

# Документация по сопровождению программного обеспечения

---

## 1. Общие сведения

Настоящий документ содержит описание процессов, обеспечивающих поддержание жизненного цикла программного средства, включая разработку, сопровождение, устранение неисправностей, выпуск обновлений и совершенствование функциональности.

Программное средство представляет собой веб-приложение, состоящее из серверной и клиентской частей и предназначенное для автоматизации процессов медицинских организаций.

Архитектура программного средства включает следующие основные компоненты:

- серверную часть, реализованную на языке программирования **Python** с использованием фреймворка **Django**;
- клиентскую часть, реализованную с использованием **Angularjs**;
- систему управления базами данных **PostgreSQL**;
- систему кеширования **Redis**;
- систему контроля версий **Git** для управления исходным кодом;
- систему автоматизации сборки и публикации **Jenkins**;
- контейнерную инфраструктуру **Docker** для развертывания компонентов системы.

Исходный код программного средства хранится в репозитории Git и сопровождается системой контроля версий.

Сборка программного средства выполняется автоматически средствами Jenkins, включая формирование и публикацию контейнерных образов программного средства.

## 2. Цели поддержания жизненного цикла программного средства

Поддержание жизненного цикла программного средства осуществляется в целях:

- обеспечения стабильной и корректной работы программного средства;
- своевременного устранения выявленных неисправностей;
- совершенствования функциональности программного средства;
- повышения надежности, производительности и безопасности системы;
- обеспечения воспроизводимости процессов сборки и развертывания;
- выпуска обновлений и новых версий программного средства;

- адаптации программного средства к изменяющимся требованиям пользователей.

### 3. Процессы жизненного цикла программного средства

Жизненный цикл программного средства включает следующие процессы:

- разработка программного средства;
- управление исходным кодом;
- автоматизированная сборка и выпуск версий;
- тестирование изменений;
- эксплуатация программного средства;
- сопровождение и техническая поддержка;
- устранение неисправностей;
- совершенствование программного средства.

#### 3.1 Управление исходным кодом

Разработка программного средства осуществляется с использованием системы контроля версий **Git**.

Система контроля версий используется для:

- хранения исходного кода;
- управления изменениями;
- обеспечения совместной работы разработчиков;
- отслеживания истории изменений;
- восстановления предыдущих версий программного средства при необходимости.

Изменения исходного кода вносятся разработчиками в отдельные ветки репозитория. После проверки корректности изменений и их согласования изменения включаются в основную ветку разработки.

Использование системы контроля версий обеспечивает прослеживаемость всех изменений и позволяет контролировать процесс разработки и сопровождения программного средства.

#### 3.2 Процесс сборки программного средства

Сборка программного средства выполняется автоматически с использованием системы непрерывной интеграции **Jenkins**.

Процесс сборки включает следующие этапы:

1. получение актуальной версии исходного кода из репозитория Git;
2. установка зависимостей серверной части программного средства;

3. сборка клиентской части приложения;
4. подготовка серверной части приложения;
5. выполнение проверок корректности сборки;
6. формирование контейнерного образа программного средства;
7. публикация собранного программного продукта.

Использование автоматизированной сборки позволяет:

- обеспечить воспроизводимость сборки;
- сократить вероятность ошибок при сборке;
- ускорить выпуск обновлений программного средства.

### **3.3 Управление версиями программного средства**

Для программного средства ведётся управление версиями.

Каждая версия программного средства может включать:

- исправления выявленных неисправностей;
- улучшения существующих функций;
- реализацию новых функциональных возможностей;
- обновления компонентов программной инфраструктуры.

Информация о внесённых изменениях фиксируется в системе управления исходным кодом и сопровождается описанием изменений.

Выпуск новой версии сопровождается автоматизированной сборкой программного средства и публикацией новой версии программного продукта.

### **3.4 Устранение неисправностей**

Неисправности программного средства могут выявляться:

- пользователями программного средства;
- специалистами технической поддержки;
- в ходе тестирования;
- в процессе эксплуатации системы.

Процесс устранения неисправностей включает следующие этапы:

1. регистрация обращения пользователя;
2. анализ поступившей информации;
3. локализация причины возникновения неисправности;
4. разработка исправления;

5. проверка корректности исправления;
6. включение исправления в очередную версию программного средства.

Исправления могут выпускаться как в составе плановых обновлений, так и в виде отдельных исправлений при необходимости.

### 3.5 Совершенствование программного средства

Совершенствование программного средства осуществляется на основании:

- обращений пользователей;
- предложений специалистов сопровождения;
- изменений требований к программному средству;
- необходимости повышения производительности и безопасности.

Совершенствование может включать:

- реализацию новых функций;
- оптимизацию производительности системы;
- улучшение пользовательского интерфейса;
- повышение надежности и устойчивости системы.

Все изменения проходят этапы разработки, проверки и включаются в очередную версию программного средства.

### 3.6 Обработка обращений пользователей и техническая поддержка

Техническая поддержка пользователей программного средства осуществляется посредством электронной почты и системы регистрации обращений.

Пользователи могут направлять обращения следующими способами:

- по электронной почте: [info@med-logic.ru](mailto:info@med-logic.ru)
- через систему регистрации обращений (тикетов): [support.med-logic.ru](https://support.med-logic.ru)

Обращения пользователей могут включать:

- сообщения о неисправностях программного средства;
- запросы на консультации по использованию системы;
- предложения по совершенствованию функциональности программного средства;
- запросы на изменение или расширение возможностей системы.

Все поступающие обращения регистрируются и анализируются специалистами сопровождения.

### 3.7 Тестирование изменений

Перед выпуском обновлений проводится проверка работоспособности программного средства.

Проверка включает:

- тестирование серверной части программного средства;
- проверку взаимодействия компонентов системы;
- тестирование пользовательского интерфейса;
- регрессионную проверку основных сценариев работы системы.

Проверка проводится специалистами разработки и сопровождения.

## 4. Информация о персонале, необходимом для сопровождения

Для сопровождения программного средства требуется персонал, обладающий следующими знаниями и компетенциями:

- знание языка программирования **Python**;
- опыт разработки с использованием фреймворка **Django**;
- знание технологий **Angularjs, TypeScript, HTML, CSS**;
- навыки работы с системой контроля версий **Git**;
- навыки сопровождения процессов автоматизированной сборки **Jenkins**;
- опыт сопровождения веб-приложений.

В сопровождении программного средства могут участвовать следующие специалисты:

Роль	Основные функции
Backend-разработчик	сопровождение серверной части программного средства, исправление дефектов, разработка новых функций
Frontend-разработчик	сопровождение клиентской части приложения, реализация пользовательских интерфейсов
DevOps-инженер	сопровождение инфраструктуры сборки и развертывания программного средства
Специалист технической поддержки	обработка обращений пользователей, регистрация и первичный анализ проблем

В рамках небольших проектов допускается совмещение нескольких ролей одним специалистом при наличии необходимой квалификации.

## 5. Результаты процессов сопровождения

Результатами выполнения процессов сопровождения программного средства являются:

- исправления выявленных неисправностей;
- выпуск новых версий программного средства;
- улучшение функциональности системы;
- повышение надежности и стабильности работы;
- актуализация исходного кода и документации;
- повышение качества и удобства использования программного средства.