

Программное обеспечение
«Servicepipe Application Delivery Controller»
**Описание процессов жизненного цикла программного
обеспечения**

Москва — 2026 г.

Оглавление

1 Общие сведения	3
1.1 Назначение документа	3
1.2 Сведения о программном обеспечении	3
1.3 Правообладатель и область применения	3
2 Процессы жизненного цикла ПО	4
2.1 Модель жизненного цикла	4
2.2 Анализ требований	4
2.3 Планирование работ и версий	5
2.4 Проектирование	5
2.5 Разработка	5
2.6 Управление зависимостями	6
2.7 Сборка и подготовка объектного кода	6
2.8 Тестирование	7
2.9 Подготовка релиза	7
2.10 Внедрение	7
3 Процессы сопровождения и обеспечения жизненного цикла	8
3.1 Поддержка версий и развитие функциональности	8
3.2 Устранение сбойных ситуаций и дефектов	8
3.3 Эксплуатационная и консультационная поддержка	9
3.4 Документирование	9
3.5 Обучение и поддержание квалификации персонала	10
3.6 Обеспечение безопасности и устойчивости	10
4 Управление конфигурацией, исходным кодом и релизными артефактами	10
4.1 Хранение исходного кода	10
4.2 Хранение объектного кода и поставочных артефактов	11
4.3 Контроль изменений и трассируемость	11
5 Персонал, обеспечивающий жизненный цикл ПО	12
6 Инфраструктура разработки, сборки, тестирования и поддержки	13
6.1 Инфраструктура хранения и разработки	13
6.2 Инфраструктура сборки и тестирования	13
6.3 Инфраструктура технической поддержки	14

1 Общие сведения

1.1 Назначение документа

Настоящий документ содержит описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения «Servicerpipe Application Delivery Controller» (далее — ПО «Servicerpipe ADC», ADC, программное обеспечение). В документе рассматриваются процессы анализа требований, проектирования, разработки, тестирования, выпуска, внедрения, сопровождения, документирования, управления версиями, управления исходным и объектным кодом, а также процессы обеспечения квалификации персонала, участвующего в создании и сопровождении программного обеспечения.

Документ подготовлен для использования в составе комплекта материалов, предоставляемых при подтверждении характеристик программного обеспечения и описании организационно-технических процессов его разработки и сопровождения.

1.2 Сведения о программном обеспечении

ПО «Servicerpipe ADC» предназначено для публикации прикладных сервисов и API, маршрутизации и балансировки L4/L7-трафика, включая TCP, UDP, HTTP и HTTPS, терминирования и проксирования TLS-соединений, а также централизованного управления конфигурацией доставки приложений.

Программное обеспечение размещается перед backend-сервисами заказчика, принимает входящий трафик, сопоставляет его с правилами обработки, применяет политики TLS, маршрутизации, балансировки, доступности, логирования и обработки HTTP-трафика, после чего передает запрос или соединение в выбранный Upstream.

Управление выполняется через административный контур и API управления. Пользователь может управлять публикациями приложений, маршрутами, потоковыми маршрутами, upstream-узлами, проверками здоровья, SSL-сертификатами, секретами, потребителями API, группами потребителей, плагинами, глобальными правилами, конфигурациями плагинов, метаданными плагинов и proto-файлами.

Программное обеспечение относится к серверному и связующему программному обеспечению и предназначено для эксплуатации в инфраструктуре заказчика или в private cloud. ADC может использоваться как самостоятельный программный контроллер доставки приложений и как компонент инфраструктуры публикации HTTP(S), gRPC, TCP, TLS и UDP-сервисов.

1.3 Правообладатель и область применения

Правообладателем программного обеспечения является Общество с ограниченной ответственностью «Сервиспайп» (ООО «Сервиспайп»).

Параметр	Значение
Полное наименование	Общество с ограниченной ответственностью «Сервиспайп»
Сокращенное наименование	ООО «Сервиспайп»
ИНН / КПП / ОГРН	7708257951 / 772201001 / 1157746483784
Юридический адрес	Россия, 111024, г. Москва, ул. Кабельная 2-я, д. 2, стр. 3, эт. 2, помещение XVIII, ком. 8
Телефон	+7 (495) 374-62-65

Электронная почта	welcome@servicepipe.ru
Веб-сайт	servicepipe.ru

Процессы жизненного цикла, описанные в настоящем документе, применяются к собственным компонентам Servicepipe ADC, включая runtime-модули, патчи, дополнительные сервисы, backend и frontend личного кабинета, API-адаптеры, упаковку, installer, конфигурационные шаблоны и эксплуатационную документацию. Сторонние open source компоненты рассматриваются как зависимости проекта и учитываются в рамках процесса управления зависимостями и сборкой.

2 Процессы жизненного цикла ПО

2.1 Модель жизненного цикла

Разработка и сопровождение ПО «Servicepipe ADC» осуществляются в рамках итерационно-релизной модели жизненного цикла с обязательным прохождением контрольных стадий для каждой версии программного обеспечения. Такая модель позволяет планомерно развивать функциональность, выпускать корректирующие обновления и сохранять управляемость изменений, влияющих на обработку трафика, отказоустойчивость, безопасность и административное управление.

Для каждой версии программного обеспечения выполняются следующие основные стадии: анализ требований, планирование работ, проектирование, разработка, сборка, тестирование, подготовка релиза, внедрение и сопровождение. Переход к последующей стадии выполняется после фиксации результата предыдущей стадии и устранения критичных несоответствий, препятствующих выпуску версии.

Жизненный цикл охватывает как функциональные изменения, связанные с пользовательскими возможностями ADC, так и технические изменения runtime-слоя, API управления, интерфейса администратора, поставочного installer-а, инфраструктурных интеграций и эксплуатационной документации.

2.2 Анализ требований

На стадии анализа выполняется выявление, уточнение и формализация требований к программному обеспечению. Источниками требований являются продуктовая стратегия, обращения и потребности заказчиков, результаты эксплуатации текущих версий, требования к безопасности и отказоустойчивости, требования к совместимости с инфраструктурой заказчиков, а также внутренние инициативы по развитию архитектуры и повышению наблюдаемости.

В рамках анализа определяются функциональные требования к маршрутизации, балансировке, TLS, upstream-узлам, health checks, идентификации потребителей, плагинам, логированию, статистике и сетевым режимам. Отдельно анализируются нефункциональные требования: производительность, масштабируемость, отказоустойчивость, безопасность, удобство эксплуатации, воспроизводимость сборки и возможность offline-развертывания.

Результатом стадии анализа является согласованный перечень требований к версии программного обеспечения. Требования могут оформляться в виде технического задания, продуктового описания, задач в системе управления работами, спецификаций API,

пользовательских сценариев, прототипов интерфейса или иных формализованных материалов, достаточных для последующего проектирования и реализации.

2.3 Планирование работ и версий

На стадии планирования определяется состав работ, входящих в очередную версию или корректирующее обновление. Для каждой работы устанавливаются ожидаемый результат, ответственные исполнители, зависимости от смежных компонентов, требования к тестированию и критерии готовности.

Планирование выполняется с учетом приоритетов развития продукта, влияния изменений на пользователей, необходимости устранения дефектов, сроков поставки, совместимости с существующими конфигурациями и готовности документации. Изменения, затрагивающие runtime-обработку трафика, схему хранения конфигурации, API управления или installer, рассматриваются как изменения повышенной значимости и требуют дополнительной архитектурной проверки.

Результатом стадии планирования является утвержденный набор задач версии, предварительный перечень релизных артефактов, план тестирования и перечень документации, которую необходимо создать или актуализировать.

2.4 Проектирование

На стадии проектирования разрабатываются архитектурные и технические решения, обеспечивающие реализацию согласованных требований. Проектирование охватывает runtime-слой обработки трафика, backend личного кабинета и API управления, frontend личного кабинета, installer, схемы данных, интеграции с инфраструктурными сервисами, форматы конфигурации и механизмы наблюдаемости.

В рамках проектирования определяются состав и границы компонентов, модели данных, контракты API, структура форм пользовательского интерфейса, порядок взаимодействия компонентов, требования к хранению и применению конфигурации, а также сценарии отката или безопасного внедрения изменений.

Особое внимание уделяется решениям, влияющим на обработку HTTP(S), gRPC, TCP, TLS и UDP-трафика, корректность работы health checks, выбор backend-нод, TLS-терминацию, работу с сертификатами, применение плагинов, rate limiting, логирование и сетевые режимы Active/Active или Active/Standby.

Результатом стадии проектирования является комплект проектных решений, содержащий достаточные сведения для разработки: описание архитектуры, схемы взаимодействия, спецификации API, макеты интерфейса, описание изменений конфигурации, требования к миграциям и критерии проверки.

2.5 Разработка

На стадии разработки осуществляется программная реализация функциональности в соответствии с архитектурными и техническими решениями. Разработка ведется с использованием системы контроля версий Git, размещенной на собственной инфраструктуре ООО «Сервисайп». Все изменения исходного кода фиксируются в репозиториях и сопровождаются историей изменений.

Разработка охватывает следующие группы компонентов: runtime-модули и патчи, дополнительные сервисы, backend личного кабинета и API управления, frontend личного кабинета, installer и bootstrap-скрипты, Ansible playbook-и, шаблоны конфигурации, тесты и документацию.

В ходе разработки выполняются реализация новых возможностей, доработка существующей функциональности, исправление выявленных дефектов, актуализация тестов, подготовка миграций и изменение поставочных материалов. Изменения проходят внутренний контроль: проверку соответствия требованиям, анализ влияния на смежные компоненты, код-ревью и подготовку к тестированию.

Результатом стадии разработки является набор изменений, интегрированный в соответствующие ветки репозитория и подготовленный к сборке тестовой или релизной версии программного обеспечения.

2.6 Управление зависимостями

ПО «Servicerpipe ADC» использует собственные компоненты ООО «Сервиспайп» и набор сторонних компонентов под свободными лицензиями. Сторонние компоненты рассматриваются как зависимости проекта; исключительные права на них как на самостоятельные проекты не заявляются.

В рамках управления зависимостями выполняются фиксация используемых версий, учет лицензий, хранение сведений о составе поставки, проверка совместимости и воспроизводимости сборки. Для npm/yarn-зависимостей состав фиксируется lockfile и отчетами о лицензиях. Для runtime-зависимостей и инфраструктурных компонентов фиксируются версии, используемые в поставке и manifest-е релиза.

К ключевым зависимостям и инфраструктурным компонентам относятся Apache APISIX, NGINX/OpenResty, LuaJIT, etcd, PostgreSQL, ClickHouse, VictoriaMetrics, Vector, Kafka, Docker/OCI runtime, keepalived, Patroni и иные компоненты, необходимые для работы конкретной поставки.

Результатом процесса управления зависимостями является контролируемый перечень собственных и сторонних компонентов, позволяющий воспроизвести сборку, проверить лицензионный состав и обеспечить сопровождаемость поставки.

2.7 Сборка и подготовка объектного кода

На стадии сборки выполняется преобразование исходного текста программного обеспечения в объектный код и поставочные артефакты. Сборка выполняется на контролируемой инфраструктуре ООО «Сервиспайп» с использованием зафиксированных версий исходного кода, зависимостей, конфигураций сборки и manifest-а версии.

В результате сборки формируются RPM-пакеты, архивы, Docker/OCI-образы, binary payload, конфигурационные шаблоны, installer, материалы offline-поставки и сопроводительные файлы. В состав поставки включаются собственные компоненты Servicerpipe ADC и необходимые зависимости, прошедшие проверку и размещенные в контролируемых хранилищах артефактов.

Результатом стадии является сформированный комплект объектного кода и релизных артефактов, пригодный для установки на тестовом стенде, приемочном стенде или в инфраструктуре заказчика.

2.8 Тестирование

На стадии тестирования проводится проверка реализованной версии программного обеспечения на соответствие функциональным и нефункциональным требованиям. Тестирование выполняется на тестовых стендах, воспроизводящих типовые эксплуатационные сценарии использования ADC.

В рамках тестирования выполняются функциональные, интеграционные, регрессионные, API-, UI-, инсталляционные и, при необходимости, нагрузочные проверки. Проверяется корректность создания и изменения сущностей ADC, применение маршрутов и upstream-настроек, работа health checks, выбор backend-нод, TLS-терминация, обработка сертификатов и секретов, применение плагинов, сбор статистики, логирование, сетевые режимы и поведение при отказах.

Для frontend и backend-компонентов проверяются формы ввода, валидация пользовательских данных, корректность GraphQL/REST-взаимодействия, авторизация, обработка ошибок и сценарии удаления или отключения сущностей. Для installer-a проверяются установка, обновление, удаление, применение inventory-шаблонов, корректность manifest-a и работоспособность offline-развертывания.

При выявлении несоответствий версия возвращается на доработку. Исправления повторно проходят проверки, связанные с измененным функционалом, а также необходимые регрессионные проверки. Результатом стадии тестирования является версия программного обеспечения, соответствующая установленным требованиям и готовая к подготовке релиза.

2.9 Подготовка релиза

На стадии подготовки релиза выполняется финальная фиксация состава версии, маркировка исходного кода, формирование релизного manifest-a, подготовка пакетов и образов, размещение артефактов в поставочных репозиториях и актуализация документации.

В состав релизных материалов включаются объектный код, installer, конфигурационные шаблоны, сведения о зависимостях, инструкции по установке и эксплуатации, описание функциональных характеристик, сведения о хранении исходного и объектного кода, а также иные документы, необходимые для эксплуатации программного обеспечения.

Результатом стадии является релизный комплект Servicepipe ADC, содержащий все материалы, необходимые для установки, обновления и сопровождения версии.

2.10 Внедрение

На стадии внедрения выполняется установка или обновление программного обеспечения в инфраструктуре заказчика, на приемочном стенде или на ином согласованном контуре. Внедрение может выполняться силами заказчика по эксплуатационной документации либо при участии специалистов ООО «Сервиспайп».

В рамках внедрения подготавливается инфраструктура, проверяются системные требования, устанавливаются компоненты ADC, применяется inventory и конфигурационные шаблоны, выполняется первичная настройка gateway-уровня, backend и frontend компонентов, проверяется доступ к панели управления и API, а также выполняются контрольные проверки публикации тестового приложения или API.

Результатом стадии внедрения является установленная и настроенная версия программного обеспечения, готовая к эксплуатации. После внедрения сведения о версии, особенностях установки и выявленных замечаниях учитываются в процессах сопровождения и дальнейшего развития продукта.

3 Процессы сопровождения и обеспечения жизненного цикла

Процессы сопровождения направлены на обеспечение стабильного функционирования ПО «Servicepipe ADC» в процессе эксплуатации, поддержание актуальности программного обеспечения, устранение выявленных дефектов, развитие функциональности и обеспечение пользователей необходимой документацией и консультационной поддержкой.

Сопровождение осуществляется организацией-разработчиком на протяжении всего срока использования программного обеспечения в рамках установленных договорных и внутренних регламентов.

3.1 Поддержка версий и развитие функциональности

Управление версиями Servicepipe ADC осуществляется на основе принятой системы нумерации релизов и release manifest-ов. В зависимости от характера изменений выпускаются корректирующие обновления, минорные версии с расширением функциональности и мажорные версии, содержащие существенные архитектурные изменения.

В рамках поддержки версий выполняются:

- выпуск корректирующих обновлений для устранения дефектов;
- выпуск минорных версий с расширением функциональных возможностей;
- выпуск мажорных версий при значимых изменениях архитектуры или модели поставки;
- ведение журнала изменений и фиксация состава релиза;
- поддержание совместимости обновлений с существующими конфигурациями, если это предусмотрено характером изменений;
- актуализация пользовательской, эксплуатационной и технической документации.

Решения о выпуске новых версий принимаются с учетом результатов эксплуатации текущих версий, обращений пользователей, необходимости устранения выявленных недостатков, развития функций Application delivery, требований к безопасности и инфраструктурной совместимости.

3.2 Устранение сбойных ситуаций и дефектов

ООО «Сервиспайп» обеспечивает регистрацию и обработку обращений, связанных с некорректной работой программного обеспечения, ошибками установки, отклонениями в работе runtime-слоя, API управления, интерфейса администратора, логирования, статистики или сетевых функций.

Процесс устранения сбойной ситуации включает:

- регистрацию обращения или внутреннего инцидента;
- первичную классификацию по влиянию на доступность, безопасность и функциональность;
- сбор диагностических данных, журналов, конфигураций и описания сценария;
- анализ и, при необходимости, воспроизведение ситуации на тестовом стенде;
- определение причины сбоя или дефекта;
- подготовку корректирующего решения, настройки обходного решения или рекомендаций;
- тестирование исправления и регрессионную проверку затронутых функций;
- выпуск корректирующего обновления либо включение исправления в плановый релиз;
- информирование заявителя о результате обработки обращения.

Если исправление требует изменения исходного кода, соответствующие изменения проходят стандартные стадии жизненного цикла: разработку, сборку, тестирование и подготовку релиза. Если причина связана с конфигурацией или эксплуатационной средой, пользователю предоставляются рекомендации по настройке или восстановлению работоспособности.

3.3 Эксплуатационная и консультационная поддержка

Эксплуатационная поддержка включает консультации по установке, обновлению, настройке и использованию Servicepipe ADC. Пользователям предоставляются рекомендации по созданию маршрутов, потоковых маршрутов, upstream-пулов, health checks, сертификатов, секретов, потребителей, групп потребителей и конфигураций плагинов.

Специалисты сопровождения помогают анализировать типовые проблемы эксплуатации: недоступность backend-нод, некорректный выбор маршрута, ошибки TLS, проблемы с health checks, нарушения формата JSON-конфигурации, ошибки авторизации API, отсутствие статистики или некорректную доставку логов.

Обращения пользователей принимаются по установленным каналам связи службы технической поддержки ООО «Сервиспайп», включая договорные каналы взаимодействия и электронную почту организации.

3.4 Документирование

Документация на программное обеспечение формируется и поддерживается в актуальном состоянии на протяжении всего жизненного цикла. Документация отражает назначение продукта, функциональные характеристики, порядок установки, эксплуатации, настройки, диагностики и сопровождения программного обеспечения.

Процесс документирования включает:

- разработку пользовательской и эксплуатационной документации;
- подготовку описания функциональных характеристик;
- описание технических средств хранения исходного и объектного кода;
- описание процессов жизненного цикла программного обеспечения;
- актуализацию документации при выпуске новых версий;
- контроль соответствия документации реализованной функциональности;
- хранение актуальных и архивных версий документации.

Изменения функциональности, интерфейсов, API, installer-a, эксплуатационных процедур и состава поставки сопровождаются внесением соответствующих изменений в документацию.

3.5 Обучение и поддержание квалификации персонала

Персонал, участвующий в разработке и сопровождении Serviceripe ADC, обладает профильной квалификацией в области разработки программного обеспечения, сетевых технологий, эксплуатации серверной инфраструктуры, информационной безопасности, тестирования, технической поддержки и документирования.

ООО «Сервиспайп» обеспечивает поддержание квалификации персонала за счет внутреннего обучения, обмена опытом, участия специалистов в проектировании новых функций, анализа инцидентов, изучения новых технологий и обновления внутренних инструкций при внедрении новых архитектурных решений или инструментов.

Поддержание квалификации направлено на обеспечение надежности, сопровождаемости, безопасности и устойчивого развития программного обеспечения.

3.6 Обеспечение безопасности и устойчивости

В рамках обеспечения жизненного цикла выполняются мероприятия, направленные на снижение рисков, связанных с изменением программного обеспечения, обработкой конфигурации, поставкой релизных артефактов и эксплуатацией ADC в инфраструктуре заказчика.

К таким мероприятиям относятся ограничение доступа к репозиториям и хранилищам артефактов, контроль изменений исходного кода, фиксация состава версий и зависимостей, проверка корректности конфигураций, тестирование механизмов отказоустойчивости, учет требований к TLS, сертификатам, секретам, логированию и доступу к административным интерфейсам.

Для критичных изменений предусматривается дополнительная проверка влияния на доступность сервисов, обработку пользовательского трафика, совместимость с существующими конфигурациями и возможность безопасного обновления или отката.

4 Управление конфигурацией, исходным кодом и релизными артефактами

4.1 Хранение исходного кода

Исходный код проекта Serviceripe ADC хранится в группе локальных Git-репозиториях на собственной инфраструктуре ООО «Сервиспайп». Технические средства хранения исходного кода расположены на собственных серверах, размещенных в датацентре в г. Москве. Внешние SaaS-репозитории не используются как основное место хранения исходного текста заявляемого программного обеспечения.

Основные репозитории исходного кода и материалов проекта:

Репозиторий / компонент	Назначение
adc-runtime-core	Собственный runtime-слой ADC: Lua-модули, C/C++ модули, дополнительные сервисы, патчи NGINX-ядра, API-адаптеры,

	конфигурации, тесты, документация и каталог зависимостей.
adc-lk-backend	Backend личного кабинета и API управления ADC: Node.js, TypeScript, NestJS, GraphQL/REST, прикладная логика управления пользователями, настройками, журналами, отчетами и интеграциями runtime-слоя.
adc-lk-frontend	Frontend личного кабинета: React, TypeScript, Vite, Apollo GraphQL, React Query и UI для управления публикацией приложений, настройками, журналами, аналитикой и диагностикой.
adc-deploy-installer	Installer, bootstrap-скрипты, Ansible playbook-и, inventory-шаблоны, manifest версий, Makefile/скрипты упаковки, шаблоны конфигураций и материалы развертывания.

В Git-репозиториях хранятся исходный текст, конфигурации сборки, тесты, документация, сведения о зависимостях и release manifest. Изменения фиксируются в истории репозитория и позволяют восстановить состав исходного кода конкретной версии.

4.2 Хранение объектного кода и поставочных артефактов

Объектный код и релизные артефакты отделены от исходных Git-репозиториях и хранятся в поставочных хранилищах артефактов. Такой подход обеспечивает разделение процесса разработки исходного текста и процесса распространения готовых пакетов, образов и материалов поставки.

Хранилище	Тип артефактов	Назначение
adc-packages-repository	RPM/пакеты	Хранение пакетов Servicepipe ADC и включаемых сторонних компонентов, используемых installer-ом.
adc-files-repository	Архивы, images, binary payload	Хранение файлов поставки, архивов, Docker/OCI-образов и иных материалов, используемых installer-ом.

Релизные артефакты формируются из контролируемого состояния исходного кода и зависимостей. Для каждого релиза фиксируются версия, состав компонентов, используемые зависимости и материалы развертывания.

4.3 Контроль изменений и трассируемость

Контроль изменений обеспечивается за счет ведения задач разработки, фиксации изменений в Git-репозиториях, подготовки release manifest-a и ведения сведений о составе релиза. Каждое изменение связывается с соответствующим требованием, дефектом, задачей или внутренней инициативой.

Для изменений, затрагивающих обработку трафика, схему конфигурации, API управления, installer или механизмы безопасности, проводится дополнительный анализ влияния. По результатам анализа определяются необходимые тесты, миграции, изменения документации и порядок внедрения.

Трассируемость позволяет определить, какие изменения исходного кода и конфигурации вошли в конкретную версию, какие артефакты были собраны и какие проверки были выполнены перед выпуском.

5 Персонал, обеспечивающий жизненный цикл ПО

Работы по разработке, сопровождению и развитию ПО «Servicepipe ADC» выполняются специалистами ООО «Сервиспайп», обладающими соответствующей квалификацией и опытом. В жизненном цикле участвуют продуктовые, инженерные, тестовые, эксплуатационные и технические роли.

Роль	Участие в жизненном цикле	Квалификация
Руководитель продукта	Формирует требования к программному обеспечению, определяет направления развития продукта, согласует функциональный состав версий, участвует в приемке результатов и подготовке релизов.	Высшее образование; опыт управления продуктом, анализа требований и взаимодействия с пользователями программного обеспечения.
Руководитель разработки / технический лидер	Организует разработку, участвует в архитектурных решениях, распределяет задачи, контролирует качество реализации и согласованность изменений между компонентами.	Высшее профильное образование; опыт разработки программного обеспечения и проектирования серверных систем.
Архитектор / ведущий разработчик runtime-слоя	Проектирует и реализует компоненты обработки трафика, runtime-модули, патчи, API-адаптеры и интеграции с gateway-уровнем.	Высшее профильное образование; опыт разработки сетевого, серверного или высоконагруженного программного обеспечения.
Backend-разработчик	Разрабатывает backend личного кабинета, GraphQL/REST API управления, бизнес-логику, интеграции, миграции данных и сервисные задачи.	Высшее профильное образование; опыт разработки backend-сервисов, API и работы с базами данных.
Frontend-разработчик	Разрабатывает пользовательский интерфейс Application delivery, формы управления сущностями ADC, таблицы, валидацию, интеграцию с API и отображение статистики.	Высшее профильное образование; опыт разработки web-интерфейсов на React/TypeScript или аналогичных технологиях.
Инженер DevOps / инженер сборки и поставки	Поддерживает installer, bootstrap-скрипты, Ansible playbook-и, manifest-ы версий, сборку пакетов и размещение артефактов.	Высшее профильное образование; опыт Linux-администрирования, автоматизации сборки, контейнеризации и сопровождения инфраструктуры.
Инженер по тестированию	Проводит функциональное, интеграционное, регрессионное, API-, UI- и инсталляционное тестирование, проверяет исправления и готовность версии к выпуску.	Высшее профильное образование; опыт тестирования серверного, web- или инфраструктурного программного обеспечения.
Инженер внедрения / эксплуатации	Участвует в установке, обновлении, настройке, диагностике и проверке работоспособности ADC в	Высшее профильное образование; опыт эксплуатации Linux, сетевых сервисов, балансировщиков, TLS и систем мониторинга.

	стендовой или клиентской инфраструктуре.	
Инженер технической поддержки	Принимает и классифицирует обращения пользователей, собирает диагностические данные, взаимодействует с разработчиками при устранении дефектов и консультирует пользователей.	Высшее профильное образование; опыт технической поддержки серверного или сетевого программного обеспечения.
Технический писатель / системный аналитик	Готовит и актуализирует пользовательскую, эксплуатационную и техническую документацию, описания функций, инструкций и процессов.	Высшее образование; опыт подготовки технической документации и анализа программных продуктов.

Фактический количественный состав специалистов определяется штатным расписанием, планом работ и составом версии. При необходимости к работам привлекаются специалисты смежных направлений: информационной безопасности, сетевой инфраструктуры, мониторинга, сопровождения баз данных и администрирования Linux-систем.

6 Инфраструктура разработки, сборки, тестирования и поддержки

6.1 Инфраструктура хранения и разработки

Инфраструктура разработки программного обеспечения Servicepipe ADC включает внутренние Git-репозитории, серверы разработки, средства управления задачами, хранилища документации, средства сборки, тестовые стенды и поставочные репозитории артефактов.

Технические средства хранения исходного кода расположены на собственных серверах ООО «Сервиспайп», размещенных в датацентре в г. Москве по адресу: ул. Бутлерова, д. 7, ЦОД Ростелеком. Доступ к инфраструктуре предоставляется уполномоченным сотрудникам в соответствии с их ролями и необходимостью участия в работах по жизненному циклу.

Разработчики программного обеспечения осуществляют деятельность в рамках организационного контура ООО «Сервиспайп». Юридический адрес организации: Россия, 111024, г. Москва, ул. Кабельная 2-я, д. 2, стр. 3, эт. 2, помещение XVIII, ком. 8.

6.2 Инфраструктура сборки и тестирования

Инфраструктура сборки обеспечивает формирование объектного кода, RPM-пакетов, Docker/OCI-образов, архивов, binary payload, installer-a и сопроводительных материалов поставки. Сборка выполняется из контролируемого состояния исходного кода и фиксированного набора зависимостей.

Тестовые стенды используются для проверки runtime-слоя, backend и frontend компонентов, API управления, installer-a, интеграции с базами данных, хранилищами метрик и логов, а также для проверки сетевых сценариев публикации HTTP(S), gRPC, TCP, TLS и UDP-сервисов.

Для диагностики и проверки используются журналы, метрики, средства мониторинга, инструменты анализа сетевого трафика, тестовые backend-сервисы и сценарии, имитирующие типовую эксплуатацию ADC в инфраструктуре заказчика.

6.3 Инфраструктура технической поддержки

Техническая поддержка программного обеспечения осуществляется службой технической поддержки ООО «Сервиспайп». Служба поддержки обеспечивает прием, регистрацию, классификацию и обработку обращений пользователей, связанных с установкой, настройкой, эксплуатацией, диагностикой и устранением сбойных ситуаций.

Размещение службы технической поддержки осуществляется в организационном контуре ООО «Сервиспайп». Обращения пользователей принимаются по установленным договором каналам связи, а также через публичные контактные каналы организации, включая электронную почту welcome@servisepipe.ru.

Инфраструктура поддержки включает средства регистрации обращений, внутренней коммуникации, хранения эксплуатационных материалов, доступа к документации, сбора диагностической информации и передачи задач в разработку при необходимости выпуска исправления.

Настоящее описание процессов жизненного цикла применяется к текущей и последующим версиям программного обеспечения Servisepipe ADC и подлежит актуализации при существенном изменении архитектуры продукта, состава компонентов, модели поставки, инфраструктуры разработки или процессов сопровождения.