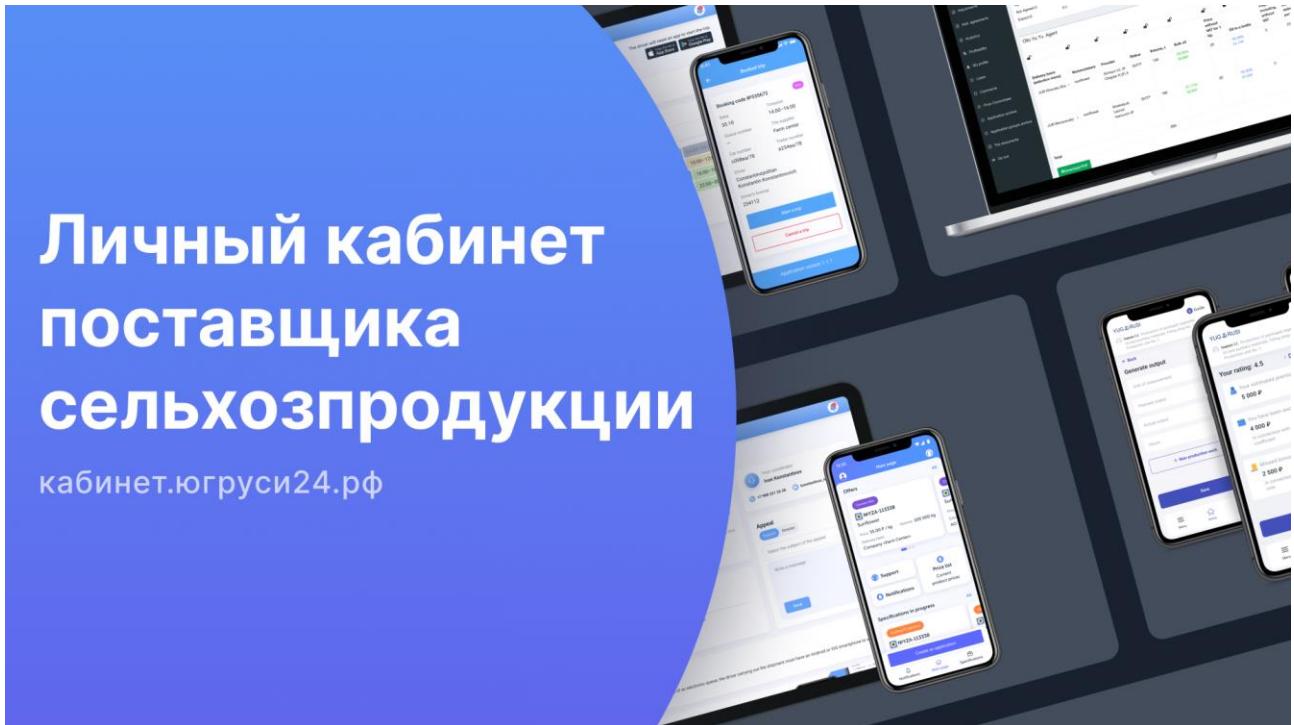


Программное обеспечение «Личный кабинет поставщика сельхозпродукции»

**Документация, содержащая описание
процессов, обеспечивающих поддержание
жизненного цикла, в том числе устранение
неисправностей и совершенствование, а также
информацию о персонале, необходимом для
обеспечения такой поддержки**

**Личный кабинет
поставщика
сельхозпродукции**

кабинет.югруси24.рф



Содержание

1. Введение.....	3
2. Сведения о команде разработки и поддержки.....	4
3. Жизненный цикл разработки и поддержки ПО «Личный кабинет поставщика сельхозпродукции».....	6

1. Введение

Настоящий документ описывает процессы, обеспечивающие поддержание полного жизненного цикла программного обеспечения (ПО) «Личный кабинет поставщика сельхозпродукции» (далее – ЛК поставщика с/х продукции, Система).

Цель документа – регламентировать деятельность по проектированию, разработке, тестированию, эксплуатации, поддержке и совершенствованию Системы, а также определить ответственных сотрудников и порядок их взаимодействия.

Документ предназначен для руководителя проектов, менеджера, аналитиков, разработчиков, тестировщиков, сотрудников технической поддержки и других специалистов, задействованных в жизненном цикле Системы.

Про веб-приложение «Личный кабинет поставщика сельхозпродукции»

Личный кабинет поставщика сельхозпродукции – это веб-приложение, которое обеспечивает автоматизацию полного цикла закупки с/х продукции.

Адрес: <http://кабинет.югруси24.рф>

Фактический адрес размещения инфраструктуры разработки:

Россия, Ростовская область, г. Ростов-на-Дону, ул. 1-я Луговая, 12а.

Фактический адрес размещения разработчиков:

Россия, Ростовская область, г. Ростов-на-Дону, ул. 1-я Луговая, 12а.

Фактический адрес размещения службы поддержки:

Россия, Ростовская область, г. Ростов-на-Дону, ул. 1-я Луговая, 12а.

ЛК поставщика с/х продукции решает следующие бизнес-проблемы:

- обеспечение прозрачности взаимодействия с поставщиком;
- автоматизация процесса подписания договоров и создания спецификаций;
- механизм рыночного определения цены на продукцию (ценовой комитет);
- упорядочение процесса отгрузки продукции посредством использования электронной очереди;
- механизм поиска объемов с/х продукции по наиболее выгодной цене и наиболее релевантных по доступности доставки (карта объемов);
- контроль ритмичности поставок с/х продукции (график завоза).

Функционал приложения:

1. Регистрация поставщика.
2. Заполнение основных данных.
3. Аккредитация.
4. Автоматическое формирование договора.
5. Подписание договора через ЭДО.
6. Подача заявки на поставку продукции по прайсу или по «своей цене».
7. Согласование заявки на поставку «по своей цене» Ценовым комитетом.
8. Создание спецификации.
9. Бронирование временного интервала в электронной очереди
10. Просмотр отгрузок.

2. Сведения о команде разработки и поддержки

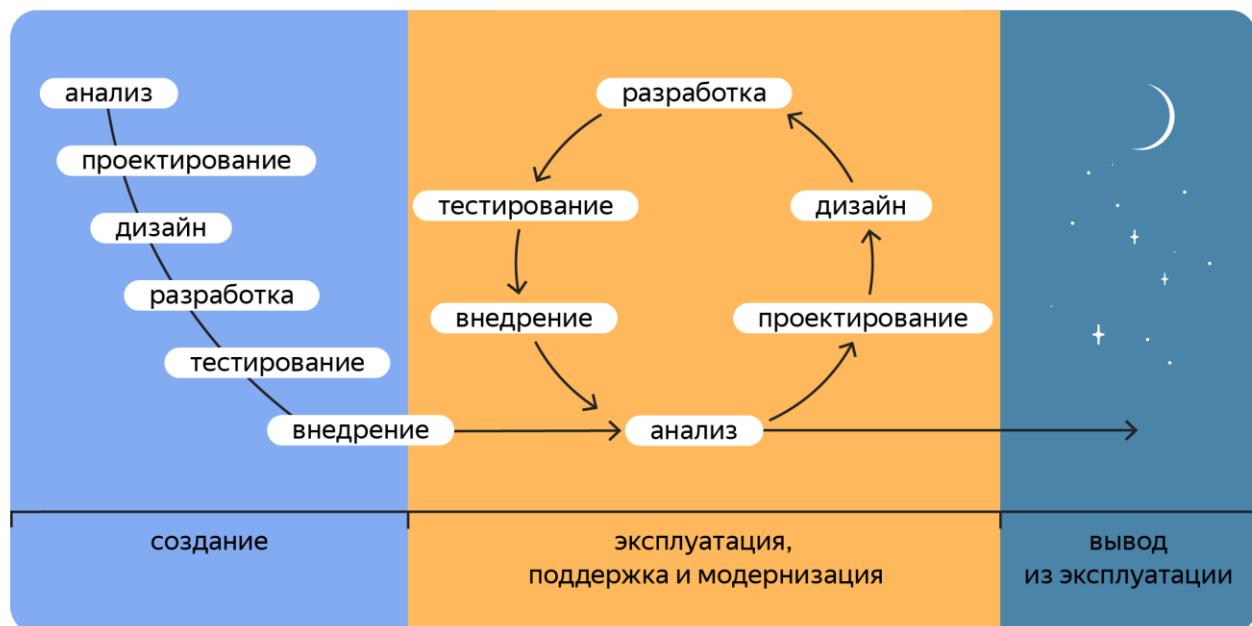
Разработку и поддержку ПО «Личный кабинет поставщика сельхозпродукции» осуществляет команда, состоящая из

сотрудников ООО «А-Софт». Основные роли и их обязанности приведены в таблице.

Роль пользователя	Функционал
Руководитель проекта (1 человек)	Определение общего видения продукта. Ответственность за распределение и контроль финансовых ресурсов проекта. Утверждение основных технических и архитектурных решений, влияющих на бюджет, сроки или стратегию развития продукта. Разрешение конфликтных ситуаций, эскалированных менеджером проекта. Взаимодействие со стейкхолдерами высшего уровня. Создание условий для эффективной работы команды, включая решение организационных и административных вопросов.
Менеджер проекта (1 человек)	Управление бэклогом задач, приоритизация задач, взаимодействие с бизнес-заказчиками, контроль сроков и бюджета, приемка работ.
Системный архитектор / Ведущий разработчик / Лид команды разработки (1 человек)	Проектирование архитектуры (микросервисы, API), принятие технических решений, код-ревью, контроль за техническим качеством проекта.
Разработчик (Backend, Frontend, Mobile) (4 человека)	Реализация нового функционала, исправление ошибок, рефакторинг, написание unit-тестов, поддержка серверной части (Node.js/NestJS), клиентской части (Vue 3, Flutter) и мобильных приложений.
Тестировщик (QA-инженер) (1 человек)	Разработка тест-планов и тест-кейсов, проведение функционального, интеграционного, регрессионного тестирования, документирование результатов
Аналитик (1 человек)	Детальный сбор, анализ и формализация бизнес-требований от заказчиков и пользователей. Разработка ТЗ, пользовательских сценариев (User Stories), нефункциональных требований. Проектирование бизнес-процессов и их оптимизация. Подготовка материалов для разработчиков и тестировщиков.

Роль пользователя	Функционал
UX/UI-дизайнер (1 человек)	Создание дизайн-макетов пользовательских интерфейсов
Специалист I линии поддержки (2 человека)	Первичный прием и обработка обращений пользователей, консультация, решение типовых проблем, сбор информации для эскалации, ведение базы знаний (Типовые вопросы и ответы).
DevOps-инженер (1 человек)	Поддержка инфраструктуры (серверы, сети), развертывание и мониторинг микросервисов, обеспечение отказоустойчивости и безопасности периметра

3. Жизненный цикл разработки и поддержки ПО «Личный кабинет поставщика сельхозпродукции»



3.1. Анализ и проектирование

- Источники запросов:** Обратная связь от пользователей (через техподдержку), инициативы бизнес-заказчиков (Дирекция по закупкам и снабжению), результаты анализа метрик (Яндекс-метрика).

- **Процесс:**

1. **Сбор требований:** Аналитик собирает и формализует бизнес-требования и пользовательские требования. Результат – пользовательские требования в формате User Stories, диаграммы вариантов использования (Use Cases).
2. **Проектирование архитектуры:** Системный архитектор проектирует изменения в архитектуре и интеграции). Результат – схемы архитектуры (C4-model, диаграммы контейнеров и компонентов). Системный аналитик составляет спецификацию требований к ПО, включающую описание изменения в модели данных (ER-диаграммы), описание требования к пользовательским интерфейсам, включая прототипы пользовательских интерфейсов, документацию API, диаграммы последовательности.
3. **Согласование:** Утверждение технических требований у бизнес-заказчика.)

3.2. Дизайн

- **Результат:** На основе пользовательских и функциональных требований, прототипов аналитика составляются дизайномакеты.

3.3. Разработка ПО

- **Методология:** Применяется гибкая методология разработки (Agile/Scrum/Kanban). Разработка ведется на основе разработанной технической документации
- **Стандарты:**
 - Использование фреймворков: NestJS (Backend), Vue 3 (Web Frontend), Flutter (Mobile).

- Следование принципам микросервисной архитектуры.
- Использование REST API и брокера сообщений (NATS) для межсервисного взаимодействия.
- Ведение программного кода в репозитории с обязательным код-ревью.
- **Интеграция:** Разработчики обеспечивают корректную передачу данных через API со всеми внешними системами (1С, Диадок, Dadata и др.), согласно утвержденным контрактам.

3.4. Тестирование

- **Этапы:**
 1. **Внутреннее закрытое тестирование:** Проводится силами QA-инженеров и разработчиков без участия пользователей.
 2. **Тестирование с участием пользователей:** Проводится совместно с представителями поставщиков, координаторами, агентами для проверки юзабилити и соответствия бизнес-процессам.
- **Процесс:** Все этапы тестирования документируются. Результаты фиксируются в акте о результатах тестирования. Только после успешного тестирования и подписания акта изменения переносятся в продуктивную среду.

3.5. Внедрение (поставка и развертывание)

- **Процесс:** Развертывание осуществляется в действующей инфраструктуре ООО «МЭЗ Юг Руси». Используется контейнеризация для обеспечения консистентности между средами (разработки, тестирования, продуктивной).
- **Ответственные:** DevOps-инженеры осуществляют деплой обновлений с минимальным временем простоя.

3.6. Эксплуатация

- **Инфраструктура:** Система работает на серверах ООО «МЭЗ Юг Руси». Данные хранятся в PostgreSQL. Используется файловое хранилище MinIO (S3). Для логирования и мониторинга задействован ELK-стек.
- **Мониторинг:** Ведутся логи работы всех компонентов системы. Осуществляется мониторинг доступности сервисов и производительности.
- **Резервное копирование:** Проводится не реже чем каждые 24 часа для всех баз данных. Срок хранения данных о поставщиках не ограничен.

3.7. Документирование

- **Перечень документации:**
 - **Техническое задание (ТЗ):** Создается в соответствии с ГОСТ 34.602-89.
 - **Проектное решение/Спецификация требований (SRS):** Детальное описание реализованного функционала и интерфейсов.
 - **Схемы архитектуры:** Диаграммы C4 (Контекст, Контейнеры, Компоненты).
 - **Инструкции для пользователей:** Руководства по эксплуатации для различных ролей.
 - **Инструкции для техподдержки:** Содержат сценарии диалогов, типовые вопросы и ответы, минимальный набор информации для эскалации обращения на II линию.
 - **Описание API:** Используется Swagger или аналогичные инструменты.
 - **Программная документация:** Описание кода, принципов работы модулей, версий использованного ПО.

- **Ответственные:** Разработчики, архитекторы, аналитики и менеджер проекта несут ответственность за актуальность документации.

3.8. Обучение и квалификация персонала

- **Техническая поддержка:** Специалисты | проходят регулярное обучение по новому функционалу Системы. В их распоряжении находится база знаний с типовыми вопросами и ответами, а также инструкции.
- **Разработчики и тестировщики:** Ответственны за поддержание своей квалификации в области используемых технологий.
- **Пользователи:** Обучение пользователей (поставщиков, агентов) проводится через раздел "Обучение" в ЛК, а также с помощью инструкций и консультаций техподдержки.

3.8. Поддержка версий и доработка

- **Управление версиями:** Используется система контроля версий (Git). Версии ПО присваиваются в соответствии с правилами семантического версионирования.
- **Доработка:** Процесс доработки инициируется через менеджера проекта, проходит стадии анализа и проектирования, дизайна, разработки, тестирования и внедрения, описанные выше. Исправления ошибок (баг-файлы) приоритизируются и выпускаются вне регулярного цикла релизов при необходимости.

3.9. Устранение сбойных ситуаций (Инцидентов)

- **Каналы поступления:** Обращения поступают через ITSM (Naumen), почту (b2b_support@yugrusi.ru), горячую линию, чаты поддержки.
- **Процесс обработки:**
 1. **1 линия поддержки:** Принимает обращение, идентифицирует проблему, используя "Сценарий диалога". Решает типовые вопросы. Если проблема не

решена, собирает "Минимальный набор информации" (ИНН, номер заявки/договора, скриншоты, текст ошибки) и эскалирует на II линию.

Телефон горячей линии: тел. 8-800-500-27-79 (доб. 4).

2. **II линия поддержки:** Анализирует проблему, проверяет логи (ELK), воспроизводит ошибку. Если проблема является программной ошибкой, создает задачу для разработчиков таск-трекере.
 3. **Разработчики:** Исправляют ошибку. Исправление проходит стадию тестирования.
 4. **Поставка исправления:** Прошедшее тестирование исправление развертывается в продуктивной среде.
- **SLA:** Время на решение проблемы – 1 рабочий день (8 рабочих часов).
 - **Эскалация бизнес-вопросов:** Вопросы, не связанные с технической неисправностью (напр., условия договора), перенаправляются в Дирекцию по закупкам и снабжению (тел. 8-800-500-27-79, доб. 5).

Прием запросов по техподдержке и устранению сбойных ситуаций осуществляют специалисты I линии поддержки (2 человека).

Режим работы специалистов техподдержки:

с понедельника по пятницу включительно, с 08.00 до 16.30, время московское, суббота и воскресенье - выходные дни.