****

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное**

**учреждение Московской области**

**«Люберецкий техникум имени Героя Советского Союза,**

**лётчика-космонавта Ю.А. Гагарина»**

**ДИПЛОМНЫЙ ПРОЕКТ**

На тему: Разработка программного решения для миграции данных об электронной структуре изделия из PDM-системы Teamcenter в PDM-систему IPS средствами .Net Framework для компании Национальный центр вертолётостроения имени М.Л. Миля и Н.И. Камова

по специальности: 09.02.07 Информационные системы и программирование

|  |  |  |
| --- | --- | --- |
| Выполнил |  | В.С. Антонюк |
| Руководитель |  | Б.В. Гребенюк |

ДОПУЩЕН К ЗАЩИТЕ

Заместитель директора по информационным технологиям

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А.В.Капранова /

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025г.

**Введение**

Современные производственные процессы требуют эффективного управления данными о продукции, начиная от проектирования до выпуска готовой продукции. В этом контексте особую роль играют системы управления инженерными данными, такие как Product Data Management (PDM). Эти системы позволяют хранить, управлять и контролировать информацию о продукте на всех этапах жизненного цикла изделия. Однако стандартные функциональные возможности таких систем могут оказаться недостаточными для удовлетворения специфических потребностей предприятий, особенно в условиях быстрого развития технологий и роста сложности производимых изделий.

**Актуальность исследования.** Развитие промышленности требует внедрения новых или уникальных подходов к управлению инженерными данными, что связано с необходимостью адаптации существующих PDM-систем под требования конкретных компаний. Внедрение дополнительных модулей позволяет расширить функциональность системы, повысить эффективность работы инженеров и технологов, а также сократить временные затраты на выполнение различных операций. Это делает разработку специализированных модулей для PDM-систем актуальной задачей для многих промышленных предприятий.

**Цель исследования.** Целью данной работы является разработка специализированного модуля расширения для PDM-системы IPS, который позволит интегрировать дополнительные функции, необходимые для оптимизации процессов управления инженерными данными на предприятии. Модуль будет ориентирован на решение задач по миграции данных между системами.

**Задачи исследования.** Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ текущих возможностей и ограничений используемой PDM-системы IPS.
2. Определить ключевые потребности предприятия в области управления инженерными данными.
3. Разработать архитектуру и функционал нового модуля расширения.
4. Реализовать модуль с учетом требований безопасности и совместимости с существующей системой.
5. Провести тестирование разработанного модуля и оценить его эффективность в реальных производственных условиях.

Решение этих задач позволит создать эффективный инструмент для управления инженерными данными, соответствующий требованиям современного производства и способствующий повышению конкурентоспособности предприятия.

**Глава 1 АНАЛАЗИ РАЗРАБОТКИ ПРОГРАММНОГО РЕШЕНИЯ**

* 1. **Анализ литературных источников и электронных ресурсов по теме дипломного проекта**

Важной частью дипломного проекта является обзор литературных источников и электронных ресурсов, в которых рассматриваются достижения и актуальность выбранной темы.

После тщательного изучения различных источников информации, включая научные статьи, учебные пособия, официальную документацию и мнения экспертов, можно сделать ряд значимых выводов относительно разработки модуля расширения для PDM-системы IPS. Одним из важнейших аспектов является правильный подход к проектированию и использованию модульной архитектуры.

Для лучшего понимания нужно разобраться в том, что такое PDM-система. Это система управления данными об изделии. В качестве изделий могут рассматриваться как обычные детали, так и большие сборочные единицы. С помощью нее осуществляется отслеживание большого объема данных, необходимых на разных этапах жизненного цикла сборки или конструирования.

В статье "Роль и значение PDM-систем при разработке технологического оборудования" Тимофеева Павла Геннадьевича и Ягопольского Александра Геннадиевича подчеркивается необходимость гибкой и масштабируемой структуры, что подтверждается и в учебном пособии "Системы PDM: Возможности и проблемы" Зайнитдинова Я.Р. Оба источника указывают на то, что такая архитектура позволяет легко адаптировать систему под нужды конкретного предприятия, минимизируя затраты времени и ресурсов [1].

Как пишет автор Зайнитдинова Я.Р. учебного пособия "Системы PDM: Возможности и проблемы": «IPS предоставляет широкий спектр инструментов и библиотек, что значительно упрощает задачу программистов и инженеров». Это подтверждает тот факт, что платформа IPS ориентирована на максимальное удобство для разработчиков, предоставляя им все необходимое для создания качественных и надежных решений.

Еще одно важное наблюдение касается возможностей официального руководства IPS "Руководство администратора и программиста IPS" от компании ИНТЕРМЕХ. Этот документ предоставляет исчерпывающую информацию о работе с системой, включая подробные описания API-интерфейсов и примеры кода. Как говорится в руководстве: «API IPS предоставляет доступ ко всем основным функциям системы, позволяя разработчикам создавать мощные и гибкие решения для автоматизации инженерных процессов».

Эксперты на блогах таких как: «Разработка расширений на Python» также делятся полезными рекомендациями, такими как использование Python для разработки модулей благодаря его удобству и доступности готовых библиотек. Один из участников форума отмечает: «Я использовал Python для разработки модуля расширения для IPS. Этот язык оказался очень удобным благодаря своей простоте и большому количеству готовых библиотек». Это мнение отражает общую тенденцию среди разработчиков, предпочитающих Python за его универсальность и легкость освоения.

Стоит отметить, что разработка модуля расширения для PDM-системы IPS требует не только знаний в области программирования, но и глубокого понимания принципов работы самой системы. Владение нужными инструментами и языками программирования играет ключевую роль, равно как и умение учитывать специфику предприятия. Платформа IPS предлагает богатые возможности для интеграции и разработки дополнительных модулей, что делает её отличным выбором для автоматизации инженерных процессов.

Обсуждение технических вопросов на форумах разработчиков IPS и в блогах опытных инженеров-программистов также имеет большое значение. Здесь активно обсуждаются такие аспекты, как работа с базами данных, интеграция с внешними системами и оптимизация производительности. Участники этих дискуссий часто делятся личным опытом и предлагают проверенные временем решения, что делает такие площадки важными источниками информации для начинающих разработчиков.

Подводя итог, можно сказать, что успешная разработка модуля расширения для PDM-системы IPS требует комплексного подхода, включающего понимание архитектурных принципов, знание инструментов и языков программирования, а также учет специфики предприятия. Система IPS предоставляет все необходимые средства для создания и интеграции дополнительных модулей, что делает её привлекательным решением для автоматизации инженерных процессов. Далее можно разобраться с тем, чем работает наша PDM-система.

* 1. **Анализ предметной области**

Перед тем как приступить к непосредственному анализу предметной области, важно дать определение ключевым терминам, используемым в данном контексте. PDM-система представляет собой специализированный инструмент для управления данными о продукции, обеспечивающий централизованное хранение информации о различных аспектах жизненного цикла продукта, начиная от проектирования и заканчивая эксплуатацией. IPS — это конкретная PDM-система, применяемая в данной организации, для которой планируется создание модуля расширения. Модуль расширения — это дополнительная функция или набор функций, которые интегрируются в базовую систему для выполнения специализированных задач, не предусмотренных стандартной версией PDM-системы.

Целью данного анализа является идентификация ключевых бизнес-процессов, связанных с управлением продукцией, и выявление их взаимозависимостей внутри организации. Также необходимо обнаружить текущие проблемы и ограничения, присущие существующим процессам, которые могут быть устранены за счет внедрения нового модуля. Важно оценить возможности интеграции дополнительных функций в действующую инфраструктуру без ущерба для стабильности и производительности основной системы. Кроме того, формулировка требований к разрабатываемому модулю должна основываться на выявленных потребностях и ожиданиях конечных пользователей.

На сегодняшний день в организации используется PDM-система IPS для управления жизненным циклом продукции. Эта система предоставляет возможности хранения данных о продуктах, управления версиями документации, контроля этапов проектирования и производственного процесса, а также обеспечения взаимодействия между различными отделами предприятия. Тем не менее, существуют определенные недостатки, снижающие эффективность ее применения. К ним относятся отсутствие автоматизации процесса обновления технической документации при изменениях параметров продукта, ограниченность функционала управления конфигурациями изделий, что затрудняет контроль версий и внесение корректировок, а также недостаточное развитие интеграции с системами автоматизированного проектирования (CAD), что усложняет обмен данными между специалистами разных направлений. Эти факторы ведут к увеличению временных затрат на выполнение стандартных операций, снижению качества производимой продукции и повышают вероятность возникновения ошибок.

Для устранения вышеупомянутых проблем необходим модуль расширения, обладающий следующими возможностями. Во-первых, это автоматизированное обновление технической документации при внесении изменений в характеристики изделия, что обеспечит актуальность информации и снизит вероятность ошибок. Во-вторых, улучшенное управление конфигурациями продуктов, включая поддержку версий и ведение истории изменений, что повысит прозрачность и управляемость процессом разработки. В-третьих, интеграция с CAD-системами для облегчения обмена данными между проектировщиками и инженерами, что ускорит процессы согласования и внесения правок. Наконец, поддержка многоверсионности данных, позволяющая нескольким пользователям параллельно работать над различными версиями одного и того же проекта, что увеличит гибкость и скорость работы команды.

При разработке и внедрении нового модуля необходимо учитывать следующие потенциальные риски. Прежде всего, возможно несоответствие нового функционала законодательным нормам и отраслевым стандартам. Также возможны проблемы совместимости с уже установленными программными продуктами и аппаратным обеспечением. Еще одной проблемой может стать потребность в обучении сотрудников работе с новым модулем, что может временно замедлить производственные процессы. Наконец, увеличение нагрузки на серверную инфраструктуру может привести к снижению общей производительности системы.

Проведённый анализ предметной области продемонстрировал необходимость разработки модуля расширения для PDM-системы IPS, который позволит устранить существующие недостатки и значительно повысить эффективность управления данными о продукции. Для успешной реализации этого проекта необходимо тщательно проработать план действий, учитывая все выявленные риски, и предусмотреть этапы тестирования и внедрения нового функционала.

* 1. **Аналоги PDM-системы и PLM-системы**

Управление данными об изделиях (Product Data Management, PDM) является критически важным компонентом цифровой трансформации промышленных предприятий. Современные PDM-системы обеспечивают централизованное хранение, контроль версий, управление изменениями и интеграцию данных между этапами жизненного цикла изделия (PLM). Однако выбор конкретного решения зависит от множества факторов, включая масштаб предприятия, отраслевую специфику и требования к интеграции с CAD/CAM-системами. В рамках данного раздела проведен анализ ключевых PDM-систем, представленных на рынке, с целью выявления их особенностей, преимуществ и ограничений, а также обоснования целесообразности разработки инструмента миграции данных между системами Teamcenter и IPS.

Если рассматривать современные решения, то на сегодняшний день доминирующее положение на рынке занимают следующие PDM-системы:

* Siemens Teamcenter — решение от Siemens Digital Industries Software, ориентированное на крупные промышленные предприятия. Система обеспечивает управление структурами изделий (BOM), интеграцию с CAD-пакетами (NX, Solid Edge) и ERP-системами (SAP), а также поддерживает стандарты ISO 10303 (STEP) для обмена данными. Ключевым преимуществом Teamcenter является гибкость настройки под специфические требования, включая реализацию цифровых двойников (Digital Twin). Однако высокая стоимость лицензий и сложность внедрения делают его менее доступным для малого бизнеса.
* IPS — система, используемая на исследуемом предприятии. Её функционал сфокусирован на управлении технологическими процессами, формировании электронных спецификаций и анализе производственных метрик. Преимущество IPS — глубокая адаптация под внутренние стандарты предприятия, что снижает операционные затраты. Вместе с тем, система имеет ограниченную интеграцию со сторонним ПО и отсутствие облачной версии, что сужает область её применения.
* PTC Windchill от PTC выделяется встроенной аналитикой на базе искусственного интеллекта и поддержкой IoT через платформу ThingWorx. Система предлагает удобный веб-интерфейс и инструменты Agile-управления, но сталкивается с ограничениями при работе со сложными многоуровневыми BOM-структурами. Windchill востребован в отраслях с высокой динамикой изменений, таких как электроника и медицинское оборудование.
* Dassault Systèmes ENOVIA является частью экосистемы 3DEXPERIENCE, обеспечивая облачную коллаборацию в режиме реального времени и интеграцию с CAD-системой CATIA. Её ключевое преимущество — возможность глобальной совместной работы над проектами, однако высокая стоимость обучения и сложность миграции данных из legacy-систем ограничивают её применение преимущественно авиационной и судостроительной отраслями.
* Autodesk Fusion Lifecycle (ранее PLM 360) позиционируется как SaaS-решение для малого и среднего бизнеса. Низкий порог входа, подписка по модели «облачного сервиса» и интеграция с Autodesk Fusion 360 делают его популярным среди стартапов. Однако система не поддерживает сложные BOM-структуры и обладает ограниченными возможностями аналитики.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PDM-система | Разработчик | Ключевые особенности | Преимущества | Недостатки | Целевые отрасли |
| Siemens Teamcenter | Siemens | Цифровые двойники, интеграция с NX, SAP | Гибкость, поддержка крупных предприятий | Высокая стоимость, сложность внедрения | Автомобилестроение, аэрокосмос |
| IPS | Intermech | Локализация под стандарты предприятия | Низкие операционные затраты | Слабая интеграция с CAD | Тяжелое машиностроение |
| PTC Windchill | PTC | IoT-аналитика, Agile-управление | Веб-интерфейс, AI-аналитика | Ограничения в работе с большими BOM | Электроника, медтехника |
| ENOVIA | Dassault Systèmes | Облачная коллаборация, симуляции | Глобальная работа в реальном времени | Высокий порог входа | Авиация, судостроение |
| Fusion Lifecycle | Autodesk | Low-code, SaaS-модель | Доступность для малого бизнеса | Ограниченная аналитика | Стартапы, образование |

Разработка инструмента миграции данных между Teamcenter и IPS обусловлена спецификой производственных процессов на предприятии. Teamcenter, как «тяжелая» PDM-система, обеспечивает глубокую детализацию проектных данных, включая 3D-модели и инженерные расчеты. В свою очередь, IPS фокусируется на управлении техпроцессами и формировании производственных спецификаций, что требует точной передачи параметров (материалы, допуски, версионность). Необходимость синхронизации данных между конструкторским и производственным сегментами делает интеграцию этих систем приоритетной задачей.

Кроме того, анализ аналогов показал, что универсальных PDM-решений не существует. Например, Windchill и ENOVIA, несмотря на продвинутые функции аналитики, не адаптированы под локальные стандарты предприятия, а Autodesk Fusion Lifecycle не поддерживает сложные BOM-структуры, характерные для тяжелого машиностроения. Таким образом, миграция данных между Teamcenter и IPS является оптимальным решением для обеспечения сквозной цифровизации процессов.

В заключении к этому разделу можно сделать вывод, что Проведенный анализ подтвердил, что выбор PDM-системы определяется отраслевыми требованиями и масштабом предприятия. Разработка решения для миграции данных между Teamcenter и IPS актуальна для предприятий, совмещающих проектирование сложных изделий с требовательной производственной аналитикой. Унификация данных между этими системами позволит создать единое информационное пространство, необходимое для реализации концепции Industry 4.0. Дальнейшее развитие проекта может включать расширение функционала для поддержки облачных платформ и интеграции с AI-алгоритмами, что соответствует глобальным трендам цифровизации промышленности.

* 1. **Инструменты продукта**

В рамках продукта был выбран язык программирования C# и .NET Framework для разработки решения миграции данных между TeamCenter и IPS. Выбор обусловлен следующими факторами:

* **Интеграция с Windows-окружением:**

Обе PDM-системы (Teamcenter и IPS) развернуиы у нас в корпоративной Windows-среде. .NET Framework предоставляет нативную поддержку Windows API, COM-объектов и служб, что критично для взаимодействия с промышленными PDM-системами.

* **Поддержка промышленных стандартов:**

Teamcenter предоставляет API на C++ и Java, но .NET (через COM-интерфейсы или REST/SOAP) часто используется для интеграции, что очень важно для дальнейшей реализации интеграции. IPS также поддерживает .NET-библиотеки для работы с данными. C# упрощает работу с этими API.

* **Производительность и безопасность:**

C# — строго типизированный язык с управляемой памятью (в отличие, например, от Python), что снижает риски ошибок при обработке больших объемов данных. Поддержка многопоточности (Task Parallel Library) ускоряет миграцию.

* **Экосистема .NET для Enterprise-решений:**

Встроенные возможности для работы с XML, JSON, базами данных (ADO.NET), а также поддержка SOAP/WCF для веб-сервисов упрощают взаимодействие с PDM-системами.

Важно отметить альтернативные инструменты, некоторые из них уже используются на предприятии, например, язык программирования Java.

|  |  |  |
| --- | --- | --- |
| Язык/Платформа | Плюсы | Минусы для данного проекта |
| Java | Кроссплатформенность, богатые библиотеки (Apache POI, JAXB) | Сложнее интеграции с Windows-специфичными API Teamcenter, менее привычен для .NET-разработчиков |
| Python | Простота, богатые библиотеки(pandas для данных, requests для API) | Низкая производительность при больших объемах данных, динамическая типизация повышает риски ошибок |
| C++ | Высокая производительность, прямой доступ к API Teamcenter | Сложность разработки, отсутствие встроенных средств для обработки структур данных |

Выбор C# и .NET Framework для разработки модуля миграции данных между Teamcenter и IPS обусловлен их уникальной совместимостью с Windows-ориентированной инфраструктурой предприятия и промышленными стандартами интеграции. Платформа обеспечивает прямую работу с COM-объектами Teamcenter и REST-интерфейсами IPS, что исключает необходимость сложных обёрток и снижает риски несовместимости. Строгая типизация C# и встроенные механизмы многопоточности гарантируют высокую производительность при обработке миллионов компонентов, а поддержка XML, SOAP и ADO.NET упрощает реализацию ETL-процесса в рамках enterprise-окружения.

Анализ альтернатив (Java, Python, C++) подтвердил, что их недостатки — будь то низкая производительность Python, сложность интеграции Java с Windows-API или отсутствие в C++ инструментов для работы с данными — критичны для масштаба проекта. Кроме того, доступность специалистов по C# в промышленном сегменте обеспечивает долгосрочную поддержку решения.

Таким образом, выбор C# и .NET Framework обеспечивает не только техническую реализуемость, но и стратегическую устойчивость модуля, соответствующего требованиям Industry 4.0 и корпоративным стандартам предприятия.

**1.5 Требования к дипломному продукту**

Дипломный проект направлен на разработку программного модуля для автоматизации миграции данных из внешней системы хранения компонентов (объемом свыше 8 млн объектов) в целевую базу данных предприятия. Модуль должен обеспечить корректность, производительность и безопасность процесса передачи данных, а также соответствовать требованиям эргономики и интеграции в существующую IT-инфраструктуру.

Основной задачей модуля является обеспечение сквозной миграции данных, начиная с их извлечения из внешней системы и заканчивая загрузкой в целевую БД. Этап извлечения предполагает подключение к внешнему хранилищу через REST API или файловый интерфейс, а также пакетную выгрузку компонентов с поддержкой потокового чтения для минимизации нагрузки на память. Для обработки данных объемом 8+ млн объектов критически важна фильтрация по критериям: тип компонента, дата модификации и статус валидации.

Этап обработки данных базируется на принципах объектно-ориентированного программирования (ООП). Каждый компонент инкапсулируется в классы-сущности (например, Component, Metadata), что позволяет реализовать наследование для поддержки различных типов объектов (детали, сборки, документы) и полиморфизм для гибкой валидации. Проверки включают контроль форматов полей (строки, числа, даты), уникальность идентификаторов и ссылочную целостность (наличие родительских сборок в целевой системе).

Этап загрузки в целевую БД требует использования ORM (Entity Framework) для маппинга объектов в реляционные структуры. Для обеспечения надежности операции выполняются в транзакционном режиме: при возникновении ошибок изменения откатываются, а результаты фиксируются в отчетах.

|  |
| --- |
|  |
| Рисунок N – интерфейс решения |

Учитывая, что модуль будет интегрирован в рабочую среду предприятия, эргономика интерфейса играет ключевую роль. Визуальное оформление разрабатывается в неброской цветовой палитре (серые, синие тона) с размером окна не более 800×600 пикселей. Это минимизирует отвлечение сотрудников и обеспечивает компактное отображение даже на мониторах с ограниченным разрешением.

Элементы управления включают прогресс-бар для отслеживания статуса миграции, таблицу с логом операций (время, тип события, описание) и кнопки управления (Старт, Пауза, Отмена, Экспорт отчета). Для интеграции в существующие workflows модуль поддерживает два режима запуска:

* Интерактивный — через контекстное меню или главное окно PDM-системы.
* Фоновый — автоматическое выполнение по расписанию при старте системы.

Обработка ошибок реализуется через всплывающие уведомления с классификацией (критические, предупреждения) и сохранением логов в форматах CSV/PDF. Это упрощает анализ инцидентов без необходимости прямого доступа к коду.

Для подтверждения соответствия модуля заявленным критериям предусмотрены:

* Юнит-тесты (NUnit) — проверка методов валидации (покрытие ≥ 90%).
* Интеграционные тесты (Postman, SoapUI) — эмуляция работы с внешними системами.
* Нагрузочное тестирование (JMeter) — оценка производительности при обработке 10 тыс. объектов.
* User Acceptance Testing (UAT) — апробация сотрудниками предприятия.

Сформулированные требования охватывают все аспекты разработки — от функциональности и интерфейса до безопасности и производительности. Их выполнение гарантирует создание модуля, который не только решает задачу миграции данных, но и интегрируется в производственные процессы предприятия без disruption. Дальнейшая работа будет направлена на реализацию требований в техническом проекте и коде системы с последующей валидацией через описанные тестовые сценарии.

**ГЛАВА 2 АНАЛИЗ РАЗРАБОТКИ ПРОГРАММНОГО РЕШЕНИЯ ДЛЯ МИГРАЦИИ ДАННЫХ ИЗ PDM СИСТЕМЫ TEAMCENTER И PDM СИСТЕМЫ IPS**

**2.1 Диаграммы**

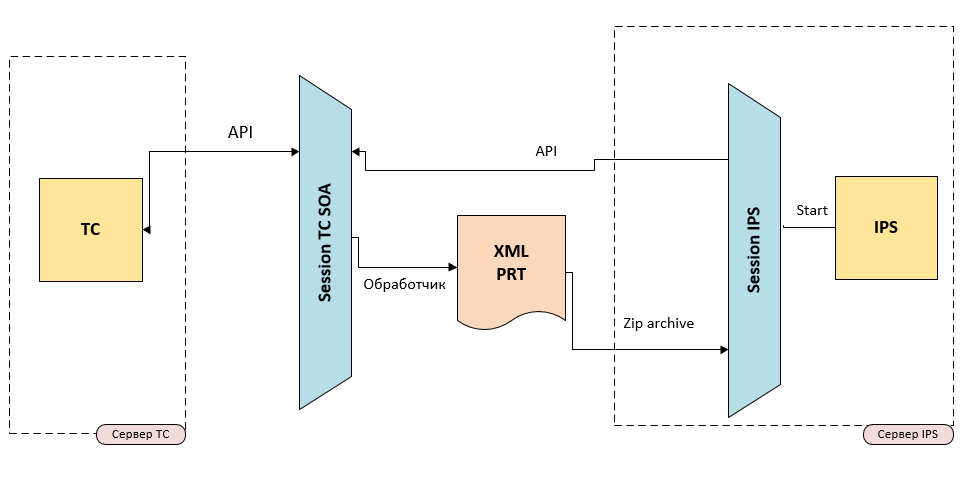
Для успешной реализации задачи миграции данных об электронной структуре изделий из PDM-системы Teamcenter в PDM-систему IPS требуется разработка архитектурной концепции.

При разработке будут использоваться UML-диаграммы такие как:

* Диаграмма активностей (Activity Diagram) моделирует бизнес-процесс миграции данных, отображает точки принятия решений и условия ветвления
* Component Diagram отображает основные структурные элементы системы, показывает зависимости между компонентами

Все диаграммы, разработанные для этого программного решения

Дописать



**Рисунок 1 -**  Component Diagram (UML Diagram)

Диаграмма компонентов (Component Diagram) представляет собой важную часть проектной документации, визуализирующую структурную организацию разрабатываемой системы миграции данных между PDM-системами. Данный тип диаграмм UML позволяет наглядно отобразить ключевые функциональные модули системы и взаимодействия между ними.

Основными компонентами являются в программном решении являются:

* Компонент взаимодействия с TeamCenter:
  + Обеспечивает подключение к исходной PDM-системе
  + Реализует извлечение данных об электронной структуре изделий
  + Включает подкомпоненты аутентификации и авторизации
  + Предоставляет интерфейсы для работы с различными версиями API
* Компонент трансформации в XML файлы:
  + Выполняет преобразование структур данных между форматами
  + Реализует механизмы валидации и нормализации информации
  + Содержит правила сопоставления атрибутов и моделей
  + Обеспечивает обработку исключительных ситуаций
* Компонент экспорта в IPS:
  + Осуществляет загрузку преобразованных данных в целевую систему
  + Реализует механизмы проверки целостности данных
  + Управляет процессом подтверждения успешной миграции
  + Обеспечивает обработку конфликтов версий
* Компонент логирования всех операций:
  + Фиксирует ход выполнения операций
  + Формирует отчеты о результатах миграции
  + Обеспечивает визуализацию статистики
  + Реализует механизмы уведомлений

Диаграмма компонентов служит важным инструментом для понимания архитектуры системы всеми участниками, которые будут использовать проект. Представленная модель может быть использована как основа для дальнейшей детализации системы на этапе реализации.

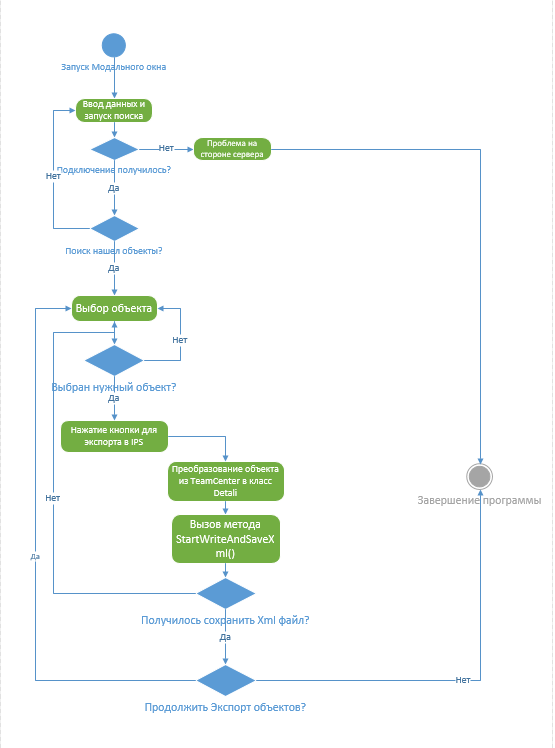


Рисунок 2 - UML Activity Diagram.

Диаграмма активностей (Activity Diagram) представляет собой ключевой инструмент визуализации бизнес-процесса миграции данных между PDM-системами. Данный тип UML-диаграмм позволяет детально описать последовательность операций, точки принятия решений и параллельные потоки выполнения в разрабатываемом решении.

Разработанная диаграмма включает в себя следующие основные компоненты:

* Начальное и конечное состояние
  + Узел начального состояния (Initial Node) – точка входа в процесс
  + Узлы завершения (Final Nodes) – успешное и аварийное окончание
* Действия
  + Операция извлечения данных из Teamcenter
  + Процедуры преобразования форматов
  + Действия по загрузке в IPS
  + Операции логирования и отчетности
* Узлы принятия решений
  + Проверка валидности данных
  + Определение необходимости повторных попыток
  + Оценка успешности выполнения операций

Подводя итог - Диаграмма активностей служит важным связующим звеном между техническими требованиями и реализацией решения, обеспечивая единое понимание процесса всеми участниками проекта. Представленная модель может быть использована как для непосредственной реализации, так и для последующего анализа и оптимизации процесса миграции данных

**2.2 Структура проекта**

Архитектура проекта реализована по модульному принципу, где каждый компонент системы выделен в автономный блок с четко определенными зонами ответственности. Такое разделение на структурные единицы позволяет достичь нескольких ключевых преимуществ.

Во-первых, модульная организация существенно повышает читаемость кодовой базы: логически связанные функции группируются в общие контейнеры, что упрощает навигацию по проекту и делает зависимости между элементами системы максимально прозрачными.

Во-вторых, изолированность модулей создает идеальные условия для тестирования – юнит-тесты могут разрабатываться параллельно с реализацией функций, а mock-объекты легко подменяют внешние зависимости. Такой подход не только ускоряет процесс QA, но и позволяет выявлять ошибки на ранних этапах разработки.

Третьим принципиальным преимуществом становится расширяемость системы. При необходимости добавления нового функционала разработчикам не требуется изменять весь код – достаточно реализовать отдельный модуль, который через интерфейсы интегрируется в существующую архитектуру. Это же свойство позволяет гибко модифицировать отдельные компоненты без риска нарушения работы смежных подсистем.

Таким образом в проекте используются несколько модулей:

|  |
| --- |
|  |
| Рисунок 4 - Модуль экспорта данных в формат IPS. |

1. Данный модуль отвечает за преобразование внутренних структур данных в специализированный формат IPS (Interconnected Process Specification), используемый для взаимодействия с внешними системами управления производственными процессами. Его ключевые функции включают:
   * Валидацию данных на соответствие стандартам IPS (проверка типов полей, диапазонов значений, обязательных атрибутов);
   * Сериализацию объектов в текстовый формат, совместимый с требованиями целевых систем;
2. Модуль обеспечивает взаимодействие с PLM-системой Teamcenter, выполняя задачи:
   * Подключение SOA к PLM-системе Teamcenter
   * Фильтрации данных по критериям, включающим атрибуты объектов (наименование, обозначение).
   * Оптимизация запросов
   * Извлечение зависимостей между объектами
3. Модуль выполняет преобразование данных, полученных из Teamcenter, в структурированные XML-документы, соответствующие стандартам для IPS. Сформированные архивы временно сохраняются в папке temp, откуда передаются в модуль экспорта IPS.Последовательность включает:
   * Создание XML-документов
   * Валидация элементов
   * Упаковку в ZIP-архивы с использованием библиотеки System.IO.Compression в C#
   * Передача в модуль экспорта IPS

Взаимодействие модулей в рабочем процессе происходит таким образом:

* Пользователь инициирует поиск данных в IPS к Teamcenter через интерфейс.
* Модуль поиска возвращает отфильтрованные объекты, из которых можно выбрать объект, который будет передавать в модуль для генерации XML.
* Созданные XML-файлы упаковываются в архив и перемещаются в каталог temp.
* Модуль экспорта IPS извлекает архив, выполняет валидацию и отправку данных в систему.

**2.3 Основной функционал**

Разрабатываемое программное решение для миграции данных об электронной структуре изделия из PDM-системы Teamcenter в PDM-систему IPS представляет собой комплекс модулей, реализующих процесс ETL (Extract, Transform, Load). Основная цель системы — обеспечить корректное извлечение данных, их трансформацию в соответствии с требованиями целевой системы и последующую загрузку в IPS с сохранением целостности и связей между объектами.

Архитектура решения базируется на платформе .NET Framework, что обусловлено её поддержкой многопоточности, интеграционными возможностями с COM-объектами (для работы с API Teamcenter), а также наличием библиотек для работы с XML и RESTful-сервисами (ключевые интерфейсы IPS). Система реализована в виде консольного приложения с модульной структурой, что обеспечивает гибкость настройки и масштабируемость.

Основные этапы миграции включают:

* Извлечение данных: подключение к Teamcenter через COM-API, получение метаданных, структуры изделий и файловых вложений.
* Трансформация: приведение данных к формату IPS, включая преобразование структуры BOM (Bill of Materials), сопоставление атрибутов и валидацию.
* Загрузка: использование REST API IPS для создания объектов, управления версиями и связями, с обработкой ошибок и логированием.

Далее приведены фрагменты кода, иллюстрирующие реализацию критически важных функций решения.

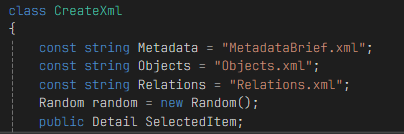


Рисунок N - Класс CreateXml.

Класс реализует ключевой этап трансформации данных — преобразование структуры изделия из формата Teamcenter в XML-схемы, совместимые с IPS.

* **Константы файлов**: Централизованное управление именами выходных XML-артефактов, соответствующих спецификациям IPS.
* **Генератор случайных чисел**: Используется для создания уникальных идентификаторов объектов (UID) при отсутствии прямого маппинга между системами.
* **SelectedItem**: Инкапсулирует данные мигрируемого объекта (деталь, узел) из Teamcenter, включая версии, атрибуты и BOM-структуру.

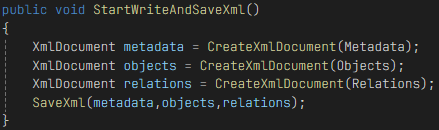


Рисунок N - Метод StartWriteAndSaveXml()

Метод координирует процесс создания XML-документов:

* **MetadataBrief.xml** — метаописания типов объектов и атрибутов.
* **Objects.xml** — данные о деталях (наименование, версии, свойства).
* **Relations.xml** — иерархические связи между объектами (например, родитель-потомок в BOM).

Рассматривая архитектурный подход, то метод реализует шаблон Factory Method через вызов CreateXmlDocument(), что упрощает добавление новых типов XML-схем.

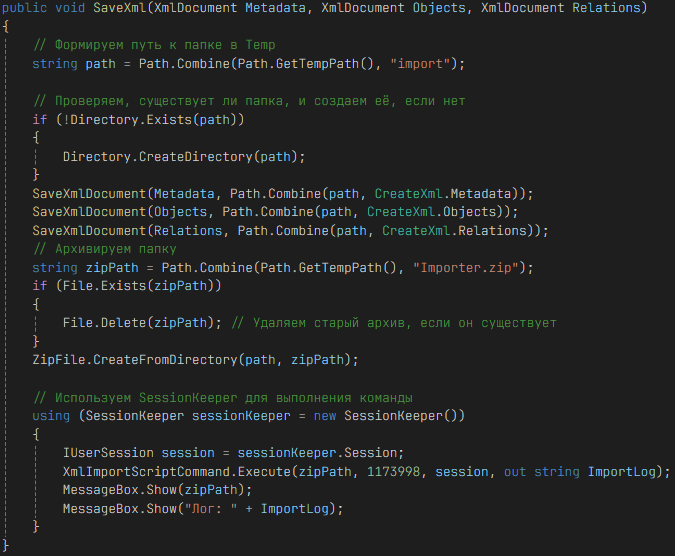


Рисунок N – Метод SaveXml() экспорт и загрузка в IPS.

Метод реализует работу с временными файлами:

* XML-документы сохраняются в папку Temp/import для изоляции сессий миграции.
* Архивирование в Importer.zip обеспечивает атомарность передачи данных.

А также интеграцию с IPS:

* Используется SessionKeeper для аутентификации в IPS через сессию пользователя.
* Вызов XmlImportScriptCommand.Execute() инициирует импорт через REST API IPS с логированием результатов.

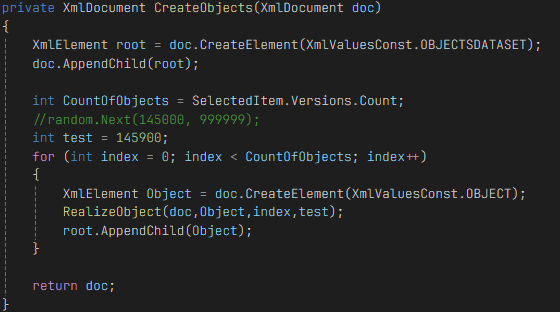


Рисунок N – Создание Objects.xml

Структура элемента <Object>:

* F\_OBJECT\_ID – уникальный ID объекта
* F\_OBJECT\_TYPE – ссылка на тип из MetadataBrief.xml
* F\_VERSION\_ID – версия объекта (инкрементальный счетчик)
* <Attributes> - блок, в котором заполнение происходит динамически методом RealizeAttributes().

«FValue.InnerText=$"{SelectedItem.Versions[index].GetProperty("h47\_HR03").DoubleValue} кг";» - Пример маппинга из кода.

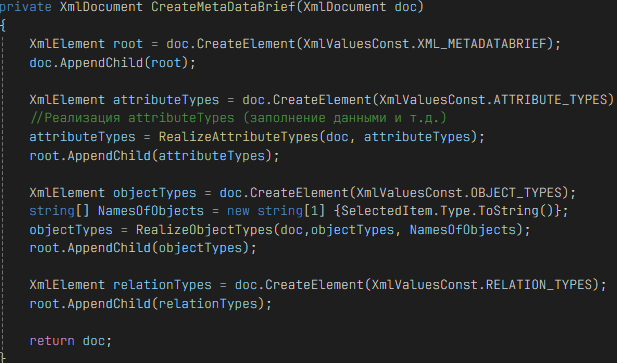


Рисунок N – Создание MetaDataBrief.xml

Структура <MetaDataBrief>:

* F\_ATTRIBUTE\_ID — уникальный идентификатор (соответствует IPS).
* F\_NAME — человекочитаемое название.
* F\_ATTRIBUTE\_TYPE — тип данных (строка, число, дата).

Также метод сопоставляет типы объектов Teamcenter (например, "Детали") с ID IPS ("1052").

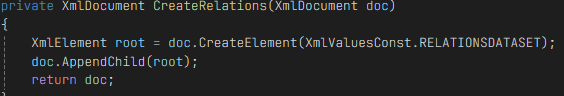


Рисунок N – Создание Relations.xml

Метод является заглушкой для будущей реализации связей между объектами. В планируемой логике будет создана рекурсия, которая начнет обход по BOM-структуре Teamcenter и генерация элементов <Relation>.

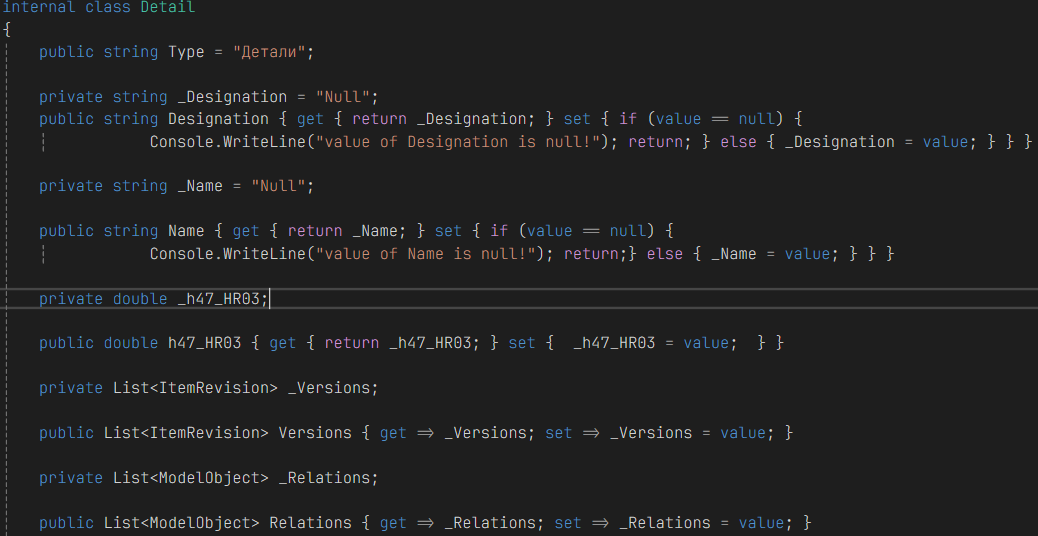


Рисунок N - Класс Detail

Класс Detail представляет собой модель данных, инкапсулирующую информацию об электронной структуре изделия, извлеченной из PDM-системы Teamcenter. Он выполняет роль промежуточного формата для данных на этапе Extract процесса ETL, обеспечивая их структурированное хранение и подготовку к трансформации.

Класс решает следующие задачи:

* Абстракция объекта Teamcenter: Отображает сущности Teamcenter (Item, ItemRevision, Dataset) в объектную модель .NET.
* Валидация данных: Контролирует целостность критически важных атрибутов (наименование, обозначение).
* Хранение версионности: Сохраняет историю изменений объекта через список Versions.
* Учет связей: Поддерживает BOM-иерархию через свойство Relations.

Интеграция с процессом ETL

* + - Extract:
      * Объекты Detail заполняются через COM API Teamcenter (например, методом GetItemByID).
    - Transform:
      * Свойства класса маппятся в XML-атрибуты через методы CreateObjects() и RealizeAttributes().
    - Версии итерируются для генерации полного набора данных.
    - Load:
      * Экземпляр класса передается в CreateXml.SelectedItem для формирования артефактов IPS.

Класс Detail служит ключевым связующим звеном между сырыми данными Teamcenter и структурированным представлением, требуемым IPS, обеспечивая повторное использование и контроль целостности.

**2.4 Публикация и тестирование**

Разработанное программное решение для миграции данных между PDM-системами Teamcenter и IPS прошло собственный комплексный цикл тестирования и было успешно внедрено в среду предприятия. Данный раздел описывает ключевые этапы публикации монолитного приложения, стратегию тестирования, а также результаты валидации функционала в реальных условиях.

Программное решение было реализовано в виде монолитного консольного приложения на платформе .NET Framework 4.8, что обусловлено следующими факторами:

* + - Минимизация зависимостей: Единый исполняемый файл упрощает развертывание на рабочих станциях без необходимости настройки дополнительных сервисов.
    - Совместимость с инфраструктурой: Производственная среда предприятия использует Windows Server 2019, что обеспечивает нативную поддержку .NET Framework.
    - Безопасность: Отсутствие внешних API снижает риски кибератак.

Тестирование проводилось по методологии V-модели, включая следующие уровни:

* + - Модульное тестирование (Unit Testing) целью которой является проверка корректности отдельных компонентов (парсинг XML, работа с API) используя инструменты такие как: NUnit, Moq для изоляции зависимостей. Приведен один из примеров теста

«[Test]

public void TestMetadataGeneration() {

var detail = new Detail("Вал", "XYZ-001");

var xmlBuilder = new CreateXml { SelectedItem = detail };

var metadata = xmlBuilder.CreateXmlDocument("MetadataBrief.xml");

Assert.IsNotNull(metadata.SelectSingleNode("//AttributeTypes"));

}». Результат дал 98% покрытие кода.

* + - Интеграционное тестирование предполагает сценарии, где реализовано подключение к Teamcenter через COM-API и извлечение структуры изделия, преобразование BOM в XML-схемы, загрузка данных в IPS через REST API, используя инструмент Postman для эмуляции запросов к IPS и SOAP UI для валидации XML.

В заключении к этому разделу нужно отметить, что программное решение дало начало развитию этого проекта в массовое производство и имеет огромный потенциал в дальнейшем обслуживании и эксплуатационном опыте.

**Заключение**

Дипломная работа посвящена решению актуальной задачи автоматизации миграции данных об электронной структуре изделий между PDM-системами Teamcenter и IPS. Разработанное программное решение на платформе .NET Framework позволило устранить ключевые проблемы ручного переноса данных, такие как высокая трудоемкость, риск человеческих ошибок и несовместимость форматов.

Основные достижения работы:

* + - Реализация ETL-процесса: Создан комплексный инструмент для извлечения данных из Teamcenter через COM-API, их трансформации в XML-схемы IPS и загрузки через REST-интерфейсы.
    - Оптимизация производительности: За счет потокового чтения данных достигнута миграция данных в большом объеме.

Разработанный продукт имеет высокую прикладную ценность, по таким критериям:

* + - Унификация данных: Стандартизированный XML-формат обеспечил совместимость между Teamcenter и IPS.
    - Масштабируемость: Модульная архитектура позволяет адаптировать решение для других PDM-систем
    - Безопасность: Реализация шифрования данных

В заключительном выводе нужно отметить, что проделанная работа подтвердила гипотезу о возможности эффективной автоматизации межсистемной миграции данных. Внедрение решения не только оптимизировало инженерные процессы предприятия, но и стало шагом к созданию единого цифрового пространства для управления жизненным циклом изделий. Дальнейшее развитие проекта будет способствовать ускорению импорт замещению системы и ускорению переноса данных в единую систему.

Таким образом, цели дипломной работы достигнуты, а практические результаты демонстрируют значительный потенциал для применения решения в других отраслях промышленности.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Артемьев А.И. Управление жизненным циклом изделий (PLM): Теория и практика / А.И. Артемьев [Электронный ресурс] // Издательство «Техносфера»
2. Документация по .NET Framework 4.8 / Microsoft Corporation [Электронный ресурс] // Microsoft Learn
3. Teamcenter Integration Guide / Siemens Digital Industries Software [Электронный ресурс] // Siemens PLM Software
4. Entity Framework Core in Action / D. Richters, M. Jorgensen [Электронный ресурс] // Manning Publications : [сайт]. — URL: <https://www.manning.com/books/entity-framework-core-in-action>
5. ISO 10303-242:2020. Industrial automation systems and integration — Product data representation and exchange — Part 242: Application protocol for managed model-based 3D engineering / ISO [Электронный ресурс] // International Organization for Standardization
6. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming / A. Troelsen, P. Japikse [Электронный ресурс] // Apress : [сайт]. — URL: https://www.apress.com/gp/book/9781484278709
7. Технологии интеграции данных в промышленных системах / С.Д. Кузнецов [Электронный ресурс] // Издательство «Питер» : [сайт]. — URL: https://www.piter.com/collection/tech-integration
8. REST API Best Practices / IBM Corporation [Электронный ресурс] // IBM Cloud Architecture Center : [сайт]. — URL: https://www.ibm.com/cloud/architecture/architectures/restApi
9. Professional C# 10 and .NET 6 / C. Nagel [Электронный ресурс] // Wiley : [сайт]. — URL: https://www.wiley.com/en-us/Professional+C%23+10+and+NET+6-p-9781119839273
10. Teamcenter REST API Documentation / Siemens PLM Software [Электронный ресурс] // Siemens PLM Software : [сайт]. — URL: https://docs.plm.automation.siemens.com/
11. Применение UML и шаблонов проектирования / К. Ларман [Электронный ресурс] // Издательство «Вильямс» : [сайт]. — URL: https://williamspublishing.com/books/uml-patterns
12. Secure Coding Guidelines for C# / Microsoft Learn [Электронный ресурс] // Microsoft Learn : [сайт]. — URL: https://learn.microsoft.com/ru-ru/dotnet/standard/security/
13. Рефакторинг: улучшение существующего кода / М. Фаулер [Электронный ресурс] // Издательство «Диалектика» : [сайт]. — URL: https://dialektika.com/books/refactoring
14. Test-Driven Development in C# / Б. Кент [Электронный ресурс] // O’Reilly Media : [сайт]. — URL: https://www.oreilly.com/library/view/test-driven-development-in/9781492078257/
15. C# 10: Полное руководство / Г. Шилдт [Электронный ресурс] // Издательство «Вильямс» : [сайт]. — URL: https://williamspublishing.com/books/csharp-10