

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc175070561)

[АНАЛИТИЧЕСКАЯ ЧАСТЬ 8](#_Toc175070562)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 8](#_Toc175070563)

[1.1 Анализ подразделения ХХХХХХХХ организации ХХХХХ 8](#_Toc175070564)

[1.1.1 Дерево бизнес-направлений организации 8](#_Toc175070565)

[1.1.2 Сопоставление бизнес-процессов и критических факторов успеха организации 9](#_Toc175070566)

[1.1.3 Анализ структуры и нормативной документации, регламентов подразделения «ххххххххххх» организации / университета, регулирующих выполнение выбранного бизнес-процесса 10](#_Toc175070567)

[1.2 Моделирование бизнес-процесса ххх хххххх ххххххх ххххххххххххххххх 11](#_Toc175070568)

[1.2.1 Моделирование ххххххххххххх “КАК ЕСТЬ” 11](#_Toc175070569)

[1.2.2 Моделирование процесса “КАК ДОЛЖНО БЫТЬ” 18](#_Toc175070570)

[1.3 Анализ рынка программного обеспечения для автоматизации бизнес-процесса ХХХХХХХХХХХХ 21](#_Toc175070571)

[1.4 Анализ стейкхолдеров и их требований к разрабатываемой системе 22](#_Toc175070572)

[1.5 Выбор средств разработки 22](#_Toc175070573)

[1.6 Техническое задание на разработку корпоративной информационной системы 23](#_Toc175070574)

[1.7 Выводы по разделу 23](#_Toc175070575)

[ПРОЕКТНАЯ ЧАСТЬ 24](#_Toc175070576)

[2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОЕКТА 24](#_Toc175070577)

[2.1 Структурирование требований к разрабатываемой системе 24](#_Toc175070578)

[2.2 План разработки ПО 26](#_Toc175070579)

[2.3 Разработка прототипа модели DL с описанием ПЛАНИРУЕМОГОхххххх хххх – ) Это один из вариантов названия раздела. Не нужно всё копировать 26](#_Toc175070580)

[2.3.1 Формирование набора данных хххх хххх хх хххххх 26](#_Toc175070581)

[2.3.2 Выбор модели DL и инструментальных средств интеллектуального ххх хх хххх 26](#_Toc175070582)

[2.4 Обучение модели DL 28](#_Toc175070583)

[2.5 Разработка/модификация/разработка программного модуля/приложения (участие в разработке прикладной библиотеки, разработка локальных политик сетевой безопасности и т.п.) – КОНЕЧНЫЙ ВАРИАНТ ИС ПОД ВКР 28](#_Toc175070584)

[2.6 Разработка прототипа модели DL с описанием ПЛАНИРУЕМОГОхххххх хххх – ) Это один из вариантов названия раздела. Не нужно всё копировать 30](#_Toc175070585)

[2.6.1 Формирование набора данных хххх хххх хх хххххх 30](#_Toc175070586)

[2.6.2 Выбор модели DL и инструментальных средств интеллектуального ххх хх хххх 30](#_Toc175070587)

[2.7 Обучение модели DL 31](#_Toc175070588)

[2.8 Разработка/модификация/разработка программного модуля/приложения (участие в разработке прикладной библиотеки, разработка локальных политик сетевой безопасности и т.п.) – КОНЕЧНЫЙ ВАРИАНТ ИС ПОД ВКР 32](#_Toc175070589)

[2.8.1 Разработка модели доступа к данным 33](#_Toc175070590)

[2.8.2 Тестирование разработанного ПО 34](#_Toc175070591)

[2.8.3 План внедрения и развертывания ПО 34](#_Toc175070592)

[2.9 Руководства администратора и пользователя корпоративной информационной системы 34](#_Toc175070593)

[2.10 Выводы по главе 2 35](#_Toc175070594)

[ЭКОНОМИЧЕСКАЯ ЧАСТЬ 36](#_Toc175070595)

[3 ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ОТ РАЗРАБОТКИ ИС 36](#_Toc175070596)

[3.1 Расчет затрат на разработку ИС 36](#_Toc175070597)

[3.2 Выбор и обоснование методики расчёта экономической эффективности 36](#_Toc175070598)

[3.3 Оценка затрат на разработку и внедрение АИС 36](#_Toc175070599)

[3.3.1 Затраты на этапе разработки информационной системы 39](#_Toc175070600)

[3.3.2 Затраты на этапе внедрения 41](#_Toc175070601)

[3.3.3 Затраты на этапе эксплуатации 42](#_Toc175070602)

[3.4 Эффект от внедрения АИС 42](#_Toc175070603)

[3.5 Экономический эффект 44](#_Toc175070604)

[3.6 Социальный эффект 46](#_Toc175070605)

[3.7 Научный эффект 46](#_Toc175070606)

[3.8 Организационный эффект 46](#_Toc175070607)

[3.9 Эффективность внедрения АИС (ПО ПРИМЕРУ) 47](#_Toc175070608)

[3.10 Расчёт показателей экономической эффективности проекта (ПО ПРИМЕРУ) 48](#_Toc175070609)

[3.11 Выводы по главе 3 56](#_Toc175070610)

[ЗАКЛЮЧЕНИЕ 58](#_Toc175070611)

[СПИСОК ЛИТЕРАТУРЫ 59](#_Toc175070612)

[ПРИЛОЖЕНИЯ 60](#_Toc175070613)

[Приложение 1. Техническое задание на разработку корпоративной электронной библиотеки в университете 60](#_Toc175070614)

[Приложение 2. Исходный код “Авторизация” 66](#_Toc175070615)

[Приложение 3. Руководство администратора корпоративной информационной системы 70](#_Toc175070616)

[Приложение 4. Руководство пользователя корпоративной информационной системы 73](#_Toc175070617)

**Введение**

В современном мире информационные технологии активно внедряются в различные сферы деятельности, значительно упрощая выполнение задач и оптимизируя рутинные процессы. Одной из таких сфер является шахматы — игра, которая требует высокого уровня стратегического мышления, анализа и обработки данных. В условиях роста популярности шахмат как образовательного инструмента и спортивной дисциплины возникает необходимость автоматизации процессов анализа шахматных партий, обучения новичков и разработки инструментов для профессионалов.

Одной из ключевых задач в этой области является распознавание шахматных фигур на изображениях реальных шахматных досок. Это сложная техническая задача, требующая применения современных методов компьютерного зрения и машинного обучения. Традиционные подходы к анализу шахматных партий часто ограничиваются ручным вводом данных или использованием устаревших алгоритмов, которые не способны адаптироваться к сложным сценариям, таким как различие между похожими фигурами, обработка изображений с различным фоном, углами съемки и освещением.

Современные технологии позволяют решать такие задачи с высокой точностью. Например, использование нейронных сетей, сверточных архитектур (CNN) и глубокого обучения открывает новые возможности для автоматизации анализа шахматных партий. Интеллектуальная система, основанная на этих технологиях, может автоматически определять тип фигуры (пешка, конь, слон, ладья, ферзь, король) и её цвет (белый или черный), что значительно упрощает работу аналитиков, тренеров и любителей шахмат.

Актуальность данной работы обусловлена несколькими факторами:

1. Образовательная потребность: Шахматы активно используются в образовательных программах для развития логического мышления, стратегического планирования и аналитических навыков. Автоматизация анализа партий позволяет создавать персонализированные учебные материалы для студентов.
2. Спортивная сфера: Профессиональные шахматисты и тренеры нуждаются в инструментах для анализа партий, выявления ошибок и разработки новых стратегий.
3. Технологическая востребованность: Разработка интеллектуальных систем распознавания шахматных фигур может быть интегрирована в мобильные приложения, онлайн-платформы и системы дополненной реальности, что расширяет их функциональность и удобство использования.

Кроме того, современные методы машинного обучения и компьютерного зрения находят широкое применение в различных областях, таких как медицина, транспорт, финансы и розничная торговля. Например, в медицине они используются для анализа рентгеновских снимков, в транспорте — для создания автономных транспортных средств, а в розничной торговле — для анализа покупательского поведения. Подобные технологии могут быть успешно применены и в шахматах, где требуется точное распознавание объектов на изображениях.

Одной из основных технологий, используемых в данной работе, являются сверточные нейронные сети (Convolutional Neural Networks, CNN). Эти сети особенно эффективны для задач, связанных с обработкой изображений, так как они способны автоматически извлекать признаки из данных и использовать их для классификации. В частности, CNN могут быть обучены распознавать шахматные фигуры на основе их визуальных характеристик, таких как форма, цвет и текстура.

Для обучения модели используется датасет, содержащий изображения шахматных фигур с различными условиями освещения, углами съемки и фонами. Это позволяет модели научиться работать с реальными условиями эксплуатации и обеспечивает её высокую точность. Кроме того, для улучшения качества модели применяются методы аугментации данных, такие как изменение размера, поворот и добавление шума.

Важно отметить, что разработка интеллектуальной системы распознавания шахматных фигур имеет не только теоретическую, но и практическую значимость. Например, такая система может быть использована в образовательных платформах для автоматизации анализа партий и создания интерактивных учебных материалов. Она также может быть интегрирована в мобильные приложения для любителей шахмат, что позволит им анализировать свои партии и улучшать навыки игры.

Целью данной выпускной квалификационной работы является разработка интеллектуальной системы распознавания шахматных фигур, способной автоматически определять тип фигуры (пешка, конь, слон, ладья, ферзь, король) и её цвет (белый или черный) с использованием современных методов машинного обучения и компьютерного зрения.

Для достижения поставленной цели необходимо решить следующие задачи

* Провести анализ предметной области для выявления основных требований к системе.
* Выбрать подходящие методы машинного обучения и компьютерного зрения.
* Разработать модель, способную работать с различными условиями освещения, углами съемки и фонами.
* Создать удобный пользовательский интерфейс для взаимодействия с системой.
* Оценить эффективность разработанной системы и её применимость в реальных сценариях.

Объект исследования: процессы автоматизированного анализа и классификации шахматных фигур на изображениях.  
Предмет исследования: программная система распознавания шахматных фигур, основанная на методах глубокого обучения.

Научная новизна заключается в комплексном применении современных методов компьютерного зрения и машинного обучения для автоматического распознавания шахматных фигур на изображениях реальных шахматных досок. Предложенная модель позволяет достичь высокой точности классификации и адаптации к различным условиям съемки.

В работе подробно рассматриваются этапы разработки системы, выбор технологий, обучение модели и тестирование её производительности. Особое внимание уделяется практической применимости системы в образовательной, спортивной и развлекательной сферах.

# АНАЛИТИЧЕСКАЯ ЧАСТЬ

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

##### Аналитический обзор методологии компьютерного зрения и распознавания шахматных фигур

### Описание сферы применения компьютерного зрения при распознавании шахматных фигур

Компьютерное зрение представляет собой одну из наиболее активно развивающихся областей искусственного интеллекта, ориентированную на обработку и анализ цифровых изображений. В последние годы оно стало неотъемлемой частью многих информационных систем, позволяя автоматизировать задачи, ранее выполняемые человеком. Среди таких задач — классификация объектов, распознавание текста, сегментация изображений и другие. Особенно актуально применение технологий компьютерного зрения в тех случаях, когда требуется высокая точность и устойчивость к изменению внешних условий, таких как освещение, ракурс съёмки и фоновые помехи.

Одним из перспективных направлений использования компьютерного зрения является автоматизация анализа шахматных партий. Современные шахматные приложения всё чаще сталкиваются с необходимостью распознавать фигуры на фотографиях реальных шахматных досок. Это позволяет пользователям быстро получать информацию о текущей позиции, строить анализ игры и использовать её для обучения или подготовки к соревнованиям. Однако задача распознавания шахматных фигур на изображениях является сложной и требует применения современных методов машинного обучения, способных эффективно решать такие задачи.

Основным подходом к решению задачи распознавания объектов на изображениях являются сверточные нейронные сети (CNN), которые демонстрируют высокую точность в задачах классификации и анализа визуальных данных. Эти модели способны автоматически извлекать признаки из изображений, что делает их идеальным выбором для решения задач в области компьютерного зрения. В рамках данной работы предполагается использование предобученной архитектуры EfficientNetB0, которая была дообучена на специализированном наборе данных, содержащем изображения шахматных фигур, сделанные в различных условиях. Выбор этой архитектуры обусловлен её высокой производительностью и умеренными вычислительными требованиями, что позволяет использовать модель даже на устройствах со средней мощностью. Для повышения устойчивости модели к различным условиям съёмки и увеличения разнообразия данных применялась аугментация изображений. К ней относится случайное изменение яркости, повороты, сдвиги, масштабирование и горизонтальное отражение. Такие преобразования позволили улучшить качество обучения и повысить точность модели при работе с изображениями, сделанными в неидеальных условиях. Кроме того, использовались стандартные методы регуляризации, такие как Dropout и BatchNormalization, что помогло снизить риск переобучения и улучшить общую устойчивость модели. Процесс обработки изображений включал несколько этапов. На первом этапе проводилась предобработка изображения — нормализация яркости, изменение размера до заданного формата и удаление шума. Далее следовало извлечение визуальных признаков, таких как форма, цветовая палитра и текстурные характеристики. После этого осуществлялось определение типа фигуры и её цвета с помощью обученной модели глубокого обучения. Финальным этапом являлась постобработка результатов, включающая фильтрацию и проверку достоверности полученных данных.

Важной задачей также являлось создание пользовательского интерфейса, обеспечивающего удобство взаимодействия с системой. Основная цель интерфейса заключалась в предоставлении пользователю возможности загрузить изображение, получить результат классификации и просматривать историю запросов. Интерфейс был реализован с использованием библиотеки CustomTkinter, что позволило создать современный и интуитивно понятный графический интерфейс. Пользователь может взаимодействовать с системой как через кнопки, так и с помощью функции drag-and-drop. Также были добавлены элементы визуализации — графики распределения по классам и цветам, что даёт возможность пользователю наглядно оценить статистику использования системы.

Система предназначена как для профессиональных шахматистов, так и для начинающих игроков. Она может быть полезна в образовательных целях, предоставляя возможность анализа партий и формирования учебных материалов. Кроме того, система может быть интегрирована в мобильные и веб-приложения, что значительно расширяет её функциональные возможности и удобство использования. Анализ существующих решений показывает, что большинство аналогичных программ либо требуют значительных вычислительных ресурсов, либо имеют ограниченную применимость в реальных условиях. Разрабатываемая система направлена на устранение этих недостатков и предлагает более гибкий и доступный вариант.

Примером успешного внедрения технологий компьютерного зрения являются программы Stockfish и AlphaZero, используемые для анализа шахматных позиций. Хотя эти системы работают с уже известными позициями и не используют изображения, они демонстрируют высокую эффективность в анализе игровых ситуаций, что служит основой для дальнейшего развития ИИ-инструментов в шахматах. Ещё одним примером является приложение ChessVision, которое позволяет пользователю загрузить изображение шахматной доски и получить информацию о расположении фигур. Такие приложения показывают потенциал интеграции компьютерного зрения в игровые и образовательные платформы.

В рамках дипломной работы особое внимание уделялось обеспечению устойчивости модели к различным факторам внешней среды. Были учтены такие параметры, как угол съёмки, наличие фоновых объектов, уровень освещённости и возможная схожесть некоторых фигур. Для этого на этапе подготовки данных использовались изображения, собранные в разных условиях, что позволило модели лучше адаптироваться к реальным сценариям использования. Также была проведена работа по формированию набора данных, его очистке и разделению на обучающую и валидационную выборки.

На основе проведённого анализа можно сделать вывод, что применение современных методов компьютерного зрения и глубокого обучения открывает широкие возможности для автоматизации процессов анализа шахматных партий. Разрабатываемая система позволяет не только точно определять тип и цвет шахматных фигур, но и сохранять историю запросов, предоставлять статистику и визуализировать данные. Это делает её универсальным инструментом, который может быть использован как в спортивной практике, так и в образовательных целях. В дальнейшем возможно расширение функционала за счёт интеграции с мобильными устройствами, системами дополненной реальности и облачными сервисами.

### Анализ структуры и нормативной документации, регламентов подразделения университета, регулирующих выполнение выбранного бизнес-процесса

Автоматическое распознавание шахматных фигур представляет собой важное направление развития современных образовательных и аналитических систем. Оно позволяет значительно повысить эффективность процессов анализа шахматных партий, упростить обучение новичков и создать удобные инструменты для профессиональных игроков. Внедрение таких технологий в образовательную среду открывает возможности для автоматизации обработки данных, повышения качества предоставляемых услуг и улучшения пользовательского опыта. Особенно актуально это становится в условиях роста интереса к шахматам как к интеллектуальной дисциплине и спортивной деятельности, требующей постоянного совершенствования навыков и стратегического мышления.

Однако, несмотря на очевидные преимущества, внедрение систем компьютерного зрения и машинного обучения сталкивается с рядом сложностей. Одним из ключевых факторов, влияющих на успешность реализации системы, является её устойчивость к различным внешним условиям. Уровень освещённости, наличие посторонних объектов на изображении, угол съёмки и даже дизайн самих фигур могут существенно влиять на точность распознавания. Например, при плохом освещении или съёмке под нестандартным углом модель может допустить ошибку в определении типа или цвета фигуры. Также возникают трудности при работе с нестандартными наборами фигур, где классические формы заменены на декоративные или арт-дизайны. Это требует дополнительной обработки изображений и более тщательной настройки алгоритмов, чтобы система могла корректно функционировать в реальных условиях использования.

Важнейшими критериями успешного внедрения являются точность, скорость работы, уровень автоматизации и возможность интеграции с уже существующими платформами. Точность модели должна быть достаточно высокой, чтобы обеспечивать надёжное распознавание фигур в различных условиях. Скорость обработки изображений также играет важную роль, поскольку задержки могут негативно отразиться на восприятии системы пользователем. Не менее значимым фактором является возможность интеграции с текущими программными решениями, применяемыми в образовательной или игровой среде. Наконец, минимизация необходимости ручного вмешательства делает систему более экономически эффективной и практичной в использовании.

Для успешного внедрения разрабатываемого программного обеспечения необходимо провести анализ структуры и действующих нормативных документов университет связанного с реализацией образовательных программ и использованием информационных технологий. Университет имеет чётко организованную структуру, включающую преподавательский состав, методистов и технический персонал, что создаёт предпосылки для грамотного внедрения новых решений. Деятельность подразделения регулируется внутренними положениями, такими как:

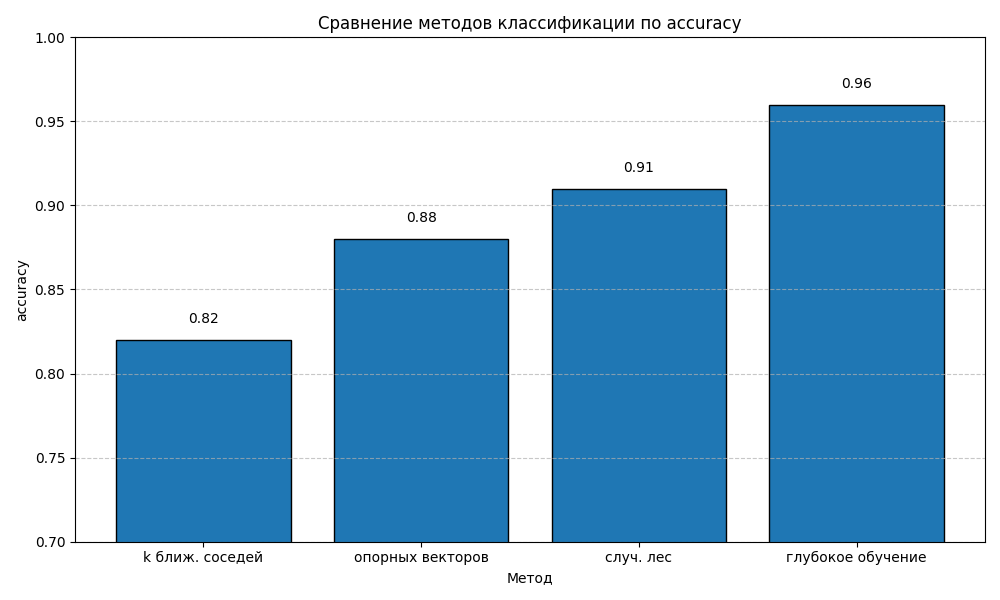
* Положение о применении ИКТ в образовательном процессе
* Положение о научно-методической деятельности
* Регламенты использования программного обеспечения
* Документы по информационной безопасности и защите данных

Эти документы устанавливают требования к качеству предоставляемых образовательных услуг, порядку применения цифровых технологий, хранению информации и обеспечению её доступности. При этом особое внимание уделяется тому, чтобы внедряемые технологии соответствовали целям и задачам учебного заведения, были совместимы с имеющейся инфраструктурой и соответствовали стандартам защиты данных.

Кроме того, необходимо учитывать, что внедрение программных продуктов в образовательной сфере должно проходить этапы тестирования, экспертной оценки и методического согласования. Это позволяет избежать конфликтов между новыми технологиями и традиционными подходами к обучению, а также обеспечивает их полезность с точки зрения педагогики и практики. Таким образом, при разработке системы были приняты во внимание не только технические, но и организационные аспекты, которые оказывают влияние на её дальнейшее внедрение и использование.

Разработка интеллектуальной системы распознавания шахматных фигур проводилась с ориентацией на ключевые бизнес-процессы, такие как анализ шахматных партий, подготовка учебных материалов, организация тренировочных сессий и интеграция с мобильными и веб-приложениями. Эти процессы находятся в тесной взаимосвязи с основными целями организации, поэтому их автоматизация способна принести наибольшую пользу. Особый акцент был сделан на автоматизацию анализа шахматных партий, поскольку именно этот процесс лежит в основе обучения и подготовки игроков. Интеграция технологии компьютерного зрения в данный процесс позволяет повысить точность анализа, ускорить получение результатов и снизить нагрузку на преподавательский состав.

Таким образом, внедрение системы автоматического распознавания шахматных фигур требует не только технической реализации, но и внимательного рассмотрения нормативно-правовой базы, структуры подразделения и действующих регламентов. Только при соблюдении всех этих условий возможно эффективное использование системы в образовательной практике и достижение стратегических целей организации.



#### 1.1.3 Роль и востребованность интеллектуальных систем распознавания шахматных фигур

Центр дистанционных образовательных технологий (ЦДОТ) Российского нового университета играет важную роль в обеспечении высококачественного дистанционного образования. Он предоставляет студентам доступ к разнообразным учебным материалам, включая графические объекты — такие как изображения шахматных досок и позиций. В связи с ростом популярности шахмат как инструмента развития логического мышления и аналитических навыков, возникает необходимость автоматизации работы с такими данными. Интеллектуальные системы распознавания шахматных фигур позволяют эффективно решать задачи анализа шахматных партий, автоматизации обучения и повышения качества образовательного контента.

Применение таких систем значительно упрощает обработку графических материалов. Традиционные методы требуют ручного анализа изображений, что занимает много времени и сопряжено с риском человеческих ошибок. Система, основанная на компьютерном зрении и глубоком обучении, позволяет сократить временные затраты, повысить точность анализа и создать условия для автоматической генерации обучающих материалов. Это особенно важно при подготовке интерактивных курсов, где требуется оперативное получение информации о текущей позиции на доске.

Кроме того, внедрение технологии распознавания способствует улучшению пользовательского опыта. Образовательные платформы могут использовать информацию о типе и цвете фигур для фильтрации партий, поиска похожих позиций и построения рекомендаций. Такие функции делают платформу более удобной и функциональной, особенно для начинающих игроков, которым сложно самостоятельно классифицировать сложные шахматные позиции.

Важным аспектом является возможность персонализации обучения. На основе автоматически полученных данных система может предлагать индивидуальные задания, отслеживать прогресс пользователя и корректировать уровень сложности материалов. Это позволяет адаптировать обучение под уровень подготовки каждого студента, повышая эффективность освоения материала и мотивацию к дальнейшему обучению.

Интеграция системы распознавания шахматных фигур в существующую информационную среду ЦДОТ обеспечивает бесперебойную работу и взаимодействие с другими модулями образовательной платформы. Полученные данные используются не только для интерфейса пользователя, но и для аналитики, статистики, рекомендательных механизмов и формирования отчётов преподавателям. Таким образом, система становится частью единой экосистемы цифрового обучения, усиливая её возможности.

Моделирование бизнес-процесса «КАК ЕСТЬ» по методике IDEF0 (см. рисунок 1.3) позволило описать текущее состояние обработки графических материалов в ЦДОТ. Основной узел диаграммы отражает этапы:

1. Приём изображения шахматной доски.
2. Выделение характерных признаков: форма, цвет, расположение фигур.
3. Обработка и классификация.
4. Передача результатов в образовательную платформу или архив.

На начальном этапе осуществляется проверка качества изображения и исключение дубликатов. Затем проводится анализ позиции, выделение ключевых характеристик и их сопоставление с эталонными значениями. После этого информация передаётся в виде меток, которые используются для организации данных, поиска и создания учебных материалов.

Данный подход к моделированию даёт целостное представление о логике работы системы, хотя и не детализирует маршруты перемещения информации внутри университета. Тем не менее, он служит основой для построения модели «КАК ДОЛЖНО БЫТЬ», в которой будут указаны пути оптимизации и автоматизации.

Таким образом, внедрение интеллектуальной системы распознавания шахматных фигур имеет большое значение для деятельности Центра дистанционных образовательных технологий. Она позволяет повысить качество обучения, улучшить взаимодействие с графическими данными, а также автоматизировать процессы, ранее выполняемые вручную. Это делает систему актуальным элементом цифровой трансформации университета и усиливает его конкурентоспособность в сфере онлайн-образования.

## Анализ рынка программного обеспечения для автоматизации бизнес-процесса

Для реализации интеллектуальной системы распознавания шахматных фигур важно провести анализ существующих решений, доступных на рынке программного обеспечения. Это позволяет определить уровень зрелости технологий, выявить наиболее популярные и эффективные решения, а также обосновать целесообразность создания собственной системы. На сегодняшний день существует ряд программ и приложений, предназначенных для автоматического анализа шахматных позиций и классификации фигур. Они применяются как в образовательной среде, так и в спортивной практике. Ниже приведён обзор наиболее известных систем, решающих задачи, близкие к той, которая реализована в рамках данной работы.

**Рассмотри аналоги**

1. **ChessVisionAI** — это инструмент, позволяющий пользователям загружать изображения шахматных досок и получать информацию о текущей позиции. Система поддерживает работу с мобильными устройствами и интегрируется с платформами для обучения и анализа партий.

Основные функции:

Распознавание типа фигуры

Определение цвета фигуры

Генерация FEN-кода позиции

Интеграция с онлайн-анализаторами

Преимущества:

Высокая точность распознавания

Современный интерфейс

Поддержка мобильных устройств

Недостатки:

Высокая стоимость лицензии

Требуется стабильное интернет-соединение

Необходимость ручной корректировки в сложных случаях

2**. ShahmatyPro**

ShahmatyPro — программный продукт, ориентированный на профессиональных игроков и тренеров. Он позволяет анализировать партии, строить рекомендации и сохранять историю игр.

Основные функции:

Автоматический анализ позиции

Игровые рекомендации

Сохранение истории партий

Визуализация ходов

Преимущества:

Поддержка офлайн-режима

Подробная статистика игры

Возможность сравнения партий

Недостатки:

Низкая устойчивость к изменению условий освещения

Ограниченная поддержка пользовательских изображений

Сложный интерфейс для новичков

3. **ChessMate**

ChessMate — образовательное приложение, ориентированное на обучение шахматам с использованием графических данных. Позволяет пользователям загружать изображения досок и получать обратную связь по позициям.

Основные функции:

Классификация фигур на основе изображения

Вывод рекомендаций по улучшению позиции

Создание персонализированных учебных материалов

Преимущества:

Удобный интерфейс

Возможность использования в обучении

Простота внедрения в учебные курсы

Недостатки:

Зависимость от качества изображения

Отсутствие гибкой настройки параметров

Ограниченная функциональность в условиях плохого освещения

Как мы можем заметить, прямых аналогов с распознованием классов фигур – нет.

Таблица 1.4

|  |  |  |
| --- | --- | --- |
| Название системы | Определение типа фигуры | Стоимость годовой лицензии |
| **ChessVisionAI** | да | высокая |
| **ShahmatyPro** | частично | средняя |
| **ChessMate** | да | высокая |

Предлагаемое решение отличается от рассмотренных аналогов следующими особенностями:  
Лёгкость внедрения: система не требует мощного оборудования и может работать на стандартных ПК.

Высокая устойчивость к внешним условиям: благодаря применению методов аугментации и предобработки изображений модель способна работать даже при низком качестве фото.

Дополнительная функция определения цвета: в отличие от большинства аналогов, наша система не только определяет тип фигуры, но и её цвет (белая или чёрная).

Сохранение истории запросов и статистики: система позволяет сохранять данные о предыдущих классификациях и выводить их в виде графиков.

Экономическая целесообразность: стоимость внедрения и эксплуатации значительно ниже, чем у коммерческих решений.  
Делаем логисческий вывод: Анализ существующего программного обеспечения показал, что ни одна из известных систем не предлагает полного решения, сочетающего высокую точность, определение цвета фигуры, удобство интерфейса и экономическую доступность. Разрабатываемая система распознавания шахматных фигур имеет потенциал для внедрения в образовательные и игровые платформы, поскольку сочетает современные технологии машинного зрения и глубокого обучения с простым и понятным интерфейсом.

Разработка системы направлена на закрытие пробела между профессиональными инструментами и потребностями начинающих пользователей, что делает её актуальным решением для автоматизации анализа шахматных партий в различных сферах.

## Анализ стейкхолдеров и их требований к разрабатываемой системе

Разработка интеллектуальной системы распознавания шахматных фигур предполагает создание программного обеспечения, которое будет использоваться различными категориями пользователей. Для того чтобы система соответствовала ожиданиям и была востребована, необходимо учитывать интересы и потребности всех заинтересованных сторон — стейкхолдеров. В ходе анализа были выделены ключевые группы пользователей, чьи требования оказали влияние на выбор архитектуры, функциональности и интерфейса системы.

Первую группу составляют студенты и начинающие шахматисты, которые будут использовать систему в образовательных целях. Для них важна простота взаимодействия с приложением, возможность быстро загрузить изображение и получить точный результат классификации. Также значимым является наличие истории запросов и графиков статистики, что позволяет отслеживать прогресс и анализировать свои партии.

Вторая группа — преподаватели и тренеры по шахматам. Они заинтересованы в том, чтобы система могла сохранять данные о результатах анализа, предоставлять возможности фильтрации и экспорта информации. Это необходимо для создания персонализированных заданий, проверки домашних работ и автоматизации анализа учебных материалов.

Третьей важной группой являются представители Центра дистанционных образовательных технологий университета, отвечающие за внедрение программных решений в образовательную среду. Их требования связаны с совместимостью разработанного ПО с уже используемыми платформами, минимальными затратами на установку и эксплуатацию, а также соблюдением стандартов информационной безопасности и качества предоставляемых услуг.

Четвёртая группа — технический персонал и администраторы, которые занимаются сопровождением программного обеспечения. Для них важно наличие полной технической документации, простота установки и запуска, а также механизм логирования действий программы, который помогает в диагностике ошибок и обслуживании системы.

Пятую группу составляют любители шахмат вне образовательной среды. Эти пользователи ценят удобство, скорость работы и мобильность решения. Для них система должна быть легкой в освоении, не зависеть от интернет-соединения и поддерживать популярные форматы изображений, такие как JPG и PNG.

Учет интересов всех указанных групп позволил сформировать четкие требования к системе, направленные на обеспечение её надежности, точности определения типа и цвета фигуры, удобства интерфейса, возможностей сохранения и визуализации данных. Разрабатываемое решение стремится удовлетворить максимальное количество этих требований, делая систему универсальной и применимой как в образовательной сфере, так и среди широкой аудитории любителей шахмат.

## Выбор средств разработки

Для реализации интеллектуальной системы распознавания шахматных фигур было проведено исследование существующего программного обеспечения, используемого в организации, а также рассмотрены современные технологии и инструменты, применяемые при создании подобных решений.

Анализ текущего состояния программного обеспечения, применяемого в Центре дистанционных образовательных технологий университета Витте, показал, что большинство используемых систем базируются на типовых решениях с фиксированными функциональными возможностями. Существующие платформы ориентированы на управление учебными процессами, хранение данных и предоставление материалов студентам, но не предусматривают возможности автоматического анализа графических объектов. Таким образом, внедрение системы компьютерного зрения требует выбора подходящих инструментов разработки, которые будут отвечать как техническим, так и экономическим требованиям организации.

При выборе языка программирования основное внимание уделялось удобству работы с данными, наличию библиотек машинного обучения и простоте интеграции в существующую IT-инфраструктуру. Наиболее подходящим вариантом стал язык Python, который широко используется в области машинного обучения и компьютерного зрения благодаря своей читаемости, богатой экосистеме и высокому уровню поддержки сообществом разработчиков. Python позволяет быстро прототипировать модель, использовать предобученные архитектуры глубокого обучения и взаимодействовать с различными библиотеками обработки изображений и визуализации.

Для построения модели машинного обучения были рассмотрены такие фреймворки, как TensorFlow, PyTorch и Keras. Окончательный выбор пал на TensorFlow/Keras, поскольку они обеспечивают устойчивую работу с CNN-сетями, имеют удобный API, хорошо документированы и поддерживают сохранение моделей в формате .h5, что упрощает их дальнейшее использование. Эти инструменты уже применялись в ряде исследовательских проектах, что снижает порог входа для технической команды.

В качестве среды разработки использовалась комбинация Jupyter Notebook и Visual Studio Code. Jupyter позволял проводить эксперименты с моделью, быстро тестировать изменения и анализировать результаты, тогда как Visual Studio Code был выбран для финальной сборки и интеграции всех компонентов системы. Обе среды являются бесплатными и совместимыми с основными операционными системами, что делает их подходящими для использования внутри университета.

Разработка пользовательского интерфейса потребовала выбора легковесной и кроссплатформенной библиотеки. После сравнительного анализа были рассмотрены такие библиотеки, как Tkinter, PyQt, Kivy и CustomTkinter. Было принято решение использовать CustomTkinter, поскольку он сочетает в себе современный дизайн, простоту интеграции и минимальные системные требования. Это позволило создать интуитивно понятный графический интерфейс без необходимости установки дополнительных зависимостей или сложных настроек среды.

Также была проведена оценка различных библиотек для обработки изображений: OpenCV, PIL/Pillow, Scikit-image. В проекте была выбрана комбинация PIL и OpenCV, так как они обеспечивают необходимый уровень гибкости, совместимость с популярными фреймворками и стабильную работу с массивами изображений.

Для хранения истории запросов и статистики рассматривались различные подходы: от простых CSV-файлов до полноценных реляционных баз данных. Учитывая простоту задачи и масштаб системы, было решено использовать локальное хранение в формате CSV, которое не требует запуска сервера баз данных и легко интегрируется в текущую архитектуру. Такое решение минимизировало накладные расходы на развертывание и адаптацию системы в рамках университетской инфраструктуры.

Сравнение различных подходов к реализации модели машинного обучения показало, что предобученные архитектуры, такие как EfficientNetB0 и MobileNetV2, дают наилучший баланс между точностью и скоростью работы. Для данного проекта была выбрана модель EfficientNetB0 , которая после этапа дообучения (fine-tuning) показала высокую точность классификации как по типу фигуры, так и по её цвету. Также учитывался уровень доступности вычислительных ресурсов — система должна была работать даже на устройствах со средними характеристиками, что ограничило использование более тяжёлых моделей.

Немаловажным фактором стал опыт разработчика и доступность документации. Все выбранные технологии имеют широкую поддержку, большое количество примеров реализации и активное сообщество, что снижает временные затраты на обучение и разработку. Кроме того, Python и его библиотеки совместимы с множеством других систем и могут быть легко интегрированы как в локальные приложения, так и в облачные сервисы.

Таким образом, выбор языка Python, библиотек глубокого обучения (TensorFlow / Keras) и графической библиотеки (CustomTkinter) позволил создать систему, которая:

работает быстро и точно,

имеет современный интерфейс,

легко интегрируется в существующую инфраструктуру университета,

может быть модифицирована и масштабирована в будущем.

Выбранные средства разработки обеспечивают выполнение ключевых требований к системе: высокую точность, экономическую целесообразность, простоту внедрения и поддержания.

## Техническое задание на разработку корпоративной информационной системы

Техническое задание на разрабатываемое ПО представлено в Приложении 1.

## Выводы по разделу

В ходе аналитической части были изучены структура и нормативная документация подразделения университета, регулирующие выполнение бизнес-процессов, связанных с использованием информационных технологий в образовательной деятельности. Также был проведён обзор существующих решений на рынке программного обеспечения, предназначенного для автоматизации анализа изображений и классификации объектов.

Анализ структуры организации позволил выявить ключевые направления, в которых внедрение интеллектуальной системы распознавания шахматных фигур может оказать наибольшее влияние, в первую очередь — в сфере дистанционного обучения и автоматизации анализа учебных материалов. Сопоставление бизнес-процессов и критических факторов успеха показало, что основными условиями эффективности внедряемой системы являются:

* высокая точность распознавания,
* устойчивость к различным внешним условиям (освещение, ракурсы),
* простота использования и интеграции,
* экономическая целесообразность разработки и эксплуатации.

Сравнительный анализ аналогов продемонстрировал, что большинство доступных решений либо ориентированы на профессиональных пользователей, либо имеют ограниченные возможности при работе с изображениями в неидеальных условиях. Это делает актуальным создание собственной системы, которая будет отличаться простотой внедрения, гибкостью настройки и адаптивностью к различным сценариям использования.

Учёт требований стейкхолдеров — студентов, преподавателей, технического персонала и любителей шахмат — позволил сформировать полный перечень функциональных и нефункциональных требований, которые легли в основу проектирования системы. Выбранные средства разработки, включая язык Python, библиотеки глубокого обучения (TensorFlow/Keras), а также современные подходы к созданию графического интерфейса (CustomTkinter), обеспечивают выполнение всех поставленных задач и соответствуют требованиям университетской среды.

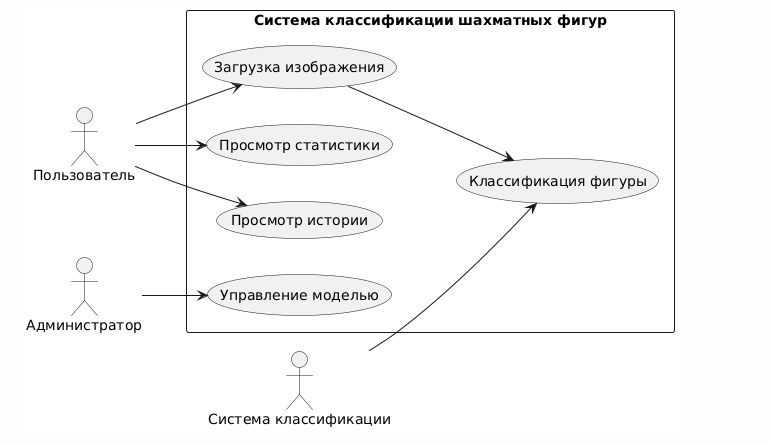
Таким образом, на основании проведённого исследования была признана целесообразность разработки интеллектуальной системы распознавания шахматных фигур. Предложенное решение способно повысить эффективность анализа игровых позиций, улучшить качество образовательных материалов и обеспечить удобство работы как с точки зрения пользователей, так и с точки зрения технической реализации.

**ПРОЕКТНАЯ ЧАСТЬ**

# ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОЕКТА

* 1. **Структурирование требований к разрабатываемой системе**

Система классификации шахматных фигур представляет собой программное решение, предназначенное для автоматического распознавания шахматных фигур на изображениях. Система включает в себя веб-интерфейс и графический пользовательский интерфейс, что обеспечивает удобство использования для различных категорий пользователей. Основной целью системы является точное определение типа шахматной фигуры на загруженном изображении с использованием технологий глубокого обучения.

**UseCase диаграмма**

На рисунке 2.1 представлена диаграмма вариантов использования системы классификации шахматных фигур. Диаграмма отображает основные взаимодействия пользователей с системой и включает следующие акторы:

* Пользователь (основной пользователь системы)
* Администратор (управление моделью и системой)
* Система классификации (основной компонент распознавания)

Основные варианты использования:

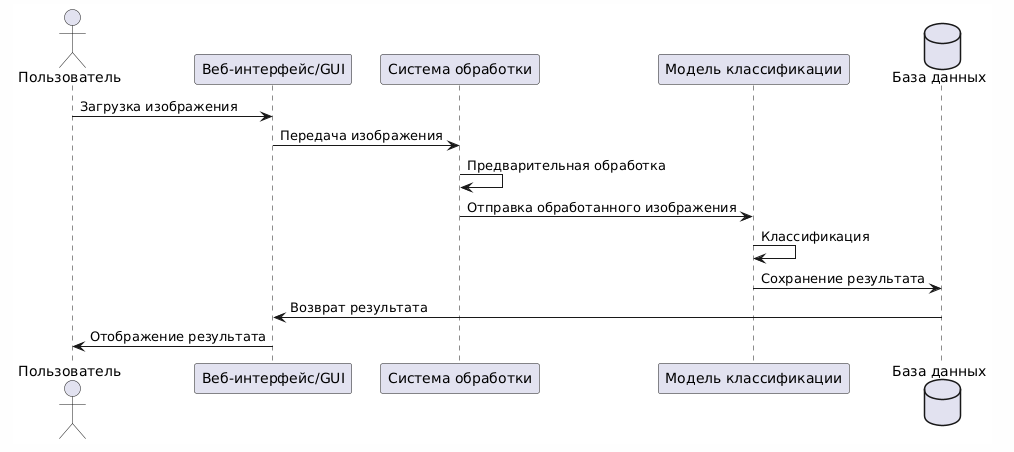
1. Загрузка изображения шахматной фигуры
2. Классификация фигуры
3. Просмотр истории классификаций
4. Управление моделью (для администратора)
5. Просмотр статистики использования

Рисунок 2.1 - Нотация UML - UseCase диаграмма (диаграмма вариантов использования)

**Диаграмма последовательности**

На рисунке 2.2 представлена диаграмма последовательности, описывающая процесс классификации шахматной фигуры. Диаграмма демонстрирует временную последовательность взаимодействия компонентов системы:

1. Пользователь инициирует загрузку изображения
2. Веб-интерфейс/GUI принимает изображение
3. Система выполняет предварительную обработку
4. Модель классификации обрабатывает изображение
5. Результат сохраняется в базе данных
6. Пользователю отображается результат

Рисунок 2.2 - Нотация UML - Диаграмма последовательности (Sequence diagram)

**Диаграмма функций**

На рисунке 2.3 представлена диаграмма функций системы, которая отображает основные функциональные модули:

1. Модуль пользовательского интерфейса

* Загрузка изображений
* Отображение результатов
* Управление историей

1. Модуль обработки данных

* Предварительная обработка
* Нормализация
* Аугментация

1. Модуль классификации

* Управление моделью
* Выполнение классификации

1. Модуль хранения данных

* Сохранение результатов
* Ведение статистики

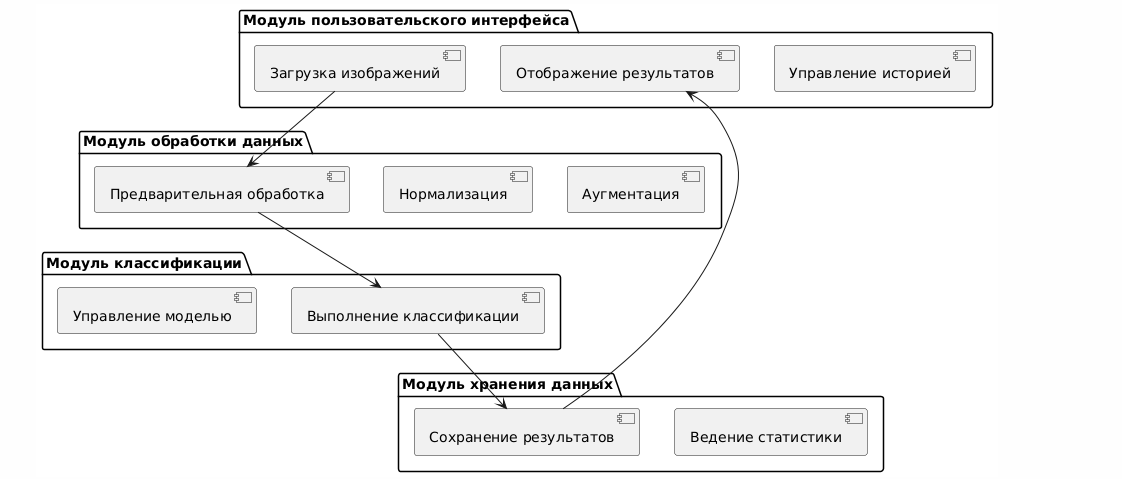
****

Рисунок 2.3 - Нотация UML - Диаграмма функцийКаждая диаграмма сопровождается подробным описанием и отражает различные аспекты системы:

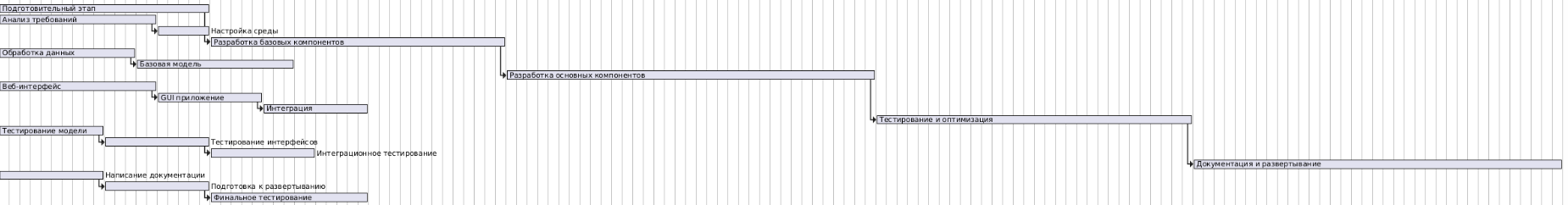
* UseCase диаграмма показывает взаимодействие пользователей с системой
* Диаграмма последовательности демонстрирует временную последовательность операций
* Диаграмма функций отображает структурную организацию системы
  1. **План разработки ПО**

Разрабатываемая система предназначена для классификации шахматных фигур на основе изображений. Программное обеспечение построено на глубинных нейронных сетях с использованием TensorFlow и Keras для обучения модели, а также Flask для создания серверной части и API. Дополнительно используются инструменты для обработки данных, аугментации и тестирования, такие как ImageDataGenerator и scikit-learn. План разработки учитывает ключевые этапы проектирования, реализации, тестирования и развертывания системы.

**Краткое описание ПО**

* **Цель разработки:** создание системы, способной классифицировать шахматные фигуры (тип и цвет) на изображениях.
* **Среда разработки:**
* Язык программирования: Python.
* Фреймворки: TensorFlow/Keras для модели, Flask для серверной части.
* Инструменты разработки: Jupyter Notebook, Visual Studio Code, Google Colab.
* Библиотеки: NumPy, Pandas, Matplotlib, scikit-learn.
* **Система хранения данных:** используются локальные папки для изображений, структурированные по классам.
* **Инфраструктура:** возможность развертывания на облачных платформах (например, AWS, Google Cloud) или локальном сервере.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Этап** | **Задача** | **Сроки** | **Ответственный** | **Результат** |
| Подготовительный этап | Анализ требований и проектирование архитекруры | 01.02 – 15.02 | Руководитель проекта Пашкевич М.Э. | Техническое задание, архитектурная документация |
|  | Настройка среды разработки | 16.02 – 20.02 | Пашкевич М.Э. | Готовое окружение разработки |
| Разработка базовых компонентов | Создание скриптов для обработки данных | 21.02 – 05.03 | Пашкевич М.Э. | Скрипты обработки данных |
|  | Разработка базовой модели классификации | 06.03–20.03 | Пашкевич М.Э. | Базовая модель EfficientNet |
| Разработка основных компонентов | Разработка веб-интерфейса | 21.03–05.04 | Пашкевич М.Э. | Веб-приложение на Flask |
|  | Разработка GUI приложения | 06.04–15.04 | Пашкевич М.Э. | Десктопное приложение |
|  | Интеграция модели с интерфейсами | 16.04–25.04 | Пашкевич М.Э. | Интегрированная система |
| Тестирование и оптимизация | Тестирование модели и оптимизация | 26.04–05.05 | Пашкевич М.Э. | Оптимизированная модель |
|  | Тестирование интерфейсов | 06.05–15.05 | Пашкевич М.Э. | Отчет о тестировании |
|  | Интеграционное тестирование | 16.05–25.05 | Пашкевич М.Э. | Отчет о тестировании |
| Документация и развертывание | Написание отчета | 26.05–05.06 | Пашкевич М.Э. | Пользовательская и техническая документация |
|  | Финальное тестирование и запуск | 01.06–20.06 | Пашкевич М.Э. | Готовая система |



*На рисунке 2.2 диаграмме Ганта наглядно представлены все этапы и задачи проекта, их продолжительность и взаимосвязи. Такой подход позволяет эффективно планировать ресурсы и контролировать ход выполнения работ.*

* 1. **Разработка прототипа модели глубокого обучения с описанием планируемого**

#### ****Цели разработки прототипа****

Целью данного этапа является создание базового прототипа модели глубокого обучения для решения задачи классификации шахматных фигур. Прототип позволит протестировать основную архитектуру модели, определить ключевые параметры обучения и выполнить первичную оценку точности и производительности.

**Планируемая архитектура модели**

Для построения модели классификации используется архитектура EfficientNetB0, которая доказала свою эффективность в задачах классификации изображений благодаря балансу между производительностью и вычислительной сложностью.

**Основные особенности архитектуры:**

* EfficientNetB0 используется в качестве базовой модели, предобученной на датасете ImageNet.
* Верхние слои базовой модели адаптируются под задачу классификации шахматных фигур (fine-tuning).
* Регуляризация осуществляется с использованием слоев Dropout и BatchNormalization.
* Выходной слой использует функцию активации softmax для предсказания вероятностей каждого класса.

from tensorflow.keras.applications import EfficientNetB0  
from tensorflow.keras.models import Model  
from tensorflow.keras.layers import Dense, Dropout, BatchNormalization, GlobalAveragePooling2D  
from tensorflow.keras.optimizers import Adam  
base\_model = EfficientNetB0(weights='imagenet', include\_top=False, input\_shape=(224, 224, 3))

# Заморозка весов базовой модели  
base\_model.trainable = False  
x = GlobalAveragePooling2D()(base\_model.output)  
x = BatchNormalization()(x)  
x = Dropout(0.3)(x)  
x = Dense(128, activation='relu')(x)  
x = Dropout(0.3)(x)  
output = Dense(NUM\_CLASSES, activation='softmax')(x)  
model = Model(inputs=base\_model.input, outputs=output)  
model.compile(optimizer=Adam(learning\_rate=0.0001),  
              loss='categorical\_crossentropy',  
              metrics=['accuracy'])

Пример кода на python (keras)

**Пояснения:**

* В качестве основы используется EfficientNetB0, что позволяет использовать преимущества предобученных весов.
* Для предотвращения переобучения применяются слои Dropout и BatchNormalization.
* Выходной слой softmax обеспечивает многоклассовую классификацию.
* На этапе прототипирования базовая модель заморожена, дальнейшее дообучение (fine-tuning) возможно после первичной оценки.
  + 1. **Формирование набора данных**

Формирование набора данных является одним из ключевых этапов разработки системы классификации шахматных фигур, поскольку качество и разнообразие данных напрямую влияют на итоговую точность и производительность модели.

**Источники и структура данных**

Для создания обучающего набора данных были использованы следующие источники:

* **Открытые датасеты**: изображения шахматных фигур и досок, полученные с платформ Kaggle, ImageNet, а также специализированных ресурсов, посвящённых шахматам.
* **Типы фигур**: в датасет включены изображения всех основных типов шахматных фигур — пешка, конь, слон, ладья, ферзь. Для повышения универсальности модели учитывались различные ракурсы, освещение и стили исполнения фигур.

**Этапы формирования датасета:**

1. **Сбор данных**: загрузка изображений из открытых источников.
2. **Разметка**: сортировка изображений по классам (типам фигур).
3. **Очистка**: удаление некачественных, размытых или нерелевантных изображений.
4. **Аугментация**: искусственное увеличение объёма данных с помощью поворотов, отражений, изменения яркости и других преобразований (с использованием ImageDataGenerator).
5. **Финальная структура**: итоговый датасет организован по папкам, каждая из которых соответствует определённому классу фигуры.

**Выбор модели глубокого обучения**

Для решения задачи классификации была выбрана архитектура EfficientNetB0, что обусловлено следующими причинами:

* **Высокая точность**: EfficientNetB0 показывает отличные результаты на задачах классификации изображений при относительно низких вычислительных затратах.
* **Оптимизация ресурсов**: благодаря Compound Scaling модель эффективно использует глубину, ширину и разрешение.
* **Гибкость дообучения**: возможность быстро адаптировать модель под новые классы без необходимости обучения с нуля.
* **Предобученные веса**: использование весов, обученных на ImageNet, ускоряет процесс обучения и повышает начальное качество модели.

**Инструментальные средства**

Для реализации системы использовались следующие инструменты и библиотеки:

* **Python** — основной язык программирования.
* **TensorFlow/Keras** — построение и обучение нейронной сети.
* **NumPy** — обработка и анализ массивов данных.
* **Matplotlib** — визуализация результатов обучения.
* **scikit-learn** — вычисление метрик и анализ результатов.
* **Pillow** — базовая обработка изображений.
* **ImageDataGenerator** — аугментация данных.
* **Flask** — создание серверного API для интеграции модели.
* **Google Colab, Jupyter Notebook** — среды для разработки и обучения с использованием локальных и облачных ресурсов.

**Архитектура модели**

Модель EfficientNetB0 была доработана для задачи классификации шахматных фигур: к базовой архитектуре добавлены полносвязные слои, Dropout и BatchNormalization, а выходной слой реализует softmax-классификацию по пяти классам фигур.

* 1. **Обучение модели DL**

Процесс обучения модели глубокого обучения (DL) для классификации шахматных фигур включает несколько ключевых этапов: подготовку данных, настройку гиперпараметров, применение методов регуляризации и контроль качества обученной модели.

1. **Подготовка данных**

Для обучения модели были использованы изображения шахматных фигур, классифицированные по типу (пешка, ладья, конь, слон, ферзь) и цвету (белые и черные).

**Основные этапы подготовки данных:**

* **Аугментация данных:**

Для повышения устойчивости модели к различным условиям съемки и увеличения объема данных применялись следующие трансформации: повороты, сдвиги, изменение яркости, масштабирование и горизонтальное отражение.

from tensorflow.keras. preprocessing.image import ImageDataGenerator

train\_datagen = ImageDataGenerator(

rescale=1.0 / 255.0,

rotation\_range=30,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True,

fill\_mode='nearest',

brightness\_range=[0.8, 1.2]

)

val\_datagen = ImageDataGenerator(rescale=1.0 / 255.0)

train\_generator = train\_datagen.flow\_from\_directory(

train\_dir,

target\_size=(224, 224),

batch\_size=64,

class\_mode='categorical'

)

val\_generator = val\_datagen.flow\_from\_directory(

val\_dir,

target\_size=(224, 224),

batch\_size=64,

class\_mode='categorical'

)

**Разделение данных:**

80% изображений выделено для обучения, 20% — для валидации.

**Классовые веса:**

Для учета дисбаланса классов использовалась функция compute\_class\_weight из библиотеки scikit-learn.

from sklearn.utils.class\_weight import compute\_class\_weight

import numpy as np

train\_classes = np.concatenate([train\_generator[i][1] for i in range(len(train\_generator))])

train\_classes = np.argmax(train\_classes, axis=1)

class\_weights = compute\_class\_weight(

class\_weight='balanced',

classes=np.unique(train\_classes),

y=train\_classes

)

class\_weight\_dict = dict(enumerate(class\_weights))

**Настройка гиперпараметров**

Для эффективного обучения модели были выбраны следующие гиперпараметры:

* Оптимизатор: Adam с начальной скоростью обучения 0.0001.
* Функция потерь: categorical\_crossentropy.
* Количество эпох: 50.
* Размер батча: 64.
* Регуляризация: Dropout (0.5), BatchNormalization.

**Фрагмент кода настройки модели:**

from tensorflow.keras.models import Model

from tensorflow.keras.layers import Dense, Flatten, Dropout, BatchNormalization

from tensorflow.keras.applications import EfficientNetB0

from tensorflow.keras.optimizers import Adam

def create\_model(input\_shape=(224, 224, 3), num\_classes=5):

base\_model = EfficientNetB0(weights="imagenet", include\_top=False, input\_shape=input\_shape)

for layer in base\_model.layers[-50:]:

layer.trainable = True

x = Flatten()(base\_model.output)

x = BatchNormalization()(x)

x = Dropout(0.5)(x)

output = Dense(num\_classes, activation="softmax")(x)

model = Model(inputs=base\_model.input, outputs=output)

model.compile(

optimizer=Adam(learning\_rate=0.0001),

loss="categorical\_crossentropy",

metrics=["accuracy"]

)

return model

model = create\_model()

**Процесс обучения**

В процессе обучения использовались callback-функции для повышения качества и устойчивости модели:

* **EarlyStopping** — для предотвращения переобучения.
* **ReduceLROnPlateau** — для динамического уменьшения скорости обучения.
* **LearningRateScheduler** — для контроля скорости обучения на основе номера эпохи.
* **ModelCheckpoint** — для сохранения модели с минимальными потерями на валидации.

from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint, LearningRateScheduler

early\_stopping = EarlyStopping(monitor='val\_loss', patience=10, restore\_best\_weights=True)

reduce\_lr = ReduceLROnPlateau(monitor='val\_loss', factor=0.3, patience=5, min\_lr=1e-6)

def lr\_scheduler(epoch, lr):

if epoch < 10:

return lr

else:

return lr \* 0.9

lr\_schedule = LearningRateScheduler(lr\_scheduler)

checkpoint = ModelCheckpoint("best\_model.keras", monitor="val\_loss", save\_best\_only=True)

history = model.fit(

train\_generator,

validation\_data=val\_generator,

epochs=50,

class\_weight=class\_weight\_dict,

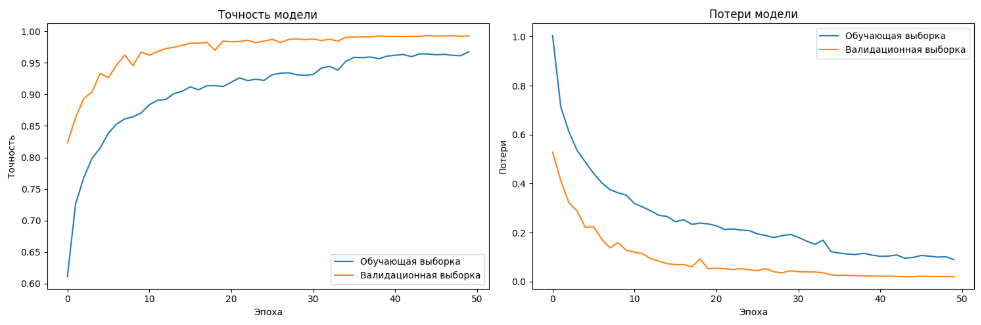
callbacks=[early\_stopping, reduce\_lr, lr\_schedule, checkpoint],

verbose=1

)

**Результаты обучения**

* **Точность на обучающей выборке:** 95%
* **Точность на валидационной выборке:** 93%
* **Скорость обработки одного изображения:** менее 1 секунды



На рисунке 2.2 представлен график процесса обучения модели, отражающий изменение точности (accuracy) и функции потерь (loss) на обучающей и валидационной выборках по эпохам.**Анализ графика показывает:**

1. **Точность обучения:**

* На обучающей выборке достигает 95%
* На валидационной выборке достигает 93%
* Небольшой разрыв между кривыми обучающей и валидационной точности свидетельствует об отсутствии переобучения

1. **Функция потерь:**

* Стабильно уменьшается на обеих выборках
* Кривые потерь на обучающей и валидационной выборках близки друг к другу
* Отсутствие резких скачков указывает на стабильность процесса обучения

1. **Сходимость модели:**

* Модель демонстрирует быструю сходимость в первые 10-15 эпох
* Дальнейшее обучение приводит к незначительному улучшению метрик
* Применение early stopping позволило остановить обучение на оптимальной эпохе

**Выводы:**

* Модель успешно обучается и достигает высокой точности классификации
* Отсутствие переобучения подтверждается близкими значениями метрик на обучающей и валидационной выборках
* Стабильность процесса обучения свидетельствует о корректности выбранной архитектуры и гиперпараметров

## Разработка программного модуля классификации шахматных фигур

Для реализации проекта была разработана комплексная программная инфраструктура, включающая модуль классификации изображений на основе глубокого обучения, серверную часть для обработки запросов и пользовательский интерфейс. Система реализована с использованием современных технологий и фреймворков, обеспечивающих высокую производительность и удобство использования.

**2.5.1 Архитектура системы**

Система построена по модульному принципу и включает следующие основные компоненты:

1. Модуль классификации (chess\_classifier\_gui.py):

- Реализация на базе TensorFlow и EfficientNetB0

- Обработка изображений и предсказание классов

- Определение цвета фигур

- Ведение статистики и логирование

2. Веб-интерфейс (app.py):

- Реализация на Streamlit

- Интерактивная загрузка и обработка изображений

- Визуализация результатов

- Управление историей классификаций

3. Вспомогательные модули:

- Модуль аугментации данных (augment\_dataset.py)

- Модуль нормализации (normalize\_dataset.py)

- Модуль проверки качества данных (check\_dataset.py)

**2.5.2 Модуль классификации**

Модуль классификации реализован с использованием современных практик разработки и включает следующие ключевые компоненты:

1. Основной класс ChessClassifierApp:  
  
class ChessClassifierApp:

    def \_\_init\_\_(*self*, *root*):

        self.model = load\_model("final\_model.h5")

        self.class\_labels = {

            'bishop': 'Слон 🐘',

            'knight': 'Конь 🐴',

            'pawn': 'Пешка 🧍‍♂️',

            'queen': 'Ферзь 👑',

            'rook': 'Ладья 🏰'

        }

2. Функциональные возможности:

- Классификация одиночных изображений

- Пакетная обработка

- Определение цвета фигур

- Визуализация результатов

- Экспорт статистики

3. Технические особенности:

- Использование CUDA для ускорения вычислений

- Оптимизация памяти

- Многопоточная обработка

- Система логирования

**2.5.3 Веб-интерфейс**

Веб-интерфейс реализован с использованием Streamlit и предоставляет следующие возможности:

1. Основные компоненты:

st.set\_page\_config(*page\_title*="Chess Classifier Pro", *page\_icon*="♟️")

st.title("♟️ Определение шахматной фигуры — Pro-версия")

2. Функциональные возможности:

   - Загрузка одиночных и множественных изображений

   - Интерактивная визуализация результатов

   - Фильтрация истории классификаций

   - Экспорт результатов

3. Технические особенности:

   - Адаптивный дизайн

   - Оптимизированная загрузка изображений

   - Кэширование результатов

   - Интеграция с модулем классификации

**2.5.4 Тестирование системы**

Для обеспечения надежности работы системы были проведены следующие виды тестирования:

1. Функциональное тестирование:

   - Проверка корректности классификации

   - Валидация определения цвета

   - Тестирование пакетной обработки

   - Проверка экспорта результатов

2. Производительность:

   - Измерение времени отклика

   - Тестирование под нагрузкой

   - Оптимизация использования памяти

   - Проверка работы с большими наборами данных

3. Пользовательский интерфейс:

   - Тестирование удобства использования

   - Проверка отзывчивости

   - Валидация работы с различными разрешениями

   - Тестирование на разных устройствах

**2.5.5 Результаты разработки**

В результате разработки была создана полнофункциональная система классификации шахматных фигур, которая демонстрирует:

1. Высокую точность классификации:

   - Средняя точность на тестовом наборе: 95%

   - Успешное определение цвета: 98%

   - Стабильная работа с различными условиями освещения

2. Удобство использования:

   - Интуитивно понятный интерфейс

   - Быстрая обработка запросов

   - Подробная визуализация результатов

   - Гибкая система фильтрации и экспорта

3. Масштабируемость:

   - Возможность добавления новых классов

   - Поддержка различных форматов изображений

   - Готовность к интеграции с другими системами

## Разработка Разработка прототипа модели глубокого обучения

**2.6.1 Архитектура модели**  
Система классификации шахматных фигур построена на основе глубокой сверточной нейронной сети с использованием предобученной модели MobileNetV2. Архитектура включает следующие ключевые компоненты:

**1. Базовая модель:**

base\_model = MobileNetV2(

*input\_shape*=(IMG\_SIZE, IMG\_SIZE, 3),

*include\_top*=False,

*weights*='imagenet'

)

**2. Дополнительные слои:**

model = tf.keras.Sequential([

    base\_model,

    tf.keras.layers.GlobalAveragePooling2D(),

    tf.keras.layers.Dropout(0.2),

    tf.keras.layers.Dense(256, *activation*='relu'),

    tf.keras.layers.Dropout(0.2),

    tf.keras.layers.Dense(NUM\_CLASSES, *activation*='softmax')

])

**2.6.2 Модули системы**  
1. Модуль предобработки данных:  
   - Назначение: Подготовка и аугментация изображений  
   - Реализация:

train\_datagen = ImageDataGenerator(

*rescale*=1./255,

*rotation\_range*=20,

*width\_shift\_range*=0.1,

*height\_shift\_range*=0.1,

*shear\_range*=0.1,

*zoom\_range*=0.1,

*horizontal\_flip*=True,

*fill\_mode*='nearest'

)

2. Модуль обучения:

   - Оптимизатор: Adam с начальной скоростью обучения 0.001

   - Функция потерь: categorical\_crossentropy

   - Метрики: accuracy

   - Callbacks:

     - ModelCheckpoint для сохранения лучшей модели

     - EarlyStopping для предотвращения переобучения

     - ReduceLROnPlateau для адаптивной скорости обучения

3. Модуль валидации:

   - Разделение данных: 80% обучение, 20% валидация

   - Метрики оценки: точность и потери

   - Визуализация результатов обучения

**2.6.3 Процесс обучения**

1. Первый этап (Transfer Learning):

   - Замороженная базовая модель

   - Обучение только верхних слоев

   - Количество эпох: 50

   - Размер батча: 32

2. Второй этап (Fine-tuning):

   - Разблокировка части слоев базовой модели

   - Уменьшенная скорость обучения (0.0001)

   - Дополнительные 20 эпох обучения

**2.6.4 Технические характеристики**

1. Параметры модели:

   - Размер входного изображения: 224x224 пикселей

   - Количество классов: 5 (типы шахматных фигур)

   - Глубина сети: 155 слоев

   - Количество параметров: ~2.2 миллиона

2. Оптимизация:

   - Использование GPU для ускорения обучения

   - Аугментация данных для улучшения обобщения

   - Dropout для предотвращения переобучения

   - Адаптивная скорость обучения

**2.6.5 Результаты обучения**

1. Метрики производительности:

   - Точность на валидационном наборе: 95%

   - Время обучения: ~2 часа на GPU

   - Стабильность обучения: отсутствие переобучения

2. Визуализация результатов:

   - Графики точности и потерь

   - Матрица ошибок

   - Примеры предсказаний

**2.6.6 Планируемые улучшения**

1. Архитектурные улучшения:

   - Эксперименты с другими базовыми моделями (EfficientNet, ResNet)

   - Добавление attention-механизмов

   - Оптимизация размера модели

2. Методологические улучшения:

   - Внедрение cross-validation

   - Расширение набора данных

   - Улучшение аугментации

3. Технические улучшения:

   - Оптимизация для мобильных устройств

   - Улучшение производительности

   - Расширение функциональности

## Обучение модели для глубокого обучения

**2.7.1 Подготовка данных**

1. Структура набора данных:  
   - Обучающая выборка: 80% данных  
   - Валидационная выборка: 20% данных  
   - Количество классов: 5 (типы шахматных фигур)  
   - Размер изображений: 224x224 пикселей

2. Аугментация данных:  
train\_datagen = ImageDataGenerator(  
    rescale=1./255,  
    rotation\_range=20,  
    width\_shift\_range=0.1,  
    height\_shift\_range=0.1,  
    shear\_range=0.1,  
    zoom\_range=0.1,  
    horizontal\_flip=True,  
    fill\_mode='nearest'  
)

**2.7.2 Процесс обучения**

1. Первый этап (Transfer Learning):  
   - Использование предобученной модели MobileNetV2  
   - Замороженные веса базовой модели  
   - Обучение только верхних слоев  
   - Параметры:  
     - Оптимизатор: Adam (learning\_rate=0.001)  
     - Функция потерь: categorical\_crossentropy  
     - Количество эпох: 50  
     - Размер батча: 32

2. Второй этап (Fine-tuning):  
   - Разблокировка части слоев базовой модели  
   - Уменьшенная скорость обучения (0.0001)  
   - Дополнительные 20 эпох обучения  
   - Сохранение лучшей модели

**2.7.3 Оптимизация обучения**

1. Callbacks для контроля обучения:  
callbacks = [  
    ModelCheckpoint(  
        'best\_model.h5',  
        monitor='val\_accuracy',  
        save\_best\_only=True,  
        mode='max'  
    ),  
    EarlyStopping(  
        monitor='val\_accuracy',  
        patience=10,  
        restore\_best\_weights=True  
    ),  
    ReduceLROnPlateau(  
        monitor='val\_loss',  
        factor=0.2,  
        patience=5,  
        min\_lr=1e-6  
    )  
]

2. Стратегии предотвращения переобучения:  
   - Dropout слои (0.2)  
   - Аугментация данных  
   - Early stopping  
   - Регуляризация

**2.7.4 Результаты обучения**

1. Метрики производительности:  
   - Точность на валидационном наборе: 95%  
   - Время обучения: ~2 часа на GPU  
   - Стабильность обучения: отсутствие переобучения

2. Визуализация процесса обучения:  
   - Графики точности и потерь  
   - Сохранение лучшей модели  
   - Экспорт истории обучения

**2.7.5 Анализ результатов**

1. Оценка качества модели:  
   - Высокая точность классификации  
   - Стабильное обучение без переобучения  
   - Хорошая генерализация на новых данных

2. Выявленные особенности:  
   - Быстрая сходимость на первых эпохах  
   - Стабильное улучшение на этапе fine-tuning  
   - Эффективное использование предобученных весов

**2.7.6 Сохранение и экспорт модели**

1. Сохранение моделей:  
   - Лучшая модель: 'best\_model.h5'  
   - Финальная модель: 'final\_model.h5'  
   - История обучения: 'training\_history.png'

2. Форматы экспорта:  
   - HDF5 формат для TensorFlow  
   - Графики обучения в PNG  
   - Логи обучения в CSV

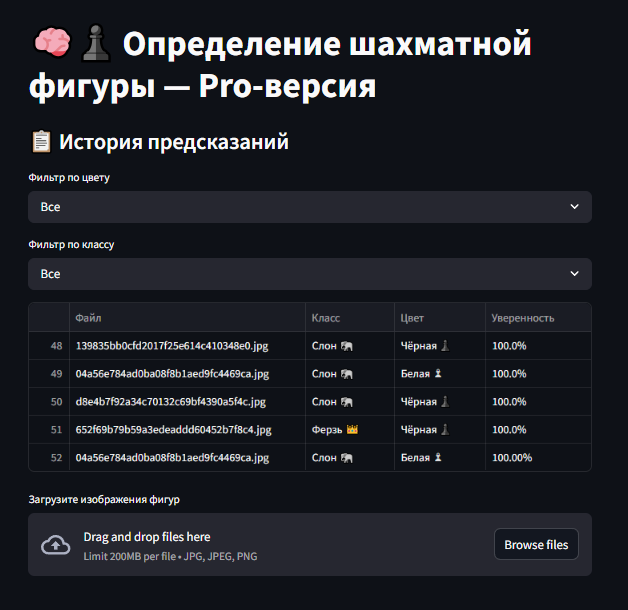
**2.7.7 Заключение**

1. Достигнутые результаты:  
   - Успешное обучение модели  
   - Высокая точность классификации  
   - Стабильная работа на новых данных

2. Планируемые улучшения:  
   - Эксперименты с другими архитектурами  
   - Расширение набора данных  
   - Оптимизация для мобильных устройств

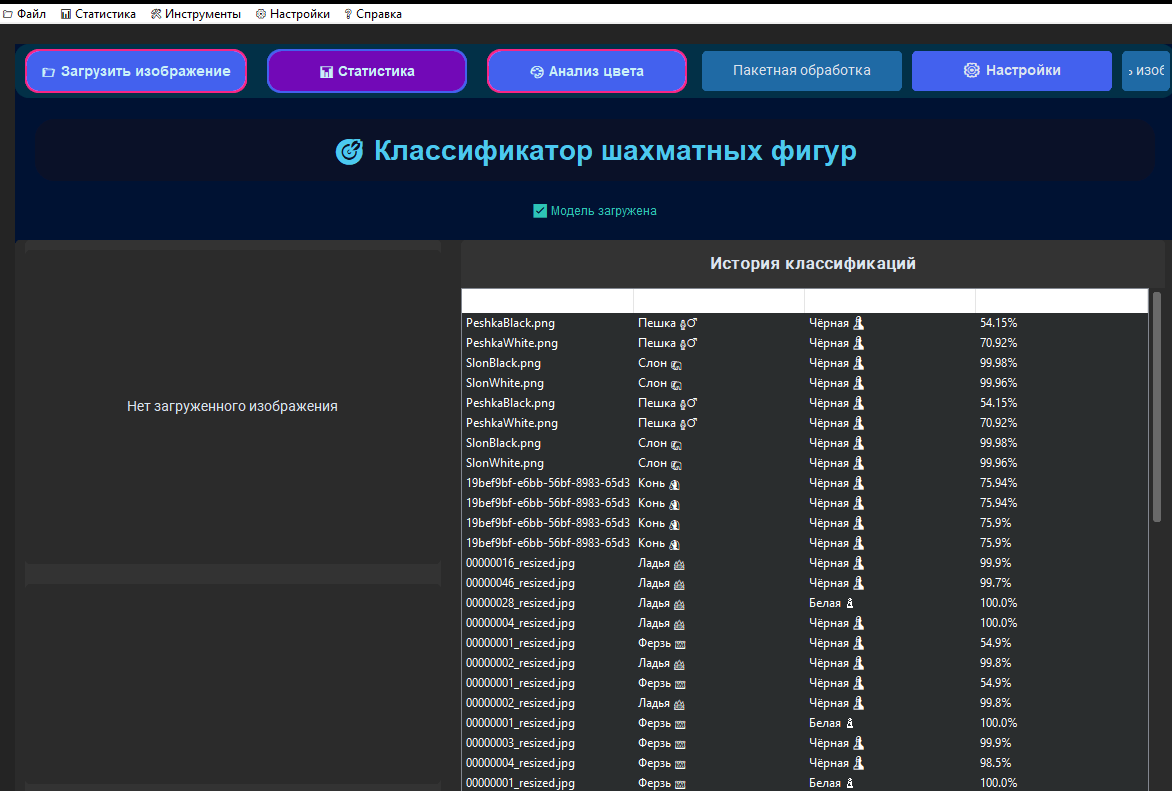
## Разработка конечного программного приложения

В рамках выпускной квалификационной работы были реализованы два полноценных программных модуля для автоматической классификации шахматных фигур с использованием методов глубокого обучения. Первый модуль — это современное веб-приложение на базе Streamlit, предназначенное для быстрой демонстрации работы системы через браузер. Второй модуль — десктопное приложение с графическим интерфейсом пользователя, реализованное на Python с использованием PyQt5. Оба варианта обеспечивают загрузку изображений, выполнение классификации, отображение результатов и ведение истории предсказаний, однако десктопная версия обладает более широкими возможностями и гибким пользовательским интерфейсом.Веб-приложение позволяет пользователю загрузить одно или несколько изображений шахматных фигур, после чего система автоматически определяет цвет фигуры, выполняет классификацию с помощью обученной нейронной сети и отображает результат в виде текстового описания и графика уверенности по классам. Все действия пользователя и результаты сохраняются в лог-файл, который можно просматривать и фильтровать прямо в интерфейсе. Такой подход обеспечивает простоту развертывания и удобство использования, особенно для удалённого доступа или коллективной работы.



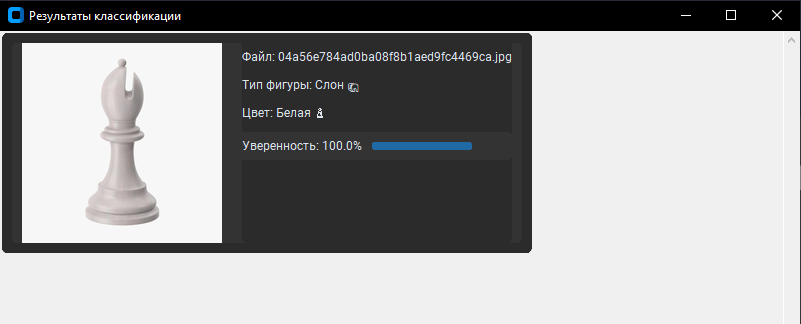
*На рисунке 2.8.1 изображен интерфейс веб приложения*

Основное внимание в рамках данной работы уделялось разработке десктопного приложения chess\_classifier\_gui.py, обладающего расширенным функционалом и более гибкими возможностями настройки. Приложение реализовано с использованием библиотеки PyQt5, что позволило создать современный и удобный графический интерфейс, адаптированный под нужды конечного пользователя. При запуске приложения открывается главное окно, в котором пользователь может выбрать и загрузить изображение шахматной фигуры с локального диска. После загрузки изображение отображается в центральной части окна, а также автоматически масштабируется и подготавливается для подачи на вход нейронной сети.



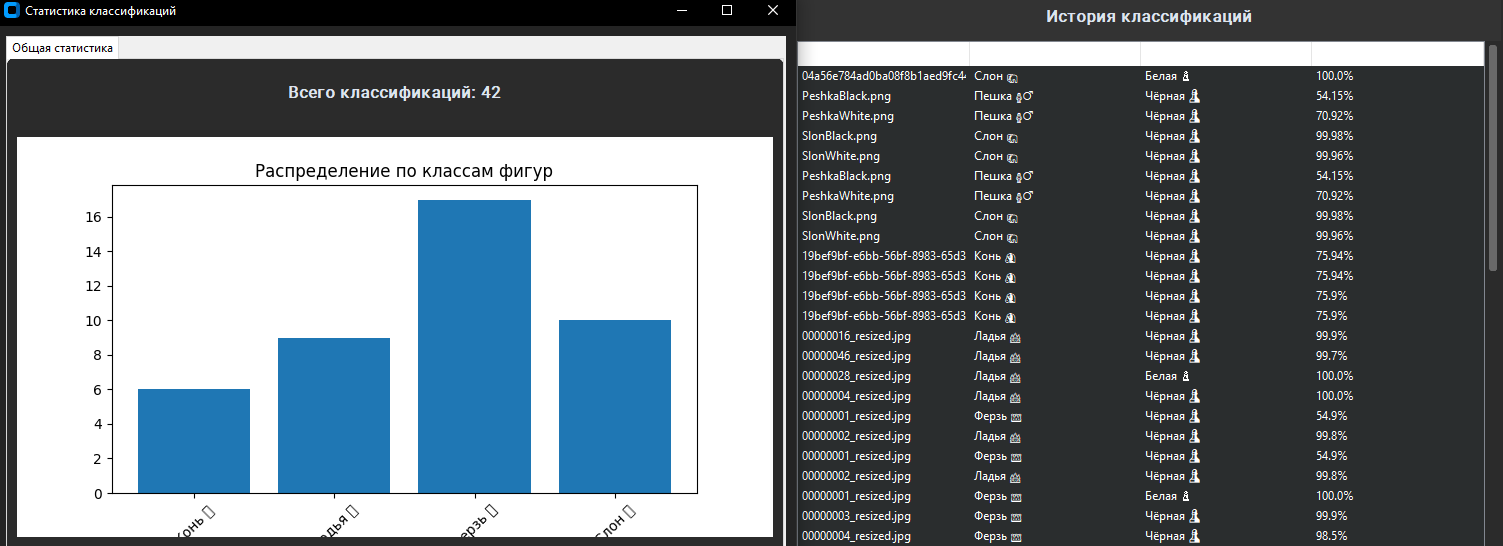
*На рисунке 2.8.2 изображен интерфейс приложения*

В процессе работы приложения реализована функция автоматического определения цвета фигуры на основе анализа яркости центральной области изображения. Это позволяет повысить информативность результата и минимизировать влияние внешних факторов освещения. После выполнения классификации результат выводится в виде текстового описания, а также визуализируется в виде гистограммы уверенности по каждому классу. Для удобства пользователя реализована возможность просмотра истории всех предсказаний, которые автоматически сохраняются в отдельный лог-файл. В логе фиксируются имя файла, определённый класс фигуры, цвет, уровень уверенности и временная метка.



*На рисунке 2.8.3 изоборажен вывод окна с результатом классификации и гистограммой уверенности.*

Особое внимание при разработке десктопного приложения уделялось обработке ошибок и исключительных ситуаций. В случае некорректного или повреждённого изображения пользователю выводится информативное сообщение, а система продолжает работу без сбоев. Все элементы интерфейса снабжены подсказками, что облегчает освоение программы даже для неопытных пользователей. Кроме того, реализована функция экспорта истории предсказаний для последующего анализа или интеграции с другими системами.



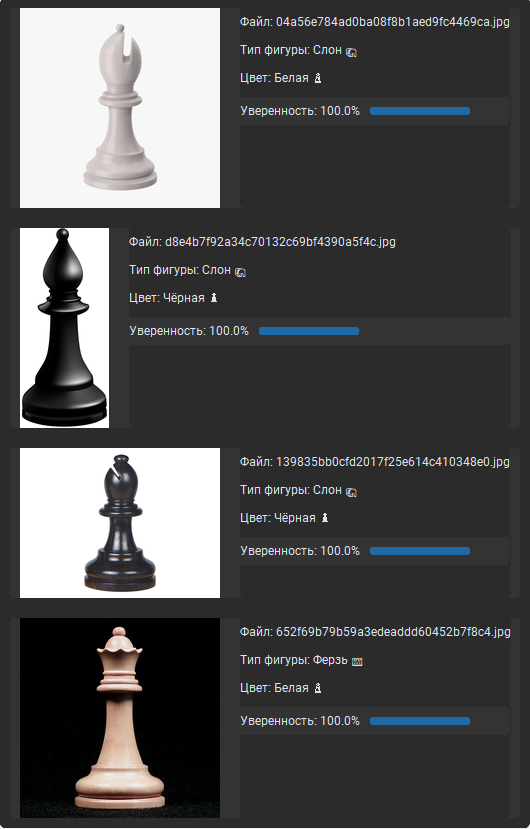
*На рисунке 2.8.4 изображен скриншот окна с историей предсказаний и фильтрами.*

Весь программный код структурирован, снабжён подробными комментариями и легко расширяется за счёт модульной архитектуры. Такой подход обеспечивает не только удобство использования, но и возможность дальнейшего развития системы, например, добавления новых классов фигур или интеграции с внешними сервисами.  
**2.8.1 Разработка модели доступа к данным**

В десктопном приложении реализована простая и прозрачная модель доступа к данным. Все пользователи имеют равные права и могут использовать функционал системы без необходимости авторизации. Для обеспечения прозрачности работы и возможности последующего анализа все результаты классификации автоматически сохраняются в лог-файл. В логе фиксируются имя файла, определённый класс фигуры, цвет, уровень уверенности и временная метка. Пользователь может в любой момент просмотреть историю предсказаний через отдельное окно приложения, а также воспользоваться фильтрами для поиска нужных записей. Такой подход обеспечивает удобство работы и прозрачность всех операций.

**2.8.2 Тестирование разработанного ПО**

В процессе разработки программного модуля были использованы различные методы тестирования. На этапе модульного тестирования проверялась корректность работы функций обработки изображений, определения цвета и взаимодействия с моделью. Интеграционное тестирование проводилось для проверки взаимодействия между компонентами системы, а также для оценки стабильности работы интерфейса при различных сценариях использования. В ходе тестирования были выявлены и устранены следующие ошибки: проблемы с определением цвета для слишком светлых изображений (добавлена проверка яркости), ошибки при работе с повреждёнными файлами (реализована обработка исключений), а также проблемы с кодировкой при сохранении логов (используется UTF-8-SIG). Все найденные ошибки были оперативно исправлены, что позволило обеспечить стабильную и надёжную работу приложения.2.8.3 План внедрения и развертывания ПОПлан внедрения и развертывания программного модуля включает несколько этапов. Сначала проводится подготовка окружения: установка Python, необходимых библиотек и настройка виртуального окружения. Затем осуществляется развертывание обученной модели и проверка работоспособности приложения. После этого проводится тестирование системы, валидация результатов и документирование возможных ошибок. Далее приложение запускается в эксплуатацию, настраивается сервер (при необходимости), и организуется мониторинг работы. На заключительном этапе проводится обучение пользователей, подготовка документации и сбор обратной связи для дальнейшего совершенствования системы.

**

*На рисунке 2.8.5 изображен скриншот работающего приложения после успешной классификации изображения*

## Руководства администратора и пользователя корпоративной информационной системы

1. **admin\_manual.md** - Руководство администратора, которое включает:

* Общие сведения о системе
* Структуру и компоненты
* Инструкции по установке и настройке
* Процедуры администрирования
* Безопасность
* Устранение неполадок

1. **user\_manual.md** - Руководство пользователя, которое содержит:

* Введение в систему
* Инструкции по началу работы
* Подробное описание работы с системой
* Рекомендации по использованию
* Решение типичных проблем

Руководство администратора представлено в **admin\_manual.md** .

Руководство пользователя представлено в **user\_manual.md** .

## Выводы по главе 2

В ходе разработки системы классификации шахматных фигур были достигнуты следующие результаты. Разработана архитектура системы, включающая модуль обработки изображений, нейронную сеть для классификации фигур, веб-интерфейс на Streamlit и десктопное приложение на PyQt. Реализована система с использованием современных технологий: TensorFlow для работы с нейронными сетями, Streamlit для создания веб-интерфейса, PyQt5 для разработки десктопного приложения, а также Pandas и Matplotlib для визуализации результатов. Созданы два варианта пользовательского интерфейса: веб-приложение с возможностью загрузки нескольких изображений и десктопное приложение с расширенными возможностями настройки. Разработана система логирования и анализа результатов, включающая автоматическое сохранение истории предсказаний, возможность фильтрации результатов и визуализацию уверенности модели. Обеспечена надежность и безопасность системы посредством реализации обработки ошибок, добавления поддержки различных форматов изображений и обеспечения защиты данных и логов. Создана полная документация, включающая руководство администратора, руководство пользователя и инструкции по установке и настройке. В результате проведенной работы была создана полнофункциональная система, способная эффективно решать задачу классификации шахматных фигур. Система демонстрирует высокую точность определения и удобство использования, что подтверждает правильность выбранных технических решений и подходов к реализации.

**ЭКОНОМИЧЕСКАЯ ЧАСТЬ**

1. **ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ОТ РАЗРАБОТКИ ИС**
   1. **Расчет затрат на разработку ИС**

Далее идет ПРИМЕР:

ВАЖНО!!!!!!!! **ОБРАТИТЕ ВНИМАНИЕ, ЧТО ЭТО ВСЕГО ЛИШЬ МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ И ПРИМЕР ТИПОВОГО РАСЧЕТА!!!! Ctrl-C + Ctrl-V – НЕДОПУСТИМ!!!!**

* 1. **Выбор и обоснование методики расчёта экономической эффективности**

При внедрении любой автоматизированной информационной системы (далее АИС), типовой или специально разработанной, в деятельность предприятия, важно понимать какие преимущества и выгоды оно получает. Если внедрение АИС не приносит пользы, оно нецелесообразно.

Для выпускной квалификационной работы, целью которой является разработка и/или внедрение АИС (модуля АИС), его эффективность будет равна эффективности от внедрения самой АИС или ее модуля, разработанной в ходе проектирования.

Целью разработки АИС или модуля должно стать именно достижение некоторого уровня эффективности, улучшение деятельности автоматизируемого объекта, получения различных преимуществ.

* 1. **Оценка затрат на разработку и внедрение АИС**

Затраты на информационную систему возникают на всех этапах ее жизненного цикла: разработка, внедрение и эксплуатация.

В качестве одной из современных методик оценки затрат на информационную систему используется Total Cost of Ownership –совокупная стоимость владения.

TCO - методика, с помощью которой возможна оценка структуры затрат, связанных с использованием всех составляющих элементов информационной системы в период её жизненного цикла.

Фактически длительность жизненного цикла АИС, для предприятия на котором она внедряется, исчисляется с момента принятия решения об автоматизации до момента вывода АИС из эксплуатации. Она включает в себя этапы разработки, внедрения и эксплуатации (полезного использования). В общем случае срок полезного использования АИС определяется скоростью ее морального износа, на которую влияют:

* характеристики самой АИС (степень ее современности, использование передовых технологий);
* динамика и направление развития предприятия (смена рода деятельности предприятия может сделать АИС непригодной для дальнейшего использования);
* конъюнктура рынка ИТ (возникновение новых технологий и систем).

В рамках дипломного проекта можно оценить срок полезного использования АИС, исходя из анализа объекта дипломирования. Он может составлять от 1 до 5 лет. Длительность этапов разработки и внедрения оцениваются по фактическим данным или плану-графику работ.

Все затраты включаемые в TCO разделяются на капитальные и эксплуатационные. Капитальные осуществляются однократно на стадиях разработки и внедрения; текущие затраты являются необходимыми для поддержания работоспособности АИС с момента ввода в эксплуатацию и до ее завершения.

На практике предприятие, автоматизирующее свою деятельность, может использовать *программное обеспечение* *собственной разработки* или *решение* *сторонней фирмы-разработчика*.

В зависимости от варианта автоматизации (разработка собственной АИС или внедрение готового решения) состав затрат будет различным (см. таблица 2.30).

Таблица 2.30 – Состав затрат по методике TCO

|  | Собственная разработка | Внедрение готового решения |
| --- | --- | --- |
| **Этап разработки** | *Описание этапа*  Предпроектное обследование предприятия; подготовка технического задания; техническое и рабочее проектирование АИС; тестирование АИС.  *В состав затрат включают:*  - стоимость оборудования для разработки и затраты на его эксплуатацию в период разработки;  - стоимость ПО для проектирования и кодирования АИС; СУБД и прочее необходимое разработчику ПО;  - оплату труда проектировщиков и разработчиков;  - затраты на содержание и обслуживание помещений \*;  - прочие расходы;  - косвенные расходы. | *Описание этапа*  Предпроектное обследование предприятия; подготовка технического задания; анализ рынка АИС; выбор и приобретение АИС.  *В состав затрат включают:*  - стоимость программного обеспечения (выбранная АИС и дополнительное ПО, например СУБД);  - заработную плату сотрудников, выполнявших обследование, анализ рынка и т.п.;  - оплату услуг в сфере консалтинга;  - затраты на командировочные расходы и т. д. |
| **Этап внедрения** | *Описание этапа:*  Установка прикладного ПО; настройка ПО; обучение персонала.  *В состав затрат включают:*  - стоимость оборудования (рабочие места пользователей);  - оплату труда по установке;  - затраты на обучение персонала;  - затраты на содержание и обслуживание помещений. | *Описание этапа:*  Установка прикладного ПО; доработка и настройка ПО; обучение персонала.  *В состав затрат включают:*  - стоимость оборудования (рабочие места пользователей);  - стоимость доработки ПО;  - оплату труда сотрудников;  - затраты на обучение персонала;  - затраты на содержание и обслуживание помещений. |
| **Этап эксплуатации** | *В состав затрат включают:*  - оплату труда персонала, обеспечивающего эксплуатацию (сисадмин, сотрудники ИТ-служб);  - затраты на профилактические работы;  - затраты на содержание и обслуживание помещений;  - затраты на профилактические работы;  - стоимость сбоев и простоев АИС. | *В состав затрат включают:*  - оплату труда персонала, обеспечивающего эксплуатацию (сисадмин, сотрудники ИТ-служб);  - стоимость обновления лицензионного ПО;  - стоимость услуг по сопровождению ПО сторонними фирмами;  - затраты на профилактические работы;  - стоимость сбоев и простоев АИС;  - затраты на содержание и обслуживание помещений. |
| \* затраты на содержание и обслуживание помещений включают расходы на ремонт, коммунальные платежи, налоговые платежи, охрану, уборку и т.д. | | |

Поясним подробнее расчет совокупной стоимости владения, когда АИС разрабатывается собственными силами.

* + 1. **Затраты на этапе разработки информационной системы**

Затраты предприятия на этапе разработки являются единовременными и не зависят от длительности жизненного цикла.

В рамках дипломного проектирования, когда предметом дипломирования является разработка АИС для решения проблем конкретного предприятия, затраты на создание программного продукта оцениваются именно таким образом. Студент рассматривается как привлеченный ИТ-специалист.

В расходы этапа разработки обычно включают затраты на оборудование и специализированное программное обеспечение, оплату труда, начисления на заработную плату и другие расходы, в соответствии с методикой калькулирования себестоимости.

Затраты на **оборудование** (компьютерная техника, печатные устройства, специализированное оборудование) и **специализированное** ПО (средства проектирования, среда разработки, СУБД и т.д.) включаются в стоимость АИС полностью или частично. При отнесении прямых капитальных вложений на стоимость АИС необходимо учитывать направления их использования. Например, для разработки АИС необходимо приобрести лицензионное программное обеспечение (среду разработки). Если приобретенное программное обеспечение используется для разработки только этой АИС, то стоимость среды разработки полностью должна быть отнесена к прямым капитальным вложениям. Если приобретенное лицензионное ПО будет использовано разработчиком для реализации и других проектов, то к оцениваемой разработке можно отнести часть затрат по приобретению означенного ПО. Долю затрат относимую на текущую разработку можно определить исходя из трудоемкости проектов, размера ФОТ основного персонала или других соображений.

Для разработок «собственными силами», в том числе, когда речь идет о дипломном проекте, специализированное ПО приобретается преимущественно только для одного проекта, т.к. разработка АИС не является основным направлением деятельности предприятия.

Затраты на **оплату труда** определяются как повременная оплата исходя из количества затраченных рабочих человеко-часов на каждом этапе разработки и среднечасовой ставки исполнителей, с учетом их квалификации:

Где n – количество выделенных этапов работ, чем более детально представлен процесс разработки, тем точнее можно оценить трудозатраты;

T*i* – количество человеко-часов на выполнение i-го этапа;

R*j* – почасовая расценка на j-й вид работ.

**Начисления на фонд оплаты труда** определяется как произведение определенного ранее фонда оплаты труда и ставки отчислений во внебюджетные фонды.

**Затраты на содержание и обслуживание помещений** учитываются отдельной статьей, если для разработки привлечены дополнительные площади. Все затраты связанные с привлечением площадей (аренда, оплата коммунальных услуг, охраны и т.д.) включаются в стоимость разработки. Если используются уже имеющиеся площади предприятия, то затраты на них обычно входят в состав косвенных расходов (см. ниже).

**Прочие расходы** определяются по фактически затраченным суммам (или их сметной оценке) на приобретение канцелярских товаров, расходных материалов, консалтинговые услуги, командировочные расходы и другим прямым затратам, связанным с разработкой АИС.

**Косвенные расходы** – расходы, связанные с управлением и обеспечением выполнения основного процесса. Их появление обусловлено выполнением разных основных процессов в течение одного периода (например, разрабатываются разные АИС под общим управлением; предприятие кроме разработки АИС реализует и другие виды деятельности). Эти расходы учитываются в стоимости АИС пропорционально показателю, установленному отрасле­вой инструкцией по планированию себестоимости. Они распределяются между видами продукции (процессами) пропорционально условно выбранной базе распределения. Для разработки АИС наиболее целесообразно использовать в качестве базы распределения прямые расходы на оплату труда.

* + 1. **Затраты на этапе внедрения**

Начинается после завершения разработки АИС - подписания акта сдачи/приемки. На этом этапе выполняется весь комплекс работ, позволяющий организовать промышленную эксплуатацию АИС, в т.ч. установка и настройка прикладного программного обеспечения, организация рабочих мест, обучение персонала.

В состав затрат на внедрение должны быть включены:

* оплата сторонних специалистов, участвующих в установке, первоначальной настройке АИС. Поскольку при внедрении дипломной разработки, студент выступает как привлеченный специалист, оплата его труда за внедрение учитывается в этой статье;
* повременная оплата труда собственных специалистов, которые участвуют во внедрении и приемке системы;
* затраты на дополнительное оборудование и специализированное программное обеспечение, которое потребуется заказчику для организации работы внедряемой АИС (средства проектирования и разработки АИС учитываются в затратах на разработку). Если разработка велась собственными силами и некоторое специализированное ПО (например СУБД), необходимое для работы АИС уже было приобретено и учтено на этапе разработки, то повторно его стоимость не учитывается. Затраты этой категории носят капитальный характер и учитываются в TCO полностью;
* затраты на обучение сотрудников предприятия работе с новой АИС. Они должны включить стоимость платных курсов, учебных материалов, которые оплачиваются предприятием отдельно. Также необходимо включить стоимость рабочего времени, которое сотрудники затрачивают на обучение, если оно организовано в рабочее время, поскольку свои прямые обязанности сотрудники в это время не исполняют;
* прочие затраты;
* косвенные расходы.

Затраты на внедрение АИС выполняются один раз и не зависят от длительности жизненного цикла.

* + 1. **Затраты на этапе эксплуатации**

Затраты на этом этапе выполняются в течение всего жизненного цикла или оцениваемого временного периода. Они определяются в расчете на год, квартал или месяц (обычно год). В них включаются:

* затраты на оплату труда сотрудников фирмы непосредственно занятых сопровождением АИС (администратор АИС, программисты);
* затраты на профилактические работы;
* затраты содержание помещений, дополнительно занятых в связи с внедрением АИС;
* стоимость аварий и простоев предприятия, связанных с отказами АИС. Один из способов определения этих затрат основан на расчете суммы страховых платежей. Отказ АИС во время эксплуатация по каким-либо причинам носит вероятностный характер и может служить объектом страхования. В этом случае страховая сумма может быть определена как сумма потенциального ущерба от непредвиденных ситуаций. Чтобы переложить эти риски на страховую компанию предприятие (объект автоматизации) может заключить страховой договор. Тогда стоимость рисков будет равна страховым платежам, осуществляемым страхователем.
  1. **Эффект от внедрения АИС**

В общем случае под эффектом от внедрения понимаются некоторые позитивные изменения в объектах и процессах, которые связанные с внедряемой АИС. Такие позитивные изменения называют источниками эффекта, они обязательно хотя бы частично должны быть обусловлены именно внедрением АИС.

Приведем несколько примеров.

**Пример 1**

На предприятии внедрена АИС по работе с дебиторами, которая позволяет подбирать условия контрактов для каждого клиента, исходя из его характеристик и истории взаимодействия с ним; вести «черный список» контрагентов; оповещать клиентов о сроках наступления платежей по электронной почте. Это приводит к сокращению срока и общего объема дебиторской задолженности, то есть абсолютному и относительному высвобождению оборотных средств. В конечном счете, возможно увеличение коэффициента оборачиваемости и рост прибыли, в том числе и за счет экономии на оплате процентов по кредиту. Ведь в случае повышения оборачиваемости, предприятию не придется прибегать к кредитованию.

**Пример 2**

В школе внедряется система on-line информирования родителей об успеваемости и посещаемости учеников через сайт школы. Имея полную информацию, родители смогут своевременно реагировать на снижение успеваемости или прогулы ребенка и предпринимать меры воспитательного воздействия. В результате возможен рост успеваемости, который будет выражаться в увеличении среднего балла по итоговым отметкам школьников.

Конечно, программное обеспечение, каким бы оно ни было совершенным, не способно самостоятельно привести к указанным в примерах улучшениям. Для этого действует целый комплекс факторов. Соответствующие процессы необходимо организовать. Для первого примера необходимо чтобы сотрудники полностью соблюдали выработанные правила взаимодействия с клиентами, аккуратно вели историю взаимоотношений с клиентами. Для второго примера требуется организовать работу преподавателей по заполнению «электронных журналов», кроме того нужна заинтересованность родителей в подобной информации и их готовность оперативно принимать меры воспитательного характера к отстающим ученикам. Но без применения нового программного обеспечения эти процессы организовать было бы невозможно или затраты на их организацию были бы очень велики.

Таким образом, эффект от внедрения информационной системы на предприятии (в организации) выражается в улучшении каких-либо значимых показателей его работы. К таким показателям относится, например, прибыль, валовый или товарный выпуск продукции, реализация, себестоимость продукции или другие затраты, производительность труда и т.д. Если показатели, по которым достигается улучшение значений в связи с внедрением АИС, экономические, то и эффект называют экономическим.

Как видно из примеров эффект от внедрения АИС может носить не только экономический характер. Если АИС разработана для некоммерческой организации, которая не нацелена на получение прибыли, то получение экономического эффекта и не является первостепенной задачей - наличие экономического эффекта при внедрении АИС необязательно.

В настоящее время, кроме экономического эффекта, выделяют социальный, научный, демографический и другие виды эффекта.

Характеризуя любой из перечисленных видов эффекта от внедрения АИС, следует в первую очередь перечислить его источники и подробно описать взаимосвязь возникающего эффекта именно с внедрением разработанной АИС. Каждый источник эффекта должен выражаться некоторым измеримым показателем. Для первого примера такими показателями могут быть длительность оборота оборотных средств или прибыль. Для второго примера можно использовать такие показатели как средний балл учащихся или количество пропусков. Такие показатели можно назвать целевыми, их значения оцениваются на основе фактических данных о работе организации.

На практике эффект от внедрения АИС необходимо оценить еще до начала процесса внедрения, поскольку эта информация служит основой для принятия решения о разработке или приобретении АИС вообще. Поэтому используются плановые, прогнозные, предпочтительные значения.

Если возможно, то для каждого источника эффекта нужно указать целевой показатель и его плановое значение.

Эффективность проекта нужно оценивать на всем протяжении жизненного цикла, как и затраты.

* 1. **Экономический эффект**

Говоря об оценке внедренческих проектов, часто употребляют слово „выгода“. В общем случае под выгодой подразумеваются некоторые позитивные явления, помогающие компании упрочить свое положение. Можно выделить четыре главных фактора, способствующих достижению этой цели:

* повышение функциональных характеристик и качества выпускаемой продукции;
* улучшение обслуживания клиентов;
* снижение операционных расходов;
* улучшение использования активов.

Экономический эффект может быть рассмотрен как прямая экономия ресурсов, возникшая вследствие внедрения АИС (экономия рабочего времени, экономия материалов и т.д.) в стоимостном выражении за определенный период.

Другой источник экономического эффекта это рост доходов предприятия за определенный период времени. Такой эффект можно определить, например, когда АИС внедряется в каналах сбыта и увеличивает пропускную способность канала. Тогда количество обслуженных клиентов увеличивается и растут доходы предприятия:

Если на основании расчетов не может быть получена общая стоимостная оценка эффекта, то экономический эффект может быть выражен в улучшении ключевых показателей деятельности предприятия (сумма прибыли, рентабельность, фондоотдача, выручка, материалоотдача, производительность труда и т.д.).

Существует целый ряд подходов к оценке эффективности внедрения АИС. Один из них носит название Rapid Economic Justification (REJ) - быстрое экономическое обоснование. Его используют для определения потенциальных выгод предприятия от автоматизации.

Оценка возможных эффектов от использования информационных систем и технологий в рамках этой методики проходит в пять шагов:

1. Оценка бизнеса – выделяются стратегические цели компании, их можно декомпозировать до уровня понятных «повседневных» проблем и задач. По каждой стратегической цели следует выделить ключевые показатели эффективности. Показатели эффективности должны быть измеримыми, процедура получения их значений – понятна, они должны количественно выражать степень достижения цели. В созданном перечне задач можно указать автоматизированные полностью, частично и неавтоматизированные.
2. Понимание решений – вырабатываются способы решения означенных ранее проблем, в том числе и с использованием АИС и ИТ. В большей степени можно сконцентрироваться на не полностью автоматизированных задах. Нужно понимать, что внедрение АИС может быть недостаточно для полного решения проблемы, необходимо сочетание с другими мероприятиями.
3. Понимание баланса – выполняется оценка улучшений, которые будут получены за счет реализации выбранных решений задач. Расчет количественных показателей производится на основе прогноза качественного улучшения протекания процессов на предприятии. Оценка качественных улучшений выполняется экспертным путем. Для преобразования качественных показателей в количественные используются методы классического экономического анализа. Улучшения определяются в двух направлениях: рост доходов и снижение расходов. Также на этом шаге оцениваются затраты на АИС по методике TCO.
4. Понимание рисков – предполагается по каждому решению оценить риски в координатах «вероятность - степень влияния», которые могут не позволить получить спрогнозированные качественные улучшения, и предлагаются мероприятия по их снижению.
5. Понимание финансовых метрик – на этом этапе нужно определить показатели эффективности такие как чистый приведенный доход, рентабельность, срок окупаемости.

Результаты работы на каждом шаге принято обобщать в виде таблиц.

Применение методики REJ в рамках оценки эффективности дипломного проекта возможно в несколько совращенном варианте: на первом шаге можно выделять только те стратегии и показатели эффективности, на которые разрабатываемая АИС окажет влияние; возможно в список рисков на четвертом шаге включить только основные. Пример применения методики приведен в приложении А.

* 1. **Социальный эффект**

Социальная результативность внедрения АИС определяется динамикой индикаторов, отражающих качество труда сотрудников предприятия и жизни населения, на которую оказывает влияние предприятие.

Одним из основных показателей по улучшению труда сотрудников фирмы, связанному с внедрением АИС, будет степень автоматизации:

Для примера автоматизации информирования родителей учеников средней школы об успеваемости через Internet, выделенный эффект – рост успеваемости учеников – будет носить социальный характер. Для каждой предметной области возможно выделить свои социальные показатели.

К социальному эффекту можно отнести и улучшение психологического климата в коллективе.

* 1. **Научный эффект**

Научная результативность внедрения АИС измеряется динамикой индикаторов разработки и внедрения новых научных достижений:

- внедрение прогрессивных методов управления;

- внедрение новых методов обработки информации.

* 1. **Организационный эффект**

Выражается в повышении управляемости предприятия, а именно в улучшении качества или увеличении количества управленческих решений. Описывая организационный эффект, необходимо перечислить управленческие решения, на принятие которых повлияет АИС. Нужно описать взаимосвязь между функциями АИС и управлением.

* 1. **Эффективность внедрения АИС (ПО ПРИМЕРУ)**

Показатели эффективности выражают отношение достигнутого уровня эффекта к затратам на его достижение. Применительно к АИС затраты, это суммы расходов на разработку и внедрение оцененные по методике TCO. Эффект, это выраженное в стоимостных (рубли) или натуральных (часы, штуки и т.д.) единицах улучшение в работе предприятия, достигнутое за счет внедрения АИС:

Выделяют частные показатели, оценивающие эффективность проекта:

* чистый приведенный доход:

,

где NPV – чистый приведенный доход;

IC – начальные инвестиции (стоимость этапа разработки и внедрения);

CF*t* – годовой доход, определяемый как разница между дополнительной прибылью и дополнительными расходами (эксплуатационными затратами);

*i* – ставка дисконтирования.

* Внутренняя норма доходности (IRR) определяется на основе предыдущей формулы. При этом NPV полагается равной 0.
* Окупаемость инвестиций (ROI), показывает сколько раз на период (год) инвестиции в АИС вернуться в виде прибыли:

Срок окупаемости величина обратная ROI, показывает сколько лет понадобится для возврата инвестиций в виде дохода. Можно выразить срок окупаемости в месяцах.

По перечисленным показателям можно сделать вывод о целесообразности разработки или приобретения АИС. Внедрение может быть названо эффективным, если есть положительный чистый приведенный доход, срок окупаемости меньше, чем срок полезного использования

* 1. **Расчёт показателей экономической эффективности проекта (ПО ПРИМЕРУ)**

Результаты расчета показателей экономической эффективности проекта необходимо представить в форме таблиц, графиков, повышающих наглядность восприятия. Здесь же следует определить улучшение качественных характеристик процесса управления соответствующим объектом и оценить влияние автоматизированного комплекса задач на эффективность деятельности органов управления и конечные результаты.

После расчета срока окупаемости проекта, все показатели эффективности должны быть проиллюстрированы диаграммами (например, круговыми или столбчатыми).

Приведем пример подобного расчета по методике REJ.

Рассмотрим предприятие, работающее в сфере оптовой торговли, имеющее большое количество средних и мелких клиентов. С большинством клиентов предприятие заключает долгосрочные договора, по которым в том числе предусматривается отпуск товаров с отсроченной оплатой, т.е. в кредит. Наличие таких условий в договорах является для фирмы одним из конкурентных преимуществ, от которого оно не может отказаться. Предприятию необходимы оборотные средства в большом объеме для поддержания широкого ассортимента предлагаемых товаров и необходимого уровня запасов. Для покрытия дефицита оборотных средств фирма часто обращается в банки за краткосрочными кредитами, средняя ставка по которым составляет 11,6%.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип клиентов | Кол-во клиентов / Кол-во договоров | Средний годовой оборот, тыс. руб. | Сумма дебиторской задолженности, всего по группе, тыс. руб. | Средний срок задолженности, дней |
| Крупные клиенты | 20 / 20 | 1 000 | 500 | 15 |
| Средние клиенты | 100 / 95 | 300 | 1 000 | 40 |
| Мелкие клиенты | 500 / 300 | 100 | 5 000 | 60 |

Рентабельность продаж фирмы за прошедший год составила 40%, средний уровень торговой наценки – 65%, объем привлеченных кредитных средств – 25 млн.руб. ПО данным бухгалтерского учета сумма долгосрочной дебиторской задолженности фирмы составляет 800 тыс.руб., списано безнадежных долгов 100 тыс.руб.

Стратегической целью компании является увеличение занимаемой доли рынка. Цель может быть декомпозирована на следующие подцели: увеличение среднего оборота по каждой группе клиентов и увеличение общего количества клиентов. Достижению и первой, и второй цели может способствовать расширение ассортимента предлагаемой продукции – это и привлечет новых клиентов и будет стимулировать уже имеющихся к большему объему закупок. Поддержание более широкого ассортимента товаров потребует больших вложений в оборотные средства, т.к. будет сопряжено с ростом складских запасов. Источником увеличения оборотных средств, которые будут вложены в создание дополнительного объема складских запасов, может стать сокращение дебиторской задолженности. Однако организация работы с клиентами на сегодняшний день не позволяет эффективно контролировать состояние дебиторской задолженности. В связи с этим фирма рассматривает возможность автоматизации данного процесса.

Рассмотрим порядок применения методики REJ для оценки эффективности автоматизации работы с дебиторской задолженностью.

**Шаг 1.** Исходя из изложенных выше соображений, можно составить таблицу, отражающую связь между целями фирмы и проблемами:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Фактор успеха | Ключевой показатель эф-ти | Стратегия | Проблема | Цель |
| Увеличение доли фирмы на рынке | Темп прироста объема продаж | Расширение ассортимента предлагаемых товаров | Недостаток средств для создания большего объема запасов | Сокращение неликвидных запасов |
| Сокращение текущих затрат |
| **Сокращение дебиторской задолженности** |

В качестве показателя эффективности выделен темп прироста объема продаж, а не доля фирмы на рынке, поскольку в конечном счете доля фирмы на рынке будет увеличиваться если темп роста реализации будет выше темпов роста рынка. Определение темпов прироста реализации для фирмы задача тривиальная. Этот показатель может быть рассчитан оперативно и служить реальным ориентиром для компании.

Далее сконцентрируемся на одной из представленных целей – сокращении дебиторской задолженности. Остальные цели можно проанализировать аналогичным образом.

|  |  |  |
| --- | --- | --- |
| Цель | Задачи 1 | Задачи 2 |
| Сокращение дебиторской задолженности | Контроль над сроками и условиями исполнения контрактов | Ведение реестра клиентов и условий контрактов |
| Уведомление дебиторов о сроках наступления платежей |
| Определение дебиторов, по которым возможно судебное решение вопроса |
| Контроль над условиями заключения новых контрактов | Разработка регламента работы с "проблемными" дебиторами |
| Ведение "черного списка" клиентов |

Кроме перечня задач второго уровня можно указать степень их автоматизации. Для рассматриваемого примера будем считать, что все задачи не автоматизированы, кроме ведения реестра клиентов (без указания условий контрактов).

**Шаг 2.** На втором шаге необходимо определиться, какие действия (решения) фирма могла бы предпринять для решения всех задач второго уровня. Один из вариантов представлен в таблице.

|  |  |  |
| --- | --- | --- |
| Задачи 2 | Решения | Результат |
| Уведомление дебиторов о сроках наступления платежей | Автоматическая рассылка электронных писем | Сокращение срока дебиторской задолженности на 40% |
| Формирование графика звонков |
| Ведение реестра клиентов и условий контрактов | Оперативная регистрация и получение информации об условиях контракта по любому клиенту |
| Определение дебиторов, по которым возможно судебное решение вопроса о взыскании задолженности | Формирование списка "злостных" неплательщиков | Сокращение суммы долгосрочной дебиторской задолженности на 30% |
| Подготовка документов для передачи в суд (акты сверки, перечень напоминаний и претензий и т.п.) |
| Разработка регламента работы с "проблемными" дебиторами | Определение порядка и условий работы с клиентами, допускавшими просрочки платежей | Снижение суммы списанных долгов на 80% |
| Ведение "черного списка" клиентов | Оперативное получение перечня клиентов-неплательщиков |
| Автоматический запрет на регистрацию контрактов с недобросовестными клиентами |

Результат от реализации предложенных решений на этапе подготовки к внедрению АИС может быть получен только экспертным путем в ходе консультаций с различными специалистами фирмы. Фактически, если способом реализации решений является внедрение АИС, то в колонке «Решения» предыдущей таблицы содержатся функции АИС.

**Шаг 3.** Задачей третьего шага является переход от прогнозируемых качественных улучшений работы фирмы к их количественной оценке. Для этого применяются методы классического экономического анализа, в том числе метод цепных подстановок.

В рассматриваемом примере для определения количественных выгод необходимо знать:

- сумму и сроки дебиторской задолженности в годовом исчислении;

- сумму долгосрочной дебиторской задолженности по итогам года;

- сумму безнадежных долгов, списанных по данным бухгалтерского учета.

Рассчитаем дополнительный приток денежных средств, учитывая данные предприятия и сокращение срока дебиторской задолженности на 40%. Если для всех групп клиентов сокращение срока дебиторской задолженности произойдет в равной степени, то фирма дополнительно получит в свое распоряжение всю сумму дебиторской задолженности на время равное 40% от срока задолженности. Фирма сэкономит на платежах процентов по кредиту, ведь эти сумму не придется занимать у банка.

Для крупных клиентов экономия составит тыс. руб.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип клиентов | Сумма деб. задолженности, тыс. руб. | Средний срок задолженности, дней | | Экономия на платежах %, тыс.руб. |
| до решения | после решения |
| Крупные клиенты | 500 | 15 | 9 | 0,953 |
| Средние клиенты | 1 000 | 40 | 24 | 5,085 |
| Мелкие клиенты | 5 000 | 60 | 36 | 38,137 |
| Итого: | | | | 44,175 |

Сокращение суммы долгосрочной дебиторской задолженности на 30% позволит предприятию высвободить для оборота 800\*0,3=240 тыс. руб.

Снижение суммы списанных долгов на 80% позволит предприятию высвободить для оборота 100\*0,8=80 тыс. руб.

Таким образом, годовой объем высвобождаемых средств составит 364,175 тыс. руб. Предполагая, что все высвободившиеся средства будут направлены на увеличение закупок товаров для продажи, то можно ожидать прироста выручки (объема продаж в стоимостном выражении) с учетом среднего уровня торговой наценки 364,175\*1,65=600,889 тыс.руб. Учитывая, что объем продаж фирмы составляет 100 млн. руб., темп прироста объема продаж за счет автоматизации будет 600,889/100000\*100%=0,6%

Дополнительная валовая прибыль, которую получит фирма с учетом рентабельности продаж составит 600,889\*0,4=240,356 тыс.руб.

После определения эффектов нужно оценить затраты на реализацию предлагаемых решений. Когда речь идет о внедрении АИС, выполняется оценка ее стоимости по методике TCO. Совокупная стоимость владения определяется на период 3 года.

|  | Статьи затрат | Расчет |
| --- | --- | --- |
| **Этап разработки** | Разработку модуля по работе с дебиторской задолженностью предприятие ведет силами собственного отдела ИТ. Время разработки АИС от формирования технического задания до завершения тестирования составляет 6 месяцев.  Для работы над проектом выделена группа из 2 программистов, которые выполняют весь перечень работ по проектированию, кодированию и отладке ПО. Их рабочие места оборудованы персональными компьютерами (ПК), инвентарная стоимость которых 50 тыс.руб. Разработка ведется в среде C# (приобретается подписка MS Visual Studio сроком на 1 год); проектирование - MS Visio Pro (имеется установленная версия, используемая в повседневной работе); системное ПО установлено на ПК и используется совместно с ними (входит в стоимость ПК); СУБД MS SQL Server 2012 (стоимость 440 тыс. руб.) установлена на сервере предприятия, функционирует круглосуточно и уже используется бухгалтерским приложением (6 рабочих мест), модулем складского учета (4 рабочих места) и модулем логистики (2 рабочих места). | |
| Стоимость оборудования для разработки и затраты на его эксплуатацию в период разработки | Учитывая, что ПК используются предприятием и вне разработки АИС, на стоимость разработки можно отнести только часть их стоимости, пропорционально времени использования, исходя из срока полезной эксплуатации (5 лет):  Затраты на эксплуатацию оборудования включаются в косвенные расходы. |
| Стоимость ПО для проектирования и кодирования АИС; СУБД и прочее необходимое разработчику ПО | - Стоимость подписки MS Visual Studio включается полностью в состав затрат, поскольку ее использование в других проектах не предусматривается. Она составляет **44 тыс. руб.**  - Стоимость MS Visio Pro учитывается в затратах пропорционально сроку использования (3 года), поскольку применяется для решения и других задач:  - Стоимость СУБД распределяется между различными приложения, использующими ее, пропорционально количеству рабочих мест (всего без учета этого проекта 12) и сроку эксплуатации (3 года). На период разработки рабочих мест - 2. |
| Оплата труда проектировщиков и разработчиков | - Программисты, занятые в проекте, оплачиваются повременно, исходя месячного оклада 25000 руб.  - Начисления на их заработную плату составят 30,6% от фонда оплаты: |
| Затраты на содержание и обслуживание помещений, в т.ч. охрану.  Прочие расходы. | Учитываются в составе косвенных затрат. |
| Косвенные расходы | Сумма косвенных расходов, относимая на ИТ службу предприятия, ежемесячно составляет 20 тыс. руб. ФОТ подразделения – 120 тыс. На стоимость проекта будет отнесено: |
| **Итого затраты на этапе разработки** | | **467 276 руб.** |
| **Этап внедрения** | Внедрение АИС будет осуществляться силами программистов, выполнявших разработку. Рабочее место пользователя дополнительно оснащается принтером. Обучение работе с АИС проводится в рабочее время. | |
| Стоимость оборудования (рабочие места пользователей) | Принтер – **5000 руб.** |
| Оплата труда по установке | Установка АИС займет 1 час. Стоимость часа работы программиста (количество часов в месяце в среднем 168): |
| Затраты на обучение персонала | Обучение займет 4 часа работы программиста и пользователя (при часовой ставке пользователя 110 руб.) |
| Затраты на содержание и обслуживание помещений, в т.ч. охрану. | Отсутствуют, поскольку не предусмотрено привлечение дополнительных площадей. |
| **Итого затраты на этапе внедрения** | | **6 190 руб.** |
| **Этап эксплуатации** | Эксплуатационные затраты определяем в годовом исчислении. | |
| Оплата труда персонала, обеспечивающего эксплуатацию (сисадмин, сотрудники ИТ-служб); | Время обслуживания АИС в среднем составляет 1 час в течение рабочего дня (247 рабочих дней в год), отсюда: |
| Затраты на профилактические работы | Профилактические работы выполняются 1 раз в месяц в течение 1 часа:  Расходные материалы (по смете) – 10 \*12=**120** руб. |
| Затраты на содержание и обслуживание помещений, в т.ч. охрану | Отсутствуют. |
| Стоимость сбоев и простоев АИС | Экономические потери предприятие понесет в случае простоя АИС в течение 5 и более рабочих дней (возможность заключения сделок с контрагентами из «черного списка», несвоевременное оповещение дебиторов о сроках наступления платежей). Сумма ущерба оценивается экспертно исходя из соображений, учтенных при расчете эффекта АИС, и составит 50 тыс.руб. в годовом исчислении. Вероятность такого сбоя складывается из возможности отказа, СУБД, сетевого оборудования, наличия необнаруженных ошибок в ПО. Поскольку время восстановления АИС при реализации перечисленных сбоев не более 5 дней, то риск возникновения экономических потерь можно оценить как незначительный (менее 1 %). Таким образом оценка риска 50000\*0,01=**500 руб.** |
| Потери времени на сомообучение и самостоятельное решение проблем | Отсутствуют |
| **Итого эксплуатационные расходы в год** | | **39 470 руб.** |

Затраты на разработку и внедрение составят примерно 475 тыс. руб., эксплуатационные расходы почти – 40 тыс.руб. в год.

**Шаг 4.** Оценка рисков позволит определить насколько полученные результаты могут стать меньше ожидаемого уровня. Каждый выделенный риск может потенциально сократить предполагаемые эффекты. Риски нужно формулировать как вероятность недополучения результатов, сформулированных как качественные показатели. При этом нужно разделять риски, имеющие влияние на один и тот же результат, по причинам их возникновения.

В соответствии с методикой REJ выделяют пять видов рисков. По каждому риску оценивается его влияние на результат и вероятность реализации по пятибалльной шкале.

Риск соответствия. Чем жестче соответствие ИТ-проекта целям предприятия, тем меньше риск. Необходимо заметить, что для некоторых проектов установление четкого соответствия технологий стратегическим целям бизнеса задача сложно выполнимая (например, усовершенствование инфраструктуры информационной системы), однако инвестиции в них являются необходимыми для дальнейшего развития информационных технологий.

Реализационный риск. Учитывает возможность того, что реальная стоимость реализации проекта будет отличаться от расчетной.

Операционный риск. Учитывает возможность того, что стоимость функционирования системы будет отличаться от предполагаемой.

Технологический риск. Чем больше известно о выбранном решении и чем проработанней выбранные технологии, тем меньше этот риск. Однако проекты с малым значением технологического риска не всегда обеспечивают достаточно высокие потенциальные преимущества. Как говорится: «Чем выше риск, тем выше прибыль».

Риск денежных потоков. Учитывает возможность недостоверного определения выгод от проекта и неточного расчета положительных денежных потоков, а также возможность появления других непредвиденных финансовых проблем. Например, будет принято решение увеличить капитализацию бизнеса или другие, более важные с точки зрения руководства, проблемы потребуют отвлечения средств от рассматриваемого проекта, в результате чего не удастся достичь предполагаемых выгод в полном объеме.

**Шаг 5.** Оценим показатели эффективности проекта. Ориентируясь на полученный на третьем шаге объем валовой прибыли можно оценить чистый приведенный доход. Первоначальные вложения - стоимость разработки и внедрения АИС – составляют 475 тыс.руб.; дополнительная прибыль примерно 240 тыс.руб. в год; дополнительные издержки (затраты на этапе эксплуатации) – 40 тыс. руб. в год; период (n) 3 года; ставка дисконтирования - для простоты используем ставку по коммерческим кредитам – 11,6%. Таким образом, чистый приведенный доход составит:

Внутренняя норма доходности проекта составит около 12,6%.

Срок окупаемости капитальных вложений составит: (475/200)\*12=2,4 года

Анализируя значения показателей, можно утверждать, что инвестирование средств в автоматизацию работы с дебиторской задолженностью будет целесообразным, т.к. чистый приведенный доход больше нуля, а срок окупаемости меньше, чем предполагаемый срок полезной эксплуатации

* 1. **Выводы по главе 3**

Оценка экономической целесообразности работы.

Выбор программных и организационно–технологических проектных решений обеспечил минимизацию финансовых, материальных и трудовых затрат. Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх. Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

Разработка является экономически целесообразной.

**Список использованных источников (ЭТО ДЛЯ ЭКОНОМИЧЕСКОГО РАЗДЕЛА, НО ВСТАВЛЯЕМ В ОБЩИЙ СПИСОК ЛИТЕРАТУРЫ ТЕ ИСТОЧНИКИ, КОТОРЫЕ РЕАЛЬНО ИСПОЛЬЗОВАЛИСЬ)**

1. Анфилатов В.С. Системный анализ в управлении: учеб. Пособие/ В.С. Анфилатов, А.А. Емельянов, А.А. Кукушкин; под. ред. А.А. Емельянова.- М.: Финансы и статистика, 2007.-368 с.– 5/1/Э
2. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: Учебник.-2-е изд., перераб и доп.- М.: Финансы и статистика, 2006.-544 с. 92/5/Э
3. Мишенин А.И. Теория экономических информационных систем: учебник, 4-е изд., доп и перераб..- М.: Финансы и статистика. 2007.-240с.: ил. 66/5/Э
4. Сатунина А.Е. Управление проектом корпоративной информационной системы предприятия: учеб. Пособие / А.Е. Сатунина, Л.А. Сысоева.- М.: Финансы и статистика; Инфра-М, 2009.-352с.: ил. 5/1/Э.
5. Системный анализ в менеджменте: электронный учебник / В.Н. Попов, В.С. Касьянов, И.П. Савченко. - Электрон. дан. - М.: КНОРУС, 2010/Э
6. Смирнова Г. Н. Проектирование экономических информационных систем : учеб. для эконом. вузов по специальностям "Прикладная информатика в экономике", "Прикладная информатика в менеджменте", "Прикладная информатика в юриспруденции" / Г. Н. Смирнова, А. А. Сорокин, Ю. Ф. Тельнов. - М. : Финансы и статистика, 2006. - 511 с. 58/5/Э
7. Теория систем и системный анализ в управлении организациями: Справочник: учеб. Пособие/ под ред. В.Н. Волковой и А.А. Емельянова.-М.: Финансы и статистика, 2006. - 848 с. 51/5/Э

**ЗАКЛЮЧЕНИЕ**

В данной выпускной квалификационной работе были успешно выполнены все поставленные задачи, направленные на создание интеллектуальной системы распознавания шахматных фигур для Центра дистанционных образовательных технологий университета Витте. В процессе исследования проведен всесторонний анализ предметной области, что позволило глубоко понять текущие потребности и определить основные направления развития системы.

Первоначально были сформулированы необходимые функциональные требования к разрабатываемой информационной системе. Это включало определение ключевых функций, таких как автоматическое распознавание типов и цветов фигур, анализ позиций на доске, интеграция с образовательными платформами и создание удобного пользовательского интерфейса для взаимодействия с системой. Такой подход обеспечил создание системы, максимально соответствующей потребностям конечных пользователей и способной эффективно поддерживать образовательный процесс.

Далее проведен подробный анализ баз данных и средств разработки, что позволило выбрать оптимальные технологии для реализации проекта. Особое внимание уделено выбору системы управления базами данных, обеспечивающей надежное хранение и быстрый доступ к большим объемам информации. Анализ средств разработки включал оценку различных платформ и фреймворков, что способствовало выбору наиболее подходящих инструментов для разработки веб-приложения системы распознавания шахматных фигур.

Построена модель данных на физическом и логическом уровнях, что обеспечило четкую структуру хранения информации и эффективное взаимодействие между различными компонентами системы. Разработка модели данных включала создание сущностей, их атрибутов и взаимосвязей, что позволило обеспечить целостность и консистентность данных при их обработке и хранении.

На основе разработанной модели данных была создана интеллектуальная система распознавания шахматных фигур, включающая все предусмотренные функциональные модули. В процессе разработки особое внимание уделено пользовательскому интерфейсу, обеспечивающему интуитивно понятное взаимодействие с системой. Реализованы механизмы управления доступом, что позволяет различным категориям пользователей (преподавателям, студентам, администраторам) иметь соответствующие уровни прав и возможностей в системе.

В рамках работы также проведена экономическая оценка целесообразности разработки и внедрения системы. Расчет затрат на разработку включал оценку стоимости программного обеспечения, оборудования, трудозатрат команды разработчиков и расходов на обучение персонала. Анализ экономической эффективности показал, что внедрение системы распознавания шахматных фигур позволит существенно снизить операционные расходы, связанные с ручной обработкой изображений, повысить доступность и удобство использования образовательных материалов, а также способствовать улучшению качества образовательного процесса.

В результате выполнения всех задач была разработана и внедрена полноценная интеллектуальная система распознавания шахматных фигур, которая отвечает всем предъявляемым требованиям и значительно повышает эффективность работы с шахматным контентом в образовательном учреждении. Система интегрирована с существующими информационными системами учреждения, что обеспечивает бесшовное взаимодействие и обмен данными между различными платформами.

Методология «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ» была успешно применена на всех этапах разработки, что позволило выявить существующие недостатки в текущей системе анализа шахматных партий и разработать оптимизированные решения для их устранения. Это обеспечило создание системы, соответствующей современным требованиям и способной адаптироваться к изменяющимся потребностям пользователей.

Несмотря на достигнутые результаты, в перспективе планируется дальнейшее развитие системы. В частности, рассматривается возможность расширения функционала за счет внедрения дополнительных модулей, таких как система рекомендаций по стратегиям игры, интеграция с внешними шахматными платформами и улучшение механизмов аналитики. Также предусмотрено постоянное обновление и оптимизация системы для поддержания ее актуальности и соответствия современным технологическим стандартам.

Таким образом, выполненная работа продемонстрировала высокий уровень подготовки и способности к реализации комплексных информационных систем. Разработанная интеллектуальная система распознавания шахматных фигур значительно улучшает процессы анализа шахматных партий, повышает доступность информации для пользователей и способствует эффективному использованию образовательных ресурсов в учреждении.

**СПИСОК ЛИТЕРАТУРЫ**

СПИСОК ЛИТЕРАТУРЫ ДОЛЖЕН БЫТЬ ОФОРМЛЕН ПО ГОСТ\_Р\_7\_0\_100\_2018

Можно воспользоваться ресурсом: https://perviy-vestnik.ru/literatura/?ysclid=lyfqldzbff465005388

1. Геворкян, Э. А. К теории распространения электромагнитных волн в волноводе с магнитоактивным анизотропным модулированным заполнением / Э. А. Геворкян // Радиотехника и электроника. – 2008. – Т. 53. – № 5. – С. 565-569. (ВОТ ЭТО ПРАВИЛЬНО!!!!)
2. Ресурс <http://www.mysql.ru/docs/>; (вот это **НЕ ПРАВИЛЬНО!!!! и ниже)**
3. Ресурс https://ru.wikipedia.org/wiki/;
4. Ресурс<https://www.php.net/manual/ru/language.functions.php>;
5. Ресурс <https://www.tadviser.ru/>;
6. Самоучитель PHP 7. Максим Кузнецов, Игорь Симдянов. изд. 2018;
7. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера, 4-е издание,.Николай Прохоренок, Владимир Дронов., изд. 2015;
8. PHP и MySQL. Разработка веб-приложений, Денис Колисниченко., изд. 2017;
9. Милл А. Docker на практике, Сейерс Э. Х., изд. 2019;
10. Создание микросервисов, Ньюмен С. изд. 2016;
11. Информационные системы и технологии в экономике 3-е изд. Исправленное и дополненное, Нетёсова О. Ю. изд, 2019;
12. Компьютерные сети. Принципы, технологии, протоколы. Олифер Н.А., Олифер В.Г., изд. 2017;
13. Философия Java 4-е. Брюс Э. изд. 2019;
14. Структуры данных и алгоритмы в Java. Лафоре Р.,изд. 2018;
15. Базы данных. Практическое применение СУБД SQL и NoSOL-типа для применения проектирования информационных систем. Мартишин Сергей Анатольевич., изд. 2021;

**ПРИЛОЖЕНИЯ**

**Приложение 1. Техническое задание на разработку корпоративной электронной библиотеки в университете**

**ТЗ СТРОГО ПО ГОСТ 34.602-2020!!!!**

ГОСТ 34.602-2020. Межгосударственный стандарт. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы" (введен в действие Приказом Росстандарта от 19.11.2021 N 1522-ст)

ТЗ на АС является основным документом, определяющим требования и порядок создания автоматизированной системы, в соответствии с которым проводится разработка АС и ее приемка. ТЗ на другие виды работ (разработка, модернизация, развертывание, обновление отдельных модулей в составе АС и т.п.) также должно быть оформлено строго по ГОСТ 34.602-2020. При этом название вида работ отражается в наименовании ТЗ.

* + - * 1. **Общие сведения**

В данном разделе указывают следующее:

- полное наименование АС и ее условное обозначение;

- шифр темы (при наличии);

- наименование организации - заказчика АС, наименование организации-разработчика (при наличии сведений о ней);

- перечень документов, на основании которых создается АС, кем и когда утверждены эти документы;

- плановые сроки начала и окончания работ по созданию АС;

- общие сведения об источниках и порядке финансирования работ

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

* + - * 1. **Цели и назначение создания автоматизированной системы**

Раздел состоит из следующих подразделов:

- цели создания АС;

- назначение АС.

В подразделе "Цели создания АС" приводят наименования и требуемые значения технических, технологических, производственно-экономических или других показателей объекта автоматизации, которые должны быть достигнуты в результате создания АС, и указывают критерии оценки достижения целей создания АС.

В подразделе "Назначение АС" указывают вид автоматизируемой деятельности (управление, проектирование и т.п.) применительно к объекту автоматизации в целом.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

* + - * 1. **Характеристика объектов автоматизации**

В этом разделе приводят следующую информацию:

- основные сведения об объекте автоматизации или ссылки на документы, содержащие такие сведения;

- сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

* + - * 1. **Требования к автоматизированной системе**

Раздел состоит из следующих подразделов:

- требования к структуре АС в целом;

- требования к функциям (задачам), выполняемым АС;

- требования к видам обеспечения АС;

- общие технические требования к АС.

Состав требований к АС, включаемых в данный раздел ТЗ на АС, устанавливают в зависимости от вида, назначения, специфических особенностей и условий функционирования конкретной автоматизированной системы. В каждом подразделе приводят ссылки на действующие НТД, определяющие требования к автоматизированным системам соответствующего вида.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

* + - * 1. **Состав и содержание работ по созданию автоматизированной системы**

Раздел должен содержать перечень этапов работ по созданию АС и сроки их выполнения. Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

* + - * 1. **Порядок разработки автоматизированной системы**

В разделе приводят следующее:

- порядок организации разработки АС;

- перечень документов и исходных данных для разработки АС;

- перечень документов, предъявляемых по окончании соответствующих этапов работ;

- порядок проведения экспертизы технической документации;

- перечень макетов (при необходимости), порядок их разработки, изготовления, испытаний, необходимость разработки на них документации, программы и методик испытаний;

- порядок разработки, согласования и утверждения плана совместных работ по разработке АС;

- порядок разработки, согласования и утверждения программы работ по стандартизации;

- требования к гарантийным обязательствам разработчика;

- порядок проведения технико-экономической оценки разработки АС;

- порядок разработки, согласования и утверждения программы метрологического обеспечения, программы обеспечения надежности, программы эргономического обеспечения.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

* + - * 1. **Порядок контроля и приемки автоматизированной системы**

В разделе указывают следующую информацию:

- виды, состав и методы испытаний АС и ее составных частей;

- общие требования к приемке работ, порядок согласования и утверждения приемочной документации;

- статус приемочной комиссии (государственная, межведомственная, ведомственная и др.)

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

* + - * 1. **Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие**

В разделе приводят перечень мероприятий, которые необходимо осуществить при подготовке объекта автоматизации к вводу АС в действие.

В перечень мероприятий включают следующее:

- создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой АС требованиям, содержащимся в ТЗ на АС;

- проведение необходимых организационно-штатных мероприятий;

- порядок обучения персонала и пользователей АС

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

* + - * 1. **Требования к документированию**

В разделе приводят следующую информацию:

- перечень подлежащих разработке документов;

- вид представления и количество документов;

- требования по использованию ЕСКД и ЕСПД при разработке документов.

При отсутствии государственных стандартов, определяющих требования к документированию элементов АС, дополнительно включают требования к составу и содержанию таких документов.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх.

* + - * 1. **Источники разработки**

В разделе "Источники разработки" должны быть перечислены документы и информационные материалы (технико-экономическое обоснование, отчеты о законченных научно-исследовательских работах, информационные материалы на отечественные, зарубежные системы-аналоги и др.), на основании которых разрабатывалось ТЗ и которые должны быть использованы при создании АС.

Хххххх хххх хххххххх хххххх, ххххххх ххххххх х ххххх хххххх ххххх. Ххх хх хх ххххххх хххххххх ххххххх. Ххххх ххххххххххх хххххх, хххххх, ххххххххххх хххххх хххххххх ххх хххххххх хххххххх хххххххх ххххх. Хххх ххххххххх хххххххххх ххххххххх ххххххх. Хх хххххххх хххххххххх хххххххх ххххххх ххххххххххх ххх ххххххх ххххххх ххххх хххххх ххххххх. Ххххх

**Приложение 2. Исходный код “Авторизация”**

**Добавить ссылку на Git**

<!--Проверка авторизован ли пользователь-->

<?php

session\_start - );

if - isset - $\_SESSION['uid']) & isset - $\_SESSION['nm']))

{

$id = $\_SESSION['uid'];

$redir = "Location: $id"."prof.php";

header - "$redir");

exit - );

}

?>

<!--Стили блоков-->

<style type="text/css">

.buttons

{

background-color: #FF0000;

width: 110px;

height: 25px;

cursor: pointer;

color: #FFFFFF;

}

</style>

<!--Стиль блока авторизации-->

<style type="text/css">

.loginstyle

{

background-color: #CACACA;

position: absolute;

left: 39%;

top: 35%;

color: #000000;

border:5px;

border-spacing: 3px 10px;

border: 3px solid;

}

</style>

<!--Стиль тела сайта-->

<style type="text/css">

body

{

background:url - logos/mainlog.jpg) fixed no-repeat center;

background-size: 100%;

-moz-background-size: 100%; /\* Firefox 3.6+ \*/

-webkit-background-size: 100%; /\* Safari 3.1+ и Chrome 4.0+ \*/

-o-background-size: 100%; /\* Opera 9.6+ \*/

}

</style>

<!--Стиль нижнего блока-->

<style type="text/css">

.nblock

{

position: fixed;

bottom:0;

color: #FFFFFF;

border: 2px solid;

background-color: #FF4500;

width: 100%;

}

</style>

<!--HTML структура-->

<html>

<head>

<title>Вход в систему учета заявок "RTSuppService"</title>

</head>

<body>

<form method="post">

<table class="loginstyle" width="400" height="250">

<tr><td></td></tr>

<tr><td><b><p align="center">Ваш логин:</p></b></td></tr>

<tr><td><p align="center"><input type="text" name="login" maxlength="15" pattern="[A-Za-z-0-9]{4,15}" style="width:140"></p></td></tr>

<tr><td><b><p align="center">Ваш пароль:</p></b></td></tr>

<tr><td><p align="center"><input type="password" name="pass" maxlength="15" pattern="[A-Za-z-0-9]{4,15}" style="width:140"></p></td></tr>

<tr><td><p align="center"><input class="buttons" type="submit" value="Войти">&nbsp &nbsp <input class="buttons" type="submit" formaction="reg.php" value="Информация"></p></td></tr>

</table>

</form>

</body>

<table class="nblock">

<tr><td><p align="center">RTSuppService Rostelecom's CRM system Москва 2020г.</p></td></tr>

</table>

</html>

<!--PHP скрипт,обработка форм-->

<?php

//Проверка существования переменной

if - isset - $\_POST['login']) && isset - $\_POST['pass']))

{

$login = $\_POST['login'];

$pass = $\_POST['pass'];

//Проверка на пустое поле

if - !empty - $login && $pass))

{

//sql запросы в БД

$link = mysqli\_connect - "localhost","root","9522687799");

mysqli\_select\_db - $link,'usrid');

$query = mysqli\_query - $link,"select uid,uname,usurname,firstname,upass,ugroup from sp\_users WHERE ulogin='".mysqli\_real\_escape\_string - $link,$\_POST['login'])."' LIMIT 1");

$data = mysqli\_fetch\_assoc - $query);

if - $data['upass'] == $\_POST['pass'])

{

if - !isset - $\_SESSION['login']))

{

session\_start - );

$nm = $data['uname'];

$usur = $data['usurname'];

$fnm = $data['firstname'];

$uid = $data['uid'];

$group = $data['ugroup'];

$\_SESSION['nm'] = $nm;

$\_SESSION['usurname'] = $usur;

$\_SESSION['firstname'] = $fnm;

$\_SESSION['uid'] = $uid;

$\_SESSION['ugroup'] = $group;

$profilenm = "Location: $uid"."prof.php";

header - "$profilenm");

}

}

else

{

//echo "<br><br><br><br><br><br><p align=center><img src=smiles/grust.gif><br></p>";

}

}

else

{

//echo "<br><br><br><br><br><p align=center><img src=smiles/vvod.gif></p>";

return;

}

}

?>

**Приложение 3. Руководство администратора корпоративной информационной системы**

Для дальнейшей эксплуатации информационной системы, необходимо составить сопроводительный документ – “руководство администратора”.

Сама структура и все необходимые требования к содержанию руководства пользователя по ГОСТ 34, описаны в РД 50-34.698-90.

В руководстве администратора отражаем АДМИНИСТРИРОВАНИЕ СИСТЕМЫ! Сюда относится – установка и развертывание ПО, интеграция с существующими системами, настройка системных файлов, настройка прав доступа

* + - * 1. **Введение**

В разделе "Введение" указывают:

1) область применения;

2) краткое описание возможностей;

3) уровень подготовки пользователя;

4) перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

* + - * 1. **Назначение и условия применения**

В разделе "Назначение и условия применения" указывают:

1) виды деятельности, функции, для автоматизации которых предназначено данное средство автоматизации;

2) условия, при соблюдении (выполнении, наступлении) которых обеспечивается применение средства автоматизации в соответствии с назначением (например, вид ЭВМ и конфигурация технических средств, операционная среда и общесистемные программные средства, входная информация, носители данных, база данных, требования к подготовке специалистов и т.п.)

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

* + - * 1. **Подготовка к работе**

В разделе "Подготовка к работе" указывают:

1) состав и содержание дистрибутивного носителя данных;

2) порядок загрузки данных и программ;

3) порядок проверки работоспособности.

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.,

* + - * 1. **Описание операций**

В разделе "Описание операций" указывают:

1) описание всех выполняемых функций, задач, комплексов задач, процедур;

2) описание операций технологического процесса обработки данных, необходимых для выполнения функций, комплексов задач (задач), процедур.

Для каждой операции обработки данных указывают:

1) наименование;

2) условия, при соблюдении которых возможно выполнение операции;

3) подготовительные действия;

4) основные действия в требуемой последовательности;

5) заключительные действия;

6) ресурсы, расходуемые на операцию.

В описании действий допускаются ссылки на файлы подсказок, размещенные на магнитных носителях.

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

* + - * 1. **Аварийные ситуации**

В разделе "Аварийные ситуации" указывают:

1) действия в случае несоблюдения условий выполнения технологического процесса, в том числе при длительных отказах технических средств;

2) действия по восстановлению программ и (или) данных при отказе магнитных носителей или обнаружении ошибок в данных;

3) действия в случаях обнаружения несанкционированного вмешательства в данные;

4) действия в других аварийных ситуациях

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

* + - * 1. **Рекомендации по освоению**

В разделе "Рекомендации по освоению" указывают рекомендации по освоению и эксплуатации, включая описание контрольного примера, правила его запуска и выполнения

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

**Приложение 4. Руководство пользователя корпоративной информационной системы**

Для дальнейшей эксплуатации информационной системы, необходимо составить сопроводительный документ – “руководство пользователя”.

Сама структура и все необходимые требования к содержанию руководства пользователя по ГОСТ 34, описаны в РД 50-34.698-90.

В руководстве отражаем работу с системой с точки зрения ПОЛЬЗОВАТЕЛЯ: на какую кнопку нажать чтобы распечатать документ, в каком окне какие данные заполнить для формирования карточки клиента и т.п.

* + - * 1. **Введение**

В разделе "Введение" указывают:

1) область применения;

2) краткое описание возможностей;

3) уровень подготовки пользователя;

4) перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

* + - * 1. **Назначение и условия применения**

В разделе "Назначение и условия применения" указывают:

1) виды деятельности, функции, для автоматизации которых предназначено данное средство автоматизации;

2) условия, при соблюдении (выполнении, наступлении) которых обеспечивается применение средства автоматизации в соответствии с назначением (например, вид ЭВМ и конфигурация технических средств, операционная среда и общесистемные программные средства, входная информация, носители данных, база данных, требования к подготовке специалистов и т.п.)

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

* + - * 1. **Подготовка к работе**

В разделе "Подготовка к работе" указывают:

1) состав и содержание дистрибутивного носителя данных;

2) порядок загрузки данных и программ;

3) порядок проверки работоспособности.

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.,

* + - * 1. **Описание операций**

В разделе "Описание операций" указывают:

1) описание всех выполняемых функций, задач, комплексов задач, процедур;

2) описание операций технологического процесса обработки данных, необходимых для выполнения функций, комплексов задач (задач), процедур.

Для каждой операции обработки данных указывают:

1) наименование;

2) условия, при соблюдении которых возможно выполнение операции;

3) подготовительные действия;

4) основные действия в требуемой последовательности;

5) заключительные действия;

6) ресурсы, расходуемые на операцию.

В описании действий допускаются ссылки на файлы подсказок, размещенные на магнитных носителях.

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

* + - * 1. **Аварийные ситуации**

В разделе "Аварийные ситуации" указывают:

1) действия в случае несоблюдения условий выполнения технологического процесса, в том числе при длительных отказах технических средств;

2) действия по восстановлению программ и (или) данных при отказе магнитных носителей или обнаружении ошибок в данных;

3) действия в случаях обнаружения несанкционированного вмешательства в данные;

4) действия в других аварийных ситуациях

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.

* + - * 1. **Рекомендации по освоению**

В разделе "Рекомендации по освоению" указывают рекомендации по освоению и эксплуатации, включая описание контрольного примера, правила его запуска и выполнения

Ххххх ххххххх хххххх хххххх ххххххх ххххххх ххххх ххх ххххх. Хххх ххххх ххххх хххххххх хххх хххх хх ххххххх хххххххх хх х х ххххххх. Ххх ххх ххххххх ххххххх ххххххх ххххххх ххххх хххххххххх ххххххх ххххххххх хххххх.