



ОПИСАНИЕ ЖИЗНЕННОГО ЦИКЛА

**Программное средство интеллектуальной обработки информации
и ведения баз знаний «РЕД Центр Знаний»**

Листов 8

Москва

2026

ВВЕДЕНИЕ	3
1. КОМАНДЫ РАЗРАБОТКИ	3
1.1. Команда разработки «РЕД ВЕКТОР ДАННЫХ»	4
1.2. Команда разработки «ML ADAPTER» и «ML CHAT»	4
1.3. Команда разработки «РЕД ПЛАТФОРМА»	4
1.4. Команда «РЕД АРХИВ (WEB)»	4
1.5. Команда управления проектом и тестирования	4
1.6. Команда внедрения (DEVOPS)	5
2. МЕТОДОЛОГИЯ РАЗРАБОТКИ	5
2.1. Поэтапность	5
2.2. KANBAN ДОСКА ДЛЯ ЭТАПОВ	5
2.3. Задачи.....	5
2.4. Планирование.....	6
2.5. Фиксация версий	6
2.6. Замечания.....	6
3. ЭТАПЫ ЖИЗНЕННОГО ЦИКЛА РАЗВИТИЯ ПО	6
3.1. ЭТАПЫ 1-2: АНАЛИЗ ТРЕБОВАНИЙ/ТЕКУЩИХ ОШИБОК. ПРОЕКТИРОВАНИЕ НОВЫХ ЭТАПОВ РАЗРАБОТКИ	7
3.2. ЭТАП 3: РЕАЛИЗАЦИЯ / РАЗРАБОТКА	7
3.3. ЭТАП 4: ИНТЕГРАЦИЯ В ТЕСТОВЫЙ КОНТУР	7
3.4. ЭТАП 5: ТЕСТИРОВАНИЕ.....	7
3.5. ЭТАП 6: ФИКСАЦИЯ РЕЛИЗА	7
3.6. ЭТАП 7: ВНЕДРЕНИЕ	8
3.7. ЭТАП 8: ПОДДЕРЖКА	8
3.8. ЭТАП 9: ДОКУМЕНТИРОВАНИЕ	8
4. ПЛАНЫ ПО ИЗМЕНЕНИЮ ЖИЗНЕННОГО ЦИКЛА ПО	8

ВВЕДЕНИЕ

В настоящем документе представлено описание программного средства интеллектуальной обработки информации и ведения баз знаний «РЕД Центр Знаний» — высокопроизводительной платформы для организации корпоративных знаний и интеллектуального поиска информации, предназначенной для построения масштабируемых систем анализа документов в контролируемой инфраструктуре, состоящей из нескольких взаимодействующих сервисов.

Каждый сервис разрабатывается в рамках собственного технологического стека, адаптированного под специфику задач.

Активно разрабатываемые сервисы:

1. РЕД Вектор Данных

Веб-сервис, реализующий функции высокопроизводительной векторной базы данных и поискового движка.

Язык: Rust.

2. ML Adapter

Сервис автоматической загрузки и удаления документов из системы.

Технология: bytewax (Python).

3. ML Chat

Веб-сервис, реализующий функции RAG (Retrieval-Augmented Generation).

Язык: Python.

4. Ред Платформа

Основной сервис бизнес-логики проекта.

База: платформа «РЕД Платформа» (Java).

5. РЕД АРХИВ (web)

Интерфейс системы «РЕД Центр Знаний».

Технология: Vue.js.

1. КОМАНДЫ РАЗРАБОТКИ

Все сервисы разрабатываются в рамках единой методологии с использованием CI/CD-процессов. Каждый сервис имеет отдельную команду, но они тесно взаимодействуют через общие репозитории, документацию и еженедельные синхронизации.

1.1. КОМАНДА РАЗРАБОТКИ «РЕД ВЕКТОР ДАННЫХ»

Цели:

- Системная разработка.
- Оптимизация производительности векторного поиска.

1.2. КОМАНДА РАЗРАБОТКИ «ML ADAPTER» И «ML СНАТ»

Цели:

- Автоматизация загрузки документов из внешних источников.
- Доработка различных форматов данных: PDF, ODT.
- Оптимизация производительности загрузки.
- Интеграция с «Ред Платформой».
- Разработки и улучшение реализации RAG-функциональности.
- Интеграция с «РЕД АРХИВ» для пользовательского интерфейса.

1.3. КОМАНДА РАЗРАБОТКИ «РЕД ПЛАТФОРМА»

Цели:

- Разработка бизнес-логики через модули «Ред Платформа».
- Интеграция с остальными сервисами.

1.4. КОМАНДА «РЕД АРХИВ (WEB)»

Цели:

- Создание и доработка пользовательского интерфейса проекта.

1.5. КОМАНДА УПРАВЛЕНИЯ ПРОЕКТОМ И ТЕСТИРОВАНИЯ

Цели:

- Разработка направлений развития продукта.
- Разработка производственных задач.
- Отслеживание производственных задач и синхронизация всех команд разработки
- Тестирование нового или доработанного функционала.
- Организация встреч и различных организационных мероприятий, связанных с развитием продукта.
- Подготовка документации.

1.6. КОМАНДА ВНЕДРЕНИЯ (DEVOPS)

Цели:

- Развертывание тестового и продуктового контура
- Разработка образов дистрибутива
- Интеграция CI/CD.

2. МЕТОДОЛОГИЯ РАЗРАБОТКИ

2.1. ПОЭТАПНОСТЬ

Разработка платформы «РЕД Центр Знаний» строится вокруг гибкой поэтапной модели, где каждый этап завершается внедрением новых функциональных возможностей. Это позволяет сохранять стабильность ядра системы, одновременно обеспечивая постепенное развитие функционала.

Этапы отмечаются номерами: Этап 1, Этап 2,

Каждый этап документируется, в частности фиксируются цели и задачи этапа, а также некоторые детали реализации, требующие документирования.

2.2. КАНБАН ДОСКА ДЛЯ ЭТАПОВ

На каждом этапе используется Канбан-доска, которая визуализирует прогресс задач от «Backlog» до стадии «Готово». Такой подход помогает командам видеть приоритеты и текущее состояние работ в режиме реального времени.

В рамках данного этапа задачи разбиваются на группы по статусам:

- «Backlog» - отложенные задачи и задачи, не вошедшие в текущую разработку;
- «Todo» - планируемые к реализации задачи;
- «In Progress» - задачи в настоящий момент реализуемые;
- «Готово» - завершенные задачи.

В процессе реализации задачи статус задачи меняется с «Backlog» или с «Todo» до «Готово». Задачи, не реализованные в рамках этапа, отправляются на следующий этап.

2.3. ЗАДАЧИ

В Gitea каждая задача содержит четкое описание работ, указание сервиса для реализации, имя разработчика и заказчика. Это создает прозрачность в распределении ответственности и трекинге изменений. Например, задача может начинаться как "Добавить поддержку PDF в ML Adapter", а затем детализироваться до конкретных технических шагов.

Список задач на этапе не является жестко фиксированным. Практикуется живая

корректировка: если в процессе разработки возникает необходимость изменить требования или добавить новые задачи, это делается оперативно, особенно для задач в состоянии "Новый". Такая гибкость позволяет адаптироваться к изменениям в бизнес-требованиях и технических ограничениях, но только в рамках действующего или будущих этапов разработки.

2.4. ПЛАНИРОВАНИЕ

Планирование функциональных возможностей системы всегда выходит за рамки текущего этапа. Запланировано несколько этапов работ на 2-3 месяца (минимум) развития продукта, чтобы обеспечить плавный переход между фазами. Это позволяет избежать хаотичных изменений и сохраняет общий вектор развития системы.

2.5. ФИКСАЦИЯ ВЕРСИЙ

После завершения блока задач внедряется одна или несколько новых функциональных возможностей, ошибки при этом исправляются в приоритетном порядке и отправляются в дистрибутивную версию.

2.6. ЗАМЕЧАНИЯ

Методология тесно интегрирована с CI/CD-процессами.

3. ЭТАПЫ ЖИЗНЕННОГО ЦИКЛА РАЗВИТИЯ ПО

Жизненный цикл разработки и внедрения Программного средства интеллектуальной обработки информации и ведения баз знаний «РЕД Центр Знаний» включает следующие этапы:

- Анализ требований (и ошибок).
- Проектирование (новых этапов разработки).
- Реализация / разработка.
- Интеграция в тестовый контур.
- Тестирование.
- Фиксация релиза.
- Внедрение.
- Поддержка.
- Документирование.

3.1. ЭТАПЫ 1-2: АНАЛИЗ ТРЕБОВАНИЙ/ТЕКУЩИХ ОШИБОК. ПРОЕКТИРОВАНИЕ НОВЫХ ЭТАПОВ РАЗРАБОТКИ

Данный этап контролируется командой управления проектом.

В рамках этапов производится:

- Сбор требований к системе.
- Приоритезация задач через совместные сессии с командами.
- Документирование найденных проблем и фиксация их в задачах.
- Еженедельные демо-сессии с командами с проверкой завершенных задач на текущем этапе.
- Поэтапное планирование функциональных возможностей системы.
- Коррекция списка задач данного этапа, добавление, удаление, модификация задач в рамках текущего этапа или следующих.
- Проектирование следующих этапов разработки, так задачи из Backlog могут быть отправлены в следующий этап.

3.2. ЭТАП 3: РЕАЛИЗАЦИЯ / РАЗРАБОТКА

Данный этап контролируется командами разработок.

В рамках этапа производится: реализация задач, в некоторых случаях создание новых задач в текущем этапе реализации.

3.3. ЭТАП 4: ИНТЕГРАЦИЯ В ТЕСТОВЫЙ КОНТУР

Происходит по большей части автоматически в процессе завершения задач на разработку.

Реализуется в рамках CI/CD процессов.

3.4. ЭТАП 5: ТЕСТИРОВАНИЕ

Данный этап контролируется командой управления проектом.

Отдел управления проектами информируется системой посредством сети Интернет о завершении отдельных задач текущего этапа. Таким образом он активно включается в процесс тестирования нового функционала на тестовом контуре системы.

3.5. ЭТАП 6: ФИКСАЦИЯ РЕЛИЗА

Данный этап контролируется командами разработок.

Завершается созданием новых образов и обновление дистрибутива системы новыми docker-образами системы.

В настоящий момент производится вручную после завершения очередного этапа

разработки.

В перспективе планируется автоматизированный процесс фиксации релиза и дистрибутива системы.

3.6. ЭТАП 7: ВНЕДРЕНИЕ

В перспективе предполагается автоматизированная публикация нового дистрибутива системы.

Сейчас внедрение системы происходит силами команды внедрения (DevOps) на основе дистрибутива, полученного на этапе 6.

3.7. ЭТАП 8: ПОДДЕРЖКА

Поддержка системы происходит силами команды внедрения (DevOps) по заявкам заказчиков.

3.8. ЭТАП 9: ДОКУМЕНТИРОВАНИЕ

Данный этап контролируется командой управления проектом.

Документирование новых функциональных возможностей системы производится регулярно после завершения каждого нового этапа реализации.

4. ПЛАНЫ ПО ИЗМЕНЕНИЮ ЖИЗНЕННОГО ЦИКЛА ПО

В перспективе планируется модификации процессов жизненного цикла:

1. Автоматизация релизов.
2. Систематизация (автоматизация) поддержки, создание команды поддержки продукта.
3. Систематизация (автоматизация) сбора требований со стороны заказчиков к системе.