

Документация по процессам жизненного цикла и поддержке

1. Модель жизненного цикла

- **Методология:** гибрид Scrum/Kanban.
- **Этапы:**
 1. Анализ требований и формирование бэклога.
 2. Проектирование интерфейсов и API.
 3. Разработка и unit-тестирование.
 4. Интеграционное тестирование с реальным роботом.
 5. Приёмочные испытания.
 6. Релиз и передача в сопровождение.
 7. Эксплуатация и обратная связь пользователей.

2. Управление требованиями

- Требования фиксируются в системе управления проектами. Для каждого требования создаётся карточка с описанием, критичностью и критериями приёмки.
- Изменения требований согласуются с владельцем продукта и фиксируются в журнале изменений.

3. Проектирование

- Архитектурные решения документируются: схемы взаимодействия UI и `RobotClient`, перечень API команд (`/api/connect`, `/api/run`, `/api/stop`, `/jog...`).

4. Разработка

- Среда разработки: IntelliJ IDEA, JDK 17.
- Код оформляется согласно стандартам Google Java Style, используются аннотации Lombok (`@Slf4j`, `@Setter`) для сокращения шаблонного кода.
- Обязателен code review минимум двумя разработчиками перед merge.

5. Тестирование

- **Unit-тесты:** покрывают логику клиента `RobotClient` (мокирование HTTP-запросов).
- **Интеграционные тесты:** проверка взаимодействия контроллеров `DeviceDialogController` и `DeviceControlController` с тестовым REST-эмулятором робота.
- **UI-тесты:** сценарии с использованием TestFX (автоматическое воспроизведение нажатий кнопок управления осями, проверка появления `Toast`).
- **Регрессионные тесты:** после каждого релиза.

6. Контроль качества

- CI/CD проверяет проект на сборку и запуск тестов (`mvn verify`).
- Статический анализ и анализ зависимостей.
- Ручная проверка на стенде с реальным устройством перед релизом.

7. Управление изменениями

- Все изменения оформляются через Change Request.
- Перед выкладкой формируется план изменений с указанием отката.
- Журнал сопровождения фиксирует дату, ответственного, версию, набор обновлённых модулей.