

УТВЕРЖДЕН  
ru.red-soft.0000001-03 36 01

Система интеллектуальных помощников «Цифровая приемная»

Описание жизненного цикла и сопровождение продукта

ru.red-soft.0000001-03 36 01

Листов 11

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата

## **АННОТАЦИЯ**

Система интеллектуальных помощников «Цифровая приемная» (СИП «Цифровая приемная») – программное обеспечение для автоматизации процесса обработки входящей, исходящей и внутренней корреспонденции. СИП «Цифровая приёмная» использует технологии «Машинное зрение» и «Обработка текстов на естественных языках» (NLP). СИП «Цифровая приемная» предполагает информационное взаимодействие со смежными информационными системами по технологии REST API.

В настоящем документе представлено описание системы управления жизненным циклом разработки и сопровождения СИП «Цифровая приемная», которая используется в ООО «Ред Софт» при разработке СИП «Цифровая приемная».

**СОДЕРЖАНИЕ**

<b>1. ПО ДЛ</b>	<b>4</b>
1.1. Общие сведения.....	4
1.2. Доступ к системе контроля версий.....	4
1.3. Возможности системы контроля версий Git.....	4
1.4. Возможности системы жизненного цикла GitLab.....	5
1.5. Возможности системы автоматизации развертывания и управления приложениями Docker.....	5
1.6. Возможности системы оркестрации Docker Compose .....	6
<b>2. МОДЕЛЬ РАБОТЫ.....</b>	<b>7</b>
<b>3. ЖИЗНЕННЫЙ ЦИКЛ ЗАДАЧ .....</b>	<b>9</b>
<b>4. ВНЕДРЕНИЕ И СОПРОВОЖДЕНИЕ .....</b>	<b>10</b>
<b>Перечень принятых сокращений .....</b>	<b>11</b>

## 1. ПО ДЛЯ ОБЕСПЕЧЕНИЯ ЖИЗНЕННОГО ЦИКЛА И СОПРОВОЖДЕНИЯ

### 1.1. Общие сведения

Для обеспечения жизненного цикла разработки и сопровождения СИП «Цифровая приёмная» используется широкий ряд различного ПО:

При разработке СИП «Цифровая приёмная» для обеспечения контроля версий применяется программный продукт Git, который является свободно распространяемой по лицензии GNU/GPL v2 распределенной системой управления версиями. В РЕД СОФТ используется версия Git не ниже 2.0.

Для управления центральным репозитарием используется система GitLab, специальный инструмент жизненного цикла DevOps с открытым исходным кодом, представляющий систему управления репозиториями кода для Git. В текущий момент для разработки СИП «Цифровая приёмная» используется версия GitLab Community Edition 13.1.3.

Для создания версий для тестирования и релизов используется Docker, это специализированное программное обеспечение для автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации. Для разработки СИП «Цифровая приёмная» используется версия Docker не ниже 17.

Для удобного управления несколькими контейнерами используется система оркестрации контейнеров Docker Compose. В основном используется самая актуальная версия системы.

### 1.2. Доступ к системе контроля версий

В ООО «Ред Софт» доступ к репозиториям системы Git и к системе GitLab обеспечивается из контролируемых зон объектов ООО «Ред Софт» удаленно с использованием шифруемых протоколов Secure Shell (SSH), HTTPS. Доступ разработчиков обеспечивается внутри корпоративной виртуальной частной сети (VPN), организуемой сертифицированными средствами межсетевого экранирования и построения VPN каналов.

### 1.3. Возможности системы контроля версий Git

Основными возможностями системы Git являются:

- возможность расширения функционала. Система спроектирована как набор программ, специально разработанных с учетом их использования в сценариях, это позволяет создавать специализированные системы контроля версий на базе Git и пользовательские интерфейсы;
- сохранение полной истории изменений файлов с возможностью возврата к предыдущим версиям. В истории сохраняются контрольные суммы изменений, поэтому она устойчива от внесения правок, а сам репозиторий представляет собой не сколько файлы, сколько изменения файлов;
- система поддерживает очень быстрое создание веток, это в том числе влияет на модель разработки продукта при использовании системы Git;
- синхронизация изменений между пользователями и автоматическое слияние изменений;
- репозиторий Git является распределенным, а потому устойчивым от потерь;

- возможность фиксации изменений в ветках. Возможность пометки определенных изменений тегами.

#### 1.4. Возможности системы жизненного цикла GitLab

GitLab это интегрированный продукт, который автоматизирует весь жизненный цикл разработки

программного обеспечения. GitLab расширяет возможности Git. Ключевые возможности GitLab:

- организация публичных и частных репозиторий;
- облачный хостинг;
- неограниченное число пользователей, расширенное управление правами, группами;
- импорт проектов;
- вики-страницы;
- интеграция по API;
- проставление лейблов и вет;
- использование поиска и шаблонов;
- комментирование, объединение;
- встроенные возможности для CI / CD автоматизации релизов — GitLab-CI;
- отслеживание изменений и прогресса;
- отслеживание времени;
- запросы на слияние кода;
- возможность проведения рецензирования кода.

#### 1.5. Возможности системы автоматизации развертывания и управления приложениями Docker

Для ускорения и одновременно упрощения и улучшения качества продуктов в РЕД СОФТ используется система контейнеризации Docker. Эта система используется для развертывания СИП «Цифровая приёмная» в тестовой, разработки и в продуктовой средах.

Docker – это программная платформа для быстрой разработки, тестирования и развертывания приложений. Система койтенеризации Docker упаковывает ПО в стандартизованные блоки, которые называются контейнерами. Каждый контейнер включает все необходимое для работы приложения: библиотеки, системные инструменты, код и среду исполнения. Благодаря Docker можно быстро развертывать и масштабировать приложения в любой среде.

Ключевые возможности Docker:

- минимальное потребление ресурсов, по сравнению с виртуальной машиной. Контейнеры не виртуализируют всю операционную систему (ОС), а используют ядро

хоста и изолируют программу на уровне процесса. Последний потребляет намного меньше ресурсов локального компьютера, чем виртуальная машина;

- быстрая доставка контейнеров/образов. Быстрое развертывание ПО. Легкая переносимость приложения на основе контейнеров Docker можно эффективно и быстро переносить в разные системы, даже разные операционные системы;
- изоляция ПО. Процессы эффективно изолируются друг от друга, также ОС изолируется от контейнеров, что минимизирует риски нарушения безопасности и работоспособности инфраструктуры;
- простота работы. Упрощает задачи настройки, доставки, развертывания и масштабирования ПО. Повышает качество установки ПО и минимизирует человеческие ошибки;
- стандартизация операций. Все операции по работе с контейнерами и образами Docker стандартизированы;
- поддерживается системами непрерывной интеграции. В частности, поддерживается системой GitLab-CI.

#### 1.6. Возможности системы оркестрации Docker Compose

К основным возможностям системы оркестрации Docker Compose можно отнести:

- простота настройки и использования. Все настройки хранятся в отдельной папке с одним файлом конфигурации – `docker-compose.yml`. Управление сервисами: запуск, установка, настройка происходят простыми операциями и занимают минимальное время. Также возможно при этом организовывать сложные взаимосвязанные системы на базе нескольких приложений описанием в одном файле настроек;
- возможно создание нескольких изолированных сред на одном хосте, что также полезно при масштабировании систем.

## 2. МОДЕЛЬ РАБОТЫ

Общий процесс подготовки новых версий СИП «Цифровая приёмная» в целом соответствует схеме (см. рис. 1).

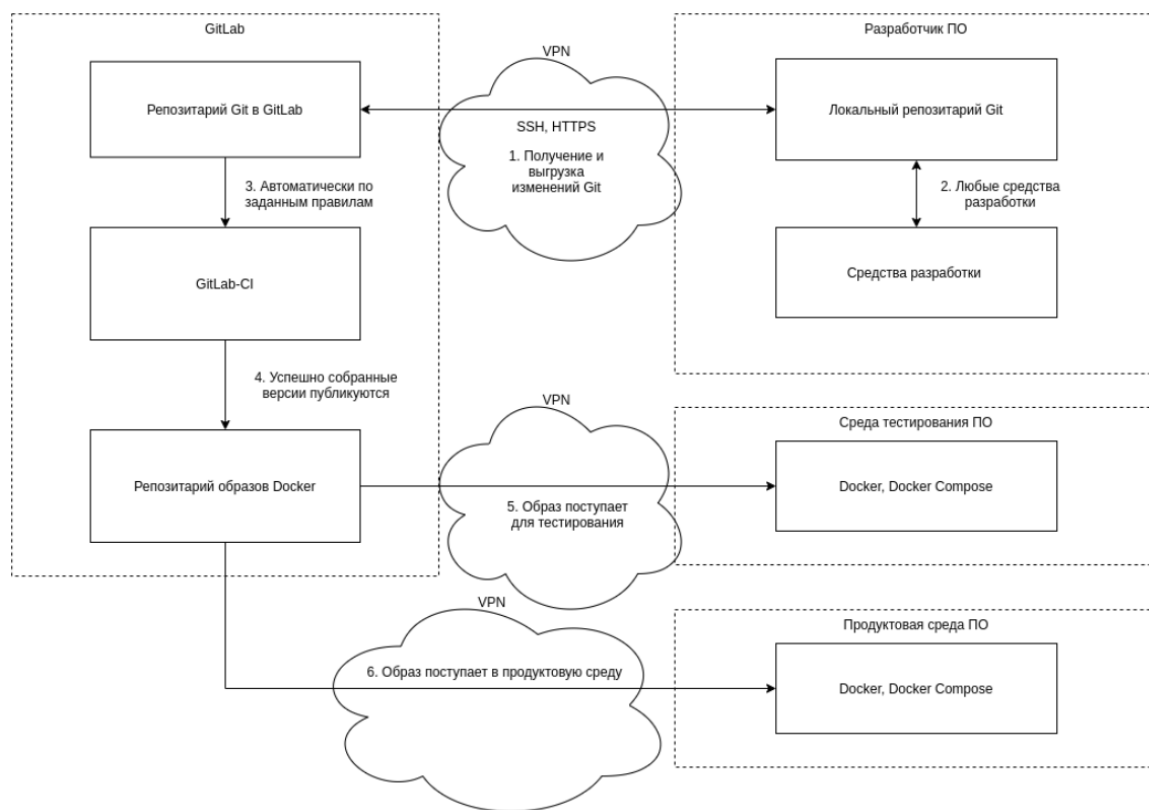


Рисунок 1

2.1. Актуализация исходного кода (см. рис. 1 шаг 1) — осуществляется клонированием репозитория продукта или обновлением локального репозитория Git в основной ветке разработки (master). Этот этап осуществляется или консольным клиентом Git или любым другим клиентом Git.

2.2. Разработчик создает новую ветку от актуализированной основной ветки разработки для внесения изменений в исходный код продукта средствами клиента Git.

2.3. Разработчик вносит изменения в исходный код программы (см. рис. 1 шаг 2) любыми удобными для себя средствами разработки.

2.4. Разработчик фиксирует изменения и отправляет их в удаленный репозиторий Git на сервере GitLab (см. рис. 1 шаг 1), создавая при этом в удаленном репозитории новую ветку – копию локальной ветки исходного кода, созданной в пункте п. 2.2.

2.5. Разработчик локально тестирует все изменения в версии программы.

2.6. Когда накопится достаточно изменений для новой версии в нескольких ветках или в одной новой ветке, данные изменения сливаются в главную ветку master. Также в случае критических исправлений все изменения поступают в главную ветку проекта. Без тега подготавливаемая в ветке master версия становится основным кандидатом в релиз.

2.7. Система GitLab-CI для проекта СИП «Цифровая приёмная» настроена таким образом, что при поступлении новых изменений в ветку master запускается автоматический пайплайн на

сборку образа Docker новой версии ПО. Система СИП «Цифровая приёмная» состоит из нескольких репозитариев нескольких различных компонент. Так что слитие (merge) изменений в главную ветку (master) любого из проектов СИП «Цифровая приёмная» продуцирует автоматическое создание Docker образа с готовым ПО соотв. проекта (см. рис. 1 шаг 3).

2.8. В случае если произошла ошибка сборки в группу разработки проекта Telegram шлется уведомление об ошибке при сборке проекта.

2.9. Успешно созданные Docker образы компонент системы СИП «Цифровая приёмная» автоматически регистрируются в репозиторий образов Docker (см. рис. 1 шаг 4). Новые Docker образы компонент системы СИП «Цифровая приёмная» регистрируются в репозитории сборки Docker под тегами master (по имени ветки, для которой строился проект).

2.10. В тестовой среде происходит настройка запуска готового проекта СИП «Цифровая приёмная» (см. рис. 1 шаг 5). Для этого из репозитория образов Docker выкачиваются последние (актуальные) версии (версии помеченные тегом master) всех компонент системы СИП «Цифровая приёмная». Делается это командой docker pull в ручном режиме.

2.11. Тестовая среда СИП «Цифровая приёмная» запускается посредством ранее настроенной системы оркестрации Docker Compose.

2.12. Тестовое приложение СИП «Цифровая приёмная», состоящее из нескольких компонент, и они же кандидаты на новые версии тестируется (см. рис. 1 шаг 5). В случае выявленных проблем все шаги повторяются сначала для устранения ошибок и доработки функционала.

2.13. В случае если приложение готово к созданию версии, все ошибки устранены, все требования к новой версии выполнены: все репозитории ветки master всех компонент системы СИП «Цифровая приёмная» фиксируются тегом с номером версии в формате .vM.m.B., где M — главный номер версии, m — младший номер версии, B — .build., используется для указания минимальных изменений в проекте. Версии компонент повышаются.

2.14. Система GitLab-CI для проекта СИП «Цифровая приёмная» настроена таким образом, что при фиксации изменений в тагах запускается автоматический пайплайн на сборку образа Docker новой версии, но уже релиза с именем версии указанной в таге шаги 3-4 повторяются (см. рис. 1). В репозитории образов Docker фиксируются новые версии образов Docker отмеченные тегом в формате .vM.m.B.. Это ремизные сборки компонент системы СИП «Цифровая приёмная» и кандидаты на установку в продуктивную среду.

2.15. Данные образы поступают на установку в продуктивную среду выполнения системы СИП «Цифровая приёмная» (см. рис. 1 шаг 6), в ручном режиме, аналогично тому, как настраивается и запускается тестовая среда см. п. 2.11.



### **3. ЖИЗНЕННЫЙ ЦИКЛ ЗАДАЧ**

В применяемой при разработке СИП «Цифровая приёмная» используется следующая схема жизненного цикла задач:

3.1.1. Регистрация ошибки или требований к доработке системы в задаче GitLab. Прием заявок на доработку, а также регистрация ошибок организована через каналы Telegram.

3.1.2. Далее задачи распределяются по исполнителям, а также им назначается при необходимости срок выполнения.

3.1.3. В момент, когда разработчик приступает к выполнению задачи, задача переводится со статуса «Открытый» в статус «В работе».

3.1.4. При завершении задачи задача переводится со статуса «В работе» на статус «На проверке». Тогда же задача проверяется тестировщиком или аналитиком или заказчиком на тестовом стенде.

3.1.5. В случае успешного выполнения задачи она переводится на статус «Закрит». Если же при проверке задачи были выявлены ошибки, задача вновь переводится на статус «В работе».

#### **4. ВНЕДРЕНИЕ И СОПРОВОЖДЕНИЕ**

В период эксплуатации и сопровождения системы СИП «Цифровая приёмная» ООО «Ред Софт» оказывает помощь в установке, настройке, устранении недостатков в работе программного обеспечения, а также компания осуществляет обучение персонала, эксплуатирующего программное обеспечение.

Сопровождение Системы включает в себя такие работы как:

- исправление ошибок и устранение неполадок, возникающих при работе СИП «Цифровая приёмная»;
- оптимизация работы СИП «Цифровая приёмная» при различных условиях эксплуатации;
- обновление и доработка СИП «Цифровая приёмная» по требованиям заказчика;
- настройка резервного копирования данных СИП «Цифровая приёмная»;
- профилактические работы по обслуживанию;
- актуализация программной документации.

Техническая поддержка Системы осуществляется в формате консультирования пользователей и администраторов СИП «Цифровая приёмная» по вопросам установки, переустановки, администрирования и эксплуатации по каналам связи (телефону, электронной почте или письменно по запросу).

## ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ

API	Программный интерфейс приложения, интерфейс прикладного программирования
DevOps	Методология автоматизации технологических процессов сборки, настройки и развёртывания программного обеспечения
Docker	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации
Git	Свободно распространяемая по лицензии GNU/GPL v2 распределенная система управления версиями
GitLab	Веб-инструмент жизненного цикла DevOps с открытым исходным кодом, представляющий систему управления репозиториями кода для Git с собственной вики, системой отслеживания ошибок и другими функциями.
GNU/GPL v2	GNU General Public License (переводят как Универсальная общественная лицензия GNU (версия 2))
NLP	Natural Language Processing. Аналог термина «Обработка текстов на естественных языках» на английском языке
REST	Архитектурный стиль взаимодействия компонентов распределённого приложения в сети
Telegram	Кроссплатформенный мессенджер, позволяющий обмениваться сообщениями и медиафайлами многих форматов
ПО	Программное обеспечение
СИП «Цифровая приёмная»	Краткое наименование программного обеспечения Система интеллектуальных помощников «Цифровая приемная»