

Платформа видеоаналитики AISee для решения отраслевых
задач

ОПИСАНИЕ ПРОЦЕССОВ, ОБЕСПЕЧИВАЮЩИХ
ПОДДЕРЖАНИЕ ЖИЗНЕННОГО ЦИКЛА

2024

АННОТАЦИЯ

В настоящем документе приведено описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения «AISee» - системы мониторинга на кассах самообслуживания.

Настоящий документ содержит:

- Назначение и функциональные характеристики системы;
- Роли исполнителей, принимающих участие в процессах жизненного цикла программы;
- Модель жизненного цикла системы.

Описание жизненного цикла приведено в соответствии с ГОСТ Р ИСО/МЭК 12207-2010.

Содержание

1. Общие сведения.....	5
1.1. Назначение системы.....	5
1.2. Возможности системы.....	5
1.3. Состав системы.....	5
2. Роли исполнителей.....	6
3. Стадии жизненного цикла Системы.....	10
3.1. Перечень стадий жизненного цикла Системы.....	10
3.2. Стадия сбора и анализа требований.....	10
3.3. Стадия проектирования архитектуры и разработки.....	11
3.4. Стадия передачи.....	13
3.5. Стадия эксплуатации и сопровождения.....	14
3.6. Стадия изъятия и утилизации.....	16

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящем документе используются следующие термины:

Термин	Определение
БД	База данных
ЖЦ	Жизненный цикл
ПО	Программное обеспечение
ИТ	Информационные технологии
Поставщик	Организация или лицо, которое вступает в соглашение с приобретающей стороной на поставку продукта или услуги
Продукт, система	Программа для ЭВМ, обеспечивающая автоматизация деятельности
Разработчик	Организация, которая выполняет разработку задач (в том числе анализ требований, проектирование, приемочные испытания) в процессе жизненного цикла
СУБД	Система управления базами данных
ТП	Техническая поддержка
ИИ	Искусственный интеллект
Grafana	Платформа для мониторинга, анализа и визуализации данных

1. Общие сведения

1.1. Назначение системы

AISee – платформа видеоаналитики, которая одновременно получает видеопотоки с видеокамер клиента и данные из различных информационных систем. Искусственный интеллект, являющийся неотъемлемой частью AISee, в режиме реального времени производит различного рода манипуляции с потоками данных и выдаёт результат согласно бизнес-логике, заложенной в алгоритмы платформы.

1.2. Возможности системы

Система выполняет следующие функции:

1. Обработка видеопотока данных с видеокамер, установленных непосредственно над Кассами самообслуживания
2. Получение потока операций с сервера Кассы самообслуживания, совершаемых клиентом
3. Распознавание товара
4. Анализ и выявление подозрительных действий клиента
5. Вывод оповещения на Кассу самообслуживания о блокировке для клиентов, сотрудников сектора касс/клиентского сервиса
6. Вывод оповещения на АРМ мониторинговой комнаты
7. Мониторинг и предоставление отчетности

Для осуществления пунктов 3, 4 «AISee» с помощью ИИ производит действия по обнаружению и классификации отдельных товаров в кадре.

1.3 Состав системы

При проектировании Системы использовались принципы микросервисной архитектуры. Основные компоненты Системы представлены на рисунке 1.

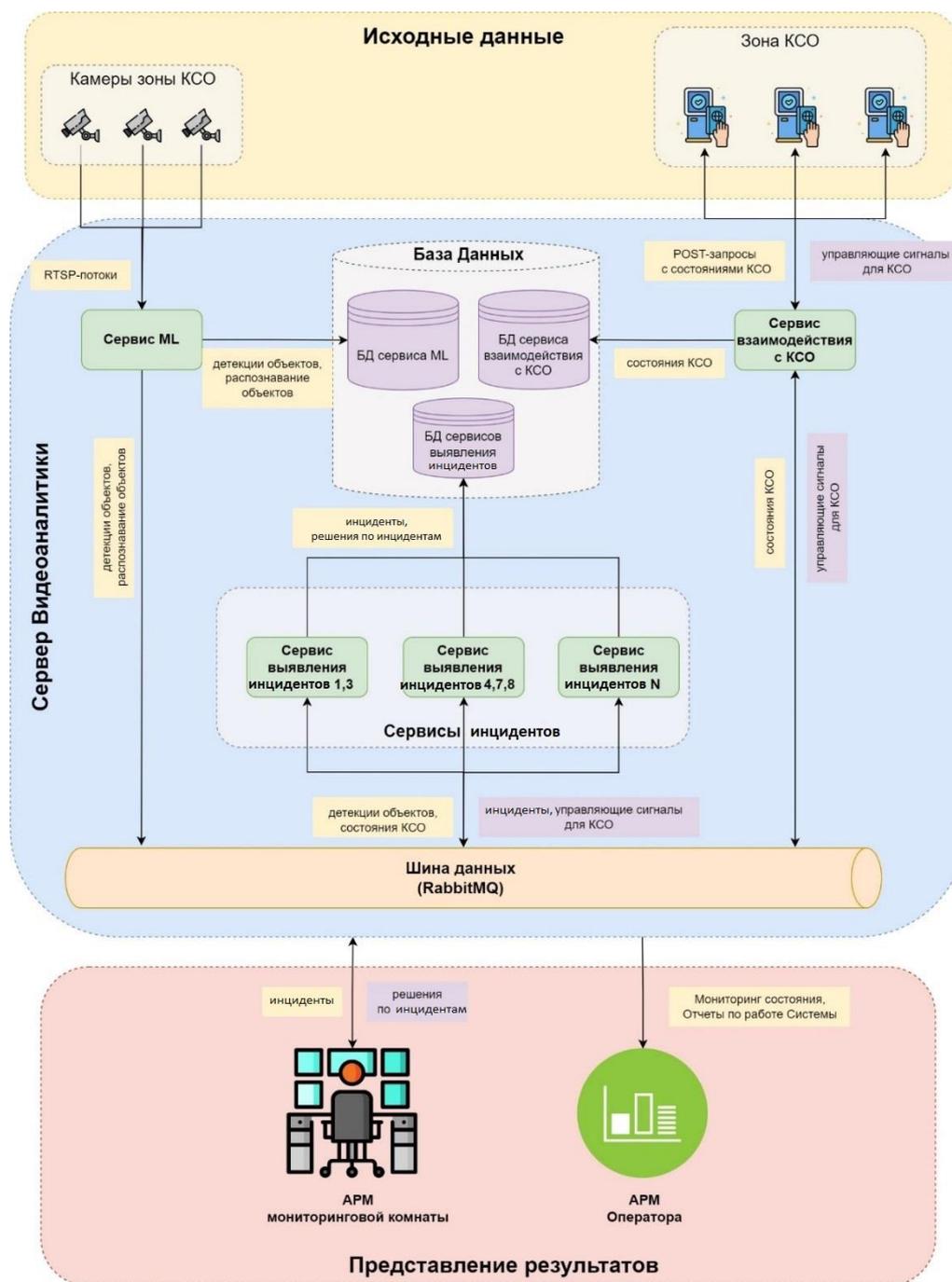


РИСУНОК 1. СХЕМА ПРОЕКТА.

Основными компонентами Системы являются:

1. Сервис ML

Обработывает поступающие видеопотоки с камер видеонаблюдения, проводит поиск объектов в кадре видеопотока для последующего использования (товары, сканеры штрих кодов, покупатели, покупательские тележки, корзинки и пр.), определяет категорию обнаруженных объектов.

2. Сервис взаимодействия с КСО

Получает состояние Кассы самообслуживания для последующего анализа, передает Кассе управляющие сигналы.

3. Сервис выявления инцидентов

Сопоставляет данные, полученные от Сервиса ML, с данными от Сервиса взаимодействия с КСО и оценивает вероятность того или иного инцидента.

4. Шина данных

Обеспечивает обмен данными между Сервисом ML, Сервисом взаимодействия с КСО и Сервисами выявления инцидентов.

5. База данных

Отвечает за хранение всех видов информации, с которой работает Система.

6. АРМ мониторинговой комнаты

Автоматизирует рабочее место мониторинговой комнаты для оперативного информирования сотрудников мониторинговой комнаты о выявленных инцидентах на Кассах самообслуживания.

7. АРМ оператора

Позволяет выводить статистическую и аналитическую информацию по результатам работы Системы в режиме реального времени, с формированием отчетов о выявленных инцидентах, обеспечивает возможность мониторинга работоспособности различных компонентов Системы в понятном графическом интерфейсе.

2. Роли исполнителей

1. Функциональный заказчик

В обязанности функционального заказчика входят:

- формирование основных бизнес-требований к Системе;
- контроль соответствия результата заданию.

2. Руководитель проекта

В обязанности руководителя проекта входят:

- управление процессом разработки Системы;
- планирование и согласование сроков и ресурсов;
- контроль соответствия работ согласованному с функциональным заказчиком плану проекта;
- формирование и предоставление функциональному заказчику промежуточных и итоговых отчетов о ходе разработки Системы;
- принятие решения о выпуске релиза Системы.

3. Ведущий программист (архитектор)

В обязанности ведущего программиста входят:

- принятие ключевых решений по архитектуре Системы;
- подготовка состава задач на реализацию изменений в исходных кодах;
- определение требований к исполнителям;
- определение требований к технологиям производства и инструментам разработки;
- разработка и внесение изменений в проектную документацию.

4. Программист

В обязанности программиста входят:

- реализация функциональности подсистем или модулей Системы согласно заданию Ведущего программиста и в соответствии с планом, подготовленным руководителем проекта;
- первичный контроль качества и работоспособности разрабатываемых компонентов Системы;
- подготовка технического описания реализации функциональных возможностей разрабатываемых компонентов Системы;
- устранение ошибок ПО и исправление дефектов, обнаруженных в процессе тестирования;
- предоставление отчётов о состоянии выполнения заданий.

5. Инженер тестирования

В обязанности инженера тестирования входят:

- проверка работоспособности Системы в соответствии с техническими условиями;
- подготовка стендов для проверки работоспособности Системы в соответствии с набором сценариев использования;
- подготовка отчётов о составе и опасности обнаруженных в ПО дефектов;
- воспроизведение на стендах проблем, обнаруженных инженером технической поддержки;
- планирование сроков и ресурсов на выполнение тестирования ПО;
- предоставление отчётов о результатах тестирования Системы.

6. Инженер поддержки процесса и инфраструктуры разработки ПО

В обязанности инженера поддержки процесса и инфраструктуры разработки ПО входят:

- поддержка и развитие сборочной системы (сборка исполняемых модулей и библиотек, запуск различных утилит для подготовки дистрибутива продукта);
- развитие утилит контроля регламента разработки (качество кода, контроль переводов ресурсов, контроль информации о версии, контроль настроек проектов и т.п.);

- стандартизация окружения разработки;
- настройка инфраструктуры с учётом особенностей разрабатываемого ПО;
- поддержка разработки, основанной на отдельных ветках под каждую новую функцию продукта;
- создание шаблонов сборки для веток;
- выполнение регламента работы с итерационными ветками разработки.

7. Аналитик

В обязанности аналитика входят:

- формирование системных требований к продукту;
- контроль соответствия результата системным и бизнес-требованиям;
- разработка технической и пользовательской документации на Систему;
- документирование изменений, производимых в Системе;
- контроль документов проекта на соответствие принятому стандарту документирования;
- принятие участия в решении запросов, переданных во вторую линию поддержки.

8. Инженер технической поддержки

В обязанности инженера технической поддержки входят:

- приём и маршрутизация заявок от пользователей Системы;
- консультация пользователей по возникающим вопросам, связанным с Системой;
- управление критическими инцидентами и оповещение пользователей о статусе заявки;
- сопровождение и участие в обновлении установленного ПО;
- диагностика и устранение неисправностей.

3. Стадии жизненного цикла Системы

3.1. Перечень стадий жизненного цикла Системы

В процессе развития Системы используется спиральная модель жизненного цикла и релизный подход при обновлении. Этапы жизненного цикла Системы проходят несколько итераций. Результат каждой итерации представляет собой очередную версию программного продукта, которая постоянно совершенствуется.

Модель жизненного цикла Системы включает в себя следующие стадии:

- 1) стадия сбора и анализа требований;
- 2) стадия проектирования архитектуры и разработки;
- 3) стадия передачи;
- 4) стадия эксплуатации и сопровождения;
- 5) стадия изъятия и утилизации.



РИСУНОК 2. СПИРАЛЬНАЯ МОДЕЛЬ

3.2. Стадия сбора и анализа требований

В рамках данной стадии осуществляется взаимодействие с функциональным заказчиком, направленное на выявление требований к Системе, потребностей в изменении существующей Системы, разработке дополнительных подсистем, компонентов.

На данной стадии применяются следующие процессы:

- процесс определения цели и назначения Системы;
- процесс сбора требований;
- процесс систематизации и анализа требований;

Роли исполнителей, действующих на данной стадии:

- функциональный заказчик;
- аналитик;
- руководитель проекта;
- ведущий программист.

Результатом данной стадии является техническое задание на проектирование и разработку Системы.

3.3. Стадия проектирования архитектуры и разработки

В рамках стадии проектирования определяются функциональные возможности Системы и ее место в программной архитектуре предприятия, определяется перечень компонентов Системы и необходимых доработок.

На данной стадии применяются следующие технические процессы:

- процесс проектирования архитектуры;
- процесс реализации;
- процесс комплексирования системы;
- процесс тестирования системы;
- процесс менеджмента документации.

Роли исполнителей, действующих на данной стадии:

- руководитель проекта;
- ведущий программист;
- программист;
- аналитик;
- инженер тестирования;
- инженер поддержки процесса и инфраструктуры разработки ПО.

К моменту старта реализации подготавливается среда разработки и определяется методология управления разработкой.

Комплексирование (сборка) приложений Системы осуществляется из исходного кода. Исходный код хранится в системе управления версиями. Для формирования сборки приложений Системы разработаны специализированные автоматизированные процедуры.

В результате работы инструментов сборки на выходе получают:

- готовые к установке приложения/компоненты Системы;
- набор скриптов для модификации модели данных Системы.

На данной стадии осуществляется тестирование Системы в различных контурах:

- контур разработки – модульное тестирование;

- контур тестирования – интеграционное и системное тестирование;
- контур регрессионного тестирования – регрессионное тестирование;
- контур нагрузочного тестирования – нагрузочное тестирование.

Применяемые виды тестирования:

- 1) Модульное тестирование – изолированные испытания отдельных программных модулей Системы. Объектом испытания может служить отдельная функция, метод, процедура, модуль или программный объект. Если модуль взаимодействует с внешними системами, для тестирования применяются специализированные программные модули - «заглушки», имитирующие внешние системы.
- 2) Интеграционное тестирование – испытания, при которых программные модули Системы объединяются логически и тестируются как группа. Целью тестирования является выявление проблем взаимодействия отдельных компонентов Системы или взаимодействия Системы с внешними системами.
- 3) Регрессионное тестирование – испытания ранее протестированной Системы, позволяющие убедиться, что внесенные изменения не повлекли за собой появления дефектов в той части программы, которая не менялась.
- 4) Системное тестирование – испытания, направленные на подтверждение корректности поведения Системы в целом, подразумевающее сквозную проверку взаимодействия всех компонентов Системы и взаимодействия Системы с внешними системами.
- 5) Юзабилити-тестирование (проверка эргономичности) – исследование, выполняемое с целью определения, удобен ли пользовательский интерфейс Системы для его предполагаемого применения. Проверка эргономичности – метод оценки удобства продукта в использовании, основанный на привлечении пользователей в качестве испытателей и суммировании полученных от них выводов.

Решения о переводе Системы на следующую фазу разработки или следующую стадию жизненного цикла принимаются в зависимости от результатов тестирования в соответствующих контурах:

- тестирование в контуре разработки – решение о возможности установки сборки в тестовый контур;
- тестирование в контуре тестирования – решение о возможности включения изменения в сборку для регрессионного тестирования и нагрузочного тестирования;

- тестирование в контурах регрессионного и нагрузочного тестирования – решение о начале квалификационного тестирования Системы (стадия приёмки).

Стадия завершается готовностью релиза Системы к предварительным испытаниям, готовностью программной документации в следующем составе:

- 1) Программа и методика испытаний;
- 2) Руководство администратора;
- 3) Руководство менеджера.

Результатом стадии разработки также могут являться дополнительные требования, протокол выполненных работ и прочие документы, предназначенные для использования на следующих стадиях.

3.4. Стадия передачи

В рамках стадии выполняется подготовка экземпляра системы к опытно- промышленной эксплуатации, подготовка конечных пользователей.

Стадия может включать в себя процессы улучшения программного продукта.

В рамках данной стадии:

- передаётся на утверждение функциональному заказчику Программа и методика испытаний (ПМИ);
- производится установка и настройка Системы;
- проводится приёмка Системы, включающая:

1. Предварительные испытания:

В ходе предварительных испытаний проверяется работоспособность и соответствие техническому заданию, устраняются выявленные неисправности и недостатки. Испытания должны включать системное тестирование и нагрузочное тестирование. После проведения предварительных испытаний Система передается в опытную эксплуатацию.

2. Опытную эксплуатацию:

В ходе опытной эксплуатации выполняется работа с программным продуктом по назначению, собираются статистические данные о характеристиках и результатах функционирования, в том числе проводится юзабилити-тестирование. При необходимости осуществляется доработка программного продукта.

3. Приёмочные испытания:

Испытания проводятся после завершения опытной эксплуатации. По результатам приемочных испытаний принимается решение о выпуске релиза Системы (начале серийной эксплуатации Системы).

На данной стадии применяются следующие технические процессы:

- процесс установки и конфигурации Системы;
- подключение технических средств;
- установка прав доступа.

Роли исполнителей, действующих на данной стадии:

- функциональный заказчик;
- руководитель проекта;
- ведущий программист;
- аналитик;
- инженер тестирования;
- инженер поддержки процесса и инфраструктуры разработки ПО.

Результатом стадии является протокол об окончании опытно-промышленной эксплуатации и готовность к промышленной эксплуатации.

3.5. Стадия эксплуатации и сопровождения

Началом стадии применения Системы служит её установка и передача для применения по назначению в соответствии с моделью процесса поставки по ГОСТ Р ИСО/МЭК 12207:2010.

В рамках настоящего документа процесс поставки программного обеспечения для автоматизации процесса классификации и обработки инцидентов и заявок в ITSM-системе на базе алгоритмов искусственного интеллекта «АИК» не рассматривается.

Сопровождение Системы состоит в обеспечении техническим обслуживанием и сопровождением, и другими видами поддержки функционирования и использования Системы в соответствии с согласованными условиями поставки Системы приобретающей стороне.

В рамках данной стадии:

- осуществляется настройка Системы;
- выполняются консультации пользователей Системы;
- осуществляется контроль работы Системы, и анализ собранных данных об отклонениях и отказах;
- определяются новые требования к Системе и формирование решений, необходимых для реализации данных модификаций;

- производятся модификации Системы;
- по мере необходимости обновляется связанная с изменениями системная и программная документация;
- обновленные компоненты Системы помещаются в среду приобретающей стороны;
- сведения о модификации Системы доводятся до всех затронутых обновлениями сторон.

На данной стадии применяются следующие технические процессы:

- процесс функционирования программных средств.
- процесс сопровождения программных средств.

Роли исполнителей, действующих на данной стадии:

- приобретающая сторона;
- поставщик;
- аналитик;
- ведущий программист;
- программист;
- инженер технической поддержки.

Техническая поддержка пользователей осуществляется через Центр поддержки пользователей. Все обращения в Центр поддержки пользователей регистрируются и классифицируются по типам (Таблица 1).

№	Тип	Формат текста описания заявки
1	Инцидент	Несоответствие работы Системы утвержденным техническим заданиям и эксплуатационной документации, вывод сообщений об ошибках
2	Запрос на изменение	Замечание/предложение по работе Системы (запрос на изменение/расширение функциональности)
3	Консультация	Просьба пользователя о пояснении работы в Системе, уточнение каких-либо непонятных моментов

Устранение ошибок в работе Системы производится путём изменения настроек среды выполнения Системы или модификации программного кода Системы с выпуском новой сборки, которая передается приобретающей стороне. При необходимости корректируется документация на Систему.

На основании анализа обращений пользователей Системы, собранных данных о недостатках и отказах Системы функциональный заказчик может самостоятельно принять

решение о модернизации Системы. Модернизация подразумевает развитие функциональных возможностей Системы, повышение удобства использования и администрирования. После модернизации производится выпуск нового релиза Системы. Для выполнения модернизации Системы инициируется переход на стадию проектирования архитектуры и разработки.

В случае заинтересованности в расширении функциональности Системы приобретающая сторона направляет поставщику новые требования к Системе. Требования анализируются поставщиком, далее согласовываются условия оказания услуг по доработке Системы, сроки и стоимость работ. Для выполнения доработки Системы инициируется переход на стадию проектирования архитектуры и разработки.

В случае принятия приобретающей стороной решения о завершении эксплуатации Системы инициируется переход на стадию изъятия и утилизации.

3.6. Стадия изъятия и утилизации

В соответствии с ГОСТ Р ИСО/МЭК 12207-2010 модель процесса прекращения применения программных средств предусматривает прекращение деятельности организации (обладателя исключительного права) по поддержке функционирования и сопровождения или деактивирует, демонтирует и удаляет поврежденные программные продукты, отправляя их в финальное состояние и возвращая окружающую среду в приемлемые условия.

Прекращение применения Системы как серийного программного продукта означает деактивацию и удаление Системы из среды конкретного покупателя (приобретающей стороны).

Деятельность организации (подразделения) по поддержке функционирования и сопровождению может быть прекращена по решению обладателя исключительного права на программное обеспечение «AISee».

Причиной перевода Системы в данную стадию может служить замещение новой системой, катастрофический отказ, неэффективность дальнейшего применения.

На данной стадии применяются следующие технические процессы:

- процесс прекращения применения программных средств.

Роли исполнителей, действующих на данной стадии:

- приобретающая сторона;
- поставщик;
- функциональный заказчик;
- аналитик;

– инженер технической поддержки.

Все заинтересованные стороны оповещаются о планах и действиях по выводу Системы из эксплуатации.

Вся связанная документация по разработке, журналы и коды помещаются в архивы. Используемые данные или данные, связанные с прекращением применения Системы, должны быть доступны в соответствии с требованиями законодательства, локальных нормативных актов, соглашений и условий поставки.