

**Общество с ограниченной ответственностью
«РЖД – Цифровые пассажирские решения»**

*ОГРН 5167746236906 ИНН 9717045555
107078, г. Москва, ул. Новорязанская, д. 8А, стр. 2*

**ПРОГРАММА ДЛЯ ЭВМ
«АВТОПРОТОКОЛ»**

**Описание процессов, обеспечивающих поддержание
жизненного цикла программного обеспечения**

На 13 листах

Москва, 2026

АННОТАЦИЯ

Настоящий документ описывает совокупность организационных и технических процессов, обеспечивающих поддержание жизненного цикла программы для ЭВМ «Автопротокол» (далее – ПО, Сервис) на всех стадиях его существования: от планирования и разработки до сопровождения эксплуатации, устранения неисправностей, совершенствования и вывода из эксплуатации. Отдельные разделы посвящены организации технической поддержки пользователей, порядку устранения неисправностей, выявленных в ходе эксплуатации, порядку совершенствования ПО, а также сведениям о персонале, необходимом для обеспечения указанных процессов.

Процессы жизненного цикла ПО организованы правообладателем – ООО «РЖД – Цифровые пассажирские решения». Поддержание жизненного цикла осуществляется силами правообладателя на территории Российской Федерации без привлечения иностранных лиц.

СОДЕРЖАНИЕ

1. Общие сведения.....	4
2. Средства обеспечения жизненного цикла	4
3. Процесс разработки и выпуска версий	5
3.1. Управление требованиями и планирование	5
3.2. Реализация и контроль качества.....	5
3.3. Выпуск версий	6
3.4. Развертывание обновлений	6
4. Техническая поддержка пользователей	7
4.1. Организация поддержки.....	7
4.2. Каналы приема обращений и режим работы	7
4.3. Порядок обработки обращений и нормативные сроки	7
5. Устранение неисправностей, выявленных в ходе эксплуатации	8
5.1. Источники выявления неисправностей	8
5.2. Порядок устранения.....	9
5.3. Устойчивость к сбоям в ходе эксплуатации.....	9
6. Совершенствование программного обеспечения	10
6.1. Источники и планирование развития.....	10
6.2. Направления развития (дорожная карта).....	10
6.3. Управление изменениями.....	10
7. Персонал, необходимый для обеспечения поддержки.....	11
7.1. Роли и численность	11
7.2. Квалификационные требования	12
7.3. Компетенции и преемственность	12
8. Адреса осуществления деятельности и контактная информация	12

1. Общие сведения

Жизненный цикл ПО охватывает следующие стадии: формирование и анализ требований; проектирование; реализация (кодирование); тестирование и контроль качества; выпуск версий; развертывание; эксплуатация и сопровождение; совершенствование (развитие); вывод из эксплуатации. Поддержание жизненного цикла обеспечивается непрерывностью трех взаимоувязанных контуров деятельности правообладателя:

– контур разработки – планирование и реализация изменений, контроль качества, управление версиями и выпусками;

– контур технической поддержки – прием и обработка обращений пользователей и эксплуатирующих подразделений, устранение неисправностей;

– контур эксплуатационного сопровождения – мониторинг работоспособности, регламентное обслуживание, обновление развернутых экземпляров, обеспечение резервного копирования и восстановления.

Модель разработки – итеративно-инкрементальная: развитие ПО осуществляется последовательными версиями (релизами), каждая из которых проходит полный цикл проектирования, реализации, тестирования и контролируемого развертывания с подготовленной процедурой отката.

2. Средства обеспечения жизненного цикла

Для поддержания процессов жизненного цикла правообладателем используются следующие технические средства, размещенные в защищенной инфраструктуре правообладателя на территории Российской Федерации:

Средство	Назначение
Система контроля версий Git (корпоративный сервер GitLab)	Хранение и версионирование исходного текста, рецензирование изменений (merge request), управление ветками и метками релизов, отслеживание задач
Система сборки (Docker, Docker Compose)	Воспроизводимая сборка компонентов из исходного текста в контейнерные образы; идентичность сред разработки, тестирования и эксплуатации

Комплект автоматизированных тестов (pytest)	Модульные и контрактные тесты бизнес-логики компонентов; выполняются при подготовке каждого изменения и каждого релиза
Средства статического анализа кода (SAST)	Регулярный анализ исходного текста на уязвимости с триажем находок; результаты фиксируются в документации проекта
Эталонный измерительный корпус качества распознавания	Составной эталонный набор записей с разметкой («golden bench») для контроля качества транскрибации и диаризации при изменениях конвейера
Встроенные средства мониторинга ПО	Телеметрия воркеров, метрики производительности, журнал аудита настроек, секция «Состояние системы» административной панели
Стенд локальной отработки (E2E)	Локальный стенд полного цикла (включая GPU) для воспроизведения дефектов и проверки исправлений до выпуска

Подробное описание технических средств хранения исходного текста и объектного кода и средств компиляции приведено в отдельном документе «Описание технических средств хранения исходного текста и объектного кода ПО и технических средств компиляции», представляемом в комплекте документов заявления.

3. Процесс разработки и выпуска версий

3.1. Управление требованиями и планирование

Требования к развитию ПО формируются на основании: технического задания и планов развития, утверждаемых правообладателем; обращений пользователей и эксплуатирующих подразделений; результатов анализа встроенной статистики использования и телеметрии; результатов аудитов безопасности. Требования фиксируются в системе отслеживания задач корпоративного GitLab, приоритизируются руководителем разработки и включаются в план версии.

3.2. Реализация и контроль качества

Реализация изменений ведется в отдельных ветках системы контроля версий. Обязательные элементы контроля качества каждого изменения:

- рецензирование кода (code review) в форме merge request до слияния в основную ветку разработки;
- прохождение комплекта автоматизированных тестов (pytest; тесты бизнес-логики, контрактов сообщений, конфигурации, форматов результатов);
- для изменений конвейера распознавания – контрольные прогоны на эталонном корпусе с сопоставлением метрик качества (точность распознавания, качество диаризации, доля «фантомных» смен говорящих) с зафиксированными базовыми значениями;
- для изменений, затрагивающих безопасность – внеочередной статический анализ и триаж находок;
- актуализация эксплуатационной документации в составе того же изменения (документация хранится в репозитории вместе с кодом и рецензируется наравне с ним).

3.3. Выпуск версий

Выпуск версии оформляется слиянием в основную ветку с меткой версии в системе контроля версий. Для каждого релиза формируются: перечень изменений; инструкция обновления с указанием особенностей (новые параметры конфигурации, изменения схемы данных); процедура отката с контрольной точкой (резервная копия каталога приложения, тегирование текущих контейнерных образов, git-метка состояния до обновления). Изменения схемы данных выполняются аддитивно и идемпотентно, что обеспечивает совместимость отката с накопленными данными.

3.4. Развертывание обновлений

Обновление эксплуатируемых экземпляров выполняется по документированной процедуре (docs/DEPLOYMENT.md): ожидание завершения активных задач, обновление кода, пересборка образов из исходного текста, поэтапный запуск с контролем проверок живости, комплекс послерелизных проверок (смоук-задача полного цикла, контроль журналов, контроль индикаторов административной панели).

Принудительные автоматические обновления отсутствуют: момент и состав обновления полностью контролируются эксплуатирующей организацией.

4. Техническая поддержка пользователей

4.1. Организация поддержки

Техническая поддержка ПО осуществляется правообладателем по многоуровневой схеме:

Линия	Состав работ	Исполнители
1-я линия	Прием и регистрация обращений, первичная классификация, консультации по порядку работы с Сервисом, типовые вопросы	Специалисты поддержки правообладателя
2-я линия	Диагностика по журналам и телеметрии, устранение эксплуатационных инцидентов (перезапуск компонентов, разбор очереди недоставленных сообщений, корректировка настроек), эскалация дефектов	Системные администраторы
3-я линия	Анализ и устранение дефектов ПО, выпуск исправлений (hotfix) и плановых версий, доработка функциональности	Разработчики правообладателя

4.2. Каналы приема обращений и режим работы

Канал	Значение
Электронная почта поддержки	support@smarttravel.ru
Телефон	+7 (499) 490-25-58
Внутренняя система заявок (Service Desk)	Инфраменеджер
Режим работы поддержки	рабочие дни, 09:00–18:00 по московскому времени

4.3. Порядок обработки обращений и нормативные сроки

Каждое обращение регистрируется с присвоением идентификатора, классифицируется по категории (инцидент, консультация, запрос на изменение) и приоритету, после чего обрабатывается в соответствии с нормативными сроками:

Приоритет	Критерий	Время реакции	Целевое время решения
Критический	Сервис недоступен либо массово неработоспособен основной сценарий обработки	1 рабочий час	8 рабочих часов
Высокий	Существенное ограничение функциональности при сохранении основного сценария	4 рабочих часа	2 рабочих дня
Средний	Локальная ошибка с обходным решением; вопросы настройки	8 рабочих часов	5 рабочих дней
Низкий	Консультации, пожелания по развитию	1 рабочий день	в плановом порядке – в составе очередной версии

По результатам обработки заявителю направляется ответ с описанием решения либо с планом устранения; обращение закрывается после подтверждения заявителем. Статистика обращений анализируется ежемесячно и используется в планировании развития ПО (раздел 6).

5. Устранение неисправностей, выявленных в ходе эксплуатации

5.1. Источники выявления неисправностей

- обращения пользователей и администраторов по каналам поддержки (раздел 4);
- встроенный мониторинг: индикаторы живости компонентов, телеметрия воркеров, секция «Состояние системы» административной панели;
- диагностические статусы задач: каждая нештатно завершенная задача получает статус с машиночитаемой причиной; события ошибок фиксируются в статистике;
- очередь недоставленных сообщений (DLQ): изоляция «ядовитых» сообщений с сохранением полного контекста для разбора – накопление сообщений сигнализирует о систематической ошибке;
- журналы всех компонентов с унифицированными префиксами;
- результаты регулярного статического анализа кода и аудитов безопасности.

5.2. Порядок устранения

Процесс устранения неисправности включает последовательные этапы:

- регистрация и классификация – фиксация дефекта в системе отслеживания задач с присвоением приоритета по таблице раздела 4.3;
- локализация и диагностика – воспроизведение на локальном стенде полного цикла, анализ журналов, телеметрии и содержимого DLQ; для дефектов качества распознавания – прогон на эталонном корпусе;
- устранение – исправление в отдельной ветке с обязательным рецензированием и прохождением автоматизированных тестов; при возможности дефект закрепляется регрессионным тестом;
- выпуск исправления – в составе внеочередного исправления (hotfix) для критических и высоких приоритетов либо очередной плановой версии для остальных;
- установка и проверка – обновление эксплуатируемых экземпляров по процедуре раздела 3.4 с послерелизными проверками;
- уведомление – информирование заявителя и закрытие обращения; пополнение базы типовых неисправностей эксплуатационной документации.

5.3. Устойчивость к сбоям в ходе эксплуатации

Архитектура ПО минимизирует влияние неисправностей на пользователей до их устранения: автоматический повтор обработки при первичной ошибке; изоляция повторно ошибочных сообщений в DLQ без блокировки очереди; каскад фолбэков подсистемы саммаризации (деление сверхдлинных текстов, повтор с измененными параметрами, принудительное сокращение входного текста); сторожевые таймеры и автоматический перезапуск зависших компонентов; защита от бесконечной повторной обработки задач счетчиком попыток. Указанные механизмы обеспечивают деградацию отдельной задачи вместо отказа Сервиса в целом.

6. Совершенствование программного обеспечения

6.1. Источники и планирование развития

Совершенствование ПО осуществляется плановыми версиями на основе: предложений пользователей и эксплуатирующих подразделений; анализа встроенной статистики использования (объемы, категории пользователей, доля ошибок) и телеметрии производительности (RTF, скорость генерации); результатов сравнительных испытаний моделей искусственного интеллекта на эталонном корпусе; развития экосистемы свободных компонентов (обновление версий, замена устаревающих компонентов). Предложения консолидируются в дорожную карту, утверждаемую правообладателем.

6.2. Направления развития (дорожная карта)

- повышение качества распознавания и диаризации: регулярные сравнительные испытания версий моделей и параметров конвейера на эталонном корпусе с внедрением улучшений, подтвержденных метриками;
- развитие распределенных схем развертывания: переключаемый брокер сообщений (RabbitMQ/Apache Kafka), вынос вычислительных воркеров на отдельные серверы, подготовка к оркестрации Kubernetes и автомасштабированию воркеров;
- расширение средств администрирования: развитие настроек времени выполнения, интеграция с внешними системами мониторинга (Prometheus/Grafana);
- плановая замена компонентов с изменившимися условиями сопровождения (в частности, заложена среднесрочная миграция объектного хранилища на актуально сопровождаемое S3-совместимое решение);
- расширение форматов результатов и интеграционных интерфейсов для встраивания в корпоративные системы документооборота.

6.3. Управление изменениями

Каждое направление развития проходит стандартный цикл: проработка требований → проектное решение (при существенных изменениях –

с фиксацией дизайна в документации репозитория) → реализация с рецензированием и тестированием → контролируемый выпуск с процедурой отката (раздел 3). Обратная совместимость интерфейсов и контрактов сообщений контролируется контрактными тестами; изменения, затрагивающие эксплуатацию, сопровождаются актуализацией документации и инструкций обновления.

7. Персонал, необходимый для обеспечения поддержки

7.1. Роли и численность

Поддержание жизненного цикла ПО обеспечивается персоналом правообладателя в следующем составе:

Роль	Основные обязанности	Численность, чел.
Руководитель разработки (владелец продукта)	Планирование версий, приоритизация, приемка результатов, дорожная карта	1, совмещение допускается
Инженер-программист (backend/ML)	Разработка и сопровождение компонентов, конвейера распознавания и саммаризации, устранение дефектов, рецензирование кода	1, совмещение допускается
Инженер сопровождения (DevOps/системный администратор)	Развертывание, обновление, мониторинг, резервное копирование, инфраструктура разработки	1, совмещение допускается
Специалист технической поддержки	Прием и обработка обращений, консультирование пользователей, ведение базы знаний	1, совмещение допускается
Специалист по тестированию	Автоматизированные и ручные испытания, контроль качества релизов, сопровождение эталонного корпуса	1, совмещение допускается

Численность уточняется по штатному расписанию; совмещение ролей допускается при сохранении покрытия всех процессов. Указанный персонал состоит в трудовых отношениях с правообладателем; привлечение иностранных организаций и специалистов к разработке и поддержке ПО не осуществляется.

7.2. Квалификационные требования

Роль	Квалификационные требования
Инженер-программист	Высшее техническое образование; опыт промышленной разработки на Python от 3 лет; знание асинхронных архитектур, брокеров сообщений, СУБД PostgreSQL, контейнеризации Docker; для профиля ML – опыт работы с моделями распознавания речи и большими языковыми моделями, стеком PyTorch/ONNX/CUDA
Инженер сопровождения	Опыт администрирования Linux от 3 лет; уверенное владение Docker/Docker Compose, сетевыми технологиями, средствами резервного копирования; опыт эксплуатации GPU-серверов NVIDIA
Специалист поддержки	Знание функциональности Сервиса в объеме пользовательской и административной документации; навыки работы с системой заявок; грамотная письменная и устная речь
Специалист по тестированию	Опыт тестирования веб-приложений и API; владение pytest; понимание метрик качества распознавания речи

7.3. Компетенции и преемственность

Преемственность знаний обеспечивается: хранением проектной и эксплуатационной документации в репозитории вместе с исходным текстом; обязательным рецензированием изменений (исключает единоличное знание кода); базой типовых неисправностей; актами передачи исходного кода и отчетами о выполненных работах. Новые сотрудники вводятся в проект по комплекту документации репозитория (архитектура, конфигурация, эксплуатация, конвейер распознавания), поддерживаемому в актуальном состоянии.

8. Адреса осуществления деятельности и контактная информация

Показатель	Значение
Фактический адрес размещения инфраструктуры разработки ПО	107078, г. Москва, вн.тер.г. муниципальный округ Красносельский, ул. Новорязанская, д. 8А, стр. 2
Фактический адрес размещения персонала разработки и поддержки	107078, г. Москва, вн.тер.г. муниципальный округ Красносельский, ул. Новорязанская, д. 8А, стр. 2
Электронная почта по вопросам поддержки ПО	support@smarttravel.ru
Телефон	+7 (499) 490-25-58
Официальный сайт правообладателя	https://smarttravel.ru/

Вся инфраструктура разработки, тестирования и поддержки ПО (серверы системы контроля версий, сборки, стенды, серверы эксплуатации) размещена на территории Российской Федерации и находится под управлением правообладателя.