
title: Описание процессов поддержания жизненного цикла программного средства «Платформа управления бизнес-процессами DocHub»

Описание процессов поддержания жизненного цикла программного средства «Платформа управления бизнес-процессами DocHub» (далее также - DocHub)

1. Назначение документа

Документ описывает процессы поддержания жизненного цикла программного средства «Платформа управления бизнес-процессами DocHub».

Документ не содержит реквизитов организации, контактных данных и колонтитулов. Сведения об организации, которая осуществляет разработку, совершенствование, устранение неисправностей и техническую поддержку, указываются в комплекте документов отдельно.

2. Процессы реализации программного средства

2.1. Анализ требований и планирование

Требования к программному средству формируются на основании продуктовой дорожной карты, запросов пользователей, результатов эксплуатации и задач сопровождения. Для каждой существенной доработки определяются назначение, объем работ, критерии приёмки и ожидаемый результат.

2.2. Архитектура и разработка

Разработка выполняется по компонентам программного средства:

- Java/Spring Boot backend;
- серверная бизнес-логика на TypeScript/JavaScript;
- web-интерфейс на React/TypeScript;
- YAML-метаданные и системные данные;
- сборочные ресурсы, JAR и контейнерная поставка;
- внешние прикладные модули, подключаемые к платформе отдельно.

Исходный текст и конфигурация хранятся в системе контроля версий Git. Изменения выполняются в отдельных ветках или задачах, проходят проверку и включаются в релиз только после подтверждения готовности.

2.3. Сборка и тестирование

Сборка выполняется воспроизводимыми сборочными сценариями проекта и в CI/CD. В зависимости от состава изменения формируются Java-ресурсы, JAR, web-ресурсы, данные поставки и Docker-образ.

До выпуска выполняются:

- проверка сборки и состава поставки;
- unit- и интеграционные проверки изменённых компонентов;
- функциональное тестирование по согласованным сценариям;
- проверка запуска и работоспособности экземпляра;
- healthcheck и диагностические проверки;
- security-проверки поставки.

2.4. Выпуск и поставка

Релизу присваивается уникальная версия. Сведения о выпуске фиксируются в журнале изменений `CHANGELOG.md` и связываются с соответствующим состоянием репозитория.

Основные формы поставки:

- исполняемый JAR;
- Docker-образ;
- демонстрационный или проверочный комплект, если он предусмотрен поставкой;
- ZIP-модули для расширения платформы.

3. Процессы поддержки программного средства

3.1. Управление документацией

Документация создаётся, актуализируется и публикуется по мере изменения программного средства. Она включает функциональные характеристики, инструкции по установке и

эксплуатации, описание архитектуры, сведения о жизненном цикле, исходном коде, сборке и сопровождении.

3.2. Управление конфигурацией

Управление конфигурацией обеспечивает целостность программного средства и доступность согласованных версий для участников разработки и сопровождения. С помощью Git и связанных средств CI/CD:

- контролируются изменения исходного текста, метаданных и сборочных сценариев;
- фиксируются версии и составы выпусков;
- сохраняется история изменений;
- обеспечивается воспроизводимость сборки;
- контролируется соответствие поставки выбранной версии исходного текста;
- разделяются исходные тексты, промежуточные результаты и готовые артефакты.

3.3. Решение проблем

Проблемы идентифицируются, регистрируются, анализируются, контролируются и устраняются до закрытия обращения. Для анализа фиксируются версия программного средства, окружение, время возникновения, шаги воспроизведения, ожидаемый и фактический результат, журналы и иные диагностические материалы.

Классы проблем:

Класс	Описание
Блокирующий	Полный отказ работы, невозможность входа или выполнения базовых функций.
Высокий	Некорректная обработка данных или невозможность выполнения отдельного процесса.
Нормальный	Вопрос по работе программы или известный дефект с обходным решением.
Низкий	Ошибки форматирования, неточности интерфейса и прочие некритичные вопросы.

Решение проблемы может включать консультацию, подготовку обходного решения, исправление в коде, выпуск исправительной поставки или включение изменения в плановый релиз. Сроки реакции и устранения определяются договором и уровнем сервиса.

По повторяющимся и значимым проблемам проводится анализ коренных причин. По его итогам корректируются код, документация, тестовые сценарии или процессы разработки и поставки.

3.4. Техническая поддержка

Техническая поддержка принимает обращения пользователей, помогает определить причину неисправности, собирает диагностические сведения и передаёт результат в разработку при необходимости.

Каналы связи, режим работы, порядок эскалации и SLA, определяются договором и уровнем сервиса.

4. Совершенствование программного обеспечения

4.1. Общие положения

Совершенствование DocHub выполняется в соответствии с видением продукта, дорожной картой развития, результатами эксплуатации и заявками пользователей. Цель совершенствования — развитие функциональности, повышение надёжности, производительности, безопасности и удобства эксплуатации программного средства.

К совершенствованию относятся изменения:

- backend и интеграционных механизмов;
- серверной бизнес-логики;
- web-интерфейса;
- метамодели и системных данных;
- механизмов сборки, поставки, диагностики и эксплуатации;
- внешних прикладных модулей, использующих платформу.

4.2. Порядок выполнения изменений

Работы выполняются итерационно. Каждая итерация имеет определённый состав изменений и завершается выпуском версии либо переносом результата в следующий релиз.

Этап	Содержание	Результат
1. Инициирование	Регистрация запроса, проблемы или задачи развития.	Описание потребности и приоритет.
2. Анализ	Уточнение требований, ограничений, затрагиваемых компонентов и рисков.	Согласованные требования и критерии приёмки.
3. Проектирование	Выбор решения и оценка влияния на архитектуру, данные, интерфейсы и совместимость.	Описание решения и план изменения.
4. Реализация	Изменение исходного текста, метаданных, тестов и документации.	Изменения в контролируемой версии.
5. Проверка	Code review, сборка, функциональное, интеграционное и при необходимости нагрузочное тестирование.	Результаты проверок и решение о готовности.
6. Выпуск	Формирование артефактов, обновление версии и CHANGELOG.md , подготовка поставки.	Версионированный релиз.
7. Обратная связь	Контроль результата в эксплуатации и анализ новых обращений.	Подтверждение эффекта или новые задачи.

4.3. Контроль целостности изменений

В процессе совершенствования обеспечиваются:

- идентификация составных частей программного средства и их версий;
- доступ участников к согласованным модификациям и выпускам;
- регистрация статуса изменений и результатов проверок;
- согласованность исходного текста, метаданных, тестов и документации;
- хранение и поставка только проверенных составов;
- возможность восстановить предыдущее состояние при необходимости.

Существенные изменения должны иметь описание, критерии приёмки, сведения о проведённых проверках и обратимый порядок обновления.

5. Персонал

В поддержании жизненного цикла участвуют следующие постоянные роли:

- руководитель продукта или проекта;
- системный или бизнес-аналитик;
- разработчик Java/Spring Boot и интеграционных механизмов; разработчик TypeScript/JavaScript и метаданных;
- разработчик React/TypeScript-интерфейса;
- специалист сопровождения и диагностики БД, очередей и контейнеров; тестировщик;
- ответственное лицо технической поддержки.

Инфраструктура разработки, разработчики и службы поддержки располагаются по адресу 115114, г. Москва, вн.тер.г. муниципальный округ Замоскворечье, ул. Кожевническая, д. 10, стр. 1.

Отдельные специалисты могут работать удаленно по месту жительства в г. Москве.