Общество с ограниченной ответственностью «ПЛАЙНЕР»

Программа для ЭВМ

«Плайнер» (Pliner)

Система автоматизированного планирования производства

ОПИСАНИЕ ПРОЦЕССОВ ЖИЗНЕННОГО ЦИКЛА

Версия 1.0.0

г. Белгород

2026

1. Общие сведения

Документ описывает, как ООО «ПЛАЙНЕР» сопровождает программу «Плайнер» (Pliner) версии 1.0.0: устраняет неисправности, развивает функциональность, оказывает техническую поддержку, и какой персонал это обеспечивает. Документ подготовлен для включения ПО в единый реестр российских программ для ЭВМ и БД.

Таблица 1. Сведения о программе

Параметр	Значение
Наименование ПО	«Плайнер» (Pliner) — система автоматизированного планирования производства
Правообладатель	ООО «ПЛАЙНЕР», ОГРН 1263100002406, ИНН 3100054566
Адрес	308024, г. Белгород, ул. Мокроусова, д. 19, пом. 1
Версия	1.0.0
Тип поставки	серверное веб-приложение в Docker-контейнерах, установка на оборудовании заказчика или в облаке
Класс ПО	08.19 «Программы производственного планирования (APS)»; 09.02 «Средства управления производственными процессами (MES)»

Исходный код программы хранится во внутренней системе управления версиями Git (GitLab) правообладателя и разделён на пять репозиториев:

Таблица 2. Репозитории исходного кода

Репозиторий	Компонент	Технологии
management_api	сервер приложений: бизнес-логика, API, работа с БД, авторизация	Python 3.12, FastAPI, SQLAlchemy, Alembic
plancalc	ядро планирования (солвер)	C++20, Python 3.12, pybind11
shiftplan	сервис сменных заданий и плана материалов	Python 3.12, FastAPI
pliner-front	веб-интерфейс	TypeScript, React 19, Vite, Ant Design
landing	публичный сайт продукта	TypeScript, Next.js

Все компоненты разработаны правообладателем; сторонние библиотеки — свободное ПО с открытыми лицензиями (см. «Описание функциональных характеристик», раздел 5).

2. Устранение неисправностей

Неисправность — любое отклонение работы ПО от описания функциональных характеристик или эксплуатационной документации. Неисправности устраняются силами сотрудников правообладателя в рамках технической поддержки (раздел 4).

2.1. Источники выявления

- обращения пользователей в службу технической поддержки;
- автоматические тесты при сборке (модульные, интеграционные и эталонные тесты ядра планирования);
- журналы приложения: сервер приложений ведёт ежедневно ротируемые журналы со сроком хранения 30 дней (настраивается);
- проверки работоспособности контейнеров (healthcheck) и автоматический перезапуск сервисов при сбое;
- встроенная проверка технологических карт и входных данных расчёта;
- самостоятельное тестирование и ревью кода сотрудниками правообладателя.

2.2. Классификация по критичности

Таблица 3. Уровни критичности

Уровень	Описание	Примеры
1 — критический	ПО недоступно или основная функция не работает у всех пользователей; обходного пути нет	не запускается сервер приложений, невозможен вход, расчёт плана не завершается, потеря данных
2 — высокий	Не работает отдельный модуль или функция; обходной путь сложен или отсутствует	не формируются сменные задания, ошибка импорта из Excel, неверный результат расчёта
3 — средний	Функция работает с ограничениями; есть обходной путь	некорректное отображение диаграммы, ошибка в отчёте или фильтре
4 — низкий	Косметические дефекты, вопросы и консультации	опечатки, выравнивание элементов, вопросы по работе с ПО

2.3. Порядок устранения

1. **Регистрация.** Обращение регистрируется в системе учёта задач с присвоением номера; заявителю сообщается номер обращения.
2. **Классификация.** Сотрудник первой линии поддержки присваивает уровень критичности и ответственного исполнителя.
3. **Воспроизведение и диагностика.** Ошибка воспроизводится на тестовом стенде по данным заявителя (с его согласия — по резервной копии БД), анализируются журналы сервисов.
4. **Временное решение.** Для уровней 1–2 заявителю предоставляется обходной путь или выполняется откат на предыдущую стабильную версию контейнера.
5. **Исправление.** Изменение вносится в отдельной ветке Git; к исправлению добавляется автоматический тест, воспроизводящий ошибку.

6. **Проверка.** Исправление проходит ревью кода (merge request), автоматические тесты в конвейере CI и регрессионную проверку на тестовом стенде.
7. **Выпуск.** Исправление выпускается как корректирующая версия (увеличение третьего числа номера, например 1.0.1) и устанавливается обновлением Docker-образов; структура БД обновляется миграциями Alembic без потери данных.
8. **Закрытие.** Заявитель подтверждает устранение; обращение закрывается, изменение фиксируется в перечне изменений версии.

Перед установкой обновления выполняется резервное копирование БД (pg_dump); при неудаче система возвращается к предыдущей версии образов и копии БД.

3. Развитие ПО

Правообладатель постоянно развивает ПО собственными силами: улучшает алгоритмы ядра планирования, добавляет модули и интеграции, повышает производительность и безопасность.

3.1. Источники требований

- предложения пользователей, поступившие через службу технической поддержки;
- требования заказчиков, выявленные при внедрении на предприятиях;
- изменения законодательства РФ (в том числе Трудового кодекса в части сверхурочных работ и режима отдыха);
- анализ качества и производительности расчётов ядра планирования;
- обновления сторонних компонентов и устранение уязвимостей;
- действия по поддержке отечественных ОС и СУБД (Astra Linux, РЕД ОС, Postgres Pro).

3.2. Этапы разработки

1. **Анализ и планирование.** Ведущий разработчик оценивает требования, присваивает приоритет и включает их в план выпуска версии.
2. **Проектирование.** Для значимых изменений готовится техническое решение: изменения API, структуры БД, алгоритма ядра.
3. **Разработка.** Каждая задача выполняется в отдельной ветке Git; изменения БД оформляются миграциями Alembic; код проверяется статическими анализаторами (Ruff для Python, ESLint и TypeScript для веб-интерфейса).
4. **Ревью кода.** Изменения вливаются в основную ветку через merge request в GitLab после проверки вторым разработчиком.
5. **Тестирование.** Конвейер GitLab CI запускает автоматические тесты (pytest для серверных сервисов, эталонные тесты ядра планирования на фиксированных наборах данных), проверку на утечку секретов (gitleaks) и уязвимости зависимостей (Trivy). Затем версия проверяется на демонстрационном стенде.

6. **Выпуск.** Конвейер CI собирает Docker-образы и разворачивает их в целевом окружении; версия помечается тегом в Git, пользователям передаётся перечень изменений.

3.3. Версионирование и периодичность

Номер версии имеет вид X.Y.Z (семантическое версионирование).

Таблица 4. Типы выпусков

Тип выпуска	Номер	Содержание	Периодичность
Корректирующий	Z (1.0.1)	исправление неисправностей и уязвимостей	по мере готовности; критические — вне очереди
Минорный	Y (1.1.0)	новые функции без нарушения совместимости	не реже одного раза в квартал
Мажорный	X (2.0.0)	существенные изменения архитектуры или API	по плану развития

Обновления передаются пользователям в виде Docker-образов с инструкцией по установке; миграции БД применяются автоматически и сохраняют данные пользователя.

3.4. Хранение исходного кода

Исходный код и история изменений хранятся на сервере GitLab, находящемся под управлением правообладателя на территории РФ. Доступ к репозиториям имеют только сотрудники правообладателя. Параллельно ведутся локальные копии репозиториев у разработчиков.

4. Техническая поддержка

Техническую поддержку пользователей оказывает ООО «ПЛАЙНЕР» собственными силами, без привлечения сторонних организаций. Поддержка ведётся на русском языке.

4.1. Состав услуг

- консультации по установке, настройке и работе с ПО;
- приём и обработка сообщений об ошибках, устранение неисправностей (раздел 2);
- предоставление обновлений и помощь в их установке;
- помощь в начальном заполнении справочников и импорте данных из Excel;
- приём предложений по развитию ПО (раздел 3).

4.2. Каналы обращения

Таблица 5. Каналы обращения

Канал	Адрес	Для чего
Электронная почта	info@pliner.pro	основной канал; любые обращения, круглосуточный приём
Телефон	+7 919 228-57-45	критические и срочные обращения в рабочее время
Telegram	@ms_sergeevich (t.me/ms_sergeevich)	оперативные вопросы и консультации
Форма на сайте	pliner.pro	заявки на демонстрацию и общие вопросы

В обращении указываются: организация и контактное лицо, версия ПО, описание проблемы и шаги для воспроизведения, скриншоты или текст ошибки, при необходимости — журналы сервисов.

4.3. Режим работы

Понедельник — пятница, с 9:00 до 18:00 по московскому времени, кроме нерабочих праздничных дней РФ. Обращения по электронной почте принимаются круглосуточно; поступившие вне рабочего времени регистрируются в начале следующего рабочего дня. Иные режимы (в том числе 24×7) устанавливаются договором с заказчиком.

4.4. Сроки реакции и решения

Сроки считаются в рабочих часах с момента регистрации обращения. Реакция — первый ответ специалиста с подтверждением принятия в работу; решение — устранение неисправности или предоставление обходного пути.

Таблица 6. Сроки реакции и решения

Уровень критичности	Время реакции	Обходной путь	Исправление
1 — критический	до 2 часов	до 8 часов	до 3 рабочих дней
2 — высокий	до 4 часов	до 2 рабочих дней	до 10 рабочих дней
3 — средний	до 1 рабочего дня	—	в ближайшем плановом выпуске
4 — низкий, консультация	до 2 рабочих дней	—	по плану развития; консультация — до 3 рабочих дней

Срок исправления приостанавливается, пока сотрудники правообладателя ожидают от заявителя данные или доступ, нужные для диагностики. Иные сроки могут быть установлены договором с заказчиком.

4.5. Порядок обработки обращений

Обращения обрабатываются на двух линиях:

- **Первая линия** — разработчик веб-интерфейса: регистрирует обращение, классифицирует его, консультирует пользователя, решает вопросы настройки и данных.
- **Вторая линия** — разработчик профильного компонента (серверная часть и ядро планирования или веб-интерфейс): диагностирует и исправляет ошибки в коде по процессу раздела 2.3.

Заявитель получает уведомления при регистрации, смене статуса и закрытии обращения. При необходимости и с согласия заказчика специалисты подключаются к серверу удалённо или выезжают на площадку.

5. Персонал

Разработку, сопровождение и техническую поддержку ПО обеспечивают 2 сотрудника ООО «ПЛАЙНЕР». Оба — граждане Российской Федерации, находятся на территории РФ и работают по трудовым или гражданско-правовым договорам с правообладателем. Права на созданный код принадлежат правообладателю.

Таблица 7. Персонал

Роль	Численность, чел.	Обязанности	Квалификация
Ведущий разработчик (ядро планирования и серверная часть)	1	план развития и архитектура; солвер, сервер приложений, сервис сменных заданий, БД; конвейер CI/CD, установка и обновление у заказчиков, резервное копирование; ревью кода и выпуск версий; вторая линия поддержки	высшее техническое или математическое образование; C++20, Python, FastAPI, PostgreSQL, RabbitMQ; алгоритмы оптимизации и расписаний; Linux (в т.ч. Astra Linux, РЕД ОС), Docker, GitLab CI
Разработчик веб-интерфейса	1	веб-интерфейс и сайт продукта; первая линия поддержки: приём обращений, консультации, обучение пользователей; вторая линия по веб-интерфейсу	высшее или среднее специальное техническое образование; TypeScript, React, Next.js; знание функциональности ПО «Плайнер» и процессов производственного планирования

Обязанности взаимозаменяемы: оба разработчика ведут код в единых репозиториях, проверяют изменения друг друга и могут принять обращение по любому компоненту. Архитектура и процессы описаны во внутренней документации репозиториев. При росте

числа заказчиков правообладатель расширяет штат: в первую очередь — специалистами первой линии поддержки и внедрения.

6. Контактная информация

Таблица 8. Контакты службы поддержки

Параметр	**Значение**
Организация	ООО «ПЛАЙНЕР», ИНН 3100054566, КПП 310001001
Фактический адрес службы поддержки	308024, г. Белгород, ул. Мокроусова, д. 19, пом. 1
Электронная почта поддержки	info@pliner.pro
Телефон	+7 919 228-57-45
Telegram	@ms_sergeevich
Сайт	pliner.pro
Режим работы	пн–пт, 9:00–18:00 МСК

На сайте правообладателя публикуются этот документ, описание функциональных характеристик, инструкция по установке и руководство пользователя.