

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**ИНСТРУМЕНТ ПО АКТУАЛИЗАЦИИ И АНАЛИЗУ  
ИНТЕГРИРОВАННОЙ МОДЕЛИ НЕФТЕГАЗОВОГО АКТИВА**

**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

Студентки 4 курса 411 группы  
направления 02.03.02 — Фундаментальная информатика и информационные  
технологии  
факультета КНиИТ  
Москаленко Елены Евгеньевны

Научный руководитель  
доцент, к.ф.-м.н.

\_\_\_\_\_

А. С. Иванова

Заведующий кафедрой  
доцент, к.ф.-м.н.

\_\_\_\_\_

С. В. Миронов

Саратов 2026

## ВВЕДЕНИЕ

**Актуальность темы.** Современные нефтегазовые компании активно переходят к цифровизации производственных процессов, включая моделирование и анализ работы месторождений. Интегрированные модели нефтегазовых активов, объединяющие в себе данные по фонду скважин, кустовым площадкам, сети сбора, транспортировки и системам поддержания пластового давления (ППД), становятся ключевым инструментом для принятия решений. Однако эффективность таких моделей напрямую зависит от качества их актуализации — своевременного обновления с учетом реальных производственных данных.

На практике поддержание интегрированных моделей в актуальном состоянии представляет собой трудоемкую и частично ручную задачу. Исходные данные поступают из разнородных источников: фактических замеров унифицированной системой обмена информацией (УСОИ), ШТР, промысловых информационных систем, отчетов по работе скважин и кустовых площадок. Для корректного использования данные требуют унификации, расчетов поправочных коэффициентов, фильтрации аномальных значений и последующей автоматизированной загрузки в расчетную модель.

Особую сложность представляет интеграция с программным обеспечением класса PetEx, в частности с модулем GAP (General Allocation Program). Запись параметров в модель реализована через COM-интерфейс OpenServer API и требует формирования иерархических тегов объектов. На практике большинство операций выполняется вручную или с помощью несистематизированных скриптов, что делает процесс невоспроизводимым и подверженным ошибкам.

Актуальность данной работы обусловлена отсутствием готовых инструментов, объединяющих полный цикл актуализации — от предобработки данных до записи в GAP-модель и формирования отчетности — в единый воспроизводимый процесс с конфигурированием через файл без правки кода.

**Цель бакалаврской работы** — спроектировать и программно реализовать инструмент по актуализации и анализу интегрированной модели нефтегазового актива, обеспечивающий автоматизацию полного рабочего цикла с конфигурированием через YAML-файлы.

Поставленная цель определила **следующие задачи**:

- провести анализ существующих решений, разработок и подходов к поддержанию интегрированных моделей.

- изучить методологические основы расчетов, применяемых в моделях добычи и закачки.
- спроектировать модульную архитектуру инструмента с двумя независимыми функциональными блоками.
- реализовать функционал актуализации с поддержкой расчета расходов/приемистостей, фильтрации данных, расчета ПК различными алгоритмами и переноса данных в модель.
- реализовать модуль формирования результатов с генерацией многолистовых Excel-файлов, включая встроенные графики.
- вынести все пользовательские параметры в YAML-конфигурацию.
- провести тестирование созданного инструмента на реальных производственных данных и оценить соответствие его функционала требованиям.

**Методологические основы** разработки инструментов автоматизации для нефтегазовой отрасли представлены в работах И. В. Шпурова, М. Ю. Данько, К. А. Веласкеса, Д. М. Мамонова, В. В. Рокотянского, З. Ф. Исмагиловой, Д. М. Еремеева и других авторов.

**Теоретическая значимость бакалаврской работы** заключается в анализе существующих решений для актуализации интегрированных моделей, исследовании механизмов взаимодействия с GAPI через OpenServer API и обосновании выбора алгоритмов расчета паркового коэффициента.

**Практическая значимость бакалаврской работы** заключается в разработке готового к применению инструмента, сокращающего время полного цикла актуализации с 10-12 часов до 45-60 минут и исключающего ошибки ручного ввода данных.

**Структура и объем работы.** Бакалаврская работа состоит из введения, 4 разделов, заключения, списка использованных источников и 4 приложений. Общий объем работы — 58 страниц, из них 41 страница — основное содержание, включая 7 рисунков, 7 таблиц, цифровой носитель в качестве приложения, список использованных источников информации — 20 наименований.

## КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

**Первый раздел «Аналитико-методологические основы разработки инструмента по актуализации и анализу интегрированной модели нефтегазового актива»** включает описание предметной области, анализ существующих решений, методологические основы расчетов, обоснование технологического стека и требования к разрабатываемому инструменту. В подразделе 1.1 рассматривается архитектура интегрированных моделей нефтегазовых активов и механизм взаимодействия с GАР через OpenServer API. В подразделе 1.2 проводится сравнительный анализ существующих решений. В подразделе 1.3 описаны методологические основы расчетов: парковый коэффициент, газовый фактор, суммарная жидкость. В подразделах 1.4 и 1.5 представлены технологический стек и требования к инструменту.

**Анализ предметной области.** Интегрированная модель актива (ИМА) — цифровой двойник нефтегазового месторождения, объединяющий модели пласта, скважин, промысловой сети и технологического оборудования. В рамках данной работы рассматривается подмножество ИМА — модель сети сбора и транспорта, построенная в программном комплексе PetEx с использованием модуля GАР. Модель описывает гидравлически связанную систему трубопроводов, насосов, сепараторов и скважин. Модель может функционировать как в режиме добычи нефти, газа и воды (extraction), так и в режиме закачки воды для поддержания пластового давления (injection).

Взаимодействие с GАР-моделью осуществляется через OpenServer API — СОМ-интерфейс, предоставляющий методы GetValue и SetValue для чтения и записи атрибутов объектов по иерархическим тегам. Актуализация GАР-модели подразумевает периодическое обновление объемных расходов по фазам, давлений на кустах и сепараторах, а также коэффициентов настройки.

**Анализ существующих решений.** Выявлены три основные категории решений: встроенные средства PetEx, корпоративные Python/VBA-скрипты и платформы промышленной аналитики. Встроенные средства PetEx не рассчитаны на сложную предобработку данных из гетерогенных Excel-источников и не формируют структурированную отчетность. Корпоративные скрипты имеют жестко заданные параметры, не поддерживают конфигурирование и не документированы. Платформы аналитики обеспечивают визуализацию, однако не решают задачу непосредственной записи параметров в GАР-модель. Анализ показыва-

ет высокую актуальность разработки инструмента, объединяющего весь процесс в единый воспроизводимый рабочий цикл.

**Методологические основы расчетов.** Парковый коэффициент (ПК) — основной параметр актуализации, представляющий собой отношение фактического расхода по УСОИ к плановому по ШТР. В инструменте реализованы три алгоритма расчета ПК: без поправки (ПК = 1.0), индивидуальный (отдельно для каждой даты) и трендовый с Z-score фильтрацией аномалий и скользящим средним с окном 5 точек. Газовый фактор рассчитывается как отношение объемного расхода газа к расходу нефти с переводным коэффициентом 1000.

**Технологический стек.** Основным языком реализации выбран Python 3.11 как стандарт для инженерной аналитики в нефтегазовой отрасли. Для обработки табличных данных используется pandas, для численных расчетов и Z-score фильтрации — numpy. Формирование xlsx-файлов с применением стилей, формул и условного форматирования осуществляется через orepnuxl; создание графиков — через xlwings посредством COM API, поскольку orepnuxl не поддерживает ChartObjects. Доступ к OpenServer API PetEx обеспечивается библиотекой ruwin32. Конфигурирование реализовано через YAML как наиболее читаемую альтернативу JSON и INI для сложных вложенных структур.

**Второй раздел «Проектирование инструмента»** посвящен архитектурным решениям: модульной структуре, организации конфигурационного файла, примененным паттернам проектирования и схеме потока данных.

**Модульная архитектура.** Инструмент построен по двухуровневой модульной архитектуре. На верхнем уровне расположены два независимых функциональных блока: модуль актуализации (ActualizationProc) и модуль формирования результатов, представляющий собой набор специализированных процессоров листов Excel. Управление запуском осуществляется из единой точки входа main.py через флаги run\_actualization и run\_reporting в конфигурационном файле. На нижнем уровне находятся вспомогательные компоненты: базовый класс DataProc для работы с файловой системой, класс Style для единообразного оформления Excel-файлов и класс Server — обертка над OpenServer API. Взаимодействие между блоками осуществляется через общий файл config.yaml и общий результирующий файл result.xlsx.

**Структура конфигурационного файла.** Все пользовательские параметры вынесены в единый файл config.yaml, организованный по логическим секци-

ям: пути к входным и выходным данным, флаги запуска модулей, режим работы и тип объектов GAP, настройки записи в GAP (префиксы объектов, имена переменных расходов и давлений), алгоритм расчета ПК и порог отсева аномалий, список исключаемых кустов, шаблоны имен входных файлов, диапазон дат, параметры отдельных листов отчетности, а также маппинги — описания преобразований входных файлов в листы result.xlsx с переименованием столбцов.

**Паттерны проектирования.** В архитектуре инструмента применяются четыре паттерна: Strategy — алгоритм расчета ПК выбирается через параметр `pk_method` и инкапсулирован в трех независимых ветвях; Template Method — базовый класс `DataProc` определяет общий интерфейс работы с файлами, который специализированные процессоры наследуют и переопределяют только в части логики обработки данных; Facade — точка входа `main.py` предоставляет единый интерфейс запуска полного конвейера; Registry — словари `_inflow_exists_cache` и `_source_exists_cache` хранят результаты однократной проверки наличия объектов в GAP, действуя как кэш с ленивой инициализацией.

**Схема потока данных.** Поток данных организован в два независимых конвейера. Конвейер актуализации загружает входные Excel-файлы через `DataProc`, вычисляет ПК и итоговые расходы, после чего записывает их в объекты INFLOW/SOURCE GAP-модели через OpenServer API, затем записывает давления кустов и сепаратора. Конвейер отчетности загружает те же входные файлы повторно, дополнительно считывает расчетные значения из GAP-модели и формирует листы result.xlsx, после чего добавляет графики через `xlwings`.

**Третий раздел «Реализация инструмента»** описывает программную реализацию всех компонентов системы.

**Базовая инфраструктура.** Класс `DataProc` инкапсулирует операции с файловой системой и чтение Excel-файлов. Метод `getFilePathByPattern` использует `glob` для поиска файла по шаблону, что позволяет указывать маски вида «\*Кусты\*.xlsx» вместо точного имени файла с датой. Метод `getDataFrame` читает указанный лист Excel в `pandas DataFrame` с обработкой исключений: при ошибке возвращает `None` и выводит сообщение, не прерывая выполнение. Класс `Style` реализует единую систему визуального оформления через `openpyxl`: метод `stylizeCell` применяет к ячейке числовой формат, цвет фона, жирность шрифта и границы; метод `createSectionHeader` создает объединенный заголовок секции; метод `autoWidth` автоматически подбирает ширину столбцов с учетом типа

данных.

**Модуль актуализации (ActualizationProc).** Метод `_load_source_frames` загружает три входных датафрейма: дебиты, давления на кустах и общие данные ДНС, приводя даты к типу `datetime.date`. Метод `_filter_excluded_pads` удаляет из расчетов кусты из списка `exclude_pads`. Метод `_compute_pk_from_general` рассчитывает ПК по нефти, воде и газу: деление на ноль предотвращается заменой нулевых значений ШТР на NaN. Выбор алгоритма реализован в методе `_apply_pk_method` по паттерну Strategy. Трендовый алгоритм реализован в методе `_apply_trend_smoothing`: для каждой фазы вычисляется Z-score, значения с отклонением выше порога заменяются на NaN, после чего применяется скользящее среднее с окном пять точек; оставшиеся пропуски восстанавливаются исходными значениями.

Перед записью в GAP-модель метод `_build_object_registry` выполняет однократную проверку наличия объектов INFLOW и SOURCE для каждого куста и кэширует результат. Логика построена с приоритетом INFLOW над SOURCE. Проверка выполняется через считывание атрибута LABEL объекта; результат сохраняется в `_inflow_exists_cache`. Для объектов типа INFLOW данные записываются двумя способами: статический Lookup на дату среза (таблица IPR с двумя строками и одинаковыми значениями расходов) и расписание Schedule для всего диапазона дат конфига. Давление на кустах записывается через OSVAR\_CHANGE с переменной MANPRES; давление на сепараторе — как в SolverPres, так и в Schedule через тип мероприятия SEP\_PRESCHNG.

**Процессоры листов Excel.** DNSProc формирует лист DNS со сводной таблицей показателей добычи актива. Структура листа имеет двухуровневые заголовки: первый уровень объединяет группы (Oil Rate, Water Rate, Gas Rate, GOR, Liquid Rate), второй содержит конкретные параметры (ШТР, УСОИ, ПК, Настройка). Метод `processPKValues` вставляет в ячейки Excel-формулы деления с зеленым фоном при наличии числовых значений или формулу `=NA()` в противном случае; в строку заголовка вставляется формула AVERAGE. Настроечный расход вычисляется как  $\text{ШТР} \times \text{ПК}_{\text{среднее}}$ . Метод `addCharts` создает через `xlwings` пять линейных графиков; пустые ячейки не отображаются благодаря настройке `DisplayBlanksAs=1`.

PlinProc формирует лист с фактическими линейными давлениями по кустовым площадкам, применяя `pandas.pivot_table` для преобразования строчной

структуры в матрицу «дата × куст».

GarObjProc считывает расчетные давления из GAP-модели через OpenServer API; список объектов определяется методом searchObjects, перебирающим индексы до 2000 с прерыванием после 50 пустых ячеек подряд.

QProc — универсальный процессор для формирования листов расходов по фазам (нефть, вода, газ); универсальность достигается параметризацией конструктора именем целевой переменной и листа.

SREZProc формирует лист сравнения фактических и расчетных давлений на дату среза с условным форматированием по градиенту зеленый-желтый-красный.

UZAProc обрабатывает данные узловых измерительных аппаратов, извлекая номер УЗА из имени файла регулярным выражением.

SKProc формирует набор индивидуальных листов по каждой кустовой площадке с вычисляемыми показателями суммарной жидкости и отклонения давлений, добавляя на каждый лист два графика.

Точка входа main.py является фасадом всего инструмента: загружает конфигурацию из config.yaml и последовательно вызывает процессоры в порядке, обеспечивающем корректное формирование листов result.xlsx. Режим mode (extraction/injection) определяет набор запускаемых QProc-процессоров.

**Четвертый раздел «Тестирование и результаты»** содержит описание тестовых данных, результаты тестирования модулей актуализации и формирования отчетности, проверку граничных случаев и оценку производительности.

**Описание тестовых данных.** Тестирование проводилось на реальных производственных данных Актива X в среде Windows 11 с установленным PetEx 11. Временной диапазон — 365 дат (01.01.2025-31.12.2025), дата среза — 09.07.2025, количество кустовых площадок — 20, тип объектов в GAP — INFLOW, режим работы — extraction.

**Тестирование модуля актуализации.** Для алгоритма 'individual' значения ПК совпали с ручными расчетами специалистов до четвертого знака после запятой по всем 365 датам. Алгоритм 'trend' корректно выявил и отфильтровал 7 аномальных точек по нефти и 4 по воде ( $|z| > 3.0$ ), вызванных сбоями измерений УСОИ. Алгоритм 'none' выдал расходы, совпадающие с ШТР, что соответствует ожидаемому поведению. Все 20 объектов INFLOW найдены в модели; кэш заполнен за одну итерацию проверки. Выполнено 51 465 операций do\_set (43

800 — расходы в Schedule, 7 300 — давления кустов, 365 — давление сепаратора). Статический Lookup на дату среза создан корректно для всех 20 объектов. Проверка считыванием через GetValue после записи не выявила расхождений. Для подтверждения корректности учета внесенных данных выполнен прогнозный расчет модели по двум контрольным объектам — результаты соответствуют ожидаемым.

**Тестирование модуля формирования результатов.** При полном запуске result.xlsx содержит 28 листов: DNS (5 графиков), Рлин, Рлин (расчет), QнТех, QвТех, Qг, Срез (1 график), UZA (4 графика) и 20 листов S\_K по числу кустов (2 графика каждый). Все стили ячеек применялись корректно; автоподбор ширины столбцов работал верно для всех типов данных; условное форматирование листа «Срез» отображало градиент пропорционально величине относительного отклонения.

**Проверка граничных случаев.** Протестированы семь граничных ситуаций: отсутствие данных за отдельные даты (строки с NaN пропускаются при записи в Schedule), нулевые значения ШТР (деление предотвращено через replace(0:pr.nan)), недоступность OpenServer (исключение перехватывается и логируется без прерывания выполнения), отсутствие объекта INFLOW в GAP (кэш фиксирует False, куст пропускается без ошибки), повторный запуск (лист удаляется и создается заново), пустой список exclude\_pads (фильтрация не применяется), все значения ПК являются аномалиями при методе 'trend' (fillna восстанавливает исходные значения). Все случаи обработаны корректно.

**Производительность.** Доминирующим по времени этапом является взаимодействие с OpenServer API — около 90% общего времени. Полный цикл актуализации и формирования отчетности на 365 датах и 20 кустах занимает 42-56 минут. Кэширование проверок наличия объектов сократило число обращений к API для метаданных приблизительно в пять раз. При запуске только модуля отчетности (без актуализации) время составляет 4-6 минут.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был спроектирован и программно реализован инструмент по актуализации и анализу интегрированной модели нефтегазового актива. Все поставленные задачи решены в полном объеме.

В аналитической части проведен комплексный обзор предметной области: рассмотрены архитектура GАР-модели PetEx, механизм взаимодействия через OpenServer API и методология расчета паркового коэффициента. Выявлены недостатки существующих решений: отсутствие единого конфигурируемого цикла актуализации, невозможность расчета ПК несколькими методами, отсутствие автоматизированной отчетности.

При проектировании разработана двухуровневая модульная архитектура с применением паттернов Strategy, Template Method, Facade и Registry. Все пользовательские параметры вынесены в YAML-конфигурацию, что исключает необходимость правки кода при изменении входных данных или структуры модели.

В части реализации созданы следующие компоненты: модуль ActualizationProc, обеспечивающий загрузку и предобработку данных, три алгоритма расчета ПК (none / individual / trend с Z-score фильтрацией), формирование итоговых расходов и запись в GАР через OpenServer API с поддержкой INFLOW Lookup и Schedule, SOURCE, давлений кустов и сепаратора; семь процессоров листов Excel (DNSProc, PlinProc, GapObjProc, QProc, SREZProc, UZAProc, SKProc), формирующих 28 листов единого файла result.xlsx с единым образом стилем и встроенными графиками; вспомогательные классы DataProc и Style, обеспечивающие работу с файловой системой и единое оформление.

Тестирование на реальных данных Актива X подтвердило корректность расчетов ПК, успешную запись 51 465 операций в GАР-модель и правильность формирования отчетности. Семь проверенных граничных случаев обрабатываются корректно.

Практический эффект внедрения: сокращение времени полного цикла актуализации с 10-12 часов до 45-60 минут, исключение ошибок ручного ввода, унификация отчетности.

Перспективы развития инструмента включают добавление поддержки DUOTech IMA, реализацию веб-дашборда на Plotly/Dash для интерактивной

визуализации результатов, а также параллельную запись в OpenServer API для сокращения времени цикла актуализации.