

ООО «Клиенткор»
ИИ-платформа ClientCore (ClientCore AI Platform)

УТВЕРЖДАЮ
Генеральный директор ООО «Клиенткор»
_____ / Вишняков А. А.
«__» _____ 2026 г.

**ОПИСАНИЕ ПРОЦЕССОВ, ОБЕСПЕЧИВАЮЩИХ
ПОДДЕРЖАНИЕ ЖИЗНЕННОГО ЦИКЛА
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

ИИ-платформа ClientCore

Москва, 2026

Содержание

1. Общие сведения

Настоящий документ описывает процессы, обеспечивающие поддержание жизненного цикла программного обеспечения «ИИ-платформа ClientCore» (далее — Платформа), в том числе процессы устранения неисправностей, выявленных в ходе эксплуатации, и совершенствования Платформы, а также сведения о персонале, необходимом для обеспечения такой поддержки.

Правообладатель и разработчик ПО — ООО «Клиенткор». Поддержание жизненного цикла осуществляется силами штатных сотрудников правообладателя.

2. Общая организация процессов жизненного цикла

Жизненный цикл Платформы включает следующие процессы: сбор и анализ требований; проектирование; разработку; тестирование; развёртывание (выпуск версий); эксплуатацию и сопровождение; устранение неисправностей; совершенствование (развитие функциональности).

Управление задачами ведётся в системе отслеживания задач (issue tracker): каждая задача, идея или дефект оформляются карточкой с описанием, приоритетом, критериями готовности (Definition of Done) и сроком. Без оформленной карточки и срока задача в работу не принимается.

3. Управление версиями и выпуск обновлений

Исходный код Платформы хранится в системе управления версиями (Git). Применяется модель ветвления:

- ветка **dev** — текущая разработка; изменения вносятся через запросы на слияние (Pull Request) с код-ревью;
- ветка **main** — стабильная отревьюированная версия, используемая как точка отката.

Выпуск обновления на промышленную среду выполняется по регламентированной процедуре: создание резервной копии базы данных; получение кода целевой ветки; пересборка контейнеров; применение миграций схемы БД (Alembic) до старта приложения; перезапуск сервисов на новых образах; верификация работоспособности по журналам и контрольным запросам. При выявлении критической ошибки выполняется откат на стабильную версию (ветка main) и восстановление базы данных из резервной копии.

Изменения схемы базы данных вносятся исключительно через миграции Alembic (источник истины для схемы). На старте приложение проверяет соответствие схемы последней ревизии и не запускается при несоответствии, что исключает рассинхронизацию кода и данных.

4. Процесс устранения неисправностей

Устранение неисправностей, выявленных в ходе эксплуатации, осуществляется в следующем порядке:

1. Регистрация. Неисправность фиксируется: обращением пользователя, по данным мониторинга или по журналам. Создаётся карточка с описанием, шагами воспроизведения и приоритетом.

2. Диагностика. По журналам сервисов (Docker) и данным системы мониторинга ошибок (GlitchTip / Sentry SDK, привязка к request_id) локализуется причина.
3. Устранение. Исправление разрабатывается в ветке dev, сопровождается тестами, проходит код-ревью и сливается через Pull Request.
4. Выпуск. Исправление выкатывается на промышленную среду по процедуре раздела 3 с обязательной резервной копией БД и возможностью отката.
5. Верификация. Корректность исправления подтверждается на промышленной среде; карточка закрывается.

Классификация по приоритетам: P0 — критические (недоступность сервиса, потеря данных) — устраняются в первоочередном порядке; P1 — существенные; P2 — прочие. Контроль ошибок в реальном времени обеспечивается системой мониторинга с самостоятельным хостингом в инфраструктуре правообладателя.

5. Процесс совершенствования

Совершенствование Платформы (развитие функциональности, повышение производительности и надёжности) выполняется на постоянной основе:

- сбор предложений и обратной связи (в том числе автоматическое создание карточек по обратной связи пользователей из интерфейса);
- планирование работ по эпикам и спринтам с приоритизацией;
- реализация изменений по принятому процессу разработки (ветка dev → ревью → тестирование → выпуск);
- ведение журнала архитектурных решений (ADR) и технической документации в репозитории.

6. Тестирование

Применяются следующие виды проверок качества:

- модульные и интеграционные тесты серверной части (pytest): проверка миграций и соответствия схемы, валидации конфигурации, обработки ошибок, идиоматичности операций, бизнес-логики интеграций;
- автоматические тесты клиентской части (Vitest, Testing Library) и проверка типов TypeScript;
- непрерывная интеграция (CI, GitHub Actions) для клиентской части: установка зависимостей, проверка типов, прогон тестов;
- сценарные проверки (smoke, нагрузочные и комплексные сценарии) перед выпуском значимых изменений.

Контроль соответствия модели данных и схемы БД выполняется тестом, разворачивающим чистую базу и сверяющим результат миграций с описанием моделей.

7. Резервное копирование и восстановление

Для обеспечения сохранности данных выполняется регулярное резервное копирование PostgreSQL, MongoDB и объектного хранилища; копии PostgreSQL создаются перед каждым обновлением и по расписанию, хранятся в том числе во внешнем (офсайт)

хранилище. Процедура восстановления отработана и применяется при откате неудачных обновлений и при сбоях.

8. Сведения о персонале

Для поддержания жизненного цикла Платформы привлекается квалифицированный персонал правообладателя. Минимально необходимый состав по направлениям:

Функция	Кол-во штатных сотрудников	Квалификационные требования
Гарантийное обслуживание ПО	1	Высшее техническое образование; опыт сопровождения ПО; знание Linux, Docker, систем мониторинга.
Техническая поддержка ПО	1	Опыт технической поддержки; знание функциональности Платформы; навыки диагностики по журналам.
Модернизация (развитие) ПО	1	Опыт разработки на Python и/или TypeScript; знание FastAPI, SQLAlchemy, Next.js/React; навыки работы с Git, CI/CD, Docker.

Допускается совмещение функций одним специалистом при наличии соответствующей квалификации. Привлекаемый персонал обладает компетенциями в области серверной (Python 3.12, FastAPI, PostgreSQL, MongoDB) и клиентской (TypeScript, Next.js, React) разработки, администрирования контейнеризованных приложений (Docker, Docker Compose), эксплуатации СУБД и обеспечения информационной безопасности.

9. Контакты технической поддержки

Техническая поддержка осуществляется правообладателем — ООО «Клиенткор». Канал обращения: e-mail azim@clientcore.ru, тел. +7 (999) 846-96-66. Режим работы и параметры реагирования определяются условиями лицензионного договора.