

Научная статья



УДК 004.9

<https://doi.org/10.31854/1813-324X-2025-11-3-26-36>

EDN:DVEOYM

## Разработка системы AMS для университета: потребности в системе управленческого учета

- ▣ Александр Борисович Гольдштейн<sup>1,2</sup>, goldstein@sut.ru
- ▣ Сергей Викторович Кисляков<sup>1,2</sup> ✉, kislyakov@sut.ru
- ▣ Артём Алексеевич Кузнецов<sup>3</sup>, avoit5@yandex.ru
- ▣ Егор Андреевич Лочкарев<sup>4</sup>, lochkarev.egor00@mail.ru
- ▣ Илья Андреевич Рыбаков<sup>5</sup>, ifisher37bonch@gmail.com
- ▣ Даниил Игоревич Сухомлинов<sup>4</sup>, d.sukhomlinov.spb@gmail.com

<sup>1</sup>Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация

<sup>2</sup>ООО «НТЦ АРГУС», Санкт-Петербург, 197198, Российская Федерация

<sup>3</sup>АО «Селектел», Санкт-Петербург, 196006, Российская Федерация

<sup>4</sup>ООО «placeholder», Санкт-Петербург, 197191, Российская Федерация

<sup>5</sup>ООО «НТЦ ПРОТЕЙ», Санкт-Петербург, 194044, Российская Федерация

### Аннотация

**Актуальность.** Сегодня университеты – это крупные организации, обладающие большим количеством активов, начиная от недвижимости и заканчивая оборудованием и расходными материалами. Это оборудование имеет свой жизненный цикл, а университетам необходимо поддерживать не только работоспособность оборудования, но и осуществлять функции классической технической поддержки пользователей этого оборудования. Они сталкиваются с проблемами управления активами из-за устаревших бумажных методов учета. Рост объемов активов приводит к сложностям управления их жизненным циклом.

**Цель исследования** заключается в повышении эффективности управления активами университета за счет разработки системы AMS (аббр. от англ. Asset Management System) на основе открытой цифровой архитектуры (ODA, аббр. от англ. Open Digital Architecture), устраняющей пробелы в автоматизации процессов учета и анализа. Для реализации цели исследования использовался объектно-ориентированный анализ (диаграммы прецедентов и последовательности), концептуальное моделирование, а также проектирование с применением микросервисной архитектуры и стандартов ODA.

**Решение.** Разработаны функциональные требования (например, автоматизация списания оборудования, генерация инвентарных номеров) и нефункциональные (производительность, безопасность), а также архитектура системы на основе компонентов ODA.

**Новизна.** ODA пришла на смену давно существующим фреймворкам, таким как NGOSS и Framworx (TM Forum). Предложен подход к построению системы AMS на компонентах ODA. **Теоретическая значимость** определяется разработкой модели системы на новейшем фреймворке ODA, а **практическая значимость** – в непосредственно прикладном смысле данной задачи.

**Ключевые слова:** система управления активами (AMS), микросервисы, открытая цифровая архитектура (ODA), автоматизация процессов, интеграция систем, Asset Management

**Ссылка для цитирования:** Гольдштейн А.Б., Кисляков С.В., Кузнецов А.А., Лочкарев Е.А., Рыбаков И.А., Сухомлинов Д.И. Разработка системы AMS для университета: потребности в системе управленческого учета // Труды учебных заведений связи. 2025. Т. 11. № 3. С. 26–36. DOI:10.31854/1813-324X-2025-11-3-26-36. EDN:DVEOYM

Original research  
<https://doi.org/10.31854/1813-324X-2025-11-3-26-36>  
EDN:DVEOYM

# Development of Management Accounting System Model for Universities Based on Open Digital Architecture

 **Alexandr B. Goldstein**<sup>1,2</sup>, goldstein@sut.ru  
 **Sergey V. Kislyakov**<sup>1,2</sup> , kislyakov@sut.ru  
 **Artem A. Kuznetsov**<sup>3</sup>, avoit5@yandex.ru  
 **Egor A. Lochkarev**<sup>4</sup>, lochkarev.egor00@mail.ru  
 **Ilya A. Rybakov**<sup>5</sup>, ifisher37bonch@gmail.com  
 **Daniil I. Sukhomlinov**<sup>4</sup>, d.sukhomlinov.spb@gmail.com

<sup>1</sup>The Bonch-Bruевич Saint Petersburg State University of Telecommunications,  
St. Petersburg, 193232, Russian Federation

<sup>2</sup>LLC "ARGUS",  
St. Petersburg, 197198, Russian Federation

<sup>3</sup>CJSC «Selectel»,  
St. Petersburg, 196006, Russian Federation

<sup>4</sup>LLC «placeholder»,  
St. Petersburg, 197191, Russian Federation

<sup>5</sup>LLC «RTC PROTEL»,  
St. Petersburg, 194044, Russian Federation

## Annotation

**Relevance.** Today, universities are equipped with a large number of modern telecommunications and IT equipment, and other material assets. The university needs to manage the life cycle of all material assets. **The purpose of the research** is to improve the efficiency of university asset management by developing an AMS (Asset Management System) based on an open digital architecture (ODA), which eliminates gaps in the automation of accounting and analysis processes. The growth of data volumes and the need for integration with IT infrastructure underscore the relevance of developing flexible digital solutions.

**Research methods:** Object-oriented analysis, conceptual modeling, and design based on microservices architecture and ODA standards were applied.

**Results.** An AMS system was developed, including functional modules for automating equipment write-offs, generating inventory numbers, and analytics. The system integrates via API using ODA-components (TMFC014, TMFC039, etc.), ensuring flexibility and scalability.

**For the first time**, an approach to decomposing AMS into ODA-components is proposed, enhancing resource management efficiency and process automation. The work demonstrates the dependence of system flexibility on the application of ODA and creates a foundation for further scaling.

**The theoretical significance** is determined by the development of a system model on the latest ODA framework, and **the practical significance** is that the proposed result can be practically used to develop specific AMS solutions.

**Keywords:** asset management system (AMS), microservices, open digital architecture (ODA), process automation, system integration, resource management, data analytics, data security

**For citation:** Goldstein A.B., Kislyakov S.V., Kuznetsov A.A., Lochkarev E.A., Rybakov I.A., Sukhomlinov D.I. Development of Management Accounting System Model for Universities Based on Open Digital Architecture. *Proceedings of Telecommunication Universities*. 2025;11(3):26–36. (in Russ.) DOI:10.31854/1813-324X-2025-11-3-26-36. EDN:DVEOYM

### Цифровизация процессов: преимущества перед бумажным учетом

Современные университеты ежедневно сталкиваются с огромным объемом данных: это информация о сотрудниках, образовательных программах, учебных помещениях, техническом оборудовании, финансовых ресурсах и партнерских проектах. Управление всеми этими данными на бумаге или через разрозненные таблицы создает множество проблем:

- неэффективность: бумажные документы требуют много времени на обработку, поиск и внесение изменений; процессы согласования и утверждения становятся долгими, особенно, если участвует множество подразделений;
- ошибки и потери данных: ручной ввод информации увеличивает вероятность ошибок; бумажные документы подвержены рискам физической утраты (например, пожар, затопление) и износу;
- ограниченная доступность: доступ к данным ограничен только физическим местонахождением документов; трудности с предоставлением информации заинтересованным сторонам в реальном времени;
- отсутствие аналитики: отдельные электронные файлы в табличном формате или разрозненные (несвязанные) программные системы не позволяют эффективно собирать, агрегировать и анализировать данные; отсутствие аналитики осложняет принятие обоснованных решений.

Цифровая система управленческого учета решает эти проблемы, предоставляя единое пространство для хранения, анализа и управления всеми ресурсами университета.

К основным преимуществам цифровизации можно отнести:

- ускорение процессов: быстрый доступ к данным и автоматизация рутинных операций; мгновенное обновление информации;
- снижение рисков: цифровые данные защищены от потери (например, через резервное копирование); эффективные системы безопасности предотвращают несанкционированный доступ;
- прозрачность и контроль: все действия фиксируются в системе, что облегчает аудит; легче видеть полную картину текущего состояния ресурсов;
- поддержка аналитики и прогнозирования: возможности анализа больших данных позволяют оценивать эффективность и планировать развитие.

Аналогичные преимущества цифровизации демонстрируются в других образовательных системах, например, в веб-системе E-Report, которая упрощает обработку оценок и предоставляет доступ к данным в реальном времени [1].

Цифровая система управленческого учета представляет интерес для администрации вуза, так как

она открывает перед руководством университета новые горизонты управления, выгоды от которых указаны на рисунке 1.

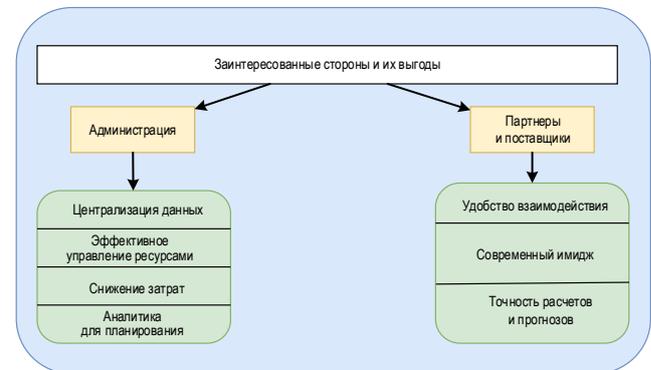


Рис. 1. Анализ востребованности продукта

Fig. 1. Analysis of Product Demand

Централизации данных свойственно хранение всех типов информации в одном месте – от информации о финансах до учета оборудования. В случае *эффективного управления ресурсами* видна текущая нагрузка и доступность аудиторий, оборудования, преподавателей. Еще одна выгода от цифровизации – *снижение затрат* (оптимизация процессов снижает административные издержки). На основе текущих данных можно прогнозировать потребности (*аналитика для планирования*).

Кроме того, для партнеров и поставщиков цифровизация делает университет более привлекательным и надежным, особенно когда реализована интеграция с другими ИТ-системами поддержки процессов и аналитики. В этой связи можно выделить ряд преимуществ: *удобство взаимодействия* (автоматизированные процессы работы с документами и договорами); *современный имидж* (цифровая система демонстрирует технологическую зрелость университета, что повышает его репутацию); *точность расчетов и прогнозов* (партнеры видят, что университет управляет своими ресурсами профессионально).

Таким образом, цифровая система становится мостом между университетом и внешними партнерами, укрепляя доверие и открывая новые возможности для сотрудничества.

### Интеграция системы управленческого учета в ИТ-инфраструктуру университета

Микросервисы – это архитектурный подход, предполагающий, что система состоит из небольших независимых сервисов, каждый из которых отвечает за выполнение определенных функций. Такой подход обеспечивает гибкость, масштабируемость и легкость интеграции с существующими системами [2]. Основными преимуществами микросервисов в разработке приложений являются гибкость интеграции, масштабируемость, минимизация рисков [3].

**Гибкость интеграции.** Микросервисы позволяют подключить новые модули системы управленческого учета (AMS, аббр. от англ. Asset Management System) к существующим решениям университета, не нарушая их работы. Например, можно интегрировать управление ресурсами с системой планирования ресурсов (ERP, аббр. от англ. Enterprise Resource Planning), системой технической поддержки или системой управления расписаниями.

**Масштабируемость.** При увеличении нагрузки (например, во время вступительной кампании)

можно масштабировать только те сервисы, которые нуждаются в повышении производительности, без необходимости расширения всей системы.

**Минимизация рисков.** В случае сбоя одного микросервиса остальные продолжают работать, что обеспечивает стабильность всей системы.

Интеграция AMS в существующую инфраструктуру университета с использованием микросервисов и компонентов ODA (аббр. от англ. Open Digital Architecture – открытой цифровой архитектуры) включает 6 этапов. Основные этапы интеграции системы отражены на схеме (рисунок 2).



Рис. 2. Этапы интеграции AMS

Fig. 2. The Main Steps of the AMS Integration

**Внедрение AMS в IT-ландшафт университета**

Функциональная архитектура системы AMS строится на основе ключевых блоков, необходимых для эффективного управления:

- 1) управление ресурсами:
  - учет финансовых, материальных и человеческих ресурсов;
  - контроль состояния и доступности оборудования, аудиторий, библиотечных фондов;
- 2) учет использования активов:
  - мониторинг использования помещений, оборудования и других университетских ресурсов;
  - учет технического состояния и планирование обновлений или ремонта;
- 3) аналитика и отчетность:
  - сбор и обработка данных о работе университета;
  - подготовка отчетов для администрации, партнеров и государственных структур.

**Проектирование компонентов и разработка модели данных системы AMS**

Для реализации системы AMS используется современный стек технологий, обеспечивающий гибкость и масштабируемость. Компоненты разрабатываются с использованием Docker / Kubernetes в качестве платформы управления контейнерами, что позволяет обеспечить их независимость и легкую интеграцию. Основными инструментами разработки являются Helm (упрощение управления развертыванием компонентов и их конфигурацией), PostgreSQL (для хранения и управления структурированными данными, связанными с активами, пользователями и процессами) и Grafana (для визуализации данных и аналитики, предоставляемой системой). Выбор технологий для AMS основан на их гибкости и масштабируемости, однако в других исследованиях для аналогичных систем управления активами в образовательных

учреждениях применяются более простые стеки, такие как PHP и MySQL, что также обеспечивает достаточную функциональность (<https://www.tmforum.org/resources/standard/open-digital-architecture-toolkit>).

### Разработка моделей данных и интеграции

В ходе исследования было применено два основных подхода в разработке и интеграции: модель данных (SID, аббр. от англ. Shared Information and Data Model) и интеграция с существующими системами. Модель SID объединяет данные и информацию и применяется в тех случаях, когда необходимо иметь легко масштабируемое и легко интегрируемое решение с другими системами ИТ-ландшафта. Такой подход обеспечивает единообразие данных и поддерживает адаптивность системы к изменяющимся потребностям университета. Применение разработок некоммерческой международной компании TMForum для решения AMS позволит реализовать легкую интеграцию с CRM, ERP и другими системами университета благодаря использованию RESTful Open APIs.

### Основные этапы проектирования системы AMS

С учетом особенностей университетской деятельности можно выделить пять основных этапов проектирования системы.

#### Этап 1. Сбор и анализ требований:

- определение ключевых задач системы: управление активами, автоматизация процессов, улучшение взаимодействия с заинтересованными сторонами;
- оценка текущей ИТ-инфраструктуры университета и определение точек интеграции;
- уточнение требований к безопасности данных и нормативным стандартам (например, защита персональных данных в соответствии с локальными и международными законами).

#### Этап 2. Проектирование архитектуры:

- разработка архитектуры на основе микросервисного подхода, обеспечивающего модульность и гибкость системы;
- создание карты компонентов системы с учетом ODA-принципов (блок управления активами, системы аналитики и мониторинга; интерфейсные модули для взаимодействия с пользователями) [4];
- определение взаимодействия между компонентами через стандартизированные Open APIs.

#### Этап 3. Разработка прототипа:

- реализация пилотного проекта на Docker/Kubernetes для проверки работоспособности ключевых функций;
- тестирование базовой функциональности (учет ресурсов, аналитика и интеграция с существующими системами).

#### Этап 4. Внедрение и интеграция:

- разработка окончательной версии системы с постепенным подключением всех функциональных модулей;
- интеграция AMS с системами университета (LMS, ERP, CRM) через единый API-шлюз;
- постепенный запуск системы, начиная с базовых модулей, с минимальными рисками для рабочих процессов.

#### Этап 5. Обучение и поддержка:

- проведение тренингов для сотрудников и предоставление инструкций по работе с системой;
- организация службы технической поддержки для решения возникающих вопросов.

### Принципы безопасности

Основные принципы безопасности, используемые при проектировании системы AMS: *конфиденциальность* (применяются механизмы шифрования данных в состоянии покоя и при передаче (например, TLS/SSL для взаимодействия через API <https://www.tmforum.org/resources>), *целостность* (данные защищаются от несанкционированных изменений через механизмы контроля версий и автоматизированный мониторинг), *доступность* (реализуется резервное копирование и план восстановления в случае аварий, что минимизирует простой системы).

В целях обеспечения безопасности системы на всех этапах – от проектирования до внедрения и эксплуатации, применяются известные подходы:

- архитектура с нулевым доверием (*от англ. Zero Trust*): каждый запрос к системе проходит многофакторную аутентификацию и авторизацию, независимо от источника [5];
- роли и ограничения доступа: пользователи системы (администрация, партнеры) получают доступ только к данным и функциональности, которые необходимы для их задач;
- мониторинг и аудит: постоянный контроль активности пользователей и компонентов для выявления подозрительных действий;
- соответствие нормативным стандартам.

### Практические результаты

Для определения более детальных требований обратимся к диаграмме прецедентов (рисунок 3), которая поможет нам визуализировать, как пользователи (акторы) взаимодействуют с системой. Это упрощает восприятие архитектуры системы и функций, которые она предлагает. Диаграмма позволяет четко определить, кто будет взаимодействовать с системой (например, сотрудники ИТ-отдела, бухгалтерия, сотрудники отдела технической поддержки и т. д.). Это помогает понять, какие роли необходимы и какие функции нужно разработать для каждой из них [2].

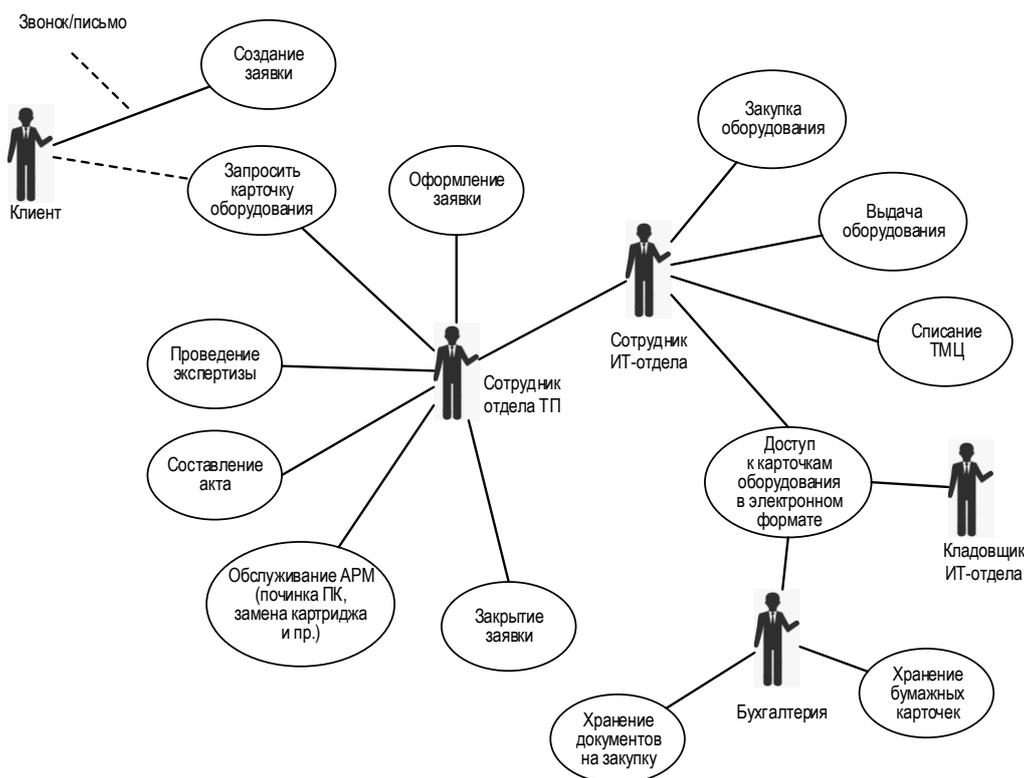


Рис. 3. Диаграмма прецедентов

Fig. 3. Diagram of Precedents

При введении системы AMS можно выделить следующих акторов и их взаимодействия (прецеденты): клиент (заказчик), сотрудник отдела технической поддержки, сотрудник ИТ-отдела, кладовщик ИТ-отдела и бухгалтерия. Указанные отделы взаимодействуют в рамках разных бизнес-процессов (см. рисунок 3).

В любом университете традиционно функционирует три взаимодействующих подразделения:

- ИТ-отдел, функции которого включают закупку оборудования и программного обеспечения, выдачу его заказчикам и учет;
- отдел эксплуатации (сервис технической поддержки), принимающий заявки на обслуживание – ремонт оборудования, замена картриджей и т. п., а также оказывающий информационную поддержку пользователям;
- бухгалтерия.

С помощью диаграмм прецедентов можно определить и задокументировать требования к системе. Каждое действие пользователей и соответствующие им прецеденты служат основой для определения функциональных требований [1].

Не менее важным инструментом в объектно-ориентированном анализе и проектировании, который используется для визуализации взаимодействий между объектами в определенной последовательности во времени, является диаграмма последовательности. Она показывает, как объекты

взаимодействуют между собой в течение определенного сценария, помогает понять, как объекты коммуницируют между собой для выполнения конкретной задачи [4] в рамках конкретных бизнес-процессов. Это особенно полезно для аналитиков и дизайнеров: можно увидеть, как разработанные компоненты будут работать вместе.

На рисунке 4 представлена диаграмма последовательности взаимодействия отделов по работе с заявкой. В первом варианте альтернативного события представлен случай, когда система AMS отсутствует. Необходимо совершать много дополнительных действий для получения карточки оборудования. В случае наличия системы AMS у всех отделов будет доступ к электронной карточке оборудования в любое время.

После описания диаграмм перейдем к описанию требований к системе AMS. Разработка этих требований опирается на лучшие практики управления активами, включая автоматизацию процессов, обеспечение безопасности и производительности, что соответствует современным стандартам систем управления активами [5, 6]. Можно выделить функциональные требования (задачи системы) и нефункциональные (характеристики работы).

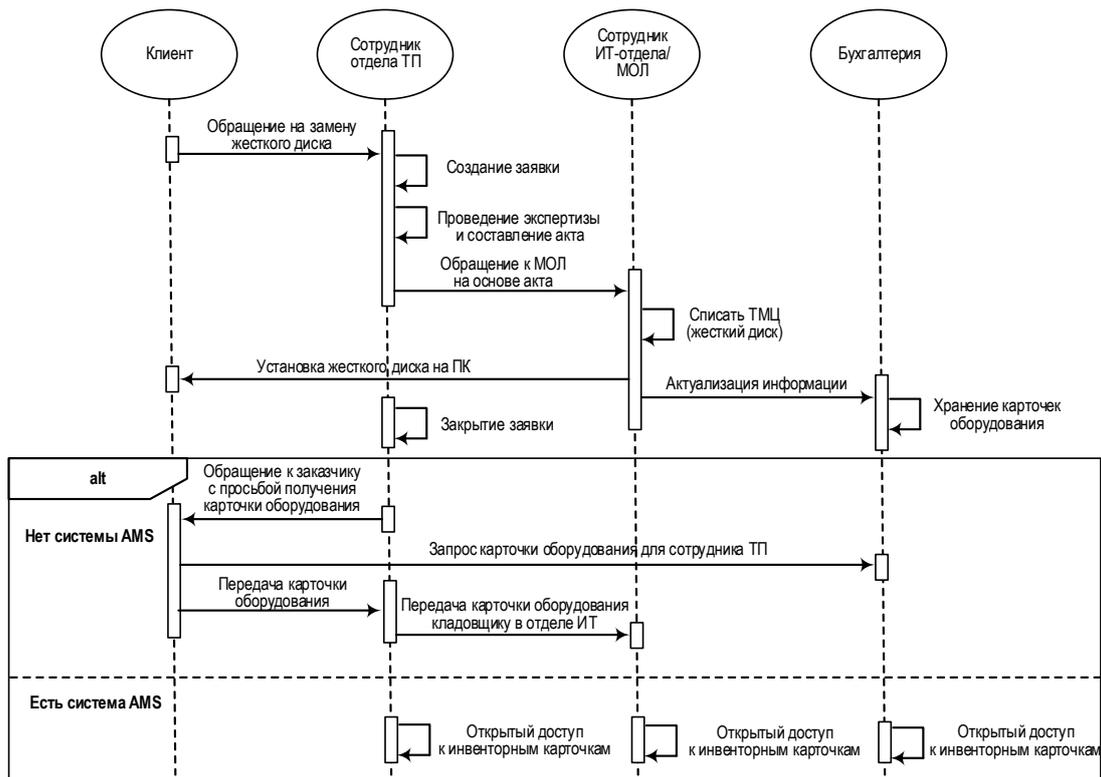


Рис. 4. Диаграмма взаимодействия отделов

Fig. 4. Departments Sequence Diagram

К функциональным требованиям можно отнести:

1) списание оборудования (автоматическое создание акта списания с подписями сторон и акта утилизации для снятия оборудования с баланса);

2) учет стоимости (добавление полей «Цена на входе» и «Остаточная стоимость» с сортировкой и списанием объектов ниже заданного порога);

3) генерация инвентарных номеров (автоматическая генерация уникальных номеров для поступающего товара с печатью для наклеивания);

4) печать ярлыков (автоматическая печать ярлыков с инвентарными номерами для инвентаризации);

5) справочник материально-ответственных лиц (учет ФИО, номера кабинета и телефона с возможностью поиска и редактирования);

6) загрузка гарантийных талонов (поддержка загрузки талонов на партию и отдельные экземпляры в систему);

7) работа нескольких пользователей (одновременная работа пользователей с уведомлениями о действиях других);

8) отслеживание конфликтов редактирования (контроль одновременного редактирования данных для сохранения их целостности);

9) описание помещения (формирование описания помещения на основе местоположения товарно-материальных ценностей);

10) генерация расписок для ремонта (автоматическое создание расписок для оформления товарно-материальных ценностей (ТМЦ) в ремонт.

Нефункциональные требования включают в себя:

1) производительность до 5 сек (обработка запросов – создание документов, сортировка, генерация номеров);

2) надежность (доступность системы – не менее 99,9 % с сохранением данных при сбоях);

3) безопасность (шифрование данных, доступ только авторизованным пользователям);

4) хранение (документы хранятся 3 года).

### Система AMS на основе открытой цифровой архитектуры

С появлением и развитием технологий, облачных вычислений и мобильных платформ возникла необходимость в создании более гибких решений, способных легко интегрироваться с различными сервисами и платформами. Переход на ODA поддерживается мировыми стандартами, разработанными TM Forum (<https://www.tmforum.org>). Традиционно, стандарты способствуют унификации и упрощению интеграции систем, повышению гибкости и масштабируемости инноваций в разработке программного обеспечения (<https://www.tmforum.org/resources/introductory-guide/oda-component-inventory-v16-0-0-ig1242>).

Процесс построения системы AMS на основе концепции ODA включает определение бизнес-требований, выбор группировок бизнес-функций из Функциональной архитектуры ODA, определения программных функций на основе структуры приложений ODA и, наконец, разбиение системы на ODA-компоненты на основе выбранных функций. Результатом будет система, состоящая из множества взаимосвязанных и функционирующих компонентов (таблица 1).

Помимо перечисленных в таблице 1 компонентов ODA, для системы AMS также были выделены все компоненты из блоков ODA Canvas Operators и Engagement Management. Это обусловлено тем, что

компоненты этих блоков отвечают за графический интерфейс (GUI, *аббр. от англ.* Graphical User Interface), управление потоками событий и организацию операционных процессов внутри среды выполнения компонентов ODA Canvas внутри ИТ-ландшафта. На рисунке 5 продемонстрировано взаимодействие компонентов ODA из блоков операционных процессов. TMFC039 Agreement Management не связан API, так как его координация и интеграция с остальными компонентами реализована через API TMF 701 Process Flow Management и TMF688 Event Management. TMFC039 необходим для реализации в система справочника, содержащего договоры с различными параметрами.

ТАБЛИЦА 1. Компоненты системы AMS

TABLE 1. AMS System Components

Название компонента	Краткое описание функциональности
Production	
TMFC014 Location Management	Отвечает за управление информацией о местоположениях, например адреса, принадлежность к различным зонам и т. д.
TMFC010 Resource Catalog Management	Организует сбор спецификаций ресурсов, определяющих требования к ним, и предоставляет функционал для отображения клиентского и технического представления ресурсов, а также управления ими на протяжении жизненного цикла
TMFC006 Service Catalog Management	Организует сбор спецификаций услуг, определяющих все требования к ним; Предоставляет функциональность для отображения клиентского и технического представления услуг, позволяя пользователям искать и выбирать необходимые услуги; также включает управление спецификациями, администрирование жизненного цикла услуг и упрощение доступа к ним
TMFC007 Service Order Management	Отвечает за доставку ресурсов клиентского сервиса (CFS) и предоставляет API <i>ServiceOrder</i> ; Организует процесс доставки, определяя возможные ресурсы и выбирая их из каталога, а также запрашивает обновление выбранных экземпляров ресурсов для доставки CFS
TMFC011 Resource Order Management	Отвечает за выполнение запросов на ресурсы в соответствии с их требованиями; Охватывает все этапы процесса заказа, включая получение и подтверждение заказов, управление доставкой с проверкой доступности ресурсов, отслеживание действий по заказу, сопоставление зависимостей ресурсов, обработку обновлений заказов и отчетность о состоянии жизненного цикла заказа
TMFC032 Supply Chain Management (без спец.)	Включает в себя задачи по планированию ресурсов и услуг, управлению информацией о них, управлению снабжением, запасами, производством, местоположением, транспортировкой, возвратами и логистикой
Intelligence Management	
TMFC038 Resource Performance Management	Обеспечивает доступность и готовность приложений, вычислительных и сетевых ресурсов; Включает в себя создание и управление экземплярами ресурсов, мониторинг и отчетность о возможностях и затратах; Основные обязанности этих процессов включают поддержку внедрения новой инфраструктуры, управление плановыми отключениями, анализ доступности и производительности ресурсов, а также выполнение мероприятий по техническому обслуживанию и ремонту; Кроме того, важными аспектами являются управление запасными частями, оценка угроз и рисков, а также мероприятия по снижению рисков и безопасной конфигурации
TMFC037 Service Performance Management	Отвечает за сбор, контроль, анализ и отчетность о производительности конечного сервиса; Включает в себя мониторинг в реальном времени для обеспечения корректной работы услуг, а также исторический анализ; Функции основаны на данных управления производительностью ресурсов и активных испытаниях производительности, что позволяет получить полное представление о качестве обслуживания; Компонент предоставляет ключевые входные данные для оценки качества сервиса
Party Management	
TMFC039 Agreement Management	Отвечает за создание, хранение, редактирование и отслеживание согласованных соглашений на протяжении всего жизненного цикла; Управляет предложениями, записями принятия и связанными с ними намерениями для установления юридически обязательных соглашений; Кроме того, компонент предоставляет рабочие процессы и шаблоны, которые упрощают сотрудничество и переговоры между сторонами, а также администрирует особенности перевода соглашений в контракты; Обеспечивает безопасное хранение, контроль версий, управление соответствием и уведомления о продлении соглашений

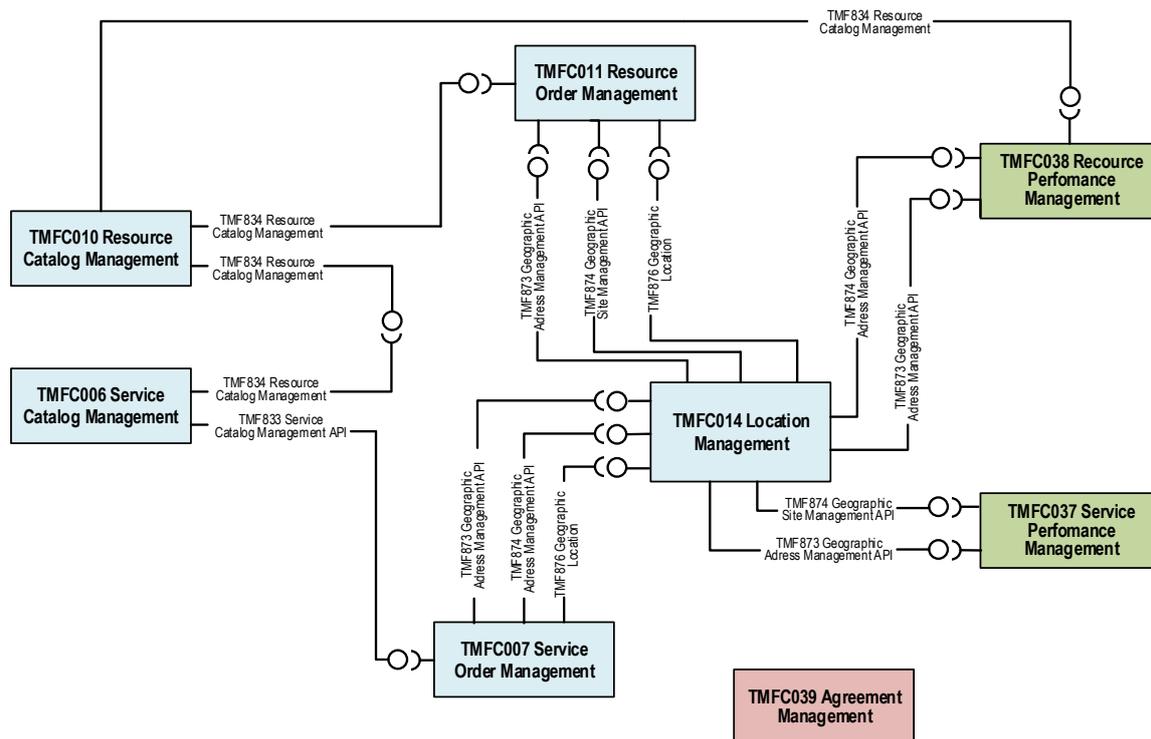


Рис. 5. Взаимодействие компонентов ODA операционных блоков через API

Fig. 5. ODA-Components Interaction through API

Таким образом, система AMS была спроектирована на основе множества компонентов ODA.

### Реализация и тестирование системы AMS

Система AMS была реализована на выделенном сервере с использованием контейнеризации через Docker, что обеспечило гибкость и масштабируемость. В качестве основного фреймворка применялся NocoDB – инструмент для создания баз данных с удобным интерфейсом, а для хранения данных использовалась система управления базами данных PostgreSQL, обеспечивающая надежность и производительность. Такой стек технологий позволил быстро развернуть систему и адаптировать ее под нужды университета.

Главная страница системы AMS, реализованная через интерфейс NocoDB (<https://nocodb.com/docs/self-hosting/installation/docker-install>), отображает доступные базы данных и проекты, предоставляя пользователю удобный доступ к управлению активами. Интерфейс минималистичен: слева расположена панель навигации с разделами *Bases* и *Projects*, а в нижней части – информация о текущем пользователе (например, `username@email.com`) и кнопка для перехода к облачной версии NocoDB. Главная страница служит отправной точкой для работы с системой, позволяя администраторам и сотрудникам быстро переходить к нужным модулям.

Управление пользователями – одна из ключевых функций AMS, обеспечивающая контроль доступа и распределение ролей. На рисунке 6 показан интерфейс управления пользователями, где отображаются их данные: имена (например, *Test\_Observer*, *Test\_Creator*), email-адреса, роли (Наблюдатель, Создатель, Владелец) и статус активности. Также указывается время последнего входа (например, «1 d ago» или «2 h ago»), что помогает отслеживать активность. Этот модуль позволяет администратору добавлять, редактировать или удалять пользователей, а также управлять их правами, что соответствует требованиям безопасности и многопользовательской работы, описанным ранее.

В качестве отправной точки для анализа, разработки и тестирования использовалась платформа «АРГУС AMS» (рисунок 7), позволяющая создавать гибкие проектные решения под самые разные архитектуры предприятий и их бизнес-процессов за счет сформированной на принципах low-code инфраструктуры и возможностей JMIX.

AMS-компонент включает 6 базовых сущностей (средний слой AMS-компонент). Выделение таких сущностей позволяет реализовывать различные сценарии управленческого учета, придавая этим понятиям различный бизнес-смысл. Эти объекты являются самостоятельными, их можно настраивать независимо.

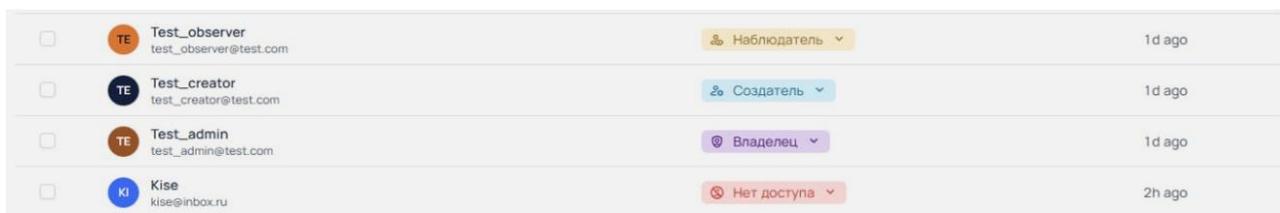


Рис. 6. Интерфейс управления пользователями

Fig. 6. User's Management Interface

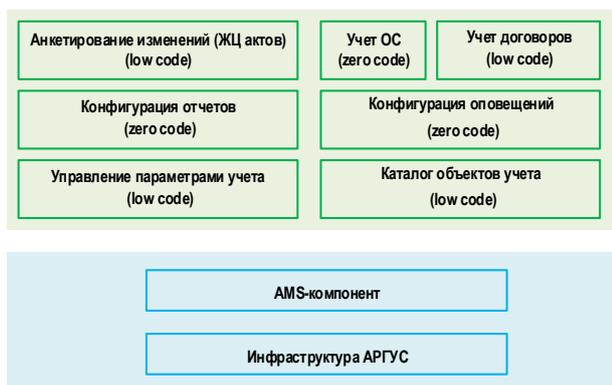


Рис. 7. Платформа «ARGUS AMS»

Fig. 7. «ARGUS AMS» Software Platform

Сущности AMS-компонента реализованы по технологии «low code», что не потребовало и не потребует писать большого количества строк кода для адаптации системы под конкретные задачи университета. Это позволит в дальнейшем не нести больших затрат на программирование и программистов. Необходимые доработки под конкретный проект и сценарии можно будет реализовывать по технологии «low code». Верхний слой платформы AMS содержит сущности, которые настраиваются и взаимодействуют в режиме «zero-code», когда не требуется знание языков программирования.

**Заключение**

В результате разработана модель системы управления активами (AMS) для университета, основанная на новейшем фреймворке – открытой цифровой архитектуры (ODA). Система решает ключевые проблемы традиционного бумажного учета, такие как неэффективность, высокий риск ошибок и отсутствие аналитических возможностей, обеспечивая автоматизацию процессов, централизованное хранение данных и интеграцию с существующей IT-инфраструктурой через стандартизированные API.

**Список источников**

1. Febriyanto E., Naufal R.S., Sulistiawati S. Planning of the Web-based E-Raport Assessment System // Aptisi Transactions on Technopreneurship. 2020. Vol. 2. Iss. 1. PP. 34–44. DOI:10.34306/att.v2i1.27
2. Кисляков С.В., Майоров В.В. Разработка бизнес-процесса подключения услуги на основе TM Forum Open Digital Architecture // XI Международная научно-техническая и научно-методическая конференция «Актуальные проблемы инфотелекоммуникаций в науке и образовании» (АПИНО 2022, Санкт-Петербург, Российская Федерация, 15–16 февраля 2022 г.). СПб.: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2022. Т. 1. С. 551–556. EDN:FFGRMD

Разработанные функциональные требования, включая автоматизацию списания оборудования, генерацию инвентарных номеров и управление справочниками, а также нефункциональные требования (производительность, безопасность, масштабируемость), позволили создать гибкое и устойчивое решение. Применение компонентов ODA, таких как TMFC014 Location Management и TMFC039 Agreement Management, обеспечило модульность системы и возможность ее дальнейшего масштабирования. Использование современных технологий (Docker/Kubernetes, PostgreSQL, Grafana) и подходов к безопасности (zero trust, шифрование данных) гарантирует надежность и защиту данных на всех этапах эксплуатации.

Новизна работы заключается в предложенном подходе к декомпозиции «монолитной» AMS на компоненты ODA с учетом специфики управления активами, что отличает решение от традиционных систем. Теоретическая значимость исследования состоит в установлении зависимости между гибкостью системы и применением ODA, а практическая – в предоставлении инструмента для автоматизации процессов, улучшения аналитики и поддержки принятия решений на основе данных.

Перспективы дальнейших исследований включают внедрение и тестирование прототипа системы в реальных условиях университета, а также расширение функциональности AMS за счет добавления модулей для прогнозирования потребностей и управления проектами с партнерами.

ODA позволяет разрабатывать системы, автоматизирующие любой бизнес, так как именно компонентный подход вкупе с открытыми или стандартными API (прикладными программными интерфейсами) дает максимальную гибкость в архитектуре и функциональности. Если за основу разработки взять платформу JMIX, то это позволит в дальнейшем не нести больших затрат на программирование и программистов. Необходимые доработки под конкретный проект и сценарии можно будет реализовывать по технологии «low code».

3. Iluore O.E., Onose A.M., Emeter M. Development of asset management model using real-time equipment monitoring (RTEM): case study of an industrial company // *Cogent Business & Management*. 2020. Vol. 7. Iss. 1. DOI:10.1080/23311975.2020.1763649

4. Гольдштейн А.Б., Кисляков С.В. Современные подходы к автоматизации бизнес-процессов операторов связи. СПб.: СПбГУТ, 2020. 84 с.

5. Гольдштейн А.Б., Кисляков С.В. Концепция открытой цифровой архитектуры: эволюция или революция? // *Вестник связи*. 2022. № 6. С. 21–25. EDN:GVAFPF

6. Гольдштейн А.Б., Кисляков С.В., Феноменов М.А. Открытая цифровая архитектура для разработки систем управления инфокоммуникациями. СПб.: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. 77 с. EDN:AOKPJR

## References

1. Febriyanto E., Naufal R.S., Sulistiawati S. Planning of the Web-based E-Raport Assessment System. *Aptisi Transactions on Technopreneurship*. 2020;2(1):34–44. DOI:10.34306/att.v2i1.27

2. Kislyakov S., Mayorov V. Development of a Business Process for Service Activation Based on TM Forum Open Digital Architecture. *Proceedings of the XIth International Conference on Infotelecommunications in Science and Education, 1–2 March 2017, St. Petersburg, Russian Federation, vol.1*. St. Petersburg: The Bonch-Bruевич Saint Petersburg State University of Telecommunications Publ.; 2022. p.551–556. (in Russ.) EDN:FFGRMD

3. Iluore O.E., Onose A.M., Emeter M. Development of asset management model using real-time equipment monitoring (RTEM): case study of an industrial company. *Cogent Business & Management*. 2020;7(1). DOI:10.1080/23311975.2020.1763649

4. Goldstein A.B., Kislyakov S.V. *Modern Approaches to Automation of Business Processes of Telecom Operators*. St. Petersburg: The Bonch-Bruевич Saint Petersburg State University of Telecommunications Publ.; 2020. 84 p. (in Russ.)

5. Goldstein A.B., Kislyakov S.V. The concept of open digital architecture: evolution or revolution? *Vestnik svyazi*. 2022;6: 21–25. (in Russ.) EDN:GVAFPF

6. Goldstein A.B., Kislyakov S.V., Fenomenov M.A. *Open Digital Architecture for the Development of Infocommunication Management Systems*. St. Petersburg: The Bonch-Bruевич Saint Petersburg State University of Telecommunications Publ.; 2024. 77 p. (in Russ.) EDN:AOKPJR

Статья поступила в редакцию 08.04.2025; одобрена после рецензирования 05.06.2025; принята к публикации 10.06.2025.

The article was submitted 08.04.2025; approved after reviewing 05.06.2025; accepted for publication 10.06.2025.

## Информация об авторах:

**ГОЛЬДШТЕЙН**  
**Александр Борисович**

доктор технических наук, профессор, профессор кафедры инфокоммуникационных систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича, директор ООО «НТЦ АРГУС»

 <https://orcid.org/0000-0002-4136-4703>

**КИСЛЯКОВ**  
**Сергей Викторович**

кандидат технических наук, доцент кафедры инфокоммуникационных систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича, аналитик ООО «НТЦ АРГУС»

 <https://orcid.org/0000-0001-8842-7903>

**КУЗНЕЦОВ**  
**Артём Алексеевич**

инженер АО «Селектел»

 <https://orcid.org/0009-0001-6624-2868>

**ЛОЧКАРЕВ**  
**Егор Андреевич**

инженер ООО «placeholder»

 <https://orcid.org/0009-0009-1418-0018>

**РЫБАКОВ**  
**Илья Андреевич**

инженер ООО «НТЦ ПРОТЕЙ»

 <https://orcid.org/0009-0006-4173-087X>

**СУХОМЛИНОВ**  
**Даниил Игоревич**

инженер ООО «placeholder»

 <https://orcid.org/0009-0008-7304-270X>

Авторы сообщают об отсутствии конфликтов интересов.

The authors declare no conflicts of interests.