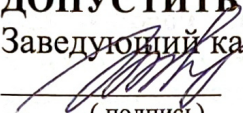


Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РтФ
Кафедра/департамент информационных технологий и систем управления

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

Заведующий кафедрой


(подпись) Е.В. Кислицын
(Ф.И.О.)

« 28 » 05 2024 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Методика управления внеплановыми проектами в области машинного обучения

Научный руководитель: Юманова Ирина Фарисовна
канд. физ.-мат. наук, доцент



подпись

Нормоконтролер: Абакумова Анна Геннадьевна



подпись

Студент группы: Кременецкий Михаил Васильевич



подпись

Екатеринбург
2024

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РИФ
Кафедра/департамент информационных технологий и систем управления
Направление подготовки 09.04.04 Программная инженерия
Образовательная программа Разработка и управление в программных проектах

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студента Кременецкого Михаила Васильевича группы РИМ-220990
(фамилия, имя, отчество)

1. Тема выпускной квалификационной работы «Методика управления внеплановыми проектами в области машинного обучения»

Утверждена распоряжением по институту от «04» декабря 2023 г. № 33.02-05/298

2. Научный руководитель

Юманова Ирина Фарисовна, доцент, канд. физ.-мат. наук
(Ф.И.О., должность, ученая степень, ученое звание)

3. Исходные данные к работе результаты производственной практики (научно-исследовательской работы)

4. Перечень демонстрационных материалов

Презентационный материал по итогам выполнения ВКР

5. Календарный план

№ п/п	Наименование этапов выполнения работы	Срок выполнения этапов работы	Отметка о выполнении
1.	1 раздел (глава)	до 12.04.2024 г.	
2.	2 раздел (глава)	до 03.05.2024 г.	
3.	3-4 разделы (главы)	до 17.05.2024 г.	
4.	ВКР в целом	до 20.05.2024 г.	

Руководитель Юманова И.Ф.

Ф.И.О.

Студент задание принял к исполнению 20.03.2024

дата

(подпись)

(подпись)

6. Допустить Кременецкого Михаила Васильевича к защите выпускной квалификационной работы в экзаменационной комиссии

Зав. кафедрой ИТиСУ

(подпись)

Е.В. Кислицын
Ф.И.О.

РЕФЕРАТ

Выпускная квалификационная работа состоит из введения, 3 глав и заключения, изложенных на 48 страницах, а также списка литературы из 50 наименований и 2 приложений. В работе имеется 9 рисунков и 1 таблица.

Ключевые слова: МЕТОДИКА, МАШИННОЕ ОБУЧЕНИЕ, УПРАВЛЕНИЕ ПРОЕКТАМИ, ПРОГРАММНЫЙ ПРОЕКТ, ПРОЕКТНОЕ ОБУЧЕНИЕ, ВНЕПЛАНОВЫЙ ПРОЕКТ.

Разработана и опробована методика для управления внеплановыми программными проектами в области машинного обучения. Рассматриваются существующие методики для управления программными проектами и причины, почему они не применимы для внеплановых проектов в области машинного обучения. Приводится свод правил, выполнение которых подразумевает разработанная методика. Описывается процесс апробации методики во время управления разработкой программного продукта в рамках учебного проекта, условия которого соответствуют условиям внепланового проекта, а также возможность масштабирования методики для управления непосредственно внеплановым проектом.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Исследование существующих методик управления программными проектами и условий внеплановой разработки	8
1.1 Каскадная модель «Waterfall».....	8
1.2 Методологии Agile.....	9
1.3 Методики управления проектами машинного обучения	15
1.4 Условия внеплановой разработки	18
2 Разработка правил методики управления внеплановыми проектами в области машинного обучения	20
2.1 Разработка продукта машинного обучения.....	20
2.2 Организация процесса разработки	22
3 Апробация методики при управлении разработкой продукта.....	25
3.1 Управление студенческим проектом.....	25
3.2 Масштабирование разработанной методики	36
ЗАКЛЮЧЕНИЕ	37
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	40
ПРИЛОЖЕНИЕ А	46
ПРИЛОЖЕНИЕ Б.....	48

ВВЕДЕНИЕ

Актуальность темы исследования. На сегодняшний день развитие и расширение сферы информационных технологий стремительно увеличивается [1], что приводит к повышенному спросу на подготовленных специалистов в разработке программных продуктов и управления проектами. В ответ на спрос государство увеличивает количество бюджетных мест в образовательных учреждениях по соответствующим направлениям.

Учреждения высшего образования, в связи с этим, разрабатывают образовательные программы, целью которых является обучение теоретическим и практическим знаниям и навыкам, необходимым в разработке программных продуктов [2]. Одним из способов, используемых учреждениями высшего образования для обучения студентов, является внедрение проектного обучения, в рамках которого группа студентов занимается разработкой по теме, предложенной сторонней компанией или предприятием. Также необходимо отметить, что одним из наиболее востребованных направлений у компаний-заказчиков является машинное обучение (на осенний семестр 2023 года в институте радиоэлектроники и информационных технологий ФГАОУ ВО «УрФУ» более половины представленных тем относятся к разработке в области машинного обучения).

Для компаний, занимающихся разработкой программных продуктов, внеплановые проекты являются обязанностью, однако методики для управления подобными проектами разработаны для графика полного рабочего дня [3]. Кроме того, разработка продукта машинного обучения имеет индивидуальные этапы разработки, отличающие ее от разработки иных программных проектов [4]. Аналогичными условиями обладают и учебные проекты, выполнение которых происходит в неопределенные укороченные временные промежутки. В рамках проектного обучения, перед студентами, помимо разработки непосредственно продукта, ставится задача по организации командной разработки [5]. Для группы студентов, у которых

отличается уровень профессиональных навыков, опыт работы в области выбранной темы, опыт работы в командной разработке и внеучебный график, организация и планирование процессов разработки является дополнительной нагрузкой и фактором, способным замедлить и сбить фокус непосредственно с разработки продукта.

Анализ актуальности обусловил выбор темы исследования: методика управления внеплановыми проектами в области машинного обучения.

Гипотеза исследования состоит в том, что применение разработанной методики внеплановых проектах в области машинного обучения позволит четко соблюдать требования заказчика, параллельно вести асинхронную разработку без потери коммуникации как внутри команды, так и между стейкхолдерами проекта, выполнять планирование задач с уверенностью достижения цели проекта.

Целью исследования является разработка и апробация методики для управления внеплановым проектом на примере проекта «Фото идентификация тигров. Модель определения стороны тигра(лево/право)», а также масштабируемость методики для управления внеплановым проектом.

Для достижения цели необходимо решить следующие **задачи**:

- 1) исследование существующих методик управления программными проектами в области машинного обучения, условий проектного практикума и внеплановой разработки, обоснование использования разрабатываемой методики;
- 2) разработка правил методики и их апробация в рамках проекта «Фото идентификация тигров. Модель определения стороны тигра(лево/право)»;
- 3) анализ результатов апробации разработанной методики в рамках проекта «Фото идентификация тигров. Модель определения стороны тигра(лево/право)», масштабирование разработанной методики.

Объектом исследования является процесс управления студенческим проектом «Фото идентификация тигров. Модель определения стороны

тигра(лево/право)», в рамках которого студенты выполняют разработку программного продукта в области машинного обучения.

Предметом исследования является использование разработанной методики для управления внеплановыми проектами в области машинного обучения.

Методы исследования включают в себя:

- 1) анализ существующих методологий для управления проектами;
- 2) сравнение рассматриваемых условий для их применения;
- 3) экспериментальное опробование разработанной методики в рамках студенческого проекта;
- 4) измерение равномерности загруженности членов команд задачами, количества запланированных, выполненных в срок, просроченных и отмененных задач;
- 5) беседа со стейкхолдерами проекта, обобщение субъективных мнений о личном опыте и впечатлениях о проекте (формат «Ретроспектива»).

Теоретической основой исследования стали:

- 1) отечественная и зарубежная литература, описывающая методологию управления программными проектами;
- 2) отечественные и зарубежные исследования по проблеме управления программными проектами;
- 3) современные концепции, раскрывающие сущность отдельных методик, понятий и определений, используемых в управлении программными проектами.

Теоретическая и практическая значимость работы заключается в том, что студенты получают возможность изучить и применить современные методы и инструменты управления проектами, что поможет им лучше понять принципы организации работы над программными проектами. Это также способствует повышению уровня теоретических знаний студентов в области управления проектами и развитию их навыков в этой области. Студенты приобретают опыт работы над реальным программным проектом, что

помогает им применить полученные теоретические знания на практике. Это также способствует развитию навыков командной работы, планирования, контроля и оценки результатов проекта. Кроме того, успешная апробация разработанной методики и ее публикация позволит студентам, управляя проектом, иметь набор правил и метрик, с помощью которых оценивать успешность достижения цели проекта и планировать предстоящие задачи.

База исследования:

- 1) информация о существующих методиках управления программными проектами;
- 2) инструменты для командного планирования, отслеживания и оценки задач;
- 3) учебный план студента магистратуры;
- 4) набор тем, представленных для формата проектного обучения.

На защиту выносятся следующие положения:

- 1) существующие методики для управления программными проектами плохо применимы для проектного обучения;
- 2) соблюдение правил разработанной методики позволяют провести разработку программного продукта в области машинного обучения студентам в рамках проектного обучения и оценить успешность его выполнения;
- 3) использование методики при разработке программного продукта позволила успешно завершить студенческий проект.

Апробация результатов исследования и публикации. Основные результаты исследования были опробованы во время управления разработкой программного продукта «Фотоидентификация тигров. Модель определения стороны» и были представлены в журнале «Инженерный вестник Дона».

Структура и объём работы. Выпускная квалификационная работа состоит из введения, 3 глав и заключения, изложенных на 48 страницах, а также списка литературы и 2 приложений. В работе имеется 10 рисунков и 1 таблица. Список литературы содержит 50 наименований. Во введении

описываются следующие составляющие исследования: актуальность темы, гипотеза, цель и задачи, объект и предмет, методы, теоретическая основа, теоретическая и практическая значимость, база, защищаемые положения, апробация результатов, структура и объем работы.

В главе 1 описывается исследование и анализ существующих методик для управления программными проектами, проектами в области машинного обучения, а также условий внеплановой разработки. В главе 2 приводится свод правил разработанной методики, соблюдение которых, согласно гипотезе исследования, должно привести к достижению целей проекта. В главе 3 описывается процесс апробации разработанной методики при управлении проектом по разработке программного продукта «Фотоидентификация тигров. Модель определения стороны», а также возможность ее масштабируемости для управления внеплановыми проектами на предприятии. В заключении приводятся основные результаты работы.

1 Исследование существующих методик управления программными проектами и условий внеплановой разработки

1.1 Каскадная модель «Waterfall»

Первые методики для управления программными проектами начали разрабатываться в 1960-х годах. Одним из самых ранних и важных вкладов в эту область была работа [6], опубликованная в 1970 году. В этой работе автор описал концепцию последовательной модели разработки программного обеспечения, которая стала основой для многих современных методологий управления программными проектами «Waterfall». Это система управления проектами, основанная на линейном, последовательном, структурированном подходе к разработке программного обеспечения. Он предполагает завершение каждого этапа проекта, прежде чем перейти к следующему.

Система состоит из нескольких отдельных этапов, включая сбор требований, разработку, внедрение, тестирование и техническое обслуживание [7]. Каждый этап имеет свой собственный набор результатов и целей, и после завершения этапа трудно вернуться назад и внести изменения, не затрагивая всего проекта.

Концепция «Waterfall» основана на предположении, что все требования могут быть определены в начале проекта и что в процессе разработки не будет никаких изменений в сфере охвата или требованиях [8]. Такой подход может быть полезен для проектов с четко определенными требованиями и стабильными условиями, но он может быть трудным для сложных проектов с меняющимися потребностями.

Одним из главных преимуществ системы «Waterfall» является то, что она обеспечивает четкую структуру для планирования проектов и управления ими. В нем также делается акцент на документации и официальных обзорах, которые могут помочь обеспечить достижение целей проекта. Однако она

может быть негибкой и может привести к задержкам, если в процессе разработки потребуются изменения.

1.2 Методологии Agile

В начале 2000-х годов были разработаны и опубликованы методологии Agile, которые предлагают гибкий и итеративный подход к управлению программными проектами, позволяющий быстро реагировать на изменения и требования заказчика [9]. Agile методологиями являются Scrum, ScrumBan, Extreme Programming, Crystal, Lean и др. [[10, 11, 12]. Гибкие методологии проектного управления являются одним из наиболее эффективных подходов к разработке и реализации проектов, которые позволяют максимально оптимизировать процесс работы команды и достичь наилучших результатов [10] Одной из основных характеристик таких методологий является разбиение проекта на управляемые разделы, длительностью от 1 до 4 недель, называемые спринтами или итерациями. На спринт назначаются задачи, выполнение которых возможно за срок его продолжительности. Во время каждого спринта команда работает совместно, чтобы обеспечить улучшение рабочего продукта и достижение поставленных целей.

В рамках гибких методологий проектного управления команда проводит регулярные встречи, такие как ежедневные стенды и спринтерские обзоры, чтобы убедиться, что все находятся в понимании общего состояния процессов и на ранней стадии выявить вопросы, которые необходимо решить. Это позволяет улучшить эффективность работы команды, сократить время на выполнение проекта и повысить качество конечного продукта. Кроме того, гибкие методологии проектного управления позволяют быстро реагировать на изменения в проекте и корректировать его направление, чтобы достичь максимальной эффективности и успеха.

Важным преимуществом гибких методологий проектного управления является их способность адаптироваться к различным условиям и требованиям

проекта. Команда может выбирать наиболее подходящие инструменты и методы работы, чтобы максимально эффективно достигнуть поставленных целей. Кроме того, гибкие методологии проектного управления позволяют команде работать более автономно и принимать самостоятельные решения, что способствует улучшению качества работы и повышению мотивации членов команды. Было показано, что гибкость улучшает результаты проекта за счет повышения удовлетворенности клиентов, сокращения времени выхода на рынок, а также повышения морального духа и производительности команды [11].

Методология **Scrum** основана на итеративном и инкрементальном подходе к разработке продукта и помогает быстро реагировать на изменения, улучшать качество продукта и повышать эффективность работы. Scrum включает в себя регулярные встречи команды, такие как планирование, ежедневные стендапы, ретроспективы и прочие составляющие, представленные на рисунке.

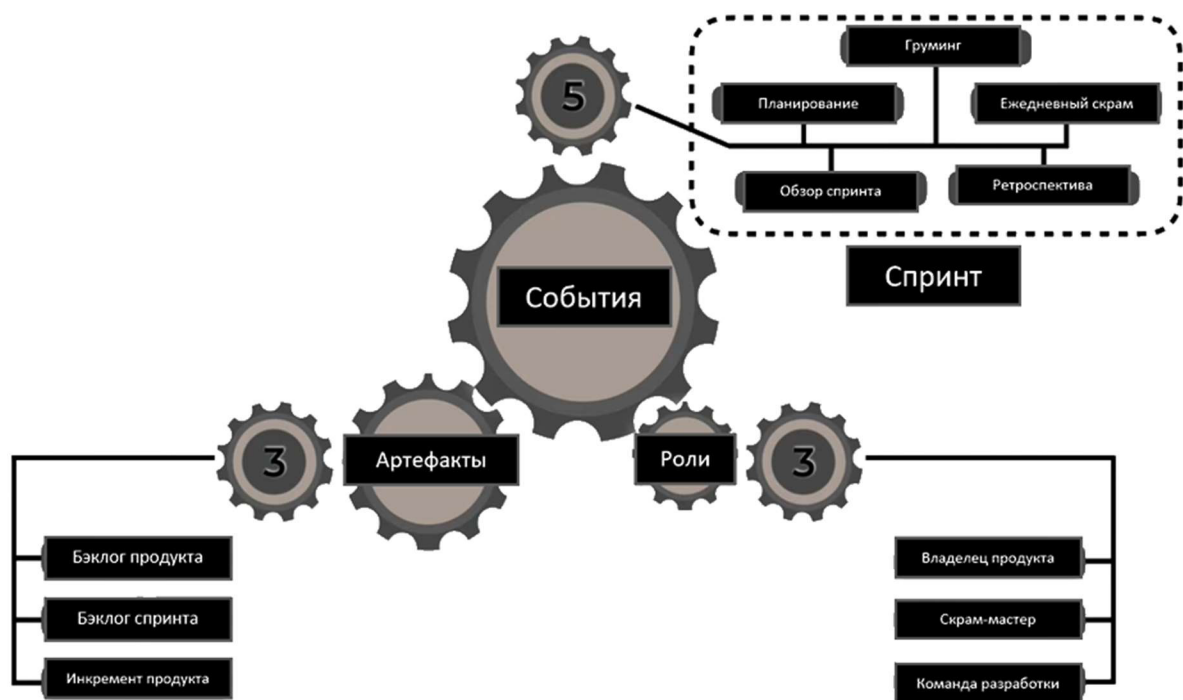


Рисунок 1 – Элементы методологии Scrum

Во время планирования определяются цель спринта и работа, которая будет сделана за этот спринт. Рекомендуемая длительность спринта определяется от 1 до 4 недель. На ежедневных стендапах каждый член команды должен ответить команде на три вопроса:

- что я сделал вчера, чтобы помочь команде достичь цель спринта;
- что я сделаю сегодня, чтобы помочь команде достичь цель спринта;
- вижу ли я какие-то препятствия, которые могут не дать команде достичь цель спринта;

Ретроспективы проводятся между спринтами. Это событие инспектирует то, как прошла работа в рамках последнего спринта. Это возможность для команды проверить себя и создать план улучшений, которые должны быть реализованы в течение следующего спринта.

ScrumBan является гибридом Scrum и Kanban и был разработан для команд, которые используют Scrum, но сталкиваются с проблемами, связанными с ограничениями времени и бюджета. Данная методология предназначена для управления проектами, которые имеют как структурированные, так и неструктурированные задачи, и может быть использована как для разработки программного обеспечения, так и для других проектов [13].

ScrumBan включает в себя элементы Scrum, такие как спринты и регулярные встречи, а также элементы KanBan, такие как визуализация рабочего процесса и управление потоком работ. Вместо того, чтобы иметь строго определенное количество задач на спринт, ScrumBan позволяет команде выбирать задачи из заранее сформированного списка и работать над ними в течение определенного периода времени.

ScrumBan также предоставляет возможность управления потоком работ с помощью ограничения количества задач, которые могут находиться в работе одновременно. Это позволяет команде более эффективно распределять задачи и управлять временем выполнения проекта.

Если проект имеет большое количество неструктурированных задач и требует быстрого реагирования на изменения, ScrumVan может быть хорошим выбором для управления данным проектом.

Экстремальное программирование (Extreme Programming) является Agile методологией разработки программного обеспечения, которая акцентирует внимание на качестве кода, коммуникации в команде разработчиков, тестировании и постоянном улучшении процесса разработки [14]. Основными отличительными особенностями методологии экстремального программирования являются парное программирование, смена позиций, тестирование до разработки, привлечение заказчика в процесс разработки.

Парное программирование заключается в том, что во время написания программного кода кем-либо из разработчиков, рядом с ним обязательно должен присутствовать второй разработчик и контролировать работу первого. Такой подход позволяет существенно улучшить качество кода, снизить число ошибок и ускорить обмен знаниями между разработчиками [15]. Пока один человек сосредоточивает усилия на стратегическом представлении об объекте, второй реализует его свойства и методы. Смена позиций разработчиков во также способствует обмену знаниями между разработчиками, устранению недопониманий в реализации тех или иных составляющих проекта, а также возможности взаимозаменяемости.

В экстремальном программировании тестирование происходит согласно методу тестирования до разработки (Test Driven Development, TDD) [16, 17], цикл которого представлен на рисунке 1. Перед написанием программного кода реализации некоей идеи (функции) разрабатывается тест, который позволяет проверить, работает ли этот будущий код реализации, или нет.

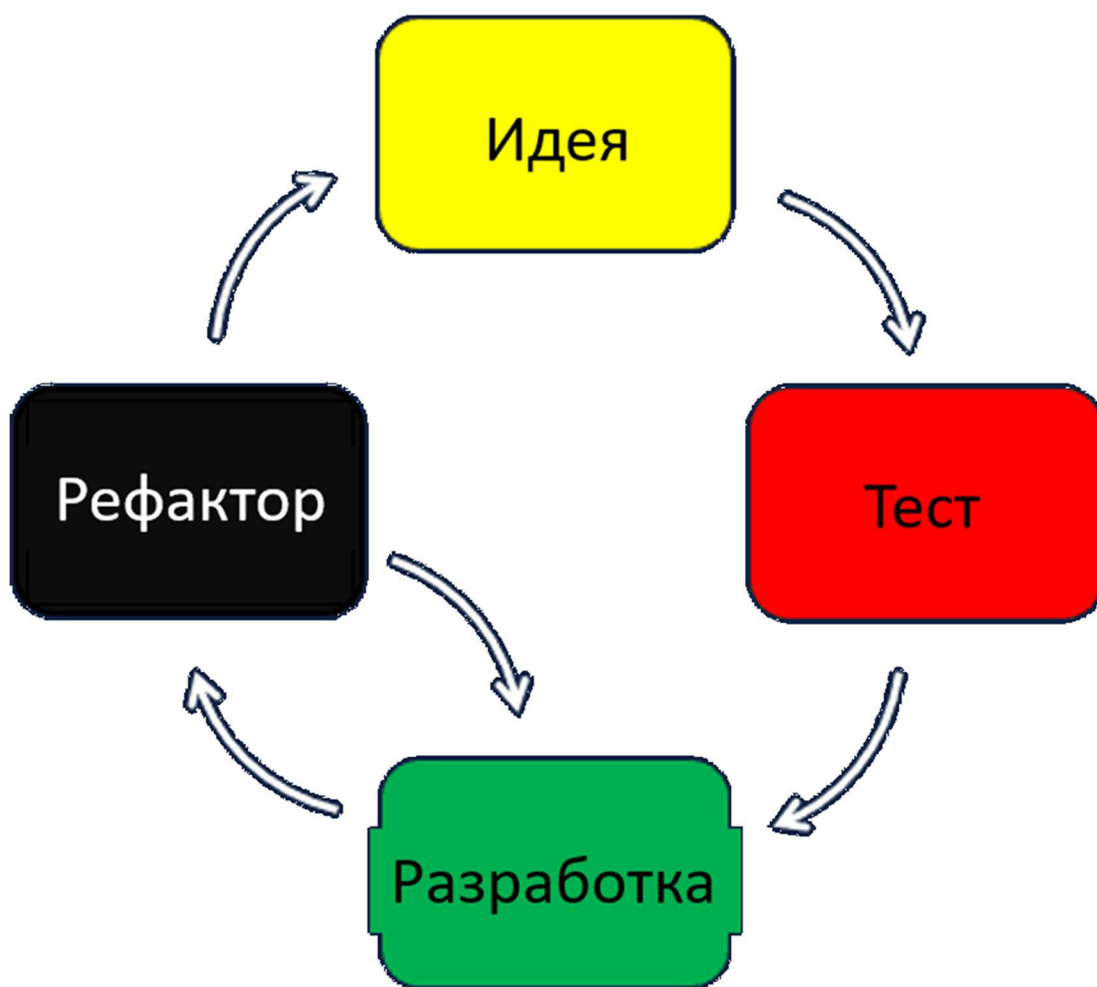


Рисунок 2 – Цикл разработки по методу TDD

Также перед реализацией прорабатываемой возможности, разработчики убеждаются в том, что тест не выдает ложноположительные результаты, это — своего рода тестирование самих тестов. Далее происходит реализация запланированной идеи и ее проверка разработанным тестом. Это позволит лучше понять, что требуется сделать разработчикам и проверить работоспособность кода сразу, как только он будет написан.

Привлечение заказчика в процесс разработки [18] позволяет исключить проблему недостатка компетенции в предметной области, в рамках которой ведется разработка программного продукта. Также привлечение заказчика позволит ускорить процесс получения обратной связи об использовании продукта и упростить процесс согласования функционала элементов продукта.

Методы **Crystal** также являются составляющей Agile методологий, направленных на повышение эффективности и результативности команд разработчиков, тесном сотрудничестве стейкхолдеров, эффективном общении, простоте, итеративной разработке [19]. Одной из ключевых особенностей методов Crystal является адаптируемость. Так как среди методов Crystal существуют 4 варианта с различным уровнем структуризации, в зависимости от масштаба проекта, у команд есть возможность выбирать и адаптировать свою практику в соответствии со своими конкретными потребностями, предпочтениями и навыками, а не придерживаться жесткого набора процедур. Методы Crystal подразумевают бережливый подход с минимальной документацией, вместо этого уделяя особое внимание прямому общению и сотрудничеству как основным средствам передачи информации и решения проблем. Команды поощряются к самоорганизации и самоуправлению, при этом руководители проектов и лидеры выступают в качестве координаторов и наставников, а не направляют или контролируют их деятельность.

Методология бережливого производства (**Lean**), как и прочие методологии Agile, включает в себя организацию итеративного подхода, открытую коммуникацию стейкхолдеров, улучшение потока процессов [20]. В методологии Lean выделено 4 основных принципа.

«Сосредоточьтесь на ценности»: методология бережливого производства делает упор на предоставлении ценности клиенту, сосредоточение внимания на разработке элементов, которые решают проблему заказчика и приносят ему пользу [21]. Расставляя приоритеты в отношении ценности, команда может избежать напрасной траты времени и ресурсов на функции, которые не являются необходимыми или не повышают ценность проекта.

Непрерывное совершенствование: проекты машинного обучения требуют постоянного совершенствования по мере поступления новых данных или получения новой информации. Методология бережливого производства поощряет постоянное совершенствование путем регулярного анализа хода

реализации проекта и внесения необходимых коррективов. Это позволяет команде адаптироваться к меняющимся требованиям и выпускать более качественный продукт.

Сотрудничество: методология бережливого производства поощряет сотрудничество между членами команды и заинтересованными сторонами. В проектах машинного обучения такое сотрудничество имеет важное значение для обеспечения точности разработанных моделей и соответствия потребностям заказчика. Работая сообща, команда может выявлять проблемы на ранней стадии и устранять их до того, как они перерастут в более серьезные проблемы.

Минимизация отходов: методология бережливого производства направлена на минимизацию отходов путем устранения ненужной работы или процессов. В проектах машинного обучения это означает сосредоточение внимания на разработке эффективных моделей. Сводя к минимуму количество отходов, команда может создавать более качественный продукт за меньшее время.

1.3 Методики управления проектами машинного обучения

Помимо методологий управления программными проектами в разработке используются методологии для решения конкретных задач. Для проектов машинного обучения это сбор и обработка данных, формулирование и экспериментальная проверка гипотез, обучение моделей, оценка результатов. Такими методологиями являются CRISP-DM [22], SEMMA[23], KDD Process[24]. Данные методологии не регламентируют роли стейкхолдеров, график встреч, итеративный подход и прочие правила, присущие методологиям Agile. Они устанавливают критерии и последовательность выполнения задач отдельно взятого разработчика в своей сфере.

Проекты, связанные с машинным обучением (ML, MO), имеют свою специфику, отличающую их от проектов по разработке программных

продуктов [25]. Основное отличие заключается в объединении с наукой о данных [26, 27], подразумевающее использование научного подхода в этапах разработки [28]. Важными критериями методологии для управления проектами машинного обучения являются визуализация рабочего процесса, ограничение незавершенного производства, постоянное совершенствование, совместный подход и ориентация на потребности клиентов [29]. Эти критерии необходимы для управления многочисленными этапами обработки данных, разработки моделей и тестирования, обеспечения ценности разрабатываемого продукта для заказчика [30]. В отличие от традиционной разработки программного обеспечения, где программисты явно задают правила и логику работы программы, в машинном обучении модели "обучаются" на основе данных, чтобы делать предсказания или принимать решения.

В проектах машинного обучения большое внимание уделяется подготовке и очистке данных, выбору подходящих алгоритмов обучения, оптимизации параметров моделей и оценке их качества. Также важным аспектом является работа с большими объемами данных и выбор подходящих инструментов для их обработки. Кроме того, в проектах машинного обучения часто используются специализированные инструменты для работы с данными, обучения моделей и их развертывания в продакшн-среде [25]. В таблице 1 представлены сходства и различия проектов по разработке программных продуктов, включающих процессы машинного обучения, от проектов по разработке прочих программных продуктов.

Таблица 1 – Сходства и различия ML проектов с прочими программными проектами

Сходства проектов ML с прочими проектами	Различия проектов ML с прочими проектами
Проекты требуют хорошего понимания требований заказчика и пользователей, а также бизнес-задач, которые должен решать разрабатываемый продукт	Проекты с машинным обучением требуют специализированных знаний в области алгоритмов машинного обучения, статистики и обработки данных.
Необходимо определить архитектуру программного продукта, выбрать подходящие технологии и инструменты для его разработки.	Разработка программного продукта с использованием машинного обучения связана с необходимостью обработки больших объемов данных и подготовкой их для обучения модели
Проекты требуют тестирования и отладки продукта перед его выпуском.	Проекты с машинным обучением требуют проведения экспериментов для определения зависимостей
Подходят для Agile методологий	Модель машинного обучения и обучающая выборка могут требовать многократной постройки
	Разработка проекта с машинным обучением подразумевает проведение исследований и экспериментов, выдвижение и проверку гипотез.

Таким образом, для управления проектами с машинным обучением из рассмотренных ранее методологий лучшим образом подходят ScrumBan, Lean, так как они нацелены на сосредоточение внимания к ценности разрабатываемого продукта для заказчика, внутрикомандного взаимодействия, контроля потока задач и визуализации текущего процесса. Использование при управлении проектом с машинным обучением согласно одной из этих методологий порядка разработки, описанного в CRISP-DM заранее определит и упростит планирование спринтов, формирование Backlog проекта, выбор критериев успешности проекта.

1.4 Условия внеплановой разработки

Одной из проблем бизнеса является периодическое отсутствие временных ресурсов для реализации проектов, выполнение которых необходимо в рамках ближайшего ограниченного периода (месяц, квартал, полугодие). Отсутствие его в планах на данный период не позволяет снизить количество ресурсов для реализации уже запланированных проектов. Такие проекты можно назвать внеплановыми.

Один из способов реализации таких проектов является использование сверхурочного времени сотрудников для проведения разработки. В данном случае работодатель не способен обязать сотрудников в работе сверхурочно [31]. Однако, согласно опросу [32], сотрудники готовы задерживаться для работы сверхурочно, однако график сверхурочной работы в данном случае не может быть обязательным. Таким образом, порядка 60 % сотрудников привлекается к сверхурочной работе не более 2 часов в день.

При рассмотрении случая разработки внепланового проекта в области машинного обучения, согласно методологиям ScrumBan и Lean, команда разработки состоит в среднем из 5 человек, значит над внеплановым проектом в лучшем случае будет трудиться порядка 3 человека 2 часа в день. Согласно формуле (1), это составляет 30 человеко-часов в неделю.

$$T_k = D_k \cdot P_{cc} \cdot P_c, \quad (1)$$

где T_k – человеко-часы работы над внеплановым проектом;

D_k – число календарных дней в периоде;

P_{cc} – средне-списочная численность разработчиков;

P_c – продолжительность сверхурочной занятости, часов.

Инструменты и правила, определяемые методологиями ScrumBan и Lean, подходят для асинхронной, ограниченной по человеческим и временным ресурсам разработки. Использование доски для визуализации процесса разработки позволяет получить актуальную информацию о задаче, выполненной другим разработчиком в день отсутствия трудящихся

сотрудников в текущий момент. Ежедневные короткие обсуждения выполненной ранее части работы и задач, принятых к выполнению в текущий момент, позволяют разработчикам иметь актуальную информацию о целостной картине разработки. Отслеживание и сокращение количества задач, не приближающих команду к достижению цели проекта, а также сосредоточенность на ценности разрабатываемого продукта для заказчика позволяет оптимизировать временной ресурс для соблюдения или сокращения срока разработки. Обе методологии позволяют быстро адаптироваться к изменениям в процессе разработки и сосредоточиться на приоритетных задачах. Они ставят целью оптимизацию потока работы, минимизацию времени цикла и устранение избыточных процессов. Это особенно важно при работе с ограниченными ресурсами.

2 Разработка правил методики управления внеплановыми проектами в области машинного обучения

2.1 Разработка продукта машинного обучения

Методика для управления внеплановыми проектами в области машинного обучения должна быть адаптирована к асинхронной работе сменяющих друг друга разработчиков, доступным трудозатратам до 30 человеко-часов в неделю, полному циклу разработки продукта машинного обучения.

Согласно работе [33], разработку продукта машинного обучения можно разделить на 4 основных этапа:

- 1) анализ данных;
- 2) подготовка к обучению;
- 3) обучение модели и оценка результатов;
- 4) внедрение обученной модели в систему продукта.

Анализ данных является этапом, определяющим вводные данные проекта. На данном этапе рассматриваются техническое задание к продукту, ценность его для заказчика, цель разработки по критериям SMART [36]. В измеримости цели следует определять метрики для анализа результатов обучения модели. Анализируются имеющиеся и подготовленные для разработки составляющие, такие как:

- подходящая для задачи заказчика модель машинного обучения;
- набор данных для обучения модели, его качество и обработка;
- уровень готовности конечного продукта для внедрения обученной модели.

При анализе данных также следует оценить достижимость поставленных целей, исходя из имеющихся ресурсов трудозатрат и запланированного объема задач.

Во время подготовки к обучению выполняется настройка среды разработки, установка необходимых библиотек для работы с данными, настройка параметров обучения модели. Также выполняется подготовка данных для обучения. Полный цикл состоит из следующих этапов:

- поиск, либо производство необходимых для проекта данных;
- разделение выборки на обучающую, валидационную и тестовую составляющие;
- обработка данных для загрузки в обучаемую модель [34].

На этапе обучения выбранной модели происходит загрузка подготовленных данных в программу, обеспечивающую их представление в векторном виде и выполняющую алгоритм установления зависимостей [35]. По окончании обучения модели выполняется оценка ее работы по определенным на этапе анализа данных критериям, тестирование на наличие ложноположительных результатов. По окончании анализа результатов обучения принимается решение о завершении работы над обучением модели, либо о проведении дополнительных экспериментов, связанных с доработкой обучающей выборки и изменением параметров обучения модели, либо модели в целом. Так как изначально неизвестно, какое количество итераций данного этапа потребуется для достижения поставленной цели, на данный этап отводится больший процент времени по сравнению с прочими.

Внедрение обученной модели в систему продукта является заключительным этапом, на котором выполняется разработка, не связанная с машинным обучением. В зависимости от определенной при анализе данных цели, выполняется разработка программной части, обеспечивающей достаточный уровень функционирования разработанной модели. К примеру, на данном этапе возможны встраивание обученной модели в некий автономный программный продукт, разработка API-сервиса для обращения по сети, разработка web-приложения для работы с обученной модели с помощью интерфейса и др.

Длительность всех этапов определяется сроком разработки проекта и может варьироваться от недели до нескольких месяцев. Достижимость целей и возможность выполнения задач проекта за поставленный срок определяется на этапе анализа данных.

2.2 Организация процесса разработки

Процесс разработки организуется с использованием следующих элементов ScrumBan:

- процесс разработки разбивается на спринты;
- перед каждым спринтом проводится встреча стейкхолдеров проекта;
- в Backlog спринта заносятся задачи, реализация которых осуществима за неделю;
- ежедневные стендап митинги перед началом сверхурочной работы для синхронизации текущих этапов разработки с актуальными данными;
- ведение KanBan доски для визуализации процесса разработки.

Длительность спринта составляет неделю, так как асинхронность ведения разработки, редкая коммуникация внутри команды разработки, растяжение трудозатрат одного дня на неделю способствуют смещению фокуса с направления разработки и ценности продукта, провоцировать разработку излишнего функционала. Принимая в расчет, что сотрудник не каждый день готов к сверхурочной работе, близкое завершения спринта и уменьшенный срок выполнения задач мотивируют разработчиков не откладывать их выполнение.

Встреча стейкхолдеров проекта перед каждым спринтом необходима для оценки результатов предыдущего спринта и их согласованности с требованиями заказчика, определения задач начавшегося спринта. Перед началом работы над проектом на встрече стейкхолдеров определяются цели проекта по критериям SMART [36], решаемая с помощью разрабатываемого

продукта проблема заказчика, возможность реализации и критерии завершенности проекта. Это позволяет определить границы выполнения проекта и уточнить, что именно необходимо заказчику.

В каждом спринте определяются задачи отдельному взятому разработчику, выполнение которых возможно за его длительность. Для равномерного распределения нагрузки на разработчиков, а также последующей оценки трудозатрат как отдельно взятых разработчиков, так и проекта в целом, каждой задаче определяется вес, соответствующий затраченным на нее человеко-часам.

Доска KanBan является инструментом, позволяющим как визуализировать процесс разработки на общем и персональном уровнях, так и передавать информацию между членами команды.

Использование правил и требований методологии Lean позволяет сократить срок разработки путем предупреждения появления ошибок на всем этапе разработки программного продукта. Для этого может использоваться разработанный ранее подход TTD. Наибольший эффект от данного подхода достигается при условии, что команда разработки имеет опыт работы с ним. Его внедрение и первичное обучение занимает дополнительное время, однако в последующем, когда команда разработки научится его применять, разработка становится быстрее и устойчивее к возможным ошибкам. Скорость разработки в данном случае обеспечивается за счет выявления проблем и ошибок на ранних этапах, что способствует их быстрому исправлению и предотвращению более серьезных проблем в будущем.

При отсутствии у команды наработок по какому-либо из четырех этапов разработки продукта машинного обучения, спринты распределяются в процентном соотношении 20 – 25 – 35 – 20 соответственно. Данное распределение времени обусловлено прогнозируемыми трудозатратами каждого из этапов разработки. Выбор и подготовка данных перед обучением является одним из важнейших этапов обучения модели машинного обучения. Объем обучающей выборки, баланс классов и качество распределения данных

наибольшим образом влияют на качество работы обученной модели и являются управляемыми параметрами, поэтому требует выделения больших временных ресурсов.

Этап обучения модели и оценки результатов подразумевает проведение экспериментов с настройкой модели и подготовке обучающей выборки в случае недостижения определенного на этапе анализа данных критерия успешности обучения модели. Выделенная на данный этап увеличенная доля времени проекта необходима для повторения действий этапа подготовки к обучению в целях доработки данных и перенастройки алгоритмов обучаемой модели с учетом результатов предыдущего обучения.

Временное соотношение 20 – 25 – 35 – 20 между этапами разработки продукта обеспечивает более детальную проработку непосредственно модели машинного обучения. При необходимости разработки сложной системы, в которой предполагается использование обученной модели, к примеру, многостраничного web-сайта или обработчика цифрового видеосигнала, следует увеличить долю времени, выделенную на последний этап, до 40 % максимум.

3 Апробация методики при управлении разработкой продукта

3.1 Управление студенческим проектом

Для апробации разработанной методики управления внеплановыми проектами в области машинного обучения был выбран студенческий проект «Фото идентификация тигров. Модель определения стороны тигра(лево/право)», условия которого соответствуют условиям внепланового проекта. Проектный практикум подразумевает разработку или исследование группой студентов темы, предложенной компаниями-партнерами высшего учебного заведения. В зависимости от сложности проекта и объема необходимых работ, проект может длиться не менее одного семестра, а в команде разработки состоит от трех до семи человек. Семестровая работа разделяется на 4 итерации. График обучения определяет 4 академических часа в неделю для выполнения задач проектного практикума, однако не регламентирует строгого времени для занятий.

Был произведен расчет времени, которое разработчик отводит для выполнения задач по проектному практикуму, путем опроса участников команды разработки. Каждый студент из команды трудоустроен на основную работу с графиком работы 5–2 с 8:00 до 17:00, а также посещает лекционные и практические занятия по прочим дисциплинам. В соответствии с анализом, представленным на рисунке 3, ресурс команды разработки для работы над проектом составляет порядка 22 часа в неделю.

Время на проект в неделю, часов

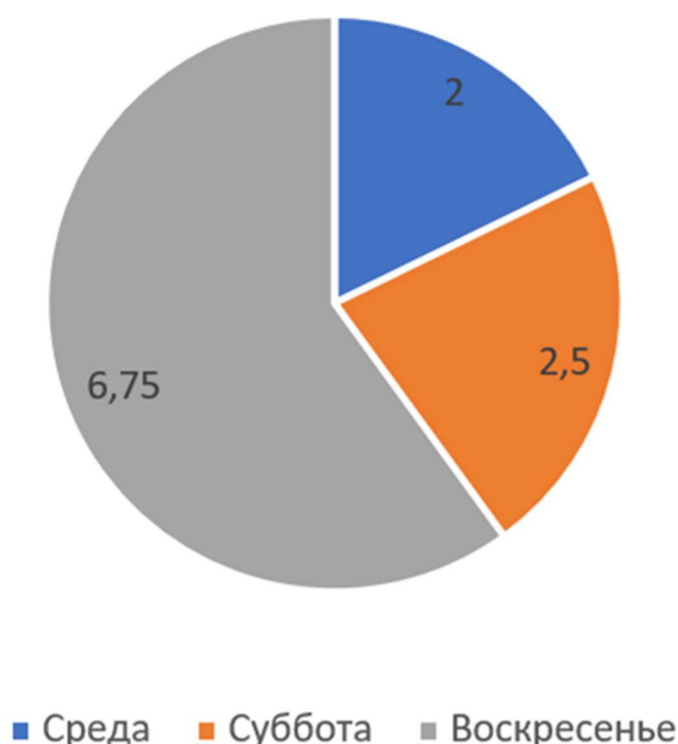


Рисунок 3 – Время, уделяемое студентами для работы над проектом

Условия работы в рамках данного проекта близки к условиям внеплановых проектов по машинному обучению: численность команды разработки и трудозатраты учебного проекта сопоставимы с данными величинами сверхурочной работы, срок разработки ограничен одним кварталом, проект включает в себя полный цикл разработки программного продукта с машинным обучением, разработка ведется асинхронно, деление учебного проекта на 4 итерации соответствует разделению разработки продукта машинного обучения.

Отличается студенческий проект от сверхурочной работы уровнем квалификации команды разработки и графиком работы разработчиками над проектом в течение дня.

В начале первой итерации состоялась встреча стейкхолдеров проекта, на которой были определены цели и задачи проекта по критериям SMART.

Конкретная цель проекта – обучить нейронную сеть классифицировать положение детектированного на фото тигра по следующим классам:

- право;
- лево;
- другое.

Измеримость заключается в обучении как минимум одной модели МО. Точность работы модели МО должна оцениваться по метрике F1.

Достижимость цели обеспечивается следующими факторами:

- доступен датасет [37] из 5487 фотографий тигров с разных сторон, среди данных присутствуют координаты ограничительной рамки, однако их поля на них занимают порядка 30 % изображения;
- доступны модели МО с открытым исходным кодом, подходящие для обучения;
- доступны научные работы с описанием использования и исследования вышеперечисленных моделей МО;
- доступны бесплатные для студентов вычислительные ресурсы для обучения моделей МО.

Проект ограничен во времени длительностью учебного семестра. Цель должна быть достигнута к 30 декабря 2023 года.

Обратная связь обеспечивается путем оценки выполнения каждого этапа на соответствие заявленных в начале спринта требованиям к результату. Синхронизация с заказчиком и согласование выполненных и предстоящих на каждый спринт задач.

Среди требований к проекту, подлежащих к пересмотру, было выделено изменение количества классов модели МО.

Также необходимо определить цель каждой из четырех итераций. Для проектов машинного обучения они были выбраны одинаковыми для каждого проекта:

- 1) поиск актуальной литературы о способах решения поставленной задачи МО, поиск набора данных и выбор доступных моделей МО;
- 2) подготовка набора данных для обучения (далее – датасет), написание кода программы для загрузки датасета и запуска обучения моделей МО;
- 3) обучение моделей МО, анализ полученных метрик. Проведение экспериментов с датасетом, параметрами обучения моделей МО;
- 4) оформление отчета о результатах обучения модели МО, подготовка файлов проекта для передачи заказчику.

В ходе первой итерации (этап анализа данных) были проанализированы модели МО для классификации изображений [38]. Важным критерием выбора была возможность отключения либо отсутствие функции построения векторов с учетом горизонтального отзеркаливания. После проведения технического анализа были выбраны модели Vision Transformer (далее – ViT) [39, 40, 41] и ConvNeXt [42, 43]. ViT – это модель для классификации изображений, основанная на архитектуре трансформеров. Авторы работы [39] продемонстрировали, что чистый трансформер, примененный непосредственно к последовательностям патчей изображений, может хорошо справляться с задачами классификации изображений без необходимости использования сверточных сетей.

Поиск датасета в данном проекте не проводился, так как заказчиком был предоставлен датасет ATRW [37], содержащий более пяти тысяч фотографий тигров. Среди данных для изображений датасета присутствуют рамки границ тигра с шириной полей порядка 10 %, как представлено на рисунке 4, и набор ключевых точек для двух тысяч изображений.



Рисунок 4 - Изображение датасета с широкими полями

Так как в датасете отсутствует классификация сторон тигров, и задача по созданию данных входит в проект, было выдвинуто два варианта ее решения. Первый заключается в просмотре членами команды всех изображений подряд и выборе класса вручную. Второй вариант заключался в использовании координат ключевых точек для разработки алгоритма определения стороны автоматически.

Задача по разработке алгоритма автоматического определения стороны была выгоднее с точки зрения временных ресурсов, так как она занимала 3 человеко-часа, а ручная разметка командой всего набора данных – 18 человеко-часов. В ходе первого часа выполнения задачи выяснилось, что в некоторых случаях качество разметки координат ключевых точек не соответствует действительности, как представлено на рисунке 5.

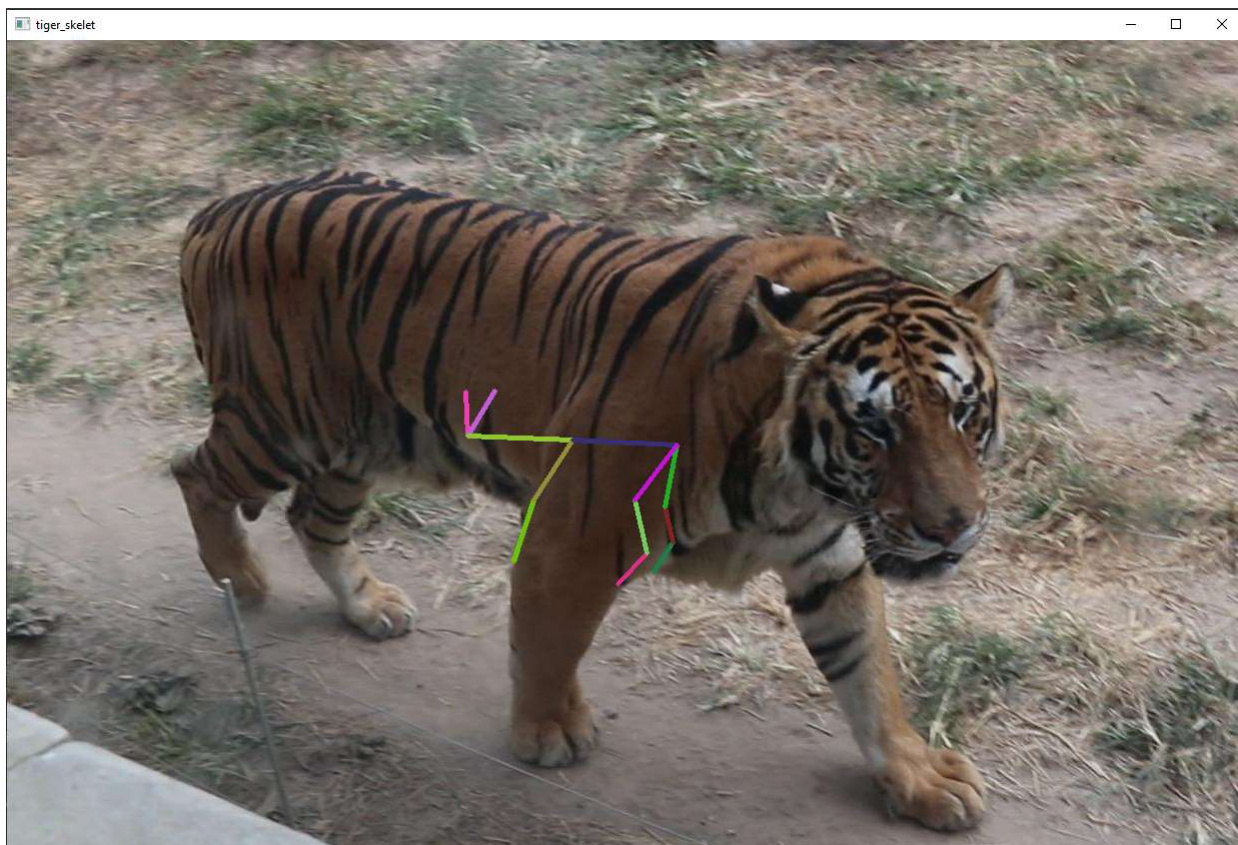


Рисунок 5 – Неверная разметка ключевых точек тигра

Так как задачи вручную проверять результаты размеченных автоматически двух тысяч изображений и вручную размечать пять тысяч изображений сопоставимы по затратам человеко-часов, было решено отказаться от варианта автоматической разметки классов стороны по координатам ключевых точек. Данная задача назначена на вторую итерацию.

К концу первой итерации были выполнены следующие задачи:

- определены модели ViT и ConvNeXt для обучения классификации сторон тигра;
- создан блокнот с моделью ViT;
- проверена и отвергнута гипотеза о автоматической разметке датасета с помощью имеющихся координат ключевых точек;
- создан проект в сервисе Roboflow [44] для ручной разметки фотографий на классы сторон.

Во время встречи стейкхолдеров после последнего спринта первой итерации также определены задачи для второй итерации:

- 1) разметка датасета на roboflow;
- 2) создание блокнота с моделью ConvNeXt;
- 3) обучение модели ViT на сформированном датасете;
- 4) обучение модели ConvNeXt;
- 5) анализ полученных метрик F1, разработка гипотез по улучшению.

Задача разметки датасета была распределена равномерно на всех членов команды. В ходе выполнения было отмечено, что не удастся однозначно определить класс, когда тигр расположен полубоком, к примеру на рисунке Рисунок 6.



Рисунок 6 – Неоднозначное расположение тигра

Было решено увеличить количество классов с целью увеличения точности работы модели на такого рода изображения. Были добавлены классы

«перед», «зад», «верх», «право с неполным рисунком», «лево с неполным рисунком», «право с нечитаемым рисунком», «лево с нечитаемым рисунком». После выполнения разметки датасета было проанализировано количество изображений всех классов в датасете, представленное на рисунке 7.



Рисунок 7 - Количество изображений всех классов в датасете

Для обучения модели классификатора репрезентативность каждого класса должна быть близка к остальным [45, 46]. Во время общения с кураторами заказчика было решено снизить количество классов, оставив «лево», «право», «перед», «другое». Неразмеченными остались изображениями, на которых представлено несколько тигров. Таким образом, был сформирован и размечен датасет из 4342 изображений.

Создание блокнота с ConvNeXt не было завершено, так как кураторы заказчика внесли дополнительное требование к проекту: обучить модель ResNet18 [47]. Так как блокнот для обучения ViT уже был подготовлен к работе, а ConvNeXt был на этапе анализа доступных механизмов, принято решение отказаться от ConvNeXt в пользу ResNet18.

Обучение модели ViT на подготовленном датасете было завершено успешно, полученное значение F1 метрики составило 0,8965. Были выявлены проблемы блокнота, связанные с заменой классификационного слоя предобученной модели ViT, отображением метрики по ходу обучения, загрузкой данных в модель. В связи с этим принято решение о подготовке другого блокнота для модели ViT. После обучения обновленного блокнота модели ViT F1 метрика составила 0,929.

Во время обсуждения полученных результатов с кураторами заказчиков было определено, что необходимо провести аугментацию обучающих данных [48, 49] с целью повышения стабильности работы модели на некачественных изображениях.

Таким образом, к концу второй итерации были выполнены следующие задачи:

- 1) размерен датасет для обучения моделей МО;
- 2) создан блокнот для обучения модели ResNet18;
- 3) обновлен блокнот для обучения модели ViT;
- 4) проведено обучение модели ViT на обновленном блокноте.

На третью итерацию было запланировано выполнение следующих задач:

- 1) обучение модели ResNet18;
- 2) аугментация обучающих данных;
- 3) обучение ViT и ResNet18 на аугментированных данных;
- 4) анализ полученных результатов работы моделей, обученных на аугментированных данных.

Для обучения модели ResNet18 во время предыдущих итераций были подготовлены блокнот и датасет, поэтому задача по ее обучению была завершена на первом спринте третьей итерации. Метрика F1 составила 0,899. Так как в течение второй итерации был выполнен переход с модели МО ConNeXt на ResNet18, произошла рассинхронизация параллельной работы с двумя моделями. Таким образом, в начале третьей итерации модель ViT была готова к экспериментам по аугментации обучающих данных, а ResNet18 только предстояло обучить на датасете. Основываясь на работе [50] и целях проекта, было принято решение не дожидаться завершения обучения модели ResNet18 и проводить задачи по аугментации обучающих данных параллельно на двух моделях, а начать проводить их на модели ViT с начала итерации.

Исходя из описания заказчиком возможных проблем с качеством поступающих на вход модели изображений, для аугментации данных были выбраны техники эластичного преобразования, преобразования в черно-белый

цвет, размытия изображения, случайной обрезки изображения. Аугментация датасета с помощью вышеописанных техник проводилась как независимо друг от друга, так и одновременно. При использовании независимых аугментаций метрика F1 увеличивалась в среднем на 0,02 для каждой техники. В случае объединения техник метрика для каждой модели снижалась в среднем на 0,03.

Четвертая итерация была посвящена анализу трудозатрат команды разработки, проведению ретроспективы и определению успешности завершения проекта.

Используя данные для оценки управления проектом, приведенные в п. 2 и представленные в приложении А. Трудозатраты разработчиков, представленные на рисунке 8, составляют в среднем 19,5 человеко-часов за спринт. Как видно по линии тренда, трудозатраты разработчиков сокращаются по мере приближения к финалу проекта.

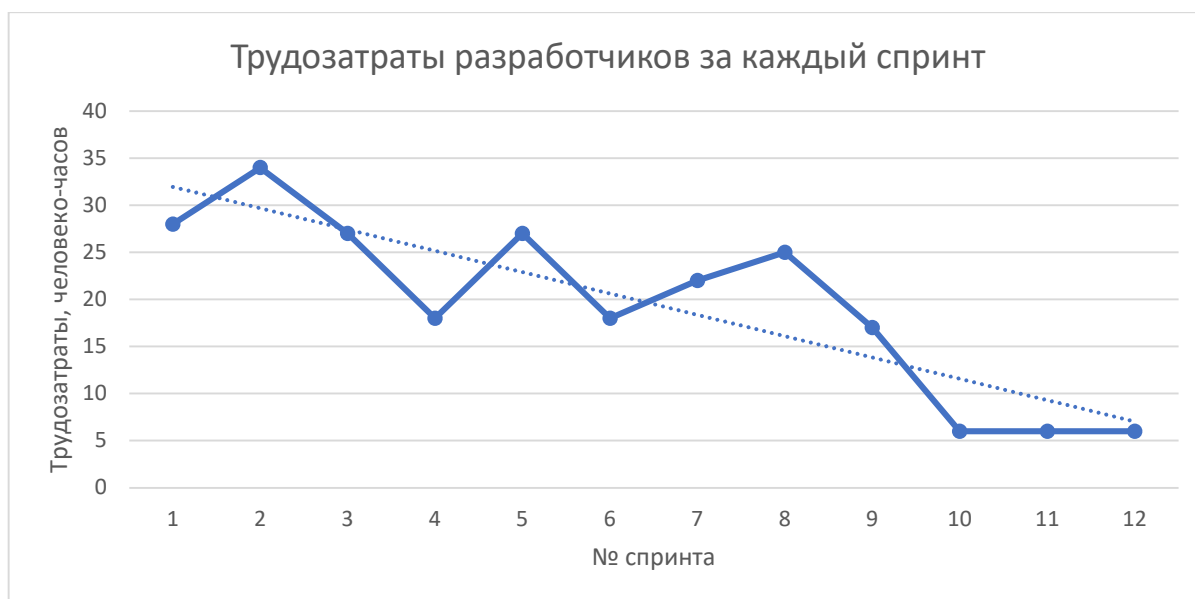


Рисунок 8 – Трудозатраты разработчиков за каждый спринт

Также были оценены трудозатраты спланированных задач, выполнение которых не принесло пользы для достижения целей проекта. Их соотношение представлено на рисунке 9. Задачи, соответствующие данным трудозатратам,

были направлены на разметку датасета для обучения моделей МО, а также анализу и подготовки модели MO ConvNeXt.



Рисунок 9 - Излишние трудозатраты в спринте

Несмотря на наличие потерь человеко-часов, цель проекта удалось достичь, соблюдши все критерии SMARTER.

В ходе апробации разработанной методики для управления внеплановыми программными проектами в области машинного обучения установлено, что, руководствуясь ею, команде студентов из 5 человек удалось успешно завершить проект «Фотоидентификация тигров. Модель определения стороны». Ключевыми отличиями учебного проекта от внепланового производственного проекта являлись:

- уровень квалификации разработчиков;
- использование общедоступных сервисов для машинного обучения;
- согласование с заказчиком не только целей и требований проекта, но и применяемых технологий;
- предоставление заказчиком набора данных для обучения.
- отсутствие задачи по встраиванию обученной модели в конечный продукт.

3.2 Масштабирование разработанной методики

Для масштабирования методики на внеплановые производственные проекты изменений не требуется. Низкий уровень квалификации студентов, обучающихся инженерии искусственного интеллекта, относительно разработчиков IT-компаний был компенсирован за счет технической помощи специалистов со стороны заказчика. Таким образом, для использования разработанной методики необходимо организовать KanBan доску команды разработки, обеспечить ежедневную сверхурочную работу части специалистов в соответствии с трудовым кодексом Российской Федерации [31], организовать еженедельную встречу стейкхолдеров проекта для подведения результатов предыдущего спринта и определения задач следующего с точки зрения приближения проекта к достижению поставленных целей.

ЗАКЛЮЧЕНИЕ

В данной работе были проанализированы существующие методики для управления программными проектами и проектами в области машинного обучения, разработана методика для управления внеплановыми проектами в области машинного обучения, выполнена апробация разработанной методики при управлении проектом «Фото идентификация тигров. Модель определения стороны тигра(лево/право)», условия которого соответствуют условиям внеплановых проектов, определена пригодность методики для масштабирования к использованию на предприятиях.

В ходе анализа рассмотренных методик для управления программными проектами установлено, что они плохо применимы для условий внеплановых проектов. Работа над внеплановым проектом в разные дни недели ведется разными разработчиками, в том числе над одной задачей, не более 2 часов в день. В существующих же методиках определена работа по графику стандартизированного рабочего дня с синхронной работой разработчиков. Также были проанализированы методики для управления проектами в области машинного обучения. Они определяют очередность этапов и технические аспекты разработки продукта, однако не содержат правил по организации и управлению процессом разработки.

Была разработана методика для управления внеплановыми проектами в области машинного обучения, содержащая правила по организации взаимодействия между стейкхолдерами проекта, порядок и этапы разработки продукта машинного обучения, инструменты для визуализации процессов и асинхронного взаимодействия разработчиков. Временные интервалы в рамках методики рассчитаны для условий внеплановых проектов:

- 1) ежедневная работа 30% команды разработчиков;
- 2) ежедневное рабочее время 2 часа;
- 3) ежедневная ротация разработчиков при выполнении задачи.

Разработанная методика была опробована при управлении проектом «Фото идентификация тигров. Модель определения стороны тигра(лево/право)». Использование разработанной методики при управлении разработкой позволило исключить этапы, которые не являются необходимыми для достижения цели, а также выполнять задачи, не привязываясь к разному графику работы членов команды, сохранив при этом последовательность и завершенность. Не используя методику, возможно, в ходе проекта могли возникать ситуации многократного выполнения одной задачи, ожидания информации о результатах завершенной задачи, разработки излишнего функционала и др. Отсутствие четкой методики может привести к неопределенности в определении целей и ожиданий от продукта. Это может привести к тому, что разработка будет направлена не в том направлении, которое требуется заказчику или пользователям. Без методики управления проектом может быть сложно эффективно распределять ресурсы, такие как время, бюджет и персонал. Это может привести к излишней затрате времени и ресурсов на непродуктивные задачи. Отсутствие методики управления проектом может привести к недостаточному контролю качества продукта. Это может привести к выявлению проблем на поздних этапах разработки, что затруднит их исправление и повысит риск недовольства заказчика. Методика управления проектом предусматривают структурированную коммуникацию внутри команды и с заказчиком. Без этого может возникнуть недопонимание и неполнота информации, что отразится на результате проекта.

По результатам проекта были обучены и протестированы две модели классификатора, определяющих сторону тигра на изображении. Цели проекта были достигнуты в полном объеме. При управлении проектом согласно методике, были отмечены следующие замечания:

- в первом, втором и четвертом спринтах было затрачено 17 человеко-часов для выполнения задач, не приблизивших команду разработки к цели проекта;

– в ходе последней итерации трудозатраты составляли 6 человеко-часов, так как проект не требовал внедрения обученной модели в основной продукт.

Установлено, что для масштабирования проекта под производственные проекты машинного обучения введения дополнительных правил не требуется. Выполненный проект был отличен от внепланового квалификацией сотрудников и графиком работы над проектом. Хотя квалифицированным разработчикам, в отличие от студентов, не требуется техническая обратная связь со стороны заказчика, еженедельное представление и обсуждение результатов спринта необходимо в процессе разработки. Обзор спринта не позволяет разработчикам откладывать работу и уходить от поставленных целей и ценностей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Волкодаева А. В., Балановская А. В., Чулков А. В. Тенденции востребованности специалистов сферы информационных технологий на рынке труда в России // НК. 2021. Вып. 10, №3. – С. 174-188.
2. Разработка и управление в программных проектах. О программе : сайт. Екатеринбург. – URL: <https://programs.edu.urfu.ru/ru/10276/> (дата обращения: 12.03.2024).
3. Котельников, И. К. Учет внеплановых работ при оценке статуса проекта по методу освоенного объема / И. К. Котельников // Достижения науки и образования. – 2016. – № 7(8). – С. 63-65. – EDN WGXCER.
4. Ali Bidhendi, Mohammad Mahoud. Chapter VIII: Artificial Intelligence and Machine Learning in Project Management / Ali Bidhendi, Mohammad Mahoud // The Future of Project Management: Adapting to Modern Needs. – Cambridge Scholars Publishing. – 218 с.
5. Андрианов, А. М. Об актуальности применения проектного обучения на примере проектного командного обучения (пко) / А. М. Андрианов // Системы компьютерной математики и их приложения. – 2020. – № 21. – С. 360-367. – EDN PUUSIY.
6. Winston W. Royce. «Managing the Development of Large Software Systems». MIT : mitpress, 2021. – С. 321-332.
7. Жук, А. И. Управление задачами в Agile и Waterfall / А. И. Жук // Студенческий. – 2019. – № 1-1(45). – С. 35-37. – EDN YWDCQX.
8. Шалин, Н. А. Анализ методологии разработки по waterfall с помощью стандарта OMG essence / Н. А. Шалин // Лучший научно-исследовательский проект : Сборник статей Международного научно-исследовательского конкурса, Санкт-Петербург, 15 апреля 2020 года / Под редакцией Р.Д. Иванова. – Санкт-Петербург: Индивидуальный предприниматель Иванов Роман Дмитриевич, 2020. – С. 13-16. – EDN VGLCQA.

9. Васильев, Н. И. Каскадная модель или Waterfall: насколько актуален и применим в современном мире? / Н. И. Васильев // СОВЕРШЕНСТВОВАНИЕ МЕТОДОЛОГИИ ПОЗНАНИЯ В ЦЕЛЯХ РАЗВИТИЯ науки : сборник статей Национальной (Всероссийской) научно-практической конференции, Екатеринбург, 22 января 2022 года. – Уфа: Общество с ограниченной ответственностью "ОМЕГА САЙНС", 2022. – С. 17-19. – EDN KSCSUN.

10. A Comparative Study of Agile Software Development Methodology and traditional waterfall model [текст] / Rupali Pawar // Software Engineering, 2015. – С. 1-8.

11. State-of-the-Art: A Systematic Literature Review of Agile Project Management in Information Systems Development [текст] / Markus Hummel // Proceedings of the 2014 47th Hawaii International Conference on System Sciences. – 2014. – С. 4712-4721.

12. Дюсембекова, Г. А. Методы управления и планирования при разработке системы оценивания эффективности сотрудников IT компании - Scrum, Kanban, Waterfall / Г. А. Дюсембекова, Р. С. Аксанов // Студенческий. – 2021. – № 20-1(148). – С. 64-67. – EDN TWWGTF.

13. Project Management Methodologies: A Comparative Analysis [текст] / Christina Chin, Eng Hwa Yap, Andrew C Spowage // Journal for the Advancement of Performance Information and Value. – Vol. 4, N 1. – 2012. – С. 106-118.

14. Брагин К. И., Агапитов Д. В., Колташев Я. А. Экстремальное программирование как метод развития гибких и профессиональных навыков студентов it-отрасли // Вестник СибГУТИ. 2023. Вып. 17, № 2. – С. 12-22.

15. D. Sihombing. Development of Engineering Team Performance Management Applications in Apartments Using Extreme Programming Methods // Jurnal Teknologi Terapan G-Tech. 2024. – Vol. 8. – С. 521-527.

16. M. U. Pervez, L. Eman. Test Driven Development: A Review. National University of Sciences and Technology, 2022. – 4 с.

17. Agha D., Sohail R., Meghji A. F., Qaboolio R., Bhatti S. Test Driven Development and Its Impact on Program Design and Software Quality: A Systematic Literature Review. – VAWKUM Transactions on Computer Sciences, 2023. – Vol. 11, № 1. – С. 268–280.
18. Rudakova L. The process of managing customer knowledge in business structures and its role in predicting the development of their needs. // экономика и управление: проблемы, решения: журнал. – 2024. – Vol. 1. – С. 51-58.
19. Anwer F., Aftab Sh., Waheed U., Muhammad S. Agile Software Development Models TDD, FDD, DSDM, and Crystal Methods: A Survey. // INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING: журнал. – 2017. – Vol. 8. – С. 1-10.
20. Agile to Lean Software Development Transformation: a Systematic Literature Review [текст] / Bruno Rossi, Filip Kišš // 2018 Federated Conference on Computer Science and Information Systems. – 2018. – С. 969-973.
21. Kondate Roja, Muthuswamy Shanthi, Rojas-Santiago Miguel. Process improvements using lean methodologies in a subassembly line. / Kondate Roja, Muthuswamy Shanthi, Rojas-Santiago Miguel // International Journal of Six Sigma and Competitive Advantage : журнал. – 2020. – № 1. – 13 с.
22. Brzozowska J., Pizoń J., Baytikenova G., Gola A., Zakimova A., Piotrowska K. Data engineering in crisp-dm process production data – case study. // Applied Computer Science: журнал. –2023. – Vol. 19. – С. 83-95.
23. Omari F. S. A combination of SEMMA & CRISP-DM models for effectively handling big data using formal concept analysis based knowledge discovery: A data mining approach // World Journal of Advanced Engineering Technology and Sciences: журнал. –2023. – Vol. 8. – С. 9-14.
24. Funes A., Dasso A. Data Mining and the KDD Process // Advanced Methodologies and Technologies in Network Architecture, Mobile Computing, and Data Analytics. – 2019. – С. 515-231.

25. Challenges in the deployment and operation of machine learning in practice [текст] / Lucas Baier, Fabian Jöhren, Stefan Seebacher // 27th European Conference on Information Systems, May 2019.

26. Amarachukwu Felix, Sai Peck Lee. Systematic Literature Review of Preprocessing Techniques for Imbalanced Data / Amarachukwu Felix, Sai Peck Lee // IET Software : журнал. – Wiley, 2019. – № 13(2). – С. 479-496.

27. Mohamed Abdeldaiem Abdelhadi Mahboub, Pyla Srinivasa Rao, T.Gopi Krishna. Innovative Approaches to Enhance Data Science Optimization / Mohamed Abdeldaiem Abdelhadi Mahboub, Pyla Srinivasa Rao, T.Gopi Krishna // International Journal of Innovative Science and Research Technology : журнал. – 2023. – Vol. 8(11). – С. 964-970.

28. Chengwu Liu. FIMO: A Challenge Formal Dataset for Automated Theorem Proving / Chengwu Liu, Jianhao Shen, Huajian Xin [и др.]. // Arxiv, Cornell University, 2023. – 2 изд. – 8 с.

29. Emmanuel A. Building Machine Learning Powered Applications: Going from Idea to Product: монография. / A. Emmanuel – 1-е изд. – O'Reilly Media, 2020, – 256 с. – ISBN 978-1492045113.

30. Brasjö C., Lindovsky M. Machine Learning Project Management. A Study of Project Requirements and Processes in Early Adoption: Master's Thesis in Master's Programme International Project Management Department of Architecture and Civil Engineering / Christoffer Brasjö, Martin Lindovsky ; «Chalmers University Of Technology» – Gothenburg, Sweden, 2019. – 89 с. – URL: <https://publications.lib.chalmers.se/records/fulltext/257253/257253.pdf>

31. Российская Федерация. Законы. Трудовой кодекс : Федеральный закон № 197-ФЗ : [Принят Государственной Думой 21 декабря 2001 года : Одобрен Советом Федерации 26 декабря 2001 года] – Москва: 2024. – 259 с.

32. Ведомости : сайт. – Москва. – URL: <https://www.vedomosti.ru/society/news/2021/11/19/896664-bolshe-pолоvini-rossiyan-priznalis-v-pererabotke> (дата обращения: 13.03.2024).

33. Harvard Business Review : сайт. – Гарвард. – URL: <https://hbr.org/2024/01/getting-machine-learning-projects-from-idea-to-execution>
34. Iqbal H. Sarker. Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective. / Iqbal H. Sarker // SN Computer Science : журнал. – 2021. – №2(5). – 22 с.
35. Iqbal H. Sarker. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions / Iqbal H. Sarker // SN Computer Science : журнал. – 2021. – № 2. – С. 419-439.
36. Акбарова С.А. Постановка целей по методике SMART и как она влияет на мотивацию сотрудников / С.А. Акбарова // COLLOQUIUM-JOURNAL : журнал. – 2019. – № 2-4. – С. 34-36.
37. CVWC 2019 : сайт. – Сеул, Корея. – URL: <https://cvwc2019.github.io/challenge.html>
38. Хабр : сайт. – URL: <https://habr.com/ru/articles/666314/> (дата обращения: 02.03.2024)
39. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale / A. Dosovitskiy, L. Beyer, A. Kolesnikov [и др.] // ICLR, 2021. – 22 с.
40. Yang Liu. A Survey of Visual Transformers / Yang Liu, Yao Zhang, Yixin Wang, Feng Hou [и др.]. Arxive, Cornell University, 2021. – изд. 2. – 21 с.
41. Nicolas Carion. End-to-End Object Detection with Transformers / Nicolas Carion, Francisco Massa, Gabriel Synnaeve [и др.]. // Arxive, Cornell University, 2020. – 26 с.
42. Li Xin Jed Lim, Ziming Qiu. μ -Net: ConvNext-Based U-Nets for Cosmic Muon Tomography / Li Xin Jed Lim, Ziming Qiu // Cornell University, 2023. – 22 с.
43. Arohan Ajit. A Review of Convolutional Neural Networks / Arohan Ajit, Koustav Acharya, Abhishek Samanta // Conference: 2020 International Conference on Emerging Trends in Information Technology and Engineering. – 5 с.

44. Roboflow : сайт. – URL: <https://app.roboflow.com> (дата обращения: 02.03.2024)
45. Ting Li, Xiaofeng Chen, Weikai Li. An Improved AdaBoost Method in Imbalanced Data Learning / Ting Li, Xiaofeng Chen, Weikai Li // Xi'an, China, International Conference on Cyber-Physical Social Intelligence, 2023. – С. 405-410.
46. Seba Susan, Amitesh Kumar. The balancing trick: Optimized sampling of imbalanced datasets – A brief survey of the recent State of the Art / Seba Susan, Amitesh Kumar // Engineering Reports : журнал. – 2020. – № 3(04). – 24 с.
47. Deep Residual Learning for Image Recognition / Kaiming He, Xiangyu Zhang, Shaoqing Ren [и др.] // – Microsoft Research, 2015. – 12 с.
48. Моисеева Е. Д. Аугментация изображений / Е. Д. Моисеева // Российская наука: тенденции и возможности : сборник научных статей. Том Часть 4. – Москва : Издательство "Перо", 2020. – С. 129-132. – EDN XCQUCS.
49. Гайер, А. В. Аугментация обучающей выборки "на лету" для обучения нейронных сетей / А. В. Гайер, А. В. Шешкус, Ю. С. Чернышова // Труды Института системного анализа Российской академии наук. – 2018. – Т. 68, № S1. – С. 150-157. – DOI 10.14357/20790279180517. – EDN YOQRFR.
50. Harold Kerzner. Project Management : монография / Harold Kerzner. – 12-е изд. – Wiley, 2017. – 597 с. – ISBN 9781119165354.

ПРИЛОЖЕНИЕ А

Анализ методик для управления программными проектами

Таблица А1 – Анализ методик для управления программными проектами

Методология	Достоинства для ml	Достоинства для внеплановых проектов	Недостатки для ml	Недостатки для внеплановых проектов
Waterfall		Ориентирование на первоначальные цели, выполнение ограниченного объема работы и жесткие сроки выполнения	Не подразумевает циклических экспериментов и тестирований	Процесс разработки не синхронизируется как внутри команды разработчика, так и между прочими стейкхолдерами
Lean	Способствует улучшению качества продукта, минимизации потерь, адаптивности к изменениям и фокусу на потребностях клиента.	Ставит потребности клиента на первый план, что помогает разработчикам машинного обучения создавать модели, которые соответствуют потребностям и ожиданиям клиентов	Предполагает оптимизацию процессов и устранение избыточности, что не совместимо с необходимостью проведения множества экспериментов и тестирований в процессе разработки.	
		Нацелена на минимизацию потери времени на излишнюю деятельность и	Не предусматривает достаточной гибкости для быстрой адаптации к изменениям требований и условий проекта, а также низкой квалификации разработчиков	

Окончание таблицы А1

Scrum	Способствует гибкости и адаптивности, итеративному подходу, прозрачности и коммуникации, а также фокусу на ценностях продукта	Ставит на первый план ценности продукта, что помогает разработчикам машинного обучения создавать модели, которые соответствуют потребностям и ожиданиям клиентов.	Предполагает разбиение проекта на короткие итерации и четкое определение задач для каждой из них. В проектах машинного обучения может потребоваться проведение множества экспериментов и тестирований, которые могут занять значительное количество времени и не подходят для коротких итераций.	Ставит на первый план прозрачность и коммуникацию между участниками проекта, что трудно достижимо в условиях проектного практикума
ScrumBan	Способствует визуализации процесса разработки, управлению потоком работы, оптимизации процесса разработки, прозрачности и коммуникации, а также фокусу на поставленных задачах.	Помогает визуализировать процесс разработки, что позволяет работать асинхронно без потери информации	-	-
Экстремальное программирование	Обеспечивает надежность разработки путем исключения ошибок	-	-	Требуется больших трудозатрат для выполнения одной задачи
				Требуется постоянного присутствия заказчика в ходе разработки

ПРИЛОЖЕНИЕ Б

Трудозатраты проекта

Таблица Б1 – Трудозатраты проекта

№ спринта	Трудозатраты разработчиков, человеко-часов												
	ML инженер		ML инженер		Инженер датасета		ML инженер		ML инженер		Менеджер проекта		Общие трудозатраты
	Завершены без пользы	Завершены успешно	Завершены без пользы	Завершены успешно	Завершены без пользы	Завершены успешно	Завершены без пользы	Завершены успешно	Завершены без пользы	Завершены успешно	Завершены без пользы	Завершены успешно	
1	2	6		6	3	2	2	6	2	6	9	2	46
2		6	2	8	2	4		6		6	4	4	42
3		4		6		6		4		3	0	4	27
4	2	3		3	2	3		3		3	4	3	26
5		4		5		5		4		6	0	3	27
6		2		4		3		2		3	0	4	18
7		3		5		2		4		3	0	5	22
8		6		3		4		2		5	0	5	25
9		3		2		3		2		3	0	4	17
10		1		1		1		1		1	0	1	6
11		1		1		1		1		1	0	1	6
12		1		1		1		1		1	0	1	6