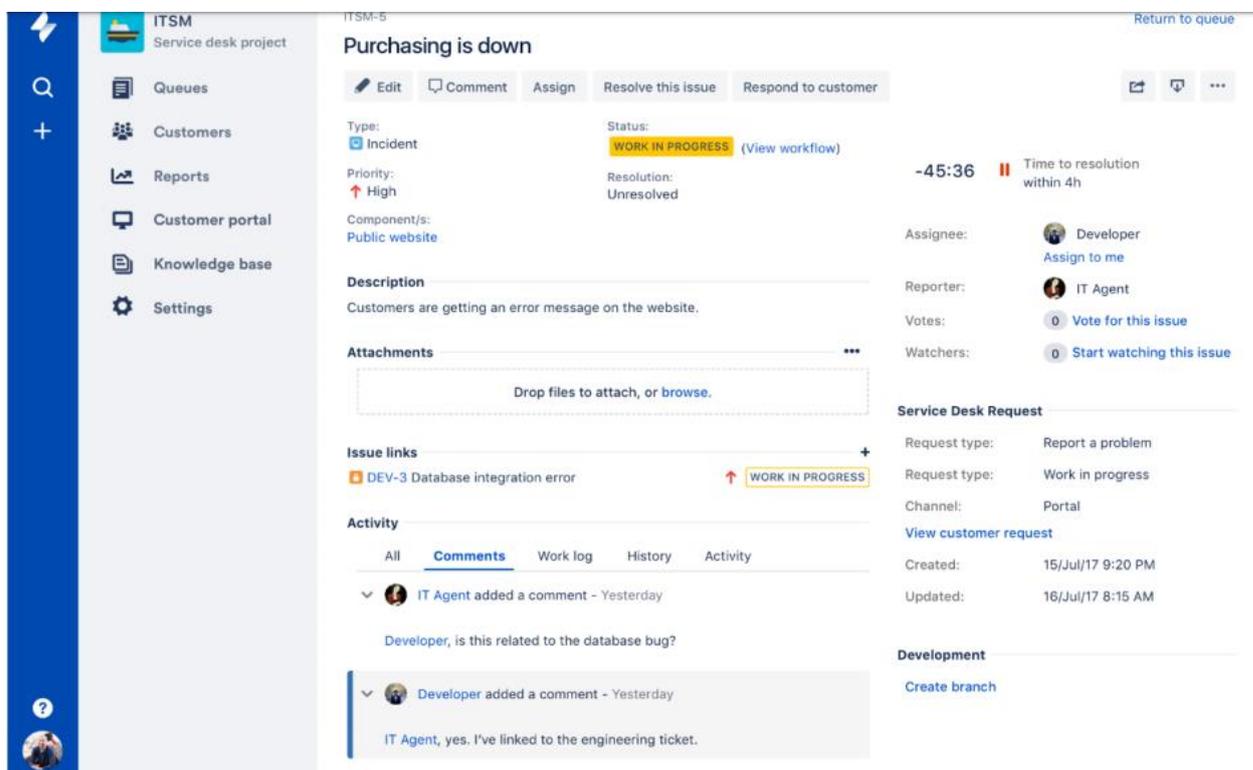


Рисунок 1.13 - Открытая задача в JIRA

Продукты и программы, построенные на платформе JIRA, помогают командам планировать, назначать, отслеживать, отчитываться и управлять работой. Платформа JIRA объединяет команды во всем: от гибкой разработки программного обеспечения и поддержки клиентов до управления списками покупок и семейными делами (рис.1.13).

Четыре продукта построены на платформе JIRA: JIRA Software, JIRA Service Desk, JIRA Ops и JIRA Core. Каждый продукт поставляется со встроенными шаблонами для различных случаев использования и легко интегрируется, поэтому команды из разных организаций могут работать лучше вместе.

Рассмотрим основные характеристики JIRA Software.

Рабочие процессы представляют собой последовательный путь от зарождения идеи до ее реализации.

Схема базового рабочего процесса показана на рис. 1.14.

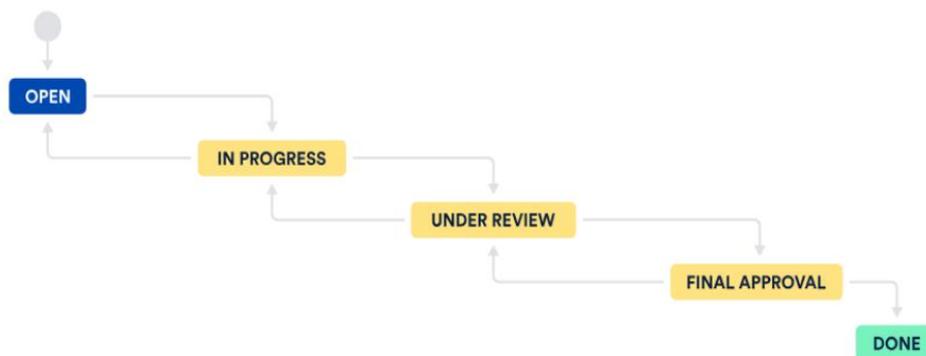


Рисунок 1.14 - Jira Workflow

В этом случае Open, Done и метки между ними отражают статус, в котором может возникнуть проблема, а стрелки - потенциальные переходы от одного состояния к другому.

Рабочие процессы могут быть простыми или сложными, с условиями, триггерами, валидаторами и функциями сообщения.

Рекомендуется для администраторов, начинающих использовать JIRA Software, сохранять максимально простой вариант своих рабочих процессов до тех пор, пока при реализации проекта не возникнет необходимость в усложнении конфигурации рабочего процесса [38].

Таблица 1.5 содержит сведения о важных характеристиках продукта Jira Software, а именно:

- назначение;
- потенциальных пользователей;
- сфере применения;
- возможности интегрирования с другими системами;
- доступных опциях для хостинга.

Таблица 1.5 — Обзор JIRA Software

Назначение продукта	Планирование, отслеживание и выпуск программного обеспечения мирового уровня.
Users	Software developers Project managers SCRUM masters
Use Cases	Отслеживание ошибок Управление проектами Менеджмент продукта Управление процессами Управление задачами Разработка программного обеспечения Agile разработка программного обеспечения
Интеграция	Confluence Bitbucket Slack GitHub
Hosting options	Cloud, Server, Data Center

Рассмотрим *положительные стороны* использования JIRA Software:

- хорошая видимость (visibility). Одним из факторов, замедляющих осуществление любого проекта (не только в области разработки программного обеспечения, но и в других профессиональных сферах, да и просто в жизни), является отсутствие четкой постановки задач и выстраивания иерархии их выполнения. JIRA устраняет эту проблему, поскольку она объединяет команды таким образом, что все члены команды получают возможность видеть продвижение в выполнении задач другими сотрудниками в режиме реального времени, так как задачи имеют теги «started» и «completed». Это помогает всем членам команды знать, на какой стадии находится проект, что ускоряет работу над программным продуктом;
- удобное определение приоритетов. Еще одно преимущество использования JIRA заключается в том, что она позволяет лучше определить порядок приоритетов задач, стоящих перед всеми членами команды, вследствие чего можно увидеть, какие задачи необходимо выполнить немедленно и какие

могут быть решены позже. Это очень полезно с точки зрения соблюдения сроков, особенно при работе над различными проектами с разными датами завершения;

- повышение производительности. Используя JIRA, члены команды получают возможность в любой момент времени видеть последовательность задач в списке (backlog), вследствие чего уменьшается время простоя, потраченное на обсуждение текущих проблем, и повышается производительность работы. Хотя время простоев может показаться незначительным, тем не менее общее их количество может привести к превышению времени, отведенного на выполнение задачи. Таким образом, JIRA способствует устранению этой проблемы и повышению производительности труда;

- обеспечение непрерывности взаимодействия между членами команды, где бы они ни находились. Еще одним преимуществом программного обеспечения JIRA является то, что JIRA поставляется с доступными мобильными приложениями. Это означает, что все члены команды могут оставаться на связи не только в офисе или дома через ноутбук, но и используя мобильные телефоны и планшеты;

- более 1000 плагинов. JIRA поставляется с более чем 1000 приложениями, которые помогут сделать программное обеспечение еще более полезным для команд гибкой разработки, два наиболее популярных из них - GreenHopper и Bonfire [36].

Рассмотрим *недостатки* использования JIRA:

- большая насыщенность функциональными возможностями, что создает дополнительные трудности не только для начинающих пользователей, но и для продвинутых, так как процесс кастомизации продукта оказывается достаточно трудоемким в силу большого количества настраиваемых параметров системы. Таким образом, требуется некоторое время, чтобы понять архитектуру программного обеспечения;

- сложный UX/UI. Обилие визуальной информации, а также наличие не всегда интуитивно понятных элементов интерфейса создает пользователю трудности взаимодействия с JIRA;

- требование наличия навыков DevOps, знания Scrum, уметь строить спринт для создания структуры работы;

- отсутствие встроенной функциональности управления календарем;

- высокая стоимость, которая составляет для команды до 10 человек 10 долларов США в месяц, до 100 человек - 7 долларов США за каждого члена в месяц.

Для лучшей организации совместной работы вместе с Jira в IT-компаниях обычно используют Confluence, так как это продукты одной компании Atlassian и они хорошо интегрированы друг с другом. Рассмотрим достоинства и недостатки работы с Confluence.

III. 1. Confluence – платформа, позволяющая публиковать web-страницы и документов в стиле wiki, а также обмениваться контентом между участниками команды и организовать обсуждение.

Назначение Confluence:

- централизация информационных потоков;
- полное обслуживание файловой системы и документации (создание, хранение, просмотр, обмен, редактирование);

- функция поиска документов или другой рабочей информации, например, писем;

- встроенный корпоративный календарь;

- постановка задач и контроль процесса исполнения;

- корпоративные чаты.

Среди недостатков данного программного продукта пользователи указывают следующие:

- неудобная система комментариев;

- плохо организованная система одновременного редактирования текстов;
- не налажено отслеживание развития проекта;
- неудобный тайм-менеджмент;
- отсутствует управление ресурсами;
- система ценообразования; цена на продукт растет с ростом количества пользователей [30].

Как видно, возможности Confluence достаточно большие. Разработчики постарались учесть все потребности работы в офисе.

Выводы к главе 1

На основании рассмотренных моделей жизненных циклов проекта, моделей и средств управления проектами можно сделать следующие выводы. Модель жизненного цикла проекта CRISP-DM наиболее адекватно отражает процессы разработки нейронной сети. Для управления процессами разработки программного продукта с использованием ИИ наиболее оптимальным будет использование комбинации гибкой модели и модели CRISP-DM.

Гибкая модель является наиболее подходящей для IT-компаний, стартапов, проектов в инновационных сферах, модель CRISP-DM позволяет управлять деятельностью отдела R&D; каскадная модель удобна в проектах, где ключевым ограничителем является срок реализации проекта, а не финансы.

Параллельное использование данных методологий позволит оптимизировать процессы и синхронизировать принципиально разную деятельность команд, работающих над созданием одного продукта.

Глава 2 МЕТОДЫ И АЛГОРИТМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ В КОМПАНИИ ООО «МАСТЕР МАИНД ИНК»

2.1 Алгоритмы процесса разработки программных продуктов и методы управления проектами на предприятии ООО «Мастер Маинд Инк»

Компания ООО «Мастер Маинд Инк» - молодая продуктовая компания на рынке IT-услуг. Она была образована в Харькове в 2017 в связи с ростом клиентского спроса на интеллектуальный продукт, то есть платформы и сервисы, созданные на основе искусственного интеллекта (ИИ) и машинного обучения (МО, Machine Learning).

Портфолио данной компании за сравнительно небольшое время ее существования составили такие решенные задачи:

- рекомендательная система для покупки сопутствующих товаров (заказчик - сеть украинских аптек);
- аналитический сервис, выполняющий прогноз потенциального оттока клиентов для предприятий, работающих в сфере обслуживания (сети фитнес-клубов, спа-салонов и прочее);
- платформа для увеличения продаж в электронной коммерции.

В настоящее время компания работает над проектом для сферы банковского и биржевого бизнеса.

Все эти программные продукты используют методы искусственного интеллекта и машинного обучения. Так, например, упомянутая платформа для увеличения продаж в электронной коммерции представляет собой самообучающийся ИИ, который использует данные о поведении пользователей и историю их покупок, чтобы стать умнее. Его основой является разработанная так называемая «система сознательных рекомендаций» (CRS), построенная с помощью оригинальных алгоритмов машинного обучения. CRS помогает идентифицировать покупки, объединенные одной целью, и предлагает купить нужный товар в нужное время с учетом глубокого анализа пользовательских предпочтений. Такие товары покупаются с максимальной вероятностью.

Процесс изготовления прикладного программного обеспечения, использующего ИИ, подразумевает более сложную структуру по сравнению с процессом изготовления типичного программного продукта.

Ниже представлена укрупненная блок-схема алгоритма создания любого программного продукта - как содержащего искусственный интеллект, так и стандартного (рис. 2.1).

Данный процесс включает в себя пять стадий: от анализа условий поставленной задачи до сопровождения готового изделия в процессе его эксплуатации заказчиком.

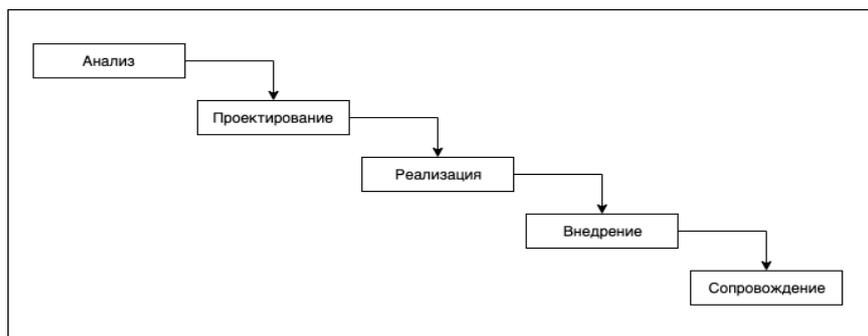


Рисунок 2.1 — Блок-схема рабочего процесса разработки программных продуктов

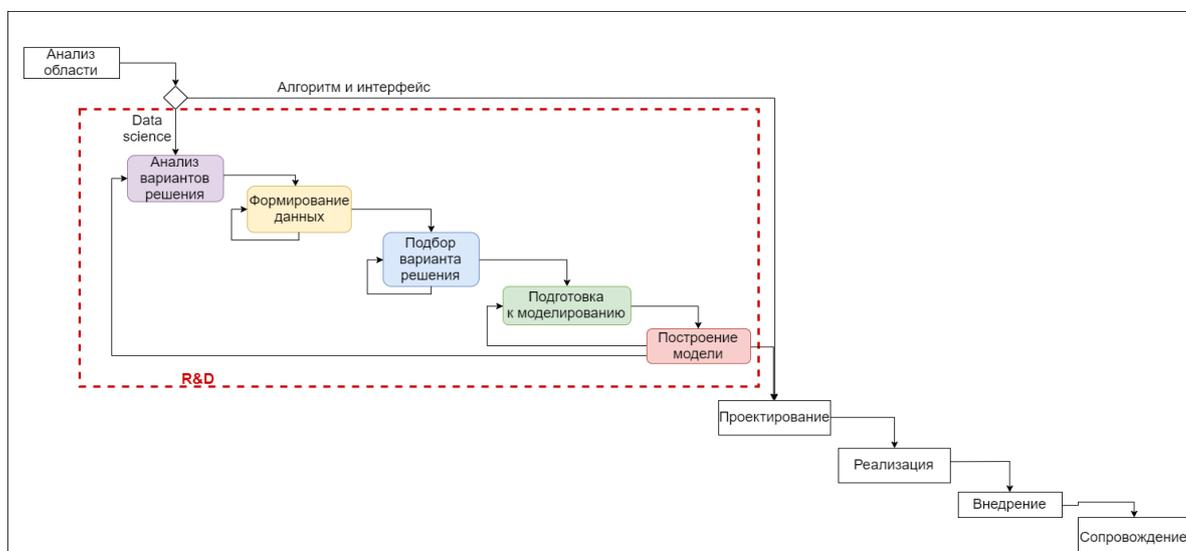


Рисунок 2.2 — Блок-схема рабочего процесса разработки программного продукта, содержащего ИИ

Однако более подробная детализация блок-схемы (рис. 2.2) в случае работы над созданием сервисов, программ, платформ с использованием ИИ (например, нейросетей) позволяет выделить в рабочем процессе целый этап, который предшествует непосредственно проектированию программного решения - исследовательскую деятельность (Research and Development - R&D), на рисунке 2.2 обозначен красным.

Такой тип деятельности присущ Data Science-проектам и направлен на исследование возможных вариантов решения поставленной задачи, а также разработку и обучение модели ИИ для каждого из вариантов.

Особенностью такой деятельности является то, что после этапа анализа предметной области невозможно однозначно переходить к этапу проектирования конечного программного продукта (как это происходит при решении стандартных задач), так как выходные данные, полученные после этапа анализа, могут быть многократно пересмотрены и уточнены.

Уточненный алгоритм этапа R&D представлен на рисунке 2.3.

Как показано на рисунке 2.3, в отличие от процесса реализации традиционных решений, жизненный цикл ИИ-проектов усложняется необходимостью проведения ряда подготовительных работ, которые имеют циклический характер. Количество этих циклов заранее неизвестно и зависит как от степени сложности поставленной задачи, так и от квалификации сотрудников на проекте и уровня развития их творческого мышления.

Более того, возможен исход, при котором детальное изучение поставленного задания показывает, что оно не имеет решения.

Узел принятия решения, обозначенный *1, отражает тот момент исследования, когда команда, основываясь на своем опыте и интуиции, выбирает направление, связанное либо с изменением параметров модели, либо с изменением «features», либо с поиском нового варианта решения поставленной задачи.

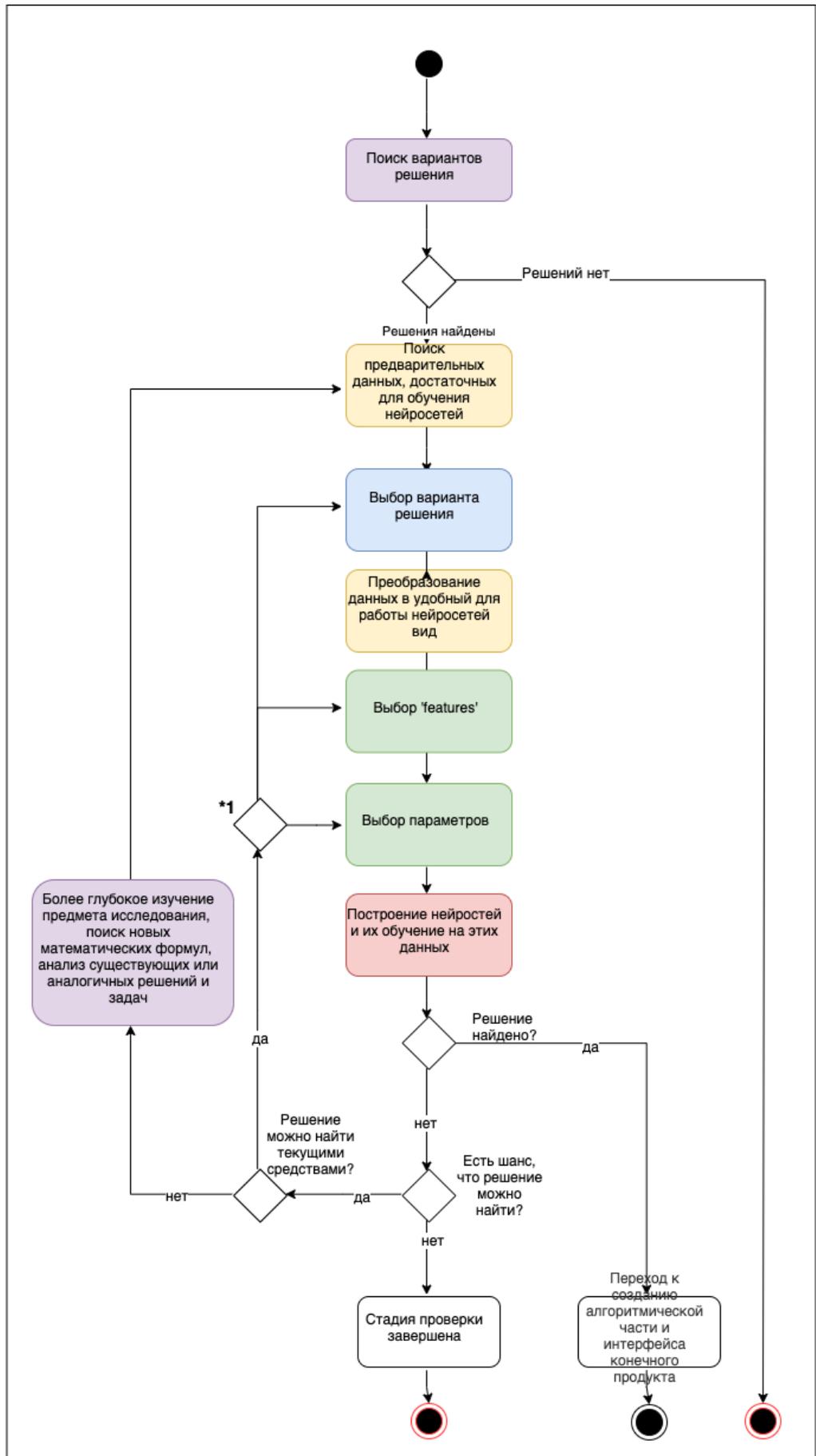


Рисунок 2.3 - Блок-схема алгоритма R&D

Циклический характер процесса создания ИИ-продукта существенно отличается от Agile-моделей (Scrum, Kanban и др.), основу которых составляют итерации, на каждой из которых мы получаем некоторую законченную часть целевого программного продукта.

В случае с ИИ-проектами завершение каждого цикла не всегда означает создание части или функции конечного продукта; напротив, нередко бывает, что исполнители проекта вынуждены вернуться к точке нулевого отсчета и начать заново работу с исходной информацией.

Кроме того, деятельность R&D основывается на работе с большими массивами данных, причем на протяжении всего жизненного цикла проекта.

Это ключевое обстоятельство вынудило специалистов Data Science искать специализированные методы работы с Big Data.

На данный момент в компаниях, специализирующихся на создании Data Science-продуктов, наиболее распространенной является CRISP-DM - методология по исследованию данных.

Таким образом, деятельность команды в процессе создания ИИ-продукта, состоит из исследовательских действий, которые всегда носят уникальный, оригинальный характер, и стандартных действий по проектированию, реализации и внедрению.

В течение всего процесса менеджмент компании ООО «Мастер Маинд Инк» для управления проектом использует Jira, а ведение документации выполняется с помощью Confluence.

Эти инструменты содержат весь необходимый функционал для управления задачами, персоналом и документацией на стандартных проектах, и идеально подходят для управления этапами проектирования и реализации, но не достаточны для управления исследовательской деятельностью на этапе R&D. Как показало проведенное автором изучение особенностей управления проектами в компании, на этапе R&D возможности Jira используются частично, фрагментарно и неэффективно.

2.2 Организация процесса исследовательской деятельности на предприятии ООО «Мастер Маинд Инк»

Рассмотрим организационную структуру компании ООО «Мастер Маинд Инк». В компании работает 15 человек, распределенных между такими отделами: финансовый -1 чел., отдел DevOps - 2 чел., специалисты по данным (Data Engineering) - 2 чел., отдел R&D - 6 чел., технический директор СТО, исполнительный директор CEO, Product Owner; разработчик сайтов готового продукта - 1 чел.

Сотрудники отдела R&D владеют следующими знаниями и умениями.

Высшее образование по прикладной математике, алгоритмы и структуры данных, Machine Learning, финансово-экономические знания, языки программирования и фреймворки Python, Flask, Docker Swarm, Kubernetes, TensorFlow, Cuda, C++, NodeJS, ClickHouse, MySQL, Redis, Kafka, R language, Bash script.

Специалисты Data Engineering обладают следующими знаниями и умениями: высшее техническое образование, алгоритмы и структуры данных, языки программирования и фреймворки C++, Cuda, MySQL, MSSQL, PostgreSQL, ClickHouse, MongoDB, RabbitMQ, Redis, Kafka, NodeJS, Bashscript.

Профессиональные знания и умения сотрудников отдела DevOps следующие: высшее математическое образование, Docker Swarm, Kubernetes, Kibana, Grafana, Git, NginX, Lua, Bash Script.

Директора компании имеют высшее техническое образование, владеют некоторыми из перечисленных языков программирования и фреймворков.

Начальник R&D-отдела имеет степень PhD по математике.

Кроме того, технический директор выполняет обязанности архитектора проектов, а также непосредственно руководит отделами.

Исполнительный директор CEO совмещает обязанности Project Manager.

Его задачи следующие:

- постановка заданий для проектной команды;
- назначение исполнителей для конкретных задач;

- отслеживание выполнения;
- создание проектной документации и т.д.

R&D-процесс подразумевает большую вовлеченность проектной команды в задачи, которые по Agile-методологиям (и другим распространенным общепринятым методологиям) выполняет проектный менеджер.

Как видно из приведенных сведений, сотрудники компании представляют собой высококвалифицированных специалистов, и обладают перекрестными умениями, что является необходимым условием работы в сфере производства ИИ-продуктов.

В процессе работы над программным продуктом все сотрудники, включая менеджмент, тесно связаны между собой, и поддерживают постоянную связь с заказчиком. Каждое утро проводятся стендап-митинги, на которых обсуждаются результаты прошедшего дня и планируются задачи на текущий день.

При этом также принимаются решения об отклонении промежуточных задач или изменении постановки задачи, если не удалось найти ее решение. Каждую неделю проводятся брейн-штормы, где на повестку выносятся задачи, выполнение которых кажется невозможным. В обсуждении поиска возможных вариантов решения принимает участие весь состав компании, в особо затруднительных случаях могут привлекаться сторонние специалисты.

Следует заметить, что результаты проводимых митингов практически не документируются.

Возникающие в процессе обсуждения мысли, идеи, выводы, предложения записываются на листочках, с помощью стикеров на доске, а порой вообще не фиксируются. Пока компания небольшая, подобная практика не является катастрофичной, но тем не менее на сегодняшний день существует большое упущение в процессах управления проектами.

Схема организационной структуры предприятия в общем виде представлена на рисунке 2.4.

подробная детализация усложняет схему исследовательской деятельности, выделение четких этапов (как это было сделано в рисунке 2.2), становится весьма затруднительным, так как реальный, физический процесс создания программного продукта является нелинейным, ситуативным, а его подпроцессы могут быть параллельны или пересекаться.

Процесс создания и обучения нейросети распределен между тремя участвующими сторонами: Product Owner (лицо, принимающее решение со стороны заказчика), специалисты по Data Science и инженеры по данным (Data Engineer). Как уже отмечалось, бывают случаи, когда исполнители меняются ролями и временно принимают на себя обязанности другого члена команды.

Проследим организационные связи между отделами и сотрудниками на предприятии ООО «Мастер Маинд Инк» на каждом этапе алгоритма R&D.

Как было ранее показано на рисунке 2.3, R&D-деятельности предшествует анализ предметной области. Результатами этого анализа должны быть:

- определение цели продукта;
- определение бизнес-требований;
- определение KPI для каждого бизнес-требования;
- определение возможных способов удовлетворения бизнес-требований,

а именно: определение проектной командой стратегии решения каждого бизнес-требования - с помощью построения ИИ или алгоритмических методов.

Предварительная ориентировочная деятельность в проектах с ИИ имеет значение большее, чем в любых других видах проектов. На этой стадии происходит формулирование критериев, которым в дальнейшем будет соответствовать готовый продукт.

Реализация любого проекта всегда начинается с формулирования потребности заказчика/рынка в нем. Product Owner должен иметь представление о том, как этот продукт или его новая функция должны работать в конечном итоге. Например, внедрение некоторого сервиса предотвратит отток посетителей или простимулирует продажи других продуктов.

Формулировка такой потребности не претендует на полное описание проекта, она только обозначает проблему: «наши клиенты должны понять, как они распоряжаются своим бюджетом», «если мы добавим функцию предсказания часа пик в аэропорту, то клиенты охотнее будут покупать приложение».

Здесь необходимо изложить цели продукта, на основании которых формируются бизнес-требования, т.е. набор той функциональности, которая определяет назначение ПО.

На этом этапе все участники проекта предлагают идеи, которые можно использовать при решении сформулированной проблемы, происходит создание некоторых первоначальных набросков возможного решения. Этим этапом обычно руководит специалист Data Science, но важно, чтобы принимали участие абсолютно все участники работы на проекте, включая менеджмент, так как возможные решения и продуктивные идеи могут лежать в совершенно различных областях знаний.

В результате этого этапа обычно становится ясно, насколько глубоким будет процесс исследования данных.

Затем следует определение количественных показателей, которыми будет определяться, достигнуто ли бизнес-требование. Этот этап заключается в совместном определении объема и ключевых показателей эффективности проекта (KPI). Затем эти KPI должны уточнены и переведены в измеримые. Возможно, удастся получить очень четкие показатели, такие как *«прогнозирование ожидаемого CTR объявления с приближением не менее X% в по меньшей мере Y% случаев, для любого объявления, показанного не менее недели, и для любого клиента с более чем двумя месяцами временных данных»*. Однако в некоторых случаях необходимо использовать качественные показатели, например, *«время, необходимое для изучения темы с использованием сгенерированных расширенных запросов, будет сокращено и/или улучшится качество результатов по сравнению с исходными запросами»*.

Это особенно верно, когда модель предназначена для оказания помощи человеку в какой-то сложной его деятельности.

Ключевые показатели могут быть уточнены в связи с имеющимися временными рамками и ресурсами. Для их уточнения нужен дальнейший сбор данных.

Обычно эти метрики (KPI) определяются проектной командой совместно с Product Owner в точных количественных показателях. Но в зависимости от ресурсов и временных ограничений они могут быть подвергнуты пересмотру после консультаций с заказчиком. Поскольку любая обратная связь означает дополнительный расход времени, то проектная команда может попытаться найти дополнительные жесткие метрики, на которые можно опереться для продолжения работы над продуктом, и избежать дополнительных консультаций с заказчиком.

Определение объема работ на этом этапе особенно важно, поскольку исследовательские проекты имеют тенденцию удлиняться и естественным образом увеличиваться в размерах и масштабах по мере появления новых возможностей при проведении исследований.

Процесс определения значений KPI, является, таким образом, итерационным. Он завершается тогда, когда найденные значения удовлетворят клиента. Здесь был замечен следующий факт, отражающий некоторый конфликт бизнес-интересов заказчика и тех возможностей программного продукта, как их оценивает проектная команда. Бизнес всегда стремится иметь максимальные показатели KPI. Разработчики утверждают, что программный продукт объективно не в состоянии (как любая модель) отразить реальную ситуацию (в предметной области) на 100%. Например, если система предсказывает поведение покупателей с точностью, меньше 95%, то такая система, по мнению большинства заказчиков, является бесполезной.

Однако во многих случаях тщательный анализ и обсуждение предположений о продукте могут привести к появлению очень ценных

решений, которые могут оказаться не такими технически сложными, как казалось на первой итерации выработки KPI.

Определить объем работ на данном этапе также является важным с психологической точки зрения. Исследователь, который начинает изучать полученную задачу, стремится изучить как можно большее количество статей и книг. Поэтому отсутствие четко очерченных границ времени, ресурсов и целей проекта может привести к долгим неделям или месяцам, потраченным на разработку прекрасных моделей, которые в конечном итоге не отвечают реальной потребности, или приводят к неучету некоторых KPI, которые могли быть определены ранее.

Анализ решения (обозначен сиреневым цветом на рисунках 2.2, 2.3 и 2.5).

На данном этапе перед проектной командой стоят задачи изучения литературы и существующих готовых решений.

Исследователь анализирует как академическую литературу, так и существующие программные решения и инструменты.

Глубина погружения в литературу зависит от особенностей проекта: является ли он фундаментальным, определяющим дальнейшую стратегию компании или будущий программный продукт предназначен для решения разовой проблемы. Также необходимо ответить на вопросы: планируется ли публикация результатов исследования по этому вопросу в академических изданиях; планирует ли исследователь стать экспертом команды по этой теме и т. п.

Например, проект предназначен для того, *чтобы помочь отделу продаж лучше прогнозировать отток клиентов и убытки от оттока. Для решения поставленной задачи требуется хорошее знание теории случайных процессов, на которой строятся многие общие решения подобных проблем.*

Необходимая глубина понимания зависит от технических аспектов проблемы. Некоторые из аспектов могут быть предвидены заранее, а некоторые обнаружены только позже.

Например, если разрабатываемый продукт должен быть интегрирован в готовую программную среду, написанную на языках Java и Scala, то приходится использовать те технологии, которые применяются в готовом продукте. И если разработчики не владеют этими языками, то их придется изучить.

При изучении литературы специалист Data Science должен подготовить для обсуждения в команде не только выбранные варианты решения проблемы, но сделать краткий обзор всей области и всех рассмотренных решений, объясняя преимущества и недостатки каждого решения и обосновать свой выбор.

После того, как найден подходящий математический аппарат, найденные направления решений необходимо оценить с точки зрения способа реализации и сложности этого способа в производстве.

Цели продукта и бизнес-требования к нему, а также структура и характеристики предлагаемых вариантов решений служат обоснованием для выбора способа хранения и обработки данных, способность к масштабированию по горизонтали и вертикали (scalability) и приблизительную оценку стоимости проекта.

Это важная проверка, которую необходимо выполнить на этом этапе, поскольку обработка данных и разработка программного обеспечения могут начинаться параллельно с разработкой модели. Кроме того, предлагаемое решение может оказаться неадекватным или слишком дорогостоящим с технической точки зрения, и в этом случае это должно быть выявлено и устранено как можно скорее. Когда технические вопросы рассматриваются до начала разработки модели, знания, полученные на этапе исследования, могут затем использоваться для предложения альтернативного решения, которое могло бы лучше соответствовать техническим ограничениям. Это еще одна причина, по которой этап исследования должен закончиться выработкой спектра решений, а не ограничиваться одним.

Подготовка данных и выбор решения (синий и желтый цвет на рисунках 2.2, 2.3, 2.5);

На этом этапе проектная команда определяет набор данных, которые потребуются в дальнейшем для обучения нейронных сетей; сбор первичных данных, обработка их (очистка от мусорной информации), преобразование в требуемый формат. Хранятся подготовленные данные вместе с запросами в базах данных компании.

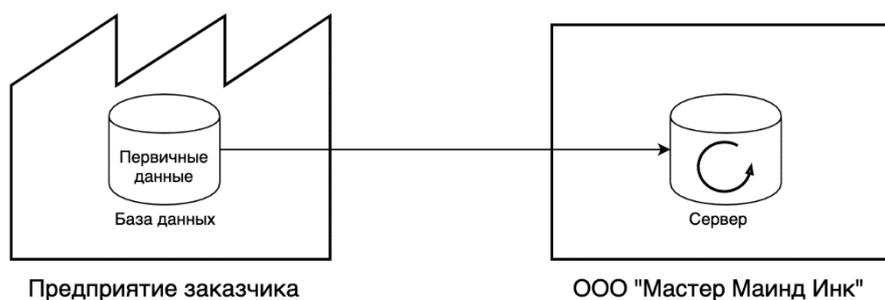


Рисунок 2.6 – Схема взаимодействия баз данных

Процесс подготовки данных часто сопровождается выгрузкой больших массивов данных (Big Data) из баз данных заказчика на сервер компании (если позволяют условия конфиденциальности). В дальнейшем происходит обработка Big Data в формат, удобный для быстрых запросов и сложных вычислений. Эти действия необходимы для начала процесса исследования данных и иногда занимают больше времени, чем ожидалось.

Далее начинается процесс исследования данных.

Теперь специалист Data Science может оперировать фактическими жесткими KPI и показателями модели.

Однако и здесь надо соблюдать динамическое равновесие между исследованием и разработкой; даже имея в виду четкие KPI, полезно не терять из виду некоторые, казалось бы, неподходящие направления решения.

Обработанные данные должны иметь формат, удобный для работы отдела Data Engineering. Однако могут быть выявлены некоторые недостатки сформированного рабочего массива данных (например, данных,

предоставленных заказчиком, может оказаться недостаточно). Data Engineer данных должен быть готов к такой ситуации, и искать дополнительный источник данных.

Следует заметить, что, хоть процесс подготовки данных и выглядит отделенным от процесса теоретического исследования и анализа вариантов решений, они обычно либо выполняются параллельно, либо чередуются между собой.

Подготовка к моделированию (зеленый цвет на рисунках 2.2, 2.3, 2.5).

Подготовка к началу разработки модели в значительной степени зависит от имеющегося технического обеспечения и объема технической поддержки, доступной Data Scientist. Возможно, что Data Scientist создаст новый репозиторий кода и запустит локальный сервер Jupyter Notebook или запросит более мощную облачную машину для выполнения вычислений.

Также может быть, что сотрудники отдела Data Science создадут код для управления версиями данных и моделей или для отслеживания и управления экспериментом. Возможно, что такая функциональность уже существует в используемых сервисах, то тогда потребуется создание некоторых настроек для распределения ресурсов или настройки пользовательских пакетов и т. д.

Этот этап связан не только с технической подготовкой, но и подготовкой ‘features’ и параметров для будущей нейросети.

Разработка и обучение модели (красный цвет на рисунках 2.2, 2.3, 2.5).

Задачами этого этапа являются развертывание модели, обучение (Machine Learning) и непрерывный мониторинг результатов обучения.

В подготовленной на предыдущем этапе среде с выбранными параметрами создается модель нейросети. Специалисты отдела Data Engineering передают в отдел Data Science наборы данных и созданные к ним запросы, подготовленные в соответствии с набором ‘features’. Отдел Data Science загружает эти данные в обучающую среду, которая использует эти данные для обучения модели. Специалисты отдела Data science осуществляют мониторинг

результатов обучения, используя функциональность сервисов. После того, как модель считается обученной, можно переходить к модельному тесту.

При разработке и обучении модели различные ее версии (и обслуживающий массив данных) должны постоянно проверяться на соответствие заранее установленным KPI.

Это дает возможность оценить степень продвижения в обучении, а также позволяет отделу Data Science решить, когда модель работает достаточно хорошо, чтобы гарантировать полное соответствие выработанным на начальном этапе KPI.

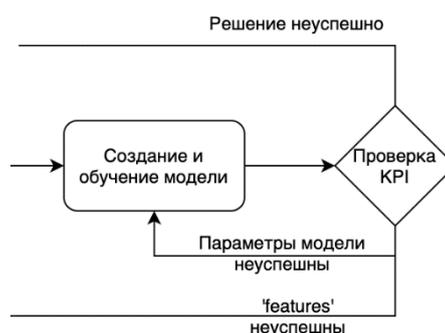


Рисунок 2.7 - Алгоритм принятия решения об успешности модели

Если результаты обучения модели не соответствуют ожидаемым показателям, то существует несколько причин такого исхода.

Специалист отдела Data Science анализирует результаты обучения нейронной сети и принимает решение о дальнейших действиях: изменить параметры, изменить «features», или данный вариант модели не пригоден для решения задачи и надо переходить к выбору следующего варианта модели.

Другим возможным результатом неудачного обучения является пересмотр цели продукта. В редких случаях изменение цели влечет за собой незначительные изменения в технической реализации проекта. Но в общем, это означает, что мы возвращаемся на фазу анализа предметной области.

Не исключен наиболее экстремальный вариант - отмена проекта; если специалист по данным уверен, что все направления исследований были

изучены, а проектный менеджер уверен, что продукт исчерпал свой потенциал, возможно, пришло время перейти к другому проекту.

Если обучение модели прошло успешно и она реализует заданные KPI, то этап R&D можно считать успешно законченным и переходить в разработке алгоритмической части конечного продукта, а также созданию интерфейса пользователя.

В ситуациях, когда расхождение между полученными и заданными значениями KPI не является критичным, требуется более тщательный их пересмотр и согласование с Product Owner.

Необходимо убедиться, что текущее значение показателя находится в интервале приемлемого разброса. При этом необходимо учитывать требования к продукту и интересы клиента.

Когда Product Owner убежден, что модель отвечает заявленным целям проекта (в удовлетворительной степени), команда может приступить к его производству программного продукта.

На основании выполненного анализа организации процесса исследовательской деятельности в компании ООО «Мастер Маинд Инк» можно сделать вывод о том, что структура этого процесса не является линейной. Развертка Agile-модели жизненного цикла проекта имеет линейную архитектуру, как представлено на рисунке 2.8.

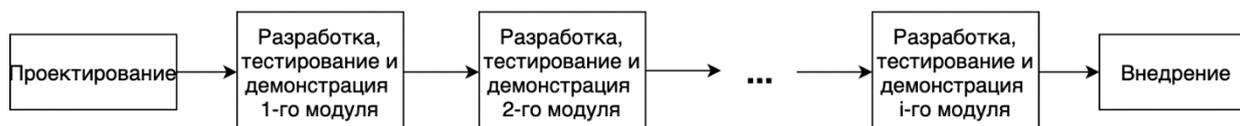


Рисунок 2.8 - Развертка Agile-модели жизненного цикла проекта

Модели Crisp DM, в отличие от Agile-моделей, можно представить в виде дерева решений, где каждая ветка отображает новый цикл разработки.

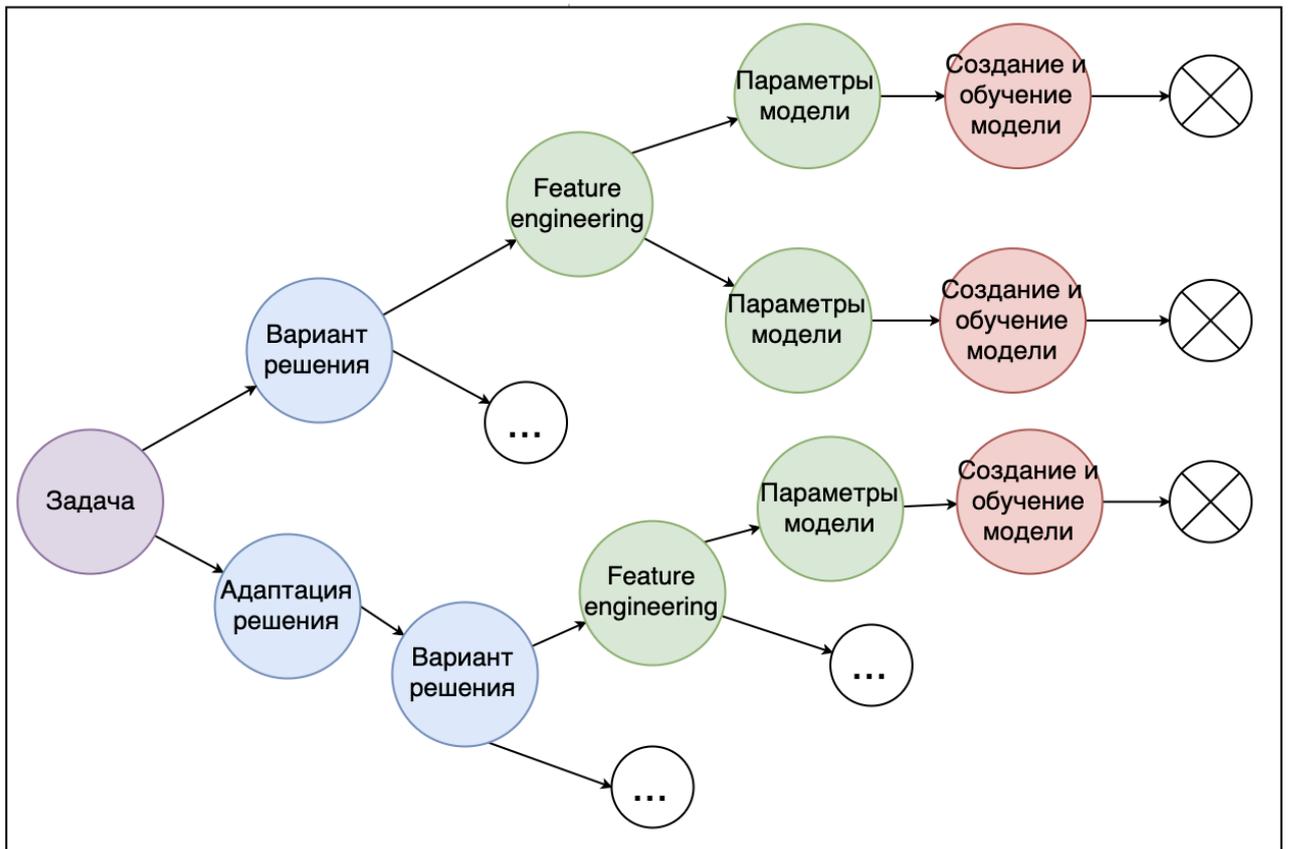


Рисунок 2.9 - Дерево решений модели Crisp DM

Корневым узлом дерева являются определенные границы проекта в виде целей, бизнес-требований и набора KPI. Узлами второго уровня можно представить найденные или адаптированные возможные решения поставленной задачи. Далее каждый узел имеет ветвление на разработанные 'features'. Из данных узлов расходятся ветви выбранных для параметров.

Каждый узел принадлежит одному из рассмотренных ранее фаз этапа R&D.

2.3 Модель системы управления проектами «As Is»

Выполненный анализ процесса создания программного ИИ-содержащего продукта, а также организационной структуры процесса в компании ООО «Мастер Маинд Инк» позволяет построить модель «As Is» («Как есть») процессов управления в компании.

Модель изображена на рисунке 2.10 приложения Б. Она отображает управленческие действия проектного менеджера в инструментах Jira и Confluence, а также действия остальных сотрудников компании, принимающих на себя функции Project Manager в соответствии с текущими потребностями проекта. Построенная модель отражает процессы управления, сопровождающие все фазы R&D-деятельности.

Процессы распределены между Project Manager, отделом Data Science и отделом Data Engineering.

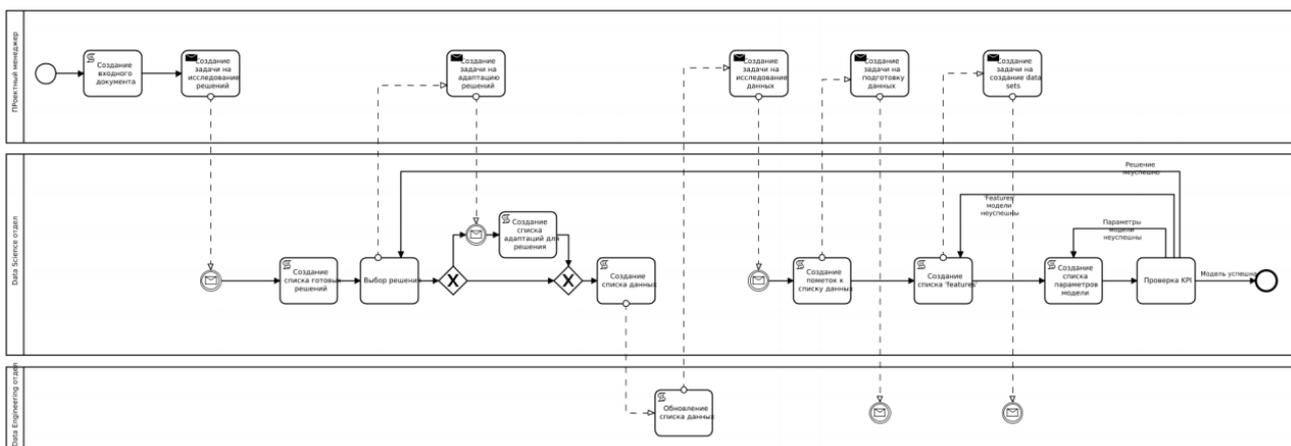


Рисунок 2.10 – Модель управления процессами «As Is»

С точки зрения управления данный R&D этап можно представить следующим образом.

Рассмотрим подробнее представленную модель процесса управления проектом, функционирующую в компании ООО «Мастер Маинд Инк».

1) создание входного документа. Этот процесс принадлежит фазе анализа предметной области. Здесь команда собирается на совещание и записывает на листочке или доске все принятые на совещании решения, а именно: список целей, список бизнес-требований, список KPI, а также пометок к каждому бизнес-требованию, с помощью чего он решается (ИИ или алгоритм). После совещания РМ делает фотографию доски или листочка и переносит все записи в Confluence;

- 2) создание задачи на исследование решений. По окончании совещания РМ создает задачу в Jira, и закрепляет задачу за сотрудником Data Science-отдела. Сотрудник, получивший задание, приступает к ее выполнению;
- 3) создание списка готовых решений. После исследования сотрудник создает список найденных решений в виде презентации и делает доклад для проектной команды;
- 4) выбор решения. После прослушивания доклада команда обсуждает его и принимает решение, какие из представленных решений перспективны, какие надо отбросить, а какие оставить «про запас»;
- 5) создание задачи на адаптацию решения. Если ни одно из выбранных решений нельзя применить в готовом виде, то РМ создает задачу в Jira на адаптацию решения и закрепляет за сотрудником Data Science-отдела;
- 6) создание списка адаптаций для решения. Сотрудник Data Science-отдела, получивший и выполнивший задание, создает список адаптаций и заводит его в Confluence;
- 7) создание списка данных. После выбора варианта решения проектная команда собирается на совещание и создается список данных, необходимых для работы над проектом. Этот пункт также может выполнить сотрудник Data Science-отдела самостоятельно. Данный список создается отдельным документом в Confluence;
- 8) обновление списка данных. Сотрудник Data Engineering-отдела распечатывает документ, созданный в п.7 и занимается сбором данных. На распечатанном документе создаются пометки о том, какие данные удалось собрать и в каком объеме;
- 9) создание задачи на исследование данных. После успешного сбора данных РМ создает задачу в Jira для начальника Data Science-отдела;
- 10) создание пометок к списку данных. Начальник Data Science-отдела исследует данные и готовит документ с пометками к собранному списку данных. Данный документ создается в Confluence;

11) создание задачи на подготовку данных. РМ создает задачу в Jira для сотрудника Data Engineering-отдела на подготовку данных. Сотрудник Data Engineering-отдела получает задачу вместе с документом из п.10;

12) создание списка «Features». После создания пометок к списку данных сотрудники Data Science-отдела проводят совещание с целью определения списка «Features». Данный список заводится в Confluence одним из сотрудников Data Science-отдела;

13) создание задачи на подготовку наборов данных. После выполнения п. 12 РМ создает задачу в Jira для сотрудников Data Engineering-отдела с прикрепленным документом из п.12;

14) создание списка параметров модели. После выполнения задачи, поставленной в п.12, сотрудники Data Science-отдела создают список параметров для текущей модели;

15) проверка KPI. После создания и обучения модели с выбранными параметрами и выбранным списком «Features» сотрудники Data Science-отдела проверяют результаты обучения модели средствами Jupiter Notebook. После этого показатели текущего цикла не сохраняются в виде документа.

Как можно увидеть, проектный менеджер создает задачи с помощью Jira. Наибольшее количество информации, подлежащей документированию, производит работа отдела Data science.

Поскольку проектный менеджмент не входит в прямые обязанности исследователей и разработчиков, то документирование выполняется ими непрофессионально. Они попросту в большинстве случаев пренебрегают вопросами сохранности потенциально полезной информации.

В результате у проектного менеджера нет визуального отображения состояния дел в любой момент времени. Он может узнать о статусе выполнения задач и данные, касающиеся текущей ветви только в устной форме.

Выполненное исследование организации управления в ООО «Мастер Маинд Инк» позволило сделать следующие наблюдения. Большинство

документации не имеет четкую структуру и создается на усмотрение исполнителя.

Не вся документация ведется с помощью корпоративной системы Confluence. Некоторые из документов заводятся в личных аккаунтах сотрудников внутри других систем или создаются в бумажном виде.

Документы, созданные вне Confluence, не всегда переносятся в Confluence;

Многие документы вовсе не создаются, обмен рабочей информацией происходит или в устной форме, или в бумажном виде.

Выводы к главе 2

Изучение методов управления проектами, принятых в компании ООО «Мастер Маинд Инк», а также алгоритма R&D-деятельности позволили сделать следующие наблюдения и выводы:

Компания сталкивается с такими проблемами, как -

- неполная документация.

Эта проблема возникает в связи с тем, в связи с тем, что создание документации традиционно является кругом обязанностей менеджерского состава и бизнес-аналитиков. В условиях исследовательской деятельности документация является технической и проектный менеджер не в состоянии ее корректно заполнить без помощи (диктовки) технических специалистов. Поэтому разработчики создают ее самостоятельно. Но поскольку подобного рода обязанности не являются традиционными для них, необходимая документация формируется некачественно, и иной раз даже вообще не создается, так как исполнители не хотят тратить на нее время и полагаются на свою память. Таким образом, ценные мысли, идеи, творческие решения могут быть утеряны. По причине несохраненной документации в некоторых случаях проектной команде приходится тратить дополнительное время на повторное обсуждение задач, способов решения, а также на повторные исследования, мозговые штурмы и т. д.

- неунифицированный формат.

По причине, описанной выше, документация создается в разных форматах. Это затрудняет процесс ее совместного использования и т. д.

- слабая визуализация.

Дерево решений не визуализируется и нигде не сохраняется. Это может влиять на общее представление о стадиях проекта, а также затрудняет проектному менеджеру задачу отчетности заказчикам о проделанной работе;

- неполный учет времени.

Задачи повторяются от ветки к ветке, некоторые задачи являются незначительными (с точки зрения времени), проектная команда к компании небольшая и каждый знает свои обязанности, по этим причинам некоторые задачи могут не создаваться в инструменте управления проектами. Это затрудняет учет времени, так как в Jira будет храниться неполная информация о задачах. Также это может породить проблему «потери» некоторых задач.

Модель CRISP-DM является на данный момент единственной приемлемой для Data Science проектов, но она обладает определенными недостатками, которые влияют на процесс управления R&D этапом. К ним относят отсутствие средств поддержки.

Данная модель стала применяться недавно к подобного рода проектам. И сами Data Science проекты появились недавно. Поэтому существующие инструменты управления не успели еще адаптироваться под нужды Data Science-проектов, а новые не появились на рынке.

Отсутствие критерия времени. Модель не подразумевает учет времени или ограничение времени каждого цикла.

Глава 3 ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ В КОМПАНИИ ООО «МАСТЕР МАИНД ИНК»

3.1 Модель автоматизированной системы управления проектами на предприятии ООО «МАСТЕР МАИНД ИНК»

Изучение способа организации управления проектами в компании ООО «Мастер Маинд ИНК» показало, что на предприятии имеются существенные возможности для повышения эффективности процесса управления проектами. Во многом ручная модель управления, установившаяся в компании, не вполне адекватна процессам создания проектов, выполняемых командой.

Для организации работы проектной команды необходим единый информационный центр, с помощью которого можно решать следующие задачи управления:

- сохранять проектную документацию;
- вести протоколы митингов, стендапов, мозговых штурмов, производственных совещаний;
- хранить презентации докладов сотрудников;
- оповещать сотрудников о событиях и планах;
- создавать реестры задач, бизнес-процессов, собственных разработок;
- создавать собственную библиотеку готовых решений по профилю компании;
- создавать «запасники» интересных идей, методов, процедур, которые могут пригодиться в будущем;
- назначать задачи и раздавать поручения;
- осуществлять обратную связь по поставленным задачам и поручениям.

Цель данного этапа диссертационного исследования - разработать модель подобного информационного центра в виде автоматизированной системы управления проектами. Эта АСУ проектами должна быть интегрирована в

существующую систему управления проектами и за счет синергии повысить эффективность управления проектами, и, значит, работы компании в целом.

Анализ методов и алгоритмов процесса управления проектами в компании ООО «Мастер Маинд Инк» показал, что управление на этапе R&D существенно отличается от процесса управления на этапе исполнения программного продукта.

Моделируемая система должна быть основана на программных продуктах, которые на данный момент используются на предприятии ООО «Мастер Маинд Инк»: Jira и Confluence. Данный продукт не будет заменять текущие системы, а будет встроен в нее с целью автоматизации рутинных процессов. Продукт не является самостоятельной и независимой системой, а является плагином для системы Jira, поэтому должен быть адаптирован к работе с ней.

Существуют следующие методы адаптации программного обеспечения:

1) параметрическая адаптация, связанная с настройкой параметров программы. Это самый простой способ адаптации; он предполагает изменение значений характеристик, регулирующих работу программы. С помощью параметрической адаптации возможно проводить необходимую настройку функций и компонентов программы, и отбирать стратегии поведения из предлагаемого набора стратегий;

2) функциональная адаптация предполагает изменение функций программного обеспечения в установленных пределах. Функциональная адаптация может выполняться совместно с параметрической. Структура и организация ПО при этом остаются неизменными;

3) организационная адаптация - перенастройка потоков и процессов в системе. Организационная адаптация означает перераспределение потоков и процессов системы как ее внутренних ресурсов, без изменения ее структуры. Может сочетаться с функциональной адаптацией;

4) структурная адаптация - изменение структуры системы. Используя этот способ адаптации, мы производим модификацию или замену одних

структурных элементов системы, или алгоритмических модулей, на другие. Адаптируемая программа становится более адекватной решаемым задачам. При этом возможно использование всех предыдущих видов адаптации системы;

5) размножение - порождение себе подобных потомков. Адаптация размножением является эффективным методом адаптации. Система производит себе подобную, но обладающую наличием свободных ресурсов и способностью к изменениям.

б) развитие - направленный процесс эволюции систем. Адаптация программного обеспечения, происходящая через развитие, подобна процессам эволюции живых систем. Адаптируемое ПО проходит процесс эволюции через накопление информации о себе, внешней среде, решаемых задачах. Проходя через этапы зарождения, становления определенных качеств, устойчивого функционирования, деградацию и гибель, ПО как система становится более приспособленным для решения задач [3].

Моделируемая система управления проектами должна представлять собой отдельный функциональный блок, встраиваемый в систему Jira и Confluence, а также взаимодействующий с ними посредством их функциональных возможностей. Таким образом, существующая система управления проектами в компании ООО «Мастер Маинд Инк», сочетающая ручное и автоматизированное, несколько меняет свою структуру, организацию и функции.

Рассмотрим, как меняются функции. В модели «As Is» функции создания целого ряда документов и задач лежали на проектной команде и проектном менеджере. Посредством R&D-плагина эта работа будет выполняться автоматически, так как в разрабатываемую систему будет заложен алгоритм R&D-деятельности, благодаря чему шаблоны документов на каждом этапе процесса управления будут формироваться автоматически; задачи и поручения также будут генерироваться с помощью новой системы благодаря функции отслеживания действий проектной команды.

Таким образом, не нарушая установленных пределов изменения функций существующей системы управления, система «То Ве» расширяет спектр функций, используемых в команде.

Организационная адаптация системы управления проектами состоит в том, что прежние потоки управленческой информации меняют свою траекторию. Взаимодействие между проектным менеджером и отделами компании будет осуществляться в новой модели через единый информационный центр управления. При этом точки входа и выхода организационных потоков остаются прежними.

Также несколько меняется структура существующей системы управления проектами. Методы ручного управления перестают быть обязательными и в системе Jiga появляется новый элемент - R&D-плагин.

На основе структурно-функционально-организационного метода адаптации была разработана модель АСУ проектами на предприятии ООО «Мастер Маинд Инк». Модель представлена следующими диаграммами:

- верхнеуровневая диаграмма экосистемы «R&D-плагин»;
- диаграмма бизнес-процессов управления «То Ве»;
- диаграмма прецедентов;
- ER-диаграмма.

Верхнеуровневая диаграмма экосистемы «R&D-плагин» показана на рисунке 3.1.

Диаграмма экосистемы показывает, с какими уже существующими подсистемами предстоит взаимодействовать новому плагину, и отображает связи между ними.

Рисунок 3.2 изображает модель взаимодействия системы управления проектами «То Ве» («Как будет») с участниками этапа R&D.

Как мы можем видеть, в данной модели появляется еще один участник - R&D Plugin. Процессы, которые присутствовали в модели «As Is», перешли из пулов участников проекта в пул плагина, что означает, что выполнение этих процессов берет на себя плагин.

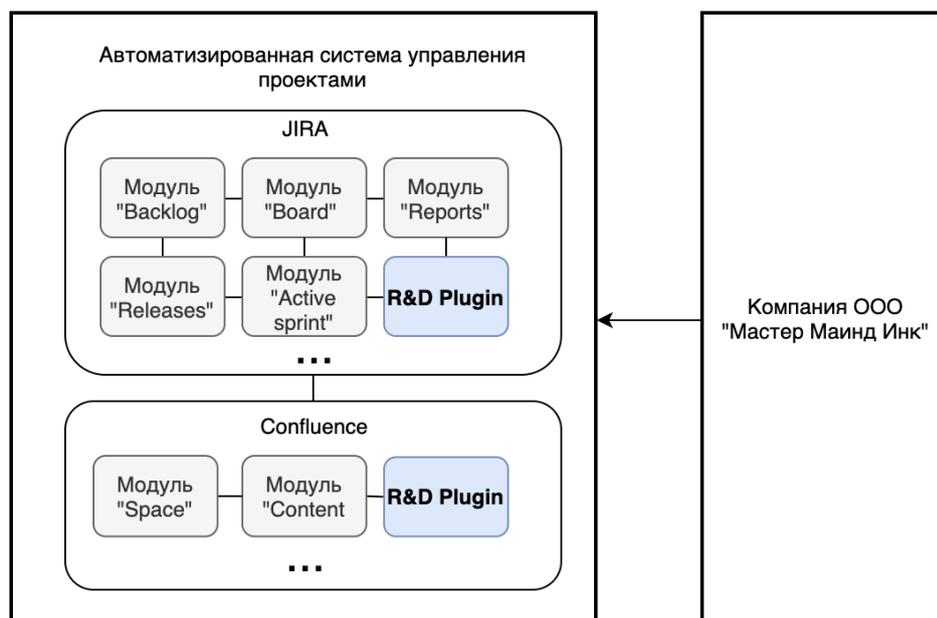


Рисунок 3.1 - Диаграмма экосистемы встраиваемого плагина для управления проектами на этапе R&D

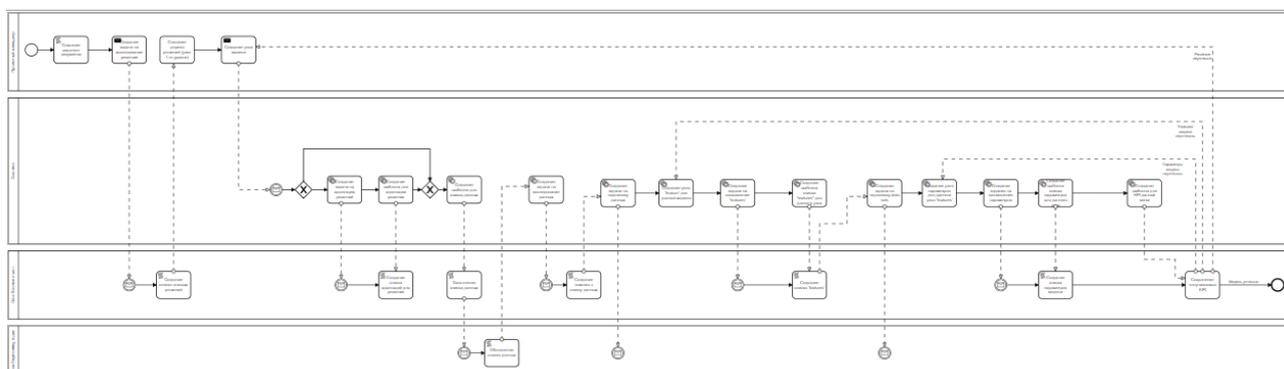


Рисунок 3.2 - Диаграмма бизнес-процессов модели «To Be» управления проектами на этапе R&D

Забирая на себя часть обязанностей остальных участников процесса, новый плагин может значительно сократить время, необходимое проектной команде на создание задач, документации и на отслеживание процесса разработки.

Рассмотрим подробнее предлагаемую диаграмму бизнес-процессов управления проектами на этапе R&D «To Be»:

1) создание входного документа. Для инициации создания нового проекта проектным менеджером плагин предлагает ему заполнить входной документ по установленному шаблону: заполнить список целей проекта, заполнить список бизнес-требований, к каждому бизнес-требованию добавить комментарий - создается ли проект с помощью ИИ либо с помощью алгоритмизированного подхода, а также создать набор KPI будущего продукта;

2) создание задачи на исследование решений. По окончании совещания PM заканчивает работу над созданием документа, проверяет заполненные данные и подтверждает создание нового проекта в системе Jira. Плагин создаёт новый проект в Jira, создает задачу на исследование решений, а также сохраняет протокол, заполненный PM в системе Confluence. Далее PM закрепляет задачу за сотрудником Data Science-отдела. Сотрудник, получивший задание, приступает к ее выполнению;

3) создание шаблона списка возможных вариантов решения. Плагин автоматически создает шаблон для найденных вариантов решения в системе Confluence. Данный шаблон в качестве ссылки будет прикреплен к задаче, созданной в п.2;

4) создание списка готовых решений. После пункта 3 сотрудник Data Science заполняет список найденных решений в документ. После совещания, на котором проектная команда будет обсуждать варианты решения, создатель списка сделает пометки, в которых расставит приоритеты для каждого найденного решения. Также помечаются решения, которые требуют адаптации;

5) создание дерева решений (узел 1-го уровня). На странице плагина проектным менеджером создается корневой узел будущего дерева решения;

6) создание узла модели. Проектный менеджер инициирует создание узла решения. АСУ создает узел решения соответственно выбранным в п.4 приоритетам и переходит к следующему этапу;

7) создание задачи на адаптацию решения. Если решение, соответствующее созданному в п. 6 узлу нельзя применить в готовом виде, то

плагин создает задачу в Jira на адаптацию решения. РМ может закрепить ее за конкретным сотрудником Data Science-отдела;

8) создание шаблона для адаптации решения. Если был выполнен, то плагин создает шаблон в системе Confluence для документа на адаптацию решения и прикрепляет его ссылкой к задаче, созданной в п.7;

9) создание списка адаптаций для решения. Сотрудник Data Science-отдела, получивший и выполнивший задание, заполняет список адаптаций и по созданному в Confluence шаблону;

10) создание шаблона для списка данных. После окончания п.6, если решение не требовало адаптаций, или после п.9, если требовало, плагин создает шаблон для списка данных, требуемых для текущего узла;

11) создание списка данных. После выбора варианта решения проектная команда собирается на совещание и создается список данных, необходимых для работы над проектом. Этот пункт также может выполнить сотрудник Data Science-отдела самостоятельно. Данный список заполняется по шаблону, созданному в Confluence;

12) обновление списка данных. Сотрудник Data Engineering-отдела распечатывает документ, созданный в п.7 или открывает его в мобильном приложении Confluence и занимается сбором данных. В документе создаются комментарии о том, какие данные удалось собрать и в каком объеме. После окончания этого этапа СТО компании отмечает на интерфейсе плагина, что сбор данных окончен;

13) создание задачи на исследование данных. После выполнения п.12 плагин автоматически создает задачу в Jira с прикрепленной ссылкой на страницу Confluence со списком данных и назначает ее выполнение начальнику Data Science-отдела;

14) создание пометок к списку данных. Начальник Data Science-отдела исследует данные и правит документ со списком данных, делая пометки. После окончания выполнения этого задания начальник Data Science-отдела отмечает на интерфейсе плагина, что задача окончена;

15) создание задачи на подготовку данных. АСУ создает задачу в Jira на подготовку данных. РМ может назначить исполнителя данной задаче. Сотрудник Data Engineering-отдела получает задачу вместе с документом из п.10.;

16) создание узла «features» для данной модели. После того, как выполнен п.15, плагин создает узел «features»;

17) создание задания на составление «features». После окончания выполнения п.16, плагин создает в Jira задачу на составление списка «features». РМ может назначить исполнителя данной задаче;

18) создание шаблона списка «features». Как только создана задача, описанная в п.17, АСУ создает шаблон списка «features» для данного узла. Данный шаблон в качестве ссылки на страницу в системе Confluence прикрепляется к задаче из п.17.;

19) создание списка «features». После создания пометок к списку данных сотрудники Data Science-отдела проводят совещание с целью определения списка «features». Данный список заполняется в Confluence одним из сотрудников Data Science-отдела в заранее созданном шаблоне. После окончания этого этапа начальник Data Science-отдела отмечает на интерфейсе плагина, что список «features» составлен;

20) создание задачи на подготовку Data Sets. После выполнения п.19 плагин создает задачу в системе Jira на подготовку Data Sets. К задаче прикрепляется документ, созданный в п.19. РМ может назначить исполнителя данной задаче;

21) создание узла параметров для данного узла «features». После создания задачи, описанной в п.20, плагин создает новый узел в системе - узел параметров;

22) создание задания на составление параметров. После создания узла параметров, АСУ создает задачу в Jira на создание списка параметров текущего варианта модели. РМ может назначить исполнителя данной задаче;

23) создание шаблона списка параметров для данного узла. Плагин создает шаблон списка параметров в Confluence и прикрепляет ссылку на данную Confluence-страницу к задаче, описанной в п.22.;

24) создание списка параметров модели. После выполнения задачи, поставленной в п.22, сотрудники Data Science-отдела заполняют список параметров для текущей модели в шаблоне, созданном плагином;

Проектная команда занимается созданием модели по заданным параметрам и обучению ее на подготовленных Data Sets. После того, как модель будет считаться обученной, команда должна проанализировать ее показатели с точки зрения достижения KPI продукта.

25) создание шаблона для KPI данной ветки. После выполнения п.24, плагин создает шаблон в Confluence для списка показателей KPI для данной ветки;

26) сохранение получившихся KPI. После окончания обучения модели сотрудник Data Science-отдела заполняет созданный в п.25 шаблон.

По окончании п.26 проектная команда принимает решение о том, созданное решение является подходящим и можно считать этап R&D завершенным, или KPI считаются не достигнутыми. В случае, если KPI считаются не достигнутыми, перед проектной командой стоит выбор о том, в какой из узлов решения возвращаться:

- создать другие параметры модели с этим же набором «features» и вернуться в п.21;
- создать новый набор «features» и вернуться в п.16;
- вернуться к выбору нового варианта решения и вернуться в п.6.

Построим диаграмму прецедентов (Use Case диаграмма), отражающую варианты взаимодействия участников R&D этапа с моделируемым плагином.

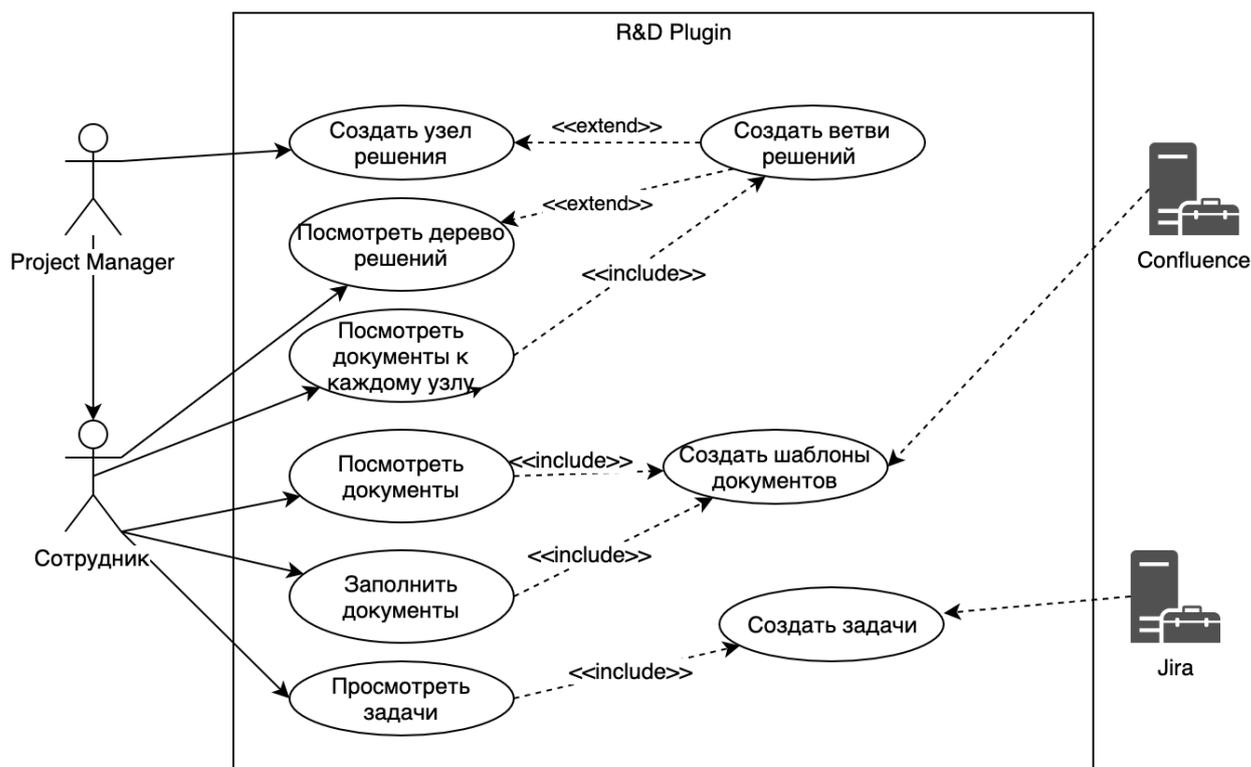


Рисунок 3.3 - Диаграмма прецедентов для модели «To Be» управления проектами на этапе R&D

Диаграмма вариантов использования позволяет описать функциональное назначение системы, а также показывает границы системы. R&D Plugin представляет собой систему, автоматизирующую процессы, которые требуют ручного управления. АСУ проектами может покрыть выполнение таких функций как создание документов и их структуры, создание задач. Данные функции являются автоматическими и выполняются в нужный момент времени на основании системы триггеров, которая коррелируется с заложенным в систему алгоритмом R&D этапа. Благодаря функциям отслеживания выполнения задач, и своевременного создания шаблонов нужной документации, система упрощает процесс создания документации к проекту, а также облегчает создание унифицированного ее формата. Также плагин позволяет удобным способом визуализировать фазы этапа R&D, а также позволяет видеть статусы каждой фазы исследования.

Для проведения структурного анализа моделируемой системы был использован метод моделирования «сущность-связь», или ER-диаграмма. Данный метод позволяет отобразить объекты, которые будут использоваться в системе, а также способы их взаимодействия. Диаграмма ER АСУ ИИ-проектами на R&D-этапе приведена на рисунке 3.3.

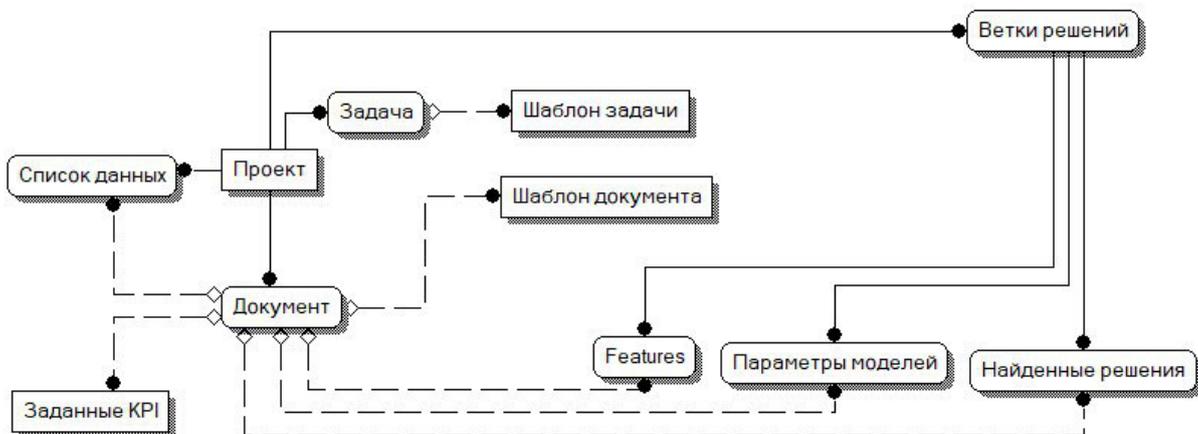


Рисунок 3.4 - ER-диаграмма модели «To Be» управления проектами на этапе R&D

ER-диаграмма отражает взаимодействие между ключевыми сущностями данной системы (проект, документы, задачи и ветки решений) в статике.

Совокупность описанных диаграмм (диаграмма экосистемы, диаграмма бизнес-процессов, диаграмма прецедентов и ER-диаграмма) дает полное описание модели «To Be» информационной системы управления ИИ-проектами на этапе R&D на предприятии ООО «Мастер Маинд Инк».

3.2 Качественная оценка модели автоматизированной системы управления проектами на предприятии ООО «МАСТЕР МАИНД ИНК»

С целью качественной оценки предложенной модели автоматизированной системы управления проектами были применены следующие методы:

- оценка степени автоматизации бизнес-процессов в сравнении моделей «As Is» и «To Be»;
- экспертных оценок.

Для выявления слабых и сильных сторон разработанной модели АСУ проектами, а также потенциала ее дальнейшего развития и определения возможных рисков был применен SWOT-анализ.

Оценка степени автоматизации проводилась по методике, описанной Кораблевым И.Г. [11].

Для проведения оценки степени автоматизации необходимо было подсчитать количество рутинных или ручных действий, выполняемых в настоящее время в компании (модель «As Is»), и количество тех же действий после возможного внедрения плагина (модель «To Be») и автоматизации процесса управления.

Наибольшее количество рутинных действий приходится на два процесса: создание задачи в системе Jira и создание документа в системе Confluence. Создание задач является рутинным действием, так как на этапе R&D задачи в основном являются краткосрочными, но их достаточно много, поэтому проектному менеджеру иногда является нецелесообразным тратить время на создание и управление задачами в системе Jira.

Создание документа является еще и трудоемким само по себе, так как система Confluence не самый удобный инструмент.

Также неудобством является то, что документы, в силу специфики этапа R&D, приходится создавать техническим специалистам.

Подобные обязанности редко возлагаются не на проектных менеджеров, поэтому технические специалисты часто вместо полноценного создания документа прикрепляют фотографию того же документа, написанного от руки, а то и вовсе пренебрегают выполнением данной задачи.

Оценка степени автоматизации проводилась на основании порядковой шкалы 10-ти уровней автоматизации (LOA) Т. Шеридана и В. Вепланка [41] (таблица 3.1).

В таблице 3.2 перечислены элементарные действия пользователя в процессе создания задачи и создания документа As Is и дана оценка степени их автоматизации.

Таблица 3.1 — Шкала уровней автоматизации Шеридана и Вепланка

Уровень автоматизации (LOA)	Описание
1	Компьютер не предлагает помощь: человек должен принимать все решения и выполнять все действия сам
2	Компьютер предлагает человеку полный набор решений/действий, альтернативы (пример – работа с электронным справочником)
3	Компьютер предлагает полный набор решений/действий, альтернативы и сужает выбор до нескольких вариантов
4	Компьютер предлагает одну альтернативу
5	Компьютер предлагает одну альтернативу и автоматически выполняет это предложение, если человек соглашается
6	Компьютер предлагает одну альтернативу и выполняет это предложение, если человек в течение ограниченного времени не накладывает вето на автоматическое выполнение операции
7	Компьютер выполняет операции автоматически, обязательно информируя человека
8	Компьютер выполняет операции автоматически информирует человека, только если компьютер спросит
9	Компьютер выполняет операции автоматически информирует человека, только если он (компьютер) решит
10	Компьютер решает всё и действует автономно, не обращая внимание на человека

Рассмотренные процессы (создание документа и создание задачи) неоднократно повторяются на протяжении всего этапа R&D.

Найдем общее количество операций, выполняемых участниками проектной команды на одной полной ветке дерева решений.

Также разделим эти операции на 2 категории - ручные ($LOA < 6$) и автоматические ($LOA \geq 6$) - и найдем процент содержания каждой из категорий операций на этапе R&D.

Таблица 3.2 — Степень автоматизации процессов создания задачи и создания документов As Is

Название процесса	Описание операций в данном процессе	LOA
Создание документа	Открыть Confluence	1
	Открыть нужную папку	1
	Создать документ	1
	Создать структуру документа	1
	Заполнить документ	1
	Опубликовать документ	1
Создание задачи	Открыть Jira	1
	Открыть Backlog	1
	Создать задачу	1
	Описать задачу	1
	Назначить исполнителя	1
	Перенести в текущий спринт	1

Таблица 3.3 — Степень автоматизации для процесса As Is на одной полной ветке дерева решений

Вид операции	Кол-во операций в задаче	Кол-во задач на одной ветке	Общее кол-во операций на этапе R&D для одной ветке	Процент операций на этапе R&D
Процесс создания задачи				
Ручные операции	6	8	48	100%
Автоматические операции	-	-	-	-
Процесс создания документа				
Ручные операции	6	6	36	100%
Автоматические операции	-	-	-	-

Как видно из таблицы 3.3, процессы создания задачи и создания документа являются ручными на 100%.

Построим таблицы, аналогичные таблицам 3.2 и 3.3, для модели To Be.

Таблица 3.4 — Степень автоматизации процессов создания задачи и создания документов To Be

Название процесса	Описание операций в данном процессе	LOA
Создание документа	Открыть Confluence	1
	Открыть нужную папку	10
	Создать документ	10
	Создать структуру документа	10
	Заполнить документ	1
	Опубликовать документ	1
Создание задачи	Открыть Jira	10
	Открыть Backlog	10
	Создать задачу	10
	Описать задачу	10
	Назначить исполнителя	1
	Перенести в текущий спринт	10

Таблица 3.5 — Степень автоматизации процесса To Be на одной полной ветке дерева решений

	Кол-во операций в задаче	Кол-во задач на одной ветви	Общее кол-во операций на этапе R&D для одной ветви	Процент операций на этапе R&D
Процесс создания задачи				
Ручные операции	1	8	8	14,3%
Автоматические операции	6	8	48	85,7%
Процесс создания документа				
Ручные операции	3	5	15	50%
Автоматические операции	3	5	15	50%

Анализ таблиц 3.4 и 3.5 показывает, что для одной ветки исследования степень автоматизации процессов создания задач увеличилась на 85,7%, а степень автоматизации процесса создания документа на 50%.

Увеличение уровня автоматизации процессов приводит также к росту некоторых качественных показателей управления, среди которых можно указать улучшение визуального представления циклов решения, потенциальное

сохранение большего количества данных о создаваемых программных решениях и многие другие.

Практическая актуальность и целесообразность разработанной модели автоматизированной системы управления проектами на предприятии «ММИ» подтверждается отзывами экспертов - техническим директором и исполнительным директором компании ООО «Мастер Маинд Инк» (приложения Г и Д).

Для оценки эффективности разработанной модели АСУ проектами бы применен SWOT-анализ. Он позволил выявить достоинства и недостатки системы, на которые следует обратить внимание при дальнейшей разработке программного продукта. Также были выявлены потенциальные направления развития и способы монетизации программного продукта. Последней задачей SWOT-анализа стало выявление возможных рисков, связанных с разработкой моделируемого программного продукта.

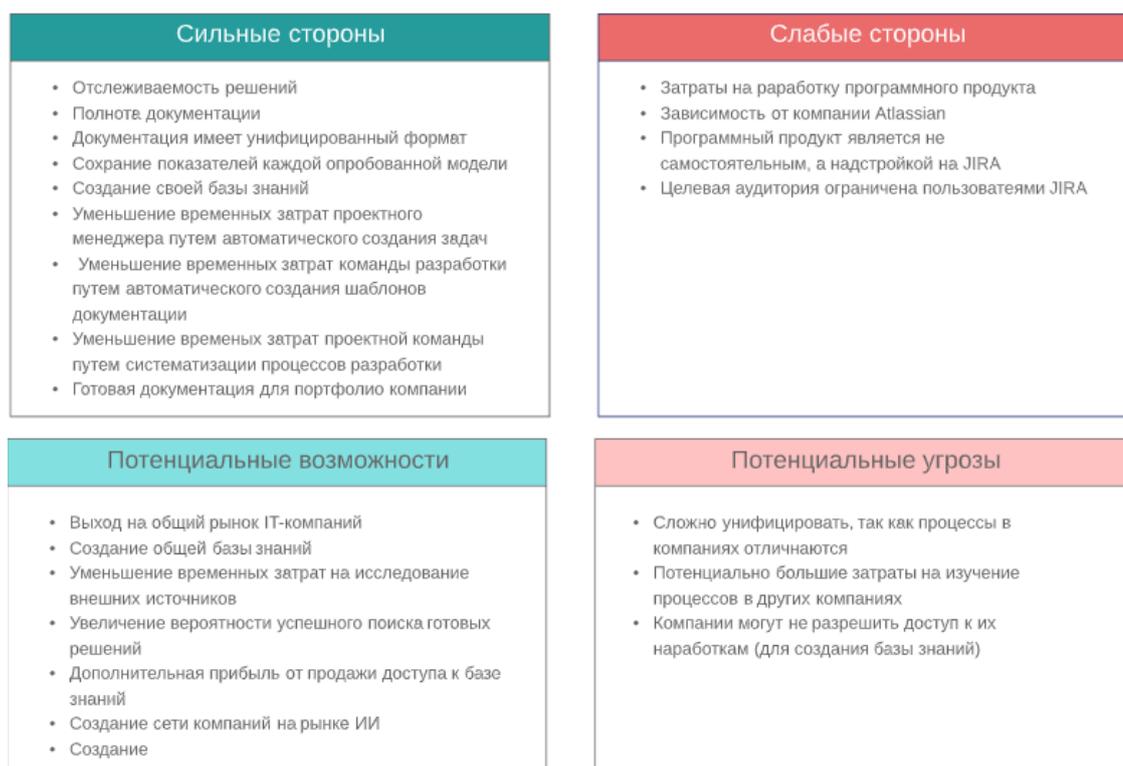


Рисунок 3.5 - SWOT-анализ АСУ проектами в компании ООО «Мастер Маинд Инк»

SWOT-анализ показал, что предложенная система имеет значительное количество сильных сторон и преимуществ перед старой моделью управления. Модель To Be предлагает такие улучшения как: упорядочивание документооборота, увеличение степени визуального и функционального комфорта работы всех участников проекта, улучшение коммуникации между ними и тд.

Слабые стороны в основном связаны с «привязкой» к системе Jira и их зависимости таким образом от ценовой политики компании Atlassian и от того, насколько стабилен будет функционал продуктов Atlassian. Также к уязвимостям предложенного плагина можно отнести сужение целевой аудитории до пользователей продуктов Atlassian, так как не все data science компании используют Jira и Confluence.

Выводы к главе 3

Исследование методов, средств и существующих информационных систем управления проектами; подробный анализ деятельности участников проектной команды на этапе R&D; разработанный алгоритм процесса проектной деятельности на этапе R&D; изученная модель «As Is» существующих процессов управления в компании ООО «Мастер Маинд Инк» позволили разработать модель «To Be» автоматизированной системы управления проектами, которая учитывает недостатки управления, имеющиеся в компании, например, неналаженный документооборот и наличие рутинных действий.

Модель «To Be» предлагает адаптировать существующую в компании систему управления проектами Jira и Confluence, встроив в нее плагин управления этапом «R&D». При этом (в установленных пределах) несколько изменяется структура, функции и организация информационных потоков существующей системы.

Модель «To Be» полностью описывается четырьмя диаграммами: диаграмма экосистемы встраиваемого плагина для управления проектами на

этапе R&D, диаграмма бизнес-процессов модели «To Be» управления проектами на этапе R&D, диаграмма прецедентов для модели «To Be» управления проектами на этапе R&D, ER-диаграмма модели «To Be» управления проектами на этапе R&D.

Предложенного описания модели «To Be» с помощью указанных диаграмм достаточно для того, чтобы приступить к написанию SRS и разработке плагина.

ЗАКЛЮЧЕНИЕ

Основные научные результаты диссертации заключаются в следующем.

В соответствии с целью исследования, было проведено теоретическое обоснование модели автоматизированной системы управления проектами в компании ООО «Мастер Маинд Инк», которая специализируется на разработке программных продуктов, содержащих искусственный интеллект и применяющих машинное обучение. Анализ первоисточников по теме исследования - научных статей, учебных пособий по теории управления проектами, материалов сети Интернет, где содержится большое количество значимой информации о практическом опыте использования различных информационных систем управления проектами позволил теоретически обосновать необходимость и возможность использования существующих средств и инструментов для построения модели ИС управления ИИ-проектами на предприятии.

Для достижения поставленной цели были изучены современные методы, инструменты и информационные системы управления проектами, определены их преимущества и недостатки; путем наблюдения за деятельностью проектной команды компании «Мастер Маинд Инк» по созданию ИИ-продукта было проанализировано ее содержание и выявлено большое количество рутинных действий, неуправляемых, дублирующих и потому бесполезных работ, а также недостатки документооборота. Поскольку управление проектами в компании ведется в системе Jira, было принято решение о разработке модели АСУ проектами с целью дальнейшего создания на ее основе плагина, встраиваемого в Jira.

Теоретическое обоснование и анализ процессов управления на предприятии «Мастер Маинд Инк» позволили разработать модель автоматизированной системы управления проектами, полное описание которой составлено из четырех диаграмм: диаграммы экосистемы встраиваемого плагина для управления проектами на этапе R&D, диаграммы бизнес-процессов модели «To Be» управления проектами на этапе R&D, диаграмма прецедентов

для модели «To Be» управления проектами на этапе R&D, ER-диаграмма модели «To Be» управления проектами на этапе R&D.

Была произведена качественная оценка разработанной модели с применением таких методов экспертных оценок и сравнительного анализа уровня автоматизации бизнес-процессов предприятия при использовании моделей управления «As Is» и «To Be». Было показано, что модель «To Be» позволяет увеличить степень автоматизации создания задач на 85,7%, а создания документа на 50%.

SWOT-анализ модели «To Be» показал сильные и слабые стороны новой модели, а также возможные риски и потенциал дальнейшего развития.

В целом, **новизна** исследования и его **теоретическую значимость** заключается в том, что предложенная модель системы управления проектами направлена в первую очередь на управление этапом R&D, который наиболее трудно поддается управлению в силу специфики научно-исследовательской деятельности; для создания адекватной модели управления был разработан алгоритм процесса проектной деятельности на этапе R&D; предложен способ представления жизненного цикла проекта и фаз проектной деятельности в виде дерева решений, явным преимуществом которого перед иными моделями жизненных есть визуальное отражение исходных точек начала следующего цикла.

Рекомендации по практическому применению разработанной модели и дальнейшее развитие темы диссертационной работы заключаются в следующем:

1) воплощение модели «To Be» в виде плагина R&D и внедрении его в систему управления проектами компании ООО «Мастер Маинд Инк»;

2) управление временем. На данный момент предложенная модель нигде не учитывает такой важный фактор, как время. Нигде в модели не присутствует контроль времени, то есть не установлен параметр и не разработана функция, регулирующая количество времени, необходимое для:

- изучения предметной области проекта,

- поиска адекватных и доступных для реализации решений формулируемых задач;
- повторения цикла исследования (если не повезло найти сразу нужное решение).

Также не определена протяженность временного интервала обучения модели.

Проблема регулирования времени в научно-исследовательских проектах, преследующих бизнес-цели, является очень важной, так как увеличение количества времени на R&D-этапе является вполне естественным. Исследователь стремится найти наилучшее решение, как с точки зрения эффективности, так и с эстетической и познавательной.

Трудность заключается в том, что количество времени в такого рода исследованиях действительно трудно регулировать. Планирование работ и прогноз того, сколько потребуется часов, дней или месяцев на исследовательскую часть разработки проекта (estimate) зависит от многих факторов, как внешних, так и внутренних: сложности предметной области, ее уровня изученности и разработанности в науке; наличия математических и прикладных процедур, необходимых для создания нейронной сети; сложности реализации самого проекта; профессионального уровня сотрудников, и даже психологического личностного профиля каждого; сложившихся взаимоотношений внутри компании и с заказчиком и др.

Тем не менее управление временем в компании выполняется - каждым сотрудником, проектным менеджером и Product Owner-ом.

Следовательно, можно попробовать разработать некоторые плавающие временные коэффициенты, управляющие процессом создания проекта. В процессе накопления опыта выполнения проектов компанией будет происходить настройка временного коэффициента и совершенствоваться система управления проектами.

3) в процессе расширения «библиотеки готовых решений» по профилю компании возможна ее интеграция с внешними подобными библиотеками;

4) создание рекомендательной подсистемы, наблюдающей за работой исследователя на этапе «анализ решений». Изучая запросы исследователя в конкретной области, такая подсистема сможет помогать сотруднику в поиске необходимого решения, осуществляя поиск источников.

Экономичность и результативность предложенной модели автоматизированной системы управления проектами обеспечивается следующими ее свойствами:

- автоматизация рутинной работы на предприятии;
- автоматизация ручных методов управления;
- сохранение всех документов, сопутствующих процессу создания проекта;
- направления организационных потоков через единый информационный центр;
- сокращение времени, необходимого для каждого коммуникативного акта;
- независимость от взаимной территориальной удаленности сотрудников на проекте;
- повышение комфорта и скорости взаимодействия между сотрудниками.

Достоверность и обоснованность результатов обеспечивается выбранной теоретической и методологической основой исследования; аргументированностью выводов, подтвержденных оценками экспертов - директоров компании «Мастер Маинд Инк»; соответствием методов исследования цели, предмету и задачам.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Арутюнова, Д.В. Инновационный менеджмент: [учебное пособие] – Ростов-на-Дону: Изд-во ЮФУ, 2014. – 152 с.
2. Александрова, Т.В. Повышение эффективности проектного управления в организации на основе гибкой методологии Agile // Экономика и бизнес: теория и практика. 2019. №9.
3. Баканов, А. Б., Дрождин, В. В., Зинченко, Р. Е., Кузнецов, Р. Н. - Методы адаптации и поколения развития программного обеспечения // Известия ПГПУ им. В. Г. Белинского. 2009. № 13 (17). С. 66-69
4. Бэгьюли, Ф. Управление проектом: пер. с англ. / Ф. Бэгьюли – М.: Гранд ФАИР-ПРЕСС, 2002. – 202 с.
5. Вареникова, О.В., Бобылева, А.А., Голубев, Д.В. Управление проектами в электроэнергетике/ Colloquium-journal. – 2019. - №13(37). – С.43-56
6. Вертакова, Ю. В. Управленческие решения: разработка и выбор: учеб. пособие / Ю. В. Вертакова, И. А. Козьева, Э. Н. Кузьбожев; под общ. Ред. Проф. Э. Н. Кузьбожева. — М.: КНОРУС, 2005. — 352 с.
7. Голубев, С.А. Управление венчурными проектами.-СПб, СпбГТУ, 2009
8. Дерябкин, В.П. Проектирование автоматизированных систем обработки информации и управления: курс лекций / В.П. Дерябкин. - Самара: СГАУ, 2001. – 120 с.
9. Заренков, В. А. Управление проектами: учебное пособие. - 2 изд. - СПб: АСВ, 2010. - 312 с.
10. Ивасенко, А. Г. Управление проектами / А. Г. Ивасенко, Я. И. Никонова, М. В. Каркавин – Ростов-на-Дону : Феникс, 2009. – 327 с.
11. Кораблев, И. Г. Оценка уровня автоматизации бизнес-процессов предприятия // Вестник Череповецкого государственного университета. 2016. №1 (70). URL: <https://cyberleninka.ru/article/n/otsenka-urovnya-avtomatizatsii-biznes-protsesov-predpriyatiya> (дата обращения: 16.12.2019)

12. Кузнецова, Н. В. Методы принятия управленческих решений / учебное пособие. — Москва, Инфра-М, 2015. — 222 с.
13. Кузнецова, Н. В. Анализ подготовки и принятия управленческих решений в практике: к вопросу выбора оптимального метода // Молодой ученый. — 2016. — №27. — С. 425-433.
14. Лифшиц, А. С. Управленческие решения: учеб. пособие. — М.: КНОРУС, 2009. — 248 с.
15. Мазур, И. И. Управление инвестиционно-строительными проектами: международный подход / И. И. Мазур. М. : Омега-Л, 2011. 736 с.
16. Макашов, П. Л., Романенко, Н. А. Сервис-ориентированный подход к управлению ИТ проектами на примере использования программного продукта "Jira" // Современные информационные технологии и ИТ-образование. - 2015. - №11. – С.12-16.
17. Николаенко, В. С. Разработка принципов управления ИТ-проектом // Вестн. Том. гос. ун-та. - 2015. -№390. –С.56-60.
18. Норенков, И.П. Автоматизированные информационные системы: учеб. пособие / И.П. Норенков. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2011. – 342 [2] с.: ил. – (Информатика в техническом университете).
19. Озерова, Т. Системная триада как основа управления проектами на предприятиях общественного питания / Т. Озерова // РИСК : ресурсы, информация, снабжение, конкуренция. 2011. № 2, ч. 1. С. 35–38.
20. Основы управления проектами: [учеб. пособие] / Л. Н. Боронина, З. В. Сенук; М-во образования и науки Рос. Федерации, Урал. федер. ун-т. – Екатеринбург : Изд-во Урал. ун-та, 2015. — 112 с.
21. Преображенская, Т.В. Управление проектами: учеб. пособие / Т.В. Преображенская, М.Ш. Муртазина, А.А. Алетдинова. – Новосибирск: Изд-во НГТУ, 2018 г.- 123 с.
22. Сербская, О.В. Современные методы управления проектами // Материалы Афанасьевских чтений, №2, 2016

23. Скотт, А. Гибкие технологии: экстремальное программирование и унифицированный процесс разработки. Wiley (ISBN 0-471-20282-7), 2002 - Scott W. Ambler, Agile Modeling: Effective Practices for Extreme Programming, 2002; перевод и издание на русском языке: ЗАО Издательский дом “Питер” (ISBN 5-94723-545-5), 2005
24. Управление проектами / В. Д. Шапиро и др. СПб.: ДваТрИ, 1996. 610 с.
25. Фунтов, В. Н. Основы управления проектами в компании. / В. Н. Фунтов – СПб.: Питер, 2011. – 393 с.
26. Чуланова, О. Л. Технология управления проектами и проектными командами на основе методологии гибкого управления проектами Agile // Вестник евразийской науки. - 2018. - №1.-С 31-35.
27. Шайхулова, А.Ф. Автоматизация и управление инновационными проектами технического перевооружения авиадвигателестроительного производства на основе каскадного метода оптимизации: Автореф...дис. кан. тех. наук. – Уфа, 2018. –20 с.
- Электронные ресурсы*
28. Agile-манифест разработки программного обеспечения // <https://agilemanifesto.org/iso/ru/manifesto.html>
29. Понятие «проект» // Pandia URL: <http://www.pandia.ru/365896/>
30. Confluence // <https://startpack.ru/application/confluence>
31. PMBOK® Guide – Sixth Edition // Project Management Institute: <https://www.pmi.org/pmbok-guide-standards/foundational/pmbok>
32. Жизненный цикл проектной задачи // Projectimo: <http://projectimo.ru/upravlenie-proektami/zhiznennyj-cikl-proekta.html>
33. Орлик, С. Основы программной инженерии по SWEBOK // <https://ligurio.github.io/swebok-ru/>
34. Advantages and Disadvantages of Trello // SoftwareDeveloperIndia: <https://www.software-developer-india.com/advantages-and-disadvantages-of-trello>.
35. What is Asana? // CompareCamp: <http://comparecamp.com/asana-reviews-pricing-benefits-and-features-analysis/>

36. 6 Benefits of Using JIRA // Dumbfunded: <https://www.dumbfunded.co.uk/guides/6-benefits-of-using-jira/>
37. Trello vs Asana: The Best Project Management App in 2017? // Process.st: <https://www.process.st/trello-vs-asana/>
38. Product Guides & Tutorials // Atlassian: <https://www.atlassian.com/software/jira/guides/getting-started/overview>
39. Project management. A Guide to the Project Management Body of Knowledge: PMBOK guide. 3rd ed. 2004. 506 p. [Electronic resource]: <http://rutracker.org/forum/viewtopic.php?t=2831128>
Литература на иностранном языке
40. A Spiral Model of Software Development and Enhancement. Barry W. Boehm, TRW Defense System Group, 1988
41. Frohm J., Stahre Jo, Winroth M. P. Levels of Automation in Manufacturing // International Journal of Ergonomics and Human Factors, Vol. 30 Issue 3, 2008.
42. Managing the Development of Large Software Systems: concepts and techniques. Rouse Winston, 1970, ICSE '87 Proceedings of the 9th international conference on Software Engineering
43. Oberlender Garold D. Project management for engineering and construction / 2nd ed. New York: McGrawHill, 2000. 368 p.
44. "Software Engineering — Guide to the software engineering body of knowledge". [ISO/IEC TR 19759:2015](#)
45. Walker Anthony. Project management in construction / 4th ed. Oxford: Blackwell Science, 2002. 289 p.

Приложение А

Модель бизнес-процессов на этапе R&D

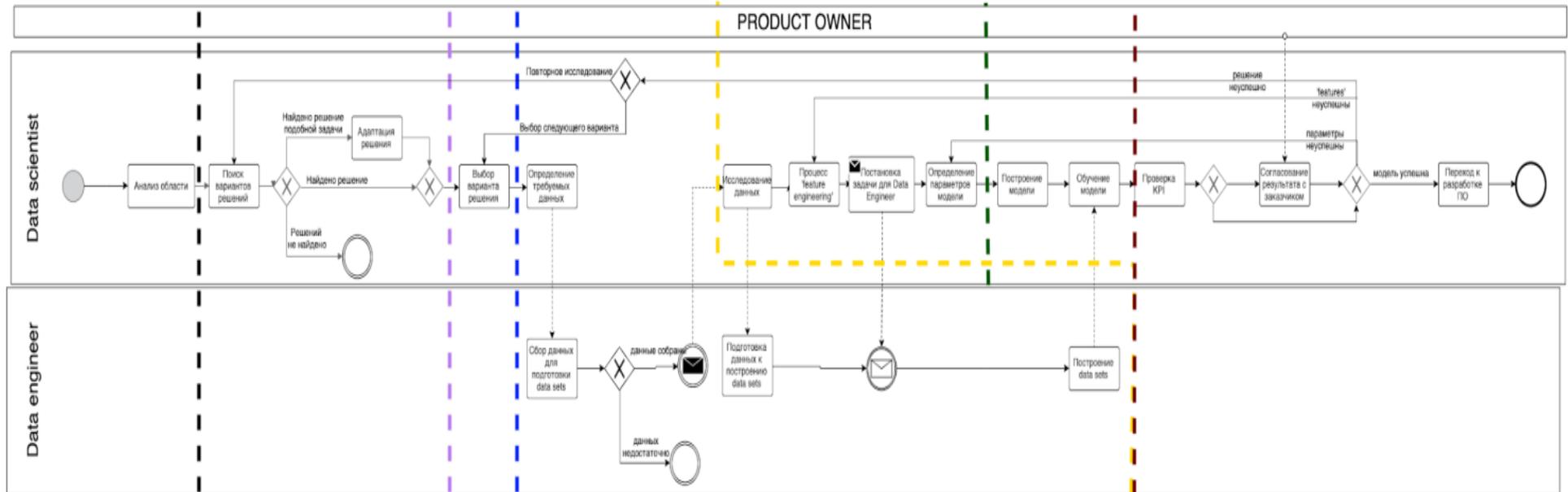


Рисунок А.1 - Модель бизнес-процессов на этапе R&D

Приложение Б

Модель “AS IS” управления проектами на этапе R&D

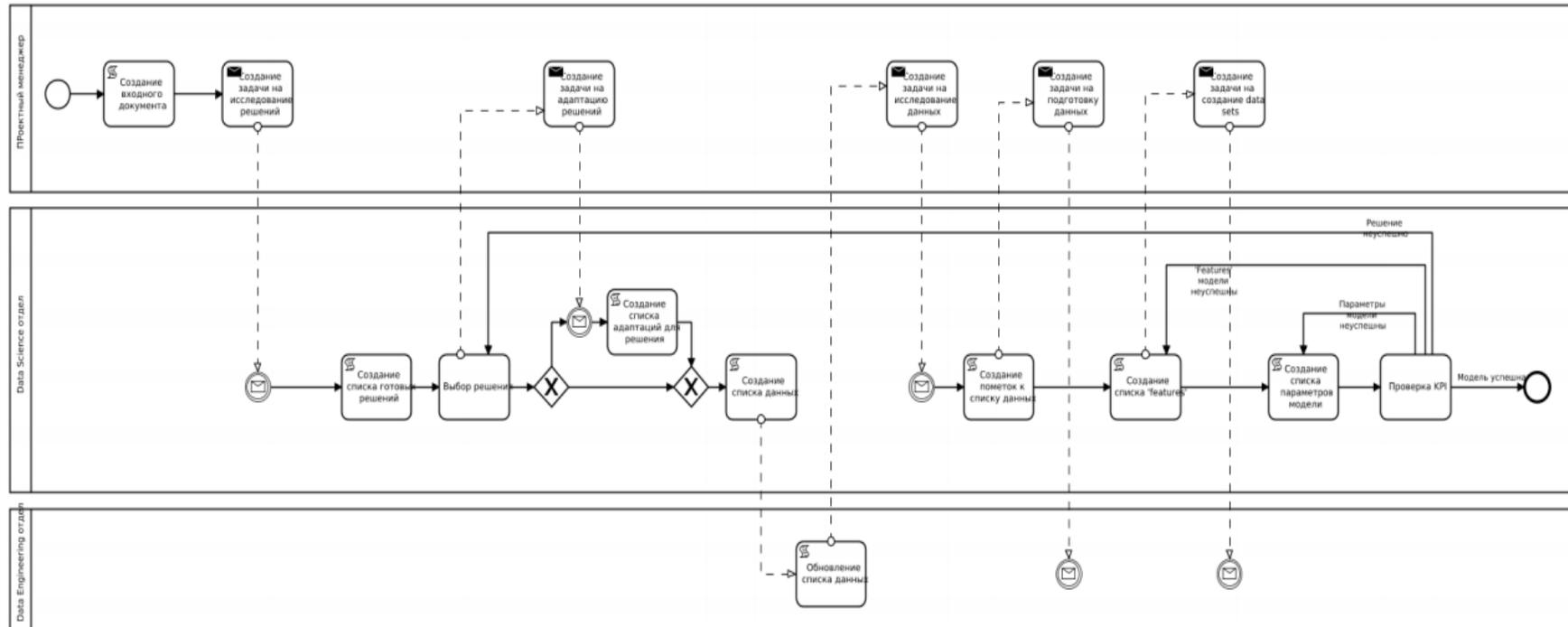


Рисунок Б.1 - Модель “AS IS” управления проектами на этапе R&D

Приложение В

Модель АСУ управления проектами на предприятии

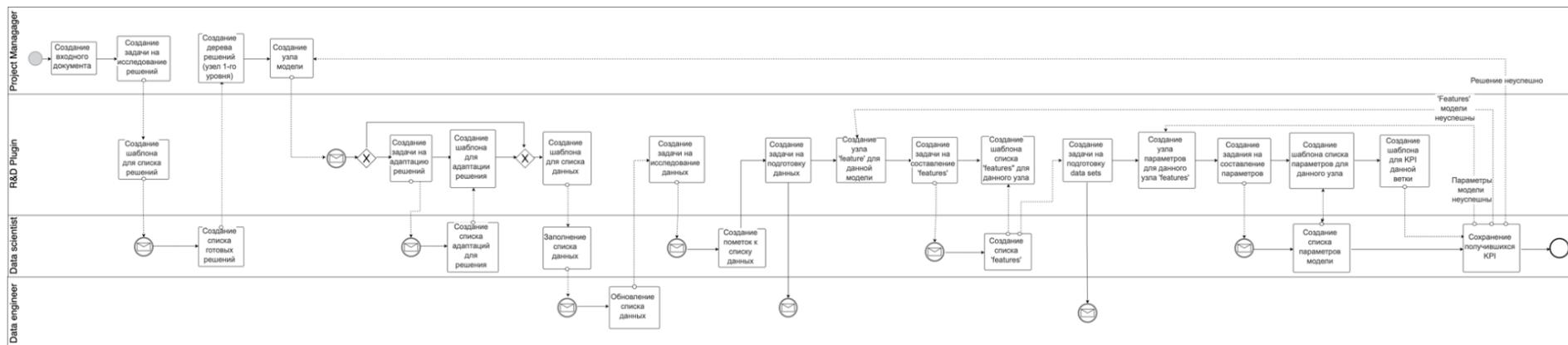


Рисунок В.1 - Модель АСУ управления проектами на предприятии ООО “Мастер Маинд Инк”