

2.2

Описание процессов жизненного цикла

Настоящий документ описывает процессы жизненного цикла программного обеспечения "Автоматизированная система управления компанией (АСК) ЛЕГАРУС", включая проектирование, разработку, тестирование, приобретение, поставку, эксплуатацию, документирование, обучение и квалификацию персонала, поддержку версий и доработку, устранение сбойных ситуаций.

Настоящий документ описывает процессы жизненного цикла программного обеспечения "Автоматизированная система управления компанией (АСК) ЛЕГАРУС", включая проектирование, разработку, тестирование, приобретение, поставку, эксплуатацию, документирование, обучение и квалификацию персонала, поддержку версий и доработку, устранение сбойных ситуаций.

1. Общие сведения

Все процессы организованы с целью обеспечения качества программного обеспечения, надежности его работы и эффективной поддержки пользователей. Информация о персонале :

Разработчики (персонал, осуществляющий совершенствование ПО): 2 человека
Системный администратор (персонал, настраивающий системное окружение): 2 человек
Специалисты технической поддержки (персонал, устраняющий сбойные ситуации): 5 человек

Фактический адрес размещения:

Адрес размещения инфраструктуры разработки: 115088, г. Москва, 2-я ул. Машиностроения, дом 25, стр.5
Адрес размещения разработчиков: 115088, г. Москва, 2-я ул. Машиностроения, дом 25, стр.5
Адрес размещения службы поддержки: 115088, г. Москва, 2-я ул. Машиностроения, дом 25, стр.5

2. Проектирование

Проектирование программного обеспечения включает следующие этапы:

Анализ требований: Изучение бизнес-задач и требований
Анализ существующей функциональности системы
Определение целей и задач нового функционала
Планирование архитектуры: Проектирование архитектурных решений на основе многоуровневой модульной архитектуры
Определение структуры данных и моделей
Проектирование API и интерфейсов взаимодействия
Планирование интеграций с внешними системами
Определение требований к производительности и масштабируемости

Проектирование базы данных: Проектирование схемы базы данных
Определение связей между сущностями Планирование индексов для
оптимизации запросов Проектирование миграций базы данных
Проектирование пользовательского интерфейса: Проектирование макетов
интерфейса Определение пользовательских сценариев Планирование
адаптивности интерфейса

3. Разработка программного обеспечения

3.1. Процесс разработки

Процесс разработки программного обеспечения включает следующие этапы:
Постановка задачи: Формулирование бизнес-задачи Описание требований к
функциональности Определение критериев приемки Планирование
архитектуры: Детальное проектирование решения Определение компонентов
и модулей Планирование этапов реализации Разработка: Реализация
функциональности согласно проекту Следование стандартам кодирования
Использование системы контроля версий (Git) Создание модульных
компонентов Код-ревью: Проверка кода другими разработчиками Проверка
соответствия стандартам кодирования Проверка архитектурных решений
Выявление потенциальных проблем и улучшений Тестирование: Модульное
тестирование Интеграционное тестирование Функциональное тестирование
Проверка соответствия требованиям Документирование: Обновление
технической документации Обновление руководства пользователя (при
необходимости) Обновление руководства администратора (при
необходимости) Документирование изменений в changelog Публикация:
Подготовка релиза Создание тега версии в системе контроля версий
Публикация новой версии

3.2. Технологический стек

Разработка ведется с использованием следующих технологий:
Backend: Python 3.10+, Django 5.2.7, Django REST Framework База данных:
PostgreSQL 14+ с расширением pgvector Кэширование и очереди: Redis 5.0+
Frontend: HTML5, CSS3, JavaScript (ES6+), HTMX, Bootstrap 5 Асинхронные
задачи: Celery 5.3.4 WebSocket: Django Channels Машинное обучение: PyTorch,
sentence-transformers

3.3. Система контроля версий

Разработка ведется с использованием системы контроля версий Git через
платформу Gitea. Характеристики системы хранения исходного кода:
Наименование: Gitea Тип: Система управления репозиториями Git
Расположение: Локальная сеть организации Доступность: Доступна только из
локальной сети организации Адрес нахождения системы: 115088, г. Москва,
2-я ул. Машиностроения, дом 25, стр. 5
Хранимые данные:

Исходный код программного обеспечения (файлы с расширениями .py, .js, .html, .css, .scss и другие) Файлы конфигурации (requirements.txt, package.json, gulpfile.js и другие) Файлы документации История изменений исходного кода Метаданные репозитория (ветки, теги, коммиты)

Безопасность:

Доступ к репозиторию ограничен локальной сетью организации Используется система аутентификации и авторизации Gitea LDAP-авторизация для централизованного управления учетными записями Двухфакторная аутентификация (2FA) для дополнительной защиты

Резервное копирование:

Резервное копирование репозитория выполняется ежедневно один раз в день Время выполнения резервного копирования: 04:00 по московскому времени Резервные копии хранятся в соответствии с политикой резервного копирования организации

3.4. Версионирование

Система использует семантическое версионирование в формате X.Y.Z, где:

X - версия API (изменяется при несовместимых изменениях API) Y - новый функционал (изменяется при добавлении нового функционала) Z - исправления (изменяется при исправлении ошибок)

Пример: версия 1.2.3 означает версию API 1, второй релиз с новым функционалом, третье исправление.

4. Тестирование

4.1. Виды тестирования

В процессе разработки выполняются следующие виды тестирования:

Модульное тестирование: Тестирование отдельных компонентов и функций Проверка корректности работы модулей в изоляции Использование фреймворка Django TestCase Интеграционное тестирование: Тестирование взаимодействия между компонентами Проверка работы сервисов и API Тестирование интеграций с внешними системами Функциональное тестирование: Тестирование пользовательских сценариев Проверка соответствия требованиям Тестирование бизнес-логики Регрессионное тестирование: Проверка работоспособности существующего функционала после изменений Автоматизированное тестирование критических сценариев

4.2. Процесс тестирования

Тестирование выполняется на всех этапах разработки:

Модульное тестирование выполняется разработчиками в процессе написания кода Интеграционное и функциональное тестирование выполняется перед код-ревью Финальное тестирование выполняется перед публикацией релиза

4.3. Автоматизация тестирования

Часть тестов автоматизирована и выполняется автоматически при коммитах в репозиторий и перед публикацией релиза.

5. Приобретение

5.1. Лицензирование

Программное обеспечение распространяется на основе лицензионного соглашения. Система лицензирования включает:

Пробный период: При первой установке предоставляется пробный период на 30 дней. В течение пробного периода система работает без ограничений. За 3 недели до истечения пробного периода система начинает показывать предупреждения. Активация лицензии: Активация выполняется через веб-интерфейс системной админки. Ввод лицензионного ключа, полученного от поставщика. Автоматическая проверка ключа на сервере лицензирования. Регистрация установки в системе лицензирования. Проверка лицензии: Ежедневная автоматическая проверка лицензии (в 3:00 утра). Проверка при каждом запросе через middleware. Обновление статуса лицензии в базе данных.

5.2. Получение лицензионного ключа

Лицензионные ключи распространяются отдельно от программного обеспечения через защищенные каналы связи (электронная почта, защищенный портал).

6. Поставка

6.1. Формы поставки

Программное обеспечение поставляется в следующих формах:

Исходный код: В виде архивов с исходным кодом (ZIP, TAR.GZ).
Установочные пакеты: Архивы с исходным кодом и инструкциями по установке.
Документация по установке и настройке.

6.2. Процесс поставки

Подготовка релиза с указанием версии Создание архива с исходным кодом
Подготовка документации по установке Передача заказчику через защищенные каналы связи

7. Эксплуатация

7.1. Установка и настройка

Процесс установки включает следующие этапы:

Подготовка инфраструктуры: Установка операционной системы (Linux)
Установка PostgreSQL, Redis, Python Настройка веб-сервера (Nginx) Настройка менеджера процессов (Supervisor) Установка приложения: Клонирование репозитория или распаковка архива Создание виртуального окружения Python Установка зависимостей Настройка конфигурации Инициализация базы данных: Создание базы данных и пользователя Применение миграций Загрузка начальных данных Создание суперпользователя Запуск сервисов: Настройка и запуск Gunicorn (WSGI сервер) Настройка и запуск Celery (фоновые задачи) Настройка и запуск Celery Beat (периодические задачи) Настройка Nginx

7.2. Эксплуатация системы

В процессе эксплуатации выполняются следующие операции:

Мониторинг: Мониторинг работы сервисов Мониторинг использования ресурсов Мониторинг логов на наличие ошибок Резервное копирование: Регулярное резервное копирование базы данных Резервное копирование медиа-файлов Хранение резервных копий в безопасном месте Обслуживание: Регулярная проверка работы системы Очистка старых данных Оптимизация базы данных

8. Документирование

8. Документирование

8.1. Виды документации

Система сопровождается следующими видами документации:

Руководство пользователя: Описание функциональности системы Инструкции по работе с модулями Описание пользовательского интерфейса Примеры использования Руководство администратора: Инструкции по установке и

настройке Описание системных настроек Инструкции по управлению пользователями Инструкции по резервному копированию и восстановлению Техническая документация: Описание технической архитектуры Описание API Описание базы данных Инструкции для разработчиков Документация по лицензированию: Описание системы лицензирования Инструкции по активации лицензии Описание процесса обновления

8.2. Процесс документирования

Документация создается и обновляется в процессе разработки:
Документация обновляется при добавлении нового функционала
Документация проверяется при выпуске новых версий Документация хранится в репозитории

8.3. Форматы документации

Документация предоставляется в следующих форматах:
Markdown файлы в репозитории HTML версии документации PDF версии

9. Обучение и квалификация персонала

9. Обучение и квалификация персонала

9.1. Обучение пользователей

Для пользователей системы предоставляется:
Руководство пользователя: Подробное описание всех функций системы
Пошаговые инструкции Примеры использования Обучающие материалы:
Видео-инструкции (при наличии) Демонстрационные данные Часто задаваемые вопросы (FAQ) Техническая поддержка: Консультации по использованию системы Помощь в решении проблем Обучение работе с новым функционалом

9.2. Обучение администраторов

Для администраторов системы предоставляется:
Руководство администратора: Подробные инструкции по установке и настройке
Описание системных настроек Инструкции по обслуживанию
Техническая поддержка: Консультации по настройке и администрированию
Помощь в решении технических проблем Обучение работе с новыми версиями

9.3. Квалификация персонала организации

Разработчики:

Высшее техническое образование в области информационных технологий
Опыт разработки на Python и Django Знание современных технологий веб-разработки Постоянное повышение квалификации

Системный администратор:

Высшее техническое образование в области информационных технологий
Опыт настройки и администрирования Linux-систем Знание систем виртуализации и контейнеризации Опыт настройки веб-серверов (Nginx, Apache) Знание систем управления процессами (Supervisor, systemd) Опыт настройки баз данных (PostgreSQL) Знание систем кэширования и очередей (Redis) Опыт настройки систем контроля версий (Git, Gitea) Постоянное повышение квалификации

Специалисты технической поддержки:

Техническое образование в области информационных технологий Опыт работы с Linux-системами Знание PostgreSQL, Redis, веб-серверов Опыт администрирования веб-приложений Постоянное обучение новым версиям системы

10. Поддержка версий и доработка

10. Поддержка версий и доработка

10.1. Процесс обновления системы

Процесс обновления системы включает следующие этапы:

Создание резервной копии: Резервная копия базы данных Резервная копия медиа-файлов Резервная копия конфигурационных файлов Остановка сервисов: Остановка всех сервисов через Supervisor Обновление кода: Получение новой версии из репозитория или архива Обновление зависимостей Python Применение миграций: Применение миграций базы данных Загрузка новых фикстур (при необходимости) Сбор статических файлов: Сбор статических файлов для веб-интерфейса Перезапуск сервисов: Запуск всех сервисов Проверка работоспособности Проверка работы: Проверка доступности сайта Проверка работы очередей задач Проверка логов на наличие ошибок

10.2. Откат к предыдущей версии

В случае возникновения проблем после обновления возможен откат к предыдущей версии:

Остановка сервисов Восстановление кода из резервной копии Восстановление базы данных из резервной копии Перезапуск сервисов

10.3. Доработка функциональности

Доработка функциональности выполняется в рамках процесса разработки:
Постановка задачи на доработку Планирование архитектуры решения
Разработка функциональности Код-ревью Тестирование Документирование
изменений Публикация новой версии

10.4. Управление версиями

Управление версиями осуществляется через систему контроля версий Git:
Каждая версия помечается тегом в формате X.Y.Z Ведется история изменений
(changelog) Описание изменений в каждой версии

11. Устранение сбойных ситуаций

11. Устранение сбойных ситуаций

11.1. Организация технической поддержки

Техническая поддержка осуществляется специалистами организации по адресу: 115088, г. Москва, 2-я ул. Машиностроения, дом 25, стр.5. Состав службы технической поддержки:

5 специалистов технической поддержки Рабочее время: с 9:00 до 18:00 по московскому времени (понедельник - пятница) Возможность работы во внерабочее время при критических ситуациях

Система управления техподдержкой: Для организации работы технической поддержки в организации КИВИТ используется система АСК ЛЕГАРУС (данная система). В системе регистрируются все обращения пользователей, ведется учет заявок, контролируются сроки выполнения, осуществляется коммуникация с пользователями и документирование решений.

11.2. Каналы связи с технической поддержкой

Техническая поддержка доступна через следующие каналы:

Электронная почта: Отправка обращений на адрес технической поддержки
Получение ответов и инструкций Телефонная связь: Многоканальные номера для связи Консультации по телефону Система заявок (АСК ЛЕГАРУС):
Регистрация обращений в системе техподдержки АСК ЛЕГАРУС Отслеживание статуса обращения История обращений Классификация заявок с помощью машинного обучения Автоматическое назначение исполнителей Контроль сроков выполнения по SLA

11.3. Классификация обращений

Обращения классифицируются по следующим типам:

Инцидент: Сбой в работе системы Неисправность функциональности Проблемы с доступом Запрос: Консультация по использованию Изменение настроек Установка стандартного ПО Проблема: Повторяющийся инцидент Проблема без готового решения

11.4. Процесс устранения сбойных ситуаций

Процесс устранения сбойных ситуаций включает следующие этапы:

Регистрация обращения: Регистрация обращения в системе техподдержки
Классификация типа обращения Определение приоритета Диагностика:
Анализ проблемы Сбор информации о сбое Изучение логов системы Решение проблемы: Разработка решения Применение исправлений Тестирование решения
Закрытие обращения: Проверка решения заказчиком
Документирование решения Закрытие обращения

11.5. Сроки реагирования (SLA)

Сроки реагирования на обращения определяются приоритетом:
Приоритет Время реакции (не более) Время выполнения (не более)
Критичный 30 минут 2 часа
Высокий 1 час 8 часов
Средний 2 часа 24 часов
Низкий 8 часов 48 часов

11.6. Мониторинг и профилактика

Для предотвращения сбойных ситуаций выполняются:

Мониторинг системы: Мониторинг работы сервисов Мониторинг использования ресурсов Мониторинг логов на наличие ошибок
Профилактические работы: Регулярное резервное копирование Обновление системы безопасности Оптимизация производительности
Регламентные работы: Установка обновлений безопасности Обновление прошивок сетевых устройств Проверка логов на наличие ошибок Проверка целостности данных

11.7. Документирование решений

Все решения проблем документируются:

Описание проблемы Причина возникновения Способ решения Рекомендации по предотвращению

Документация решений используется для:

Быстрого решения повторяющихся проблем Обучения специалистов Улучшения системы

