Управление распределенным складом

Архитектор 1С

Меня хорошо видно & слышно?

Защита проекта
Тема: Управление распределенным складом

Гыдилика Александр

Ведущий программист-разработчик
1С

17 лет работы в сфере 1С

План защиты

- Цель и задачи проекта
- Какие технологии использовались
- Что получилось
- Выводы
- Вопросы и рекомендации

Цель и задачи проекта

Цель проекта: проект предусматривает ведение складского учета в распределенных узлах и управляемого из центрального офиса

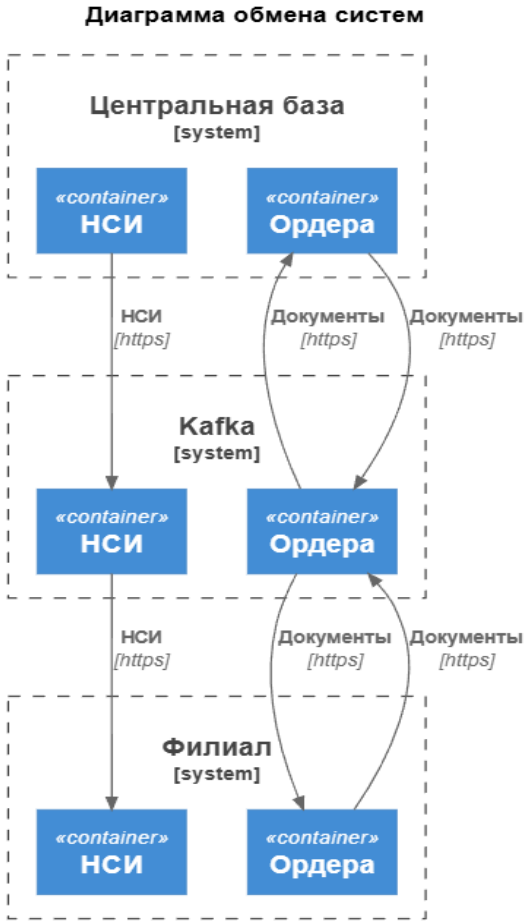
1.	Реализация складского учета товара (приемка, перемещение, списание)
2.	Фактическое выполнение складских операций на распределенных узлах
3.	Возможность оперативного увеличения количества узлов (складов)
4.	Отражение результатов работы со складскими документами в центральной базе
5.	Применение механизмов архитектурного проектирования, разработки, тестирования

Какие технологии использовались

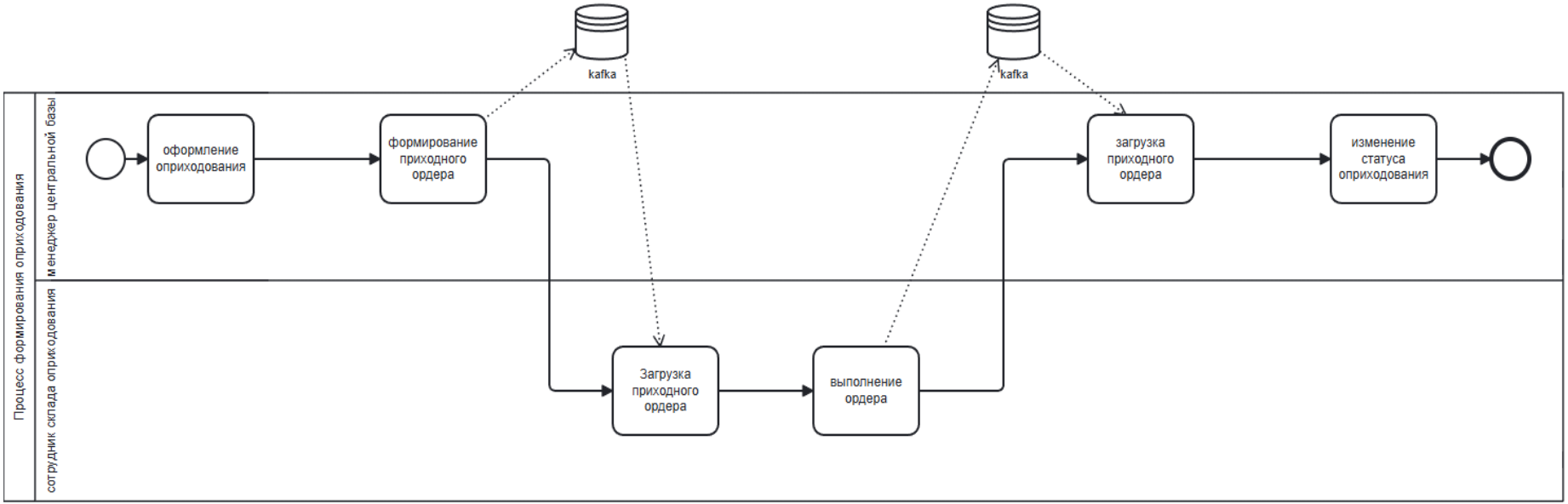
1.	1С предприятие 8, 1С: EDT
2.	Github + gitExtension
3.	Docker+Kafka
4.	VS Code + плагины
5.	Vanessa Automation + YAxUnit
6.	прочее

Что получилось

При помощь инструмента **VS Code + plant UML** разработана диаграмма обмена между системами

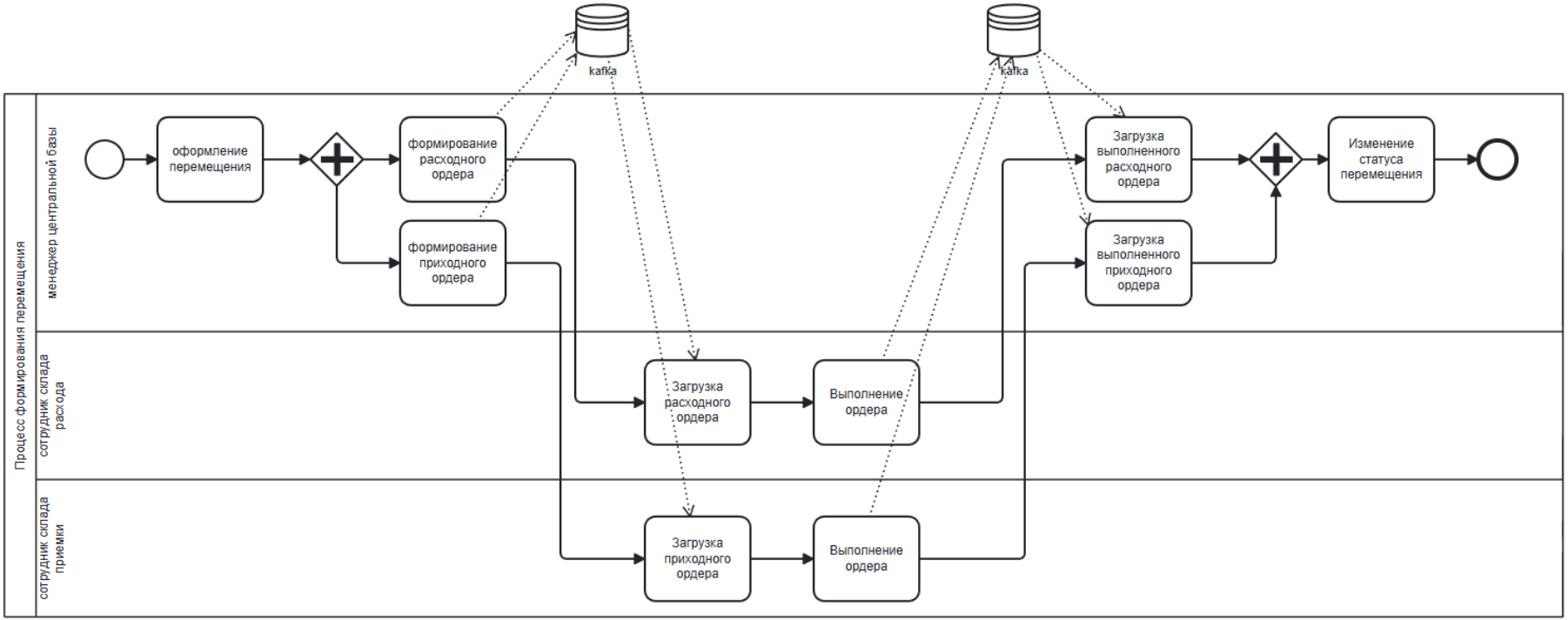


Процесс обмена документа "Оприходование"

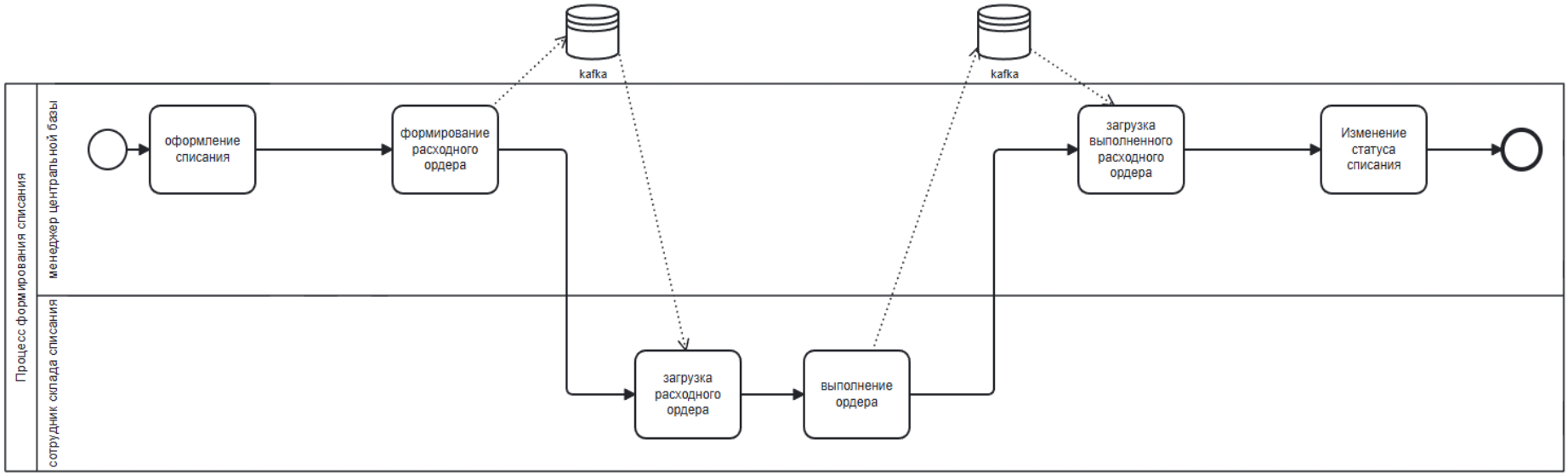


Применение системы BPMN позволяет реализовать схему процесса

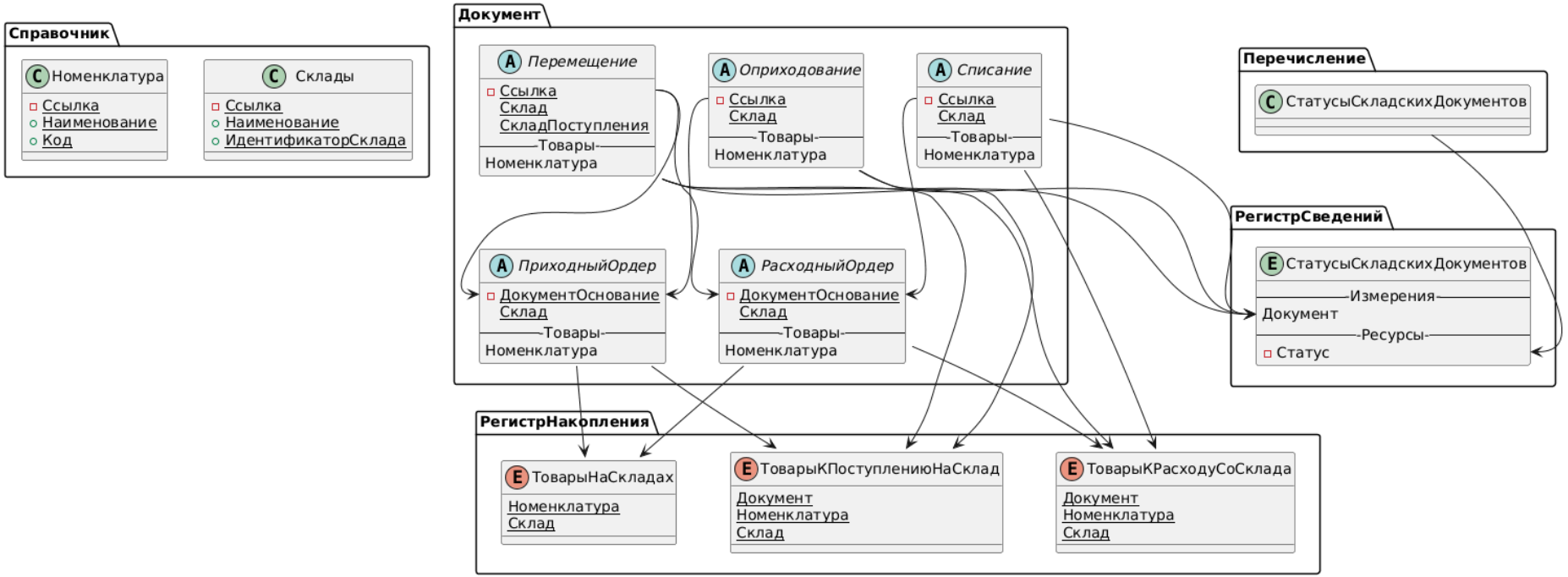
Процесс обмена документа "Перемещение"



Процесс обмена документа "Списание"



ER-диаграмма

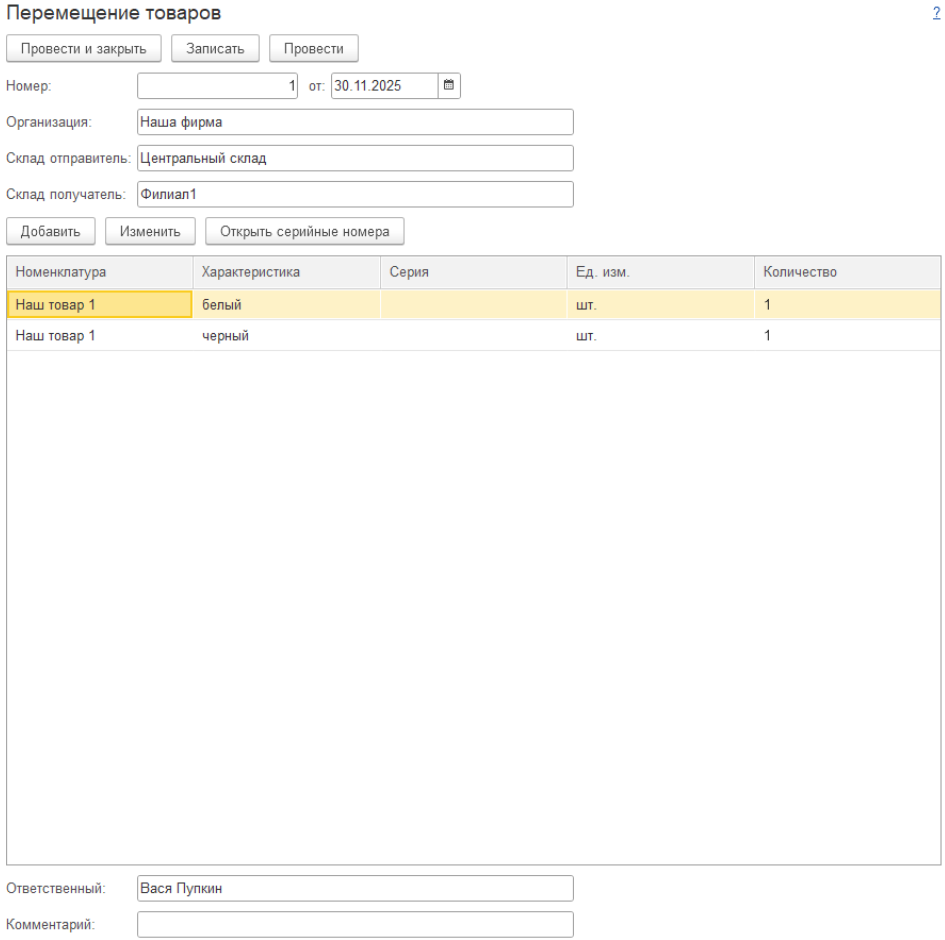


ER-схема реализована с помощью внешней обработки построения схем

Общий вид форм

Внешний вид форм максимально упрощен ввиду специфики операций и невысоким навыкам людей, которые будут с ними работать

Внешняя обработка "1С Накидка" с помощью ИИ-технологий позволяет нарисовать 1С форму не заходя в конфигуратор



Сценарное тестирование

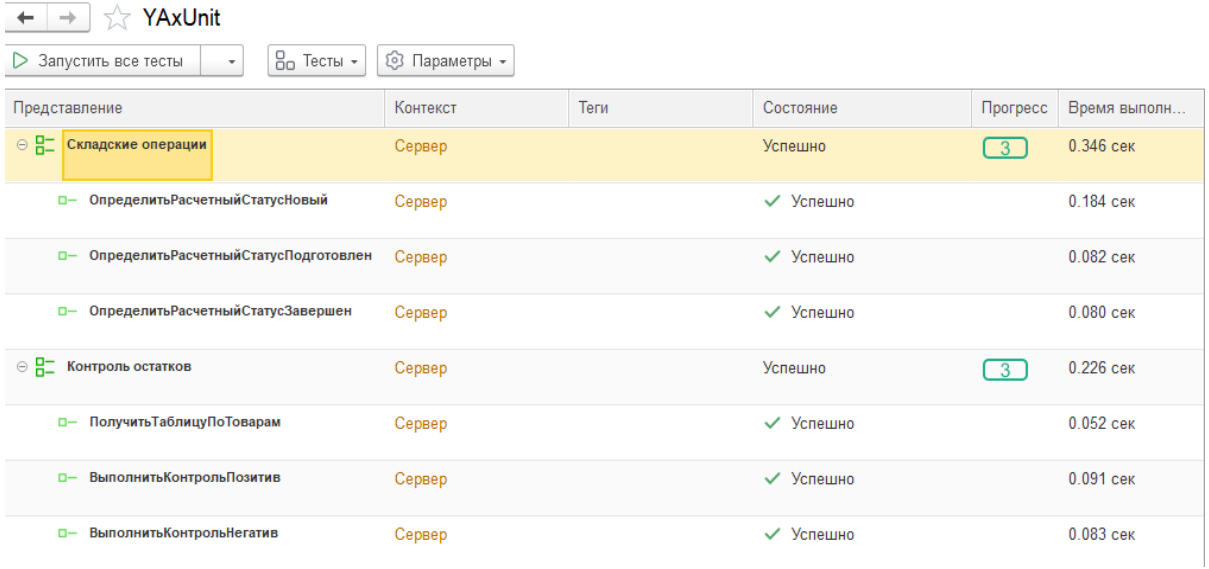
Инструмент Vanessa-automation дает возможность протестировать пользовательские сценарии взаимодействия в разных базах

Добро пожаловать! | 1_0_СозданиеИОбменДокумента

```
#language: ru

@Синхронизация

Функционал: Обмен документа

Контекст:
    Дано я подключаю TestClient "Этот клиент" логин "Admin" пароль ""
    И я закрываю все окна клиентского приложения

Сценарий: Создание документа и его выгрузка
    *Создаем документ
        И В командном интерфейсе я выбираю "Склад" "Оприходование"
        Тогда открылось окно "Оприходование"
        И я нажимаю на кнопку с именем 'ФормаСоздать'
        Тогда открылось окно "Оприходование (создание)"
        И из выпадающего списка с именем 'Склад' я выбираю точное значение "Филиал 1"
        И в таблице 'Товары' я нажимаю на кнопку с именем 'ТоварыДобавить'
        И в таблице 'Товары' из выпадающего списка с именем 'ТоварыНоменклатура' я выбираю точное значение "Элемент 105"
        И я перехожу к следующему реквизиту
        И в таблице 'Товары' в поле с именем 'ТоварыКоличество' я ввожу текст "7,000"
        И в таблице 'Товары' я завершаю редактирование строки
        И я нажимаю на кнопку с именем 'ФормаПровести'
        Тогда открылось окно "Оприходование * от *"
        И я нажимаю на кнопку с именем 'ФормаДокументПриходныйОрдерСоздатьНаОсновании'
        Тогда открылось окно "Приходный ордер (создание)"
        И я нажимаю на кнопку с именем 'ФормаЗаписать'
        И я запоминаю значение поля "Номер" как "НомерДокумента"          {"НомерДокумента": "000000033"}
        И я нажимаю на кнопку с именем 'ФормаПровестиИЗакрыть'
    *Выгружаем документ
        И В командном интерфейсе я выбираю "Синхронизация" "Синхронизация кафка"
        Тогда открылось окно "Синхронизация кафка"
        И в таблице 'Список' я перехожу к строке:
            | "Адрес сервера" | "Код"       | "Назначение" | "Наименование" | "Склад"             | "Топик" | "Центральный офис" |
            | "kafka:9092"    | "000000002" | "Документы"  | "Документы"    | "Центральный склад" | "docs"  | "Да"               |
        И в таблице 'Список' я выбираю текущую строку
        Тогда открылось окно "Документы (Синхронизация кафка)"
        И я нажимаю на кнопку с именем 'ВыгрузитьДанные'
    *Переходим в подчиненную базу
        И я подключаю TestClient "Филиал" логин "Admin" пароль ""
```

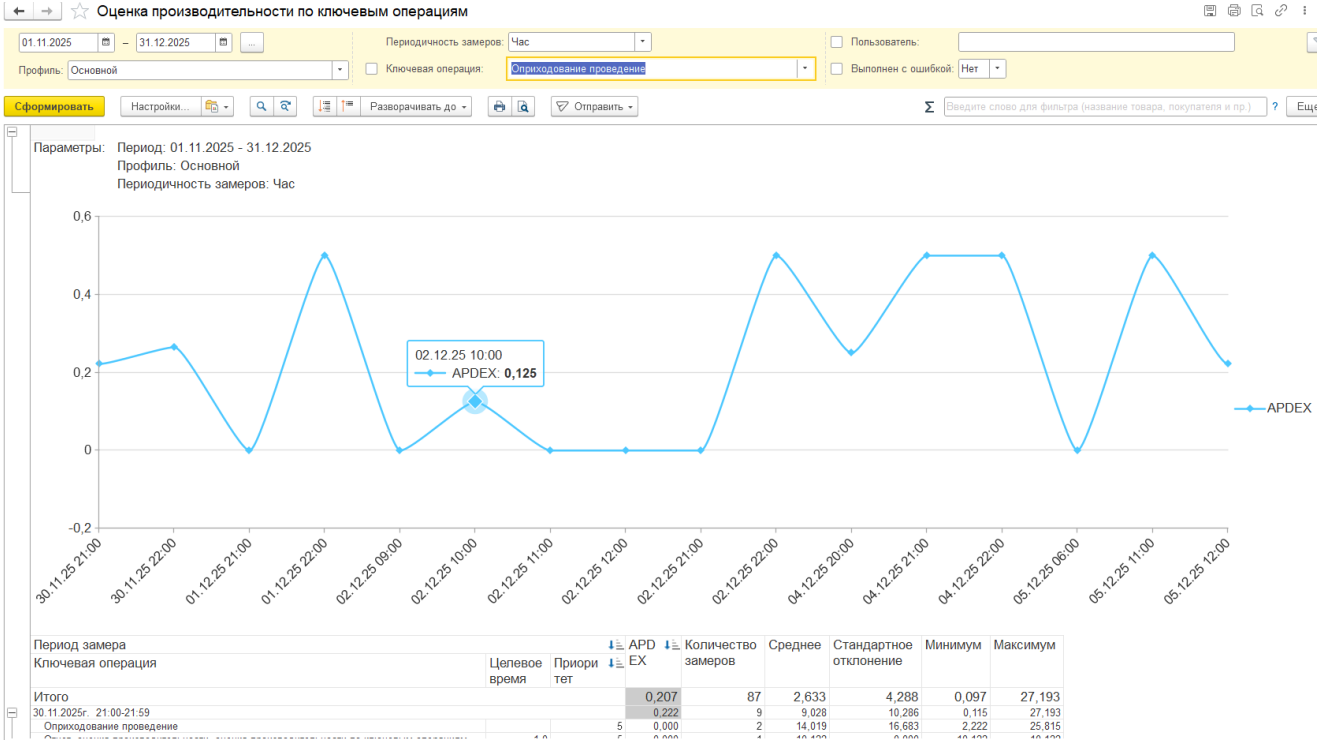
Unit-тесты

Благодаря Unit тестам всегда можем выявить ошибки как при разработке нового, так и при изменении текущего функционала. Своевременная разработка таких тестов способствует написанию сложных алгоритмов путем разбиения на множество более простых



APDEX

Встроенные модули БСП позволяют анализировать производительность механизмов и своевременно принимать меры по оптимизации функционала



Период замера / Ключевая операция	Целевое время	Приоритет	APDEX	Количество замеров	Среднее	Стандартное отклонение	Минимум	Максимум
Итого			0,207	87	2,633	4,288	0,097	27,193
30.11.2025г. 21:00-21:59			0,222	9	9,028	10,286	0,115	27,193
Оприходование проведение		5	0,000	2	14,019	16,683	2,222	25,815

Выводы

1.	Комплексное использование современных механизмов проектирования и разработки упорядочивает процесс и позволяет избегать ошибок построения бизнес-логики, программирования, конфликтов взаимодействия
2.	Применение технологии Git позволяет избегать потерь доработок и других полезных файлов. Применение шаблона упорядочивает хранение данных проекта
3.	Автоматизированное тестирование ускоряет процесс разработки на дистанции
4.	Построение схем и процессов до начала разработки упрощает сам процесс

Вопросы и рекомендации

\+ если есть вопросы

— если вопросов нет

Спасибо за внимание!