

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

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(Наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки / специальности)

Бизнес-информатика

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Автоматизированная система учета технического
обслуживания и ремонта оборудования на предприятии

Обучающийся

П.А. Домницкий

(Инициалы Фамилия)

(личная подпись)

Руководитель

О.В. Оськина

(ученая степень (при наличии), ученое звание (при наличии) , Инициалы Фамилия)

Тольятти 2024

Аннотация

Выпускная квалификационная работа посвящена автоматизированной системе учета технического обслуживания и ремонта оборудования на предприятии.

Целью данной работы является разработка автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии, которая позволит автоматизировать процессы планирования, контроля и учета ТОиР.

Актуальность выбранной темы заключается в следующем. Современные предприятия стремятся к повышению эффективности своей деятельности, что невозможно без надежного и бесперебойного функционирования оборудования. Техническое обслуживание и ремонт (ТОиР) оборудования играют ключевую роль в обеспечении стабильности производственных процессов и минимизации непредвиденных простоев.

Задачи работы:

- рассмотреть причины, обуславливающие необходимость автоматизации учета ТОиР;
- проанализировать существующие системы учета ТОиР, выявить их преимущества и недостатки;
- обосновать выбор языка программирования и базы данных для разработки системы;
- разработать структурную схему системы, описывающую основные функциональные блоки;
- разработать пользовательский интерфейс программы, обеспечивающий удобство работы с системой;
- провести расчет затрат на разработку системы;
- оценить срок окупаемости системы, основываясь на возможной экономии и улучшении производственных процессов.

Результатом работы будет являться то, что предложенная автоматизированная система учета будет иметь значительное влияние на повышение производственной эффективности, что в долгосрочной перспективе принесет значительную экономическую выгоду для предприятия.

Первая глава работы раскрывает актуальность разработки системы учета технического обслуживания и ремонта оборудования на предприятии, выявляет значимость системы учета ТОиР, содержит существующие системы учета технического обслуживания и ремонта оборудования на предприятии, их подвиды, а также определяет ключевую роль системы учета ТОиР.

Вторая глава посвящена основным языкам программирования, подходящим для разработки подробной системы, определяет их преимущества, обосновывает выбор системы базы данных, факторы базы данных, определяются наиболее подходящие варианты баз данных для системы учета ТОиР, представляет собой разработку структурной схемы системы учета технического обслуживания и ремонта оборудования на предприятии, проводятся алгоритмы работы программного обеспечения, описывается интерфейс программы, определяются этапы и задачи.

Третья глава, посвящается расчету затрат на разработку автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии, содержит расчет срока окупаемости автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии.

Данная выпускная квалификационная работа состоит из 51 страницы текста, 7 рисунков и 33 источника.

Содержание

Введение.....	5
1. Актуальность темы исследования	7
1.1 Актуальность разработки системы учета технического обслуживания и ремонта оборудования на предприятии	7
1.2 Существующие системы учета технического обслуживания и ремонта оборудования на предприятии	11
2. Разработка системы учета технического обслуживания и ремонта оборудования на предприятии	16
2.1 Выбор и обоснование языка программирования	16
2.2 Выбор и обоснование базы данных.....	25
2.3 Разработка структурной схемы системы учета технического обслуживания и ремонта оборудования на предприятии	28
2.4 Описание интерфейса программы.....	33
3. Расчет экономической эффективности	44
3.1 Расчет затрат на разработку автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии	44
3.2 Расчет срока окупаемости автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии	46
Заключение	49
Список используемой литературы	51

Введение

Актуальность работы. Современные предприятия стремятся к повышению эффективности своей деятельности, что невозможно без надежного и бесперебойного функционирования оборудования. Техническое обслуживание и ремонт (ТОиР) оборудования играют ключевую роль в обеспечении стабильности производственных процессов и минимизации непредвиденных простоев.

Однако ручное управление этими процессами зачастую приводит к ошибкам, задержкам и недостаточной оптимизации. Внедрение автоматизированной системы учета ТОиР позволяет не только повысить точность учета операций, но и существенно улучшить планирование и контроль ремонтных мероприятий, что, в свою очередь, способствует снижению эксплуатационных затрат и увеличению срока службы оборудования.

Таким образом, разработка и внедрение автоматизированной системы учета ТОиР является важной задачей для предприятий, ориентированных на долгосрочное развитие и конкурентоспособность.

Цель работы: является разработка автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии, которая позволит автоматизировать процессы планирования, контроля и учета ТОиР.

Задачи работы:

- рассмотреть причины, обуславливающие необходимость автоматизации учета ТОиР;
- проанализировать существующие системы учета ТОиР, выявить их преимущества и недостатки;
- обосновать выбор языка программирования и базы данных для разработки системы;

- разработать структурную схему системы, описывающую основные функциональные блоки;
- разработать пользовательский интерфейс программы, обеспечивающий удобство работы с системой;
- провести расчет затрат на разработку системы;
- оценить срок окупаемости системы, основываясь на возможной экономии и улучшении производственных процессов.

1. Актуальность темы исследования

1.1 Актуальность разработки системы учета технического обслуживания и ремонта оборудования на предприятии

В современных условиях эффективное управление производственными процессами на предприятии невозможно без надежной системы учета технического обслуживания и ремонта оборудования (ТОиР). С каждым годом растет зависимость бизнеса от высокотехнологичного оборудования, и его бесперебойная работа становится ключевым фактором успешности и конкурентоспособности предприятия. Актуальность разработки системы учета ТОиР обусловлена несколькими важными аспектами.

Простой оборудования из-за поломок или непредвиденного выхода из строя может привести к значительным финансовым потерям для предприятия. Эффективная система учета ТОиР позволяет вовремя отслеживать состояние оборудования, планировать профилактические ремонты и своевременно устранять неполадки.

Система учета ТОиР помогает предприятиям рационально распределять ресурсы на ремонтные работы. За счет планирования и прогнозирования необходимых мероприятий по ТОиР можно избежать излишних расходов, связанных с аварийными ремонтами. Кроме того, такая система позволяет контролировать использование запасных частей и материалов, предотвращая их перерасход или нехватку.

Регулярное техническое обслуживание, своевременный ремонт и правильная эксплуатация оборудования существенно увеличивают его срок службы. Система учета ТОиР фиксирует все операции по обслуживанию и ремонту, что позволяет отслеживать историю работы каждого агрегата, выявлять наиболее проблемные узлы и принимать меры по их модернизации или замене.

Изношенное или неисправное оборудование представляет серьезную угрозу для безопасности сотрудников. Система учета ТОиР способствует поддержанию оборудования в надлежащем состоянии, что снижает вероятность аварий и инцидентов на производстве. Таким образом, предприятие не только выполняет требования по охране труда, но и обеспечивает защиту жизни и здоровья своих сотрудников.

Во многих отраслях промышленности действуют строгие нормативные требования к техническому состоянию оборудования. Наличие системы учета ТОиР позволяет предприятию вести документированный учет всех операций по обслуживанию и ремонту, что является необходимым условием для прохождения проверок и аудитов.

Внедрение системы учета ТОиР особенно актуально для предприятий, где используется сложное и дорогостоящее оборудование. Это касается как крупных производственных компаний, так и средних и малых предприятий. В условиях конкуренции и постоянно растущих требований к качеству продукции и скорости ее выпуска, система учета ТОиР становится важным инструментом для достижения производственной стабильности и экономической эффективности.

В настоящее время на многих предприятиях, особенно на тех, где внедрение цифровых технологий еще не достигло высокого уровня, система учета технического обслуживания и ремонта (ТОиР) оборудования осуществляется вручную с использованием бумажных носителей. Этот метод, несмотря на свою простоту и доступность, имеет ряд особенностей и ограничений, которые влияют на эффективность управления техническим обслуживанием и ремонтом.

Вручную система учета ТОиР начинается с заведением специальных журналов и карт, где фиксируются все операции по обслуживанию и ремонту оборудования. В каждом цехе или отделе может вестись отдельный журнал, в котором для каждого агрегата заводится своя карточка. Эти карточки или

страницы журнала заполняются ответственными лицами, чаще всего мастерами, механиками или другими сотрудниками, назначенными для этой задачи. Информация вносится вручную, что требует особого внимания к деталям и аккуратности, так как ошибки и неточности могут привести к искажению данных.

Процесс заполнения журналов включает в себя указание таких данных, как наименование оборудования, его инвентарный номер, дата проведения обслуживания или ремонта, вид выполненных работ, перечень замененных деталей и узлов, а также фамилия и подпись исполнителя. В некоторых случаях в журнале также указываются причины поломки и рекомендации по дальнейшей эксплуатации. Если на предприятии существует регламент по периодичности ТОиР, то в журнале отмечаются даты плановых осмотров и проверок.

Заполненные журналы и карты хранятся в специально отведенных местах – это могут быть архивы, шкафы или полки в кабинете ответственного сотрудника. Важно обеспечить надежное хранение этих документов, так как они могут понадобиться при проведении аудитов, проверок или при возникновении спорных ситуаций. В связи с тем, что бумажные носители подвержены риску утраты или повреждения (например, из-за пожара, затопления или просто небрежного обращения), предприятиям приходится уделять внимание созданию резервных копий, что также осуществляется вручную.

Одним из наиболее сложных и трудоемких этапов работы с бумажными носителями является поиск необходимой информации. Например, для анализа состояния конкретного оборудования или планирования ремонта сотруднику может потребоваться просмотреть несколько журналов и карт, сверяя данные между собой. Это занимает много времени и увеличивает вероятность ошибок, особенно на крупных предприятиях с большим количеством техники.

Ведение учета ТОиР на бумажных носителях сопряжено с рядом проблем. Во-первых, это трудоемкость и медлительность процесса, что может замедлить принятие решений, связанных с ремонтом или обслуживанием оборудования. Во-вторых, это высокая вероятность ошибок при заполнении документов вручную – неправильные записи, пропуски и т.д. В-третьих, бумажные журналы и карты сложно интегрировать с другими системами управления предприятием, что ограничивает возможности автоматизации и анализа данных.

Хотя ручное ведение учета ТОиР все еще используется, оно постепенно уступает место автоматизированным системам, которые позволяют существенно упростить и ускорить процесс, уменьшить количество ошибок и повысить общую эффективность управления техническим обслуживанием и ремонтом оборудования на предприятии.

Таким образом, несмотря на то, что ручное ведение учета ТОиР на бумажных носителях до сих пор распространено, оно все более явно демонстрирует свои ограничения, подталкивая предприятия к переходу на современные цифровые решения.

1.2 Существующие системы учета технического обслуживания и ремонта оборудования на предприятии

Существующие системы учета технического обслуживания и ремонта (ТОиР) оборудования на предприятии делятся на два основных типа: ручные (традиционные) и автоматизированные (цифровые). Каждая из этих систем имеет свои особенности, преимущества и ограничения.

1. Ручные системы учета ТОиР

Бумажные журналы и карты. Это наиболее традиционный способ ведения учета, где все данные фиксируются вручную на бумажных носителях. В каждой производственной единице или цехе ведутся специальные журналы учета, где регистрируются все операции по техническому обслуживанию и ремонту оборудования. Для каждого агрегата могут заводиться отдельные карточки, в которых указываются даты проведенных работ, типы выполненных ремонтов, замененные детали, а также фамилии и подписи исполнителей. Этот метод отличается простотой и доступностью, но требует значительных затрат времени на заполнение и обработку данных, что повышает вероятность ошибок и затрудняет поиск нужной информации.

Экземплярные журналы и ведомости. В некоторых случаях вместо одного журнала для всего оборудования могут вестись отдельные ведомости для каждого типа работы или группы оборудования. Это позволяет более детально фиксировать информацию, но усложняет управление, так как данные оказываются разбросанными по различным документам.

2. Автоматизированные системы учета ТОиР

С развитием информационных технологий предприятия все чаще переходят на автоматизированные системы учета ТОиР, которые позволяют значительно упростить управление процессами обслуживания и ремонта оборудования.

CMMS (Computerized Maintenance Management System). Это специализированные программные системы, предназначенные для управления техническим обслуживанием и ремонтом оборудования. CMMS-системы позволяют автоматизировать весь процесс учета, начиная от планирования и регистрации работ и заканчивая анализом данных и отчетностью. Такие системы включают в себя модули для ведения истории обслуживания, учета запчастей, планирования ремонтов, управления заявками на ремонт и контроля за выполнением работ. CMMS-системы интегрируются с другими системами управления предприятием (ERP, SCADA и др.), что позволяет объединить учет ТОиР с управлением производственными процессами и ресурсами.

ERP-системы (Enterprise Resource Planning). В крупных предприятиях учет ТОиР часто интегрирован в более широкие ERP-системы, которые охватывают все аспекты управления предприятием, включая финансы, производство, логистику и персонал. В таких системах модуль ТОиР является частью общего комплекса управления активами (EAM) и обеспечивает не только учет и планирование ТОиР, но и управление затратами, анализ эффективности использования оборудования и прогнозирование потребностей в ремонтах и обслуживании.

IoT и умные датчики. Современные предприятия все чаще используют технологии Интернета вещей (IoT) для улучшения учета и управления ТОиР. Умные датчики, установленные на оборудовании, собирают данные в режиме реального времени, передают их в систему управления, где они анализируются для предсказания возможных отказов и планирования профилактического обслуживания. Такие системы позволяют перейти от реактивного к предиктивному обслуживанию, что значительно снижает риск поломок и увеличивает срок службы оборудования.

Мобильные приложения. В некоторых автоматизированных системах учета ТОиР используются мобильные приложения, которые позволяют

техническому персоналу оперативно вносить данные о проведенных работах, получать задания на ремонт и обслуживание, а также фиксировать проблемы непосредственно на месте. Это ускоряет процесс обмена информацией и снижает риск потери данных.

Преимущества автоматизированных систем

Автоматизированные системы учета ТОиР имеют ряд преимуществ по сравнению с традиционными методами. Во-первых, они позволяют значительно снизить количество ошибок, возникающих при ручном вводе данных. Во-вторых, такие системы обеспечивают быстрый доступ к необходимой информации, что упрощает анализ и принятие решений. В-третьих, автоматизация учетных процессов позволяет более эффективно управлять ресурсами и снижать затраты на обслуживание и ремонт оборудования.

Выбор системы учета

Выбор системы учета ТОиР зависит от размера предприятия, сложности используемого оборудования и уровня цифровизации процессов. Малые и средние предприятия могут начать с простых CMMS-систем, которые легко адаптируются под их потребности, тогда как крупные корпорации обычно интегрируют учет ТОиР в общую ERP-систему, обеспечивая полную синхронизацию всех бизнес-процессов.

В целом, современные системы учета ТОиР играют ключевую роль в обеспечении бесперебойной работы оборудования, увеличении его срока службы и оптимизации производственных процессов на предприятии.

По результатам проделанной работы в данной главе, можно сделать следующие выводы:

«Внедрение автоматизированной системы учета технического обслуживания и ремонта (ТОиР) оборудования на предприятии приносит значительные преимущества, которые способствуют оптимизации работы, снижению издержек и увеличению срока службы оборудования» [18].

Рассмотрим ключевые выводы, которые можно сделать на основе использования такой системы.

АСУ ТОиР – это автоматизированная система управления техобслуживанием и ремонтами оборудования на предприятии, которая позволяет существенно повысить надежность оборудования за счет своевременного выполнения профилактических работ. Благодаря автоматизации процесса планирования и контроля, предприятия могут избегать непредвиденных поломок и простоев.

Регулярное обслуживание, как показывает практика, значительно продлевает срок службы оборудования, минимизируя необходимость в дорогостоящих аварийных ремонтах и обеспечивая стабильность производственных процессов.

Автоматизированные системы учета ТОиР позволяют предприятиям более эффективно управлять ресурсами, выделяемыми на обслуживание и ремонт. Благодаря автоматическому учету запасных частей и материалов, предприятия могут оптимизировать свои запасы и гарантировать наличие необходимых компонентов.

Автоматизированная система учета ТОиР предоставляет мощные инструменты для контроля и анализа процессов технического обслуживания и ремонта. Система позволяет отслеживать выполнение планов, управлять заявками на ремонт и анализировать эффективность выполненных работ. Автоматизированная система также помогает соблюдать нормативные требования и стандарты, обеспечивая документированное подтверждение выполнения всех необходимых операций.

Одним из важных факторов внедрения автоматизированной системы учета ТОиР является возможность интеграции с другими системами управления предприятием, такими как ERP или SCADA. Такая интеграция создает единое информационное пространство, в котором данные о состоянии

оборудования, ресурсах и производственных процессах тесно связаны между собой.

Автоматизированная система учета ТОиР помогает преодолеть ограничения традиционных методов, таких как использование бумажных журналов и карточек. Осуществлять ремонтные работы и предотвращать неисправности, чтобы избежать снижения производительности.

Особенности и взаимосвязь всех элементов обслуживания Ручные методы, хотя и просты, не обеспечивают достаточного уровня точности и оперативности, что может приводить к ошибкам, потерям данных и затруднениям в управлении процессами. Автоматизация процессов снижает вероятность ошибок, ускоряет обработку данных и улучшает качество управления.

Автоматизированные системы учета ТОиР продолжают развиваться, внедряя современные технологии, такие как Интернет вещей (IoT) и машинное обучение, которые открывают новые возможности для предиктивного обслуживания и повышения эффективности работы оборудования. Эти технологии позволяют не только реагировать на поломки, но и прогнозировать их, делая управление ТОиР еще более проактивным и эффективным.

Внедрение автоматизированной системы учета ТОиР является необходимым моментом для любого предприятия, стремящегося к повышению эффективности и надежности своих производственных процессов. Такие системы обеспечивают более высокий уровень контроля, точности и оперативности, что приводит к снижению затрат, увеличению срока службы оборудования и повышению производственной надежности.

В условиях растущей конкуренции и повышающихся требований к качеству и скорости производства, использование автоматизированных систем учета ТОиР становится необходимым условием для достижения долгосрочного успеха и устойчивого развития предприятия.

2. Разработка системы учета технического обслуживания и ремонта оборудования на предприятии

2.1 Выбор и обоснование языка программирования

Выбор языка программирования для создания автоматизированной системы учета технического обслуживания и ремонта (ТОиР) оборудования на предприятии является решающим шагом, влияющим на эффективность, масштабируемость и удобство будущего программного продукта.

Для принятия обоснованного решения следует учитывать такие факторы, как производительность, безопасность, возможности интеграции, доступность библиотек и фреймворков, а также популярность и поддержка сообщества разработчиков. Рассмотрим основные языки программирования, подходящие для разработки подобной системы, и обоснуем их выбор.

Java – один из наиболее популярных и проверенных временем языков программирования, широко используемый для разработки корпоративных приложений. Java предлагает разработчикам удобный и надежный инструментарий для создания масштабируемых и переносимых программных решений.

Основные преимущества Java включают в себя высокую переносимость (код Java можно запускать на различных платформах без изменений), надежность, масштабируемость и богатую экосистему. Java предоставляет обширный набор библиотек и фреймворков, таких как Spring и Hibernate, которые упрощают разработку сложных и масштабируемых систем.

В контексте создания системы учета ТОиР на предприятии Java подходит благодаря способности обрабатывать большие объемы данных и поддерживать многопользовательские режимы работы. Кроме того, Java интегрируется с другими корпоративными системами, такими как ERP (Enterprise Resource Planning) и SCM (Supply Chain Management), что позволяет обеспечить единое информационное пространство на предприятии.

C# – это кроссплатформенный язык общего назначения, который делает разработчиков продуктивным при написании высокопроизводительного кода. С миллионами разработчиков C# является самым популярным языком.

Язык программирования, разработанный компанией Microsoft, подходит для создания корпоративных приложений, включая системы учета ТОиР. C# особенно удобен для разработки приложений под Windows, что делает его идеальным для предприятий, использующих эту операционную систему.

C# предоставляет мощные инструменты для разработки и богатую экосистему, включающую .NET Framework и .NET Core, которые поддерживают кроссплатформенность и позволяют создавать надежные и производительные приложения. C# также обладает хорошей поддержкой для интеграции с различными базами данных и другими корпоративными системами, что способствует построению эффективной системы учета ТОиР.

Python – еще один язык программирования, заслуживающий внимания при разработке системы учета ТОиР. Python известен своей простотой и высокой читаемостью кода, что ускоряет процесс разработки и облегчает последующее сопровождение программного обеспечения (рис.1).



Рисунок 1 – Язык программирования Python

Python предлагает обширную экосистему библиотек и фреймворков, таких как Django и Flask, которые позволяют быстро разрабатывать веб-приложения. Это делает Python хорошим выбором для создания интерфейсов системы учета ТОиР, особенно если планируется разработка веб-версии или интеграция с облачными сервисами. Python активно используется в области анализа данных и машинного обучения, что открывает возможности для внедрения предиктивного анализа в систему ТОиР.

JavaScript, особенно в связке с Node.js, является мощным инструментом для разработки веб-приложений, включая системы учета ТОиР. Node.js позволяет использовать JavaScript для серверной части приложения, что упрощает создание полноценных веб-сервисов. Это особенно полезно, если требуется разработка системы с веб-интерфейсом, доступной с различных устройств.

JavaScript хорошо поддерживается сообществом, что обеспечивает доступ к большому количеству библиотек и инструментов. Он также легко интегрируется с фронтенд-фреймворками, такими как React или Angular, что позволяет создать современный и удобный пользовательский интерфейс.

Выбор языка программирования для разработки системы учета ТОиР зависит от конкретных потребностей предприятия и планируемого окружения для эксплуатации системы. Java и C# подходят для создания масштабируемых и надежных корпоративных систем с глубокой интеграцией с существующими платформами. Python является отличным выбором для быстрой разработки и внедрения современных аналитических возможностей. JavaScript с Node.js идеально подходит для создания веб-ориентированных приложений, доступных на различных устройствах.

Оптимальный выбор языка программирования зависит от целей проекта, специфики ИТ-инфраструктуры предприятия и опыта команды разработчиков. Если важна высокая интеграция с существующими системами и стабильность под нагрузкой, Java или C# станут лучшими решениями. Если требуется быстрая разработка и расширенные аналитические возможности, стоит обратить внимание на Python. Для веб-ориентированных приложений с широким доступом идеально подойдет JavaScript с Node.js.

«Кроме того, важно учитывать долгосрочные перспективы проекта и планы по его развитию. Например, если в будущем система учета ТОиР должна будет расширяться и интегрироваться с новыми сервисами, то язык программирования должен поддерживать гибкость и масштабируемость. В этом контексте, такие языки, как Java и C#, особенно хороши для проектов, которые требуют долгосрочной поддержки и регулярных обновлений» [10].

Если предприятие планирует активно использовать технологии интернета вещей (IoT) для мониторинга состояния оборудования в реальном времени, выбор языка программирования также может зависеть от способности системы взаимодействовать с различными устройствами и

сенсорами. В таких случаях можно рассматривать Python или JavaScript, поскольку они обеспечивают отличные инструменты для разработки интеграций с IoT и анализа данных, поступающих от оборудования.

Python особенно полезен для создания систем, которые предполагают работу с большими объёмами данных или требуют реализации машинного обучения для предсказания возможных отказов оборудования. Python имеет отличную библиотечную поддержку в этих областях (например, библиотеки Scikit-learn, TensorFlow, Pandas), что может существенно сократить время на разработку модулей прогнозирования и анализа.

С другой стороны, если планируется создание распределённой системы с использованием микросервисной архитектуры, то JavaScript с Node.js может стать оптимальным выбором благодаря лёгкости его внедрения и возможности разворачивания микросервисов на различных платформах. Это особенно актуально для веб-систем, которые должны работать круглосуточно и поддерживать большое количество пользователей.

Также следует учитывать квалификацию команды разработчиков. Если в штате компании больше специалистов, работающих с Java или C#, имеет смысл выбирать эти языки, так как они уже известны команде и позволят сократить время на обучение и адаптацию. Если же предприятие использует современные подходы к разработке, такие как DevOps или Agile, и предпочитает быстрое прототипирование, то Python или JavaScript могут оказаться более подходящими благодаря своей простоте и скорости разработки.

Для успешного выбора языка программирования также стоит учитывать требования к поддерживаемым платформам и архитектуре системы. Например, если система учета ТОиР должна работать на разных операционных системах, таких как Windows, Linux или macOS, то Java может оказаться наилучшим выбором благодаря своей кроссплатформенности. Приложения на Java работают в среде Java Virtual Machine (JVM), что

позволяет запускать их на любой системе, где установлена эта среда, минимизируя проблемы совместимости.

Также необходимо учитывать требования к интерфейсу пользователя. Если система должна обеспечивать интерактивный и интуитивно понятный интерфейс, то использование JavaScript в сочетании с современными фронтед-фреймворками, такими как React или Angular, может значительно улучшить пользовательский опыт. Это позволит создать динамичные веб-приложения, которые легко адаптируются под различные устройства и разрешения экранов, что особенно важно для мобильных пользователей.

Кроме того, следует обратить внимание на сообщества и доступные ресурсы для выбранного языка программирования. Языки с большим сообществом и обширной документацией, такие как Python и JavaScript, облегчают процесс разработки благодаря доступу к множеству библиотек, инструментов и готовых решений. Это может существенно ускорить процесс разработки и сократить время на решение типичных задач.

Также важно предусмотреть возможные риски, связанные с выбором языка. Например, если предприятие выберет язык с низким уровнем популярности, это может создать сложности в будущем с поддержкой и наймом новых разработчиков. Поэтому стоит ориентироваться на языки, которые активно используются в промышленности и имеют долгую историю развития, такие как Java, C# и Python.

Кроме того, не стоит забывать о вопросах безопасности, особенно если система будет обрабатывать конфиденциальные данные или будет доступна из интернета. Языки программирования имеют разные механизмы защиты, и важно выбрать тот, который обеспечивает высокий уровень безопасности и позволяет реализовать необходимые меры по защите данных.

Итак, выбор языка программирования для разработки автоматизированной системы учета ТОиР требует всестороннего анализа потребностей предприятия, квалификации команды, долгосрочных планов по

развитию и архитектуры системы. Учитывая все вышеперечисленные факторы, можно сделать обоснованный выбор, который позволит создать эффективное и надежное решение, способствующее повышению производительности и оптимизации процессов учета технического обслуживания и ремонта оборудования. Такой подход не только повысит качество и скорость разработки, но и обеспечит долгосрочную устойчивость и развитие системы в условиях меняющегося рынка.

«Далее, следует рассмотреть, как выбор языка программирования влияет на дальнейшее сопровождение и развитие системы. Постоянное обновление и улучшение функциональности – важные аспекты успешной работы автоматизированной системы учета ТОиР. Выбранный язык программирования должен обладать достаточной гибкостью и поддержкой, чтобы без особых затруднений вносить изменения и добавлять новые функции»[12].

Например, использование Python для разработки может значительно упростить процесс внесения изменений благодаря его синтаксической простоте и широкому выбору библиотек. Это делает язык подходящим для сценариев, требующих быстрой адаптации к новым требованиям бизнеса или изменениям в законодательстве, связанным с безопасностью и качеством обслуживания оборудования.

Кроме того, важно учитывать не только возможности самого языка, но и доступные инструменты и фреймворки, которые могут облегчить разработку. Например, использование фреймворков для разработки веб-приложений на JavaScript, таких как Express.js для серверной части или React.js для клиентской, позволяет быстро создавать масштабируемые и высокопроизводительные приложения. Это особенно актуально для систем, которые будут активно использоваться различными пользователями через веб-интерфейс.

В дополнение к этому, можно рассмотреть вопрос о совместимости с различными технологиями, которые могут быть использованы на предприятии. Например, если у предприятия уже есть решения на C# и .NET, имеет смысл продолжить использовать эту технологию, чтобы избежать дополнительных сложностей с интеграцией. Это также позволит легче обучать новых сотрудников и поддерживать существующие системы.

Также следует не забывать о важности тестирования и обеспечения качества разрабатываемого ПО. Некоторые языки программирования, такие как Java, имеют встроенные инструменты для автоматизированного тестирования, что позволяет разработчикам создавать надежные системы с минимальными ошибками. Использование таких инструментов значительно сокращает время на тестирование и улучшает качество кода, что особенно важно для систем, ответственных за учет и контроль технического обслуживания.

Наконец, стоит упомянуть о возможности обучения и развития команды разработчиков. Инвестирование в обучение сотрудников новому языку или технологии может принести значительные долгосрочные преимущества для предприятия. Поэтому важно оценивать не только текущие знания команды, но и возможности их дальнейшего развития в выбранной технологии.

Параллельно с вопросом обучения важно учитывать доступность документации и ресурсов, связанных с выбранным языком программирования. Хорошая документация и активное сообщество разработчиков могут значительно упростить процесс решения возникающих проблем и ускорить разработку новых функций. Например, языки программирования с обширными сообществами, такие как Python и JavaScript, предлагают множество ресурсов, включая онлайн-курсы, форумы и библиотеку открытого кода, что может быть полезно для команд, работающих над проектами.

В условиях стремительно меняющегося технологического ландшафта предприятия также должны быть готовы к интеграции новых технологий в

существующие системы. Выбор языка программирования должен учитывать возможность бесшовной интеграции с другими технологиями, такими как облачные сервисы, IoT-устройства и системы анализа данных. Например, язык Python поддерживает множество библиотек для работы с API, что упрощает интеграцию с внешними сервисами и устройствами.

Кроме того, языки, такие как Java и C#, предлагают богатые возможности для работы с большими данными и могут легко интегрироваться с платформами, поддерживающими машинное обучение и аналитические инструменты. Это становится особенно актуальным для систем учета ТОиР, которые могут извлекать полезные инсайты из больших объемов данных о состоянии оборудования.

«Не менее важным аспектом является обеспечение безопасности системы. Выбранный язык программирования должен поддерживать современный уровень безопасности, включая механизмы защиты от распространенных уязвимостей, таких как SQL-инъекции, межсайтовый скриптинг (XSS) и другие угрозы. Языки, такие как C# и Java, предоставляют встроенные средства для реализации безопасных практик разработки, что позволяет создавать более защищенные приложения»[11].

Системы учета ТОиР могут обрабатывать чувствительные данные, такие как информация о сотрудниках и технические характеристики оборудования, что требует строгого соблюдения стандартов безопасности и конфиденциальности. Поэтому важно выбирать язык, который позволяет легко реализовывать требования к безопасности и соответствия.

Таким образом, выбор языка программирования для системы учета технического обслуживания и ремонта оборудования имеет множество последствий для дальнейшего сопровождения и развития системы. Важными аспектами являются гибкость языка, доступность инструментов и фреймворков, возможности тестирования и качества кода, а также обучение и

развитие команды разработчиков. Каждый из этих факторов вносит свой вклад в успешность проекта и его устойчивость к изменениям в требованиях бизнеса.

Качественно выбранный язык программирования станет основой для создания эффективного и надежного решения, способствующего оптимизации процессов учета ТОиР. Важно подходить к этому выбору со всей серьезностью, анализируя не только текущие потребности, но и будущее развитие системы, что позволит предприятию не только адаптироваться к изменениям, но и опережать конкурентов на рынке.

2.2 Выбор и обоснование базы данных

В рамках разработки автоматизированной системы учета технического обслуживания и ремонта (ТОиР) оборудования на предприятии важным этапом является выбор подходящей базы данных. Правильный выбор базы данных напрямую влияет на производительность, безопасность, масштабируемость и возможность интеграции системы с другими приложениями. Рассмотрим ключевые факторы, которые следует учитывать при выборе базы данных.

Факторы выбора базы данных:

- тип данных и их объем. В системах учета ТОиР предполагается работа с различными типами данных: информация о оборудовании, планы и отчеты по техническому обслуживанию, результаты диагностики, запчасти и многое другое. Объем данных может варьироваться в зависимости от масштаба предприятия и количества оборудования. Для систем, работающих с большим объемом структурированных данных, реляционные базы данных (РСУБД) будут предпочтительнее, так как они обеспечивают надежное хранение и высокую степень организованности данных;

- скорость обработки данных. Важно, чтобы выбранная база данных обеспечивала высокую производительность при выполнении операций.

«Системы учета ТОиР должны обрабатывать запросы на получение информации о состоянии оборудования, планах обслуживания и истории ремонтов в режиме реального времени. Это особенно актуально для предприятий, где требуется быстрое реагирование на запросы пользователей» [15]. В этом контексте следует рассмотреть базы данных, которые поддерживают эффективные индексации и кэширование;

- масштабируемость. Система должна быть готова к расширению в будущем. Это означает, что база данных должна поддерживать возможность добавления новых данных и пользователей без значительного ухудшения производительности. Реляционные базы данных, такие как PostgreSQL или MySQL, хорошо справляются с задачами масштабирования, а также предлагают возможности горизонтального и вертикального масштабирования;

- безопасность. Поскольку система учета ТОиР будет обрабатывать чувствительную информацию, важно, чтобы база данных обеспечивала высокий уровень безопасности. «Это включает в себя шифрование данных, а также возможность настройки ролей и прав доступа» [8]. Базы данных, такие как Microsoft SQL Server и Oracle Database, предоставляют множество встроенных функций для обеспечения безопасности и защиты данных;

- совместимость и интеграция. При выборе базы данных также необходимо учитывать возможность интеграции с другими системами, используемыми на предприятии. Важно, чтобы база данных поддерживала распространенные протоколы и форматы обмена данными, такие как RESTful API или ODBC. Это обеспечит легкость интеграции с ERP-системами, системами управления производством и другими корпоративными приложениями.

На основе вышеизложенных факторов можно выделить несколько подходящих вариантов баз данных для системы учета ТОиР:

1. PostgreSQL.

Это мощная реляционная база данных с открытым исходным кодом, которая отлично подходит для работы с большими объемами структурированных данных. PostgreSQL предлагает высокую степень безопасности, гибкие возможности масштабирования и богатый набор функциональных возможностей, таких как поддержка JSON для работы с полуструктурированными данными. Она также хорошо подходит для аналитических задач и обеспечивает надежную производительность при сложных запросах.

2. MySQL.

Эта популярная реляционная база данных является отличным выбором для проектов, где важна скорость разработки и простота использования. MySQL обеспечивает хорошую производительность, имеет широкое сообщество и много доступных ресурсов. Она идеально подходит для веб-приложений и небольших до средних проектов, требующих эффективного управления данными.

3. Microsoft SQL Server.

Это решение подойдет предприятиям, которые уже используют экосистему Microsoft. SQL Server обеспечивает высокую производительность, продвинутые функции безопасности и возможности интеграции с другими продуктами Microsoft. Это делает его особенно привлекательным для компаний, которые ориентированы на платформу Windows.

4. MongoDB.

Если требуется работа с неструктурированными данными или значительное разнообразие типов данных, стоит рассмотреть NoSQL решения, такие как MongoDB. Эта документно-ориентированная база данных позволяет легко хранить и обрабатывать данные, что может быть полезно в ситуациях, когда структура данных может изменяться. MongoDB также обеспечивает хорошую производительность и масштабируемость.

Выбор базы данных для автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии является ключевым этапом разработки, который влияет на эффективность работы системы в целом.

При принятии решения необходимо учитывать такие факторы, как объем и тип данных, скорость обработки, масштабируемость, безопасность и возможность интеграции. На основе анализа можно выбрать оптимальное решение, которое обеспечит надежную и эффективную работу системы, способствуя повышению производительности и оптимизации процессов учета ТОиР.

2.3 Разработка структурной схемы системы учета технического обслуживания и ремонта оборудования на предприятии

На рисунке (2) представлена структура разрабатываемого программного обеспечения для автоматизированной системы, предназначенной для учета технического обслуживания и ремонта оборудования на предприятии.

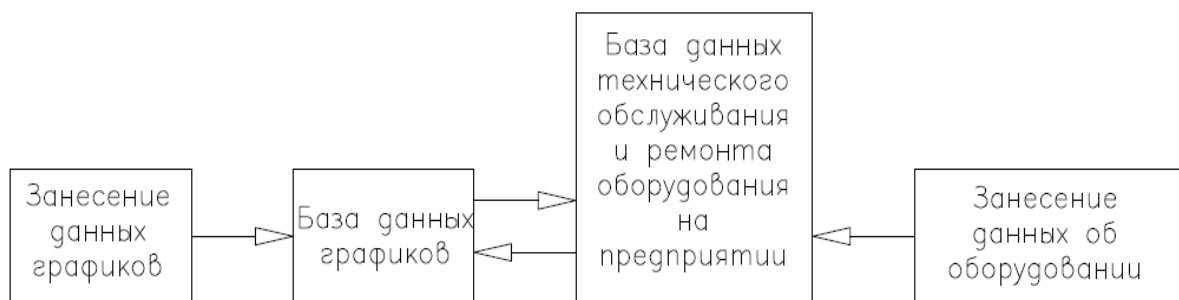


Рисунок 2 – Структура разрабатываемого программного обеспечения автоматизированной система учета технического обслуживания и ремонта оборудования на предприятии

В автоматизированной системе учета технического обслуживания и ремонта оборудования на предприятии в базу данных вносится информация о графике ремонта и технического обслуживания, включая даты начала и окончания. Каждому графику присваивается индивидуальный идентификатор, который обеспечивает возможность мониторинга технического ремонта и обслуживания оборудования. В информационной системе хранятся данные о каждом оборудовании на предприятии.

Эти записи содержат:

- идентификатор оборудования;
- название оборудования;
- дата проведения технического обслуживания;
- дата ремонта;
- полные ФИО и должность лица, отвечающего за ремонт или обслуживание.

«Техническое обслуживание и ремонт оборудования на предприятии осуществляется строго в соответствии с утвержденным графиком. Если оборудование не занесено в графике, система автоматически выдает предупреждение и блокирует его. Таким образом, система предотвращает не плановое техническое обслуживание и ремонт оборудования на предприятии» [19].

Алгоритм функционирования системы можно визуализировать с помощью блок-схемы (рис.3). Блок-схема является специализированным инструментом, который применяется для разработки новых систем или анализа и оптимизации уже существующих. Она дает общее представление о ключевых элементах системы, основных участниках и их взаимодействиях. Блок-схема позволяет быстро оценить систему и выявить проблемные области, но может не содержать всех деталей, необходимых для полного планирования или реализации проекта. На ней не отображаются все элементы

кода программы, что делает ее компактным, но менее детализированным инструментом.

«Главное предназначение блок-схемы – демонстрация потоков информации, а не описание всех процессов, происходящих между вводом и выводом данных. Это позволяет сосредоточиться на ключевых аспектах системы, используя так называемый принцип «черного ящика»» [10].

Использование такого подхода облегчает понимание общих принципов работы системы даже для пользователей или руководителей, которые не являются специалистами в области программирования. Блок-схема помогает визуализировать структуру системы, ключевые взаимосвязи между ее компонентами и порядок выполнения задач, что способствует лучшему восприятию архитектуры и позволяет эффективнее управлять процессами разработки, тестирования и эксплуатации программного обеспечения.

Кроме того, блок-схема является ценным инструментом на этапе планирования и обсуждения, так как позволяет быстро оценить потенциальные узкие места, улучшить взаимодействие модулей системы и выявить области, требующие дальнейшей оптимизации.

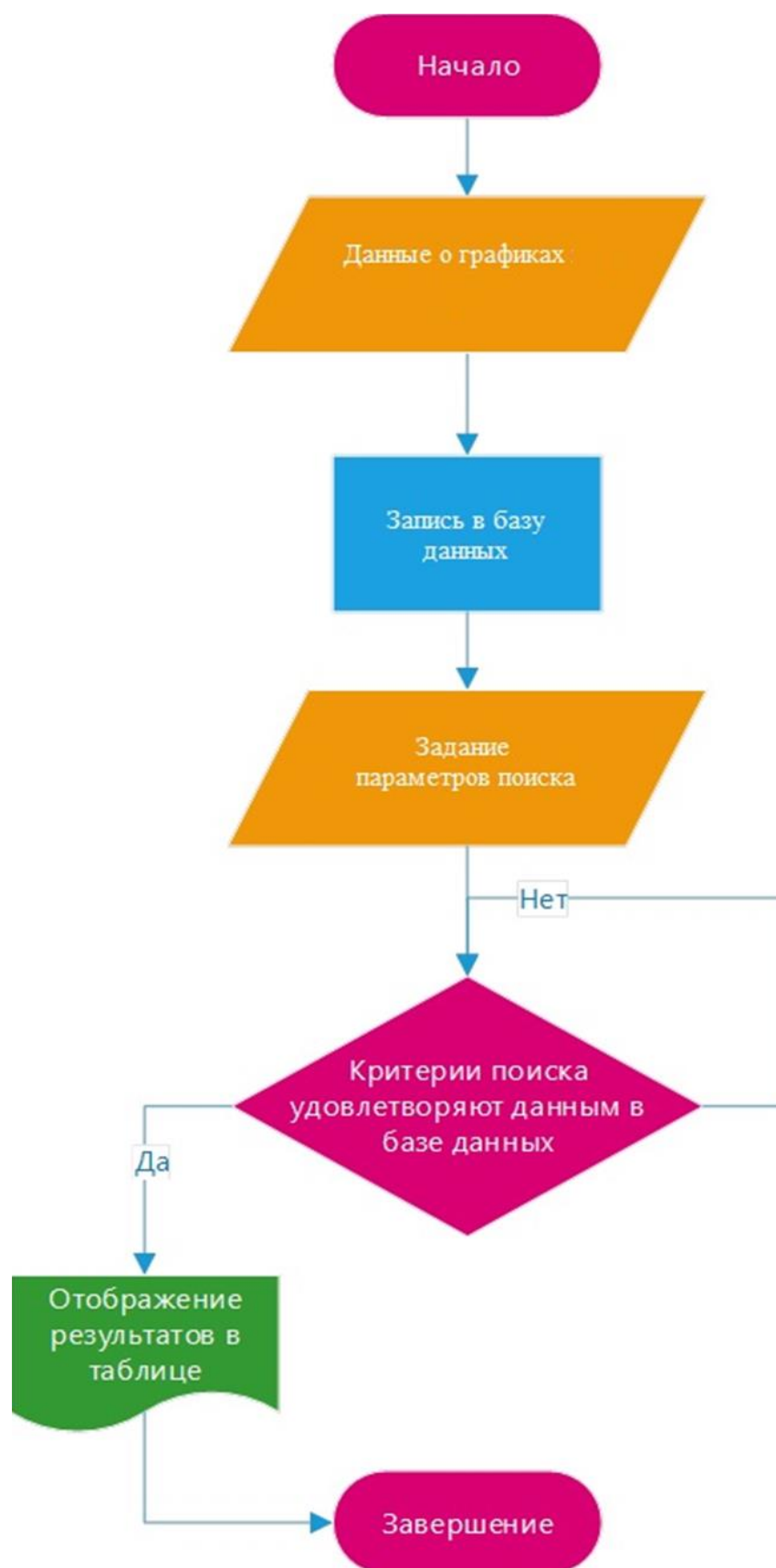


Рисунок 3 – Алгоритм ПО «АСУ»

Алгоритм работы программного обеспечения «АСУ» заключается в следующем:

Сначала нужно ввести информацию о графиках и оборудовании:

- номер оборудования;
- название оборудования;
- дата проведения технического обслуживания;
- дата ремонта;
- полные ФИО ответственного лица;
- должность ответственного лица.

После того как данные об оборудовании и графиках технического обслуживания и ремонта введены и сохранены в базе данных, система начинает процесс мониторинга и контроля за соблюдением сроков обслуживания:

1. Мониторинг графиков ТО и ремонта. Программное обеспечение автоматически сравнивает текущие даты с запланированными в системе датами технического обслуживания и ремонта. Если приближается срок выполнения плановых работ, система генерирует уведомление для ответственных лиц, напоминая о необходимости проведения ТО или ремонта.

2. Автоматическое уведомление. В случае, если график технического обслуживания или ремонта нарушен (например, срок уже наступил, но работы не были выполнены), система отправляет соответствующие оповещения пользователям с высоким приоритетом, предупреждая о потенциальных рисках из-за несвоевременного проведения работ.

3. Анализ данных и отчетность. Пользователи могут запрашивать отчеты на основании данных, содержащихся в базе. Отчеты могут содержать информацию о:

- текущем состоянии оборудования;
- графиках технического обслуживания и выполненных ремонтах;
- выявленных проблемах и неисправностях;

- эффективности работы оборудования после ремонта.

4. Обновление информации о состоянии оборудования. После проведения технического обслуживания или ремонта, ответственные лица вносят обновленные данные в систему. Информация о дате проведения работ, использованных запчастях, времени простоя оборудования и результатах обслуживания заносится в базу данных. Это помогает вести полную историю эксплуатации каждого устройства на предприятии.

5. Интеграция с другими системами. При необходимости система может обмениваться данными с другими информационными системами предприятия через интеграционные интерфейсы (API). Например, информация о состоянии оборудования может передаваться в ERP-систему для учета в планировании производственных процессов.

6. Анализ и прогнозирование. Программное обеспечение также может выполнять анализ накопленных данных, чтобы выявить повторяющиеся неисправности и предложить оптимальные графики обслуживания. Это позволяет повысить надежность работы оборудования и снизить количество внеплановых ремонтов.

Таким образом, алгоритм работы системы охватывает все этапы – от ввода данных до контроля за проведением работ, анализа и прогнозирования, что обеспечивает непрерывный процесс технического обслуживания и ремонта оборудования на предприятии.

2.4 Описание интерфейса программы

Интерфейс программного обеспечения «АСУ» был разработан в соответствии с техническим заданием. Он оформлен в максимально удобном и простом формате для комфортного использования программы.

Процесс разработки ПО проходит через несколько этапов, которые представлены на (рис.4). Все этапы подробно обсуждаются в главах данного

исследования. Теперь более подробно изучим этапы разработки программного обеспечения «АСУ».

«На этапе определения задачи необходимо подготовить техническое задание для разработки программы. Это задание для создания программного обеспечения системы, поддерживающей принятие решений по оптимизации численности сотрудников, представляет собой формализованный документ, в котором изложены ключевые технические требования к разрабатываемому ПО» [10].

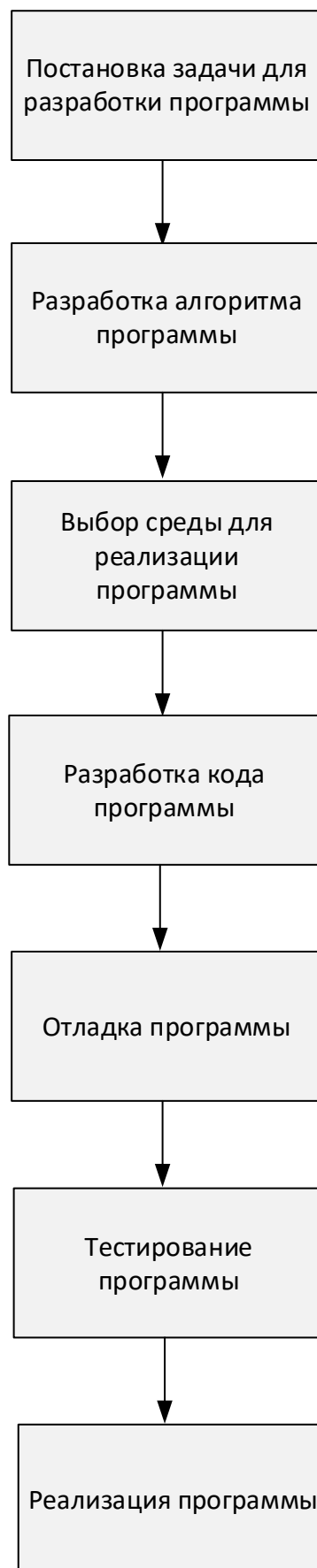


Рисунок 4 – Этапы разработки программы ПО «АСУ»

«Техническое задание является обязательным элементом при разработке любого программного обеспечения. Оно должно учитывать все нюансы и детали. Без тщательно подготовленного технического задания невозможно разработать полноценное ПО, способное эффективно решать поставленные задачи» [16].

Кроме того, техническое задание способствует предотвращению недоразумений между заказчиками и разработчиками, если какие-либо аспекты будут упущены в процессе работы. Подготовка технического задания осуществляется в соответствии с ГОСТ 34.602.

Структура технического задания должна содержать следующие разделы:

- введение;
- основания для разработки;
- назначение разработки;
- требования к программе или программному изделию;
- требования к программной документации;
- стадии и этапы разработки;
- порядок контроля и приемки;
- приложения.

Техническое задание на разработку ПО «АСУ».

Введение

Настоящее техническое задание посвящено созданию программного обеспечения «Автоматизированная система учета технического обслуживания и ремонта оборудования на предприятии» (ПО «АСУ»).

Основания для разработки.

Основанием для создания «Автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии» (ПО «АСУ») является задание на выполнение выпускной квалификационной работы.

Назначение разработки.

Разработка направлена на автоматизацию учета технического обслуживания и ремонта оборудования на предприятии.

Требования к программе или программному изделию.

Программное обеспечение должно выполнять следующие функции:

- учет технического обслуживания оборудования;
- учет ремонта оборудования.

Требования к программной документации.

«Создаваемое ПО должно быть самодокументируемым, включая все необходимые комментарии в коде. Система должна содержать справочную информацию о работе и предоставлять подсказки пользователям. В документацию также должно входить руководство для пользователей» [5].

Стадии и этапы разработки.

Процесс создания «Автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии» (ПО «АСУ») включает:

- разработку интерфейса программы;
- создание базы данных;
- реализацию алгоритма поиска заявок по различным критериям;
- тестирование программы;
- её отладку.

Порядок контроля и приемки.

Контроль и приемка осуществляются посредством тестирования программного обеспечения в производственных условиях.

Интерфейс автоматизированной системы учета технического обслуживания и ремонта оборудования (ПО «АСУ») оформлен в максимально удобном формате. В верхней части находятся поля для ввода данных (рис.5).

The screenshot displays the 'АСУ' (ASU) software interface. On the left, there are six input fields for data entry: 'Номер оборудования' (Equipment Number), 'Наименование оборудования' (Equipment Name), 'Дата технического обслуживания оборудования' (Equipment Maintenance Date), 'Дата ремонта оборудования' (Equipment Repair Date), 'Фамилия, имя, отчество ответственного лица' (Responsible Person's Surname, Name, Patronymic), and 'Должность ответственного лица' (Responsible Person's Position). On the right, there are six buttons for database management: 'Занести данные в таблицу' (Load data into table), 'Очистить базу данных' (Clear database), 'Открыть базу данных' (Open database), 'Найти в базе данных' (Find in database), 'Очистить таблицу' (Clear table), and 'Удалить значение из базы данных' (Delete value from database).

Рисунок 5 – Поля ввода данных

В поля ввода данных вносятся данные, которые попадают в базу данных:

- номер оборудования;
- наименование оборудования;
- дата технического обслуживания оборудования;
- дата ремонта оборудования;
- фамилия, имя, отчество ответственного лица;
- должность ответственного лица.

Далее, расположены кнопки управления базой данных (рис.6)

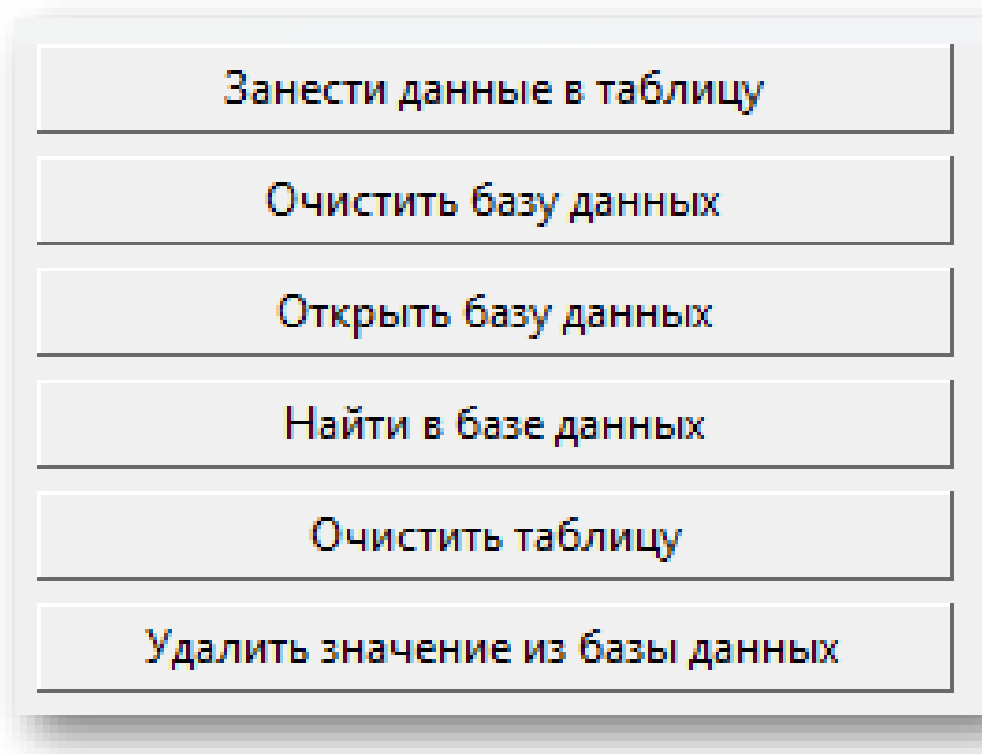


Рисунок 6 – Кнопки управления данными

С помощью этих кнопок можно выполнять следующие действия:

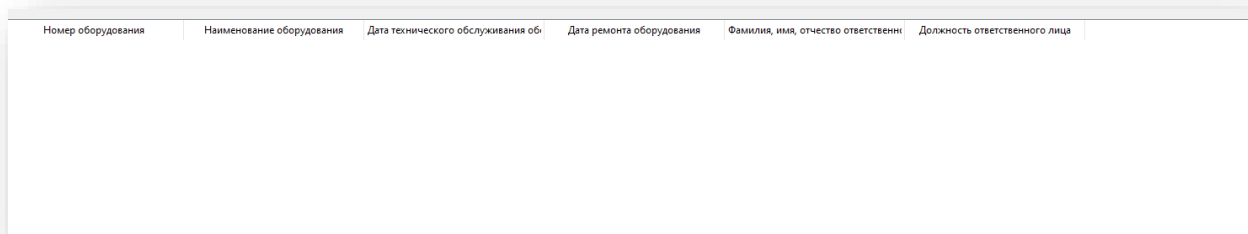
- вносить данные в таблицу;
- очищать базу данных;
- открывать базу данных;
- искать необходимое значение в базе данных;
- очищать таблицу в интерфейсе программы;
- удалять нужное значение из базы данных.

«После заполнения всех полей и нажатия кнопки «Добавить в таблицу», введенная информация отобразится в таблице ниже и автоматически сохранится в базе данных. База данных располагается в той же папке, что и программа, и создается автоматически при первом запуске» [4].

Для поиска нужного значения необходимо ввести его в поле и нажать «Поиск в базе данных». Найденное значение отобразится в таблице на вкладке

«Графики» (рис. 7). Перед поиском таблицу следует очистить с помощью кнопки «Очистить таблицу».

Кнопка «Очистить базу данных» удаляет все записи без возможности восстановления.



The screenshot shows a software interface for managing equipment data. At the top, there is a search bar with a magnifying glass icon and a button labeled 'Найти' (Find). Below the search bar is a table with the following columns: 'Номер оборудования' (Equipment Number), 'Наименование оборудования' (Equipment Name), 'Дата технического обслуживания об' (Technical Service Date), 'Дата ремонта оборудования' (Repair Date), 'Фамилия, имя, отчество ответственного' (Responsible Person's Name), and 'Должность ответственного лица' (Responsible Person's Position). The table is currently empty.

Номер оборудования	Наименование оборудования	Дата технического обслуживания об'	Дата ремонта оборудования	Фамилия, имя, отчество ответственного	Должность ответственного лица
--------------------	---------------------------	------------------------------------	---------------------------	---------------------------------------	-------------------------------

Рисунок 7 – Поиск в таблице по заданному значению

Отладка программы – это процесс выявления и исправления ошибок, а также оптимизации кода в программном обеспечении. Этот этап разработки является неотъемлемой частью жизненного цикла программного продукта и играет ключевую роль в обеспечении его надежности и функциональности.

Основная цель отладки – устранение дефектов, которые могут препятствовать нормальной работе программы. Ошибки могут проявляться в различных формах: от синтаксических ошибок, которые мешают компиляции, до логических ошибок, которые могут вызывать неправильное поведение программы.

Отладка ПО «АСУ» производилась через симуляцию всех операций, таких как:

- заполнение полей ввода;
- сохранение данных в базу;
- поиск информации в базе данных по установленным параметрам;
- очистка таблиц;
- полное удаление данных из базы.

В результате отладки программы, сбоев в работе системы выявлено не было. Это подтверждает, что код и алгоритмы программы функционируют корректно.

Процесс тестирования программных решений направлен на оценку их способности выполнять все предусмотренные задачи. Существует множество разновидностей тестирования, каждая из которых ориентирована на достижение конкретных целей и применяет различные подходы:

- приемочное тестирование его цель – убедиться, что система работает в соответствии с требованиями и ожиданиями пользователей. Это финальная проверка перед запуском продукта в эксплуатацию;

- интеграционное тестирование направлено на проверку взаимодействия между различными модулями или компонентами программы. Оно помогает выявить проблемы, возникающие при взаимодействии частей системы;

- модульное тестирование сосредоточено на проверке отдельного программного модуля. Этот тип тестирования позволяет изолировать и тестировать каждый компонент, что упрощает процесс отладки;

- функциональное тестирование проверяет, соответствует ли программа заявленным функциональным требованиям. Используются сценарии, основанные на реальных условиях эксплуатации, чтобы удостовериться, что все функции работают как задумано;

- проверка производительности: тестирование работы программного обеспечения при разных уровнях нагрузки. Например, нагрузочное тестирование применяется для оценки эффективности системы в условиях реальной эксплуатации;

- стресс-тестирование анализирует, как система работает при нагрузках, превышающих обычные эксплуатационные нормы, чтобы выявить ее устойчивость и способность восстанавливаться после критических сбоев;

- регрессионное тестирование проводится после внесения изменений в код, чтобы убедиться, что новые исправления или функции не повлияли на существующую функциональность.

Для тестирования ПО «АСУ» были выполнены следующие типы тестов:

- тестирование на соответствие требованиям;
- модульное тестирование отдельных компонентов;
- проверка функциональности системы;
- оценка производительности.

В процессе тестирования программа показала отличные результаты, и нарушений в работе выявлено не было.

Во второй главе была рассмотрена комплексная разработка системы учета технического обслуживания и ремонта оборудования на предприятии.

Для реализации системы был выбран язык программирования, который обеспечивает высокую производительность и удобство работы. Обоснование выбора основывалось на анализе потребностей предприятия, включая требования к функциональности, надежности и масштабируемости системы.

Выбор базы данных также был критически важен для обеспечения надежного хранения и быстрого доступа к информации. База данных была выбрана с учетом ее функциональных возможностей, поддержки транзакций и эффективной работы с большим объемом данных.

«Структурная схема системы была разработана с акцентом на модульность и простоту взаимодействия между компонентами. Это позволит легко добавлять новые функции и адаптировать систему под меняющиеся требования предприятия» [14].

Описание интерфейса программы подчеркивает важность пользовательского опыта. Интерфейс был разработан с учетом удобства и интуитивной понятности, что обеспечит высокую степень удовлетворенности пользователей и минимизацию времени на обучение.

Таким образом, разработка системы учета технического обслуживания и ремонта оборудования на предприятии представляет собой многоэтапный процесс, который включает выбор технологий, создание структурной схемы и разработку пользовательского интерфейса.

Эти аспекты в совокупности обеспечивают эффективность системы, ее устойчивость к изменениям и удобство в использовании, что в конечном итоге способствует повышению эффективности работы предприятия.

3. Расчет экономической эффективности

3.1 Расчет затрат на разработку автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии

Расчет затрат на разработку автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии.

1. Затраты на проектирование и анализ требований.

Сбор и анализ требований: $80 \text{ часов} \cdot 3,500 \text{ руб/час} = 280,000 \text{ руб.}$

Создание технического задания: $40 \text{ часов} \cdot 3,500 \text{ руб/час} = 140,000 \text{ руб.}$

Итого: 420,000 руб.

2. Затраты на разработку программного обеспечения.

Выбор и обоснование языка программирования:

$20 \text{ часов} \cdot 3,500 \text{ руб/час} = 70,000 \text{ руб.}$

Разработка серверной части: $200 \text{ часов} \cdot 3,500 \text{ руб/час} = 700,000 \text{ руб.}$

Разработка клиентской части (интерфейса):

$150 \text{ часов} \cdot 3,500 \text{ руб/час} = 525,000 \text{ руб.}$

Интеграция с базой данных: $100 \text{ часов} \cdot 3,500 \text{ руб/час} = 350,000 \text{ руб.}$

Итого: 1,645,000 руб.

3. Затраты на базу данных.

Лицензия на СУБД: 150,000 руб. (лицензия на 1 год).

Установка и настройка базы данных: $30 \text{ часов} \cdot 3,500 \text{ руб/час} = 105,000 \text{ руб.}$

Итого: 255,000 руб.

4. Затраты на тестирование.

Проведение функционального тестирования.

$100 \text{ часов} \cdot 3,500 \text{ руб/час} = 350,000 \text{ руб.}$

Исправление ошибок и доработка: $80 \text{ часов} \cdot 3,500 \text{ руб/час} = 280,000 \text{ руб.}$

Итого: 630,000 руб.

5. Затраты на внедрение и обучение.

Внедрение системы на предприятии: $40 \text{ часов} \cdot 3,500 \text{ руб/час} = 140,000 \text{ руб.}$

Обучение сотрудников: $30 \text{ часов} \cdot 3,500 \text{ руб/час} = 105,000 \text{ руб.}$

Итого: 245,000 руб.

6. Затраты на техническую поддержку (первый год).

Поддержка и обновления: 10% от общей стоимости разработки = 387,000 руб.

Общая стоимость разработки.

Сумма всех затрат:

Проектирование и анализ требований: 420,000 руб.

Разработка ПО: 1,645,000 руб.

База данных: 255,000 руб.

Тестирование: 630,000 руб.

Внедрение и обучение: 245,000 руб.

Техническая поддержка: 387,000 руб.

Общая сумма: 3,582,000 руб.

Расчет оплаты труда программистов может варьироваться в зависимости от их квалификации, опыта и объема работы. Ниже приведен расчет, исходя из различных ролей и часов работы программистов на разработку автоматизированной системы учета технического обслуживания и ремонта оборудования.

Ставки зарплаты программистов.

Младший программист (Junior): 2,500 руб/час.

Программист (Middle): 3,500 руб/час.

Старший программист (Senior): 4,500 руб/час.

Нагрузка по ролям.

1. Младший программист (30% времени).

Часы работы: 20 часов.

Оплата: $20 \text{ часов} \cdot 2,500 \text{ руб/час} = 50,000 \text{ руб.}$

2. Программист (Middle) (50% времени).

Часы работы: 50 часов

Оплата: $50 \text{ часов} \cdot 3,500 \text{ руб./час} = 175,000 \text{ руб.}$

3. Старший программист (20% времени)

Часы работы: 30 часов.

Оплата: $30 \text{ часов} \cdot 4,500 \text{ руб./час} = 135,000 \text{ руб.}$

Общая оплата труда программистов.

Теперь сложим оплату труда всех программистов:

1. Младший программист: 50,000 руб.

2. Программист (Middle): 175,000 руб.

3. Старший программист: 135,000 руб.

Общая сумма:

$50,000 \text{ руб.} + 175,000 \text{ руб.} + 135,000 \text{ руб.} = 360,000 \text{ руб.}$

Таким образом, сумма оплаты труда программистов за определенный период разработки составит 360,000 руб. Это значение может варьироваться в зависимости от конкретных условий работы, объема задач и других факторов, таких как налоги и дополнительные выплаты.

3.2 Расчет срока окупаемости автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии

«Расчет срока окупаемости автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии можно провести, учитывая затраты на разработку системы и ожидаемую экономию или доход от ее внедрения» [17].

1. Исходные данные.

Общие затраты на разработку системы: 3,582,000 руб. (из предыдущих расчетов).

Ожидаемая экономия или доход от внедрения системы в год: 1,200,000 руб.

2. Расчет срока окупаемости.

Срок окупаемости (S) рассчитывается по формуле:

$$S = \frac{\text{Общие затраты}}{\text{Ежегодная экономия или доход}} \quad (8)$$

Подставим известные значения:

$$S = \frac{3,582,000 \text{ руб.}}{1,200,000 \text{ руб./год}} \approx 2,99 \text{ года}$$

3. Интерпретация результатов.

«Таким образом, срок окупаемости автоматизированной системы учета технического обслуживания и ремонта оборудования составит примерно 3 года. Это означает, что через три года после внедрения системы предприятие начнет получать чистую экономию или доход, превышающий первоначальные затраты на разработку системы»[13].

В третьем разделе был проведен расчет экономической эффективности автоматизированной системы учета технического обслуживания и ремонта оборудования на предприятии.

В результате анализа затрат на разработку системы было установлено, что общие расходы составляют 3,582,000 рублей. Эти затраты включают в себя проектирование, разработку программного обеспечения, затраты на базу данных, тестирование, внедрение и обучение сотрудников, а также техническую поддержку на первый год. Подробный анализ затрат позволяет понять, какие компоненты разработки требуют наибольших инвестиций, и выделить ключевые направления для оптимизации.

Оценка срока окупаемости показала, что система окупится примерно за 3 года. Это означает, что через три года после внедрения система начнет приносить чистую экономию или доход, превышающие первоначальные затраты. Ожидаемая ежегодная экономия составила 1,200,000 рублей, что свидетельствует о высокой эффективности системы и ее способности значительно улучшить финансовые показатели предприятия.

Таким образом, результаты расчетов подтверждают целесообразность инвестиций в автоматизированную систему учета технического обслуживания и ремонта оборудования. Внедрение данной системы не только оправдывает затраты, но и создает возможности для дальнейшего улучшения процессов управления, повышения производительности и снижения затрат на обслуживание оборудования. Это в конечном итоге может привести к улучшению конкурентоспособности предприятия и повышению его финансовых результатов.

Заключение

В процессе проведенного исследования была разработана система учета технического обслуживания и ремонта оборудования на предприятии, которая направлена на повышение эффективности управления производственными ресурсами.

В первом разделе работы была рассмотрена актуальность разработки автоматизированной системы учета, приведен обзор существующих решений и их недостатков. Было установлено, что современное предприятие нуждается в системе, которая способна повысить точность учета, снизить человеческий фактор и обеспечить удобство мониторинга состояния оборудования.

Во втором разделе была произведена разработка системы. В ходе работы был выбран и обоснован язык программирования, а также база данных, которые наиболее соответствуют требованиям предприятия. Разработана структурная схема системы, а также предложен интерфейс программы, ориентированный на пользователей. «В результате проведенных исследований и проектирования была создана основа для внедрения системы, которая позволит автоматизировать процессы учета технического обслуживания и ремонта» [9].

В третьем разделе работы проведены расчеты экономической эффективности внедрения разработанной системы. Рассчитаны затраты на разработку и внедрение системы, а также определен срок ее окупаемости. Полученные результаты показывают, что система обладает высокой экономической целесообразностью, что подтверждает актуальность ее разработки и внедрения на предприятии.

Таким образом, проведенная работа доказывает, что внедрение автоматизированной системы учета технического обслуживания и ремонта оборудования позволит значительно оптимизировать производственные

процессы, повысить уровень контроля за состоянием техники и сократить издержки, связанные с ее простоем и ремонтом

Внедрение системы способствует не только улучшению показателей работы оборудования, но и повышает уровень ответственности за выполнение технического обслуживания со стороны персонала, снижает риски поломок и аварийных ситуаций. Это, в свою очередь, ведет к снижению затрат на внеплановые ремонты и повышению общей производительности предприятия.

Система учета технического обслуживания и ремонта оборудования позволит оперативно получать данные о техническом состоянии каждого единичного оборудования, планировать график обслуживания, учитывать затраты на ремонты и отслеживать историю выполнения всех работ. Это даст возможность не только улучшить управление техническим состоянием оборудования, но и повысить эффективность эксплуатации производственных мощностей в целом.

В результате проведенного исследования и разработки системы можно сделать вывод, что предложенная автоматизированная система учета будет иметь значительное влияние на повышение производственной эффективности, что в долгосрочной перспективе принесет значительную экономическую выгоду для предприятия.

Список используемой литературы

1. Microsoft Dynamics [Электронный ресурс] : Microsoft Dynamics. URL: <https://dynamics.microsoft.com/ru-ru/>
2. ГОССТАНДАРТ РОССИИ Руководящий документ IDEF0-2000. Методология функционального моделирования IDEF0 : введ. Постановлением Госстандарта России от 2000 г. №3. URL: <https://www.rst.gov.ru/portal/gost>
3. Oracle Siebel Applications | Oracle Россия и СНГ [Электронный ресурс] : Oracle. URL: <https://www.oracle.com/ru/applications/siebel/>
4. Госреестр. URL: <https://rocpeestr.pф/GosReestr/Show/10032143>).
5. Моделирование бизнеса — IDEF, UML, ARIS [Электронный ресурс] : Business Analysis. URL: <https://analytics.infozone.pro/business-modeling-idef-uml-aris>
6. Анисифоров А.Б., Анисифорова Л.О. Методики оценки эффективности информационных систем и информационных технологий в бизнесе: учебное пособие [Электронный ресурс]. URL: <https://elibr.spbstu.ru/dl/2/3876.pdf/download/3876.pdf>
7. ГОСТ Р 50779.42-99 Государственный стандарт российской федерации. Статистические методы. Контрольные карты Шухарта [Электронный ресурс]. URL: <https://docs.cntd.ru/document/1200025672>
8. A Guide to the Entity Relationship Diagram (ERD) [Электронный ресурс]. URL: <https://www.databasestar.com/entity-relationship-diagram/>
9. Axure RP - UX Prototypes, Specifications, and Diagrams [Электронный ресурс]. URL: <https://www.axure.com/>
10. Туккель, И. Л. Управление инновационными проектами [Электронный ресурс] : учебник / И. Л. Туккель, А. В. Сурина, Н. Б. Культин ; под. ред. И. Л. Туккеля. - 2-е изд., доп. - Санкт-Петербург: БХВ-Петербург,

2020. - 416 с. - (Учебная литература для вузов). - URL: <https://znanium.com/catalog/product/1818470>

11. Шишов, О. В. Технические средства автоматизации и управления [Электронный ресурс] : учебное пособие / О. В. Шишов. - Москва : ИНФРА-М, 2021. - 396 с. - (Высшее образование: Бакалавриат). – URL: <https://znanium.com/catalog/product/1157118>

12. Дубнов, П. Ю. Access 2000. Проектирование баз данных / П. Ю. Дубнов. - М.: ДМК, 2017. – 272 с.

13. Andrew Beak. PHP 7 Zend Certification Study Guide : Apress, 2017. URL: <https://itbook.store/books/9781484232453>

14. Jesper Wisborg Krogh. MySQL Concurrency : Apress, 2021. URL: <https://itbook.store/books/9781484266519>

15. MySQL 8.0 Reference Manual [Электронный ресурс] : URL: <https://dev.mysql.com/doc/refman/8.0>

16. Михеев А.Г. Процессное управление на свободном программном обеспечении [Электронный ресурс]/ А.Г. Михеев. – Электрон. Текстовые данные. –М.: Интернет–Университет Информационных Технологий (ИНТУИТ), 2016. 230с. URL: <http://www.iprbookshop.ru/39562.html>

17. Сорокин А.А. Реинжиниринг бизнес–процессов [Электронный ресурс]: учебное пособие / А.А. Сорокин, А.Ю. Орлова. – Электрон. Текстовые данные. – Ставрополь: Северо–Кавказский федеральный университет, 2014. 212с. URL: <http://www.iprbookshop.ru/63003.html>

18. Эксперт Online. Наиболее распространенные информационные системы управления персоналом [Электронный ресурс]: официальный сайт компании. URL: http://expert.ru/ratings/table_50227/20. Ковязин, А. Н. Архитектура, администрирование и разработка приложений баз данных в InterBase/FireBird/Yaffil / А. Н. Ковязин, С. М. Востриков. - М.: Кудиц-образ; Издание 4-е, 2017. – 496 с.

19. Crystal Reports, Developer for Visual Studio Downloads – Business Intelligence (BusinessObjects) - SCN Wiki [Электронный ресурс] URL: <https://wiki.scn.sap.com/wiki/display/BOBJ/Crystal+Reports%2C+Developer+for+Visual+Studio+Downloads>

20. Салтан А.А. Основы бизнес-информатики : учебник / А.А. Салтан, В.В. Иванова, Т.А. Лезина – СПб. : СПбГУ, 2014. – 244 с. – Текст : – URL: <http://znanium.com/catalog/product/941009>

21. Бабич А. В. Введение в UML [Электронный ресурс]: учебное пособие. Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2022. 198 с. URL: <https://www.iprbookshop.ru/120473.html>

22. UML 2.ru – Сообщество Аналитиков [Электронный ресурс]. URL: <https://www.uml2.ru/>

23. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose [Электронный ресурс]: учебное пособие. М. : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. 317 с. URL: <https://www.iprbookshop.ru/97554.html>

24. Молоткова Н. В., Хазанова Д. Л. Реинжиниринг бизнес-процессов [Электронный ресурс]: учебное пособие. Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2019. 81 с. URL: <https://www.iprbookshop.ru/99785.html>

25. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose : учебное пособие. М. : ИНТУИТ, Ай Пи Ар Медиа, 2020. 317 с. [Электронный ресурс]. URL: <https://www.iprbookshop.ru/97554.html>

26. Моделирование информационного обеспечения [Электронный ресурс]. URL: <https://intuit.ru/studies/courses/2195/55/lecture/1636?page=3>

27. Молоткова Н. В., Хазанова Д.Л. Реинжиниринг бизнес-процессов: учебное пособие. Тамбов : ТГТУ. 81 с. [Электронный ресурс]. URL: <https://www.iprbookshop.ru/99785.html>

28. Сколько стоят услуги программистов? [Электронный ресурс]. URL: <https://www.kadrof.ru/articles/46641>

29. Учебно-методическое обеспечение практики обучающихся колледжа как одно из условий реализации ФГОС СПО [Электронный ресурс]. URL: <https://e-koncept.ru/2015/95618.html>

31. Business Process Model and Notation (BPMN) [Электронный ресурс]. URL: <https://www.omg.org/spec/BPMN/2.0/PDF>