ОПИСАНИЕ ПРОЦЕССОВ, ОБЕСПЕЧИВАЮЩИХ ПОДДЕРЖАНИЕ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, И ИНФОРМАЦИЯ О ПЕРСОНАЛЕ, НЕОБХОДИМОМ ДЛЯ ИХ ОБЕСПЕЧЕНИЯ

программы для ЭВМ «Версиум AI»

1. Общие сведения

Наименование	Версиум AI (Versium AI)
Правообладатель	ООО «ВЕРСИУМ», ИНН 6670527467, ОГРН 1246600061014
Адрес правообладателя	620078, Свердловская обл., г. Екатеринбург, пер. Лобачевского, д. 1, кв. 23
Тип программы	Самостоятельно функционирующая программа для ЭВМ, серверное приложение с веб-интерфейсом
Размещение	В контуре заказчика (on-premise), в Docker-контейнерах на серверах под управлением Windows или Linux
Сопровождение	Осуществляется правообладателем
Обращения в поддержку	info@versium.ru, www.versium.ru, а также каналы, указанные в договоре (лицензионном соглашении)

Настоящий документ описывает процессы, обеспечивающие поддержание жизненного цикла программы «Версиум AI» (далее – программа): разработку и выпуск версий, распространение и обновление, техническую поддержку, устранение неисправностей и совершенствование, а также содержит сведения о персонале, необходимом для обеспечения такой поддержки.

2. Модель жизненного цикла

Программа развивается по итерационной модели: доработки и исправления выполняются непрерывно и поставляются заказчикам в виде очередных выпусков. Каждое изменение проходит этапы, приведённые в таблице.

Этап	Содержание работ
Формирование требований	Анализ требований заказчиков, обращений поддержки, требований законодательства в области защиты информации; техническая спецификация
Проектирование	Проектные решения, описание программных интерфейсов (контрактов) серверной части и веб-интерфейса, изменения схемы базы данных
Разработка	Реализация изменений вместе с автоматическими тестами по внутренним стандартам разработки
Тестирование	Автоматические тесты, код-ревью, проверка на тестовом стенде

Этап	Содержание работ
Выпуск	Сборка образов компонентов из зафиксированного состояния исходного кода, присвоение номера версии
Распространение	Публикация выпуска на сервере дистрибутивов и в реестре образов, подготовка офлайн-комплекта
Эксплуатация	Установка и использование программы заказчиком, администрирование, резервное копирование
Сопровождение	Техническая поддержка, устранение неисправностей, совершенствование, поставка обновлений
Вывод из эксплуатации	Сохранение необходимых данных, удаление компонентов и данных программы по решению заказчика

Исходный код, проектная документация, стандарты разработки и описания программных интерфейсов хранятся в системе контроля версий Git на сервере правообладателя, размещённом на территории Российской Федерации; доступ предоставляется только персоналу правообладателя. Разработка ведётся в основной ветке, крупные доработки – в отдельных ветках с включением в основную после приёмки. История изменений сохраняется полностью, что позволяет однозначно установить состав любого выпуска и воспроизвести его сборку.

3. Управление версиями и выпуском

3.1. Нумерация и выпуск версий

Номер версии формируется по шаблону `ГГГГ.К.ММДДЧЧММ`: год, квартал, месяц, день, час и минута (UTC) фиксации исходного кода, из которой собран выпуск (например, `2026.4.10271530`). Номер однозначно связывает выпуск с состоянием исходного кода и отображается в веб-интерфейсе программы.

Выпуск выполняется в следующем порядке:

- сборка Docker-образов компонентов (серверная часть, веб-интерфейс, исполнитель, веб-сервер) из основной ветки после прохождения всех проверок и публикация их в реестре образов правообладателя (`registry.versium.ru`);
- формирование комплекта поставки: файлы конфигурации контейнеров для установки all-in-one и с внешней СУБД, офлайн-комплект с образами всех компонентов и описание выпуска (`manifest.json`) с контрольными суммами SHA-256;
- публикация на сервере дистрибутивов `releases.versium.ru` и отметка выпуска в перечне версий `versions.json` как актуального стабильного (`latest-stable`).

На сервере дистрибутивов сохраняется несколько последних выпусков, что позволяет установить конкретную версию или вернуться к предыдущей.

3.2. Распространение

- **онлайн-установка** – сценарий установки загружается с сервера дистрибутивов, по умолчанию устанавливает актуальный стабильный выпуск (версия может быть указана явно), проверяет контрольную сумму комплекта и получает образы из реестра по учётным данным, предоставленным заказчику;

- **офлайн-установка** – комплект с образами всех компонентов устанавливается без обращения к серверу дистрибутивов, реестру образов и сети Интернет; применяется для изолированных контуров.

3.3. Обновление у заказчика

Обновление выполняется администратором заказчика тем же сценарием в режиме обновления, без переустановки программы:

- диагностика установки (режим `update-check`) без внесения изменений;
- резервное копирование базы данных; копия включает и ключи защиты данных, хранящиеся в базе данных;
- обновление (режим `upgrade`): получение комплекта новой версии, проверка контрольной суммы, замена конфигурации контейнеров, загрузка образов и перезапуск;
- автоматическое применение миграций схемы базы данных при запуске серверной части; миграции применяются последовательно, поэтому обновление с предыдущих выпусков выполняется непосредственно до актуального;
- проверка работоспособности: состояние контейнеров, служебная проверка доступности, номер версии.

```
bash install-versium-ai.sh update-check
docker compose exec -T postgres pg_dump -U versiumai versiumaidb > backup.sql
bash install-versium-ai.sh upgrade
```

Сценарий `install-versium-ai.sh` загружается с сервера дистрибутивов или входит в офлайн-комплект. Параметры установки, сертификаты TLS и тома данных при обновлении не изменяются, данные и настройки сохраняются. В изолированном контуре обновление выполняется из офлайн-комплекта новой версии; при внешней СУБД резервное копирование выполняется штатными средствами заказчика. Возврат к предыдущей версии выполняется установкой предыдущего выпуска с восстановлением базы данных из резервной копии.

4. Техническая поддержка

Техническая поддержка осуществляется правообладателем и включает консультирование по установке, настройке и использованию программы, помощь в диагностике, приём и обработку сообщений об ошибках и уязвимостях, предоставление обновлений и приём предложений по совершенствованию. Объём, режим и сроки поддержки определяются договором (лицензионным соглашением).

Обращения принимаются по электронной почте info@versium.ru и по иным каналам, указанным в договоре. В обращении указываются организация и контактное лицо, номер версии и вариант установки, операционная система сервера, описание проблемы и порядок её воспроизведения, время возникновения, фрагменты журналов и снимки экрана. Конфиденциальные сведения в журналах предварительно обезличиваются заказчиком. Удалённое подключение к установке заказчика выполняется только по согласованию с заказчиком и его средствами доступа.

Каждое обращение регистрируется, получает номер и приоритет; заказчик информируется о ходе работ. Обращение закрывается после предоставления решения и подтверждения заказчиком.

Приоритет	Критерии	Порядок обработки
Критический	Программа неработоспособна или недоступна для всех пользователей; угроза безопасности или утраты данных	В первоочередном порядке, временное решение, при необходимости внеочередной выпуск
Высокий	Недоступна существенная функция, обходного решения нет	В приоритетном порядке, исправление в ближайшем выпуске
Средний	Нарушена отдельная функция, есть обходное решение	Исправление в плановом выпуске
Низкий	Консультации, незначительные дефекты, предложения	В порядке очереди или в плане развития

Конкретные сроки реакции и решения по каждому приоритету определяются договором (лицензионным соглашением).

5. Устранение неисправностей

- **приём и анализ** – обращение регистрируется и классифицируется, специалист поддержки запрашивает недостающие сведения и при возможности предлагает обходное решение;
- **воспроизведение и диагностика** – неисправность воспроизводится на тестовом стенде с той же версией и аналогичной конфигурацией; причина устанавливается по журналам контейнеров, журналу установки и обновления, служебным проверкам (доступность, версия, состояние лицензии), журналу аудита и телеметрии (OpenTelemetry);
- **исправление** – разработчик вносит изменение в исходный код и добавляет автоматический тест, воспроизводящий неисправность;
- **проверка** – код-ревью, полный набор автоматических тестов, проверка на тестовом стенде;
- **выпуск и доставка** – исправление включается в очередной выпуск, для критических неисправностей – во внеочередной; выпуск публикуется на сервере дистрибутивов, для изолированных контуров передаётся офлайн-комплект, заказчик уведомляется;
- **закрытие** – после установки обновления и подтверждения заказчиком.

6. Совершенствование программы

Источниками совершенствования являются предложения заказчиков, анализ обращений технической поддержки, изменения законодательства в области защиты информации и персональных данных, развитие языковых моделей и протоколов взаимодействия с ними, изменения поддерживаемых операционных систем, СУБД и средств контейнеризации.

Предложения регистрируются в перечне задач развития, оцениваются руководителем продукта и архитектором и включаются в план выпусков с учётом приоритета. Каждая доработка выполняется по процессу раздела 2. Совместимость с данными и настройками ранее установленных версий сохраняется: изменения схемы базы данных выполняются миграциями, применяемыми автоматически при обновлении. Основные направления развития: состав поддерживаемых провайдеров языковых моделей, в том числе локальных; правила обнаружения и обезличивания данных; AI-агенты, рабочие процессы и подключение MCP-серверов; аудит, политики доступа и администрирование; совместимость с отечественными операционными системами и СУБД.

7. Обеспечение качества и безопасности

Разработка ведётся по внутренним стандартам: архитектурным правилам, правилам оформления кода серверной части и веб-интерфейса, стандартам автоматических тестов. Программные интерфейсы описываются контрактами до начала реализации. Перед фиксацией каждого изменения автоматически выполняются проверка форматирования, сборка, при которой любое предупреждение компилятора считается ошибкой, и проверка кода веб-интерфейса линтером. При сборке применяются статические анализаторы исходного кода .NET (правила надёжности, безопасности и стиля) и строгая проверка типов TypeScript. Каждый набор изменений до включения в основную ветку проходит независимую проверку (код-ревью) на соответствие архитектурным правилам, контрактам и стандартам.

Вид тестов	Объект проверки	Средства
Модульные (серверная часть)	Доменная логика, валидация, маскирование, лицензирование, исполнитель	xUnit
Интеграционные (серверная часть)	Программные интерфейсы в сборе с реальной СУБД PostgreSQL в изолированных контейнерах	xUnit, Testcontainers
Модульные (веб-интерфейс)	Компоненты, состояние и логика интерфейса	Vitest
Сквозные	Пользовательские сценарии в браузере на развёрнутой программе	Playwright

Меры обеспечения безопасности:

- **секреты** – ключи доступа к провайдерам хранятся в базе данных в зашифрованном виде и не отображаются в открытом виде, допускается только их замена;
- **контроль зависимостей** – версии библиотек управляются централизованно и фиксируются, при сборке выполняется проверка зависимостей на известные уязвимости, при выходе исправлений базовых образов выполняется пересборка;
- **целостность дистрибутива** – контрольные суммы SHA-256 проверяются сценарием установки, доступ к реестру образов – по учётным данным;
- **лицензионный ключ** – подписывается электронной подписью правообладателя и проверяется локально, без обращения к внешним серверам;
- **уязвимости** – сообщения об уязвимостях обрабатываются с критическим приоритетом: оценка, исправление, внеочередной выпуск, уведомление заказчиков.

Функционирование установленной программы и её обновление из офлайн-комплекта не требуют обращения к ресурсам, размещённым за пределами Российской Федерации.

8. Эксплуатация и вывод из эксплуатации

В ходе эксплуатации администратор заказчика контролирует состояние компонентов, выполняет регулярное резервное копирование базы данных, управляет сроком хранения журнала аудита, устанавливает обновления и продлевает лицензию. При выводе из эксплуатации заказчик сохраняет данные, подлежащие хранению (в том числе журнал аудита), после чего удаляет контейнеры, образы и тома данных программы; уничтожение данных выполняется по внутренним правилам заказчика.

9. Персонал, необходимый для обеспечения поддержки

Для поддержания жизненного цикла программы, включая техническую поддержку, устранение неисправностей и совершенствование, необходим персонал следующих ролей.

Роль	Функции	Требуемая квалификация	Числ.
Руководитель продукта	Планирование развития и выпусков, приоритеты, взаимодействие с заказчиками	Высшее техническое образование, опыт управления разработкой ПО от 3 лет	1
Архитектор	Архитектура, проектные решения, контракты интерфейсов, ревью ключевых изменений	Высшее техническое образование, проектирование серверных систем на .NET от 5 лет, PostgreSQL, контейнеризация	1
Разработчик серверной части	Функции и исправления, миграции базы данных, модульные и интеграционные тесты	Высшее техническое образование, C#/.NET от 3 лет, ASP.NET Core, Entity Framework Core, PostgreSQL	от 2
Разработчик веб-интерфейса	Функции и исправления веб-интерфейса, тесты интерфейса	Техническое образование, TypeScript и React от 2 лет	от 1
Инженер по тестированию	Воспроизведение неисправностей, сквозные тесты, приёмка выпусков	Тестирование веб-приложений и API от 2 лет, автоматизация тестирования	1
DevOps-инженер	Сборка и публикация выпусков, реестр образов, сервер дистрибутивов, тестовые стенды, помощь в установке	Администрирование Linux и Windows Server от 3 лет, Docker, сети и TLS, резервное копирование PostgreSQL	1
Специалист технической поддержки	Приём и регистрация обращений, консультации, первичная диагностика, информирование заказчиков	Техническое образование, администрирование Linux и Windows, Docker, основы СУБД и сетей	от 1
Специалист по информационной безопасности	Анализ уязвимостей, контроль зависимостей, требования безопасности, участие в ревью	Высшее образование или переподготовка в области ИБ, опыт от 2 лет	1

Минимально необходимая численность персонала составляет 9 человек; при увеличении количества установок и обращений численность специалистов поддержки и разработчиков увеличивается соразмерно нагрузке. Персонал должен знать архитектуру программы, внутренние

стандарты разработки и эксплуатационную документацию и соблюдать конфиденциальность сведений заказчиков. Непрерывность сопровождения обеспечивается хранением исходного кода, документации и полной истории изменений в системе контроля версий, что позволяет передавать задачи между специалистами без потери знаний.