**Производственные и бизнес системы на примере одной из систем**

**1 слайд**

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

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

Рассмотрим сопровождение этой системы на примере нефтеперерабатывающего завода.

**2 слайд**

Нефтедобыча и нефтепереработка в нашей стране - это очень крупный сектор экономики.

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

И для каждого из этих заводов спутником прикреплён офис компаний Сибинтек и Сибинтек-Софт.

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 ИТ процессов для всех предприятий компании.

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

Но для начала нужно понимать, что СИБИНТЕК и СИБИНТЕК–СОФТ – это дочерние компании Роснефти.

Для полного понимания системы, о которой сегодня пойдёт речь нужно немного углубиться в прикладную область, а именно в нефтепереработку.

**3 слайд**

Структура технической поддержки производственных систем.

Вся поддержка всегда крутится вокруг заказчика, а точнее вокруг системы, которой он пользуется.

И если рассматривать наше место в структуре технической поддержки производственных систем, то мы относимся к 2-ой, 3-линиям поддержки.

1-линия поддержки занимается решением простых задач (консультации) и распределением, маршрутизацией обращений от пользователя по различным проблемам ИТ.

2-линия поддержки - это наше управление. Мы занимаемся сопровождением и внедрением продуктов, которые уже находятся в промышленной эксплуатации. Мы отвечаем за решение задач, которые прямо сейчас волнуют пользователя.

3-линия поддержки - это эксперты в части знаний работы системы и бизнес процессов, которые общаются с заказчиками, формируют тз для разработчиков, решают задачи сложного уровня. Также они занимаются формированием пакетов обновления информационных систем. Непосредственно само обновление и необходимые доп. настройки осуществляет 2-линия.

4-Поддержка от разработчика. Это люди, которые общаются только с 3-ей линией. «Больше людей они не видят». Сидят и пишут код по ТЗ, которое им было сформировано.

5-Далее отдел безопасности, который контролирует защищенность всех и вся. Отдельный отдел, которого боятся все, ведь если нарушать их правила, то могут уволить)

6-Поддержка серверов и серверного оборудования – это, так называемые, сетевики и системные администраторы, которые всё знают об межсетевых экранах, шлюзах, коммутаторах, серверах и т.п.

7-Выполнение нетиповых запросов на изменение делают абсолютно все выше сказанные группы поддержки. Это то, что не входит в их стандартный функционал и является дополнительным объемом работы.

8- За управление сервисом и организацией работ по развитию системы отвечает отдельное подразделение, но частично его функции ложатся на плечи всех линий поддержки, ведь эта работа всегда комплексная и требует изменение на всех уровнях поддержки.

**4 слайд**

Предметной частью 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 его с IT технологиями.

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

Представьте завод, как огромную производственную цепочку, где каждая установка — это отдельный цех со своей специализацией.

Вот как это работает:

1. Началом производства является база нефти, где принимается сырая нефть, поступающая по трубопроводу, либо после отгрузки танкера и ЖД цистерн.

2. Начинается нефтепереработка с установки ЭЛОУ-АВТ-6, которая предназначена для обессоливания и обезвоживания сырой нефти, атмосферной перегонки обессоленной нефти, вакуумной перегонки мазута, стабилизации бензина с целью получения сырья для других установок. Из сырой нефти, поступающей на установку ЭЛОУ-АВТ-6, на блоке обессоливания удаляются хлористые соли, вода и механические примеси. Обессоленная и обезвоженная нефть с блока ЭЛОУ поступает на атмосферный блок установки, где отбираются светлые фракции бензина, керосина и дизельного топлива, а мазут подается на вакуумный блок для получения вакуумного газойля и гудрона. Отбираемые на блоке АТ (атмосферная трубчатка) бензиновые фракции направляются на блок стабилизации.

Затем на АВТ (атмосферно-вакуумная трубчатка) нефть нагревают до 350°C и разделяют в 65-метровой колонне.

Вверх поднимаются легкие фракции (бензиновые).

В середине — дизельные.

Внизу остается мазут.

3. УПЭС - Предназначена для производства элементарной серы из сероводородного газа и регенерации насыщенного раствора МЭА.

4. УПБ - Предназначена для получения битумов строительных, дорожных и кровельных марок посредством окислительных емкостей.

5. Л-35/100-300 и Л-35/100-600 – принимает прямогонную бензиновую фракцию и посредством реакторных блоков гидроочистки, блоков стабилизации и каталитического риформинга происходит приготовление высокооктановых бензинов.

6. Л-24-6 – Предназначен для повышения качества дизельной фракции с установки ЭЛОУ АВТ.

7. Изомеризацию не рассказываем, но если есть вопрос (Входит в состав комплекса изомеризации и предназначена для переработки пентан-гексановой фракции высокооктановый компонент товарного бензина по технологии низкотемпературной изомеризации «Изомалк-2».)

8. Отгрузка – Финальная часть производства, где готовая продукция с паспортом качества отгружается в жд/авто/танкер/трубопровод и направляется к потребителям.

**5 слайд**

В соответствии с моделью CIM все информационные системы предприятия делятся на уровни по двум критериям: дискретность оси времени, в единицах которой функционируют информационные системы, и объемы данных, обрабатываемых на каждом уровне модели. В рамках концепции CIM информационные системы образуют пирамиду. Системы верхнего уровня оперируют данными на относительно больших временных промежутках, а нижнего — имеют дело с большим потоком данных реального времени. Каждое сечение пирамиды имеет площадь, пропорциональную объему обрабатываемых данных. На вершине этот объем минимален, в основании — максимален. Для связи дискретной оси времени наверху пирамиды с событиями реального времени в ее основании используются промежуточные системы цехового уровня, (Production Control, или Manufacturing Execution Systems — MES).

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-система (Supervisory Control And Data Acquisition) 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-система (Manufacturing Execution System) — это комплекс программных средств для управления производственными процессами на предприятиях.

Она служит мостом между планированием на уровне предприятия и фактическим производством, обеспечивая эффективное выполнение производственных операций в реальном времени. MES получает данные из SCADA или напрямую от датчиков.

Основные функции MES-систем:

* Контроль производственных операций. Система позволяет в режиме реального времени следить за выполнением производственных заданий, отслеживать отклонения от плана и оценивать загрузку оборудования.
* Сбор и обработка данных. Системы фиксируют ключевые параметры: от производительности оборудования до объёмов продукции и расхода ресурсов. Это помогает выявлять проблемные участки, оптимизировать процессы и повышать эффективность.
* Контроль качества. MES интегрируют инструменты проверки качества, позволяя оперативно выявлять и устранять дефекты. Чаще всего по средствам диспетчеризации.
* Прослеживаемость производства. Системы хранят полную историю создания продукции: от использованных материалов до этапов сборки. Это упрощает решение вопросов с клиентами и помогает соответствовать требованиям законодательства.
* Интеграция с другими системами. MES становятся частью общей цифровой экосистемы предприятия, эффективно взаимодействуя с ERP, APS и другими инструментами.

4 уровень — управление бизнес-процессами в ERP

ERP (enterprise resource planning) – 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.

**6 слайд**

Одной из наиболее важных MES систем является АСОДУ (Автоматизированная система оперативно-диспетчерского управления)  
АСОДУ — это комплекс приложений для сбора, анализа, представления и архивирования информации, поступающей с разных цехов и участков предприятия или локальных подсистем АСУТП (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.
* Отображение расчётных данных по метрологическим методикам.
* Управление производством посредством контроля движения нефтепродуктов.

Почему это важно?

Без АСОДУ управление сложными производствами было бы медленным и рискованным.

Также у нас теперь весь завод может уместиться на одном сайте, который получает оперативные данные с каждого датчика, с каждой смежной информационной системы.

Почему без АСОДУ невозможно современное производство?

Ручной сбор данных устарел:

Раньше инженеры тратили часы на обход оборудования с блокнотами. Теперь АСОДУ делает это автоматически — без ошибок и задержек.

Человеческий фактор:

Оператор физически не может отслеживать сотни параметров одновременно. АСОДУ фиксирует все отклонения и выделяет самые важные.

Экономический эффект:

Внедрение АСОДУ на НПЗ может дать до 20% снижения аварийности и 5–15% экономии ресурсов за счёт оптимизации процессов.

**7 слайд**

Но для корректного отображения с обработки данных для системы АСОДУ необходим очень мощный инструмент. Надо понимать, что 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:

БДРВ - Базы данных временных рядов

Если у вас есть упорядоченные по времени данные с временными метками, такие как метрики от инфраструктуры или данные датчиков, может быть полезно использовать одну из баз данных временных рядов.

Принцип хранения данных в базе данных временных рядов. Как мы видим, важной особенностью является наличие столбца соответствующего времени события.

Общие характеристики баз данных временных рядов:

* Данные временных рядов всегда собираются на протяжении определенного периода времени.
* Данные из рабочих нагрузок являются новыми и записываются как вставки. Уже существующие данные не обновляются путем замены.
* Когда данные записываются, они автоматически назначаются последнему интервалу времени.

Базы данных временных рядов часто используются для осуществления мониторинга различных метрик (будь то загрузка CPU, или показатели работы какого-либо датчика).

**8 слайд**

БДРВ, которое используем мы:

PI System — это интеллектуальная платформа, которая:

* Собирает данные с датчиков, оборудования и других систем. (Таких как система отгрузки причал/жд/авто, лаборатория и т.п.)
* 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.

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**

Plant Information System (PI System) – комплекс программного обеспечения, обеспечивающий управление данными, включая сбор, хранение, обработку и их представление по компании, предприятиям и отдельным процессам. PI System состоит из следующих компонентов:

* Источники данных: программное обеспечение, генерирующее данные. Они могут быть разнообразными и подключаться к узлам интерфейса несколькими способами. Серверные приложения обработки данных PI ACE и Totalizer также являются источниками данных, хотя и могут располагаться на компьютере, где установлены PI сервера
* Интерфейсы: программное обеспечение, получающее данные от источников данных и отправляющее их на PI сервера. Каждому типу источников данных необходим соответствующий интерфейс PI, который может его интерпретировать. В OSIsoft имеется более 400 различных интерфейсов
* Сервера PI: служат для хранения данных и выступают в качестве серверов данных для клиентских приложений на основе Microsoft Windows. PI Server можно также применять для взаимодействия с данными, сохраненными во внешних системах
* Серверные приложения PI: в PI System включены многие продукты так называемого среднего слоя, которые выступают в роли серверных приложений. В их число входят: PI ACE, PI Notifications, объектные базы данных PI-AF и веб-порталы на основе Microsoft SharePoint и SAP NetWeaver
* Клиентские приложения: операторы, инженеры, менеджеры и другой персонал компании используют разнообразные клиентские приложения для подключения к серверам PI и серверам приложений PI для просмотра интересующих данных.

Pi System представляет из себя систему MES уровня, необходимую для диспетчерского управления техническим процессом. Она обеспечивает сбор, хранение, анализ и визуализацию данных из таких источников, как: АСУТП, средств ручного ввода, лабораторного и интегрированного с PI программного обеспечения. На Саратовском НПЗ система представлена такими компонентами, как:

* PI Server – ядро системы PI System. Он получает данные и распределяет их в реальном времени с помощью компонентов PI System и всей информационной инфраструктуры.
* PI Data Archive – специализированная БД для хранения данных временных рядов, с уникальным алгоритмом сжатия и скоростью восстановления данных. PI Data Archive был специально спроектирован для обработки и хранения данных реального времени и, в отличие от реляционных хранилищ, является базой данных временных рядов. Благодаря этому он легко масштабируется и может хранить огромное количество точных, согласованных производственных данных за много лет, с разрешением времени вплоть до микросекунд, доступных оперативно и без потери точности.
* PI Asset Framework (AF) – инструмент для создания организационных и технологических моделей и схем производства, состоящих из элементов и их взаимосвязей и взаимозависимостей. Элементы модели представляют собой как физическое оборудование (резервуары, теплообменники, смесители, измерители и т.д.), так и более абстрактные логические конструкции (эффективность, экология и т.д.). Разработанная модель позволяет включить в себя организационно-технологическую структуру предприятия, логику его функционирования, принципы контекстно-зависимого представления данных о производственном процессе. Основываясь на единой модели, доступной любым приложениям PI System, можно проводить различного рода анализ, обработку и представление информации о производстве, поступающую в реальном времени. Например, согласование технологических данных, расчет эффективности, анализ простоев технологического оборудования.
* PI ProcessBook – основное клиентское приложение в PI System для разработки графического интерфейса пользователя, основанного на мнемосхемах, трендах, объединяющих и структурирующих всю производственную информацию, необходимую для быстрого и эффективного принятия решений. Приложение PI ProcessBook содержит одну или более книг ProcessBook, представляющих собой наборы "записей отображений". Запись отображения в PI ProcessBook может содержать данные из следующих источников (по отдельности или одновременно):
* Архивы данных PI на одном или нескольких серверах, в том числе данных, поступающих от тегов измерения и расчетных данных
* Другие базы данных, поддерживающие ODBC.
* Электронные таблицы, документы, схемы, графики, фотографии и другие приложения Windows, подключаемые через OLE и встраиванием
* Процедуры Visual Basic for Applications

С помощью PI ProcessBook можно создавать графические мнемосхемы, содержащие текущие и архивные данные о производственном процессе, включая специализированные инструменты для быстрой оценки состояния производства всеми сотрудниками. Инструменты рисования и возможность встраивать тренды текущих и архивных данных дают пользователям всего предприятия невероятно богатое и ясное представление о производственных процессах. Microsoft Visual Basic for Applications (VBA), встроенный в ProcessBook, позволяет строить экраны для любых частей производственного процесса.

PI ProcessBook позволяет отображать данные из любой системы, работающей в сфере бизнеса и управления операциями. В среде ProcessBook легко обработать и проанализировать данные, построить тренд. Экраны PI ProcessBook могут быть связаны друг с другом для мгновенного доступа к дополнительным данным, позволяя вам быстро обнаруживать производственные проблемы.

**10 слайд**

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.

**11 слайд**

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

C#,Python, 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.

IIS:

IIS (Internet Information Services) — это веб-сервер, разработанный компанией Microsoft для своих операционных систем.

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.

Маленькое уточнение:

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

Наше управление не разрабатывает с нуля какие-либо системы и продукты. Мы занимаемся реляционным программированием внутри уже созданных приложений, используя данные технологии, для решения поступающих задач. Об одной из них Вам сейчас расскажет мой коллега Даниил.

**12 слайд**

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 при 15 градусах, градуировочную таблицу как резервуара, так и понтона и т.д.

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

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

**13 слайд**

Маленький пример того, как входные, выходные 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 использую скриншот из приложения SMT (System Management Tools) и описание маленького тестового резервуара.

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 раз в 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. 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.

**14 слайд**

Теперь к практической задаче, с которой мы столкнулись.

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

Ведётся обсуждение, планирование трудозатрат, дополнительных входных 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 с ключевыми пользователями системы и начинается работа.

**15 слайд**

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, расчетные 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 одном нефтеперерабатывающем заводе их около 80 резервуаров, а если взять другой завод, 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 подразумевает за собой использование на входе намного большего количества входных параметров, чем раньше, то при старой схеме работы расчетов сервер приложения «упадёт» при растущей нагрузке на БДРВ примерно через сутки, двое.

Помимо этого, расчёты не срабатывают мгновенно, т.к. много проверок входных параметров, частые их конвертации и прочее. Если расчёт не успеет отработать в срок, то следующий триггер будет стоять в очереди и может образоваться большая очередь, которая повлечёт за собой переполнение оперативной памяти сервера.

**16 слайд**

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 сервер БДРВ 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.

**17 слайд**

**Финал**

Подводя итоги нашего выступления, хочется отметить, что работа с производственными системами вроде АСОДУ на базе 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 оборудования до бизнес-аналитики верхнего уровня.

Вы освоите:

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

От нас зависит прибыльность бизнеса заказчика, ведь производство должно происходить в круглосуточном режиме, без остановок и задержек. Мы не просто решаем проблемы заявителей, но и поясняем, что именно пошло не так. И таким образом влияем на профессионализм сотрудников заказчика.

Приглашаю всех в ряды Сибинтек-Софта. Подумайте над стартом своей карьеры вместе с нами. У нас вы точно получите разнообразные и интересные задачи, для решений которых, вы изучите новые it-технологии.

Вопросы?