**1 слайд ВСТУПЛЕНИЕ**

Добрый день, меня зовут Карпенко Дмитрий. Я ведущий специалист Управление автоматизации процессов блока "Материальные потоки и балансы" компании Сибинтек-СОФТ.

Если сказать общими словами, то отдел в котором я работаю занимается сопровождением и внедрением производственных и бизнес систем. Об одной из них мы сегодня Вам и расскажем.

**2 слайд СИБИНТЕК СЕГОДНЯ**

НПЗ и нефтедобыча в нашей стране - это очень крупный сектор экономики.

Заводы Роснефти расположены во многих крупных городах: Туапсе, Рязань, Комсомольск, Куйбышевск, Сызрань, Ачинск, Ангарск, Рязань, Саратов и многие другие…

И для каждого из этих заводов спутником прикреплён офис компании Сибинтек.

Мы занимаемся информационными технологиями на данных предприятиях. В 1999 году было принято решение, что сотрудники заводов должны заниматься технологическими процессами, а IT – сегмент поручить отдельной компании, которая централизованно будет оказывать сопровождение для всех заводов компании.

Таким образом мы уже 25 лет на рынке и постоянно растём.

Но для начала нужно понимать, что СИБИНТЕК и СИБИНТЕК – СОФТ – это дочерняя компания Роснефти, и для полного понимания системы, о которой сегодня пойдёт речь нужно немного углубиться в прикладную область, а именно в нефтепереработку.

**3 слайд Автоматизация бизнес-процессов нефтеперерабатывающего завода**

Сама прикладная часть для нашей производственной системы является нефтепереработка.

Сразу скажу, что я составлю для Вас общую картину не вдаваясь в подробности.

Если вы уже интересуетесь нефтепереработкой, либо химическими технологическими процессами производства, то тут вы сможете реализовать полностью свой потенциал совмещая с IT технологиями.

Но если вы далеки от этого, то в ходе работы вы сами всё узнаете т.к. мы много взаимодействуем с заводчанами, а они иногда очень любят рассказывать о своей работе).

**Объяснение схемы 1 вариант:**

Представьте завод как огромный конвейер, где каждая установка — это отдельный цех со своей специализацией. Вот как это работает:

**Подготовка нефти (ЭЛОУ-АВТ-6)**

"Мойка нефти": На установке **ЭЛОУ** (электрообессоливающая установка) нефть очищают от солей и воды — как промывают овощи перед готовкой.

"Разделение на фракции": Затем на **АВТ** (атмосферно-вакуумная трубчатка) нефть нагревают до 350°C и разделяют в 65-метровой колонне:

Сверху улетают легкие фракции (бензиновые)

В середине — дизельные

Внизу остается мазут

**Производство топлива (Каталитические установки)**

**Л-24/6** — "Бензиновая кухня":  
Перерабатывает тяжелые фракции в высокооктановый бензин с помощью катализатора

**Л-35/100-300** и **Л-35/100-600** — "Дизельные цеха":  
На этих установках получают зимнее и летнее дизтопливо. Цифры 300/600 — это производительность в тыс. тонн в год

**Дополнительная переработка**

Мазут с АВТ отправляют на **битумные установки** (для дорожного покрытия)

Газы перерабатывают на **ГФУ** (газофракционирующая установка)

**Как это контролируется?**  
Каждая установка оснащена:

500+ датчиков на **Л-35/100-600**

Автоматикой, передающей данные в **АСОДУ**

Системой аварийной остановки (срабатывает за 2 секунды)

**4 слайд Где мы в Пирамиде CIM**

*Каждая установка НПЗ генерирует огромный объем данных — от показаний датчиков до управляющих сигналов. Но как эти данные собираются и превращаются в полезную информацию? Вот где вступает* ***иерархия автоматизации****, известная как пирамида CIM. Давайте разберём её уровни...*

**0 уровень – полевая автоматика**

На этом уровне находятся датчики, сенсоры, клапаны, приводы и другие исполнительные устройства. Они фиксируют и передают данные на следующие уровни.

**1 уровень – управление оборудованием в PLC/PAC**

PLC (программируемые логические контроллеры) и PAC (программируемые автоматизированные контроллеры) управляют логикой процессов локально. Например, если температура опустилась ниже заданного уровня, контроллер включает нагреватель.

**2 уровень — диспетчерское управление и сбор данных в SCADA**

SCADA помогает контролировать и мониторить показатели в реальном времени. Система собирает данные с PLC/PAC и передает их на устройство, где операторы отслеживают статус оборудования. MES, в свою очередь, использует данные из SCADA для дальнейшего анализа, планирования, координации и управления производством.

С развитием интернета вещей, MES может получать информацию напрямую от датчиков. Это возможно с помощью IoT-платформ.

**3 уровень — управление процессами в MES**

MES получает данные из SCADA или напрямую от датчиков и передает их в ERP-систему. Существуют также гибриды MES и ERP для управления производственными процессами, запасами, закупками и планированием.

**4 уровень — управление бизнес-процессами в ERP**

[ERP (enterprise resource planning)](https://www.purrweb.com/ru/uslugi/razrabotka-erp-sistem/) – это софт для контроля бизнес-процессов компании. Система объединяет управление финансами, закупками, продажами, заказами, отчетностью и персоналом. ERP-системы не управляют производством напрямую. Они передают данные в MES, чтобы корректировать планы на выпуск продукции.

**5 слайд Интегрированная система управления предприятием**

**Что это?**  
АСОДУ — это централизованная система для мониторинга и управления технологическими процессами на производстве (например, на нефтеперерабатывающих заводах).

**Основные функции**:

Сбор данных с датчиков и оборудования в реальном времени.

Визуализация процессов (например, через мнемосхемы).

Контроль аварийных ситуаций и формирование отчетов.

**Почему это важно?**  
Без АСОДУ управление сложными производствами было бы медленным и рискованным.

**6 слайд Зачем нужен PI System**

**Как предприятия справляются с огромным потоком данных?**  
Современные заводы, такие как нефтеперерабатывающие, оснащены тысячами датчиков, которые непрерывно собирают данные о температуре, давлении, расходе и других критически важных параметрах.

**Проблема:**

Данные поступают в режиме реального времени и в огромных объемах.

Если их просто накапливать, они превращаются в «информационный шум», который сложно анализировать и использовать.

**Решение:**  
PI System — это интеллектуальная платформа, которая:

* **Собирает данные** с датчиков, оборудования и систем.
* **Структурирует и хранит** их в удобном для анализа виде.
* **Предоставляет данные** в реальном времени, чтобы инженеры и операторы могли быстро принимать решения.

**PI System — это «мозг» предприятия, который превращает данные в полезные знания.**

**7 слайд Компоненты системы**

**PI System** — это промышленная платформа класса MES, обеспечивающая полный цикл работы с оперативными данными: от сбора до аналитики и визуализации. На Саратовском НПЗ система интегрирована с технологическим оборудованием, лабораториями и системами ручного ввода данных.

**Ключевые компоненты:**

* **PI Server**  
  Центральный узел системы, обеспечивающий:

Сбор данных с тысяч датчиков в реальном времени

Распределение информации между подсистемами

Управление безопасностью и доступом

* **PI Data Archive**  
  Специализированная база данных временных рядов с уникальными возможностями:

Хранение данных с точностью до микросекунд

Эффективное сжатие без потери точности

Быстрый доступ к архивным данным за 10+ лет

Поддержка до 500,000 значений в секунду

* **PI Asset Framework (AF)**  
  Интеллектуальная подсистема для:

Создания цифровых двойников оборудования

Построения иерархических моделей производства

Контекстного анализа данных (КПД, энергопотребление)

Прогнозирования состояния оборудования

* **PI ProcessBook**  
  Основной инструмент визуализации:

Создание интерактивных мнемосхем

Отображение трендов в реальном времени

Интеграция с Excel и другими BI-инструментами

Поддержка VBA для кастомизации интерфейсов

* **PI ACE**  
  Вычислительная среда для:

Создания сложных инженерных расчетов

Автоматического пересчета показателей

Прогнозной аналитики и оптимизации процессов

**Интеграционные возможности:**

* + **PI OPC Interface** для подключения к АСУТП
  + **PI SDK/API** для разработки custom-решений
  + **PI ODBC/OLEDB** для интеграции с корпоративными системами
  + **PI ActiveView** для веб-публикации мнемосхем

**8 слайд Представление информации реального времени на производстве**

Мы рассмотрели, как PI System организует сбор и обработку данных. Но сердцем этой системы является специализированное хранилище — PI Data Archive. Чем оно отличается от обычных баз данных?

**Специализация под временные ряды**

Оптимизирована для записи и чтения данных с временными метками (температура, давление и т.д.).

Поддерживает микросекундное разрешение времени.

**Эффективное хранение**

Сжатие данных до **90%** без потери точности.

Хранение истории за **10+ лет** без снижения производительности.

**Высокая производительность**

Запись до **500,000 значений в секунду**.

Мгновенный доступ к данным за любой период.

**Преимущества для промышленности:**

✅ **Надёжность**

Автоматическая буферизация данных при обрывах связи.

Резервирование и восстановление без потерь.

✅ **Интеграция**

Совместимость с **Grafana**, **Power BI** и другими инструментами.

Поддержка **SQL-запросов** через PI ODBC.

**9 слайд Польза для производства**

**PI System помогает решать ключевые задачи на производстве:**

Контроль производственных процессов

Система непрерывно отслеживает параметры (температуру, давление, расход).

Пример: Если температура в котле выходит за допустимые пределы, PI System сразу отправляет предупреждение.

**Автоматические расчеты**

* PI System автоматически рассчитывает важные показатели, такие как:
* Баланс массы сырья и продуктов.
* Энергопотребление оборудования.
* KPI (ключевые показатели эффективности) установок.

**Снижение аварийности**

* Анализ данных позволяет предсказать возможные проблемы, например:
* Износ оборудования.
* Риск аварийных ситуаций.
* Это помогает предотвратить простои и снизить затраты на ремонт.

**Отчеты без ручной работы**

* Инженеры больше не тратят время на ручной сбор данных.
* С помощью PI DataLink отчеты создаются автоматически, что экономит время и снижает вероятность ошибок.

**10 слайд Используемые технологии и инструменты**

Для разработки, администрирования и поддержки решений на базе PI System мы используем следующие технологии и инструменты:

**C# и VB.Net**:

Основные языки программирования для создания приложений, интеграции и автоматизации процессов.

Пример: Разработка скриптов для обработки данных или интеграции PI System с другими системами.

**.Net Framework**:

Платформа для разработки надежных и производительных приложений.

Пример: Создание для работы с данными.

**SQL и SQL Server**:

Для работы с базами данных, хранения и анализа информации.

Пример: Настройка и оптимизация баз данных, выполнение сложных запросов.

**Visual Studio**:

Основная среда разработки для создания, тестирования и отладки приложений.

Пример: Разработка модулей для PI System или интеграционных решений.

**Дополнительные технологии (упоминаем без деталей):**

* *Grafana* – визуализация данных
* *Apache Kafka* – потоковая обработка
* *Apache Airflow* – оркестрация задач
* *Keycloak* – управление доступом
* Prometheus –БДРВ для ТИС-Переработки.

**Требования к знаниям**

*«Желательные навыки кандидата»*

* Базовый SQL (SELECT, JOIN, агрегатные функции).
* Опыт работы с C# или VB.NET (написание скриптов).
* Понимание принципов работы с API.

**Устное уточнение:**  
*«Глубокие знания не требуются — всему научим. Важна готовность работать с кодом и данными»*

**НАЧАЛО ПРАКТИКИ**

**11 слайд Как мы видим PI System?**

Дима рассказал о технологиях и инструментах, которые мы используем в работе с PI System.

Я хочу вам рассказать о небольшой практической задаче, с которой мы столкнулись, как устроились на работу.

Но перед этим в общих чертах расскажу вам маленькую вводную часть.

Мы разрабатываем и поддерживаем **точную систему расчёта масс и объёмов** для резервуарного парка завода.

Для этого мы используем готовый набор расчётных модулей в виде DLL. Это позволяет нам упростить работу, снизить риски допущения ошибок и ускоряет обработку данных.

В реальной системе оператор получает готовые значения, не задумываясь о том, как они рассчитываются.

А мы для этого используем как входные так и выходные параметры. Для этого предлагаю обратиться к схеме на слайде.

Я представил вам типичный вертикальный резервуар с минимальным набором параметров.

Для расчёта мы забираем, например, уровень, температуру, плотность пи 15 градусах, градуировочную таблицу как резервуара так и понтона и т.д.

Затем, используя эти данные, получаем соответствующие выходные значения: масса общая, полезная, свободная, такие же объёмы, плотность при 20.

Для большинства параметров, что мы забираем есть датчики, мы с них получаем значения от 1 секунды до минуты. Например, уровни у нас поступают с прибора каждые 20 секунд, следовательно, каждые 20 секунд мы будем забирать из базы все необходимые входные параметры, рассчитываем, и записываем в базу готовые значения на ту же временную метку.

**12 слайд Представление объектов и временных рядов в приложениях**

Маленький пример того, как параметры и сам резервуар отражены у нас в системе.

Можно было бы прикрепить скриншот из БД, чтобы продемонстрировать, но тогда слайда точно не хватит.

Поэтому я использовал приложение SMT (System Management Tools) и выбрал маленький тестовый резервуар.

Все динамически изменяющиеся параметры мы храним в тегах (временной ряд), а чтобы закрепить их за резервуаром, помещаем их в alias этого резервуара.

Если у нас есть параметры, изменение которых происходит не чаще, чем через полгода, то помещаем его в проперти и храним их как строку или массив строк.

На прошлом слайде я упоминал, что для расчёта мы используем входные параметры. А расчёт срабатывает, если есть триггеры, мы, в качестве триггеров, используем уровень и температуру, так как они чаще всего меняются. Мы видим их значения, как указано на правом скриншоте. Значение и временная метка, обновляющаяся каждые 20 секунд.

**13 слайд Схема расчёта после перехода**

Теперь к практической задаче, с которой мы столкнулись.

Кто-то с завода подал обращение, где просит устроить переход на новую расчётную библиотеку.

Ведётся обсуждение, планирование трудозатрат, дополнительных вводных данных для успешного внедрения расчётного модуля.

Отличительная черта нового модуля от старого заключалась в том, что на вход требовалось намного больше входных параметров, нежели раньше.

Изменение согласуется и начинается работа.

**14 Схема расчёта до перехода**

До перехода на новый модуль абстрактная схема работы алгоритма выглядела следующим образом.

Как только срабатывал один из триггеров, то запускался расчёт по этому объекту. Мы вытягивали из базы актуальные значения по тегам, по пропертям, по константам.

После того, как отработал расчёт, полученные значения мы записываем в соответствующие теги.

У данного подхода есть свои недостатки, самый глобальный из них: постоянно обращаться к базе данных для получения одних и тех же значений.

Взять, например, уровень – мы настроили поступление по нему данных каждые 20 секунд. На Саратовском НПЗ около 80 резервуаров, а если взять, например,

Рязанский НПЗ, то там уже более 130 резервуаров. Итого, мы каждые 20 секунд обращаемся к базе минимум 15 раз для одного объекта. А их 130 и это без потоков и других расчётов.

Так как новый модуль подразумевал за собой использовать на входе намного больше параметров, нежели раньше, то при такой схеме сервер упадёт при растущей нагрузке на SQL примерно через сутки, двое.

Помимо этого расчёты не срабатывают мгновенно, ибо там полно проверок входных параметров, частые их конвертации и прочее. Если расчёт не успеет отработать в срок, то следующий триггер будет стоять в очереди и может накопиться очередь, которая повлечёт за собой перегруз оперативной памяти.

**15 Схема расчёта после перехода с оптимизацией**

Чтобы решить эту проблему, мы внедрили простое, но эффективное решение - промежуточный буфер для статических данных. Суть в том, что теперь все постоянные параметры (проперти и константы) загружаются в память всего один раз при первом обращении, а не запрашиваются из базы при каждом расчете. С динамическими тегами, конечно, такой фокус не пройдет - их значения постоянно меняются.

Работает это по принципу интеллектуального кэша: система сначала проверяет - есть ли нужные параметры объекта в оперативной памяти. Если находит - использует их. Если нет - однократно подгружает из базы и сохраняет для последующих обращений.

Такая оптимизация дала внушительный результат - нагрузка на SQL-сервер значительно сократилась, особенно при массовых расчетах, когда десятки объектов требуют одних и тех же справочных данных. Это как если бы вместо того, чтобы каждый раз бегать в библиотеку за справочником, вы взяли его один раз и положили на свой рабочий стол.

Подводя итоги нашего выступления, хочется отметить, что работа с промышленными системами вроде PI System — это уникальная возможность для комплексного профессионального роста. Вы не просто прокачаете технические навыки работы с данными и системами реального времени, но и глубоко погрузитесь в технологические процессы производства. Это направление учит видеть полную картину — от датчиков на оборудовании до бизнес-аналитики верхнего уровня.

Вы освоите:

* Работу с Big Data в промышленных масштабах
* Создание цифровых моделей производства
* Оптимизацию реальных технологических процессов

Если у вас остались вопросы о нашей работе или возможностях системы — будем рады на них ответить!