|  |
| --- |
| титул |
|  |

|  |
| --- |
| **ДОПУСКАЮ К ЗАЩИТЕ** |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| Заместитель директора  *О.В.Корешков*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |
| (дата) |

**ДИПЛОМНАЯ РАБОТА**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Разработка и исследование методов создания и использования систем управления изменениями в информационных технологиях | | | | |
| (тема) | | | | |
| Выпускная квалификационная работа должна быть выполнена в виде: | | | | |
| дипломной работы и демонстрационного экзамена | | | | |
| студентом группы | | 3-ИСП11-18 | | |
|  | | (номер группы) | | |
| Н. Д. Зверевым | | | |  |
| (И.О.Фамилия) | | | | (подпись, дата) |
| Основная профессиональная образовательная программа по специальности | | | | |
| 09.02.07 Информационные системы и программирование | | | | |
| (шифр и наименование специальности) | | | | |
| Форма обучения | | очная | | |
| Руководитель |  | преподаватель |  | В. А. Мартиросян |
|  |  | | |  |
|  |  | | | (подпись, дата) |
| Консультант |  | преподаватель |  | В. А. Мартиросян |
| По практич.части |  | | |  |
|  |  | | | (подпись, дата) |
| Председатель предметной (междисциплинарной, модульной) комиссии | | | | |
| В.Д. Травин | | |  |  |
| (И.О.Фамилия) | | |  | (подпись, дата) |

Москва

2024

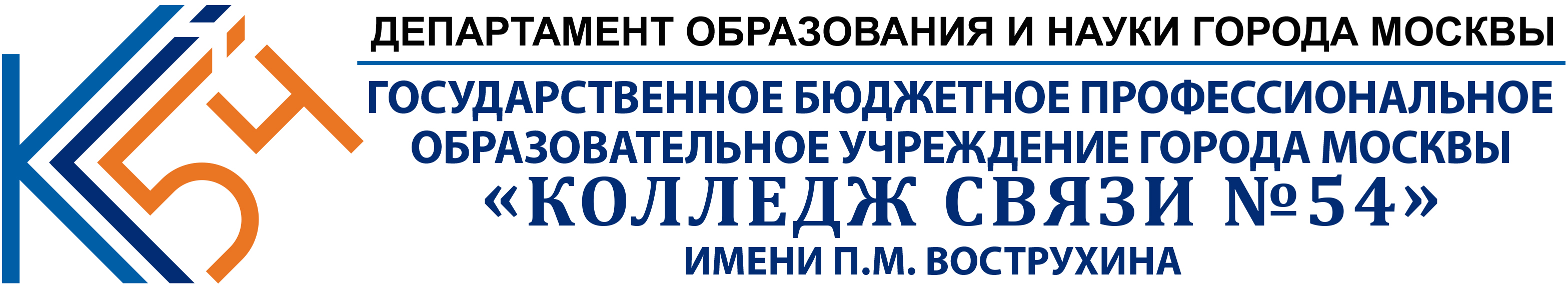
|  |
| --- |
| титул |
|  |

|  |
| --- |
| **УТВЕРЖДАЮ** |
| Заместитель директора  *О.В.Корешков*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |
|  |
| (дата) |

**ЗАДАНИЕ НА ДИПЛОМНУЮ РАБОТУ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студенту Звереву Никите Денисовичу | | | | | |
| (фамилия, имя, отчество полностью) | | | | | |
| **I. Тема дипломной работы** | | | | | |
| Разработка и исследование методов создания и использования систем управления изменениями в информационных технологиях | | | | | |
| **II. Срок сдачи студентом законченной работы** | | | ***«01» июня 2024 г.*** | |
| **III. Исходные данные** | | | | | |
|  | *Программные средства: среда разработки PyCharm Community Edition 2024.1* | | | | |
|  | *Практические навыки по компетенции информационные системы и программирование, полученные в период прохождения преддипломной практики в предприятии* | | | | |
| **IV. Перечень подлежащих разработке вопросов** | | | | | |
| 1. | *Проанализировать конкурсное задание и инфраструктурный лист ДЭ;* | | | | |
| 2. | *Проанализировать особенности предметной области;* | | | | |
| 3. | *Произвести сравнительный анализ программных средств;* | | | | |
| 4. | *Спроектировать и разработать систему контроля версий (схемы базы данных, прототипа графического приложения);* | | | | |
| 5. | *Разработать систему контроля версий (создание базы данных в СУБД MS SQL Server, графического приложения на языке программирования Python в среде разработки PyCharm Community Edition 2024.1* | | | | |
| **V. Перечень графического/иллюстрационного материала** | | | | | |
| 1. | *Схемы базы данных, а также самой базы данных с необходимой структурой;* | | | | |
| 2. | *Скриншоты экрана разработанного графического приложения;* | | | | |
| 3. | *Презентация* | | | | |
| **VI. Дата выдачи задания** | | | «\_12\_\_»\_\_\_\_марта\_2024\_\_г | |
| Руководитель | | В. А. Мартиросян | |  | |
|  | |  | | (подпись) | |
| Задание принял к исполнению | | Н.Д. Зверев | |  | |
|  | |  | | (подпись) | |

«\_12\_\_»\_\_\_\_марта\_2024\_\_г



КАЛЕНДАРНЫЙ ПЛАН ДИПЛОМНОЙ РАБОТЫ

(с указанием сроков выполнения отдельных этапов)

для специальности 09.02.07 Информационные системы и программирование

(группа 3ИСП11-18)

|  |  |  |
| --- | --- | --- |
| **Сроки преддипломной практики** | |  |
|  | Выбор темы, руководителя, оформление заявления | до 02.02.2024г. |
|  | Утверждение темы дипломной работы | до 10.03.2024 г |
|  | Выполнение задания по теме дипломной работы | с 13.03.2024 по 24.05.2024 |
|  | Предоставление отчета по практике руководителю | До 14.05.2024 |
|  | Аттестация по практике | 14.05.2024 |
| **Подготовка дипломной работы** | |  |
|  | Утверждение задания на дипломную работу | с 14. 05.2024 -31.05.2024 |
|  | Подбор и анализ исходной информации |
|  | Подготовка и утверждение плана (оглавления) дипломной работы |
|  | Работа над разделами (главами) и устранение замечаний руководителя дипломной работы | с 01.06.2024 - 07.06.2024 |
|  | Согласование содержания дипломной работы, устранение замечаний | с 08.06.2024 - 14.06.2024 |
|  | **Оформление и представление руководителю полного текста работы. Получение отзыва руководителя дипломной работы** | с 08.06.2024 - 14.06.2024 |
|  | **Предоставление студентом готовой дипломной работы рецензенту** |
|  | **Сдача ДЭ** | с 28.05.2024- 21.06.2024 |
|  | **Предзащита дипломной работы** | с 8.06.2024 - 14.06.2024 |

**Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.А Мартиросян**

План принял к исполнению «12» марта 2024 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc169278080)

[ГЛАВА 1. ОСНОВЫ СИСТЕМ УПРАВЛЕНИЯИЗМЕНЕНИЯМИВ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ 7](#_Toc169278081)

[1.1 Теоретические аспекты управления изменениями 7](#_Toc169278082)

[1.2 Методы создания систем управления изменениями 8](#_Toc169278083)

[1.3 Ключевые принципы разработки систем управления изменениями 10](#_Toc169278084)

[1.4 Технологии и инструменты для систем управления изменениями 12](#_Toc169278085)

[1.5 Методы исследования эффективности систем управления изменениями 15](#_Toc169278086)

[ВЫВОДЫ 17](#_Toc169278087)

[ГЛАВА 2. РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПРОГРАММНОГО РЕШЕНИЯ 18](#_Toc169278088)

[2.1 Технологические аспекты разработки приложения 18](#_Toc169278089)

[2.2 Постановка задачи 18](#_Toc169278090)

[2.3 Анализ ТЗ 19](#_Toc169278091)

[2.3.1 Определение основных требований и функциональности 19](#_Toc169278092)

[2.3.2 Технологии и инструменты 20](#_Toc169278093)

[2.4 Проектирование базы данных 20](#_Toc169278094)

[2.4.1 Определение структуры базы данных для хранения информации о пользователях и версий кода 20](#_Toc169278095)

[2.4.2 Разработка схемы базы данных 21](#_Toc169278096)

[2.5 Создание интерфейса и функционала 21](#_Toc169278097)

[2.5.1 Разработка пользовательского интерфейса 21](#_Toc169278098)

[ВЫВОДЫ 24](#_Toc169278099)

[ЗАКЛЮЧЕНИЕ 25](#_Toc169278100)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 26](#_Toc169278101)

[ПРИЛОЖЕНИЯ 28](#_Toc169278102)

# ВВЕДЕНИЕ

В современном мире информационные технологии играют ключевую роль в различных сферах бизнеса и общественной жизни. С постоянным развитием технологий и увеличением сложности IT-проектов, важно обеспечить эффективное управление изменениями в информационных системах. Отсутствие системы управления изменениями может привести к снижению производительности, непредвиденным ошибкам, а также потере ценных данных.

Тема дипломной работы «Разработка и исследование методов создания и использования систем управления изменениями в информационных технологиях» является актуальной и важной в современной обстановке. Эффективное управление изменениями позволяет сделать IT-проекты более адаптивными к изменяющимся требованиям бизнеса и повысить их успешность.

Исследование данной темы позволит выявить лучшие практики, методики и инструменты для создания и использования систем управления изменениями в информационных технологиях. Актуальность работы заключается в том, что она способствует повышению профессионализма специалистов в области IT-управления, обеспечивает надёжность и устойчивость информационных систем к изменениям, а также способствует оптимизации процессов разработки и поддержки IT-проектов.

Данная дипломная работа направлена на исследование современных методов управления изменениями в IT-проектах, их практическую реализацию с использованием инновационных подходов и средств, а также на анализ эффективности внедрения систем управления изменениями в информационных технологиях.

Целью данной дипломной работы является создание методологии, которая поможет оптимизировать процессы управления изменениями в IT-проектах, повысить их эффективность, уменьшить риски и повысить качество окончательного продукта.

Для выполнения поставленной цели необходимо выполнить следующие задачи:

1. Изучение существующих подходов: проанализировать существующие методики и инструменты управления изменениями в информационных технологиях и выявить их сильные и слабые стороны.
2. Разработка методологии: разработать собственную методологию создания и внедрения систем управления изменениями, учитывающую специфику IT-проектов и требования бизнеса.
3. Практическая реализация: провести практическую реализацию разработанных методов на примере конкретного IT-проекта для проверки их эффективности и применимости.
4. Оценка результатов: оценить эффективность и влияние внедрения систем управления изменениями на процессы разработки, качество продукта и уровень удовлетворённости пользователей.
5. Формирование рекомендаций: предложить рекомендации для улучшения управления изменениями в информационных технологиях на основе полученных результатов и анализа.

Объектом дипломной работы является система управления изменениями в области информационных технологий.

Предметом исследования данной дипломной работы являются методы разработки и использования систем управления изменениями в информационных технологиях. Основное внимание уделяется анализу существующих методов управления изменениями, разработке новых подходов, исследованию их применимости в информационных технологиях с целью повышения эффективности управления процессами изменений в IT сфере.

# ГЛАВА 1. ОСНОВЫ СИСТЕМ УПРАВЛЕНИЯИЗМЕНЕНИЯМИВ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ

## 1.1 Теоретические аспекты управления изменениями

Понятие изменений в информационных технологиях

Изменения в информационных технологиях– это процессы, методы и технологии, направленные на внедрение новых или улучшение существующих информационных систем, программного обеспечения, аппаратного обеспечения, архитектуры сетей и других IT-решений в организации. Эти изменения могут касаться обновления технологического стека, повышения безопасности информационных ресурсов, оптимизации бизнес-процессов, внедрения цифровых инноваций и других аспектов, целью которых является улучшение эффективности и конкурентоспособности организации.

Принципы систем управления изменениями

Существует несколько основных принципов управления изменениями:

1. Прозрачность и открытость: обеспечение доступа к информации о планируемых изменениях и их результатах.
2. Участие заинтересованных сторон: вовлечение сотрудников и руководителей в процесс принятия и внедрения изменений.
3. Обратная связь и обучение: систематический сбор обратной связи и обучение персонала для успешной адаптации к изменениям.
4. Управление рисками и качеством: анализ и управление рисками, а также обеспечение высокого качества внедряемых изменений.
5. Контроль и оценка: постоянный контроль и оценка результатов внедрения изменений для корректировки стратегии при необходимости.

Роль и значение управления изменениями в IT-проектах

Управление изменениями в IT-проектах играет ключевую роль и имеет следующие значения:

1. Минимизация рисков: идентификация, анализ и управление рисками, связанными с изменениями в проекте.
2. Обеспечение качества: контроль и оценка внедряемых изменений для обеспечения качества продукта.
3. Повышение эффективности: эффективное управление процессом изменений для сокращения времени и затрат.
4. Коммуникация и вовлечённость: улучшение коммуникации и вовлеченности участников проекта для успешной реализации изменений.
5. Адаптация к изменениям: гибкое реагирование на изменения и эффективное принятие корректирующих мер.

## 1.2 Методы создания систем управления изменениями

Ознакомление с основными методами и инструментами управления изменениями

Основные методы управления изменениями в сфере IT:

1. ITIL (IT Infrastructure Library): набор bewek Р. А. и др. основных рекомендаций по управлению ИТ-сервисами, включая процессы управления изменениями.
2. Agile и DevOps: гибкие методологии разработки и управления проектами, способствующие быстрой и качественной реализации изменений.
3. Change Management Frameworks:фреймворки управления изменениями, такие как PROSCI ADKAR Model или Kotter’s 8-Step Process, обеспечивающие систематический и структурированный подход к изменениям.
4. Kanban и Scrum:методики управления процессами и работой, которые могут быть применены для управления изменениями в IT-проектах.
5. Continuous Integration/Continuous Delivery (CI/CD):автоматизированные процессы сборки, тестирования и развертывания программного обеспечения для эффективного управления изменениями.

Основные инструменты управления изменениями в сфере IT:

1. Системы управления конфигурациями (Configuration Management Systems)для отслеживания изменений в ИТ-инфраструктуре.
2. Системы отслеживания ошибок и запросов (Issue Tracking Systems) для учета и управления изменениями и запросами на изменения.
3. Автоматизированные алгоритмы оценки рисков при внедрении изменений для предотвращения возможных негативных последствий.
4. Системы мониторинга и аналитики изменений для отслеживания и анализа воздействия внесенных изменений на IT-среду и бизнес-процессы.

Сравнительный анализ различных подходов к управлению изменениями

1. ITIL против Agile и DevOps:
   1. ITIL: подходит для управления IT-сервисами, включая процессы управления изменениями, но склонен к более формальным и медленным изменениям;
   2. Agile и DevOps: гибкие методологии, способствующие быстрой и качественной реализации изменений, но могут требовать дополнительных усилий для организации процессов управления изменениями.
2. Change Management Frameworks против Kanban и Scrum:
   1. Change Management Frameworks: предлагают систематический и структурированный подход к изменениям, что важно для крупных и сложных изменений;
   2. Kanban и Scrum: методики управления процессами, которые могут быть более гибкими и приспосабливающимися к изменениям в реальном времени.
3. Continuous Integration/Continuous Delivery (CI/CD) против автоматизированные алгоритмы оценки рисков:
   1. CI/CD: позволяют автоматизировать процессы сборки, тестирования и развертывания программного обеспечения, обеспечивая быструю поставку изменений;
   2. Автоматизированные алгоритмы оценки рисков: помогают предотвращать возможные негативные последствия при внедрении изменений, что важно для обеспечения стабильности и безопасности.

## 1.3 Ключевые принципы разработки систем управления изменениями

Проектирование системы управления изменениями в информационных технологиях

Ключевые этапы проектирования системы управления изменениями в информационных технологиях:

1. Определение целей системы управления изменениями: идентификация основных целей и задач системы управления изменениями в информационных технологиях, например, улучшение процесса внедрения новых технологий, снижение рисков изменений, повышение принятия изменений среди сотрудников и т.д.
2. Анализ потребностей и требований: изучение потребностей организации, особенностей информационных технологий, структуры и культуры организации для установления конкретных требований к системе управления изменениями.
3. Выбор подходов и методов управления изменениями: определение подходов к управлению изменениями, которые наилучшим образом соответствуют целям и требованиям системы управления в информационных технологиях (например, ADKAR модель, Kotter’s 8-ступенчатая модель изменений, PROSCI ADKAR модель).
4. Разработка структуры системы управления изменениями: создание организационной системы управления изменениями, определение ролей и обязанностей участников, разработка процессов и процедур управления изменениями.
5. Определение инструментов и технологий:выбор специализированных инструментов и технологий, которые помогут эффективно реализовать систему управления изменениями в информационных технологиях (например, программное обеспечение для управления проектами, системы отслеживания изменений и т.д.)
6. Планирование внедрения системы:разработка плана внедрения системы управления изменениями, определение этапов, сроков, ресурсов и ответственных лиц, проведение оценки рисков и разработка стратегии обучения и коммуникации сотрудников.

Раскрытие принципов разработки систем изменений

1. Гибкость и адаптивность:система должна быть способной приспосабливаться к изменяющимся потребностям и условиям среды.
2. Простота и понятность:разработка системы изменений должна быть понятной и простой для всех участников процесса.
3. Автоматизация и эффективность:использование автоматизированных средств для ускорения и снижения вероятности ошибок в процессе разработки изменений.
4. Тестирование и обратная связь:проведение тестирования изменений и сбор обратной связи для обеспечения качества и корректности внедрения.
5. Безопасность и конфиденциальность:уделять особое внимание обеспечению безопасности и конфиденциальности в процессе разработки и внедрения изменений.

## 1.4 Технологии и инструменты для систем управления изменениями

Обзор современных технологий и инструментов для управления изменениями

1. Инструменты управления проектами:
2. Jira: популярный инструмент для управления проектами, задачами и изменениями;
3. Asana: платформа для планирования и управления проектами с функциональностью совместной работы;
4. Trello: гибкая система карточек для организации проектов и задач.
5. Инструменты для коммуникации:
6. Slack: платформа для коммуникации и совместной работы в реальном времени;
7. Microsoft Teams: инструмент для коммуникации и совместной работы внутри организации;
8. Zoom: популярное приложение для проведения онлайн-встреч и конференц-звонков.
9. Инструменты управления изменениями:
10. Change Management Tool (CMT): интегрированный инструмент для управления и отслеживания изменений в организации;
11. Change Control Software: программное обеспечение для контроля и управления изменениями в информационных технологиях;
12. IT Service Management (ITSM) Tools: инструменты для автоматизации процессов управления изменениями в информационных технологиях.
13. Инструменты для анализа данных и мониторинга:
14. Power BI: инструмент для визуализации данных и анализа результатов изменений;
15. Splunk: платформа для мониторинга и анализа данных и событий в реальном времени.

Их применение в практике и преимущества использования

1. Jira:
2. Применение: управление проектами, задачами, отслеживание изменений.
3. Преимущества: гибкая настройка, система управления проектами и задачами, интеграция с другими инструментами.
4. Asana:
5. Применение: планирование и управление проектами, коммуникация.
6. Преимущества: интуитивный интерфейс, совместная работа, гибкие конфигурации задач.
7. Trello:
8. Применение: организация проектов и задач в виде карточек.
9. Преимущества: простота использования, визуальное представление задач, гибкость конфигурации досок.
10. Slack:
11. Применение: коммуникация в реальном времени, обмен файлами.
12. Преимущества: гибкие интеграции, синхронизация с другими сервисами, создание каналов для обсуждений.
13. Microsoft Teams:
14. Применение: коллаборация, видеоконференции, обмен документами.
15. Преимущества: интеграция с другими сервисами Microsoft, удобство коммуникации внутри организации.
16. Zoom:
17. Применение: онлайн-встречи, видеоконференции, вебинары.
18. Преимущества: простота использования, высокое качество видео и звука, возможность участия большого числа участников.
19. Change Management Tool (CMT):
20. Применение: управление и отслеживание изменений.
21. Преимущества: централизованный контроль изменений, автоматизация процессов управления изменениями, улучшение прозрачности и видимости.
22. Change Control Software:
23. Применение: контроль изменений в информационных технологиях.
24. Преимущества: мониторинг изменений, обеспечение безопасности и целостности системы, улучшение процесса принятия решений.
25. IT Service Management (ITSM) Tools:
26. Применение: управление IT-сервисами и изменениями.
27. Преимущества: повышение эффективности, автоматизация процессов, улучшение качества обслуживания.
28. Power BI:
29. Применение: визуализация данных, аналитика.
30. Преимущества: лёгкость использования, настраиваемы отчёты и дашборды, интеграция с различными источниками данных.
31. Splunk:
32. Применение: мониторинг, анализ и визуализация данных и событий.
33. Преимущества: обнаружение и предотвращение инцидентов, аналитика безопасности, поиск и коррекция проблем.

## 1.5 Методы исследования эффективности систем управления изменениями

Методики оценки эффективности внедрённых систем управления изменениями

1. Ключевые показатели эффективности (KPI):определение и измерение показателей, таких как время реакции на изменения, уровень удовлетворенности пользователей, уровень создания ценности для бизнеса.
2. Опросы и обратная связь:проведение опросов и интервью сотрудников и руководителей для выявления мнения о системе управления изменениями и ее влиянии на процессы.
3. Анализ изменений в процессах:изучение изменений в бизнес-процессах и технологических процессах после внедрения системы управления изменениями.
4. Использование метрик и отчетности:оценка с помощью ключевых метрик, формирование отчетов о производительности и эффективности системы управления изменениями.
5. Сравнительный анализ до и после внедрения:сопоставление результатов и показателей до и после внедрения системы для оценки достигнутых изменений.

Исследовательские подходы для изучения влияния систем на результаты проектов

1. Анализ случаев использования:проведение глубокого анализа реальных проектов в сфере информационных технологий, в которых применялись различные системы управления изменениями. Этот подход позволит выявить преимущества и недостатки каждой системы, а также их влияние на результаты проектов.
2. Опросы и интервью:проведение опросов и интервью с участниками проектов, менеджерами по управлению проектами и специалистами по управлению изменениями. Это позволит собрать мнения и оценки относительно эффективности различных систем управления изменениями и их влияния на результаты проектов.
3. Количественный анализ данных:сбор и анализ количественных данных о проектах, включая сроки выполнения, бюджеты, качество и другие метрики проектного успеха. Это позволит провести статистическое сопоставление результатов проектов с применением различных систем управления изменениями.
4. Экспериментальные исследования:проведение контролируемых экспериментов, в которых одним группам проектов будет применяться определенная система управления изменениями, а другим – другая. После завершения проектов будут сравниваться их результаты для выявления различий и определения влияния систем управления изменениями.
5. Метрики проектного успеха:определение и использование специальных метрик для оценки успешности проектов в зависимости от применяемой системы управления изменениями. Это может включать такие показатели, как время выполнения проекта, бюджетные затраты, уровень удовлетворенности заказчиков и т.д.

# ВЫВОДЫ

Проведённый анализ теоретических аспектов управления изменениями в информационных технологиях позволяет сделать следующие выводы:

1. Системы управления изменениями являются неотъемлемой частью процессов в сфере информационных технологий. Понимание теоретических основ управления изменениями позволяет эффективно управлять процессами изменений в организации.
2. Разработка систем управления изменениями требует применения методов, учитывающих специфику организации и особенности ее информационной инфраструктуры. Важно учитывать ключевые этапы создания таких систем для успешной их реализации.
3. Основные принципы разработки систем управления изменениями включают в себя учёт интересов заинтересованных сторон, адаптивность к изменениям внешней среды, прозрачность процессов и ответственность за принимаемые решения.
4. Для эффективной работы систем управления изменениями необходимо использовать современные технологии и инструменты, способствующие автоматизации процессов управления изменениями и обеспечению контроля за ними.
5. Исследование эффективности систем управления изменениями позволяет оценить их влияние на производительность организации, выявить слабые места и обосновать необходимость внесения изменений для повышения качества управления изменениями в информационных технологиях.

Таким образом, внимательное изучение теоретических аспектов систем управления изменениями в информационных технологиях позволяет сформировать базу для дальнейшего практического исследования и разработки в данной области.

# ГЛАВА 2. РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПРОГРАММНОГО РЕШЕНИЯ

## 2.1 Технологические аспекты разработки приложения

Разработанная система контроля версий на языке программирования Python с использованием библиотеки Tkinter в среде PyCharm Community Edition 2024.1 представляет собой современное решение для управления изменениями в коде. Она обеспечивает эффективное отслеживание правок, совместную работу и масштабируемость благодаря модульной структуре и системе плагинов. Использование кросс-платформенной библиотеки Tkinter позволило создать удобный пользовательский интерфейс, а интеграция с мощной средой разработки PyCharm Community Edition 2024.1 упростила процесс создания и тестирования приложения.

Для хранения данных различных версий кода была выбрана база данных Room, которая предоставляет удобный и эффективный способ организации информации. Запросы к базе данных писались на языке SQLite, что обеспечивает высокую производительность и надежность операций с данными.

Таким образом, разработка системы контроля версий основана на передовых технологиях, которые обеспечивают удобство использования, высокую производительность и надежность работы. В результате получается инновационное приложение, которое отвечает потребностям пользователей в удобном и эффективном инструменте для создания и управления различными версиями кода.

## 2.2 Постановка задачи

Для разработки системы контроля версий необходимо выполнить следующие задачи:

1. Анализ ТЗ:
   1. Понять основные требования и функциональность;
   2. Определить технологии и инструменты, необходимые для создания системы контроля версий.
2. Проектирование базы данных:
   1. Определить структуру базы данных для хранения пользователей и версий кода;
   2. Разработать схему базы данных, учитывая все данные, хранящиеся в ней: идентификационный номер пользователя, имя пользователя, код пользователя.
3. Создание интерфейса и функциональности:

3.1 Разработать пользовательский интерфейс для возможности сохранения различных версий кода, возможности отслеживания авторства изменений, возможности откатываться к предыдущим версиям кода, а также возможности просмотра предыдущих версий кода.

## 2.3 Анализ ТЗ

## 2.3.1 Определение основных требований и функциональности

Изучить требования к системе контроля версий, выделить основные функциональные возможности.

Система контроля версий должна содержать следующие функциональные возможности:

1. Сохранение различных версий кода
2. Возможность отслеживать авторство изменений
3. Возможность откатываться к предыдущим версиям кода
4. Возможность просмотра предыдущих версий кода

## 2.3.2 Технологии и инструменты

Разработка системы контроля версий производилась в среде разработки PyCharm Community Edition 2024.1 на языке программирования Python с использованием библиотеки Tkinter.

В качестве базы данных используется Room, а для запросов к ней SQLite.

## 2.4 Проектирование базы данных

## 2.4.1 Определение структуры базы данных для хранения информации о пользователях и версий кода

Для проектирования базы данных необходимо создать таблицу со следующими полями:

1. ID (Идентификационный номер пользователя):
   1. Тип данных: INTEGER (целое число);
   2. Автоинкрементируемый первичный ключ, уникальный для каждой записи.
2. Имя пользователя:
   1. Тип данных: STRING (текстовая строка);
   2. Содержит имя пользователя.
3. Версия кода:
   1. Тип данных: STRING (текстовая строка);
   2. Содержит описание версии кода.

## 2.4.2 Разработка схемы базы данных

База данных SQLite, которая позволяет просматривать список различных версий кода, их изменения, а также их авторов.



Рисунок 1. Схема базы данных

## 2.5 Создание интерфейса и функционала

## 2.5.1 Разработка пользовательского интерфейса

При запуске приложения отображается главное окно с тремя полями:

* Автор: поле для ввода имени автора текущей версии кода;
* Код: текстовое поле для ввода кода;
* Кнопки: три кнопки для сохранения, просмотра и отката версий кода.

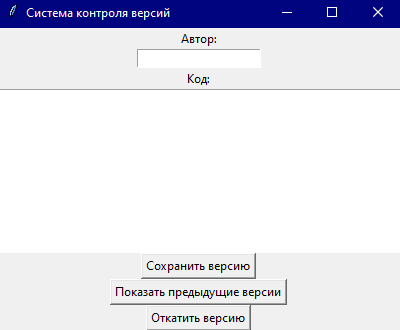


Рисунок 2. Главная страница системы контроля версий

Чтобы сохранить новую версию кода, введите своё имя в поле «Автор» и введите код в текстовое поле «Код». Затем нажмите кнопку «Сохранить версию». Приложение сохранит версию кода вместе с именем автора.

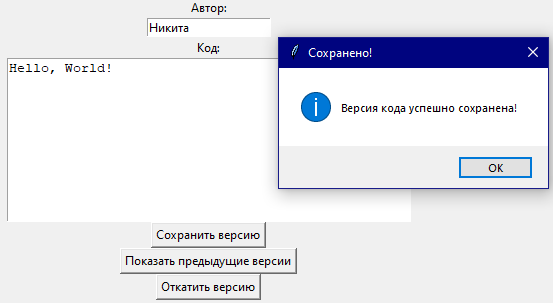


Рисунок 3. Интерфейс для сохранения системы контроля версий

Чтобы откатить код к предыдущей версии, выберите версию из списка в окне «Показать предыдущие версии» и нажмите на кнопку «Откатить версию». Приложение откатит код к выбранной версии.

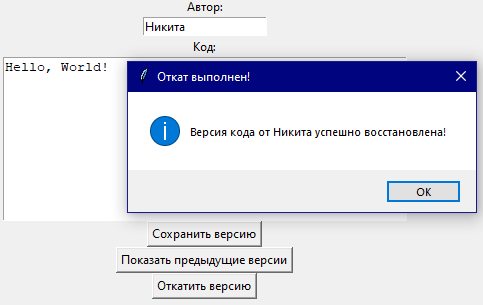


Рисунок 3. Интерфейс для отката к предыдущим версиям кода

Чтобы просмотреть список всех сохранённых версий кода, нажмите кнопку «Показать предыдущие версии». Приложение отобразит список всех версий в новом окне.

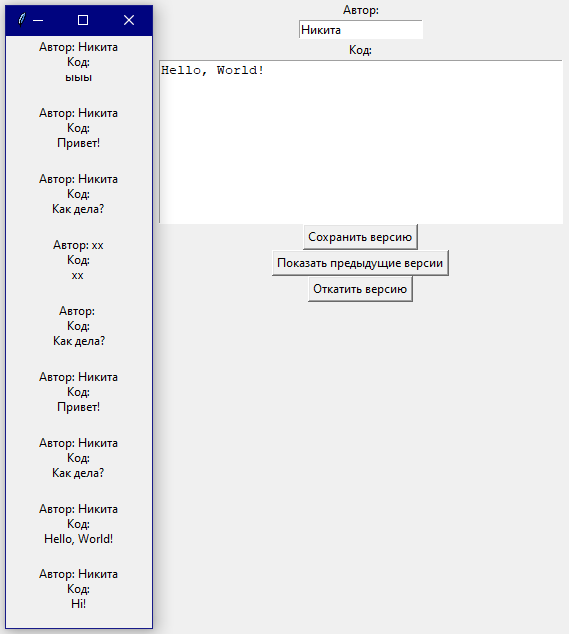


Рисунок 4. Интерфейс для просмотра предыдущих версий кода

# ВЫВОДЫ

В результате выполнения дипломной работы была разработана система контроля версий для добавления различных версий кода и управлению ими с учётом потребностей пользователей и их функциональных требований.

Основной целью создания системы контроля версий было разработать удобный инструмент, который позволит пользователям создавать различные версии кода и отслеживать их изменения с использованием современных технологий.

Для реализации системы контроля версий были использованы такие современные технологии и инструменты, как Python с использованием библиотеки Tkinter в среде разработки PyCharm Community 2024.1, Room в качестве базы данных, а для запросов к ней SQLite.

Таким образом, основной целью при создании системы контроля версий было предложить пользователям удобный и безопасный способ отслеживания различных изменений в версиях кода, сочетающий в себе современные технологии и функциональность.

# ЗАКЛЮЧЕНИЕ

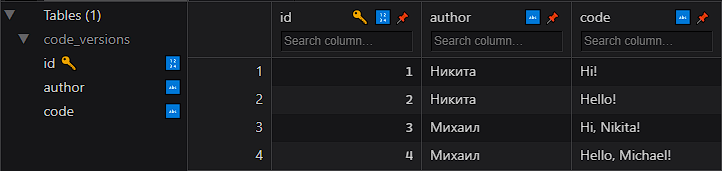
В ходе выполнения дипломной работы были достигнуты все поставленные цели и задачи. Проведён анализ существующих подходов к управлению изменениями в IT-проектах, разработана собственная методология создания и внедрения систем управления изменениями, проведена её практическая реализация на примере конкретного IT-проекта. Оценена эффективность и влияние внедрения системы управления изменениями на процессы разработки, качество продукта и уровень удовлетворённости пользователей. На основании полученных результатов и анализа сформированы рекомендации по улучшению управления изменениями в IT-проектах.

Разработанная методология позволяет оптимизировать процесс управления изменениями, повысить его эффективность и уменьшить риски. Внедрение систем управления изменениями на основе данной методологии приводит к улучшению качества конечного продукта, сокращению сроков разработки и повышению удовлетворённости пользователей.

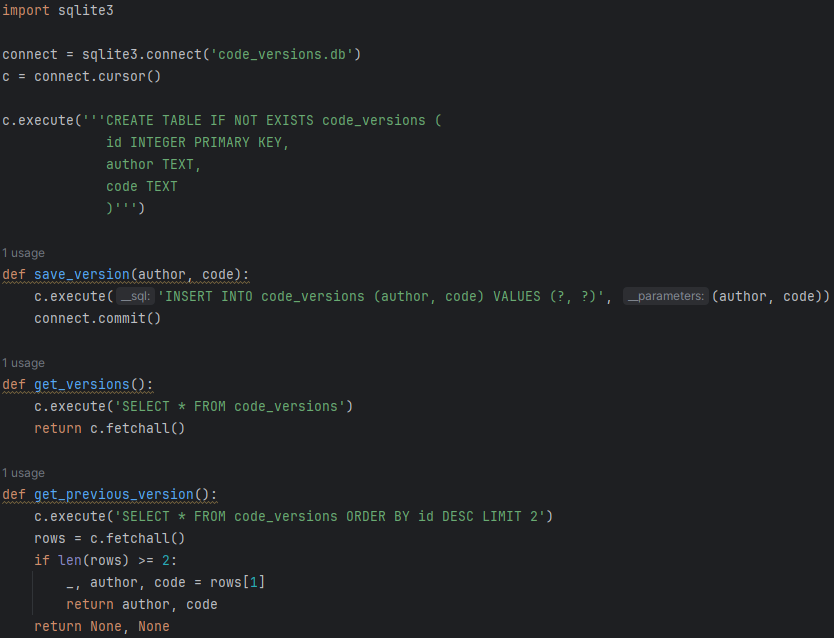
# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Как управлять изменениями с помощью системы service desk : научная статья – ITSM365 – 2022. – (https://itsm365.com/blog/upravlenie-izmeneniyami)
2. Методологии управления проектами: 12 популярных подходов : научная статья – Team Asana – 2024. – (https://asana.com/ru/resources/project-management-methodologies)
3. Пишем десктоп-приложение на Python с помощью Tkinter : научная статья – Skillbox Media – 2022. – (https://skillbox.ru/media/code/pishem-desktopprilozhenie-na-python-s-pomoshchyu-tkinter/)
4. Основы методологии DevOps : научная статья – Хабрух Нататья – 2021. – (https://proglib-io.turbopages.org/proglib.io/s/p/osnovy-metodologii-devops-2021-02-20)
5. Подробное руководство по управлению изменениями в ITIL : руководство – ServiceDesk Plus – 2020. – (https://www.manageengine.com/ru/service-desk/itil-change-management/what-is-change-management.html)
6. Работа с SQLite в Python : научная статья – Малеваний Андрей – 2023. – (https://habr.com/ru/articles/754400/)
7. Система контроля версий: определение, функции, популярные решения : блог – Редакция сайта GeekBrains – 2021. – (https://gb.ru/blog/sistema-kontrolya-versij/)
8. Система контроля версий : научная статья – Yandex Cloud – 2024. – (https://yandex.cloud/ru/docs/glossary/vcs?utm\_referrer=https%3A%2F%2Fyandex.ru%2F)
9. Современные процессы управления изменениями на предприятиях IT-сферы : научная статья – Авдеева И.Л., Марков Р.А. – 2020. – (https://cyberleninka.ru/article/n/sovremennye-protsessy-upravleniya-izmeneniyami-na-predpriyatiyah-it-sfery?ysclid=lwpa0tk9u868567112)
10. Управление изменениями: подробный гайд : научная статья – Skillbox Media – 2022. – (https://skillbox.ru/media/management/chto-takoe-upravlenie-izmeneniyami-i-kak-ono-pomozhet/?ysclid=lwpduexv4z670038579)
11. Управление качеством в IT-проектах: методы и стратегии : научная статья – Горбунов В.В. – 2022. – (https://apni.ru/article/6425-upravlenie-kachestvom-v-it-proektakh-metodi)
12. Методологии управления проектами: разбираем ключевые для IT- и digital-проектов : блог – Голава Алина – 2023. – (https://netology.ru/blog/08-2023-project-management-methodologies)
13. Методологии управления проектами: 12 популярных подходов : научная статья – Team Asana – 2024. – (https://asana.com/ru/resources/project-management-methodologies)
14. Prosci на практике - системное управление изменениями : блог компании – Тинькофф – 2024. – (https://habr.com/ru/companies/tinkoff/articles/803989/)
15. 33 полезных инструмента для веб-разработки : научная статья – Яндекс Образование – 2021. – (https://education.yandex.ru/journal/33-poleznykh-instrumenta-dlya-veb-razrabotki)

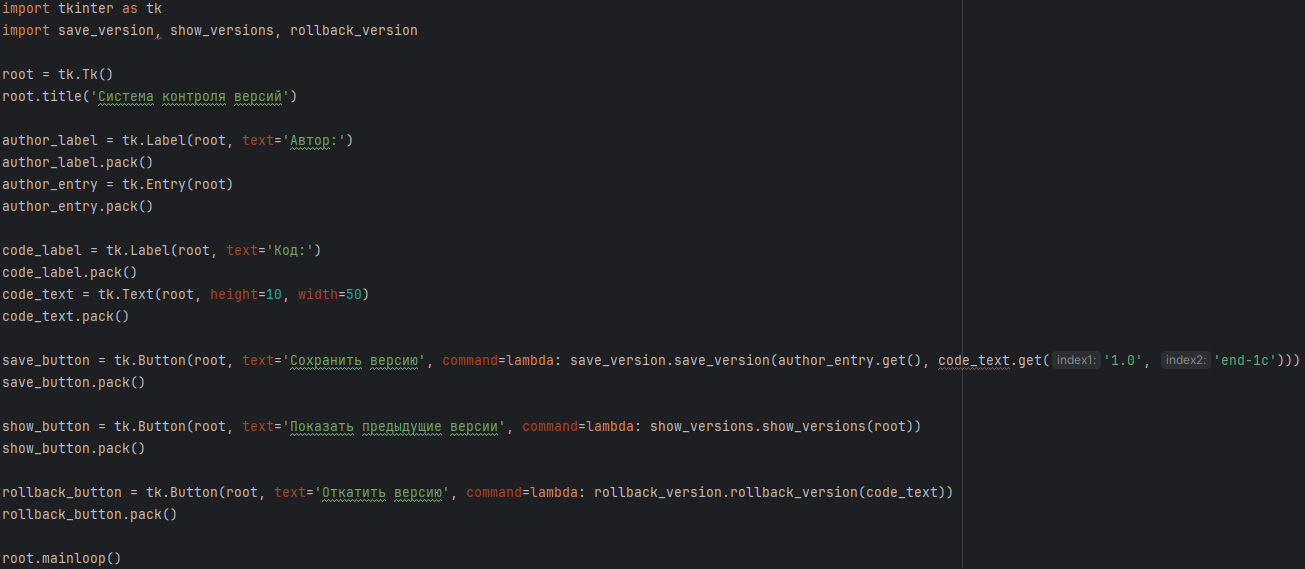
# ПРИЛОЖЕНИЯ



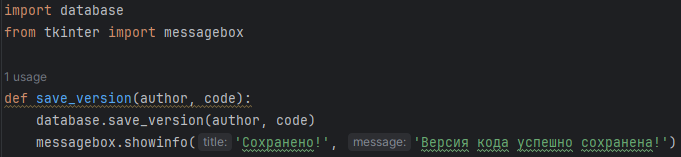
Приложение 1. Схема базы данных «save\_versions.db»



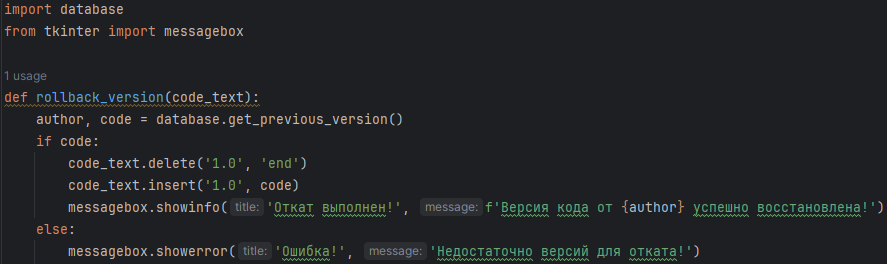
Приложение 2. Создание базы данных «save\_versions.db» с необходимой структурой



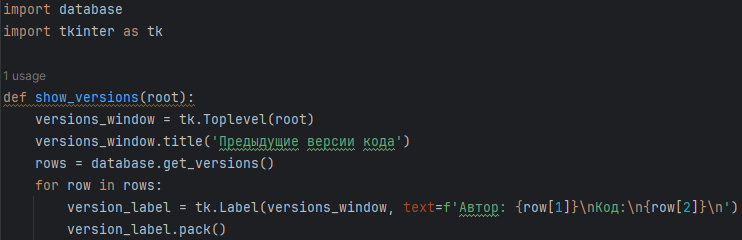
Приложение 3. Реализация графического интерфейса системы контроля версий



Приложение 4. Реализация функции сохранения текущей версии кода



Приложение 5. Реализация функции отката к предыдущей версии кода



Приложение 6. Реализация функции отображения предыдущих версий кода