

Описание процессов жизненного цикла программного обеспечения

Программное обеспечение: драйвер банковских POS-терминалов для платформы Эвотор

Документ описывает процессы жизненного цикла: проектирование, разработка, тестирование, приобретение, поставка, эксплуатация, документирование, обучение и квалификация персонала, поддержка версий и доработка, устранение сбойных ситуаций.

Правообладатель обеспечивает полный цикл создания и сопровождения драйвера. Процессы выполняются последовательно при выпуске новой версии и параллельно на стадии эксплуатации (поддержка, устранение сбоев, обучение). Ниже каждый процесс описан применительно к данному ПО.

1. Состав процессов жизненного цикла

Процесс	Цель	Основной результат
Проектирование	Зафиксировать архитектуру, интерфейсы и сценарии обмена с Эвотор и терминалом	Архитектура слоёв, контракт IPaySystem, протоколы SA и UsbProxy
Разработка ПО	Реализовать драйвер как Android-приложение под SDK Эвотор	Исходный код, posconnector.ini, манифест VID/PID
Тестирование	Подтвердить оплату, возврат, сверку, USB/сеть и обработку ошибок	Протокол проверки на кассе и терминале
Приобретение	Получить ПО торговой точкой для использования на кассе Эвотор	Право на установку выбранного варианта поставки
Поставка	Передать заказчику установочный пакет конкретной версии	APK pbf_evotor_driver_<версия>_<flavor>.apk
Эксплуатация	Обеспечить безналичные операции на подключённом терминале	Рабочий экземпляр драйвера на кассе
Документирование	Сопроводить ПО описаниями архитектуры, функций и сопровождения	Комплект эксплуатационной и технической документации
Обучение и квалификация	Подготовить кассира, администратора и поддержку к работе с драйвером	Персонал, способный выполнять свои роли
Поддержка версий и доработка	Выпускать исправления и новые версии без потери совместимости с SDK	Новая версия APK (versionName / versionCode)
Устранение сбойных ситуаций	Восстановить работу кассы и разобрать причину сбоя	Устранённый инцидент, запись в журнале

2. Проектирование

На стадии проектирования определяются границы ПО: драйвер — плагин платёжной системы Эвотор, а не самостоятельная касса и не банковский хост. Фиксируются:

- структурные элементы и интерфейсы (AIDL SDK, IPaySystem, callback сокета и COM);
- слоистая архитектура с прокси-каналом 127.0.0.1:9876 (логический обмен не пишет в COM напрямую);
- сценарии USB и виртуального (сетевого) подключения;
- набор бизнес-операций: оплата (Sale), возврат (Return), сверка (Settlement);
- модель ошибок для кассира (PayResult с текстом на чеке, коды EnumErrorsDriver);

- требования к совместимости: SDK Эвотор, POS-терминалы, Android API 23+.

Результат проектирования — описание технической архитектуры, перечень бизнес-функций и правила соединения слоёв. Изменения архитектуры (новые VID/PID, виртуальное устройство, новый тип USB-кадра) проходят повторное проектирование до включения в разработку.

3. Разработка ПО

Разработка ведётся как Android-приложение (Java/Kotlin, Gradle). Сборка: minSdk 23, compileSdk/targetSdk 35, Java 17. Исходный код хранится в системе контроля версий. Реализуются модули:

- PBFaisinoService — точка входа, жизненный цикл экземпляров USB и виртуальных устройств;
- AisinoDevice — исполнение payment / payback / closeSession;
- POSConnectorInterface / SAPacket / InitParser — SA-протокол и posconnector.ini;
- PBFProxyProtocol и USB COM-менеджер — доставка кадров на терминал;
- настройки, журналирование, выгрузка логов TMS, обработка аварий.

Варианты поставки задаются product flavors (free, paid, freeLegacy, paidLegacy) с разными applicationId. Версия фиксируется полями versionName и versionCode. Имя артефакта: pbf_evotor_driver_<versionName>_<flavor>.apk.

4. Тестирование

Тестирование подтверждает, что драйвер выполняет бизнес-функции на целевой кассе и терминале. В проекте предусмотрены модульные (JUnit) и инструментальные (AndroidJUnitRunner) проверки. Основной объём — стендовые и приёмочные испытания на оборудовании.

Вид проверки	Что проверяется
Модульная / инструментальная	Сборка, контекст приложения, базовые автопроверки в составе Gradle.
USB-сценарий	Обнаружение по VID/PID, разрешение USB, готовность COM, оплата / возврат / сверка, отключение кабеля, повторное подключение.
Сетевой сценарий	Виртуальное устройство, прямой IP, сканирование сети и сверка TID, сохранение найденного адреса.
Негативные	Таймаут ACK и полного ответа, обрыв связи, параллельная операция, пустая/повреждённая posconnector.ini, отсутствие терминала в сети.
Результат на кассе	PayResult: код «0», RRN, слип; текст ошибки на чеке; отсутствие гонок за COM.
Диагностика	Запись логов, выгрузка ZIP по порту 10115 (команда 0x21), контрольная сумма MD5.

К выпуску допускается версия, на которой подтверждены БФ-01...БФ-03 (оплата, возврат, сверка) хотя бы в одном штатном режиме подключения (USB или сеть) и корректная выдача ошибки кассиру при типичных сбоях связи.

5. Приобретение

Приобретение — получение торговой точкой права использовать драйвер на кассе Эвотор. ПО не распространяется как исходный код для самостоятельной сборки конечным кассиром. Заказчик получает согласованный вариант поставки (бесплатный или платный flavor) и право установки APK на свои кассы Эвотор.

- субъект приобретения: организация — владелец касс Эвотор / интегратор;
- объект: драйвер конкретной версии и варианта (applicationId);
- условия: наличие кассы Эвотор, POS-терминала из поддерживаемого перечня VID/PID, USB или общая IP-сеть;
- после приобретения выполняется поставка и ввод в эксплуатацию (разделы 6 и 7).

6. Поставка

Поставка — передача заказчику установочного пакета и комплекта документации. Носитель поставки — файл APK, сформированный сборкой:

- pbf_evotor_driver_<версия>_<flavor>.apk;
- в поставку входят сведения о версии (versionName, versionCode, дата сборки) и инструкция по установке на Эвотор;
- установка выполняется штатными средствами кассы / магазина приложений Эвотор как драйвер категории PAYSYSTEM;
- после установки платформа регистрирует сервис PBFaisinoService по intent-filter SDK.

Повторная поставка (обновление) — новый APK с увеличенным versionCode. Совместимость по applicationId сохраняется в пределах выбранного flavor.

7. Эксплуатация

Эксплуатация начинается после установки и подключения терминала. Сервис при старте копирует posconnector.ini во внутреннее хранилище (если копии нет), восстанавливает конфигурацию, включает журналы и принимает экземпляры устройств.

7.1. Ввод экземпляра в работу

- USB: подключение кабеля → addUsbDevice → AisinoDevice, COM 115200 8N1, прокси :9876, состояние «готово».
- Сеть: addNewVirtualDevice, IP/порт из настроек; при недоступности адреса — поиск в сети по TID.

7.2. Штатная работа

- Кассир выполняет оплату, возврат, кассовый отчёт. Драйвер обменивается с терминалом и возвращает PayResult.
- Одновременно на экземпляре идёт только одна операция.
- Администратор меняет IP, порт, таймаут, TID, уровень логов на экране настроек.

7.3. Вывод экземпляра

- `destroy(instanceId)`: ожидание окончания операции (до 5 с), остановка прокси и сервера логов.
- Удаление приложения с кассы завершает эксплуатацию ПО; терминал как отдельное устройство не стирается.

8. Документирование

Документирование сопровождает все стадии цикла. Комплект по данному ПО включает:

Документ	Назначение
Описание технической архитектуры	Элементы, интерфейсы, соединения, архитектурный стиль
Функциональные характеристики	Режимы USB/сеть, операции SDK, ошибки, настройки, диагностика
Перечень бизнес-функций	БФ-01...БФ-10, соответствие кнопкам кассы Эвотор
Описание процессов жизненного цикла	Настоящий документ
Техническая документация драйвера	Протоколы SA и UsbProxy, сценарии, коды ошибок, логирование
Конфигурация <code>posconnector.ini</code>	Состав полей операций Sale / Return / Settlement
Документация SDK Эвотор	Контракт драйвера платёжной системы (внешний стандарт интеграции)

Документация обновляется при выпуске версии, меняющей архитектуру, набор функций, протоколы или порядок сопровождения. Версия драйвера отображается в настройках (PBF Driver v.<versionName>).

9. Обучение и квалификация персонала

Для работы с ПО не требуется разработка на Android. Квалификация определяется ролью.

Роль	Необходимая квалификация	Чему обучают
Кассир	Работа на кассе Эвотор (штатный интерфейс продажи и возврата)	Запуск оплаты / возврата / кассового отчёта; действия при тексте ошибки на чеке (повтор, проверка кабеля/сети, обращение к администратору)
Администратор кассы	Подключение периферии Эвотор, базовые сетевые сведения	Установка APK, USB-разрешение, настройки IP/порта/таймаута/TID, различие USB и виртуального устройства
Специалист сопровождения	Чтение журналов, понимание кодов <code>EnumErrorsDriver</code>	Выгрузка логов TMS (порт 10115, команда 0x21), типовые сбои связи, критерии эскалации разработчику
Разработчик ПО	Android, SDK Эвотор, протоколы SA / UsbProxy	Архитектура слоёв, правила доработки без нарушения контракта <code>IPaySystem</code>

Обучение кассира опирается на уже знакомый интерфейс Эвотор: отдельные «экранные» операции драйвера для оплаты не вводятся. Администратор и поддержка получают регламент ввода в эксплуатацию и перечень типовых неисправностей (раздел 11).

10. Поддержка версий и доработка

Каждая поставка идентифицируется `versionName` (для пользователя) и `versionCode` (для установки обновления). Доработка выполняется при:

- ошибках эксплуатации (таймауты, обрыв USB, разбор ответа, конфигурация);
- появлении новых моделей терминала (добавление VID/PID в манифест и ProbeTable);
- изменениях SDK Эвотор или протокола терминала (SA, UsbProxy, NonBtp);
- запросах на настройки, логирование, сетевой поиск TID.

Порядок доработки: регистрация изменения → проектирование затронутых слоёв → разработка → тестирование на кассе и терминале → обновление документации → выпуск APK → поставка заказчику. Обратная совместимость: контракт IPaySystem и категория PAYSYSTEM сохраняются; flavor (applicationId) при обновлении не меняется, чтобы касса приняла пакет как обновление, а не как другое приложение.

Поддерживается линейка установленных версий до перевода торговой точки на новый APK. Конфигурация на кассе (рабочая копия ini, IP, таймаут) при обновлении, как правило, сохраняется во внутреннем хранилище приложения, если не меняется applicationId.

11. Устранение сбойных ситуаций

Сбой — ситуация, когда операция не завершается успехом или терминал недоступен. ПО само фиксирует сбой (код, текст, журнал) и возвращает кассиру результат на чеке. Устранение идёт по уровням.

Уровень	Кто действует	Типовые ситуации и меры
Касса (автоматически)	Драйвер	Таймаут АСК/ответа; обрыв USB (устройство «не готово», попытка открыть порт снова); отказ терминала (ТВ — занят, МЕ — формат); конфликт операций; восстановление posconnector.ini при повреждении; запись CrashHandler.
Торговая точка	Кассир / администратор	Повтор операции; проверка кабеля USB и питания; проверка, что касса и терминал в одной сети; сверка IP/порта/TID в настройках; переподключение терминала; сообщение «Оборудование не готово».
Сопровождение	Техподдержка / TMS	Выгрузка ZIP-логов (порт 10115, команда 0x21, MD5); разбор кодов 100–600 EnumErrorsDriver; печать критической ошибки на принтере кассы; рекомендации по переустановке APK.
Разработка	Разработчик ПО	Дефект в обмене, разборе SA, прокси или USB. Исправление выпускается как новая версия (процесс поддержки версий).

Критерий закрытия сбоя: повторная успешная операция (оплата, возврат или сверка) либо подтверждённая недоступность оборудования вне зоны ответственности драйвера (терминал выключен, нет USB-разрешения, другая сеть). Все программные сбои оставляют след в журнале.

12. Взаимосвязь процессов

Проектирование, разработка и тестирование образуют цикл выпуска версии. Приобретение и поставка передают версию заказчику. Эксплуатация порождает потребность в документировании, обучении, поддержке версий и устранении сбоев. Устранённый дефект возвращается в разработку и тестирование и поставляется как обновление. Документирование и обучение обновляются вместе с версией, если изменились функции, настройки или порядок действий при сбое.