

Содержание

Введение.....	3
1 Современное состояние проблемы использования сервис-ориентированной системы.....	6
1.1 Анализ литературы по методам и средствам моделирования систем основанных на СОА.....	6
1.2 Постановка проблемы, цели и задач для дальнейшего исследования.....	15
2 Методология использования сервис-ориентированной архитектуры.....	17
2.1 Существующие методы сервис-ориентированной информационной системы	17
2.2 Сравнение работы организации без использования СОА и с внедренной системой	24
2.3 Характеристика и моделирование существующих архитектур	26
2.4 Описание и сравнение сервис-ориентированной архитектуры и микросервисов	33
2.5 Методы внедрения сервис-ориентированной архитектуры	40
3 Разработка улучшенной СОА системы и ее апробация	47
3.1 Создание улучшенной модели системы основанной на СОА.....	47
3.2 Сравнительный анализ улучшенной модели СОА с существующими аналогами	56
3.3 Апробация улучшенной СОА системы.....	59
Заключение	69
Список используемых источников.....	71

Введение

В современном мире факторами успешной работы организаций и предприятий является эффективное использование тех или иных существующих на данный момент технологий и систем.

В наше время экономическим условиям свойственно меняться очень быстро, поэтому IT-подразделениям следует своевременно ставить перед собой такие задачи, как существенное изменение или дальнейшее развитие работоспособности применяемых в тех или иных организациях или предприятиях информационных систем.

Также следует ввести в дальнейшее использование новые технологии, их решения, внедрение которых должно осуществляться с учетом уже существующих механизмов инфраструктур.

IT-архитекторы исследуют сервис-ориентированную архитектуру (SOA) и дают ему такую характеристику – это такой подход, который объединяет бизнес-процессы и механизмы, тесно связанные друг с другом и их можно использовать многократно в течение длительного времени, а также изменять или объединять в соответствии с изменениями в конкретной организации.

Цель представленной работы провести анализ методов и средств моделирования систем на базе SOA, сравнить способы их внедрения и предложить улучшенную версию.

Для достижения поставленной цели были сформулированы следующие **задачи**:

- ознакомиться с научными работами на данную тематику или близкую к ней;
- ознакомиться с существующими методами сервис-ориентированной информационной системы. Изучить преимущества и недостатки рассмотренных методов;
- рассмотреть способы внедрения SOA системы;

- разработать улучшенную модель СОА системы, провести апробацию;
- сделать выводы по полученным данным.

Гипотеза исследования магистерской диссертации:

Проектирование программного обеспечения на основе СОА с возможностью интеграции дополнительных модулей позволит упростить процесс внедрения и сопровождения данного ПО.

Объектом исследования магистерской диссертации являются методы и средства моделирования систем с сервисно-ориентированной архитектурой.

Предметом исследования диссертации является процесс использования СОА системы.

Методологической основой исследования являются работы российских и зарубежных специалистов, посвященные проблеме изучения возможности использования системы основанной на сервис-ориентированной архитектуре.

Научная новизна заключается в разработке улучшенной модели сервис-ориентированной архитектуры.

Теоретической основой диссертационного исследования являются научные труды российских и зарубежных специалистов, проводивших исследование и внедрение данной системы.

Теоретическая значимость состоит в уточнении и развитии теоретических основ внедрения системы в организации.

Практическая значимость заключается в том, что применение сервис-ориентированной архитектуры позволит снизить расходы на внедрение и сопровождение, и увеличить производительность системы.

Публикации по теме исследования. Основные результаты теоретической части исследования изложены в статьях:

1. Талицких В.А. Анализ методов внедрения сервис-ориентированных информационных систем. VII Международная научно-практическая конференция (школы-семинара) молодых ученых «Прикладная математика и

информатика: современные исследования в области естественных и технических наук» – 2021 – с. 64 – 68.

Положения, выносимые на защиту:

- улучшенная модель сервис-ориентированной архитектуры;
- методы внедрения SOA системы и какой наилучший;
- результаты тестирования работы системы с SOA;
- апробация системы, разработанной на основе сервис-ориентированной архитектуры.

Диссертация состоит из введения, трех частей, заключения и списка литературы.

Первая часть содержит описание исследований современного уровня проработанности научной проблемы на основе изучения и анализа, как зарубежных, так и СНГ источников и материалов, анализ различных точек зрения.

Вторая включает в себя сравнительный анализ существующих методов SOA. Также содержит в себе варианты моделирования сервис-ориентированной системы и существующие методы внедрения. После, приводится сравнение популярного конкурента с объектом исследования и выбирается наилучшее решение для данной системы.

Третья часть включает в себя тестирование работы SOA системы, улучшение этой системы, описание и сравнение систем с SOA, аналогами, основанными на SOA и без использования данной архитектуры. А также, апробация полученных данных в ходе сравнения систем основанных на сервис-ориентированной архитектуре.

Работа изложена на 78 страницах и включает в себя 27 рисунков, 7 таблиц, 64 источника.

1 Современное состояние проблемы использования сервис-ориентированной системы

1.1 Анализ литературы по методам и средствам моделирования систем основанных на СОА

Во время поиска информации по данной теме мною было проанализировано и изучено большое количество источников.

Источники, которые использовались в качестве основы, являются разносторонними и включают в себя информацию о различных аспектах сервис-ориентированных информационных системах, включают в себя также положительные стороны в использовании и их интеграции в бизнес-процессы предприятий и организаций.

Работа с СОА требует разработки и применения различных методов управления, способных учитывать различные типы архитектуры [29, 31, 32, 33]. Все это выражается в следующих положениях:

- логика бизнес-процессов позволяет сервисам взаимодействовать или быть доступными;
- высокоуровневые компонентные сервисы основаны на развитии функциональности системы;
- архитектура системы, созданная на основе динамических отношений между ИТ-службами и бизнес-службами, адаптируется к определенным бизнес-требованиям;
- каждый элемент сервиса связан с определенной бизнес-функцией, также называемой конкретной задачей;
- своевременное использование стандартов на этапах разработки и сборки повышает частоту и качество взаимодействия всех компонентов друг с другом;

– композитная архитектура имеет такую особенность, как глубокий отклик на последние инновации как в бизнес-услугах, так и в информационных технологиях.

В качестве подробного анализа и последующего описания предлагаемых решений и мнений было принято решение рассмотреть лишь некоторые из приведенных статей.

Для одного из примеров можно рассмотреть статью [28]. Материал опубликован авторами О.Н. Граничин и И.Л. Шеронов, которые рассматривают проблемы эффективной организации работы системы в Высшей школы менеджмента Санкт-Петербургского государственного университета, которая построена на сервисно-ориентированной архитектуре. В своей статье авторы проанализировали современные достижения мировой индустрии в области информационных технологий и пришли к выводу, что наиболее современный подход к разработке системно-технических решений должен базироваться на концепции SOA.

Стандартными концепциями представленной архитектуры являются:

- информационная услуга;
- композитное приложение.

Чтобы получить более четкую картину, нужно рассмотреть каждую концепцию более подробно.

Под информационной услугой предполагается то, что прикладная функция автоматизированной системы может использоваться на протяжении всего цикла разработки необходимых приложений, создающих прикладную логику для определенных автоматизированных процессов.

Информационная услуга характеризуется такими свойствами:

- возможность многократного использования;
- она может быть идентифицирована одним или же несколькими интерфейсами, которые независимы в отношении друг к другу;

– для каждой из служб существует вероятность, называемая протоколами связи, которые, в свою очередь, могут позволить службам иметь общие соединения друг с другом.

Другой концепцией является составное или композитное приложение, которое представляет собой программное решение конкретной задачи путем объединения логики приложения процесса и источников данных в информационных сервисах. В основном составные приложения соединяются вместе в одном из различных процессов, таких как системные операционные процессы, которые соединяют различные этапы друг с другом, соединяя их в один интерфейс.

Применение данного подхода при построении архитектуры системы позволяет:

– существуют инструменты безопасности, которые позволяют доставлять сообщения вовремя и безопасно;

– скорость разработки значительно снижает затраты на достижение поставленной цели и значительно увеличивает скорость разработки необходимых приложений;

– различные транспортные протоколы и стандарты форматирования сообщений могут использоваться друг с другом.

Принимая во внимание все положительные стороны и методы этого подхода, необходимо рассмотреть вопрос о том, как уменьшить потребность в создании новых программ.

Упрощение сферы управления и взаимодействия устраняет необходимость создания новых программ и сопутствующего программного обеспечения, а также позволяет повторно использовать старые программы, что сокращает время разработки.

Существует также рационализация унаследованных процессов, что в свою очередь помогает сократить общее количество процессов, что в свою очередь приводит к использованию простых протоколов, снижая различные

затраты, поддерживаемые всеми приложениями, которые используются в процессах.

Следует помнить о том, что одним из обязательных условий создания данных и реализации системной архитектуры SOA – это единая инфраструктура, использующая все доступные и разрешенные описания сервисов, сообщений и протоколов доступа пользователей системы.

Авторы считают, что перед тем, как начать искать способы проектирования и запуска системы технологии SOA, некоторые условия должны удовлетворять созданные ранее:

- должно удовлетворяться такое условие – это, какие политики и процедуры планируются использоваться в будущем в поддержании проектирования, разработки;

- нужно создать набор определенных методов и инструментов, которые используются для поддержки SOA, а также методов, которые составляют требования к заказу на основе ранее предоставленной информации от организации. Именно тут будет происходить отслеживание и дальнейшая оценка эффективности за определенный отрезок времени, который соответствует циклу работы исследуемого объекта информации;

- пользователи с определенным префиксом поддерживают постоянную связь с сотрудниками, которые разрабатывают, внедряют и управляют SOA, а также внедряют их в другие организации, что по итогу создает непрерывную поддержку SOA с их стороны.

Если соблюдать все указанные шаги и условия для перехода к данной технологии, то у заказчика будут такие возможности, как:

- использование данного подхода, снижая в разы расходы на внедрение и обслуживание, также появляется возможность использовать такие приложения, которые могут быть адаптированы под те или иные запросы организации;

- создавать приложения, соответствующие требованиям заказчика и ориентированные на реализацию уникальных потребностей организации, что позволяет повысить эффективность работы сервиса, а также привести к сокращению сроков разработки приложений;

- использование таких обычных, но узкоспециализированных методов для достижения свободы в выборе платформы, инструментов разработки и программного обеспечения;

- поддержка работы для последующих программных продуктов, тесно работая друг с другом. Другими словами, создается единый сервис и к нему подключаются остальные, унаследованные права приложения.

Автор следующей статьи [13] отмечает помимо преимуществ подхода к сервис-ориентированной архитектуре и некие сложности в использовании.

Коптелов А.К. считает, что наилучший результат от СОА происходит благодаря многократному внедрению и использованию сервисов. Также, снижение затрат может быть достигнуто при автоматизации следующих процессов повторного использования уже разработанных ранее услуг, что, в свою очередь, избавляет от необходимости каждый раз создавать новые продукты.

Также СОА [45] упрощает интеграцию новых приложений в корпоративной информационной системе. Так как интеграция в техническом задании увеличивает бюджет проекта как минимум вдвое. В использовании технологии СОА новые программные продукты будут легко интегрироваться в существующую корпоративную информационную систему через некий механизм сервисов. Поэтому, за счет интеграций старых продуктов можно прийти к экономии затраченных средств [47].

Но в положительных аспектах есть и один сложный вопрос — передача данных между сервисами и разделение функциональности сервисов и используемых данных. Большинство сервисов не смогут иметь общих классификаторов, что значительно усложнит обмен данными между ними. И к сожалению, решения на данный момент нет. Технологическая интеграция не

может быть произведена до момента пока интерфейсы бизнес-процессов, а также используемые интерфейсы для их поддержки информационные системы не будут приведены.

Еще одной из известных недостатков является целостность информации, которая используется в различных сервисах. Если реализация процесса в системе будет неожиданно прервана, то придется проводить анализ не в едином журнале логов, внося изменения вручную, а работать с целой совокупности файлов логов различных отдельных сервисов, но чтобы они были, их еще следует создать заранее в систему. То есть в системе автоматизации нужно учитывать функциональность по восстановлению сбойных экземпляров процессов, без остановки её рабочей эксплуатации.

В статье [1] авторы А.Е. Сатунина, Л.А. Сысоева описали новизну SOA. Новизна этой архитектуры заключается в следующем:

- предоставляет возможность создавать новые бизнес-процессы на базе IT-сервисов и модернизировать существующие бизнес-процессы;
- новый подход к интеграции стандартных приложений;
- реализация взаимосвязи между SOA и BPM технологиями;
- новый подход к изменению и развитию функциональных возможностей информационных систем, основанный на использовании компонентов более высокого уровня.

Но есть и существенная разница между сервисами и программными компонентами. Это отражается в следующих подходах:

- сервисы реализуются в виде сложных программных пакетов, то есть сервис отражает не то, как реализовано программное обеспечение, а то, как оно используется;
- сервисы формируются с учетом потребностей компании, а дистрибуция программных компонентов должна осуществляться исключительно разработчикам программного обеспечения;

– каждая услуга всегда связана с определенной бизнес-функцией, а сами компоненты относятся к технологической категории отдельной архитектуры программного обеспечения.

Внедрение сервис-ориентированной архитектуры создает такие возможности, как для ИТ, так и для бизнеса. СОА предоставляет следующие преимущества:

- снижение затрат на ИТ-инфраструктуру;
- повышение согласованности ИТ-активов с бизнес-целями;
- сокращение времени разработки за счет повторного использования старых продуктов;
- срок окупаемости за счет снижения затрат на создание продукции;
- гибкость и адаптивность приложений благодаря проектированию систем, соответствующих стандартам.

Авторы этой статьи также дали хорошую систему, которая является ключевой отправной точкой и сценарием перехода СОА, эта система демонстрируется на рисунке 1.

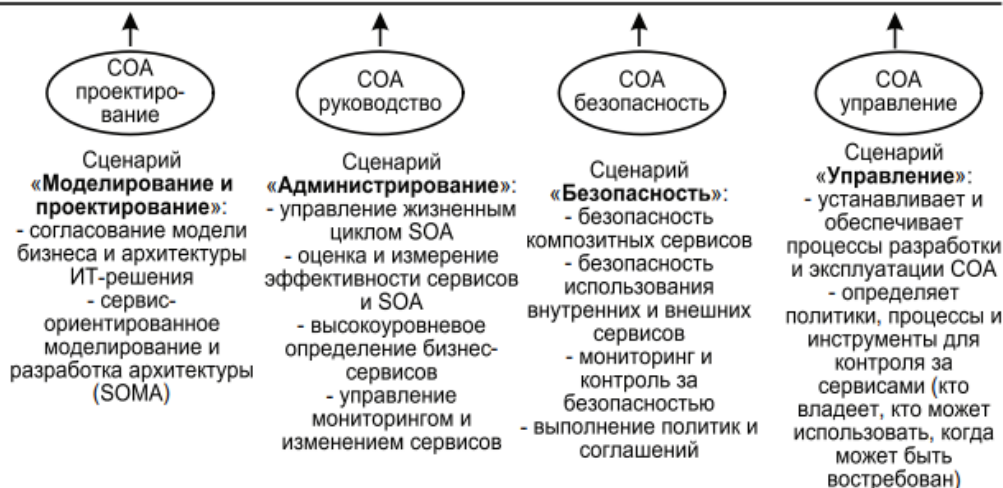
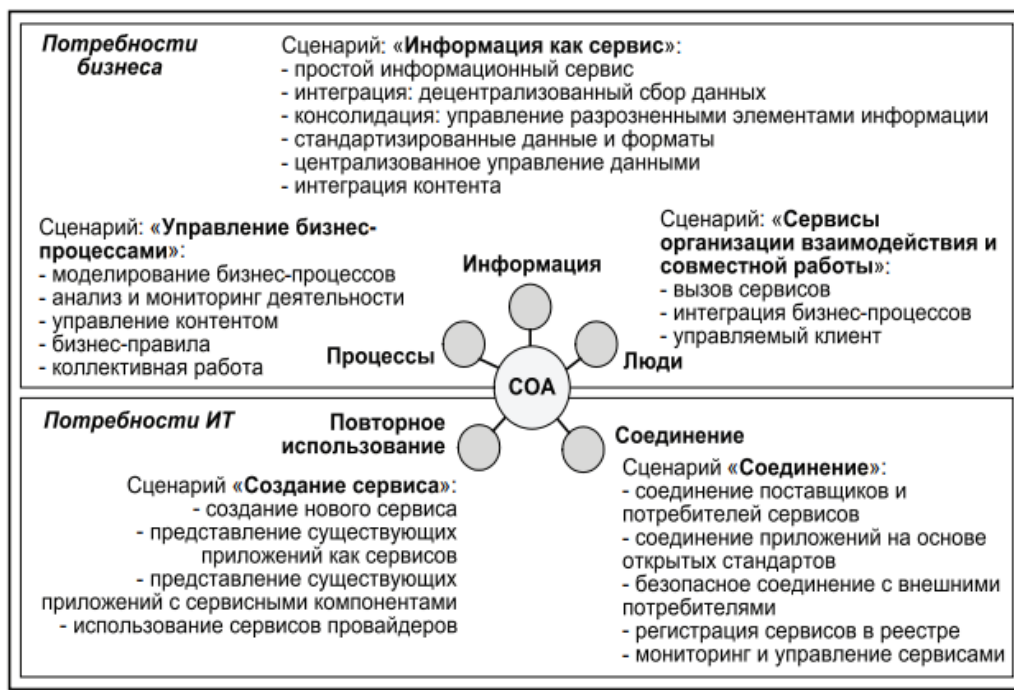


Рисунок 1 – Ключевые моменты и сценарий перехода на SOA

В источнике [2] автор указал процессы жизненного цикла управления SOA, которая в будущем может хорошо помочь в создании своего проекта.

Первой фазой является планирование, в нее входит:

- изучение методов и моделей управления;
- анализ имеющихся ресурсов и инструментов управления;
- определение роли и интересов организации;
- выделение направления стратегии в ИТ-стратегии;
- стратегия развития.

Второй шаг-это определение, оно включает в себя:

- определение организационных структур, ролей и уровней ответственности участников;
- координация политики по повторному использованию услуг в различных приложениях информационной системы;
- создание Центра передового опыта;
- определение необходимой реструктуризации существующей ИТ-инфраструктуры;
- определение рисков.

Реализация, в которую входят:

- управление безопасностью системы;
- внедрение и поддержка новых методов управления сервис-ориентированными информационными системами;
- обучение сотрудников данной методологии.

И последним этапом идет Оценка:

- контроль методов управления;
- контроль выполнения разработанных политик и возможность внести изменения;
- мониторинг, а также внедрение и поддержка новых методов анализа и соответствия уровню предоставления услуг;
- анализ соответствия услуг;
- анализ показателей эффективности;
- анализ показателей риска;
- анализ уровня повторного использования услуг.

Рассмотрев несколько основных источников, следует также затронуть и иностранные.

Например, в статье [50] выделяются 2 показателя, которые следует всегда держать в голове перед тем, как оценить работу системы, а именно:

- показатели эффективности;
- производственные затраты.

В данной статье очень подробно происходит описание этих двух показателей для успешной и эффективной работы производственной системы.

А в статье [51] подробно проводится статистика роста рынка программного обеспечения для бизнес-аналитики, который с 2012-2017 года растёт на 9.7% каждый год, также приводится анализ, как увеличить потребность в использовании COA.

В результате описания данной части научной работы были рассмотрены основные решения использования COA, выдвинуты основные положительные и отрицательные стороны данного метода, также рассмотрены этапы создания и примеры использования сервис-ориентированной архитектуры в организациях.

Рассмотрев различные статьи, в которых описано использование COA, перейдём к следующему пункту, где определим существующую проблему, поставим цель дальнейшего исследования, а так же определим задачи для её успешного выполнения.

1.2 Постановка проблемы, цели и задач для дальнейшего исследования

Основной проблемой, выдвигаемой в рамках данной научно-исследовательской работы, является пока еще растущая, но недостаточная популярность, а также сложность в ее создании на стадии планирования. На сегодняшний день в нашей стране существует достаточно малое количество организаций, которые используют эту систему. Можно отметить, что уже существуют такие организации и учреждения, которые уже внедрили эту систему. Одним из примеров является Высшая школа менеджмента СПбГУ.

Основываясь на выдвинутой проблеме, была определена основная цель научно-исследовательской работы. Её целью является рассмотрение возможности моделирования и предложения по внедрению COA системы.

На основе описанной цели в рамках данной научно-исследовательской работы выставляются следующие задачи:

- ознакомиться с научными работами на схожую или смежную тематику;
- рассмотреть различные решения СОА систем в российских, а также в зарубежных организациях;
- изучить способы внедрения СОА системы.

Выводы к разделу 1

В результате первой главы была проанализирована литература, сделаны выводы и краткий обзор некоторых источников. Для начала, в первой части работы были рассмотрены уже существующие гипотезы и решения в данной сфере. Далее были выделены основные достоинства и недостатки внедрения СОА системы, и был рассмотрен один из существующих примеров ее реализации.

Во второй части первой главы была определена основная цель исследования и на её основе выдвинуты задачи для решения поставленной проблемы. В результате изучения и анализа отобранного материала были проанализированы положительные и отрицательные стороны данного вопроса. На их основе можно сделать вывод, что создание решения и его внедрение будет положительно влиять на процесс создания организации. В свою очередь, организации и предприятия смогут организовывать или реорганизовывать рабочий процесс с меньшими затратами как средств, так и времени, благодаря повторному использованию заранее созданных программ, которые работают в единой системе.

2 Методология использования сервис-ориентированной архитектуры

2.1 Существующие методы сервис-ориентированной информационной системы

В настоящее время SOA система является востребованным направлением в организации работы предприятия или организации. Организация учебного процесса не является исключением, однако сложность заключается в том, что переход к данной системе является довольно тяжелым и требует определенных усилий в организации процесса и дополнительное время для реализации.

Для этого существует мониторинг сервисных показателей. Это единый процесс проверки и проведения системной проверки с целью оценки требований, выявления выявленных нарушений и выполнения выявленных замечаний.

В последние несколько лет вопросам сервис-ориентированной архитектуры (SOA) уделяется все больше внимания в связи с движением в направлении решения проблем, связанных с улучшением и обслуживанием различных сред. Модернизация системы программного обеспечения организаций, переход от старой системы к системе, основанной на SOA, стал основной тенденцией.

В последнее время во многих исследованиях подчеркиваются преимущества использования SOA при разработке новых ведущих мировых технологий, таких как Интернет вещи и облачные вычисления, а также микроуслуги. Это связано с тем, что SOA предлагает гибкую интеграцию и возможность повторного использования услуг благодаря своей сервисной модульной архитектуре.

SOA также предлагает прозрачность, поскольку инкапсулирует несколько приложений и источников данных в виде "черного ящика". Таким

образом, интегрированный пул ресурсов информационных технологий все еще может быть доступен, несмотря на существование различных технологий, языковых кодов, функциональных возможностей и платформ.

Сервисно-ориентированная архитектура возникла как архитектурный подход, который повышает производительность предоставления услуг существующих традиционных систем, сохраняя при этом их наиболее важные особенности. Этот подход, благодаря своей гибкости в применении, привлек внимание как академических, так и деловых кругов, особенно при разработке таких ведущих мировых технологий, как "облачные вычисления" и "интернет вещи".

Эта тема в настоящее время актуальна, поскольку сегодня наблюдается тенденция, укрепляющая взаимосвязь между ИТ-услугами и бизнесом, а также расширение сферы применения процессной модели управления, которые в итоге подстраиваются под быстрорастущий рост требований и изменений в создании бизнес-процессов.

«Целью мониторинга показателей ИТ-сервиса является выявление отклонений от требований SLA (уровень безопасности и сервиса для пользователя) и снижение рисков ИТ-сервиса при выполнении бизнес-процессов.

Также основной задачей системы мониторинга является сбор данных о состоянии сервисов и формирование их результатов. Системы мониторинга распределены по ИТ-сервисам менеджеров, бизнес-процессов и архитекторов» [39].

Структурная модель данных ИТ-сервиса в мониторинге показана на рисунке 2.

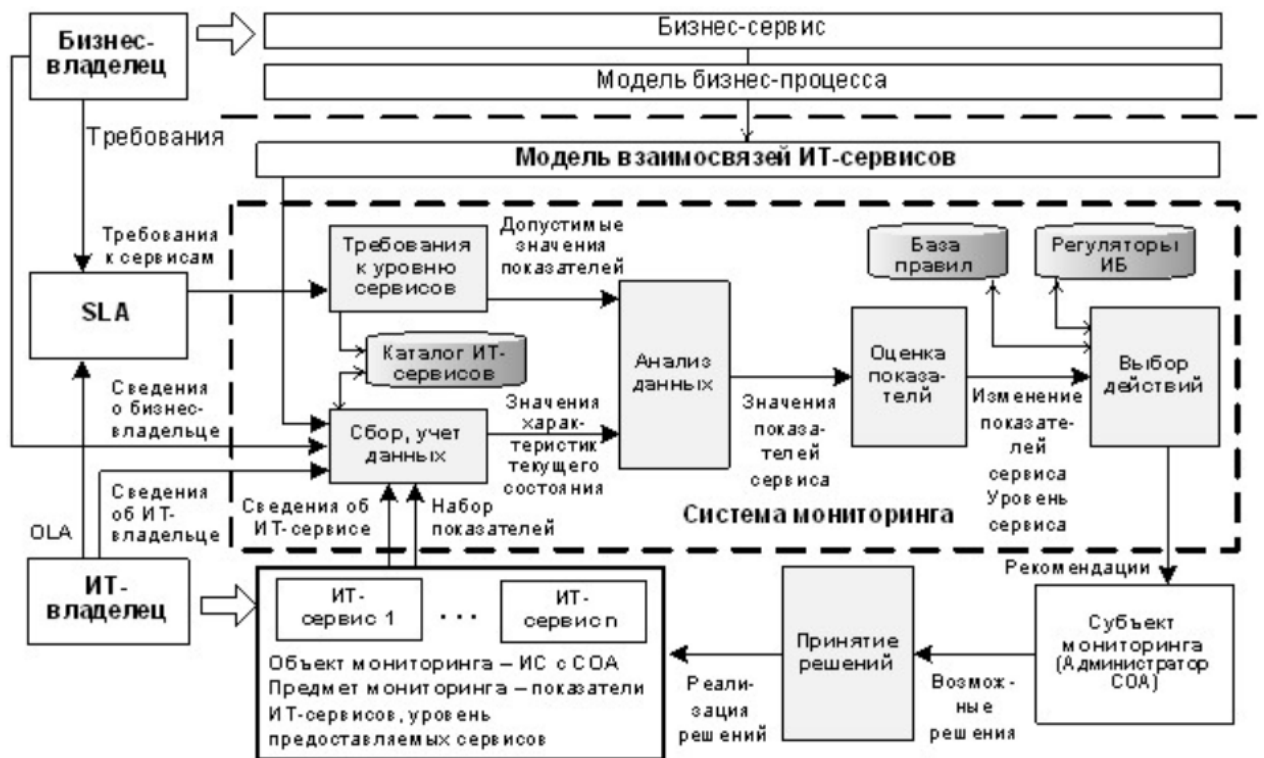


Рисунок 2 – Структурная модель мониторинга показателей ИТ-сервисов с использованием SOA

«При мониторинге измеряемых показателей используются Метрики или индикаторы, а также наборы показателей, которые предоставляют информацию, необходимую для своевременного принятия значений показателей в данном диапазоне» [43].

Согласно требованиям SLA, ИТ-сервис должен обеспечивать согласованность данных, поэтому и система мониторинга должна работать в течение длительного времени и относиться к обычным системам мониторинга.

Еще один пример работы с информационной системой SOA [46] представлен на рисунке 3.

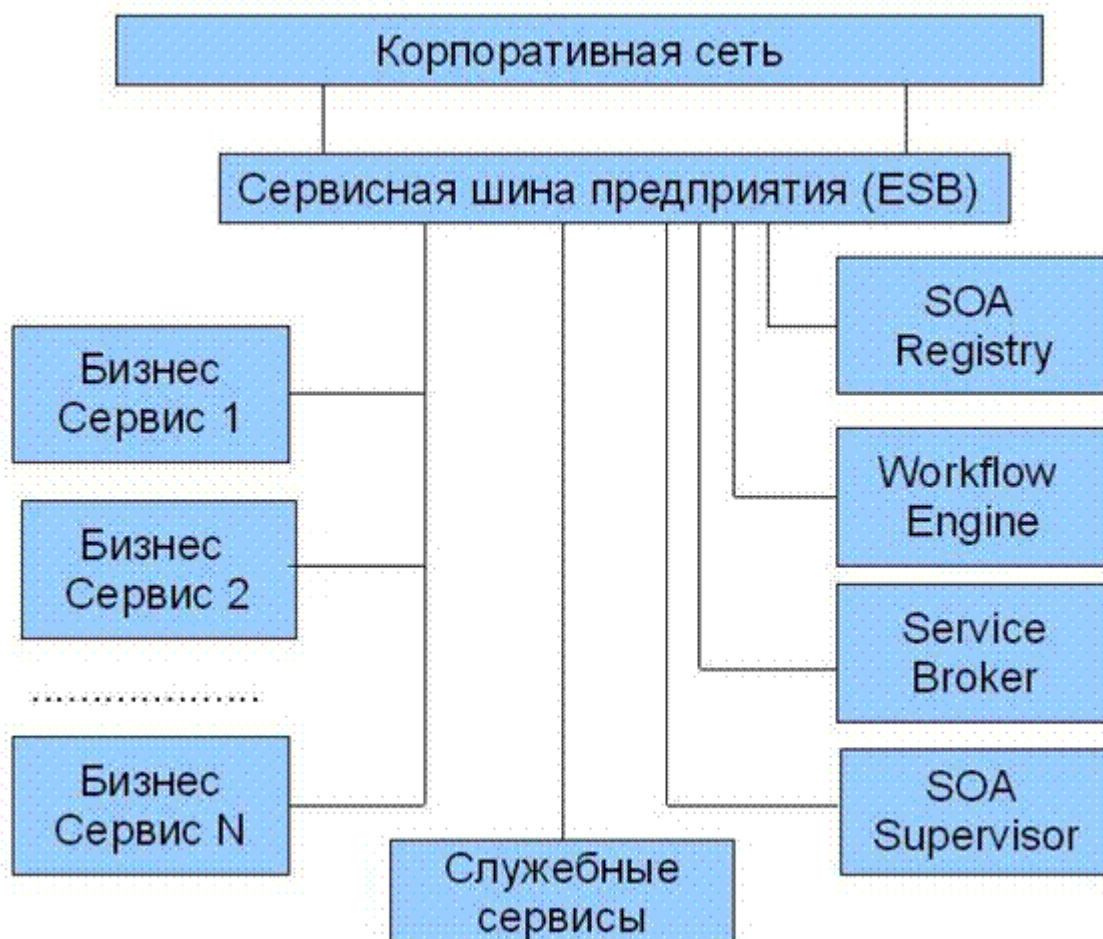


Рисунок 3 – Пример работы некоего предприятия основанного на SOA системе

Основные компоненты (показанные на рисунке 3) - сервисная шина предприятия (ESB), SOA реестр (SOA Registry), workflow engine, сервис брокер (service broker), SOA супервизор (SOA supervisor), все они взаимодействуют и играют свою роль в системе [23, 17].

Теперь рассмотрим как SOA система работает в образовательных учреждениях. Какие методы используются при создании такого формата обучения.

Саратовский государственный университет использует подобную систему.

«Основные качества SOA в образовании: во-первых, способность к оперативному управлению при изменении образовательных технологий; во-вторых, способность сохранять инвестиции за счет многократного

использования аппаратных и программных средств; в-третьих, упрощение учебного процесса и передача их на длительный срок» [30].

Сервис-ориентированная архитектура отличает влияние модификации одного компонента на другой. Это означает, что любые изменения приложения, связанные с его жизненным циклом, возможны. Кроме того, есть возможность вернуться к стабильности, если изменения приведут к ошибкам. Управление использует не только сервисы, но и оркестровку, правила, профили аудита и другие архитектурные атрибуты. Мониторинг соблюдения общих соглашений часто включает в себя уведомления, продолжительность процессов, динамику очередей, наличие заблокированных сообщений и т. д. На рисунке 4 демонстрируется пример работы СОА в образовательном учреждении.

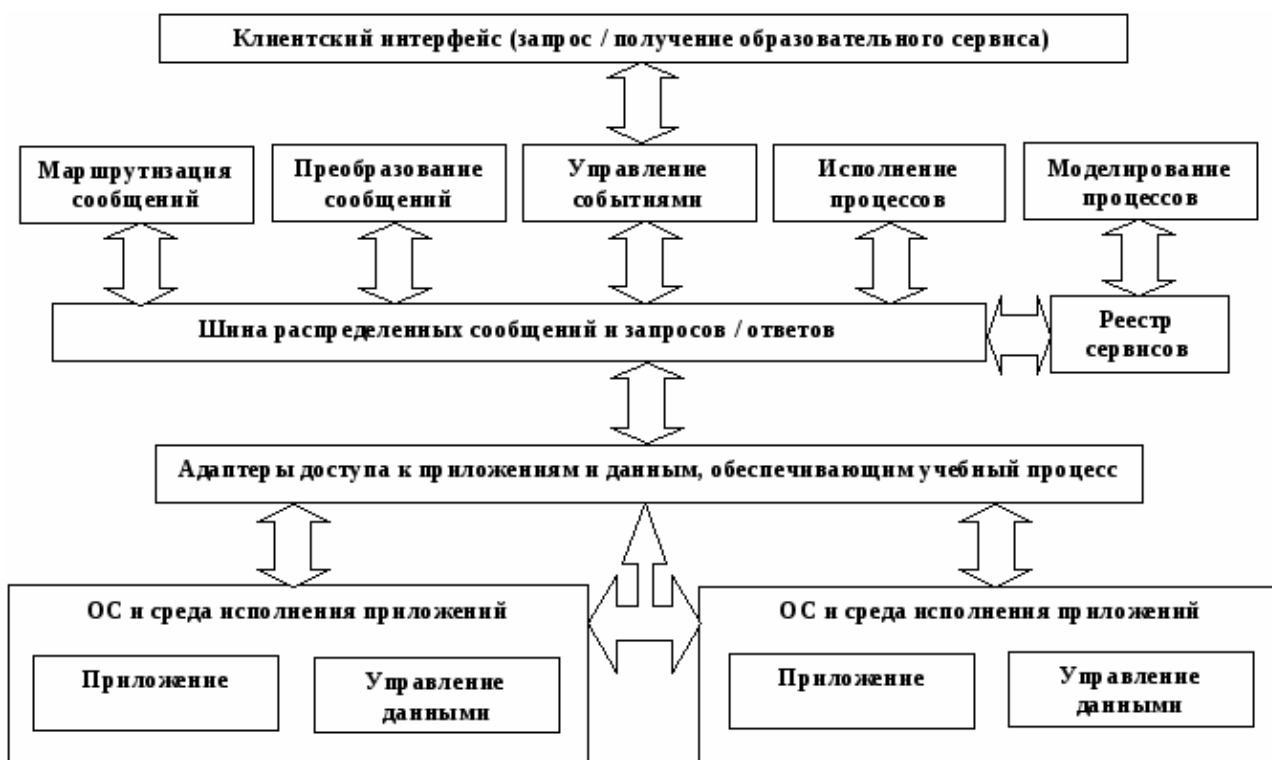


Рисунок 4 – СОА в образовании

«По сравнению с другими возможными образовательными средствами SOA, это разочарование в образовательных технологиях, в которых создаются конкретные приложения. Оркестр сервисов нейтрален в вопросах, связанных с технологиями, действиями и новостями. В этих сообщениях не содержится информации о внутренних сторонах сервиса» [30].

Нижележащие элементы архитектуры означают существующие приложения и связанные с ними хранилища данных. Приложения работают в операционных системах на некоторых платформах. Переход образовательных учреждений на сервис-ориентированную архитектуру не повлияет на этот уровень. На верхнем уровне интерфейса клиента имеются механизмы обмена информацией между Пользователем и внешними программными объектами. В современных образовательных технологиях - это корпоративный портал, созданный для внешних приложений.

В источнике [38] показывается метод работы четырех уровневой SOA. Это показано на рисунке 5.

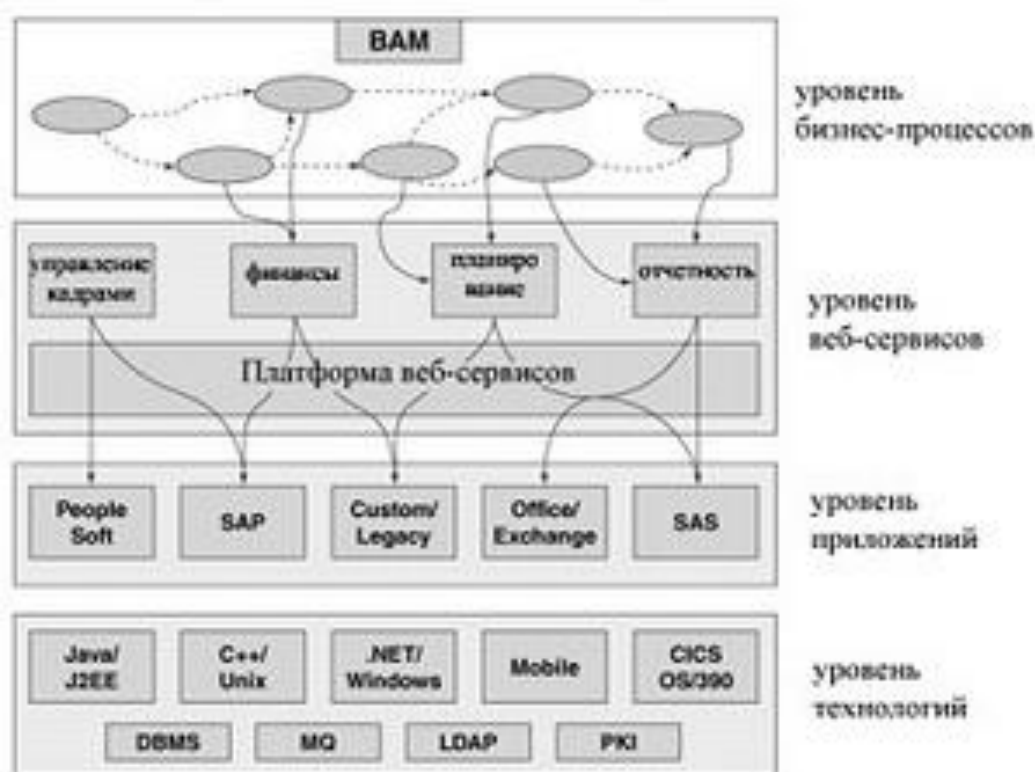


Рисунок 5 – Четырехуровневая SOA модель в прототипе организации

«Преимущество SOA - ориентированного подхода к технологиям веб-сервисов, которое решает проблемы интеграции информации изнутри, заключается в быстром изменении системы управления бизнес-процессами, создании более простых вещей, управлении и поддержке приложений.

Если взять пример с виртуальным университетом, то будут включаться следующие уровни:

- уровень Бизнес-процессов, на котором находятся организационные процессы в виде логически связанных бизнес-задач. Где представлен учебный процесс, состоящий из четырех основных этапов: введение в образовательное учреждение, обучение, итоговое тестирование, выпуск/отчисление.
- уровень функционирования образовательной среды, включающей в себя все основные группы веб-сервисов и сервисов, отвечающих определенным задачам в рамках бизнес-процессов» [38].

На рисунке 6 автор демонстрирует фрагмент системы, которая основана на SOA.

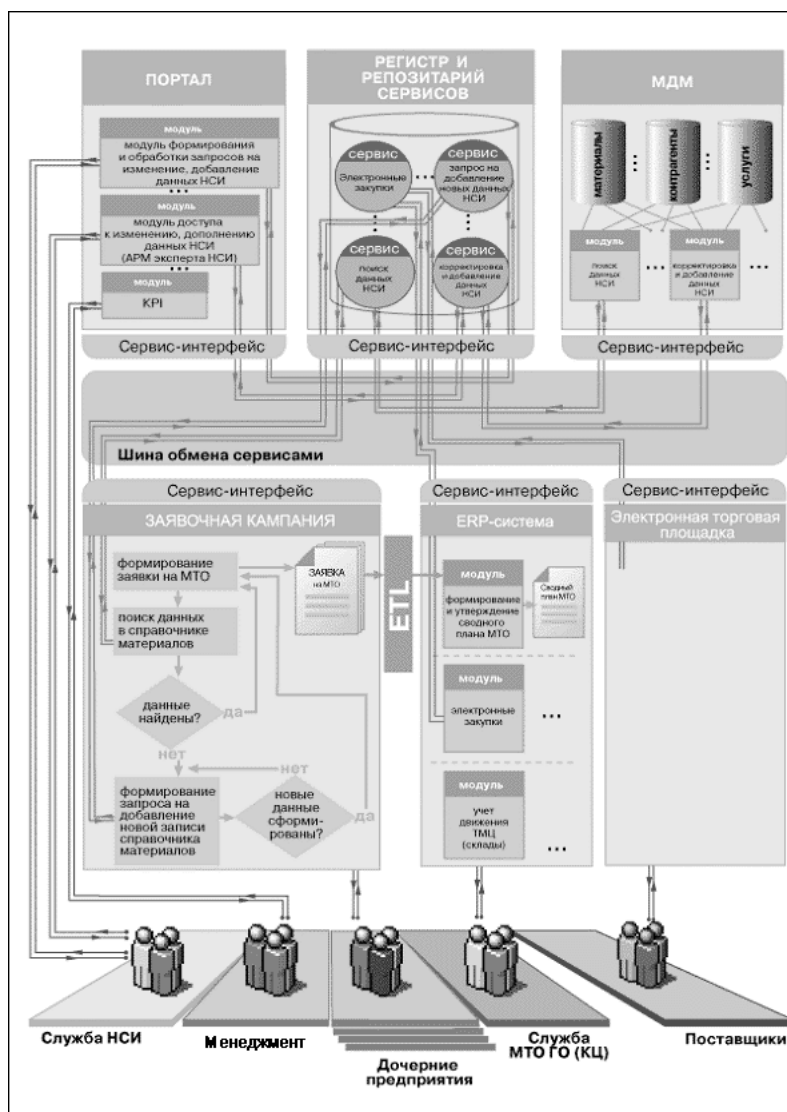


Рисунок 6 – Фрагмент системы основанной на SOA

На данном рисунке видно взаимодействие пользователей и системы, где происходит работа с подключенными модулями системы. Автор считает, что каждый из элементов сервисов должен быть доступен всегда, независимо от того, где сотрудники находятся, главное это выход в интернет.

«Спецификация "сервисы" предназначена для хранения и описания адресов сервисов (RPC), зарегистрированных в ней, включая расположение в сети, правила вызовов и т. д., За исключением запросов и обмена данными между сервисами и информационной шиной SOA, основным компонентом этой архитектуры является уникальный портал SOA в контексте данного портала» [7].

Теперь следует провести сравнение работы организации с применением SOA системы и без нее.

2.2 Сравнение работы организации без использования SOA и с внедренной системой

Одним из важных пунктов в рассмотрении и изучении данной системы является сравнение работы организации с применением данного метода и без него. Сравнение работы предприятия демонстрируется на рисунке 7.

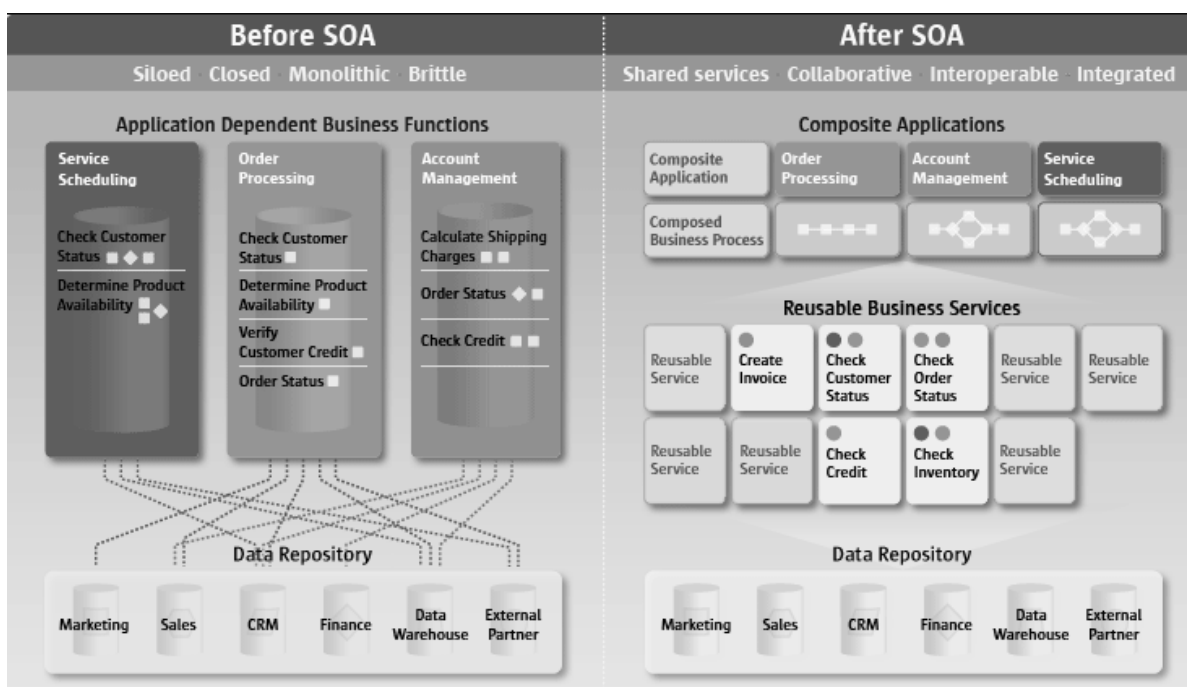


Рисунок 7 – Сравнение работы предприятия до внедрения СОА и после

В традиционных ИТ-архитектурах действия бизнес-процессов, приложения и данные заблокированы в независимых, часто несовместимых "хранилищах". Пользователи должны создавать отдельные сети, приложения и базы данных для выполнения цепочки действий, завершающих бизнес-процесс. Независимые силосы поглощают чрезмерное количество ИТ-бюджета и времени персонала для обслуживания.

Однако, сервис-ориентированная информационная система, предоставляет данные, необходимые для деятельности бизнес-процессов, в виде интегрированной службы. Пользователям больше не нужно входить в несколько систем, искать соответствующие данные и интегрировать результаты вручную. Информация отображается в виде одного приложения, доставленного на одном экране, все с одним логином [27]. А это и является преимуществом СОА системы.

Автор данного сравнения [58] ярко показал преимущество работы с данной системой.

Таким образом, в процессе написания данной части отчета научно-исследовательской работы 2 были рассмотрены существующие методы сервис-

ориентированной информационной системы и их схемы. Еще были изучены преимущества SOA системы в целом.

Перейдём к заключительной части отчёта, где опишем выводы по проделанной работе.

2.3 Характеристика и моделирование существующих архитектур

Сервис-ориентированная архитектура (SOA) – это модульный стиль проектирования, где все сервисы или службы предоставляются другим компонентам системы посредством протокола связи внутри связей.

В современном мире крайне необходимо всем компаниями уметь правильно организовывать работу, однако при выборе метода организации всей структуры мало кто берет SOA систему [37, 60].

Как показывает статистика [12, 59], данный метод только лишь начинает набирать свои обороты.

Самыми популярными подрядчиками в данной сфере являются [36]:

- Axelot (Москва), который насчитывает 63 проекта на данный момент;
- Neoflex (Москва), где 41 проект;
- Diasoft (Москва), где 39 проектов.

Среди самых популярных российских заказчиков [36] на организацию своей системы при помощи SOA являются:

- банки. ВТБ, Ренессанс-Капитал, Русфинансбанк;
- телеком. Комстар (нынешний МТС), Вымпелком;
- торговля. Евросеть, X5 Retail Group (Пятерочка, Перекресток);
- топливо-энергетический комплекс (ТЭК). Татнефть, Росэнергоатом;
- госсектор. Мэрия Москвы, Управление информатизации г.Москвы; Транспорт. Аэрофлот.

«Ее принципы не зависят от поставщиков и других технологий. В сервис-ориентированной архитектуре несколько сервисов взаимодействуют друг с другом одним из двух способов: посредством передачи данных или посредством двух или более сервисов, координирующих какую-либо деятельность» [64].

За основу СОА лежит принцип многократного использования элементов информационных технологий, убирает дублирующие элементы в программном обеспечении, а также делает универсальными типовые операции в системе, что переводит перевод операционной модели компании на централизованные процессы.

Программные компоненты распределяются по разным узлам системы и могут быть либо независимыми, либо слабо зависимыми между собой.

Программные комплексы, разработанные в соответствии с SOA, часто реализуются в виде веб-сервисов, которые внедряются при помощи стандартных протоколов, такие как SOAP, WSDL.

СОА является отличным способом слияния и неоднократного использования различным компонентов для создания сложных программных комплексов.

Всего существует три программные архитектуры:

- монолитные;
- сервис-ориентированные;
- микросервисы.

Монолитная архитектура показана на рисунке 8.



Рисунок 8 – Монолитная архитектура

Данная архитектура является последующим друг за другом шагами в использовании, то есть при использовании монолитной архитектуры при выходе из строя одного из блоков, перестает работать вся система или веб-сайт.

Например, если при использовании какого-нибудь сайта перестанет отвечать любой блок, то весь сайт перестанет отвечать на запросы, пока не будет устранена проблема. Также, при использовании монолитной архитектуры перед тем, как получить нужный результат, необходимо пройти все блоки, что создает лишнее время ожидания.

Чтобы избежать подобных проблем, были созданы две архитектуры, это сервис-ориентированная (СОА) и микросервисы. Для начала, рассмотрим СОА систему, после микросервисы, после сравним их, и рассмотрим, какая из этих архитектур лучше всего подходит для использования в разработке. На рисунке 9 показана сервис-ориентированная архитектура.

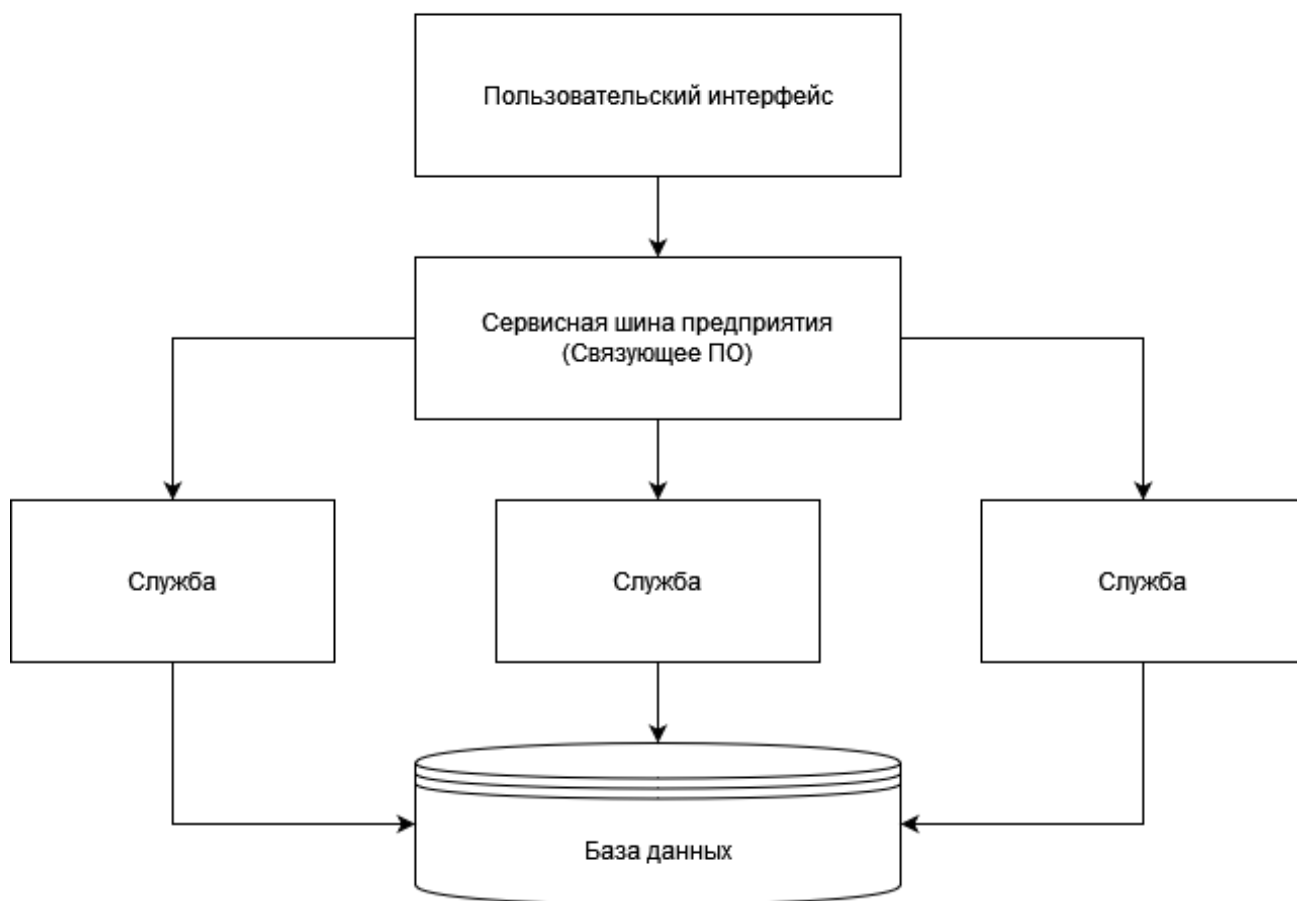


Рисунок 9 – Сервис-ориентированная архитектура

На данном рисунке показана простейшая модель СОА системы. В ней можно наблюдать те самые службы, которые можно отдельно подключать без вмешательства в другие службы.

Также стоит отметить, что данный метод хорош тем, что если какая-то служба или модуль выходит из строя, то остальные остаются работать независимо от нее и это не приводит к проблемам, так как все используется в единой базе данных [34, 48, 56]. Например, при работе на сайте перестает работать кнопка регистрации или авторизации, в монолитном варианте это привело бы к невозможности пользоваться всем сайтом, однако, в СОА пользователь может также пользоваться сайтом, но с некоторыми ограничениями.

Хорошим примером работы данной системы является взаимодействие портала Госуслуг и ведомственной системы [35]. Данная схема показана на рисунке 10.

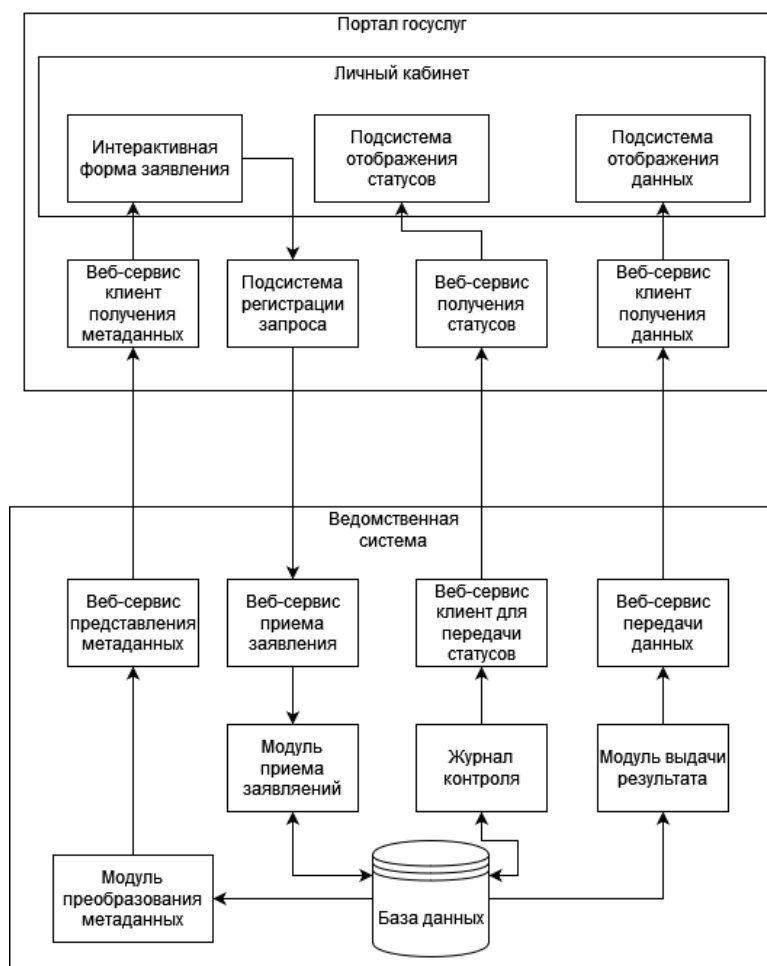


Рисунок 10 – Схема взаимодействия портала Госуслуг и ведомственной системы

Данный рисунок хорошо демонстрирует работу портала Госуслуг. Используется единая база данных у ведомства, где все хранится и в случае отказа, например, журнала контроля, то сама система будет работать и оказывать услугу, единственное, не будет в конечном итоге отображаться статус услуги.

Однако если же выйдет какой-нибудь из основных блоков, отвечающих за работу того, или иного ведомства, то будет сообщение о недоступности сервиса [11], но сам портал будет работать и дальше. Пример ошибки показан на рисунке 11.

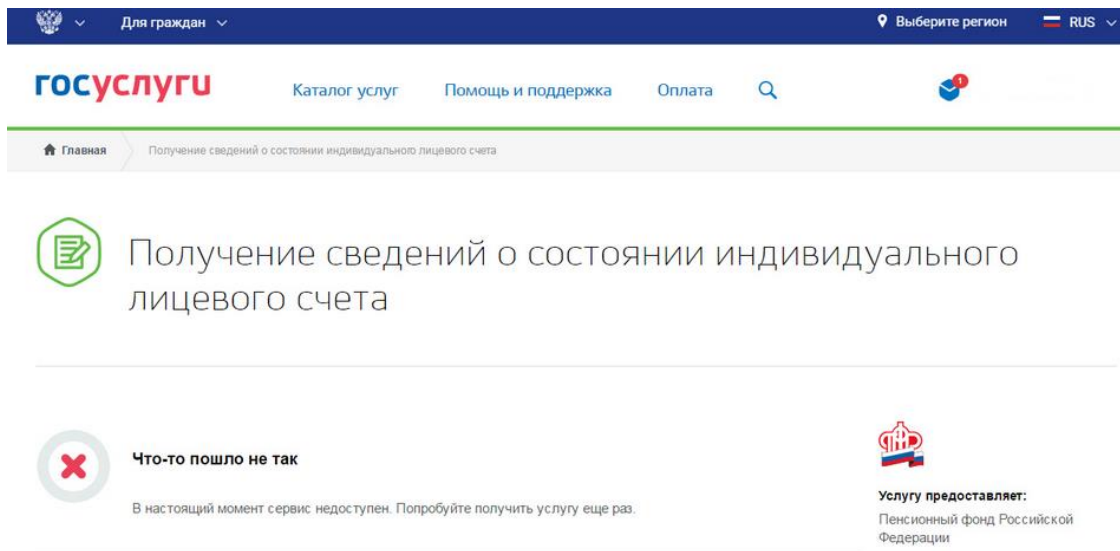


Рисунок 11 – Пример недоступности сервиса на портале Госуслуг

Теперь стоит рассмотреть третью архитектуру, а именно микросервисы. На рисунке 12 показана простейшая архитектура, созданная при помощи микросервисов.

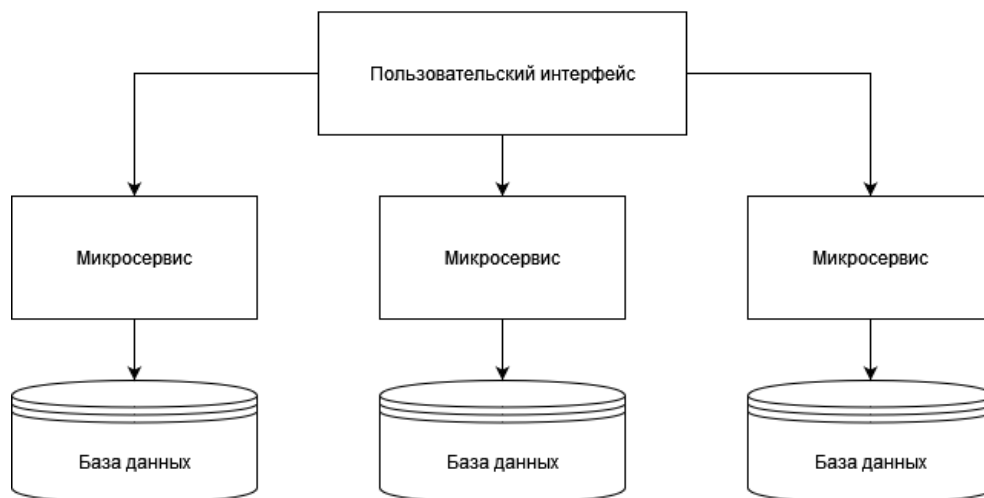


Рисунок 12 – Архитектура с использованием микросервисов

На данном рисунке демонстрируется примитивная реализация микросервисов. Как видно, пользователь через интерфейс делает запрос и через некие отдельные слабосвязанные сервисы получает результат в отдельных базах данных.

Однако стоит отметить, что в случае отказа одного из этих микросервисов система хоть и продолжит свою работу, однако у пользователя не будет

доступа к одной из баз данных. Например, если откажет база данных хранения учетных записей, то перестанет работать также и поиск по пользователям, так как все хранится в отдельной базе.

Если снова взять пример с Госуслугами, то если бы была использована эта архитектура, то при отказе одного из микросервиса были бы проблемы с отображением уже созданных данных в личном кабинете, например, перестали бы отображаться паспортные данные, так как они находились бы в отдельной базе данных. В момент, как при использовании СОА все полученные когда-то данные находятся в единой базе.

Рассмотрев более подробно все три архитектуры, можно дать им краткую характеристику.

Монолитная архитектура – состоит из взаимосвязанных, последующих друг за другом и неделимых блоков. Она хороша тем, что проста в разработке и менее затратная в финансовых вопросах, однако, совершенно не подходит в больших компаниях, так как чем больше блоков, тем медленней работает вся система в целом.

Сервис-ориентированная архитектура – создается уже с применением отдельных служб или блоков, которые уверенно связаны друг с другом, но при выходе из строя одного из них не выводит из строя всю систему. Также имеет единую базу данных.

Микросервисы – делится на мелкие, независимые друг от друга сервисы со своими базами данных. Их проще всего интегрировать друг с другом, но в случае отказа одного сервиса, не будет связи с одной из баз данных.

Рассмотрев архитектуры можно понять, что в современном мире монолитная архитектура устарела и является неактуальной, поэтому в дальнейшем рассматривать ее не будем [15, 16].

Перед тем, как сравнить СОА и микросервисы, следует отметить особенности каждой из архитектур. После этого будет составлена таблица сравнения, отмечены все преимущества и недостатки архитектур и сделаны выводы по полученным данным.

2.4 Описание и сравнение сервис-ориентированной архитектуры и микросервисов

Рассмотрев существующие архитектуры и смоделировав пример работы СОА системы на примере портала Госуслуг, следует отметить все особенности СОА и микросервисов, чтобы в дальнейшем можно было определить что лучше всего подходит для основы в использовании в разработке систем.

Особенности сервис-ориентированной архитектуры:

- сервис-ориентированная система использует интерфейсы, которые решают проблемы в интеграции больших систем;
- данная архитектура взаимодействует с клиентами и поставщиками с использованием XML;
- логирует сообщения для улучшения измерения производительности;
- логирует сообщения об обнаружении атак на систему безопасности;
- при повторном использовании служб стоимости каждой разработки и управления ПО становится ниже со временем.

Особенности микросервисов:

- модули слабо связаны между собой;
- также как и СОА может быть модульным;
- легко использует несколько технологий во внедрении нескольких функций;
- низкая стоимость изменений в системе.

Следующим этапом стоит провести сравнение между этими архитектурами [39, 61]. Таблица 1 и 2 сравнивают две архитектуры между собой.

После, будет описание этих архитектур, где сложно сказать какая лучше будет лучше без согласования с заказчиком.

Для начала, в таблице 1 будет демонстрироваться сравнение, где указано преимущество SOA над микросервисами.

Таблица 1 – Сравнение сервис-ориентированной архитектуры и микросервисами, где SOA выигрывает у микросервисов.

Сервис-ориентированная архитектура	Архитектура с использованием микросервисов
Имеет единую базу данных, в которой хранятся все полученные данные с других модулей/блоков.	Все данные полученные при помощи данной архитектуры хранятся в отдельных базах данных, которые не связаны между собой.
Связь между всеми сервисами в приложении основаным на этой архитектуре использует простые и понятные подходы.	Используются сложные API.
Максимально использует все приложения в системе.	Все сосредоточено на развязке между сервисами.
Полный стек.	Монолитный.
Совместное использование компонентов в системе.	Не включает в себя такого понятия, как совместное использование компонентов в системе.
Единое хранение всех данных в системе, полученные в системе.	Каждый сервис имеет независимое хранилище данных.
Совместное использование всех ресурсов.	Ограниченный контекст связей в системе.
Все приложения созданы для выполнения многочисленных бизнес-задач.	Создана для работы с одной бизнес-задачей.
Многопоточная архитектура с несколькими накладными расходами для обработки задач.	Однопоточный. Используется с функциями Event Loop для неблокирующей обработки задач.
Систематическое изменение, необходимое для модификации монолита.	Систематическое изменение заключается в создании нового сервиса.
Создано для максимального повторного использования сервисов приложений.	Акцент выделен на развязке.

На данном сравнении можно подтвердить, что главное преимущество сервис-ориентированной архитектуры над микросервисами является повторное

использование сервисов и нет необходимости постоянно создавать новый сервис или службу, а достаточно модифицировать нужный сервис.

Далее, на второй таблице будет сравнение двух систем, где микросервисы имеют преимущество над системой, основанной на сервис-ориентированной архитектуре.

Таблица 2 – Сравнение сервис-ориентированной архитектуры и микросервисами, где микросервисы имеют преимущества над СОА.

Сервис-ориентированная архитектура	Архитектура с использованием микросервисов
Любое крупное изменение в системе требует модификации основного блока.	Любые изменения позволяют создавать новые сервисы отдельно друг от друга.
Поддерживает несколько протоколов сообщений.	Использует легкие протоколы, такие как HTTP, REST или Thrift API.
СОА предназначен для одновременного и совместного использования ресурсов между службами.	Размещает службы, которые функционируют между собой независимо.
Менее гибкая в развертывании.	Более гибкая и простая в развертывании.
Технологический стек ниже по сравнению с архитектурой с микросервисами.	Стек может быть большим, нежели в сервис-ориентированной архитектуре.
Бизнес-единицы зависимы друг от друга.	Бизнес-единицы независимы друг от друга.
Развертывание занимает более длительное время.	Развертывание простое и менее трудоемкое.
Использует корпоративную сервисную шину (ESB) для связи.	Он использует менее сложную и простую систему обмена сообщениями
Размер программного обеспечения больше, чем у любого обычного программного обеспечения.	Размер программного обеспечения зависит от количества подключенных сервисов.
Менее масштабируемая архитектура.	Высоко масштабируемая архитектура.

По результатам данной таблице можно сделать вывод, что СОА система проигрывает у микросервисов в том, что при крупном изменении необходимо

произвести модификацию основных блоков и менее гибкая в развертывании, нежели микросервисы.

Также, система менее масштабируемая и размер итогового ПО больше, чем у аналогов.

В заключение сравнений нужно отметить сильные стороны каждой из систем, которые нельзя сравнивать, но необходимо описать, так как эти аспекты являются важным аспектом при выборе архитектуры, на которой будет работать система.

Сильные стороны сервис-ориентированной архитектуры:

- лучше для крупномасштабных интеграций;
- происходит общение через ESB;
- приложение SOA, состоит из нескольких сервисов;
- компоненты бизнес-логики хранятся в простых проводных протоколах одного домена службы (HTTP с XML JSON), API управляется с помощью SDK / клиентов;
- общее управление и стандарты.

Сильные стороны микросервисов:

- лучше для небольших и веб-приложений;
- общение через API;
- данная архитектура состоит из десятки услуг;
- бизнес-логика может существовать между служебной шиной доменов, как отдельные уровни между службами;
- спокойное управление, так как оно больше ориентировано на свободу выбора сервисов.

Сравнив эти две архитектуры, теперь следует отметить преимущества и недостатки. Это нужно для того, чтобы сделать выводы какая из представленных архитектур является более удобной во внедрении и использовании.

Преимущества сервис-ориентированной архитектуры:

- проводить изменения в системе и обновлять сервисы не составляет проблем;
- все службы имеют одинаковую структуру каталогов, что дает возможности получать доступ к данным служб из одного каталога;
- сервисы взаимодействуют с другими приложениями, используя общий язык, что приводит к независимости от платформы;
- все блоки небольшого размера по сравнению с полноценным приложением. Это нужно для того, чтобы проще отлаживать и тестировать независимые сервисы;
- позволяет повторное использование всех сервисов существующей системы, попеременно создавая новую систему;
- возможность подключать новые услуги или модернизировать существующие, чтобы удовлетворять новым бизнес-требованиям;
- можно вручную повышать производительность, функциональность службы и легко производить обновление всей системы;
- возможность регулировать или изменять различные внешне подключенные среды;
- разработка отдельных приложений без замены уже существующих приложений.

Недостатки сервис-ориентированной архитектуры:

- все входные данные должны быть проверены перед подключением в основную службу;
- является дорогостоящей архитектурой с точки зрения затрат ресурсов, развития и технологий;
- некоторым веб-службам необходимо часто отправлять и получать информацию о работе;
- слишком затратная для инвесторов;

– когда слишком много служб взаимодействуют между собой, это приводит увеличению времени запроса.

Теперь стоит рассмотреть преимущества и недостатки микросервисов.

Преимущество микросервисов:

– простой шаблон архитектуры, который намного проще понять разработчикам без должных знаний;

– IDE быстрее делает разработчиков быстрее и продуктивнее;

– веб-контейнер запускается быстрее, что ускоряет процесс развертывания и разработки;

– возможность разрабатывать, развертывать и масштабировать каждый сервис независимо от других.

Недостатки микросервисов:

– микросервисы созданы на основе монолитных приложений, поэтому он не имеет поддержки для разработки распределенных приложений;

– тестирование каждого сервиса более сложное, ибо неизвестно как будет работать с другими;

– необходимо каждый раз реализовывать механизм межсервисных коммуникаций;

– реализация вариантов использования требует координации между командами разработчиков всех сервисов;

– стоит очень дорого. Так как появляется требование поддержки различных баз данных для работы разных бизнес-задач.

Рассмотрев каждую из этих архитектур и смоделировав их, можно сделать следующие выводы и понять, какая из этих архитектур лучше всего подходит под организацию системы в компаниях.

SOA создана на повторное использование всех сервисов приложения, а у микросервисов все отдельно. SOA не монолитное, в то время, как микросервисы работают на основе монолитной архитектуры.

Приложения основаны на СОА, созданы для выполнения многочисленных задач одновременно, также использует единое хранилище всех полученных данных, в момент, когда микросервисы созданы для выполнения одной задачи и требуется создавать много баз данных для хранения всех данных.

Сервис-ориентированная архитектура предназначена для совместного использования ресурсов между службами, в тот момент, как микросервисы нужны для размещения служб, которые могут функционировать независимо.

СОА — это менее масштабируемая архитектура, в отличие от микросервисов.

Использование сервис-ориентированной архитектуры набирает популярность, но еще не стала массовой, в то время как микросервисы уже давно популярны.

На основе полученных данных можно прийти к выводу, что сервис-ориентированная архитектура — это идеальный метод архитектуры для больших и сложных бизнес-приложений. Он наиболее подходит для сред, которым необходима интеграция со многими и разнообразными приложениями.

Однако стоит понимать, что если принимать решение о выборе между СОА и микросервисами, нужно всегда помнить о функциях и сервисах, которые будут использоваться в будущем, а также о том, как система будет работать, и о том, какая масштабируемость нужна в итоге.

И СОА, и микросервисы предоставляют отличные варианты в зависимости от ключевых различий в каждой организации, ориентируясь на их потребностях в разработке.

2.5 Методы внедрения сервис-ориентированной архитектуры

СОА представляет собой набирающую обороты систему, которая позволяет организовывать структуру организаций более продуктивно. Использование СОА подразумевает использование модульного метода, где каждый модуль выполняет свою работу и это позволяет при выходе одного из модулей не прекращать работу всей системы.

Развитие современных информационных технологий определило использование сервис-ориентированной архитектуры [14]. СОА позволяет при необходимости, без особых проблем использовать неоднократно одну и ту же систему, подключая или убирая нужные модули. Это приводит к сокращению времени разработки и последующих затрат на обновление модулей.

В настоящее время данная архитектура постепенно набирает обороты, но, к сожалению, еще менее популярна, нежели монолитная архитектура или микросервисы, которые работают на основе монолитной архитектуры. С каждым годом появляются новые организации, которые постепенно меняют или организуют свою структуру, беря за основу сервис-ориентированную архитектуру.

Перед внедрением сервис-ориентированной архитектуры следует определить несколько аспектов:

- определить цели применения СОА;
- определиться с программными инструментами и их стоимостью, которые будут применяться в результате внедрения;
- определиться с методами внедрения.

Для проведения подобных изменений, нужно рассмотреть различные методы внедрения инноваций, которые могут применяться и при внедрении. Рассмотрим известные методы подробнее [22, 25]:

- метод конкретизирующих документов [3]. Является самым распространенным методом. Его основная идея строится на растолковании

более подробно, что есть СОА, для чего он нужен и написании инструкций, положений и регламентов для каждого этапа внедрения. Данный метод направлен на снижение масштабов изменений в организациях-заказчиках. В рамках данного способа внедрения создается конкурс желающих протестировать работу сервис-ориентированной архитектуры, заключается договор, создается система с использованием данной архитектуры и на этой базе проводится эксперимент. Далее подготавливаются инструкции, которые уточняют действия прописанных положений. Объемное внедрение в данном методе полностью пресекается. В рамках данного метода разрабатываются положения для интеграции этой системы в организации-заказчики [40, 44];

– метод «частичного внедрения». Представленный метод основывается на разделении процесса внедрения СОА на простые составные элементы. Нужный итоговый результат сформировывается на основе постепенного внедрения этих составных элементов, которые разделили на начальном этапе. В качестве примера можно рассмотреть реорганизацию не всей организации, а лишь какого-либо из отделов, чтобы можно было понять принцип работы и нужно ли это им. При этом работа самой организации остается незатронутой [21, 42]. Благодаря такому методу появляется возможность вернуть все в исходное положение, не откатывая назад всю структуру организации;

– метод «вечного эксперимента» [20]. В рамках данного способа процесс внедрения СОА проходит при помощи проведения испытаний на платформе одной организации-заказчика. В результате проведения подобных испытаний формируется оценка получаемых результатов в течение довольно длительного промежутка во времени. В случае успешной апробации производится внедрение в другие организации-заказчики. Однако бывают случаи, когда осуществляется искусственное удержание, утверждая, что еще не все исследовано, что полученных данных недостаточно и нужно заново проводить эксперименты и заново собирать всю информацию. Так может длиться до бесконечности;

– метод «отчетного внедрения» [26]. Данный метод предполагает, что сервис-ориентированная архитектура либо практически не используется в организациях, либо используется не в полную силу. Например, кто-то пытался внедрить, но ничего не вышло. Либо, какой-либо из стадий внедрения отказались от этой архитектуры. Тем не менее, при составлении отчетной документации записано, что уже используется в полной мере. Как правило, метод используется в тех случаях, когда квалификация сотрудников и местных IT-специалистов в организации не позволяет использовать сервис-ориентированную архитектуру в своей структуре, а обучить сотрудников слишком дорого;

– метод «параллельного внедрения» [41]. В рамках данного метода организация-заказчик совмещает использование новой архитектуры параллельно устаревшей. При этом процесс работы в организации полностью не замещает стандартные методы работы, а применяется параллельно. Например, перейдя с бумажного варианта работы на работу с системой, которая основана на СОА, где часть сотрудников начинает активно пользоваться новой системой, а часть из них сидит на бумажном, т.е. на устаревшем методе, что приводит к конфликтам и проблемам. В данном случае следует постепенно всех переводить на новую систему, упраздняя старые методы.

Помимо приведенных методов так же могут использоваться другие методы, применяемые при внедрении инновационных технологий [5]. Приведенные методы в свою очередь могут использоваться и при внедрении сервис-ориентированной архитектуры.

Принудительный метод [19, 25]. Предполагает вынужденное внедрение в организации. В этом случае организация в короткие сроки обязана перестроиться под использование сервис-ориентированную архитектуру, изменяя и реорганизовывая всю свою структуру. Процесс является дорогостоящим и нежелательным в социальном плане, может вызывать различные конфликты. Однако основным достоинством данного метода

является превосходство во времени стратегического планирования. Метод используется в условиях резкой нехватки времени, а так же когда природа сопротивления определена, и использование силы не требуется.

Метод адаптивных отклонений [6, 9, 18]. При использовании данного подхода учитываются стратегические изменения, которые осуществляются на протяжении длительного периода. Процесс интеграции СОА системы разделяется на этапы и все происходит постепенно. Для управления процессом выделяется отдельная проектная группа. Данное решение используется при возможности легкого определения опасностей или благоприятных возможностей, а так же предсказываются все дальнейшие действия по внедрению и работе. Метод эффективен при легком предвидении различных опасностей, вызываемых при внедрении этой архитектуры, которые в случае чего не потребуют моментального решения.

Метод при решении кризисных ситуаций. Этот метод крайне необходим, когда другие решения уже не помогают. В этом случае, решается угроза прекращения существования организации. Применение сервис-ориентированной архитектуры в теории может привести к спасению организации, путем сокращения отделов и реорганизации всей структуры.

Метод управления сопротивлением. Этот метод включает в себя сочетания метода адаптивных отклонений и принудительного. Все исходит путем исследования работы организации и принимается решение как лучше всего проводить внедрение СОА, либо в срочном и принудительном порядке, либо постепенно.

Ниже, в таблице 3 приведена краткая характеристика методов внедрения информационных технологий, которые можно использовать во время внедрения сервис-ориентированной архитектуры.

Таблица 3 – Сравнение методов внедрения инновационных технологии сервис-ориентированной архитектуры.

Метод	Условия применения	Преимущества	Недостатки
Принудительный	Большая срочность	Быстрота изменений	Большое сопротивление
Метод адаптивных отклонений	Небольшая срочность	Слабое сопротивление	Медлительность
Кризисные решения	Большая срочность	Спасение организации от развала	Жесткий дефицит времени и риск неудачи
Метод управления сопротивлением	Производится исследование для определения	Возможность выбора решения	Сразу непонятно как действовать

Исходя из полученных данных, можно сделать вывод, что все представленные методы по-своему рабочие, однако, для эффективного внедрения следует использовать либо метод «конкретизирующих документов», либо метод «частичного внедрения», так как они являются более эффективными.

К сожалению, метод «вечного эксперимента» не подходит, потому что в современном мире тянуть резину на бесконечные эксперименты нет никакого смысла. Достаточно рассмотреть на уже готовых примерах и сделать вывод, нужно ли это, или нет.

Метод «отчетного внедрения» не подходит тем, что по сути, этой системы нет вообще, лишь по бумаге, что не подходит для современного решения.

Метод «параллельного внедрения» не подходит, потому что появляется лишняя работа и конфликт между отделами и между структурами, так как часть уже сидит на новой архитектуре, а часть использует еще старую. Из-за этого организация может лишь навредить себе.

Что касается методов внедрения инновационных технологий, что все они отлично подходят под внедрение сервис-ориентированной архитектуры, нужно

лишь перед выбором понять в каком положении находится организация-заказчик и что у них по срокам работы.

Но самым оптимальным является метод управления сопротивлением, так как в нем можно перед началом внедрения рассмотреть сроки и необходимость изменений в структуру организации-заказчика.

В результате написания данной главы в научно-исследовательской работе были рассмотрены существующие методы внедрения, которые помогут внедрять сервис-ориентированную архитектуру.

Также были рассмотрены методы инновационных технологий СОА в организации, применение которых возможно во время создания и организации структуры работы организации-заказчика.

В рамках данной научно-исследовательской работы были определены условия применения представленных методов, рассмотрены преимущества и недостатки некоторых методов.

Так же для примера были использованы все четыре метода внедрения инноваций: принудительный, метод адаптивных отклонений, метод кризисных решений и метод управления сопротивлением. На их основе было проведено сравнение по способам и условиям применения, так же выделены основные преимущества и недостатки и составлена сравнительная таблица.

Выводы к разделу 2

Были рассмотрены методы, которые существуют на данный момент и изучен материал по данной теме. Для начала, выделены основные достоинства и недостатки внедрения СОА системы, и был рассмотрен один из существующих примеров его реализации.

Было проведено сравнение работы предприятия с внедрением СОА системы и без нее. В результате изучения и анализа отобранного материала были проанализированы положительные и отрицательные стороны данного вопроса, а также преимущество и удобство данной системы в организации. На их основе можно сделать вывод, что создание решения и его внедрение будет положительно влиять на процесс создания организации. Это связано с тем, что

благодаря внедрению СОА системы появляется возможность пользоваться полноценно всеми возможностями системы независимо от устройства, главное иметь доступ к интернету.

Также были рассмотрены три программные архитектуры, а именно:

- монолитные;
- сервис-ориентированные;
- микросервисы.

Были рассмотрены более подробно их преимущества и недостатки, на основе полученных данных был приведен вывод о том, что сервис-ориентированная архитектура является более оптимальным решением для организации работы организации-заказчика.

Был смоделирован пример работы одной структуры портала Госуслуг, которая использует СОА и показан пример выхода из строя одного из модулей. Был сделан вывод, что приложения, которые основаны на СОА, хорошо подходят для больших организаций.

Рассмотрены методы внедрения сервис-ориентированной архитектуры и можно сделать вывод, что для эффективного внедрения следует использовать либо метод конкретизирующих документов, либо метод «частичного внедрения», так как они являются более эффективными.

Что касается методов внедрения инновационных технологий, что все они отлично подходят под внедрение сервис-ориентированной архитектуры, но самым оптимальным является метод управления сопротивлением, так как в нем можно перед началом внедрения рассмотреть сроки и необходимость изменений в структуру организации-заказчика.

3 Разработка улучшенной СОА системы и ее апробация

3.1 Создание улучшенной модели системы основанной на СОА

СОА систему можно сравнить с компьютером. При отказе одного из компонентов, не нужно утилизировать весь компьютер, достаточно лишь заменить нерабочий или устаревший элемент на другой и компьютер и дальше будет работать. Также и с системой, основанной на СОА [53, 55, 57]. Если в организации нужно убрать, заменить модуль, или же он вышел из строя, то не надо изменять всю систему, полностью перестраивая под новые обстоятельства, достаточно лишь заменить или исправить нужный модуль [4, 8].

По результатам третьей главы было проведено сравнение существующих архитектур, выявления преимуществ и недостатков системы основанной на сервис-ориентированной архитектуры [52, 62]. По полученным данным было проведено исследование и было принято решение предложить улучшить данную архитектуру, а именно, внести изменения в модули, сделав их более гибкими, тем самым сняв нагрузку на систему [49, 54]. На рисунке 12 показана измененная СОА система.

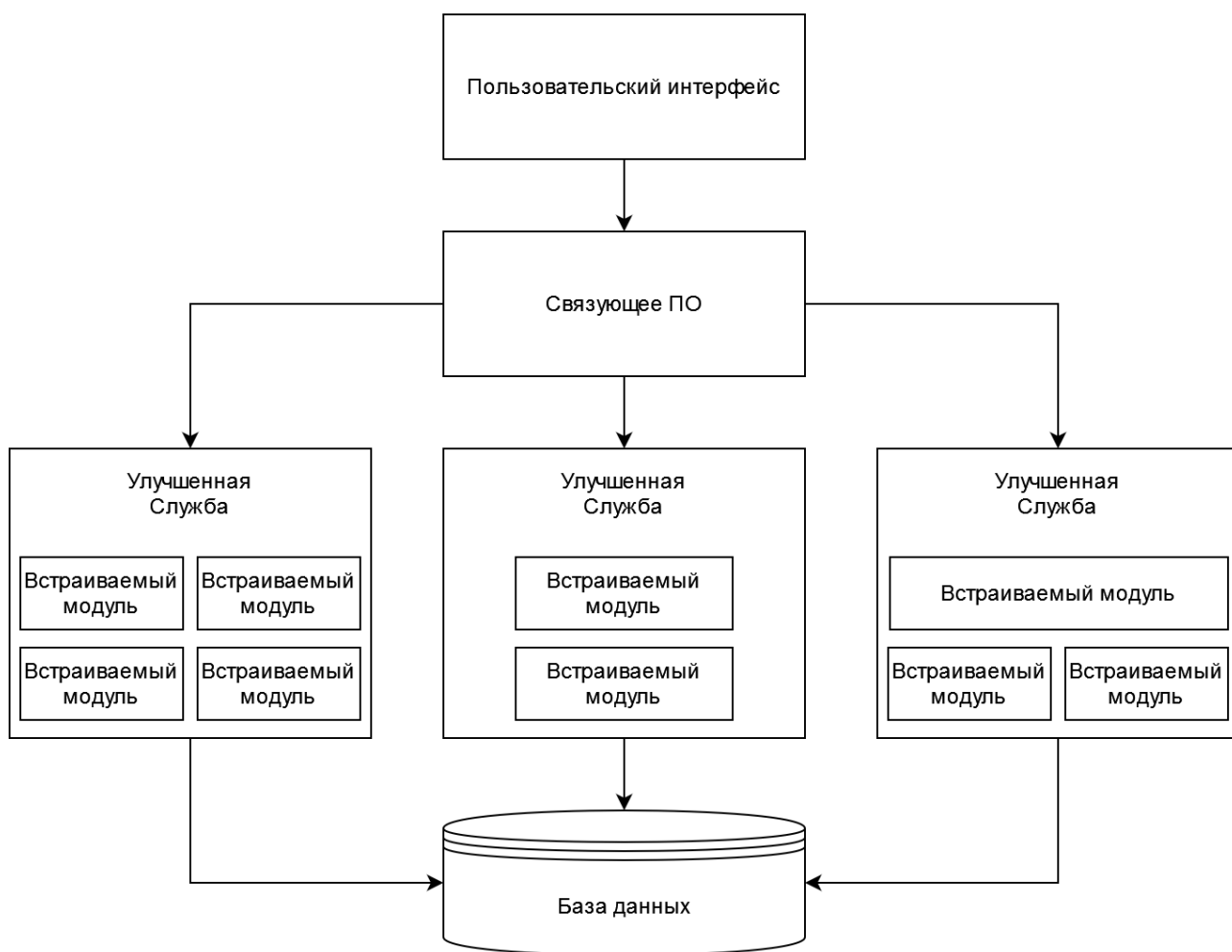


Рисунок 13 – Улучшенная СОА система

Как видно на данном рисунке, внутри каждой службы появились модули, которые можно как добавлять, так и убирать без кардинальных изменений в работе системы. В момент, как в классической сервис-ориентированной архитектуре каждую службу в случае изменений надо изменять полностью [10, 24].

В качестве примера будет произведено моделирование примерного веб-приложения, которое основано на предложенной системе, использующее сервис-ориентированную архитектуру.

На рисунке 14 демонстрируется блок-схема работы модулей внутри службы.

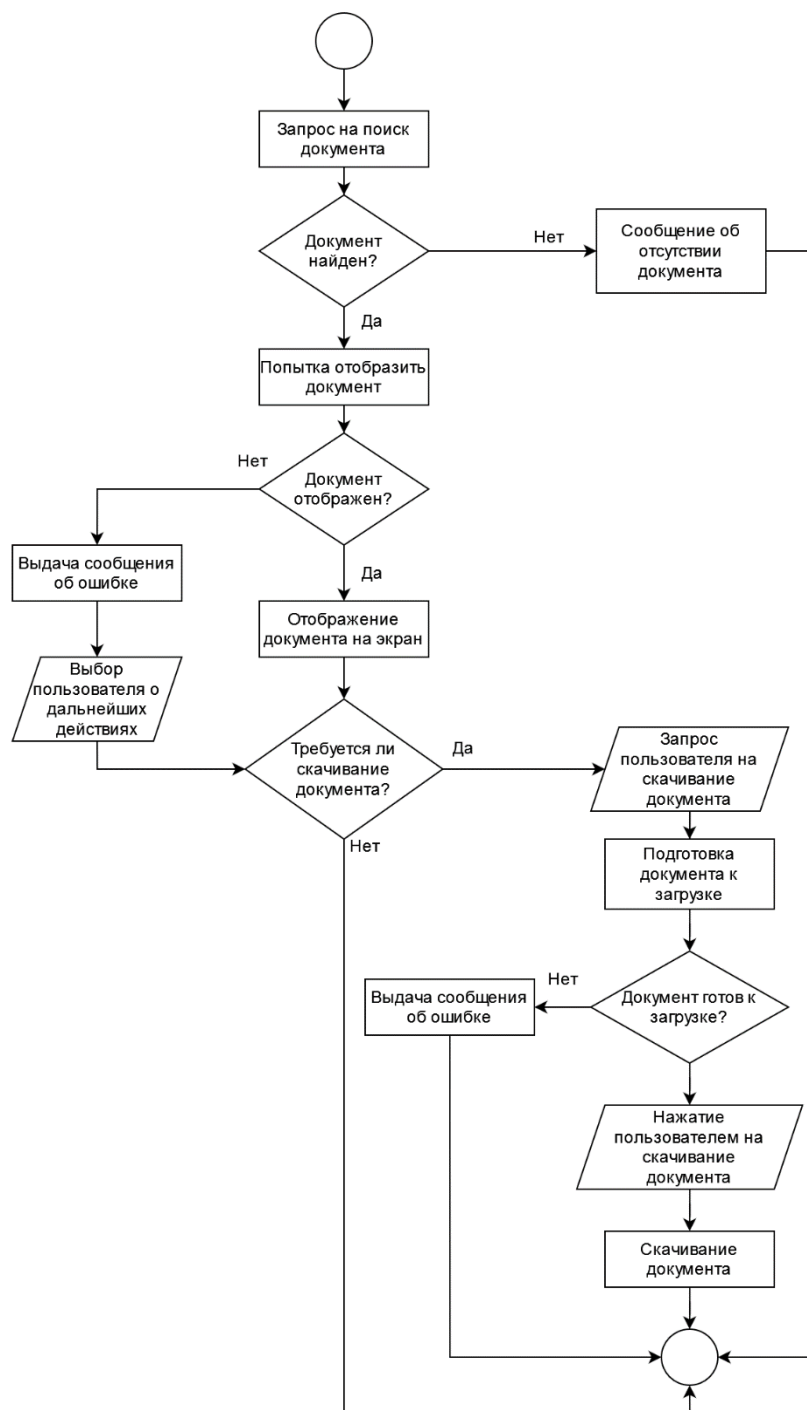


Рисунок 14 – Блок-схема работы модулей внутри службы

На данном рисунке отображена связь и работа одновременно трех модулей, а именно:

- поиск документов;
- модуль отображения документа в режиме онлайн;
- модуль загрузки документа на устройство пользователя.

Из рисунка 14 видно, что пользователь делает запрос на поиск нужного документа, если документ не найден, то нет никакого смысла обращаться к другим модулям в системе, выдается сообщение, что ничего не найдено и завершается работа.

Иначе, система пытается отобразить документ в режиме онлайн путем встроенного модуля отображения документов.

Если второй модуль отображения документов вышел из строя, то выходит соответствующая ошибка и поступает запрос на выбор, нужно ли скачать документ, задействуя третий модуль или же нет. Если нет, то работа с системой завершается.

Если же документ вначале работает исправно, то у пользователя есть возможность просмотреть документ в режиме онлайн и выбрать, нужно ли скачать его на локальное устройство.

Когда пользователь выбирает загрузку на локальный компьютер, то начинает работу третий модуль, загрузку на устройство. Если документ готов к загрузке, то пользователь нажимает на кнопку скачать и происходит загрузка.

Если же модуль неисправный, то выдается окно ошибки и завершается работа с системой.

Из улучшенной СОА системы следует отметить следующие преимущества и недостатки, описанные ниже.

Преимущества:

- в каждом отдельном блоке появляется возможность интегрировать модули в нужные блоки;
- снижаются затраты на интеграцию новых модулей;
- как система, так и блоки продолжают работать при неисправных модулях;
- исчезают ненужные связи между модулями, что делает систему более стабильной и снижает шанс поломок системы.

Также необходимо рассмотреть и недостатки, которые возникают.

Недостатки:

- появляется сложность во внедрении модулей, т.к. нужно учитывать все связи в блоках;
- нужны знания для грамотного планирования работы системы, чтобы все необходимые модули работали без конфликтов;
- увеличивается время разработки и интеграции всех модулей и блоков.

По полученным данным с составленных моделей, следующим этапом следует показать на примере, как это все будет работать в системе. Для начала, на рисунке 15 будет показана система, где все модули работают исправно.

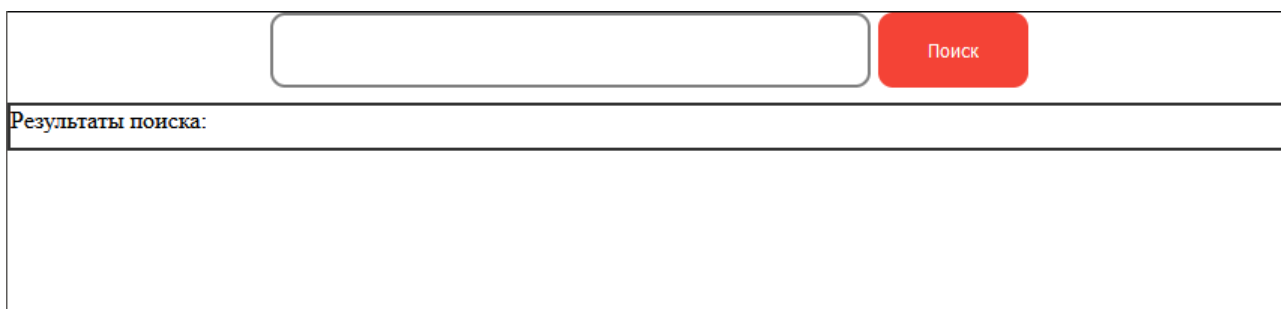


Рисунок 15 – Главное меню для поиска документов

На данной странице пользователь видит окно поиска, в котором может найти документы по нужному запросу, например, поиск дипломной работы.

На 16 рисунке показаны результаты поиска по запросу «диплом».

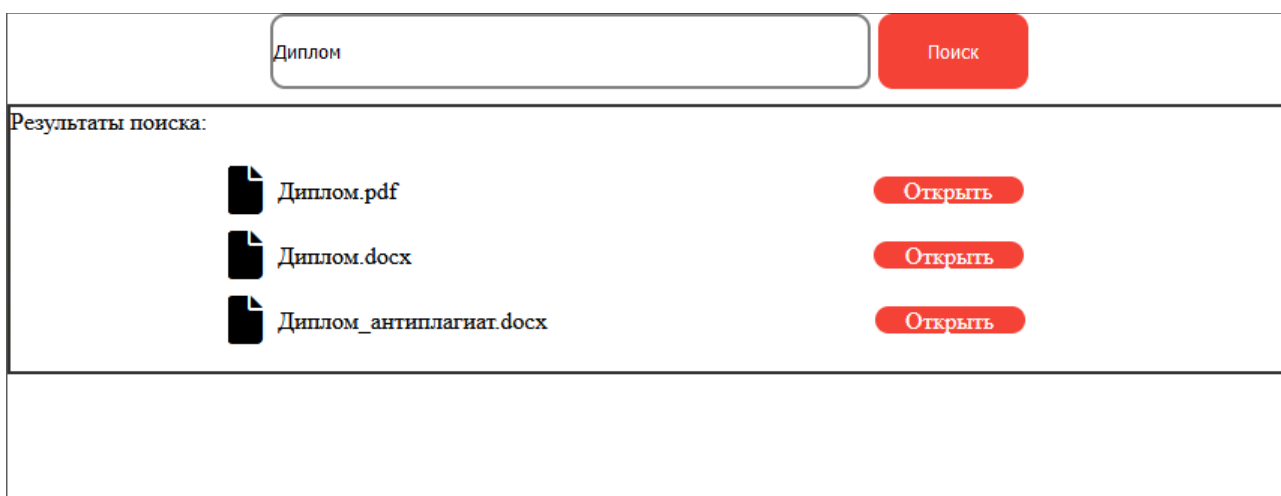


Рисунок 16 – Результат поискового запроса

На данном рисунке мы видим, что в системе имеется 3 файла, подходящие по данному запросу.

Следующим этапом пользователь выбирает, какой из документов ему необходимо просмотреть. Например, первый документ «Диплом.pdf». Пример показан на рисунке 17.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЗВАНИЕ УНИВЕРСИТЕТА»

Название факультета
Название кафедры

ФАМИЛИЯ ИМЯ ОТЧЕСТВО СТУДЕНТА

НАЗВАНИЕ ТЕМЫ ДИПЛОМНОЙ РАБОТЫ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ПО ПРОГРАММЕ ПОДГОТОВКИ БАКАЛАВРИАТА

Выполнил:
студентка ... курса ... формы обучения
направление подготовки 00.00.00 – Название
факультета, профиль – «Название профиля»

Допущена к защите ГЭК
и проверена на объем заимствования:

Заведующий кафедрой
(ученая степень, ученое звание)

(подпись) (И.О. Фамилия)

Научный руководитель
(ученая степень, ученое звание)

(подпись) (И.О. Фамилия)

«__» июня 2020 г.

Город – 2020

Закреть Скачать

Рисунок 17 – Предварительный просмотр документа

Внизу документа находятся две кнопки, а именно:

- закрыть;
- скачать.

Пользователь сам выбирает, что ему дальше делать. Если необходимо скачать файл, то жмет на зеленую кнопку, а если необходимо вернуться на главную страницу, то жмет на красную кнопку «Заккрыть».

Также, если пользователь сделает неправильный запрос, например, допустит ошибку в запросе, то система покажет, что ничего не найдено.

<input type="text" value="Деплом"/> <input type="button" value="Поиск"/>
Результаты поиска: Ничего не найдено.

Рисунок 18 – Результат поиска при запросе с ошибкой

На рисунке 18 показано, что пользователь ввел неправильный запрос «Деплом» и система выдала ответ, что ничего по данному запросу не найдено.

Однако следует также рассмотреть и различные ошибки, которые могут быть связаны с работой основных модулей, которые были внедрены, а именно:

- модуль поиска;
- модуль просмотра файла;
- модуль загрузки файла.

На рисунке 19 демонстрируется проблема с модулем поиска, то есть, пользователь делает запрос «Диплом», но модуль поиска неисправен и выдает ошибку.

<input type="text" value="Диплом"/> <input type="button" value="Поиск"/>
Результаты поиска: Ошибка поиска

Рисунок 19 – Демонстрация неисправного модуля поиска

На данном рисунке видно, что система полностью не дала сбой, пользователь не потерял доступ к системе, однако, найти ничего не может.

Теперь следует рассмотреть вариант, когда поиск работает, но отказал модуль просмотра файла. Данный вариант показан на 20 рисунке.

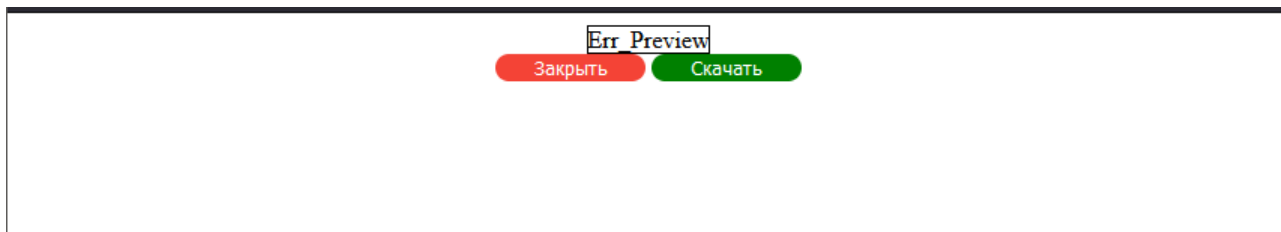


Рисунок 20 – Демонстрация неисправного модуля отображения файла

На данном этапе пользователь видит сообщение об ошибке отображения, но у него все равно открывается окно с данным файлом, чтобы пользователь по желанию мог загрузить на локальное устройство файл и просмотреть его. Или же вернуться на главное меню и продолжить работать в системе.

Последний модуль, который может быть неисправным – это модуль загрузки файла.

И на рисунке 21 показана неисправность модуля загрузки файла на локальное устройство пользователя.

При данной ошибке работает вся система, работает поиск, работает отображение, но пользователь не может скачать данный файл скачать невозможно.

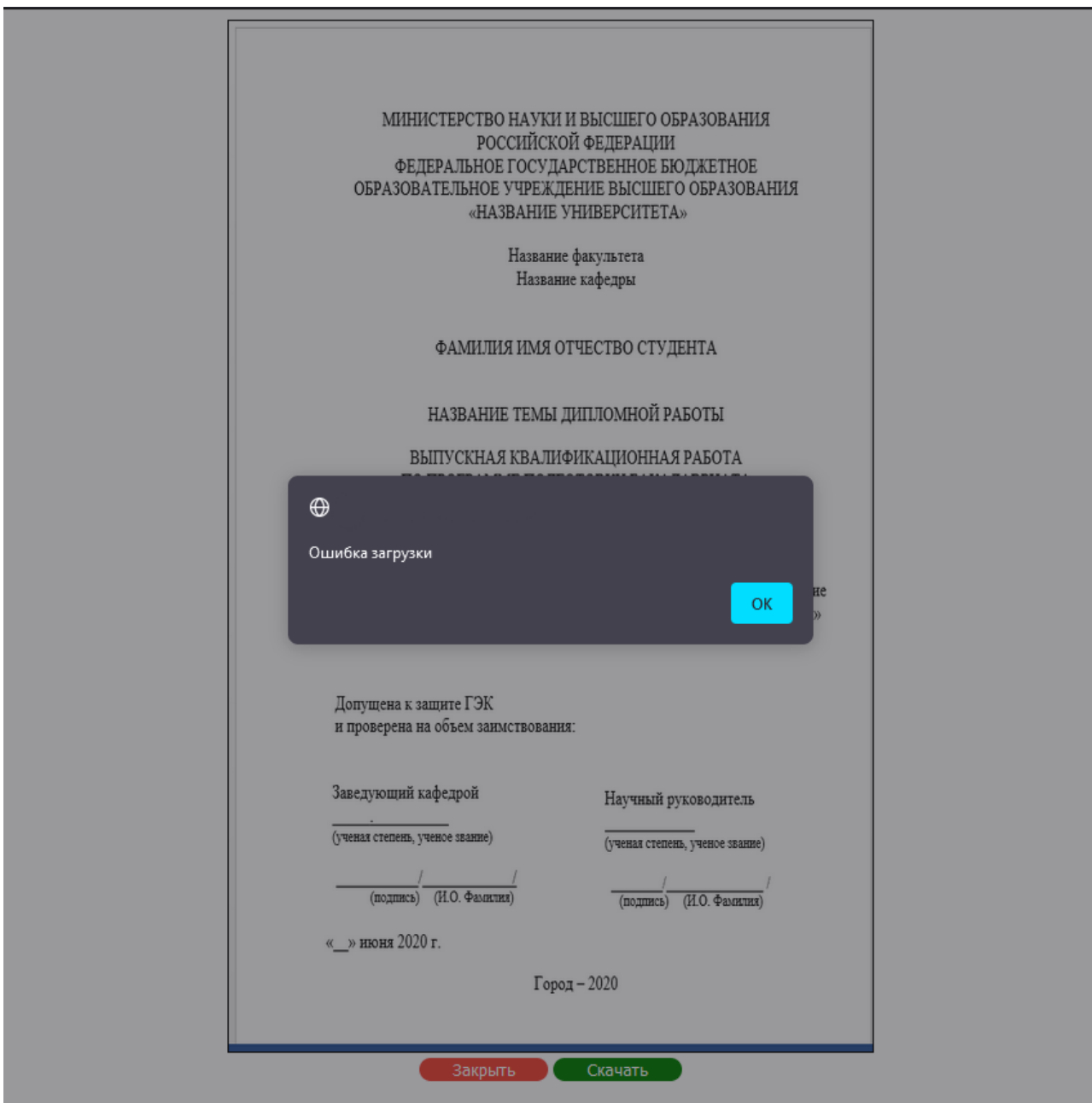


Рисунок 21 – Демонстрация неисправности модуля загрузки

При данной неисправности, когда пользователь нажимает на зеленую кнопку «Скачать», но так как модуль неисправен, то появляется сообщение «Ошибка загрузки» и при нажатии на кнопку «ОК» пользователь возвращается на просмотр документа, если конечно, модуль отображения тоже исправен.

На данном этапе была рассмотрена немного усовершенствованная система, основанная на SOA.

3.2 Сравнительный анализ улучшенной модели СОА с существующими аналогами

Перед апробацией представленной системы следует рассмотреть представленные модели.

Дэвид Вонг из «DevelopPaper» представил две вариации работы системы основанной на СОА [63]. На рисунке 22 демонстрируется 1 вариант работы СОА системы.

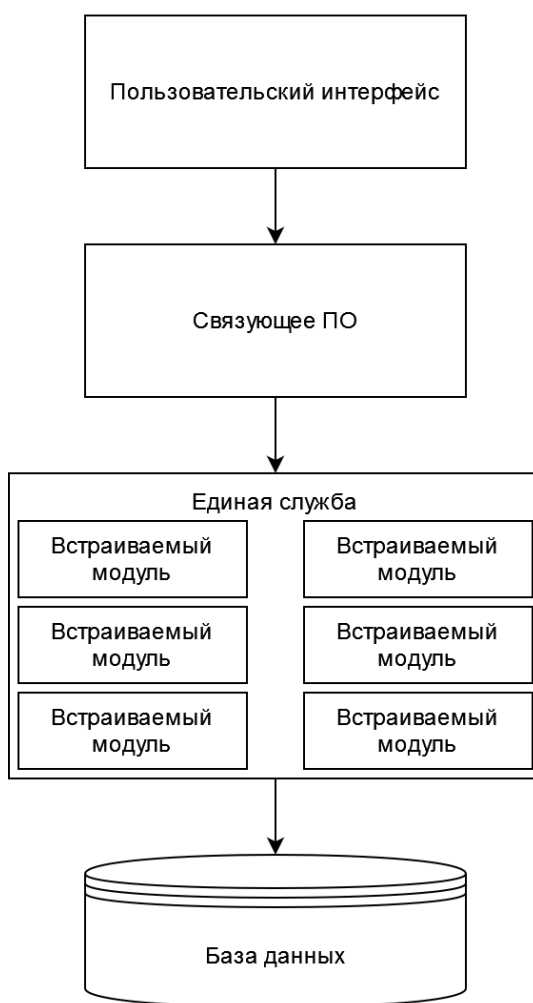


Рисунок 22 – СОА система, использующая единую службу

На данном рисунке показано, что в отличие от классической системы, основанной на СОА, используется лишь единая служба и уже к ней подключаются нужные модули.

Данный вариант крайне похож на монолитную архитектуру, за исключением поддержки модульной службы, однако, если эта служба выйдет из строя, то доступ к базе данных будет закрыт.

Также Дэвид Вонг представил еще один вариант, который крайне похож на смешение SOA системы и микросервисов. На рисунке 23 демонстрируется еще одно решение при создании SOA системы.

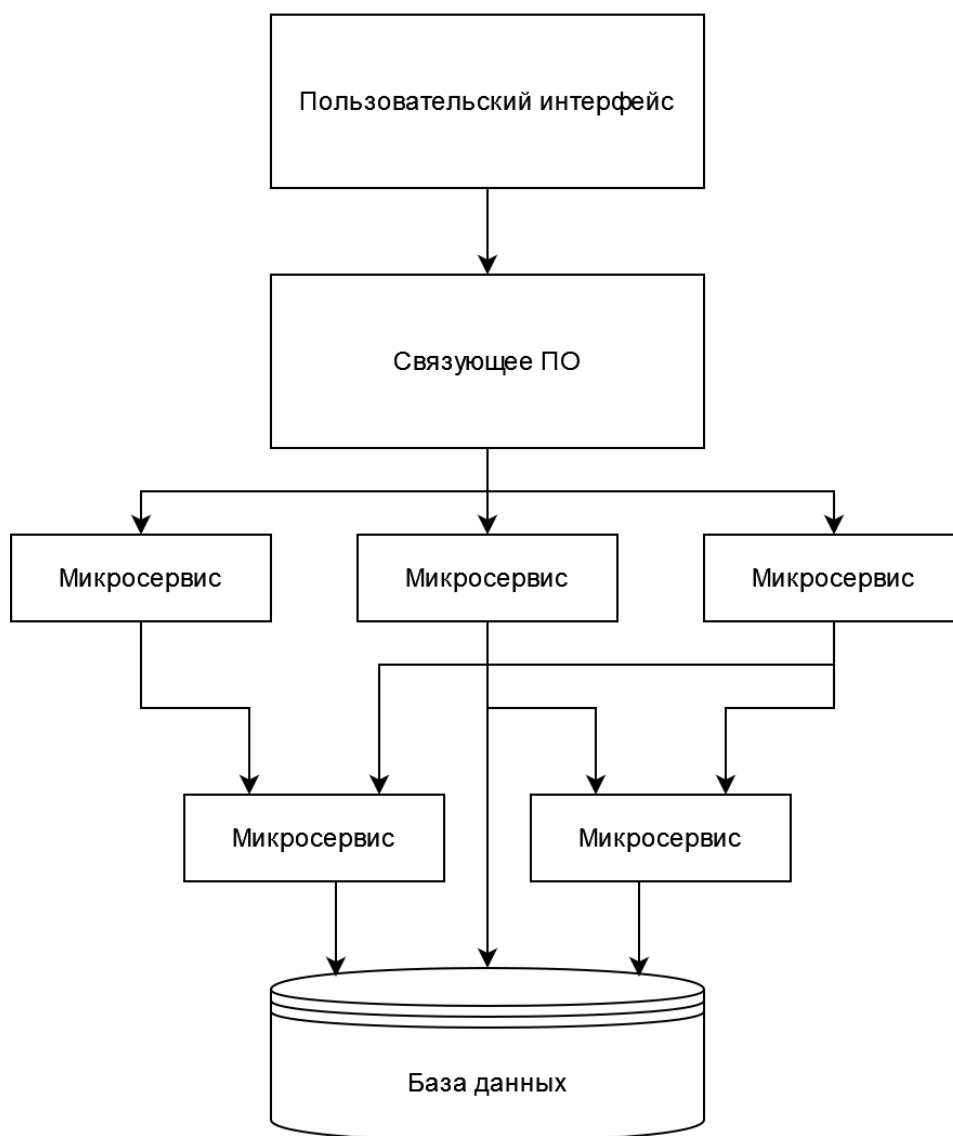


Рисунок 23 – SOA система, где вместо служб с модулями используются микросервисы

Данный вариант, в отличие от предыдущего хорош тем, что нет привязки к единой службе, но также нет возможности подключать модули не изменяя связи между всеми микросервисами.

Также всегда следует помнить, что микросервисы при создании были основаны на SOA системе, поэтому можно сказать, что микросервисы это отдельная ветвь в сервис-ориентированной архитектуры, только вместо служб с модулями используются слабосвязанные микросервисы.

Перед тем, как перейти к апробации улучшенной системы, следует провести сравнение в таблице 4, где будут показаны все преимущества и недостатки систем, основанных на SOA.

Таблица 4 – Сравнение систем основанных на SOA.

	Классическая SOA система	Улучшенная SOA система	Микросервисы	SOA с единой службой	SOA с элементами микросервисов
Добавление отдельных модулей в службы	-	+	-	+	-
Единая база данных	+	+	-	+	+
Служба кардинально не меняется при изменениях	-	+	+	+	+
Независимая работа служб	-	-	+	-	+
Масштабируемость	-	-	+	-	+
Отказоустойчивость	+	+	-	-	-

Из данной таблицы видны все преимущества и недостатки представленных систем. Как видно, классическая SOA система во многом уступает уже существующим аналогам по возможностям, однако, все эти системы основаны на ней.

Представленная улучшенная СОА система показала, что у нее достаточно преимуществ, например, в отличие от классической имеет возможность добавлять отдельные модули в службы и нет необходимости кардинально менять службы при добавлении модулей. Однако у нее есть и минусы, а именно все службы так или иначе зависят друг от друга и, к сожалению, имеет недостаток в масштабируемости системы, так как чем больше будет модулей, тем медленней будет ее работа.

Теперь следует провести апробацию эффективности, путем сравнения систем между собой и решить, что затрачивает меньше всего времени и при каких условиях.

3.3 Апробация улучшенной СОА системы

В качестве апробации системы основанной на сервис-ориентированной архитектуре было принято решение провести сравнение работы с аналогичными системами, основанными на СОА и с системой без ее использования.

Представленная в первой части данной главы улучшенная система не создана для сокращения времени отклика или же улучшения работы для пользователя, она создана для упрощения внедрения и разработки, а также для сокращения лишних расходов, не переделывая службу полностью. Поэтому, хотя она и будет работать немного быстрее классической системы и некоторых аналогов, все же, данный вариант не будет давать колоссального прироста в снижении временных затрат. В качестве доказательства, ниже приведено сравнение улучшенной версии и классической.

Таблица 5 – Сравнение СОА систем, классической и улучшенной.

	Классическая СОА система	Улучшенная СОА система
Время загрузки страницы	0.32 мс	0.38 мс
Время нахождения результатов поиска	0.45 мс	0.38 мс
Время загрузки отображения документа	0.73 мс	0.71 мс
Время загрузки при возврате на главную страницу с просмотра документа	0.21 мс	0.15 мс
Время отклика при неисправной работе поиска (Время до отображения ошибки системой)	1.99 мс	1.70 мс
Время отклика при неисправной работе отображения (Время до отображения ошибки системой)	0.95 мс	0.85 мс
Время отклика при неисправной загрузке файла (Время до отображения ошибки системой)	2.39 мс	2.13 мс

По полученным данным необходимо построить график, который покажет время отклика двух систем. Данный график показан на рисунке 24.

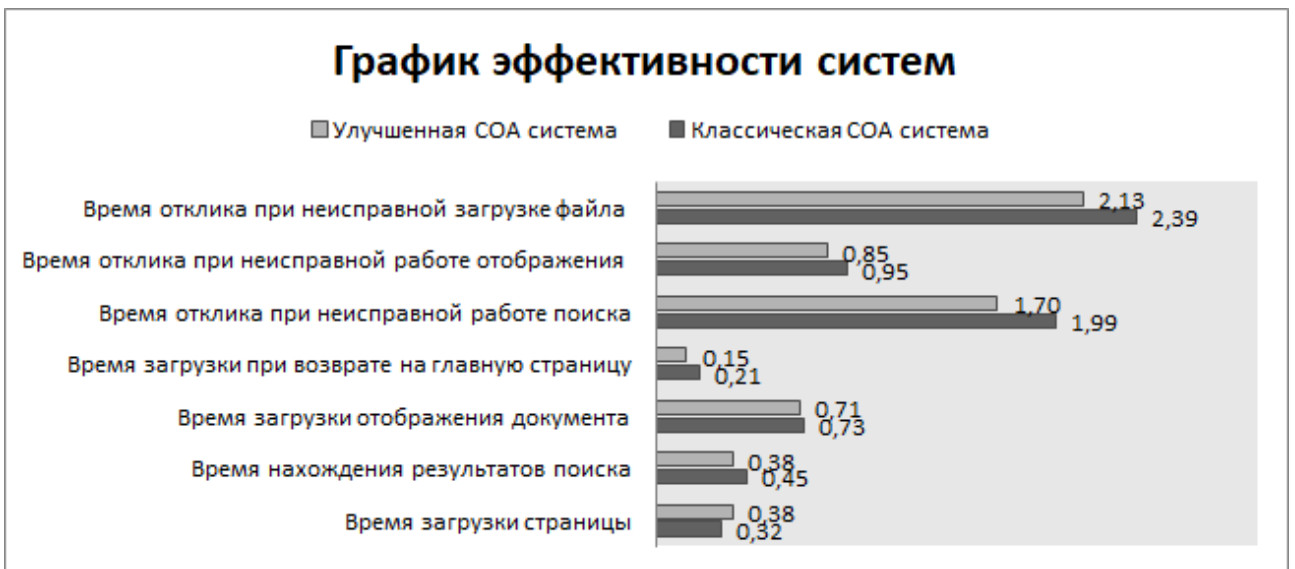


Рисунок 24 – График эффективности СОА систем

Данное сравнение показывает, что обе системы работают примерно на одном уровне и представленные изменения практически не улучшают работу системы в целом, за исключением изменения времени отклика при неисправностях. Улучшенная система показывает более быстрый отклик, примерно в 1,2 раза.

Однако если все же рассмотреть монолитную систему, как пример, то улучшенная система работает намного стабильнее, быстрее и лучше. В качестве примера возьмем тот же поисковый запрос, просмотр документа и его загрузка.

Данное сравнение необходимо для того, чтобы на примере показать колоссальное преимущество системы основанной на улучшенной сервис-ориентированной архитектуры над аналогами, использующими иные архитектуры.

В таблице 6 демонстрируется время ответа на те, или иные запросы пользователя и время отклика системы при наличии неисправностей.

Все результаты записаны в миллисекундах с момента получения запроса до момента ответа системы.

Таблица 6 – Сравнение системы использующую СОА и не использующую.

	Улучшенная СОА система	Система без СОА
Время загрузки страницы	0.38 мс	0.74 мс
Время нахождения результатов поиска	0.38 мс	0.81 мс
Время загрузки отображения документа	0.71 мс	1.24 мс
Время загрузки при возврате на главную страницу с просмотра документа	0.15 мс	0.56 мс
Время отклика при неисправной работе поиска (Время до отображения ошибки системой)	1.70 мс	- (бесконечно/страница не загружается)
Время отклика при неисправной работе отображения (Время до отображения ошибки системой)	0.85 мс	- (бесконечно/страница не загружается)
Время отклика при неисправной загрузке файла (Время до отображения ошибки системой)	2.13 мс	- (бесконечно/страница не загружается)

По полученным результатам теперь необходимо построить график, который покажет наглядно время отклика двух систем. График показан на рисунке 25.

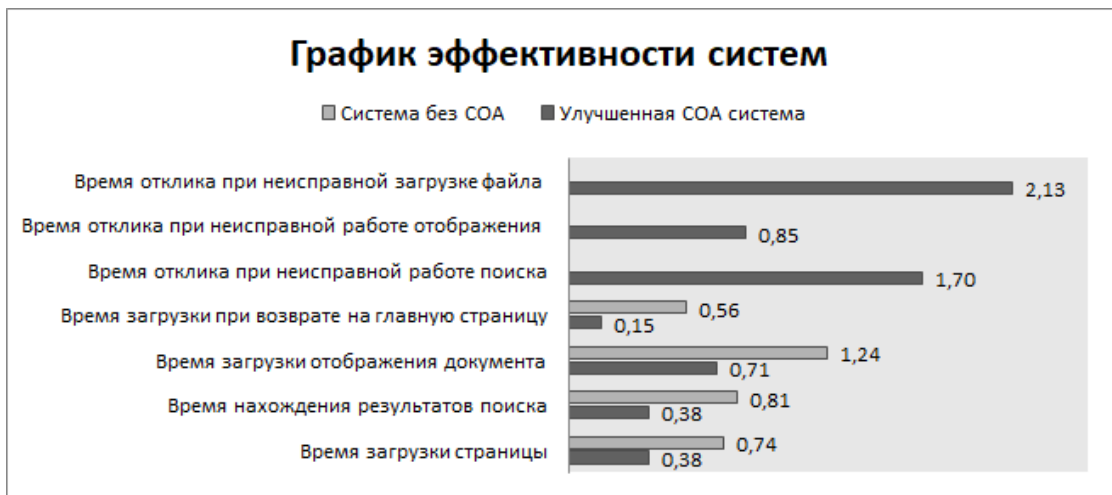


Рисунок 25 – График эффективности системы с использованием сервис-ориентированной архитектуры

По полученным результатам видно сравнение работы систем, из них следует учесть то, что при любой из неисправностей система, которая не использует сервис-ориентированную архитектуру попросту отказывает в работе и отказывает загружать страницу, выдав, например, ошибку 404 или невозможность загрузить страницу. Пример ошибки показан на рисунке 26.

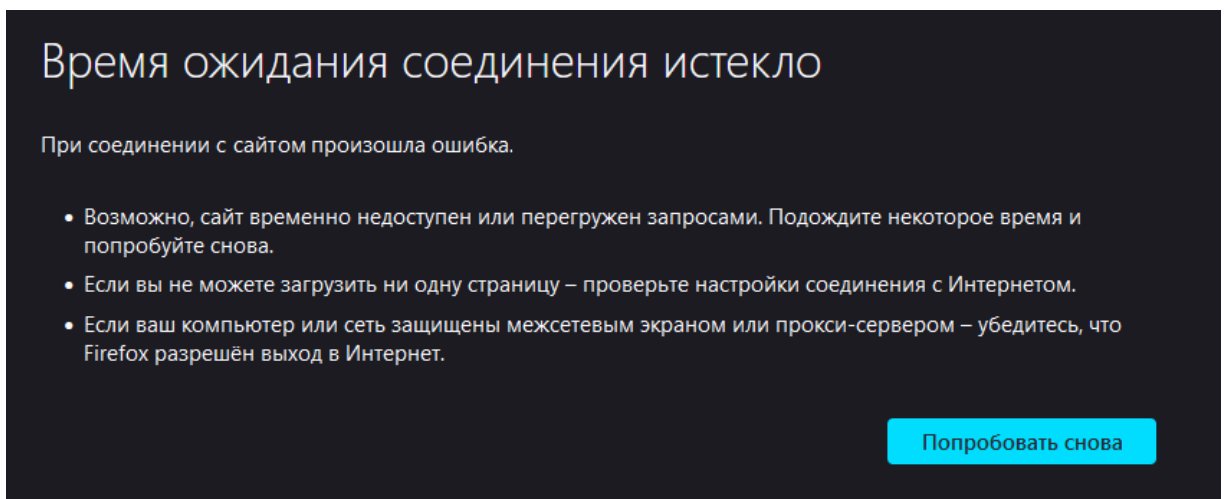


Рисунок 26 – Неисправная работа системы без использования COA

В противоположном случае, система, основанная на COA перед тем, как выдать окно с ошибкой делает неоднократный запрос, попытается открыть страницу с неисправным модулем, где будет ошибка, и пользователь по итогу все равно сможет вернуться назад, не теряя доступ.

Что касается времени отклика, то график прекрасно демонстрирует тот факт, что система, которая использует сервис-ориентированную архитектуру работает в некоторых случаях намного быстрее, чем иная система, не использующая данную архитектуру.

По результатам исследования, SOA система выполнила всю работу примерно за 1,62 миллисекунд, в тот момент, как на обработку всех запросов у системы, не использующей сервис-ориентированную архитектуру было потрачено 3,35 миллисекунд. По полученным данным можно сделать вывод, что использование SOA системы приводит к улучшению работы системы в 2,06 раз.

Последним этапом следует провести апробацию системы с аналогами основанными на сервис-ориентированной архитектуре. Данное сравнение показано на таблице 7.

Таблица 7 – Сравнение различных систем основанных на SOA.

	Классическая SOA система	Улучшенная SOA система	Микросервисы	SOA с единой службой	SOA с элементами микросервисов
Время загрузки страницы	0.32 мс	0.38 мс	0.23 мс	0.40 мс	0.31 мс
Время нахождения результатов поиска	0.45 мс	0.44 мс	0.88 мс	0.50 мс	0.71 мс
Время загрузки отображения документа	0.73 мс	0.71 мс	0.65 мс	0.80 мс	0.69 мс
Время загрузки при возврате на главную страницу	0.21 мс	0.26 мс	0.15 мс	0.33 мс	0.23 мс

Продолжение таблицы 7.

	Классическая СОА система	Улучшенная СОА система	Микросервисы	СОА с единой службой	СОА с элементами микросервисов
Время отклика при неисправной работе поиска	1.99 мс	1.91 мс	- бесконечно/ страница не загружается	2.59 мс	- бесконечно/ страница не загружается
Время отклика при неисправной работе отображения	0.95 мс	0.91 мс	- бесконечно/ страница не загружается	1.75 мс	- бесконечно/ страница не загружается
Время отклика при неисправной загрузке файла	2.39 мс	2.30 мс	- бесконечно/ страница не загружается	3.01 мс	- бесконечно/ страница не загружается

По полученным данным следует построить график эффективности. Данный график демонстрируется на рисунке 27.

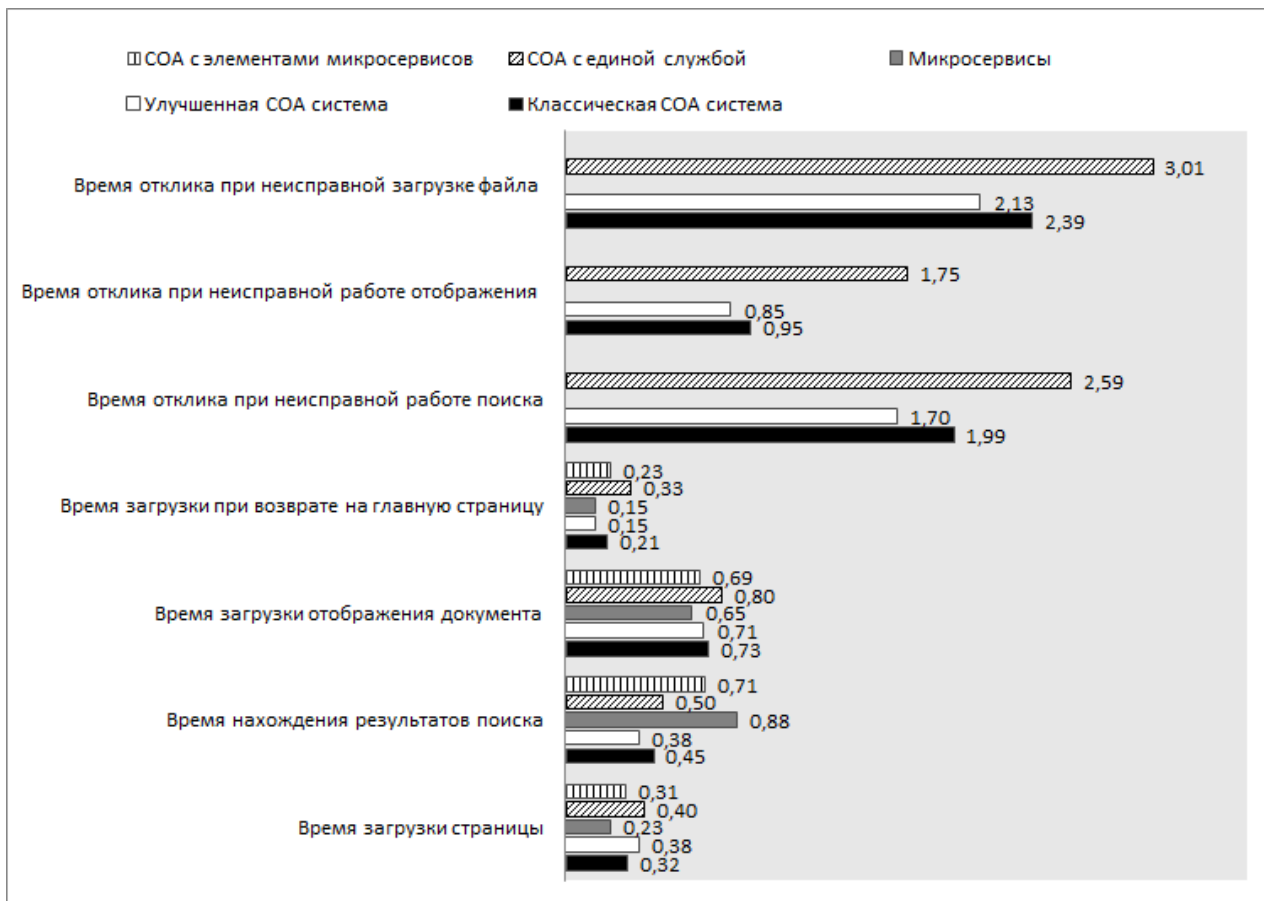


Рисунок 27 – График эффективности систем основанных на COA

По графику видно, что при исправной работе система работает на уровне аналогов, однако, при неисправностях отклик системы заметно меньше, нежели у аналогов. Микросервисы или COA с элементами микросервисов отлично показали себя в исправной работе системы, однако, во время той или иной неисправности перестали отвечать на запросы.

По полученным данным с таблицы и графика можно провести расчеты временных затрат для систем, которые продолжают работать при неисправностях. Результаты таблицы 5 показали, что улучшенная COA система работает на 1,2 раза быстрее классической, поэтому сравнить нужно лишь с COA использующей единую службу. COA с единой службой затрачивает примерно 9,38 миллисекунд, в тот момент, как улучшенная COA система всего 6,30 миллисекунд, что означает, что данная система работает в 1,5 раза быстрее.

Рассмотрев все результаты можно прийти к выводу, что использование улучшенной сервис-ориентированной архитектуры при выборе разработки системы является отличным решением, так как:

- при отказе модулей система продолжает работать;
- пользователь не теряет доступ к системе при неисправностях;
- можно добавлять или убирать модули без кардинальных изменений в структуре системы;
- время отклика меньше, чем у аналогов.

Поставленная гипотеза вначале всего исследования гласила, что использование системы, использующую сервис-ориентированную архитектуру является хорошим выбором и данное исследование это подтвердило, а значит, гипотеза доказана.

Выводы к разделу 3

В данной главе была рассмотрена немного улучшенная система, основанная на сервис-ориентированной архитектуре.

Были выделены преимущества, например:

- появляется возможность интегрировать модули в нужные блоки;
- снижаются затраты на интеграцию новых модулей;
- как система, так и блоки продолжают работать при неисправных модулях;
- исчезают ненужные связи между модулями, что делает систему более стабильной и снижает шанс поломок системы.

Однако появляются и недостатки, а именно:

- трудность во внедрении модулей, т.к. нужно производить учет всех связей в блоках;
- нужны знания для грамотного планирования работы системы, чтобы все необходимые модули работали без конфликтов;

- увеличивается время разработки и интеграции всех модулей и блоков.

На основе данной идеи была построена блок-схема, которая показывает пример работы службы, содержащей три модуля, а именно:

- модуль поиска;
- модуль отображения;
- модуль загрузки на устройство.

По полученным данным, была построена тестовая система, которая продемонстрировала работу системы, как при исправных модулях и при успешном результате, так и при неправильном запросе.

Также было необходимо провести моделирование работы системы при неисправности каждого из модулей, и продемонстрированы ошибки работы.

По полученным результатам было продемонстрировано главное преимущество всей системы в целом, в отличие от аналогов, например, монолитной системы.

Во второй части главы были рассмотрены два варианта работы СОА системы представленными Дэвид Вонгом, и на основе полученных данных была построена таблица сравнения преимуществ и недостатков СОА систем между собой.

В третьей части данной главы было проведено сравнение работы представленной СОА системы и классической, где было показано, что они работают примерно на одном уровне и улучшенная версия создана не для улучшения работы системы, затрачивая меньше времени, а создана для упрощения внедрения и сопровождения системы.

Последним этапом была проведена апробация всех систем основанных на СОА, построен на основании таблицы график и подведены итоги работы, показывающие, что работа улучшенной СОА системы в разы лучше аналогов.

Рассмотрев и получив результаты тестирования, была доказана поставленная гипотеза, а значит, что данная работа выполнена успешно.

Заключение

В ходе написания магистерской диссертации на тему «Методы и средства моделирования систем с сервисно-ориентированной архитектурой» была проанализирована существующая литература, определены основные положительные и отрицательные стороны использования сервис-ориентированной системы, а так же сформированы цели и задачи.

На основе этих данных была поставлена проблема для дальнейшего исследования. Также была выдвинута гипотеза о том, что использование системы основанной на сервис-ориентированной архитектуре является отличным выбором, так как эта архитектура имеет ряд преимуществ. У заказчиков при выборе системы в пользу СОА появится возможность сократить затраты на улучшение, так как, данную систему не нужно каждый раз пересоздавать заново, достаточно лишь изменить модули или же службы, без ущерба всей системе.

Во второй части диссертации были рассмотрены организации, в которых данная система уже внедрена, были изучены их преимущества и недостатки. По полученным данным было произведено сравнение организаций, которые используют СОА систему и которые не используют.

Следующим этапом были рассмотрены и другие архитектуры, которые могут использовать заказчик при разработке системы в организации, среди них были выделены три основные:

- монолитная архитектура;
- сервис-ориентированная архитектура;
- микросервисы.

Так как монолитная архитектура является устаревшей и имеющая достаточно недочетов, то не было никакого смысла ее рассматривать и сравнивать. Поэтому, СОА и микросервисы были сравнены между собой, отмечены их сильные и слабые стороны. Из полученного исследования было

принято решение, что СОА система подходит более для современных нужд, нежели микросервисы.

Заключительным этапом написания второй главы были рассмотрены существующие методы внедрения и выбран самый подходящий и универсальный метод.

В заключительной главе данной работы была улучшена сервис-ориентированная система, проведены тесты работы системы при внедрении следующих модулей, а именно:

- модуль поиска;
- модуль отображения;
- модуль загрузки.

Были произведены тестирования работы системы, как при исправных модулях, так и при наличии отказа модулей. По итогам тестирования было продемонстрировано, что данная система хороша тем, что при отказе того или иного модуля пользователь все равно может работать в системе, хоть и с некими ограничениями. Также, улучшенная версия СОА системы является неплохим решением, т.к. в отличие от классической версии, ее проще модифицировать, так как нет необходимости изменять всю службу полностью, а достаточно лишь подключить к ней и связать необходимые модули, в тот момент, как в классической версии нужно было полностью изменять необходимую службу.

По результатам апробации СОА системы, системы без ее использования и аналогах построенных также на основе классической СОА показало, что применение системы основанной на сервис-ориентированной архитектуре дает прирост в работе системы примерно в полтора-два раза и не лишает доступа к системе, а лишь ограничивает его. Была доказана гипотеза того, что использование СОА является отличным решением, при выборе архитектуры.

Список используемых источников

1. А.Е. Сатунина, Л.А. Сыроева. Анализ моделей перехода к сервис-ориентированной архитектуре информационной системы вуза. – Электронный ресурс – Режим доступа: <https://cyberleninka.ru/article/n/analiz-modeley-perehoda-k-servis-orientirovannoy-arhitekture-informatsionnoy-sistemy-vuza-1/viewer> (дата обращения 07.01.2021).
2. Анализ моделей управления сервис-ориентированной информационной системы – Электронный ресурс – Режим доступа: <https://cyberleninka.ru/article/n/analiz-modeley-upravleniya-servis-orientirovannoy-informatsionnoy-sistemoy-1/viewer> (дата обращения 29.12.2020).
3. Антиинновационные приемы [Электронный ресурс]. – Режим доступа: <https://megalektsii.ru/s23198t1.html> (Дата обращения: 31.12.2021).
4. Введение в сервис-ориентированную архитектуру – Электронный ресурс – Режим доступа: <http://kacit.ru/upload/iblock/cbc/cbc9bcc2c17f3aa9a017e8fad70b4c4c.pdf> (дата обращения 09.01.2022).
5. Внедрение инноваций в компании: как решать проблемы, не создавая новых [Электронный ресурс]. – Режим доступа: <https://viafuture.ru/sozdanie-startapa/vnedrenie-innovatsij> (Дата обращения: 08.01.2022).
6. Внедрение продуктовых инноваций – как бизнес работает с инновациями [Электронный ресурс]. – Режим доступа: <https://leadstartup.ru/db/product-innovation> (Дата обращения: 09.01.2022).
7. Дмитрий Гулько. Мастер-данные: найден кратчайший путь к SOA-Электронный ресурс – Режим доступа: <http://www.insapov.ru/master-data-shortest-way-to-soa.html> (дата обращения 12.05.2021).
8. И. Ю. Шполянская. Система поддержки принятия решений для проектирования сервис-ориентированной архитектуры. – Электронный ресурс – Режим доступа: <http://siit.ugatu.su/index.php/journal/article/download/20/36> (дата обращения 01.02.2022).

9. Инновационные технологии, их виды и методы внедрения [Электронный ресурс]. – Режим доступа: https://studbooks.net/1533045/marketing/innovatsionnye_tehnologii_vidy_metody_vnedreniya (Дата обращения: 09.01.2022).

10. Использование сервис-ориентированной архитектуры (SOA) для построения сложных веб проектов [Электронный ресурс]. – Режим доступа: <https://moscowpython.ru/meetup/21/soa/> (Дата обращения: 03.04.2022).

11. ИТ-АРХИТЕКТУРА ОРГАНИЗАЦИИ: SOA И УПРАВЛЕНИЕ ИНФОРМАЦИОННЫМИ ПОТОКАМИ, ЧАСТЬ 1 – Электронный ресурс – Режим доступа: <https://12news.ru/doc1793.html?ref=erpnews> (дата обращения 03.01.2022).

12. Кому нужна SOA в России – Электронный ресурс – Режим доступа: https://www.cnews.ru/articles/komu_nuzhna_soa_v_rossii (дата обращения 01.05.2021).

13. Коптелов Андрей Константинович. СЕРВИС-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА (SOA) «ЗА» И «ПРОТИВ» – Электронный ресурс – Режим доступа: <http://koptelov.info/publikatsii/soa/> (дата обращения 02.01.2021).

14. Коптелов Андрей Константинович. СЕРВИС-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА (SOA): ОПЫТ ВНЕДРЕНИЯ – Электронный ресурс – Режим доступа: <http://koptelov.info/publikatsii/service-oriented-architecture/> (дата обращения 02.01.2021).

15. Лучшая архитектура для MVP: монолит, SOA, микросервисы или бессерверная?.. Часть 1 [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/otus/blog/476024/> (Дата обращения: 08.12.2021).

16. Лучшая архитектура для MVP: монолит, SOA, микросервисы или бессерверная?.. Часть 2 [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/otus/blog/477930/> (Дата обращения: 08.12.2021).

17. Мамду Ибрахим и Гил Лонг. Часть 1. Взаимодействие SOA и EA – Электронный ресурс – Режим доступа:

<https://www.ibm.com/developerworks/ru/library/ws-soa-enterprise1/index.html> (дата обращения 05.01.2021).

18. Методы внедрения инноваций в организации [Электронный ресурс]. – Режим доступа: <https://studfile.net/preview/3873055/page:4/> (Дата обращения: 09.01.2022).

19. Методы внедрения инноваций в организации [Электронный ресурс]. – Режим доступа: https://ozlib.com/866688/ekonomika/metody_vnedreniya_innovatsiy_organizatsii (Дата обращения: 09.01.2022).

20. Методы эффективного внедрения инноваций в организацию [Электронный ресурс]. – Режим доступа: <https://apni.ru/article/510-metodi-effektivnogo-vnedreniya-innovatsij> (Дата обращения: 05.01.2022).

21. Нововведения в организациях [Электронный ресурс] – Режим доступа: <https://pandia.ru/text/78/436/12308.php> (Дата обращения: 05.01.2022)

22. Палей Т.Ф. Инновационный менеджмент / Т.Ф. Палей. – К. : Фолиантъ, 2011. – 173 с. (Дата обращения: 31.12.2021).

23. Петр Васильевич Терещенко, Виктор Андреевич Астапчук. Архитектура корпоративных информационных систем. – Новосибирск : Новосибирский государственный технический университет, 2015. – 75 с. : ил. ; Библиогр.: - ISBN 978-5-7782-2698-2. – Текст : непосредственный.

24. Построение оптимальной архитектуры и последующая разработка информационных систем. – Электронный ресурс – Режим доступа: <https://intuit.ru/studies/courses/3611/853/lecture/32047?page=3> (дата обращения 07.04.2022).

25. Принудительный метод проведения изменений [Электронный ресурс] – Режим доступа: <https://econ.wikireading.ru/45177> (Дата обращения: 09.01.2022).

26. Проблемы реализации инноваций [Электронный ресурс] – Режим доступа: https://studopedia.ru/17_21718_problemi-realizatsii-innovatsiy.html (Дата обращения 06.01.2022).

27. Разработка моделей и методов мониторинга сервис-ориентированных информационных систем – Электронный ресурс – Режим доступа: <https://www.dissercat.com/content/razrabotka-modelei-i-metodov-monitoringa-servis-orientirovannykh-informatsionnykh-sistem/read> (дата обращения 29.12.2020).

28. Сервисно-ориентированная архитектура ИС ВШМ СПбГУ и проблемы стохастической оптимизации системы – Электронный ресурс – Режим доступа: https://www.math.spbu.ru/user/gran/sb3/gran_sheron.pdf (дата обращения 01.01.2021).

29. Сервис-ориентированная архитектура (SOA). – Электронный ресурс – Режим доступа: <https://habr.com/ru/company/mailru/blog/342526/> (дата обращения 07.01.2022).

30. Сервис-ориентированная архитектура в образовании – Электронный ресурс – Режим доступа: <https://davaiknam.ru/text/servis-orientirovannaya-arhitektura-v-obrazovanii> (дата обращения 03.05.2021).

31. Сервис-ориентированная архитектура. – Электронный ресурс – Режим доступа: <http://iso.ru/ru/press-center/journal/2046.phtml> (дата обращения 06.01.2022).

32. Сервис-ориентированная архитектура. – Электронный ресурс – Режим доступа: <http://citforum.ru/internet/webservice/soa/> (дата обращения 07.01.2022).

33. Сервис-ориентированный подход к построению и функционированию корпоративной информационной системы – Электронный ресурс – Режим доступа: <https://www.science-education.ru/ru/article/view?id=1295> (дата обращения 31.12.2020).

34. Синтез сервис-ориентированной архитектуры информационной системы – Электронный ресурс – Режим доступа: http://www.rusnauka.com/11_EISN_2010/Informatica/64250.doc.htm (дата обращения 03.01.2021).

35. Система межведомственного электронного взаимодействия (СМЭВ) описание и функции системы. Информационные технологии [Электронный ресурс]. – Режим доступа: <https://ruud.ru/it/11448-sistema-mezhvedomstvennogo-elektronnogo-vzaimodejstviya-smev-opisanie-i-funkcii-sistemy-informacionnye-tehnologii/> (Дата обращения: 24.12.2021).
36. СОА (рынок России) [Электронный ресурс]. – Режим доступа: [https://www.tadviser.ru/index.php/Статья:SOA_\(рынок_России\)](https://www.tadviser.ru/index.php/Статья:SOA_(рынок_России)) (Дата обращения: 20.11.2021).
37. СОА (сервис-ориентированная архитектура) [Электронный ресурс]. – Режим доступа: <https://www.ibm.com/ru-ru/cloud/learn/soa> (Дата обращения: 10.01.2022).
38. СОА как архитектурная основа виртуального университета образования – Электронный ресурс – Режим доступа: http://www.aselibrary.ru/press_center/journal/irr/2008/number_5/number_5_3/number_5_3874/ (дата обращения 05.05.2021).
39. СОА против микросервисов [Электронный ресурс]. – Режим доступа: <https://coderlessons.com/tutorials/veb-razrabotka/arkhitektura-veb-servisov/11-soa-protiv-mikroservisov> (Дата обращения: 01.12.2021).
40. Сопротивление изменениям и методы их устранения [Электронный ресурс]. – Режим доступа: <https://zavantag.com/docs/1421/index-737292.html?page=6> (Дата обращения 01.01.2022).
41. Сопротивление инновационным изменениям [Электронный ресурс]. – Режим доступа: https://studme.org/1482111122127/menedzhment/soprotivlenie_organizatsionnym_izmeneniyam (Дата обращения: 07.01.2022).
42. Стереотипы сопротивления изменениям [Электронный ресурс] – Режим доступа: https://studopedia.su/10_166488_stereotipi-soprotivleniya-izmeneniyam.html (Дата обращения 01.01.2022).
43. Сысоев Александр Сергеевич. Разработка моделей и методов мониторинга сервис-ориентированных информационных систем –

Электронный ресурс – Режим доступа: <http://www.dislib.ru/tehnicheskie/15966-1-razrabotka-modeley-metodov-monitoringa-servis-orientirovannih-informacionnih-sistem.php> (дата обращения 07.04.2021).

44. Технологии выработки решений [Электронный ресурс] – Режим доступа: <https://monographies.ru/en/book/section?id=2381> (Дата обращения 03.01.2022).

45. Что такое SOA (сервис ориентированная архитектура)? – Электронный ресурс – Режим доступа: <https://www.livebusiness.ru/tags/soa/> (дата обращения 07.01.2021).

46. Что такое SOA? системы – Электронный ресурс – Режим доступа: <http://www.fincosoft.ru/SOA> (дата обращения 01.01.2021).

47. ШЕСТЬ ОСТРЫХ ВОПРОСОВ К SOA. – Электронный ресурс – Режим доступа: <https://12news.ru/doc2584.html> (дата обращения 08.01.2021).

48. A machine learning approach for performance-oriented decision support in service-oriented architectures – Электронный ресурс – Режим доступа: <https://link.springer.com/article/10.1007/s10844-020-00617-6> (дата обращения 27.11.2020).

49. A Service Oriented Information System: A Model Driven Approach – Электронный ресурс – Режим доступа: <https://ieeexplore.ieee.org/document/6395131> (дата обращения 05.03.2022).

50. Application of system dynamics for analysis of performance of manufacturing systems – Электронный ресурс – Режим доступа: <https://www.sciencedirect.com/science/article/abs/pii/S0278612519300895> (дата обращения 27.11.2020).

51. Business Analytics-Based Enterprise Information Systems – Электронный ресурс – Режим доступа: https://www.researchgate.net/publication/306072794_Business_Analytics-Based_Enterprise_Information_Systems (дата обращения 27.11.2020).

52. Design and Implementation of a Service-oriented Information System Architecture Based on a Case Study – Электронный ресурс – Режим доступа: <https://www.grin.com/document/71427> (дата обращения 05.03.2022).

53. Exposito Ernesto, Diop Codé. Smart SOA Platforms in Cloud Computing Architectures | Exposito Ernesto, Diop Codé. – USA : John Wiley & Sons Limited, 2014. – 224 с. : ил. ; Библиогр.: - ISBN 978-1-848-21584-9.

54. Hugo Estrada, Alicia Martinez, Luis C. Santillan and Joaquín Perez. A New Service-Based Approach for Enterprise Modeling – Электронный ресурс – Режим доступа: <http://www.scielo.org.mx/pdf/cys/v17n4/v17n4a14.pdf> (дата обращения 05.03.2022).

55. Michael Havey. SOA Cookbook – UK : Packt Publishing, 2008. – 268 с. :ил. ; Библиогр.: - ISBN 9781847195487.

56. Nicolas Arni-Bloch and Jolita Ralyté. Service-Oriented Information Systems Engineering: A Situation-Driven Approach for Service Integration – Электронный ресурс – Режим доступа: https://link.springer.com/chapter/10.1007/978-3-540-69534-9_10 (дата обращения 05.01.2021).

57. Service Oriented Architecture (SOA) – Электронный ресурс – Режим доступа: <http://www.institutio.ru/soa> (дата обращения 01.03.2022).

58. Service-driven content delivery: Do more for your patrons through Web services Marshall Breeding Director for Innovative Technology and Research Vanderbilt – Электронный ресурс – Режим доступа: <https://slideplayer.com/slide/4829527/> (дата обращения 25.05.2021).

59. Service-oriented architecture Сервис-ориентированная архитектура [Электронный ресурс]. – Режим доступа: <https://www.tadviser.ru/index.php/COA> (Дата обращения: 20.11.2021)

60. Service-oriented_architecture [Электронный ресурс]. – Режим доступа: https://www.hmoob.in/wiki/Service-oriented_architecture (Дата обращения: 09.01.2022)

61. SOA vs. Microservices [Электронный ресурс]. – Режим доступа: <https://www.integrate.io/blog/soa-vs-microservices/> (Дата обращения: 15.12.2021)
62. Thomas Usländer. Service-oriented design of environmental information systems – Электронный ресурс – Режим доступа: <https://publikationen.bibliothek.kit.edu/1000016721/1317767> (дата обращения 05.03.2022).
63. Understand the relationship between Service Oriented Architecture (SOA) and micro services [Электронный ресурс]. – Режим доступа: <https://developpaper.com/understand-the-relationship-between-service-oriented-architecture-soa-and-micro-services/> (Дата обращения: 29.03.2022)
64. What Is Service-Oriented Architecture? [Электронный ресурс]. – Режим доступа: <https://medium.com/@SoftwareDevelopmentCommunity/what-is-service-oriented-architecture-fa894d11a7ec> (Дата обращения: 25.11.2021)