# Введение

**Мировой и российский опыт говорит о том, что применение современных информационных решений позволяет значительно повысить эффективность работы государственных структур, обеспечить прозрачность процессов управления.**

**Наиболее ответственным разделом государственного управления является управление финансами, в котором, в свою очередь, первое место занимает комплекс проблем, связанных с бюджетом – его формированием и утверждением, исполнением, анализом и контролем финансовых потоков. Весь этот комплекс действий, регламентированный законодательством РФ и упорядоченный во времени, принято называть бюджетным процессом. Одной из особенностей бюджетного процесса является то, что он регулируется по одним составляющим федеральным законодательством, нормативными и инструктивными документами Министерства финансов РФ, по другим – законодательством субъектов РФ и нормативными актами муниципалитетов, под которые, как правило, инструктивных материалов создается крайне мало.**

**Возможно, как раз поэтому автоматизация бюджетного процесса силами специализированных ИТ-компаний охватывала долгое время именно процессы исполнения бюджета в регионах.**

**Однако специалисты финансовых органов нуждаются в комплексной автоматизации бюджетного процесса. И, несмотря на то что на рынке систем бюджетирования для предприятий имеется целый ряд неплохо зарекомендовавших себя продуктов, бюджетный процесс в региональных финансовых органах в принципе другой, и требует специальных методов и инструментов.**

**Финансовое управление администрации Новоегорлыкского сельского поселения поставило перед информационным отделом задачу автоматизации процесса планирования бюджета, определив ее как создание механизма, обеспечивающего многовариантность расчетов проекта бюджета, а также оперативность и точность расчетов параметров бюджета с формированием документов.**

**Таким образом, задача была поставлена сразу достаточно широко – речь идет о создании централизованной системы управления бюджетным процессом, предоставляющей сотрудникам финансового управления оперативный распределенный доступ к бюджетным данным, возможность сопоставления этих данных за различные временные периоды и по разным объектам. При этом, система должна позволять оперативно вносить изменения и обеспечивать одновременную работу большого количества пользователей.**

# Разработка требований к программному обеспечению

## Анализ предметной области

Целью программного комплекса, разработанного в рамках дипломного проекта, является автоматизация деятельности Финансового управления Администрации **Новоегорлыкского сельского поселения** в части организации бюджетного процесса и контроля над его исполнением на территории **Новоегорлыкского сельского поселения**.

Для того чтобы внести ясность в понятие бюджетный процесс, воспользуемся определением, данным этому термину в Бюджетном кодексе Российской Федерации.

Бюджетный процесс – регламентируемая нормами права деятельность органов государственной власти, органов местного самоуправления и участников бюджетного процесса по составлению и рассмотрению проектов бюджетов, проектов бюджетов государственных внебюджетных фондов, утверждению и исполнению бюджетов и бюджетов государственных внебюджетных фондов, а также по контролю за их исполнением /5/.

В структуре законодательства Российской Федерации регулирующего бюджетный процесс можно выделиться следующие правовые акты:

* Бюджетный кодекс Российской Федерации и федеральные законны о федеральном бюджете на соответствующий год принятые в соответствии с ним;
* законы субъектов Российской Федерации о бюджетах субъектов Российской Федерации на соответствующий год;
* нормативные правовые акты представительных органов местного самоуправления о местных бюджетах на соответствующий год;
* иные федеральные законы, законы субъектов Российской Федерации и нормативные правовые акты представительных органов местного самоуправления, регулирующие правоотношения, возникающие в бюджетном процессе.

Основными принципами бюджетной системы Российской Федерации являются:

* единство бюджетной системы Российской Федерации;
* разграничение доходов и расходов между уровнями бюджетной системы Российской Федерации;
* самостоятельность бюджетов;
* равенство бюджетных прав субъектов Российской Федерации и муниципальных образований;
* полнота отражения доходов и расходов бюджетов;
* сбалансированность бюджета;
* эффективность и экономичность использования бюджетных средств;
* гласность;
* достоверность бюджета;
* адресность и целевой характер бюджетных средств.

Принцип единства бюджетной системы Российской Федерации означает единство бюджетного законодательства Российской Федерации, принципов организации и функционирования бюджетной системы Российской Федерации, форм бюджетной документации и отчетности, бюджетной классификации бюджетной системы Российской Федерации, санкций за нарушение бюджетного законодательства Российской Федерации, единый порядок установления и исполнения расходных обязательств, формирования доходов и осуществления расходов бюджетов бюджетной системы Российской Федерации и бюджетных учреждений.

Принцип разграничения доходов и расходов между бюджетами разных уровней означает закрепление в соответствии с законодательством Российской Федерации доходов и расходов за бюджетами бюджетной системы Российской Федерации, а также определение полномочий органов местного самоуправления по формированию доходов, установлению и исполнению расходных обязательств.

Принцип равенства бюджетных прав субъектов Российской Федерации, муниципальных образований означает определение бюджетных полномочий органов государственной власти субъектов Российской Федерации и органов местного самоуправления, установление и исполнение расходных обязательств. Формирование налоговых и неналоговых доходов бюджетов субъектов Российской Федерации и местных бюджетов, определение объема, форм и порядка предоставления межбюджетных трансфертов в соответствии с едиными принципами и требованиями, установленными Бюджетным кодексом Российской Федерации.

Принцип сбалансированности бюджета означает, что объем предусмотренных бюджетом расходов должен соответствовать суммарному объему доходов бюджета и поступлений из источников финансирования его дефицита.

Принцип эффективности и экономности использования бюджетных средств означает, что при составлении и исполнении бюджетов уполномоченные органы и получатели бюджетных средств должны исходить из необходимости достижения заданных результатов с использованием наименьшего объема средств или достижения наилучшего результата с использованием определенного бюджетом объема средств.

Принцип гласности означает обязательное опубликование в открытой печати утвержденных бюджетов и отчетов об их исполнении, а также обязательную открытость для общества и средств массовой информации процедур рассмотрения и принятия решения по проектам бюджетов.

Принцип достоверности бюджета означает надежность показателей прогноза социально-экономического развития соответствующей территории и реалистичность расчета доходов и расходов бюджета.

Принцип адресности и целевого характера бюджетных средств означает, что бюджетные средства выделяются в распоряжение конкретных получателей бюджетных средств с обозначением направления их на финансирование конкретных целей.

В Российской Федерации устанавливается казначейское исполнение бюджетов. На органы исполнительной власти возлагается организация исполнения и исполнение бюджетов, управление счетами бюджетов и бюджетными средствами. Указанные органы являются кассирами всех распорядителей и получателей бюджетных средств и осуществляют платежи за счет бюджетных средств от имени и по поручению бюджетных учреждений.

В структуре бюджетного процесса можно выделить следующие комплексы задач:

* формирование и утверждение проекта бюджета;
* формирование и утверждение бюджетной росписи;
* ежемесячное формирование кассового плана по подведомственной сети распорядителей и получателей бюджетных средств;
* финансирование расходов подведомственных распорядителей и получателей бюджетных средств;
* контроль над поступлением доходов на счета бюджета;
* контроль над целевым расходованием бюджетных средств;
* формирование и предоставление отчетных данных в вышестоящие контролирующие органы.

К сожалению, эти комплексы задач столь объемны, что не представляется возможным описать и реализовать их все в рамках одного дипломного проекта. Представленный проект полностью охватывает реализацию первого комплекса задач из приведенного выше списка, а также от части седьмого комплекса. Следует отметить, что архитектура информационной системы, спроектированной в рамках дипломного проекта, позволяет легко расширять функциональные возможности программы, что позволяет при дальней проработке предметной области реализовать в системе недостающие комплексы подзадач бюджетного процесса.

При проектировании системы выбран объектно-ориентированный подход, который позволяет описать систему в виде взаимосвязанных сущностей представляющих собой объекты реального мира. Это позволяет значительно упростить процесс проектирования и снизить риски.

При описание системы активно используется Unified Modeling Language 2.0, который позволяет в наглядной графической форме отобразить все аспекты проектируемой системы и CASE‑средство Enterprise Architect фирмы Sparx Systems.

## Анализ существующих решений по автоматизации предметной области

Среди существующих программных решений по автоматизации деятельности администраций следует отметить автоматизированную информационную систему АС «Бюджет» НПО «Криста», которая предназначена для комплексной автоматизации деятельности финансовых органов субъектов РФ и муниципальных образований на всех этапах исполнения бюджета. Данная ИС позволяет организовать исполнение бюджета в соответствии с действующим бюджетным законодательством, обеспечивает создание системы управленческого бюджетного учета и отчетности финансового органа, поддерживает различные варианты кассового обслуживания исполнения бюджета в органах Федерального казначейства. Информационная система АС «Бюджет» обеспечивает выполнение следующих функций:

* автоматический бюджетный контроль;
* множественное визирование документов с применением ЭЦП;
* многобюджетный режим работы;
* расширенный аудит действий пользователей;
* сбора информации;
* учет государственных контрактов и договоров;
* учет бюджетных обязательств;
* электронный обмен с банком);
* сканер двухмерного штрих-кода платежных поручений.

Функция автоматического бюджетного контроля предназначена для автоматизации бюджетного контроля первичных документов в соответствии с принятым порядком организации исполнения бюджета.

Функция множественного визирования документов с применением ЭЦП позволяет внедрить во внутренний документооборот средства ЭЦП и шифрования.

Функция многобюджетный режим позволяет вести учет исполнения нескольких бюджетов в одной базе данных, используя стандартные функциональные возможности.

Функция расширенный аудит действий пользователей предназначена для мониторинга действий пользователей по изменению данных, выявления некорректных действий и восстановления данных.

Функция сбора информации позволяет организовать сбор и консолидацию произвольной оперативной и отчетной информации от нижестоящих отделов. Посредством Системы вышестоящий органы формирует запросы – задания на подготовку информации, а нижестоящие отделы предоставляют информацию в соответствии с заданиями.

Функция учет государственных контрактов и договоров предназначена для учета и контроля государственных контрактов и договоров, заключенных по результатам проведения конкурсов, а также договоров, заключаемых без проведения конкурсных процедур, расшифровки стоимости договора по этапам, по кодам бюджетной классификации и срокам оплаты, по видам продукции и ОКПД; документов исполнения.

Функция учет бюджетных обязательств предназначена для учета финансовыми органами бюджетных обязательств, вытекающих из договоров на поставку продукции, выполнение работ, оказание услуг, заключенных бюджетными учреждениями, и подлежащих оплате за счет средств соответствующего бюджета.

Функция электронный обмен с банком предназначена для осуществления полнофункционального двухстороннего обмена электронными платежными документами между ФО и органами ФК, а также между ФО и учреждениями банков.

Функция сканирования двухмерного штрих-кода платежных поручений предназначена для автоматизации ввода платежных поручений в АС «Бюджет», которая заключается в получении информации, закодированной в двухмерном штрих-коде на бумажной копии платежного поручения, и занесении этой информации в базу данных.

Комплексная система АЦК-Финансы, разработанная компанией «Бюджетные Финансовые Технологии» предназначена для обеспечения автоматизации всего процесса исполнения бюджета и управления бюджетным процессом /6/.

АЦК-Финансы обеспечивает комплексную оптимизацию и автоматизацию всех участков и всех участников бюджетного процесса, в том числе автоматизацию всех структурных подразделений внутри финансового органа, подчиненных территориальных подразделений ФО, распорядителей и получателей бюджетных средств. Кроме того, система АЦК-Финансы решает задачи связи ФО с МНС и УФК, а при необходимости и с обслуживающим банком или РКЦ.

Система АЦК-Финансы позволяет создать единый электронный документооборот, обеспечивающий полную автоматизацию процесса исполнения бюджета, охватывающий всех участников бюджетного процесса в диапазоне от конечных получателей бюджетных средств до главного распорядителя, включая все отделы, входящие в структуру финансового органа, а именно:

* бухгалтерия ФО;
* бюджетный и отраслевые отделы ФО;
* отдел ценных бумаг и неденежных форм расчетов ФО;
* отдел доходов ФО;
* казначейский отдел ФО;
* отдел управления и обслуживания государственного долга ФО;
* отдел учета бюджетных обязательств ФО;
* отдел автоматизации систем финансовых расчетов ФО;
* РБС;
* ПБС и прочие.

Система АЦК-Финансы обеспечивает автоматизацию рабочего места каждого специалиста, участвующего в бюджетном процессе, его функционирование в рамках должностных инструкций, а также связь всех автоматизированных рабочих мест с единой системой электронного документооборота с использованием при передаче электронных документов средств криптографической защиты и электронно-цифровой подписи. Используемые в системе АЦК-Финансы средства криптографической защиты информации и электронно-цифровой подписи имеют все необходимые сертификаты.

Организация единого удаленного документооборота обеспечивает создание единой информационной системы ФО, позволяющей осуществлять мониторинг исполнения регионального бюджета в режиме реального времени, вплоть до конечных ПБС вне зависимости от их территориальной удаленности.

Система АЦК-Финансы, являясь комплексным решением автоматизации ФО, помимо осуществления казначейского расхода, автоматизирует все функции ФО, такие, как:

* формирование проекта бюджета;
* проведение взаимных расчетов между уровнями бюджета;
* расчет / принятие лимитов бюджетных обязательств;
* учет бюджетных обязательств;
* исполнение бюджета по расходам;
* исполнение бюджета по доходам;
* операции с привлеченными средствами и средствами, размещенными на возвратной основе;
* бухгалтерский учет;
* учет средств, полученных от предпринимательской и иной приносящей доход деятельности;
* учет гарантий и поручительств;
* исполнение бюджета по источникам покрытия дефицита бюджета;
* исполнение доходов и расходов целевых бюджетных фондов;
* проведение операций с ценными бумагами;
* анализ исполнения бюджета, получение и свод отчетности об исполнении бюджета;
* капитальное строительство и другие.

Рассмотренные автоматизированная информационная система АС «Бюджет» и комплексная система АЦК-Финансы обеспечивает построение эффективной системы управления бюджетным процессом региона за счет централизации всей информации о ходе бюджетного процесса и автоматизации всех участков и всех участников бюджетного процесса. **Однако бюджетный процесс в региональных финансовых органах имеет ряд специфических особенностей и требует специальных методов и инструментов. В результате этого финансовое управление администрации Новоегорлыкского сельского поселения поставило перед информационным отделом задачу разработки информационной системы автоматизации процесса планирования бюджета, которая должна учитывать особенности в части многовариантности расчетов проекта бюджета, а также оперативности и точности расчетов параметров бюджета.**

## Моделирование бизнес-процессов предметной области

Для того чтобы более четко разобраться с предметной областью /30/ и понять, что требуется от проектируемой информационной системы далее приводится описание существующих в Финансовом управлении бизнес-процессов, которые подлежат автоматизации.

На рисунке 1.1 представлена схема бизнес-процесса по формированию сметы расходов распорядителя бюджетных средств.

Рисунок 1.1 – Бизнес-процесс «Формирование сметы расходов распорядителя»

Целью данного бизнес-процесса является получение сводной сметы расходов распорядителя бюджетных средств на год. Выходной документ, который формируется в результате этого бизнес-процесса, отражает предполагаемые суммы расходов распорядителя бюджетных средств на год, для которого составляется проект бюджета.

Этот документ формируется распорядителями бюджетных средств, которые по большей части, представляют собой внешние по отношению к Финансовому управлению организации. В данной ситуации может возникнуть вопрос, если эти организации являются внешними по отношению к автоматизируемому предприятия, то зачем здесь приводится описание этого бизнес-процесса. Для того чтобы прояснить эту ситуацию отметим, что само Финансовое управление является распорядителем бюджетных средств для некоторых получателей. Решением этой задачи в рамках Финансового управления администрации **Новоегорлыкского сельского поселения** занимаются специалисты бюджетного отдела.

Входной информацией для бизнес-процесса служат сметы расходов подведомственных получателей. В рамках бизнес-процесса для получения выходного документа, производится суммирование объемов денежных средств необходимых подведомственным получателям в разрезе статей расходов. Полученные суммы заносятся в выходной документ, который направляется в Финансовое управление на рассмотрение и утверждение.

Архитектура проектируемой информационной системы предполагает размещение программных модулей по составлению сметы расходов в организациях распорядителях, что позволит автоматизировать процесс передачи данных от распорядителей в Финансовое управление.

На рисунке 1.2 представлена схема бизнес-процесса по формированию сметы доходов администраторов бюджетных средств.

Целью данного бизнес-процесса является получения сметы доходов администратора бюджетных средств. Выходной документ данного бизнес-процесса отражает предполагаемые объемы поступления доходов в году, для которого составляется проект бюджета. Выходной документ данного бизнес-процесса служит входной информацией при составлении доходной части проекта бюджета.

Как и в случае со сметой расходов распорядителя бюджетных средств, смета доходов администратора бюджетных средств по большей части составляется сторонними организациями.

Финансовое управление также является администратором некоторых видов доходов. В рамках Финансового управления этой задачей занимаются специалисты отдела прогнозирования доходов и налоговой политики.

Рисунок 1.2 – Бизнес-процесс «Формирование сметы доходов администратора бюджетных средств»

На рисунке 1.3 представлена схема бизнес-процесса по формированию сметы администратора источников финансирования дефицита.

Рисунок 1.3 – Бизнес-процесс «Формирование сметы администратора источников финансирования дефицита бюджета»

Целью данного бизнес процесса является получение сметы администратора источников финансирования дефицита бюджета. Выходной документ отражает объемы средств направляемых на погашение дефицита бюджета, в случае если такой имеется.

Как и в случае вышеописанных бизнес-процессов, смета администратора источников финансирования дефицита бюджета составляется сторонними организациями, являющимися администраторами источников финансирования дефицита бюджета. Финансовое управление также является администратором некоторых источников финансирования дефицита. В рамках Финансового управления этот бизнес-процесс протекает в бюджетном отделе.

На рисунке 1.4 представлена схема бизнес-процесса по формированию доходной части проекта бюджета.

Рисунок 1.4 – Бизнес-процесс «Формирование доходной части проекта бюджета»

Целью данного бизнес-процесса является получение ведомости доходов – документа, в котором отражаются предполагаемые объемы доходов территории, для которой составляется проект бюджета.

В рамках данного бизнес процесса производится суммирование объемов доходов администраторов бюджетных средств, представленных во входных документов – сметах доходов администраторов бюджетных средств – в соответствии с бюджетной классификацией Российской Федерации.

Выходной документ данного бизнес-процесса является составной частью проекта бюджета.

На рисунке 1.5 представлена схема бизнес-процесса по формированию расходной части проекта бюджета.

Рисунок 1.5 – Бизнес-процесс «Формирование расходной части проекта бюджета»

Целью данного бизнес-процесса является получение ведомости расходов – документа, в котором отражаются предполагаемые объемы расходов территории, для которой составляется проект бюджета.

В рамках данного бизнес-процесса производится суммирование объемов расходов распорядителей бюджетных средств, представленных во входных документах – сводных сметах расходов распорядителей бюджетных средств – в соответствии с бюджетной классификацией Российской Федерации.

Выходной документ данного бизнес-процесса является составной частью проекта бюджета территории.

На рисунке 1.6 представлена схема бизнес-процесса по формированию источников финансирования дефицита бюджета.

Целью данного бизнес-процесса является получение ведомости источников финансирования дефицита бюджета – документа, в котором отражаются источники и объемы средств, направляемых на погашение дефицита бюджета.

Этот бизнес-процесс возникает только тогда, когда проект бюджета является дефицитным, т.е. объем расходов превышает объем доходов. Выходной документ данного бизнес-процесса является составной часть проекта бюджета.

Рисунок 1.6 – Бизнес-процесс «Формирование источников финансирования дефицита бюджета»

На рисунке 1.7 представлена схема бизнес-процесса направленного на поддержание проекта бюджета в актуальном состоянии в течение всего года.

Рисунок 1.7 – Бизнес-процесс «Корректировка проекта бюджета»

Так как при исполнении бюджета начальные показатели меняются, то проект бюджета необходимо поддерживать в актуальном состоянии. Это достигается за счет справок-уведомлений. Корректировка бюджета осуществляется по показателям доходов, расходов и источников финансирования дефицита бюджета.

Справки-уведомления на уменьшение объема ассигнований служат для уменьшения изначально запланированных объемов средств.

Справки-уведомления на увеличение объема ассигнований служат для увеличения изначально запланированных объемов средств.

На рисунке 1.8 представлена схема бизнес-процесса по формированию консолидированного бюджета.

Рисунок 1.8 – Бизнес-процесс «Формирование консолидированного бюджета территории»

Целью данного бизнес-процесса является получение проекта консолидированного бюджета территории.

Так как **Новоегорлыкское сельское поселения** представляет нескольких сельских поселений, то для предоставления данных в Министерство финансов Ростовской области составляется проект консолидированного бюджета территории. В этом документе отражаются общие показатели объемов доходов и расходов, а также объемов средств выделенных на покрытие дефицита бюджета **Новоегорлыкского сельского поселения**.

## Анализ и моделирование требований

###

### **Формирования функциональных требований**

Для того чтобы определить функциональные требования, предъявляемые к системе, необходимо, прежде всего, выявить лиц заинтересованных в этой системе, а затем определить тот функционал, который им требуется для осуществления своей профессиональной деятельности.

Заинтересованные в системе пользователи, которые были выявлены в процессе исследования бизнес-процессов и предпроектного обследования Финансового управления, представлены на рисунке 1.9.

Рисунок 1.9 – Пользователи системы

Дадим краткую характеристику каждому классу пользователей системы. Администраторы автоматизированной информационной системы бюджетного процесса занимается настройкой системы, управлением пользователями и пользовательскими группами, управлением правами доступа.

Специалисты бюджетного отдела – сотрудники отделов Финансового управления, которые занимаются составлением расходной части проекта бюджета и бюджетной росписи.

Специалисты отдела прогнозирования доходов и налоговой политики – сотрудники отдела Финансового управления, в задачи которых входит составление доходной части проекта бюджета и бюджетной росписи.

Распорядители бюджетных средств – это организации, управляющие распределением бюджетных средств по подведомственным получателям и осуществляющих их финансирование.

Администраторы бюджетных средств – это организации, управляющих поступлениями доходов в бюджет.

Администраторы источников финансирования дефицита – это организации, управляющие поступлениями средств, направленных на покрытие дефицита бюджета.

После того, как мы выявили основных пользователей системы, проведем анализ вариантов использования ими системы. Прецеденты фактически и являются функциональными требованиями к системе.

На рисунке 1.10 представлены варианты использования системы распорядителем бюджетных средств.

В процессе выполнения прецедента «Формирование сметы расходов» пользователь выполняет поэтапное формирование сметы расходов, последовательно вводя предполагаемые суммы расходов на год по соответствующим целевым статьям бюджетной классификации Российской Федерации.

В процессе выполнения прецедента «Утверждение сметы расходов» в системе фиксируется состояние сметы расходов, и дальнейшая ее модификации в течение года возможна только при помощи справок-уведомлений.

В процессе выполнения прецедента «Передача сметы расходов Финансовому управлению» происходит передача сметы расходов распорядителя бюджетных средств в Финансовое управление для проверки и окончательного утверждения. В дальнейшем сметы расходов распорядителей служат основой для формирования расходной части проекта бюджета.

В процессе выполнения прецедента «Внесение изменений в смету расходов» пользователь проводит корректировку сумм расходов предполагаемых в планируемом году. Выполнение этого прецедента возможно, только если смета расходов еще не утверждена, т.е. если не выполнялся прецедент «Утверждение сметы расходов на год».

В процессе выполнения прецедента «Ввод справок-уведомлений» пользователь проводит корректировку показателей сметы расходов в течение года исполнения бюджета. Этот прецедент служит для подержания сметы расходов в актуальном состоянии в течение всего года.

Рисунок 1.10 – Варианты использования системы распорядителем бюджетных средств

На рисунке 1.11 представлены варианты использования системы администратором бюджетных средств.

В процессе выполнения прецедента «Формирование сметы доходов» пользователь производит поэтапное формирование сметы доходов, последовательно вводя предполагаемые суммы доходов на год по соответствующим видам доходов бюджетной классификации Российской Федерации.

Рисунок 1.11 – Варианты использования системы администратором бюджетных средств

В процессе выполнения прецедента «Утверждение сметы доходов» в системе фиксируется состояние сметы доходов, и дальнейшая ее модификация в течение года возможна только при помощи справок уведомлений.

В процессе выполнения прецедента «Внесение изменений в смету доходов» пользователь проводит корректировку сумм доходов предполагаемых в планируемом году. Выполнение этого прецедента возможно, только если смета доходов еще не утверждена, т.е. если не выполнялся прецедент «Утверждение сметы доходов».

В процессе выполнения прецедента «Передача сметы доходов в Финансовое управление» происходит передача сметы доходов администратора бюджетных средств в Финансовое управление для проверки и окончательного утверждения.

В дальнейшем сметы доходов администраторов служат основой для формирования доходной части проекта бюджета.

В процессе выполнения прецедента «Ввод справок-уведомлений» пользователь проводить корректировку показателей сметы доходов в течение года исполнения бюджета. Этот прецедент служит для поддержания сметы доходов в актуальном состоянии в течение всего года.

На рисунке 1.12 представлены варианты использования систему администраторами источников финансирования дефицита бюджета.

В процессе выполнения прецедента «Формирование сметы источников финансирования дефицита бюджета» пользователь производит поэтапное формирование сметы источников финансирования дефицита, последовательно вводя предполагаемые суммы средств направляемых на покрытие дефицита бюджета в соответствии с бюджетной классификацией Российской Федерации.

В процессе выполнения прецедента «Утверждение сметы источников финансирования дефицита бюджета» в системе фиксируется состояние сметы источников финансирования дефицита бюджета, и дальнейшая ее модификация в течение года возможна только при помощи справок уведомлений.

В процессе выполнения прецедента «Внесение изменений в смету источников финансирования дефицита» пользователь проводит корректировку объемов средств направляемых на покрытие дефицита бюджета. Выполнение этого прецедента возможно, только если смета источников финансирования дефицита бюджета еще не утверждена, т.е. если не выполнялся прецедент «Утверждение сметы источников финансирования дефицита бюджета».

Рисунок 1.12 – Варианты использования системы администратором источников финансирования дефицита бюджета

В процесса выполнения прецедента «Передача сметы источников финансирования дефицита бюджета в Финансовое управление» происходит передача сметы источников финансирования дефицита бюджета в Финансовое управление для проверки и окончательного утверждения. В дальнейшем сметы источников финансирования дефицита бюджета служат основой для формирования источников финансирования дефицита в проекте бюджета.

В процессе выполнения прецедента «Ввод справок-уведомлений» пользователь проводить корректировку показателей сметы источников финансирования дефицита в течение года исполнения бюджета. Этот прецедент служит для поддержания сметы источников финансирования дефицита в актуальном состоянии в течение всего года.

На рисунке 1.13 представлены варианты использования системы администратором автоматизированной системы бюджетного процесса при управлении пользователями.

Рисунок 1.13 – Прецеденты управлении пользователями ИС

В процессе выполнения прецедента «Регистрация пользователя» администратор регистрирует в системе новую учетную запись.

В процессе выполнения прецедента «Блокирование пользователя» администратор временно блокирует учетную запись пользователя. Если пользователь в это время подключен к системе, то он уведомляется о том, что его учетная запись заблокирована, после чего происходит его отключение от системы.

В процессе выполнения прецедента «Удаление пользователя» администратор удаляет учетную запись пользователя и списка зарегистрированных в системе.

В процессе выполнения прецедента «Назначение прав доступа пользователю» администратор назначает пользователю права доступа к системе, которые необходимы ему для осуществления своей профессиональной деятельности.

На рисунке 1.14 представлены варианты использования системы администратором автоматизированной информационной системы при управлении пользовательскими группами.

В процессе выполнения прецедента «Регистрация пользовательской группы» администратор регистрирует в системе новую пользовательскую группу. Пользовательские группы служат для облегчения процесса администрирования системы.

В процессе выполнения прецедента «Удаление пользовательской группы» администратор удаляет выбранную пользовательскую группу из списка зарегистрированных в системе групп.

В процессе выполнения прецедента «Управление членством пользователей в пользовательских группах» администратор управляет составом пользовательских групп. Он может добавить нового члена, выбрав его из списка зарегистрированных в системе, либо наоборот исключить пользователя из группы.

В процессе выполнения прецедента «Назначение прав доступа пользовательской группе» администратор назначает пользовательской группе права доступа к системе, общие для всех членов этой группы.

На рисунке 1.15 представлены прочие варианты использования системы администратором автоматизированной системы бюджетного процесса.

Рисунок 1.14 – Прецеденты управления пользовательскими группами

В процессе выполнения прецедента «Формирование справочников бюджетной классификации» администратор производит заполнение справочников бюджетной классификации.

В процессе выполнения прецедента «Импорт справочников бюджетной классификации» администратор производит формирование справочников бюджетной классификации путем импорта данных из справочников бюджетной классификации прошлых лет. После импорта данных администратор проводит их корректировку, для того чтобы данные, содержащиеся в справочниках, соответствовали бюджетной классификации Российской Федерации для текущего года.

Рисунок 1.15 – Прочие варианты использования системы администратором

На рисунке 1.16 представлены варианты использования системы специалистом бюджетного отдела финансового управления при формировании проекта бюджета.

В процессе выполнения прецедента «Формирование сети распорядителей бюджетных средств» пользователей формирует список организаций являющихся распорядителями бюджетных средств. Каждому распорядителю присваивается код в соответствии с бюджетной классификацией Российской Федерации.

В процессе выполнения прецедента «Формирование сети администраторов источников финансирования дефицита бюджета» пользователь формирует список организаций являющихся администраторами источников финансирования дефицита бюджета. Каждому администратору присваивается код в соответствии с бюджетной классификацией Российской Федерации.

В процессе выполнения прецедента «Формирование проекта бюджета» пользователь регистрирует в системе новый проект бюджета, указывая при этом поселение и год на который составляется проект бюджета.

Рисунок 1.16 – Варианты использования формирования проекта бюджета

В процессе выполнения прецедента «Проверка и утверждение сметы расходов распорядителя» пользователь проверяет сметы, поступившие от распорядителей бюджетных средств. Если смета корректна, то пользователь утверждает ее и в дальнейшем эта смета участвует в формировании проекта бюджета. При утверждении сметы распорядителю при следующем его подключении к системе выдается сообщение о том, что его смета утверждена. Если же в смете содержатся ошибки, то специалист бюджетного отдела формирует список замечаний и отправляет смету распорядителю на доработку, о чем система также оповещает распорядителя при первом же его подключении к системе.

В процессе выполнения прецедента «Проверка и утверждение сметы администратора источников финансирования дефицита бюджета» пользователь проверяет сметы, поступившие от администраторов источников финансирования дефицита бюджета. В случае корректности сметы, пользователь ее утверждает, и в дальнейшем она участвует в формирования проекта бюджета. При утверждении сметы администратору при следующем его подключении к системе выдается уведомление о том, что его смета утверждена. Если же в смете содержатся ошибки, то специалист бюджетного отдела формирует список замечаний и отправляет смету администратору на доработку, о чем система также оповещает администратора при первом же его подключении к системе.

В процессе выполнения прецедента «Утверждение проекта бюджета» в системе фиксируется состояние проекта бюджета и дальнейшие его изменения возможны только при помощи справок уведомлений. Выполнения этого прецедента возможно только после того как утверждены сметы от всех распорядителей бюджетных средств, администраторов бюджетных средств и администраторов источников финансирования дефицита.

В процессе выполнения прецедента «Ввод справок-уведомлений» проводит корректировку показателей проекта бюджета в течение года исполнения бюджета. Этот прецедент служит для поддержания проекта бюджета в актуальном состоянии в течение всего года.

В процессе выполнения прецедента «Сравнительны анализ проектов бюджета» пользователь выбирает проекты бюджета, которые он хочет сравнить, а системе выдает ему отчет, который содержит показатели проектов бюджета и динамику роста.

На рисунке 1.17 представлены варианты использования системы специалистом бюджетного отдела при формировании проекта консолидированного бюджета территории.

Рисунок 1.17 – Варианты использование формирования проекта консолидированного бюджета территории

В процессе выполнения прецедента «Формирование проекта консолидированного бюджета» пользователь регистрирует в системе новый проект консолидированного бюджета, указывая при этом наименовании территории и год на который составляется проект бюджета, а также проекты бюджетов, которые участвуют в формировании консолидированного бюджета.

На рисунке 1.18 представлены варианты использования системы специалистом отдела прогнозирования доходов и налоговой политики.

В процессе выполнения прецедента «Формирование сети администраторов бюджетных средств» пользователь формирует список организаций являющихся администраторами бюджетных средств. Каждому администратору бюджетных средств присваивается код в соответствии с бюджетной классификацией Российской Федерации.

Рисунок 1.18 – Варианты использования прогнозирования доходов и налоговой политики

В процессе выполнения прецедента «Проверка и утверждение сметы доходов администратора бюджетных средств» пользователь проверяет сметы, поступившие от администраторов бюджетных средств. Если смета корректна, то пользователь утверждает ее и в дальнейшем эта смета участвует в формировании проекта бюджета. При утверждении сметы администратору при следующем его подключении к системе выдается сообщение о том, что его смета утверждена. Если же в смете содержатся ошибки, то специалист отдела прогнозирования доходов и налоговой политики формирует список замечаний и отправляет смету администратору бюджетных средств на доработку, о чем система также оповещает администратора при первом же его подключении к системе.

В процессе выполнения прецедента «Ввод справок-уведомлений» проводит корректировку показателей проекта бюджета в течение года исполнения бюджета. Этот прецедент служит для поддержания проекта бюджета в актуальном состоянии в течение всего года.

На рисунке 1.19 представлены варианты использования системы, характерные для всех пользователей.

Рисунок 1.19 – Варианты использования общие для всех пользователей

В процессе выполнения прецедента «Аутентификация в системе» пользователь осуществляет вход в систему посредством ввода логина и пароля.

В процессе выполнения прецедента «Смена пароля» пользователь меняет пароль для своей учетной записи.

### **Формирование нефункциональных требований**

В предыдущем разделе описаны функциональные требования, предъявляемые к разрабатываемой системе. Однако наличие описания лишь функциональных требований не является достаточным условием для начала проектирования и разработки системы, поэтому ниже приводится список нефункциональных требований, предъявляемых к разрабатываемой системе, выявленных в процессе предварительного обследования организации и опроса пользователей системы.

К операционной среде, в которой должна работать проектируемая система, предъявляются следующие требования:

* компьютер, на котором размещается серверная часть приложения, должен работать под управлением операционной системы не ниже Microsoft Windows Server 2000. Также на компьютере должны быть установленные компоненты. Net Framework 2.0;
* компьютеры, на которых размещается клиентская часть приложения, должны работать под управлением операционной среды не ниже Microsoft Windows XP Professional Edition SP2 с установленными компонентами. Net Framework 2.0;
* проектируемая система должна допускать доступ пользователей через корпоративную сеть интранет и Интернет.

К интерфейсу пользователя предъявляются следующие требования:

* клиентская часть системы должна быть выполнена в виде windows-приложения с многодокументным интерфейсом;
* формы должны быть снабжены контекстной справкой.

К производительности системы предъявляются следующие требования:

* система должна обслуживать одновременно до 100 пользователей в период пиковой активности с 9:00 до 18:00 по местному времени;
* отклик системы не должен превышать 10 секунд с момента передачи запроса.
* система должна быть доступна пользователям корпоративной сети интранет и клиентам удаленного доступа по коммутируемой линии 99% времени между 0:00 и 24:00 семь дней в неделю.

К безопасности, проектируемой системы, предъявляются следующие требования:

* все сетевые транзакции должны быть зашифрованы;
* функции системы становятся доступными пользователю только после его аутентификации в системе;
* регистрация новых пользователей в системе осуществляется только администратором системы.

Система также должна позволять экспорт выходных документов в форматы Microsoft Word и Excel.

## Спецификация состояний системы

Спецификация состояний дает статический взгляд на систему и определяется моделью классов предметной области, их атрибутами и отношениями. Для более четкого понимания предметной области ниже представлена модель ее классов, разбитая на логические части, содержащие объекты предметной области и показывающая их взаимосвязи.

Рисунок 1.20 – Объекты бюджетной классификации

Таблица 1 – Сущности бюджетной классификации

|  |  |
| --- | --- |
| **Наименование** | **Описание** |
| Budgetclassification | Бюджетная классификация |
| Revenuegroup | Группа доходов |
| Revenuesubgroup | Подгруппа хододов |
| Revenueclause | Статья доходов |
| Revenuesubclause | Подстатья доходов |
| Revenueeconomicclass | Класс экономической классификации доходов |
| Revenueprogram | Программа доходов |
| Element | Элемент бюджетной классификации |
| Revenue | Доход |
| Outlaysection | Раздел расходов |
| Outlaysubsection | Подраздел расходов |
| Outlayclause | Целевая статья расходов |
| Outlayclass | Класс экономической классификации расходов |
| Outlayprogram | Программа расходов |
| Outlaysort | Вид расходов |
| Outlay | Расход |
| Sfdgroup | Группа бюджетной классификации источников финансирования дефицита |
| Sfdsubgroup | Подгруппа бюджетной классификации источников финансирования дефицита |
| Sfdclause | Статья бюджетной классификации источников финансирования дефицита |
| Sfdsubclause | Подстатья бюджетной классификации источников финансирования дефицита |
| Sfdprogram | программа источников финансирования дефицита |
| Sfdeconomicclass | Класс экономической классификации источников финансирования дефицита |
| Sfd | Источник финансирования дефицита |

На рисунке 1.21 представлены объекты и сущности участвующие в процессах составления смет доходов, расходов и источников финансирования дефицита.

В таблице 2 представлена расшифровка объектов и сущностей, участвующих в процессах составления смет доходов, расходов и источников финансирования дефицита.

Рисунок 1.21 – Объекты и сущности процесса составления смет доходов, расходов и источников финансирования дефицита

Таблица 2 – Объекты и сущности участвующие в процессах составления смет доходов, расходов и источников финансирования дефицита

|  |  |
| --- | --- |
| **Наименование** | **Описание** |
| Legalentity | Юридическое лицо |
| Bcsteward | Распорядитель бюджетных средств |
| Outlayestimate | Смета расходов |
| Outlayestimateitem | Строка сметы расходов |
| Bcadministrator | Администратор бюджетных средств |
| Revenueestimate | Смета доходов |
| Revenueestimateitem | Строка сметы доходов |
| Sfdadministrator | Администратор источников финансирования дефицита |
| Sfdestimate | Смета источников финансирования дефицита |
| Sfdestimateitem | Строка сметы источников финансирования дефицита |
| Enquiry | Справка-уведомление на изменение выделенных ассигнований |

На рисунке 1.22 представлены субъекты и объекты, участвующие в процессе составления проекта бюджета.

Рисунок 1.22 – Объекты и субъекты процесса составления проекта бюджета

В таблице 3 представлена расшифровка субъектов и объектов, участвующих в процессе составления проекта бюджета.

На рисунке 1.23 представлены объекты, участвующие в процессе составления консолидированного бюджета территории.

Таблица 3 – Субъекты и объекты, участвующие в процессе составления проекта бюджета

|  |  |
| --- | --- |
| **Наименование** | **Описание** |
| User | Пользователь автоматизированной системы бюджетного процесса |
| Faleader | Начальник финансового управления |
| BDManager | Специалист бюджетного отдела |
| DPRManager | Специалист отдела прогнозирования доходов и налоговой политики |
| BudgetProject | Проект бюджета |
| Settlement | Поселение для которого составляется проект бюджета |

Рисунок 1.23 – Объекты процесса составления проекта бюджета

Объект ConsolidatedBudgetProject представляется проект консолидированного бюджета территории.

## Аттестация требований

Аттестация требований – процесс проверки требований на достоверность, непротиворечивость, полноту и выполнимость.

Существует набор методов аттестации, которые можно использовать как вместе, так и по отдельности:

* обзор требований – процесс просмотра системной спецификации на предмет неточных описаний и ошибок;
* прототипирование – прототип является начальной версией программной системы, которая используется для демонстрации концепций заложенных в систему, проверки вариантов требований. Прототип программного обеспечения помогает на двух этапах разработки системных требований: на этапе постановке и этапе проверки. На этапе постановки пользователь может экспериментировать с системными прототипами, что позволяет им, проверять как будет работать система. В результате могут сформироваться новые требования. На этапе проверки требований прототип позволяет обнаружить ошибки и упущения в ранее принятых требованиях;
* генерация тестовых сценариев. Требования должны быть такими, чтобы их можно было протестировать. Если тесты для требований разрабатываются как часть процесса аттестации, то это позволяет обнаружить ошибки в спецификации.

Обзор требований и прототипирование являются основными методами аттестации требований. Аттестация должна продемонстрировать, что требования действительно определяют ту систему, которую хочет иметь заказчик. Проверка требований важна, так как ошибки в спецификации требований могут привести к переделке системы и большим затратам, если будут обнаружены во время процесса разработки системы или введения ее в эксплуатацию.

В процессе моделирования требований к информационной системе диаграммы вариантов использования и диаграммы классов обсуждались с заказчиком и конечными пользователями. В процессе обсуждений были согласованы спецификации вариантов использования и состояний системы.

Прототипы пользовательского интерфейса рассматриваются в п. 3.2, а тестовые сценарии – п. 3.4 дипломного проекта.

Методология составляет основу проекта любой информационной системы. Методология реализуется через конкретные технологии и поддерживающие их стандарты, методики и инструментальные средства, которые позволяют выполнять все процессы жизненного цикла.

Существует две основных методологии проектирования:

* методология структурного проектирования;
* методология объектно-ориентированного проектирования.

Структурный подход. Сущность структурного подхода к проектированию заключается в декомпозиции задачи на автоматизируемые подсистемы, комплексы задач, задачи, функции и т.д. Каждая большая часть системы подразделяется на более мелкие.

Все компоненты информационной системы взаимосвязаны, система разрабатывается сверху − вниз. При разработке системы снизу – вверх целостность теряется, возникают проблемы состыковки компонентов.

Наиболее распространенные модели структурного подхода:

SADT – Structured Analysis and Design Techniques – описывает модели и функциональные диаграммы;

DFD – Data Flow Diagrams – диаграммы потоков данных;

ERD – Entity Relationship Diagrams – диаграммы «сущность – связь».

Объектно-ориентированный подход. Центральным понятием объектно-ориентированного подхода к проектированию является класс. Класс – это выделение из окружающего мира некой сущности, для которой определены атрибуты и операции.

Объект – это конкретная реализация некоторой сущности. В объекте инкапсулируется некоторая часть приложения, которая может представлять собой процесс, группу данных или какую-либо более сложную сущность.

Для реализации проекта был выбран объектно-ориентированный подход в силу следующих факторов:

* возможность повторного использования кода;
* повышение безопасности кода за счет инкапсуляции;
* гибкость при модификации и расширении системы;
* отсутствие необходимости разработки классов с нуля, за счет наследования;
* общая ориентированность объектно-ориентированной технологии на разработку информационных систем, как класса программного обеспечения и т.д.

Проведенный анализ предметной области выявил основные задачи, которые необходимо автоматизировать при разработке информационной системы бюджетного процесса финансового управления Новоегорлыкского сельского поселения. Рассмотрение существующих решений по информатизации управления региональными и местными бюджетами показало, что целесообразно провести индивидуальную разработку информационной системы управления бюджетным процессом финансового управления Новоегорлыкского сельского поселения. На основе моделирования бизнес-процессов были разработаны и специфицированы требования к информационной системе, которые определяют последующие этапы создания информационной системы.

В качестве методологии проектирования обосновано применение объектно-ориентированного подхода.

# Разработка информационного обеспечения

## Разработка концептуальной модели данных

Реляционная модель данных в подавляющем большинстве случаев вполне достаточна для моделирования любых данных. Однако проектирование базы данных в терминах схемы отношений на практике может вызвать большие затруднения, т.к. в этой модели изначально не предусмотрены механизмы описания семантики предметной области. С этим связано появление семантических моделей данных, которые позволяют описать конкретную предметную область гораздо ближе к интуитивному пониманию и, в то же время, достаточно формальным образом.

Следует заметить, что многие начинающие проектировщики баз данных недооценивают важность семантического моделирования. Зачастую это воспринимается как дополнительная и излишняя работа. Эта точка зрения является абсолютно неверной. Во-первых, построение мощной и наглядной концептуальной схемы базы данных позволяет более полно оценить специфику моделируемой предметной области и избежать возможных ошибок на стадии проектирования схемы реляционной базы данных. Во-вторых, на этапе семантического моделирования производится очень важная документация, которая может оказаться полезной не только при проектировании схемы реляционной базы данных, но и при эксплуатации, сопровождении и развитии уже заполненной базы данных. На практике неоднократно наблюдались жизненные ситуации, когда отсутствие такого рода документации существенно затрудняло выполнение даже небольших изменений в схеме существующей реляционной базы данных.

Полная концептуальная модель данных проектируемой системы довольно объемна, поэтому для удобства ее представления в дипломном проекте, она разбита на логические части. На рисунке 2.1 представлена часть концептуальной модели, которая отображает бюджетную классификацию доходов.

Рисунок 2.1 – Концептуальная модель бюджетной классификации доходов

В таблице 4 представлено описание классов и их атрибутов концептуальной модели бюджетной классификации доходов.

Таблица 4 – Описание классов концептуальной модели бюджетной классификации доходов

|  |  |  |
| --- | --- | --- |
| **Класс** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| Бюджетная Классификация |  | Бюджетная классификация РФ |
|  | Год | Год для которого действительна бюджетная классификация |
| Группа Доходов |  | Группа доходов |
|  | Код | Код группы доходов |
|  | Наименование | Наименование группы доходов |
| Подгруппа Доходов |  | Подгруппа доходов |
|  | Код | Код подгруппы доходов |
|  | Наименование | Наименование подгруппы доходов |
| Статья Доходов |  | Статья бюджетной классификации доходов |
|  | Код | Код статьи доходов |
|  | Наименование | Наименование статьи доходов |
| Прог Доход |  | Программа бюджетной классификации доходов |
|  | Код | Код программы доходов |
|  | Наименование | Наименование программы доходов |

На рисунке 2.2 представлена часть концептуальной модели, которая отображает бюджетную классификацию расходов.

Рисунок 2.2 – Концептуальная модель бюджетной классификации расходов

В таблице 5 представлено описание классов и их атрибутов концептуальной модели бюджетной классификации расходов.

Таблица 5 – Описание классов концептуальной модели бюджетной классификации расходов

|  |  |  |
| --- | --- | --- |
| **Класс** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| Раздел Расходов |  | Раздел бюджетной классификации расходов |
|  | Код | Код раздела расходов |
|  | Наименование | Наименование раздела расходов |
| Подразд Расходов |  | Подраздел бюджетной классификации расходов |
|  | Код | Код подраздела расходов |
|  | Наименование | Наименование подраздела расходов |
| ЦСР |  | Целевая статья расходов |
|  | Код | Код целевой статьи расходов включающий программный срез |
|  | Наименование | Наименование целевой статьи расходов |
| Вид Расходов |  | Вид расходов |
|  | Код | Код вида расходов |
|  | Наименование | Наименование вида расходов |
| Эконом Класс Расходов |  | Экономический класс расходов |
|  | Код | Код экономического класса расходов |
|  | Наименование | Наименование экономического класса расходов |
| Расход |  | Расход |
|  | Код | Код расхода |

На рисунке 2.3 представлена часть концептуальной модели, которая отображает бюджетную классификацию источников финансирования дефицита бюджета.

В таблице 6 представлено описание классов и их атрибутов концептуальной модели бюджетной классификации источников финансирования дефицита.

Рисунок 2.3 – Концептуальная модель бюджетной классификации источников финансирования дефицита

Таблица 6 – Описание классов концептуальной модели бюджетной классификации источников финансирования дефицита

|  |  |  |
| --- | --- | --- |
| **Класс** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| ГруппаИФД |  | Группа источников финансирования дефицита |
|  | Код | Код группы источников финансирования дефицита |
|  | Наименование | Наименование группы источников финансирования дефицита |
| ПодгрИстФинДеф |  | Подгруппа источников финансирования дефицита |
|  | Код | Код подгруппы источников финансирования дефицита |
|  | Наименование | Наименование подгруппы источников финансирования дефицита |
| СтатьяИФД |  | Статья источников финансирования дефицита |
|  | Код | Код статьи источников финансирования дефицита |
|  | Наименование | Наименование статьи источников финансирования дефицита |
| ПодстатьяИФД |  | Подстатья источников финансирования дефицита |
|  | Код | Код подстатьи источников финансирования дефицита |
|  | Наименование | Наименование подстатьи источников финансирования дефицита |
| ПрогИФД |  | Программа источников финансирования дефицита |
|  | Код | Код программы источников финансирования дефицита |
|  | Наименование | Наименование программы источников финансирования дефицита |
| ЭкономКлассИФД |  | Экономических класс источников финансирования дефицита |
|  | Код | Код экономического класса источников финансирования дефицита |
|  | Наименование | Наименование экономического класса источников финансирования дефицита |
| ИФД |  | Источник финансирования дефицита |
|  | Код | Код источника финансирования дефицита |

На рисунке 2.4 представлена часть концептуальной модели, которая отображает объекты, участвующие в процессе составления смет доходов администраторов бюджетных средств и доходной части проекта бюджета.

Рисунок 2.4 – Объекты, участвующие в процессах составления смет доходов администраторов бюджетных средств и доходной части проекта бюджета

В таблице 7 представлено описание классов объектов, участвующих в процессах составления смет доходов администраторов бюджетных средств и доходной части проекта бюджета.

Таблица 7 – Описание классов объектов, участвующих в процессах составления смет доходов администраторов бюджетных средств и доходной части проекта бюджета

|  |  |  |
| --- | --- | --- |
| **Класс** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| Проект Бюджета |  | Проект бюджета |
|  | Год | Год на который составляется проект бюджета |
|  | Статус | Текущее состояние проекта бюджета |
|  | Наименование | Название проекта бюджета |
| Юр Лицо |  | Юридическое лицо |
|  | ИНН | Индивидуальный номер налогоплательщика |
|  | Наименование | Название организации |
|  | Адрес | Адрес организации |
| Администратор |  | Администратор бюджетных средств |
|  | Код | Код администратора бюджетных средств |
| Смета Доходов |  | Смета доходов администратора бюджетных средств |
|  | Номер | Номер документа |
|  | Статус | Текущее состояние сметы доходов |
| Строка Смет Доход |  | Строка табличной части сметы доходов |
|  | Сумма | Предполагаемых объем доходов |
|  | Примечание | Примечание служит для любых пояснений |

На рисунке 2.5 представлена часть концептуальной модели, которая отображает объекты, участвующие в процессе составления смет расходов распорядителей бюджетных средств и расходной части проекта бюджета.

Рисунок 2.5 – Объекты, участвующие в процессах составления смет расходов распорядителей бюджетных средств и расходной части проекта бюджета

В таблице 8 представлено описание классов объектов, участвующих в процессах составления смет расходов распорядителей бюджетных средств и расходной части проекта бюджета.

Таблица 8 – Описание классов объектов, участвующих в процессах составления смет расходов распорядителей бюджетных средств и расходной части проекта бюджета

|  |  |  |
| --- | --- | --- |
| **Класс** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| Распорядитель |  | Распорядитель бюджетных средств |
|  | Код | Код распорядителя бюджетных средств |
| Смета Расходов |  | Смета расходов распорядителя бюджетных средств |
|  | Номер | Номер документа |
|  | Статус | Текущее состояние сметы расходов |
| Строка Смет Расх |  | Строка табличной части сметы расходов |
|  | Сумма | Предполагаемый объем расходов |
|  | Примечание | Комментарии |
| Справ Уведомление Расходы |  | Справка-уведомление по расходам |
|  | Номер | Номер документа |
|  | Дата | Дата внесения |
|  | Примечание | Комментарии к справке-уведомлению |
|  | Статус | Текущее состояние справки-уведомления |
| Строка Спр Увед Расх |  | Строка табличной части справки-уведомления по расходам |
|  | Сумма | Объем средств, на который увеличиваются или уменьшаются расходы |
|  | Примечание | Комментарии |
| Поселение |  | Городское или сельское поселение |
|  | Код | Код поселения |
|  | Наименование | Название поселения |
| Тип Документа |  | Тип документа основания для справок-уведомлений |
|  | Код | Код типа документа |
|  | Наименование | Название типа документа |

На рисунке 2.6 представлена часть концептуальной модели, которая отображает объекты, участвующие в процессе составления смет источников финансирования дефицита администраторов источников финансирования дефицита бюджета.

Рисунок 2.6 – Объекты, участвующие в процессе составления смет источников финансирования дефицита бюджета

В таблице 9 представлено описание классов объектов, участвующих в процессе составления смет источников финансирования дефицита бюджета.

Таблица 9 – Описание классов объектов, участвующих в процессе составления смет источников финансирования дефицита бюджета

|  |  |  |
| --- | --- | --- |
| Класс | Атрибут | Описание |
| 1 | 2 | 3 |
| Администратор ИФД |  | Администратор источников финансирования дефицита |
|  | Код | Код администратора источников финансирования дефицита |
| Смета ИФД |  | Смета источников финансирования дефицита |
|  | Номер | Номер документа |
|  | Статус | Текущее состояние сметы источников финансирования дефицита |
| Строка Сметы ИФД |  | Строка табличной части сметы источников финансирования дефицита |
|  | Сумма | Предполагаемый объем средств направляемых на покрытие дефицита |
|  | Примечание | Комментарии |
| Справка Уведомление ИФД |  | Справка-уведомление по источникам финансирования дефицита |
|  | Номер | Номер документа |
|  | Дата | Дата внесения |
|  | Примечание | Комментарии |
|  | Статус | Текущее состояние справки-уведомления |

На рисунке 2.7 представлена часть концептуальной модели, которая отображает объекты, участвующие в процессе составления консолидированного проекта бюджета территории.

Рисунок 2.7 – Объекты, участвующие в процессе составления консолидированного проекта бюджета территории

В таблице 10 представлено описание классов объектов, участвующих в процессе составления консолидированного проекта бюджета территории.

Таблица 10 – Описание классов объектов, участвующих в процессе составления консолидированного проекта бюджета территории

|  |  |  |
| --- | --- | --- |
| **Класс** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| Территория |  | Территория для которого составляется проект бюджета |
|  | Код | Код территории |
|  | Наименование | Название территории |
| Проект Консолид Бюджета |  | Проект консолидированного бюджета территории |
|  | Год | Год на который составляется проект бюджета |
|  | Наименование | Название консолидированного проекта бюджета территории |

На рисунке 2.8 представлена часть концептуальной модели, которая отображает объекты, участвующие в процессе управления пользователями проектируемой системы и их правами доступа.

Рисунок 2.8 – Объекты, участвующие в процессе управления пользователями системы и их правами доступа

В таблице 11 представлено описание классов объектов, участвующих в процессе управления пользователями системы и их правами доступа.

Таблица 11 – Описание классов объектов, участвующих в процессе управления пользователями системы и их правами доступа

|  |  |  |
| --- | --- | --- |
| **Класс** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| Пользователь |  | Пользователь системы |
|  | Логин | Имя учетной записи пользователя |
|  | Пароль | Хэш пароля учетной записи пользователя |
|  | Ф.И.О. | Имя пользователя |
| Пользовательская Группа |  | Группа пользователей |
|  | Код | Код пользовательской группы |
|  | Наименование | Название пользовательской группы |
| Право Доступа |  | Право доступа к функциональности системы |
|  | Наименование | Название права доступа |
| Группа Прав Доступа |  | Группа объединяющая несколько прав доступа |
|  | Код | Код группы прав доступа |
|  | Наименование | Название группы прав доступа |

##

## Обоснование выбора СУБД

Одним из требований заказчика является реализация информационного обеспечения на основе Microsoft SQL Server 2000 Enterprise Edition /24/, в связи с тем, что у них уже имеется лицензионная копия данного продукта. Это позволит в некоторой степени сократить первоначальные затраты на разработку и внедрение системы.

Microsoft SQL Server 2000 – это законченное решение для управления и анализа данных, позволяющее оперативно развертывать масштабируемые приложения нового поколения. SQL Server 2000 – ключевой компонент поддержки электронной коммерции, интерактивных деловых приложений и хранилищ данных, обеспечивающий масштабируемость, необходимую для поддержки растущих, динамических сред. В SQL Server 2000 предусмотрена широчайшая поддержка функций производительности и доступности, гарантирующих своевременное решение поставленных задач, а также развитой функциональности управления и настройки, позволяющей автоматизировать выполнение рутинных задач и снизить совокупную стоимость владения.

SQL Server 2000 обладает рядом возможностей, обеспечивающих легкость установки, развертывания и эксплуатации, а также поддерживающих масштабируемость, создание хранилищ данных и системную интеграцию с другим серверным программным обеспечением.

Механизм баз данных SQL Server 2000 представляет собой надежный сервер, способный управлять базами данных терабайтного объема, к которым одновременно обращаются тысячи пользователей. В то же время при работе с параметрами по умолчанию SQL Server 2000 поддерживает такие функции, как динамическая самонастройка, что позволяет не обременять пользователей решением административных задач.

Некоторые функции SQL Server 2000 увеличивают масштабируемость системы. Например, SQL Server 2000 динамически регулирует степень дробления блокировок для каждой таблицы, на которую ссылается запрос, в него также входит оптимизированная поддержка высокоскоростных операций в средах VLDB. Кроме того, SQL Server 2000 способен планировать параллельное исполнение, при котором обработка оператора SQL разделяется на несколько частей. Каждая часть может быть выполнена на отдельном процессоре, в этом случае формирование полного результирующего набора осуществляется быстрее, чем в том случае, когда отдельные части операторов выполняются последовательно.

SQL Server 2000 состоит из ряда компонентов, таких, как механизм реляционных баз данных, Analysis Services и English Query. Все эти компоненты, каждый из которых играет определенную роль, работая совместно, формируют полнофункциональную реляционную СУБД.

Механизм реляционных баз данных SQL Server 2000 – это современное ядро с высокой степенью масштабируемости, предназначенное для хранения данных. Механизм баз данных сохраняет данные в таблицах. Каждая таблица представляет определенный класс объектов, в зависимости от интересов конкретной организации. Таблица состоит из столбцов, каждый из которых представляет атрибут объекта, который она моделирует, и строк. Каждая строка представляет один экземпляр объекта, моделируемого таблицей. Приложение передает механизму баз данных оператор SQL, механизм возвращает результат в виде набора данных в табличной форме. Интернет-приложение передает механизму баз данных оператор SQL или запрос XPath, а тот возвращает результат в виде документа XML. Механизм реляционных баз данных обеспечивает поддержку стандартных интерфейсов доступа к данным, таких, как ADO, OLE DB и ODBC.

Механизм реляционных баз данных обладает высокой масштабируемостью. SQL Server 2000 Enterprise Edition поддерживает группы серверов баз данных, формирующих базы данных терабайтного объема, к которым могут обращаться тысячи пользователей одновременно. Механизм баз данных также способен динамически настраиваться путем выделения дополнительных ресурсов по мере роста числа пользователей, подключенных к базе данных, и освобождения ресурсов после отключения пользователей. Другими словами, отдельные пользователи или небольшие рабочие группы, у которых нет администраторов баз данных, могут использовать более простые редакции SQL Server. С помощью административных утилит с графическим интерфейсом из комплекта поставки продукта легко администрировать даже крупные серверы баз данных под управлением Enterprise Edition, работающие в эксплуатационном режиме.

Механизм реляционных баз данных также обладает высокой степенью зашиты. Аутентификацию при регистрации допустимо интегрировать с проверкой подлинности Windows, поэтому SQL Server не хранит никаких паролей и не пересылает их по сети. На узлах разрешается задавать аудит всех пользователей, обращающихся к базе данных, соответствующий требованиям безопасности уровня С2, и применять протокол SSL для шифрования всех данных, передаваемых между приложением и базой данных.

Репликация SQL Server 2000 позволяет поддерживать несколько копий данных на различных компьютерах с целью повышения общей производительности системы, а также обеспечивает поддержку синхронизации всех копий. Репликация – важная и мощная технология распределения данных и некоторых типов объектов баз данных по всему предприятию. В репликации SQL Server используется принцип «публикации и подписки». Издатель данных, подлежащих репликации, определяет статьи, которые надо сделать доступными для подписчиков.

Многим организациям для более эффективного принятия решений требуется централизация данных. Однако данные можно хранить в самых разнообразных форматах и в нескольких различных местах. DTS в SQL Server позволяет создавать хранилища и киоски данных путем интерактивного или автоматического импорта и передачи данных из нескольких гетерогенных источников по расписанию.

DTS SQL Server 2000 существенно повышает эффективность процесса создания хранилищ данных для оперативной аналитической обработки. Кроме того, он предоставляет средства для тонкой настройки обширных баз данных для оперативной обработки транзакций, в результате чего можно увеличить число одновременно работающих пользователей, активно добавляющих и модифицирующих данные. Структура баз данных OLTP такова, что они регистрируют подробности каждой транзакции. Попытка выполнить сложный анализ для определения трендов продаж за несколько месяцев или лет потребует просмотра огромного числа записей, а большая загруженность обработкой информации при этом снижает производительность баз данных OLTP.

Хранилища и киоски данных создаются в системе OLTP на основе данных, извлеченных и преобразованных в форму, которая лучше подходит для OLAP-обработки. Периодически осуществляется сбор строк с подробными данными OLTP в промежуточную базу данных, где они обобщаются, а итоговые данные помещаются в хранилище или киоск. DTS поддерживает извлечение данных из одного источника и выполнение сложных преобразований с последующим сохранением итоговых преобразованных данных в другом источнике данных. Этот компонент в значительной степени упрощает процесс извлечения данных из нескольких систем OLTP и создания на основе извлеченных данных хранилища или киоска данных для OLAP.

Analysis Services предоставляет инструменты для анализа данных, которые находятся в хранилищах и киосках данных, где итоговая информация содержится в таблицах фактов. Таблица фактов – центральная таблица в схеме хранилища данных, в ней хранятся численные меры и ключи, связывающие факты с таблицами измерений. Как правило, базовая таблица фактов содержит сведения, описывающие некоторые события в бизнесе, например банковские транзакции или факты продажи продукции. Приложения работают с данными Analysis Services с помощью многомерных расширений ADO и OLE DB. Обработка запросов OLAP посредством многомерных кубов Analysis Services выполняется существенно быстрее, чем с использованием подробной информации из баз данных OLTP.

В систему Analysis Services входит сервер, управляющий многомерными кубами, предназначенными для анализа. Он обеспечивает клиенту быстрый доступ к данным куба. Чтобы быстро выдавать ответы на сложные аналитические запросы. Analysis Services организует данные из хранилища в кубические массивы с помощью предварительно вычисленных агрегированных данных. Analysis Services также облегчает создание моделей извлечения информации для данных как из многомерных, так и из реляционных источников. Можно применять модели извлечения информации к обоим типам данных. Посредством службы PivotTable – компонента доступа, совместимого с OLE DB, Microsoft Excel и приложения других производителей могут получать данные с сервера и представлять их пользователю или создавать локальные кубические массивы для автономного анализа.

## Разработка физической модели данных

На основании концептуальной модели данных, описанной в первой части этой главы, была разработана физическая модель представленная ниже.

На рисунке 2.9 представлены таблицы, относящиеся к бюджетной классификации доходов. В таблице 12 представлено описание таблиц, относящихся к бюджетной классификации доходов. Во всех таблицах поле ts с типом данных timestamp используются для оптимистического управления блокировками при многопользовательской работе с проектируемой системой. Это поле в описания таблиц, в связи с ограничением на объем дипломного проекта, не приводится.

Рисунок 2.9 – Бюджетная классификация доходов

Таблица 12 – Описание физической модели бюджетной классификации доходов

|  |  |  |
| --- | --- | --- |
| **Таблица** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| BudgetClassifications |  | Бюджетные классификации за разные годы |
|  | id | Уникальный идентификатор |
|  | year | Год в течении которого действует бюджетная классификация |
| RevenueGroups |  | Группы доходов |
|  | id | Уникальный идентификатор |
|  | budgetClassificationId | Код бюджетной классификации. Внешний ключ. |
|  | sid | Код группы доходов в соответствии с бюджетной классификацией |
|  | name | Наименование группы доходов |
| RevenueSubgroups |  | Подгруппы доходов |
|  | id | Уникальный идентификатор |
|  | groupId | Код группы доходов. Внешний ключ. |
|  | sid | Код подгруппы доходов в соответствии с бюджетной классификацией |
|  | name | Наименование подгруппы доходов |
| RevenueClauses |  | Статья доходов |
|  | id | Уникальный идентификатор |
|  | subgroupId | Код подгруппы доходов. Внешний ключ |
|  | sid | Код статьи доходов в соответствии с бюджетной классификацией |
|  | name | Наименование статьи доходов |
| RevenueSubclauses |  | Подстатьи доходов |
|  | id | Уникальный идентификатор |
|  | clauseId | Код статьи доходов. Внешний ключ |
|  | sid | Код подстатьи доходов в соответствии с бюджетной классификацией |
|  | name | Наименование подстатьи доходов |
| Elements |  | Элементы бюджетной классификации |
|  | id | Уникальный идентификатор |
|  | budgetClassificationId | Код бюджетной классификации. Внешний ключ |
|  | sid | Код элемента в соответствии с бюджетной классификацией |
|  | name | Наименование элемента |
| RevenuePrograms |  | Программы доходов |
|  | id | Уникальный идентификатор |
|  | budgetClassificationId | Код бюджетной классификации. Внешний ключ |
|  | sid | Код программы в соответствии с бюджетной классификацией доходов |
|  | name | Наименование программы доходов |

На рисунке 2.10 представлены таблицы, относящиеся к бюджетной классификации расходов.

В таблице 13 представлено описание таблиц, относящихся к бюджетной классификации расходов.

На рисунке 2.11 представлены таблицы, относящиеся к бюджетной классификации источников финансирования дефицита.

В таблице 14 представлено описание таблиц, относящихся к бюджетной классификации источников финансирования дефицита.

На рисунке 2.12 представлены таблицы, относящие к процессу формированию доходной части проекта бюджета.

В таблице 15 представлено описание таблиц, относящихся к процессу формирования доходной части проекта бюджета.

Рисунок 2.10 – Бюджетная классификация расходов

Таблица 13 – Описание физической модели бюджетной классификации расходов

|  |  |  |
| --- | --- | --- |
| **Таблица** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| OutlaySections |  | Разделы бюджетной классификации расходов |
|  | id | Уникальный идентификатор |
|  | budgetClassificId | Код бюджетной классификации. Внешний ключ |
|  | sid | Код раздела бюджетной классификации расходов |
|  | name | Наименование раздела бюджетной классификации расходов |

Рисунок 2.11 – Бюджетная классификация источников финансирования дефицита

Таблица 14 – Описание физической модели бюджетной классификации источников финансирования дефицита

|  |  |  |
| --- | --- | --- |
| **Таблица** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| SFDGroups |  | Группы бюджетной классификации источников финансирования дефицита |
|  | id | Уникальный идентификатор |
|  | budgClassifId | Код бюджетной классификации. Внешний ключ |
|  | sid | Код группы источников финансирования дефицита в соответствии с бюджетной классификацией |
|  | name | Наименование группы источников финансирования дефицита |

Рисунок 2.12 – Формирование доходной части проекта бюджета

Таблица 15 – Описание таблиц физической модели данных, относящихся к процессу формировании доходной части проекта бюджета

|  |  |  |
| --- | --- | --- |
| **Таблица** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| Locations |  | Поселения, для которых формируются проекты бюджета |
|  | id | Код поселения |
|  | domains | Код территории к которой относится поселение |
|  | name | Название поселения |
| BudgetProjects |  | Проекты бюджетов |
|  | id | Уникальный идентификатор |
|  | locationId | Код поселения, которому принадлежит проект бюджета |
|  | year | Год, на который составляется проект бюджета |
|  | name | Название проекта бюджета |
|  | status | Состояние проекта бюджета |

На рисунке 2.13 представлены таблицы, относящие к процессу формированию расходной части проекта бюджета.

Рисунок 2.13 – Формирование расходной части проекта бюджета

В таблице 16 представлено описание таблиц, относящихся к процессу формирования расходной части проекта бюджета.

Таблица 16 – Описание таблиц физической модели данных, относящихся к процессу формировании расходной части проекта бюджета

|  |  |  |
| --- | --- | --- |
| **Таблица** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| BCSteward |  | Распорядители бюджетных средств |
|  | id | Уникальный идентификатор |
|  | budgProjId | Код проекта бюджета. Внешний ключ |
|  | legalId | Код юридического лица. Внешний ключ |
|  | sid | Код распорядителя в соответствии с бюджетной классификацией |
| OutlayEstimates |  | Сметы расходов распорядителей бюджетных средств |
|  | bcStewardId | Код распорядителя бюджетных средств |
|  | id | Номер документа |
|  | status | Состояние сметы расходов |
| OutlayEstimateRows |  | Строки табличной части сметы расходов распорядителя бюджетных средств |
|  | id | Уникальный идентификатор |
|  | estimateId | Код сметы расходов. Внешний ключ |
|  | outlayId | Код расхода. Внешний ключ |
|  | sum | Объем денежных средств |
|  | description | Примечание |
| OutlayEnquirys |  | Справки-уведомления по расходам |
|  | id | Номер документа |
|  | bcStewardId | Код распорядителя бюджетных средств. Внешний ключ |
|  | docId | Код документа основания. Внешний ключ |
|  | date | Дата |
|  | description | Примечание |
|  | status | Состояние справки-уведомления |
| OutlayEnquiryRows |  | Строки табличной части справок-уведомлений по расходам |
|  | id | Уникальный идентификатор |
|  | enquiryId | Код справки-уведомления. Внешний ключ |
|  | outlayId | Код расхода. Внешний ключ |
|  | summ | Объем денежных средств |
|  | description | Примечание |

На рисунке 2.14 представлены таблицы, относящие к процессу формированию источников финансирования дефицита бюджета.

В таблице 17 представлено описание таблиц, относящихся к процессу формирования источников финансирования дефицита бюджета.

Рисунок 2.14 – Формирование источников финансирования дефицита бюджета

Таблица 17 – Описание таблиц физической модели данных, относящихся к процессу формировании источников финансирования дефицита бюджета

|  |  |  |
| --- | --- | --- |
| **Таблица** | **Атрибут** | **Описание** |
| 1 | 2 | 3 |
| SFDAdministrators |  | Администраторы источников финансирования дефицита бюджета |
|  | id | Уникальный идентификатор |
|  | legalId | Код юридического лица. Внешний ключ |
|  | budgProjId | Код проекта бюджета. Внешний ключ |
|  | sid | Код администратора источников финансирования дефицита в соответствии с бюджетной классификацией |
| SFDEstimates |  | Сметы источников финансирования дефицита бюджета |
|  | sfdSdminId | Код администратора источников финансирования дефицита бюджета |
|  | id | Номер документа |
|  | status | Состояние сметы |

На рисунке 2.15 представлены таблицы, относящие к процессу формированию консолидированного проекта бюджета территории.

Рисунок 2.15 – Формирования консолидированного проекта бюджета территории

В таблице 18 представлено описание таблиц, относящихся к процессу формирования консолидированного проекта бюджета территории.

Таблица 18 – Описание таблиц физической модели данных, относящихся к процессу формировании консолидированного проекта бюджета территории

|  |  |  |
| --- | --- | --- |
| Таблица | Атрибут | Описание |
| 1 | 2 | 3 |
| Domains |  | Территории |
|  | id | Код территории |
|  | name | Название территории |
| ConsBudgetProjects |  | Консолидированные проекты бюджетов территории |
|  | id | Уникальный идентификатор |
|  | year | Год, на который составляется консолидированный проект бюджета территории |
|  | domainId | Код территории. Внешний ключ |
|  | name | Название проекта бюджета |

В процессе физического проектирования базы данных в среде MS SQL Server 2000 была создана база данных fin\_budget, состоящая из файлов данных fin\_budget.mdf и файлов журналов транзакций fin\_budget\_log.ldf. Принцип отдельного хранения данных и журналов транзакций, а также разбиение этих двух групп информации на различные файлы в SQL Server 2000 необходим для повышения надежности системы.

При создании физической модели сервера баз данных, посредством SQL Server Enterprise Manager, для всех суррогатных ключей было установлено свойство Identy, необходимое для вызова хранимой процедуры аутоинкремента. Для увеличения реактивности системы, индексам, закрепленными за суррогатными ключами присвоено значение Clustered.

На основе модели состояния системы разработана концептуальная модель данных проектируемой системы, включающая описание классов и их атрибутов. В дипломе представлены концептуальные модели бюджетной классификации доходов и расходов, источников финансирования дефицита, доходов администраторов бюджетных средств и доходной части проекта бюджета, смет расходов распорядителей бюджетных средств и расходной части проекта бюджета, смет источников финансирования дефицита бюджета, консолидированного проекта бюджета территории.

В качестве СУБД обосновано применение Microsoft SQL Server 2000 Enterprise Edition.

На основании концептуальной модели данных для Microsoft SQL Server 2000 разработана физическая модель данных.

# Проектирование программного комплекса

##

## Разработка архитектуры программного комплекса

При проектировании системы используется концепция слоев – одна из общеупотребительных моделей, применяемая разработчиками программного обеспечения для разделения сложных систем на более простые части.

Описывая систему в терминах архитектурных слоев, удобно воспринимать составляющие ее подсистемы в виде «слоеного пирога». Слой более высокого уровня пользуется службами, предоставляемыми нижележащим слоем, но тот не «осведомлен» о наличии соседнего верхнего слоя. Более того, обычно каждый промежуточный слой «скрывает» нижний слой от верхнего.

Расчленение системы на слои предоставляет целый ряд преимуществ:

* отдельный слой можно воспринимать как единое самодостаточное целое, не особенно заботясь о наличии других слоев;
* можно выбирать альтернативную реализацию базовых слоев;
* зависимость между слоями можно свести к минимуму;
* созданный слой может служить основой для несколько слоев более высокого уровня.

Архитектура проектируемого приложения основывается на трех основных слоях:

* слой представления
* слой домена
* слой источника данных

Слой представления охватывает все, что имеет отношение к общению пользователя с системой. К главным функциям этого слоя относятся отображение информации и интерпретация вводимых пользователем команд с преобразованием их в соответствующие операции в контексте домена и источника данных.

Слой источника данных *–* это подмножество функций, обеспечивающих взаимодействие со сторонними системами, которые выполняют задания в интересах приложения. Код этой категории несет ответственность за мониторинг транзакций, управление другими приложениями, обмен сообщениями. Слой источника данных управляет обращениями к базе данных, обменом сообщениями, транзакциями.

Логика домена описывает основные функции приложения, предназначенные для достижения поставленной перед ним цели. К таким функциям относятся вычисления на основе вводимых и хранимых данных, проверка всех элементов данных и обработка команд, поступающих от слоя представления, а также передача информации слою источника данных.

Также, при проектировании приложения использовалась концепция подключаемых модулей. Такой подход предполагает постепенное расширение функциональных возможностей системы за счет подключение к базовой архитектуре новых модулей.

В качестве базовой платформы для разработки приложения планируется использовать. Net Framework 2.0.

NET Framework – это управляемая среда для разработки и исполнения приложений, обеспечивающая контроль типов. Эта среда управляет выполнением программы: она выделяет память под данные и команды, назначает разрешения программе или отказывает в их предоставлении, начинает исполнение приложения и управляет его ходом, а также отвечает за освобождение и повторное использование памяти, занятой ресурсами, более ненужными программе..NET Framework состоит из двух основных компонентов: общеязыковой исполняющей среды и библиотеки классов.NET Framework.

CLR можно рассматривать как среду, управляющую исполнением кода и предоставляющую ключевые функции, такие, как компиляция кода, выделение памяти, управление потоками и сбор мусора. Благодаря использованию общей системы типов CLR выполняет строгую проверку типов, а защита по правам доступа к коду позволяет гарантировать исполнение кода в защищенном окружении.

Библиотека классов .NET Framework содержит набор полезных типов, разработанных специально для СLR и доступных для многократного использования. Типы, поддерживаемые.NET Framework, являются объектно-ориентированными, полностью расширяемыми и обеспечивают бесшовную интеграцию приложений с .NET Framework.

Конструкция.NET Framework обеспечивает межъязыковую совместимость. Проще говоря, компоненты, реализованные с применением .NET Framework, способны взаимодействовать друг с другом независимо от языка, на котором они написаны. Так, приложение на Visual Basic.NET может обращаться к DLL, написанной на С#, а та, в свою очередь, способна вызвать ресурсы, созданные на управляемом C++ или любом другом NET-совместимом языке. Межъязыковая совместимость поддерживается и для наследования в ООП, например, на основе С#-класса можно объявлять классы в программах на Visual Basic.NET и наоборот.

В качестве языка программирования, при помощи которого, будет реализовываться проектируемая система, выбран язык C#. Этот выбор обусловлен, прежде всего, тем, что данный язык разрабатывался специально для платформы. Net Framework. Он сочетает в себе мощь C++ и простоту Visual Basic.

Для доступа к данным используется набор библиотек ADO. Net.

ADO. Net – это набор библиотек, поставляемых с Microsoft. Net Framework и предназначенный для взаимодействия с различными хранилищами данных из. Net-приложений. Библиотека ADO. Net включают классы для подсоединения к источнику данных, выполнения запросов и обработки их результатов. Кроме того, ADO. Net можно использовать в качестве надежного, иерархически организованного, отсоединенного кэша данных для автономной работы с данными. Главный отсоединенный объект, DataSet, позволяет сортировать, осуществлять поиск, фильтровать, сохранять отложенные изменения и перемещаться по иерархическим данным. Кроме того, объект DataSet включает ряд функций, сокращающих разрыв между традиционным доступом к данным и программированием с использованием XML.

Для взаимодействия между удаленными узлами, на которых расположены компоненты проектируемого приложения, используется технология. Net Remoting /17, 22/.

NET Remoting – это объектно-ориентированная архитектура для поддержки распределенных приложений в Microsoft .NET. Подобно тому как .NET Framework заменяет СОМ в качестве средства разработки компонентов .NET Remoting заменяет DCOM в качестве средства создания распределенных приложений на основе.NET Framework. Более того .NET Remoting является основой для .NET Web-сервисов. Таким образом, понимание основ .NET Remoting совершенно необходимо для разработки на основе .NЕТ Framework распределенных приложений, в том числе для Интернета.

NET Remoting позволяет объектам, исполняющимся внутри разных доменов приложений и контекстов, взаимодействовать друг с другом через границы .NET Remoting. Граница .NET Remoting ведет себя, как полупроницаемая мембрана: в некоторых случаях она позволяет экземпляру пройти сквозь нее без изменений; в других – заставляет экземпляр объекта за пределами домена или контекста взаимодействовать с внутренними объектами по строго определенному протоколу.

## Разработка прототипов пользовательского интерфейса

Пользовательский интерфейс клиентской части приложения выполнен в виде единого интегрированного Windows-приложения с многодокументным интерфейсом. Клиентская часть поддерживает подключаемые модули, которые могут расширять ее функциональные возможности. При этом новые компоненты интегрируются в среду, обеспечивая единый интерфейс для работы.

На рисунке 3.1 представлена среда автоматизированной системы бюджетного процесса.

Рисунок 3.1 – Среда автоматизированной системы бюджетного процесса

Пользовательская среда автоматизированной системы бюджетного процесса состоит из нескольких основных частей:

* главное меню приложения;
* строка навигации по доступным проектам бюджета;
* панель навигации;
* основная часть.

Главное меню приложения служит для доступа ко всем функциям системы.

Строка навигации по доступным проектам бюджета служит для выбора поселения и проекта бюджета на соответствующий год, с которым будет вестись работа.

Панель навигации дублирует основные функции главного меню приложения для облегчения и ускорения доступа пользователя.

В основной части среды, выполненной в виде панели с закладками, открываются формы, непосредственно с которыми работает пользователь.

Структура главного меню приложения представлена в таблице 19.

Таблица 19 – Структура главного меню

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Название** | **Описание** |
| 1 | 2 | 3 |
| 1 | Файл |  |
| 1.1 | Войти в системы | Позволяет пользователю подключится к системе |
| 1.2 | Выйти из системы | Позволяет пользователю отключится от системы |
| 1.3 | Выход | Завершает работу приложения |
| 2 | Правка |  |
| 2.1 | Отменить | Позволяет отменить последнее произведенной пользователем действие |
| 2.2 | Повторить | Позволяет повторить отмененное ранее действие |
| 2.3 | Скопировать | Позволяет скопировать данные в буфер обмена |
| 2.4 | Вырезать | Позволяет вырезать данные в буфер обмена |
| 2.5 | Вставить | Позволяет вставить данные из буфера обмена |
| 3 | Проект бюджета |  |
| 3.1 | Доходы |  |
| 3.1.1 | Сметы доходов | Администраторам бюджетных средств позволяет вводить и передавать в Финансовое управление сметы доходов, а работникам Финансового управления проверять и утверждать сметы |
| 3.1.2 | Справки-уведомления | Администраторам бюджетных средств позволяет вводить справки-уведомления по доходам, а работникам Финансового управления проверять и утверждать их |
| 3.2 | Расходы |  |
| 3.2.1 | Сметы расходов | Распорядителям бюджетных средств позволяет вводить и передавать в Финансовое управление сметы расходов, а работникам Финансового управления проверять и утверждать сметы |
| 3.2.2 | Справки-уведомления | Распорядителям бюджетных средств позволяет вводить и передавать в Финансовое управление справки-уведомления по расходам, а работникам Финансового управления проверять и утверждать их |

Следует отметить, что пользовательский интерфейс автоматически настраивается в соответствии с правами доступа текущего пользователя, то есть пользователю отображаются только доступные ему функции.

## Проектирование структуры программного

В процессе анализа требований и объектно-ориентированного анализа предметной области основное внимание уделялось правильной организации деятельности, т.е. изучению основных целей построения автоматизированной системы бюджетного процесса. На данном этапе акцент смещается в сторону правильной реализации поставленных целей, т.е. разработке грамотного проектного решения, удовлетворяющего поставленным требованиям. И здесь на помощь приходят диаграммы взаимодействий, иллюстрирующие взаимодействия объектов в процессе выполнения системных требований /18/. Они помогают определить структуру приложения, т.е. выявить классы системы и их взаимосвязи.

На рисунке 3.2 представлена диаграмма, показывающая объекты и сообщения, передаваемые между ними в процессе аутентификации пользователя в системе.

Пользователь вводит логин и пароль своей учетной записи и нажимает на кнопку «Войти» формы AuthForm. Эта форма в свою очередь отправляет сообщение Connect управляющему объекту ClientImpl, который представляет собой клиентское приложение. Объект ClientImpl вызывает метод Connect объекта-сервера ServerImpl и передает себя в качестве параметра этого сообщения. ServerImpl проверяет наличие в системе зарегистрированной учетной записи, с введенными пользователем логином и паролем, при помощи управляющего объекта SecurityManager. Если учетная запись с введенными пользователем данными зарегистрирована в системе, то сервер регистрирует сессию для клиентского приложения и возвращает управление клиентскому приложению.

Рисунок 3.2 – Аутентификация пользователя в системе

На рисунке 3.3 представлена диаграмма, показывающая объекты и сообщения, передаваемые между ними в процессе регистрации нового пользователя в системе.

Администратор вводит на форме RegisterUserForm необходимые для регистрации нового пользователя данные, а затем запускает процедуру регистрации нового пользователя в системе. Форма RegisterUserForm передает сообщение Register управляющему объекту UserManager вместе с введенными администратором данными.

UserManager проверяет корректность введенных данных и добавляет нового пользователя в систему.

Рисунок 3.3 – Регистрация нового пользователя в системе

На рисунке 3.4 представлена диаграмма, показывающая объекты и сообщения, передаваемые между ними в процессе смены пароля на учетной записи.

Пользователь вводит свой старый пароль и новый, и запускает процедуру смены пароля. Форма ChangePasswordForm передает сообщение ChangePass управляющему объекту UserManager, который в свою очередь при помощи SecureManager проверяет наличие у пользователя права на смену пароля на текущей учетной записи. Если у пользователя это право имеется, то SecureManager посылает сообщений объекту Users, который меняет пароль на учетной записи.

В рамках дипломного проекта был разработан полный комплект диаграмм взаимодействия для всех основных и альтернативных сценариев прецедентов, но из-за ограничений накладываемых на размер дипломного проекта их не представляется возможным привести все здесь, поскольку это увеличило бы объем проекта на несколько сотен страниц.

Рисунок 3.4 – Смена пароля на учетной записи пользователя

После построения диаграмм взаимодействия можно наконец-то представить систему в виде совокупности классов и взаимосвязей между ними. Для предоставления подобного рода информации служат диаграммы классов. В приложении В представлены основные классы объектов составляющие автоматизированную систему бюджетного процесса, а в приложении Г приведены фрагменты кода.

Дадим некоторые пояснения по структуре приложения.

Центральными классами являются ServerImpl и ClientImpl, которые соответственно представляют серверную и клиентскую часть приложения. Первый объект реализует интерфейс IServer, а второй IClient. Клиентская часть приложения имеет доступ к серверу только через интерфейс IServer. Такая реализация продиктована требованиями безопасности.

При проектировании класса ServerImpl использовался паттерн Singleton, что гарантирует существование единственного экземпляра этого класса в течение всего времени работы программы.

Класс ServerImpl управляет загрузкой подключаемых модулей и предоставляет доступ клиентскому коду к управляющим объектам.

Подключаемые модули поставляются в виде динамически подключаемых библиотек. В каждой библиотеке определен один или несколько классов реализующих интерфейс IPlugin. Класс ServerImpl при запуске приложения просматривает динамически подключаемые библиотеки на наличие классов реализующих интерфейс IPlugin и в случае если такие классы обнаружены, создает объекты этих классов и вызывает метод Load. В этом методе подключаемый модуль проводит необходимую ему инициализацию и регистрирует управляющие классы объектов в системе.

Все управляющие объекты реализуют интерфейс IManager и доступ к ним осуществляется по названию менеджера через класс ServerImpl.

Рассмотрим назначение основных менеджеров.

SecurityManager агрегирует в себе функция связанные с безопасностью. К числу таких функций относится аутентификация пользователя в системе, определение прав доступа пользователя и т.д.

SessionManager управляет подключенными клиентскими приложениями. При подключении нового клиента к серверу ему автоматически присваивается сессия и вся дальнейшая работа строится на использовании этой сессии.

UserManager служит для управления пользователями. Он позволяет регистрировать новые учетные записи пользователей, удалять их и изменять регистрационные данные.

PermissionManager служит для управления правами доступа пользователей и пользовательских групп. С его помощью можно назначать или снимать права доступа с пользователя или пользовательской группы.

DatabaseManager управляет подключениями к базе данных, а также транзакциями.

DALManager управляет компонентами доступа к данным. При помощи этого менеджера в системе регистрируются DAL‑компоненты, которые осуществляют сохранение объектов в базе данных, их модификацию и удаление, а также построение объектов из данных сохраненных в базе.

BudgetManager управляет проектами бюджета. Позволяет создавать, модифицировать и утверждать проекты бюджета.

BCAdminManager управляет администраторами бюджетных средств. Позволяет назначать и удалять администраторов бюджетных средств.

BCStewardManager управляет распорядителями бюджетных средств. Позволяет назначать и удалять распорядителей бюджетных средств.

SFDAdminManager управляет администраторами источников финансирования дефицита. Позволяет назначать и удалять администраторов источников финансирования дефицита.

RevenueEstimateManager управляет сметами доходов администраторов бюджетных средств.

OutlayEstimateManager управляет сметами расходов распорядителей бюджетных средств.

SFDEstimateManager управляет сметами источников финансирования дефицита бюджета.

RevenueEnquiryManager управляет справками-уведомлениями по доходам.

OutlayEnquiryManger управляет справками-уведомлениями по расходам.

SFDEnquiryManager управляет справками-уведомлениями по источникам финансирования дефицита бюджета.

BCManager управляет объектами бюджетной классификации. Позволяет формировать справочники бюджетной классификации, вносить в них изменения.

ExchangeManager управляет обменом информацией между Финансовым управлением, администраторами и распорядителями бюджетных средств, а также администраторами источников финансирования дефицита бюджета.

Разработанные классы объединены в компоненты, представленные в таблице 20.

Таблица 20 – Компоненты автоматизированной системы бюджетного процесса

|  |  |
| --- | --- |
| **Название компонента** | **Описание** |
| 1 | 2 |
| ASBPServer.exe | Серверная часть приложения, выполненная в виде Windows‑процесса |
| ASBPClient.exe | Клиентская часть приложения |
| ASBP. Common.dll | Динамическая библиотека, объединяющая базовые классы, используемые большинством компонентов автоматизированной системы бюджетного процесса |
| ASBP. BudgetProj. Server.dll | Динамическая библиотека, объединяющая классы серверной части приложения, которые используются при формировании проекта бюджета |
| ASBP. BudgetProj. Client.dll | Динамическая библиотека, объединяющая классы клиентской части приложения, которые используются при формировании проекта бюджета |
| ASBP. Exchange. Server.dll | Динамическая библиотека, объединяющая классы серверной части приложения, которые используются при обмене данными между Финансовым управлением, администраторами и распорядителями бюджетных средств, а также администраторами источников финансирования дефицита бюджета |
| ASBP. Exchange. Client.dll | Динамическая библиотека, объединяющая классы клиентской части приложения, которые используются при обмене данными между Финансовым управлением, администраторами и распорядителями бюджетных средств, а также администраторами источников финансирования дефицита бюджета |
| ASBP. ConsBudgProj. Server.dll | Динамическая библиотека, объединяющая классы серверной части приложения, которые участвуют в процессах составления консолидированного проекта бюджета территории |
| ASBP. ConsBudgProj. Client.dll | Динамическая библиотека, объединяющая классы клиентской части приложения, которые участвуют в процессах составления консолидированного проекта бюджета территории |
| ASBP. Revenue. Server.dll | Динамическая библиотека, объединяющая классы серверной части приложения, которые используются при формировании смет доходов администраторов бюджетных средств и справок-уведомлений по доходам |
| ASBP. Revenue. Client.dll | Динамическая библиотека, объединяющая классы клиентской части приложения, которые используются при формировании смет доходов администраторов бюджетных средств и справок-уведомлений по доходам |
| ASBP. Outlay. Server.dll | Динамическая библиотека, объединяющая классы серверной части приложения, которые используются при формировании смет расходов распорядителей бюджетных средств и справок-уведомлений по расходам |
| ASBP. Outlay. Client.dll | Динамическая библиотека, объединяющая классы клиентской части приложения, которые используются при формировании смет расходов распорядителей бюджетных средств и справок уведомлений по расходам |
| ASBP.SFD. Server.dll | Динамическая библиотека, объединяющая классы серверной части приложения, которые используются при формировании смет источников финансирования дефицита бюджета и справок-уведомлений по источникам финансирования дефицита бюджета |
| ASBP.SFD. Client.dll | Динамическая библиотека, объединяющая классы клиентской части приложения, которые используются при формировании смет источников финансирования дефицита бюджета и справок-уведомлений по источникам финансирования дефицита бюджета |

## Тестирование программной системы

Процесс тестирования представляет собой эксплуатацию приложения в контролируемых условиях и изучение полученных результатов /31/. При этом проверяется работа приложения с нормальными и ошибочными данными и событиями. Следует изучить и реакцию на неожиданные ситуации.

В конечном счете, тестирование проводится не только для поиска ошибок, но и для проверки качества продукта. А так как качество – это «соответствие потребностям пользователей в решении их бизнес-задач», процесс тестирования должен способствовать достижению этой цели с помощью проверки корректности работы программы.

Для тестирования приложения, разработанного в рамках дипломного проекта, был составлен комплекс тестов, охватывающий все аспекты функционирования системы. Из-за ограничений, накладываемых на объем дипломного проекта, здесь приводится лишь несколько тестов.

Первое испытание направлено на тестировании работы автоматизированной системы бюджетного процесса по сохранению вводимых данных при вводе справок-уведомлений по доходам.

Рисунок 5.1 – Состояние таблицы RevenueEnquirys до внесения справки в систему

Рисунок 5.2 – Состояние таблицы RevenueEnquiryRows до внесения справки в систему

На рисунках 5.3 и 5.4 представлено состояние этих же таблиц после внесения справки-уведомления.

Рисунок 5.3 – Состояние таблицы RevenueEnquirys после внесения справки в систему

Рисунок 5.4 – Состояние таблицы RevenueEnquiryRows после внесения справки в систему

На основании приведенных выше данных можно сделать вывод о том, что данная функциональность системы работает корректно.

Следует отметить, что подобные тесты были проведены для всех сущностей, которые необходимо хранить в базе данных. Все эти тесты дали положительный результат, т.е. система корректно обрабатывает сохранения данных на постоянных носителях.

Вторым типом тестов, которые проводились с системой, являются тесты на корректность бизнес-логики приложения.

В качестве примера здесь приведем тест, в котором показывается, что внесение справок уведомлений по доходам, правильно влияет на состояние сметы доходов.

Из представленных данных видно, что приложение корректно обрабатывает бизнес-логику этой задачи, и внесение справок-уведомлений по доходам влияет на состояние смет доходов администраторов бюджетных средств.

Подобные тесты были проведены для всего комплекса задач, и после некоторой корректировки исходных текстов автоматизированной информационной системы бюджетного процесса они были пройдены успешно.

В процессе разработки программного обеспечения информационной системы обоснован выбор архитектуры проектируемого приложения включающей уровни представления, домена и источника данных. При проектировании приложения использовалась концепция подключаемых модулей, что предполагает постепенное расширение функциональных возможностей системы за счет подключение к базовой архитектуре новых модулей.

В качестве базовой платформы для разработки приложения использована платформа. Net Framework 2.0, которая является управляемой средой для разработки и исполнения приложений.

В качестве языка программирования, при помощи которого, реализована проектируемая система, выбран язык C#.

Для доступа к данным используется технология ADO. Net, которая представляет собой набор библиотек, поставляемых с Microsoft. Net Framework, и предназначена для взаимодействия с различными хранилищами данных из. Net-приложений.

Для взаимодействия между удаленными узлами, на которых расположены компоненты проектируемого приложения, используется технология. Net Remoting, которая является объектно-ориентированной архитектурой для поддержки распределенных приложений в Microsoft.NET.

Пользовательский интерфейс клиентской части приложения выполнен в виде единого интегрированного Windows-приложения с многодокументным интерфейсом.

В процессе проектирования приложения разработаны диаграммы взаимодействий, иллюстрирующие взаимодействия объектов в процессе выполнения системных требований и диаграммы классов. В рамках дипломного проекта разработан полный комплект диаграмм взаимодействия для всех основных и альтернативных сценариев прецедентов.

Программное обеспечение информационной системы разработано на языке C#, проведено тестирование системы.

# Управление информационным проектом

##

## Выбор жизненного цикла разработки информационной системы

Жизненный цикл информационной системы представляет собой непрерывный процесс, начинающийся с момента принятия решения о создании информационной системы и заканчивающийся в момент полного изъятия ее из эксплуатации.

Модель жизненного цикла информационной системы представляет собой некоторую структуру, определяющую последовательность осуществления процессов, действий и задач, выполняемых на протяжении её жизненного цикла, а также взаимосвязи между этими процессами, действиями и задачами. Наиболее известными жизненными циклами разработки ИС можно назвать следующие: каскад, V-образное эволюционное ускоренное прототипирование, быстрая разработка приложений, инкрементная и спиральная модели.

В каскадной модели предусмотрена последовательная организация работ. При этом основной особенностью является разбиение всей разработки на этапы, причем переход с одного этапа на следующий происходит только после того, как полностью завершены все работы на предыдущем этапе. Каскадная модель применяется в проектах, когда требования и их реализация максимально четко определены и понятны.

V-образная модель является разновидностью каскадной модели. В этой модели особое значение придается действиям, направленным на верификацию и аттестацию системы. Тестирование системы обсуждается, проектируется и планируется на ранних этапах жизненного цикла разработки. Область применения V-образной модели – когда информация о требованиях достаточно полная. Модифицированная V-образная модель включает в себя итерационные циклы внесения изменений в требования.

Модель прототипирования жизненного цикла информационной системы предполагает создание легко поддающихся модификации и расширению рабочих моделей системы. Этот подход предполагает участие конечного пользователя в течение всего процесса разработки. Процесс уточнения продолжается до тех пор пока пользователь не получит требуемую функциональность.

Основной чертой метода RAD является короткое время перехода от определения требований до создания полной системы. Метод предполагает привлечение конечного пользователя на каждой фазе проекта основывается на последовательности итераций эволюционной системы или прототипов, критический анализ которых обсуждается с заказчиком. Период итерации, как правило, 60 дней. Обязательным является применение CASE-технологий.

Инкрементная модель представляет собой процесс частичной реализации всей системы и медленного наращивания функциональных возможностей. Данная модель описывает процесс, при выполнении которого первостепенное внимание уделяется системным требованиям, а затем их реализации разработчиками.

Спиральная модель предполагает итерационный процесс разработки информационной системы. При этом возрастает значение начальных этапов жизненного цикла, таких как анализ и проектирование. На этих этапах проверяется и обосновывается реализуемость технических решений путем создания прототипов. Каждая итерация представляет собой законченный цикл разработки, приводящий к выпуску внутренней и внешней версии изделия, которое совершенствуется от итерации к итерации, чтобы стать законченной системой. Таким образом, каждый виток спирали соответствует созданию фрагмента или версии программного изделия, на нем уточняются цели и характеристики проекта, определяется его качество, планируются работы на следующем витке спирали. На каждой итерации углубляются и последовательно конкретизируются детали проекта, в результате чего выбирается обоснованный вариант, который доводится до окончательной реализации.

Использование спиральной модели позволяет осуществлять переход на следующий этап выполнения проекта, не дожидаясь полного завершения текущего – недоделанную работу можно будет выполнить на следующей итерации. Главная задача каждой итерации – как можно быстрее создать работоспособный продукт, который можно показать пользователям системы. Таким образом, существенно упрощается процесс внесения уточнений и дополнений в проект.

Приемлемая модель жизненного цикла разработки программного обеспечения для проекта выбирается на основе анализа отличительных категорий проекта, которые включают в себя анализ требований, команды разработчиков, коллектива пользователей, типа проекта и рисков.

В данном случае анализируются требования, коллектив пользователей, тип проекта и риски. Матрицы, предназначенные для осуществления процесса выбора модели жизненного цикла итогам исследования отличительных категорий проекта были получены результаты, представленные в таблице 4.1.

Таблица 20 – Результаты выбора приемлемой модели жизненного цикла разработки программного обеспечения

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Характеристика | Каскадная | V-образная | Прототипирование | Спиральная | RAD | Инкрементная |
| Требования | 2 | 5 | 2 | 5 | 4 | 2 |
| Участники команды разработчиков | 6 | 6 | 4 | 6 | 3 | 6 |
| Коллектив пользователей | 1 | 2 | 0 | 3 | 1 | 2 |
| Типы проектов и рисков | 3 | 4 | 6 | 8 | 7 | 6 |
| Итого | 12 | 17 | 12 | 22 | 15 | 16 |

Анализ показал, что наиболее приемлемым в данном случае является выбор спиральной модели жизненного цикла разработки программного обеспечения. Данная модель обеспечивает потребности организации, а также соответствует типу выполняемых работ.

## Определение цели и области действия программного проекта

При выполнении каждого проекта определяется, как минимум, одна цель. Для большинства проектов характерны несколько целей. Иногда эти цели именуются заданиями проекта, либо используется собирательное название «миссия проекта». В данном случае миссия проекта эквивалентна достижению целей и заданий проекта. Определение ясной и четкой миссии проекта является одним из простейших и наиболее экономных действий, осуществляемых при разработке всего программного проекта. Менеджмент программных проектов чаще всего связан с менеджментом ожиданий и предсказанием рисков.

Важно определить общую цель проекта, выраженную в легко понимаемых и воспроизводимых терминах. Благодаря осознанию цели достигается компактное определение проекта.

Цель данного проекта: обеспечение финансовых органов муниципальных образований эффективным и удобным инструментом, позволяющим упростить процессы управления бюджетными средствами и повысить прозрачность этих процессов посредством создания информационной системы.

Относительно данного проекта можно сказать, что он будет:

* внутренним;
* **предназначен для автоматизации процесса планирования бюджета**;
* использоваться в ф**инансовом управление администрации Новоегорлыкского сельского поселения**.

Проект не будет:

* предназначаться для использования всеми отделами **администрации Новоегорлыкского сельского поселения**;
* предназначенным для обеспечения доступа к информации не из **администрации**.

## Создание структуры пооперационного перечня работ

Под операцией понимается деятельность или процесс, выполнение которых требует некоторых временных и материальных затрат.

Структура пооперационного перечня работ представляет собой инструмент, применяемый для документирования всех рабочих операций, которые должны быть выполнены при разработке и поставке программного обеспечения надлежащим образом. При использовании такой структуры разработчикам проекта значительно проще разделить весь рабочий процесс на ряд небольших, хорошо определенных задач и действий. В частности, при наличии структуры пооперационного перечня работ облегчается планирование, в том числе календарное, и оценивание. Подобная структура представляет собой основу для осуществления мониторинга проекта, а также для создания хронологической коллекции данных.

Структура пооперационного перечня работ представляет собой иерархический перечень рабочих действий, необходимых для завершения проекта. В этот перечень включаются управленческие, административные, интегральные и программистские действия, с помощью которых:

* выполняется разработка программного обеспечения;
* происходит управление проектом;
* обеспечивается поддержка для всех действий, выполняемых в ходе осуществления проекта;
* выполняются любые другие действия, требуемы для достижения целей проекта и удовлетворения требований пользователей.

Структура пооперационного перечня работ создания информационной системы представлена на рисунке 4.1.

Структура пооперационного перечня работ представляет собой описание выполняемой работы, разбитой на отдельные ключевые компоненты вплоть до самого нижнего уровня. Таким образом, при разбиении проекта на отдельные управляемые части размер каждого компонента может быть изменен, а также возможна оценка трудозатрат, понесенных на этапе разработки этого компонента.

Рисунок 4.1 – Схема декомпозиции работ

Структура пооперационного перечня работ является ключевым рабочим продуктом, необходимым при выполнении оценок в рамках программного проекта. Для каждого различного жизненного цикла существует уникальных пооперационный перечень работ, который может использоваться в самой организации.

## Идентификация задач и действий

Действие – элемент работы, выполняемой в ходе осуществления проекта. Для действия характерна ожидаемая длительность и затраты, а также прогнозируемые требования к ресурсам. Диаграммы являются графическим средством отображения содержащейся в проектном файле информации. Из диаграмм можно получить визуальное представление о последовательности задач, их относительной длительности и длительности проекта в целом.

В качестве программы управления проектами была выбрана Microsoft Office Project 2007. В MS Project предусмотрен обширный набор возможностей по гибкому конфигурированию вида ленточных диаграмм.

В рамках программы MS Project задача – это одно из мероприятий, направленных на достижение цели проекта; основными параметрами задачи являются даты начала и завершения, длительность, трудоемкость, а также виды и количество ресурсов, необходимых для ее выполнения. Каждая задача в пределах проекта должна иметь уникальное имя.

Microsoft Project обеспечивает управление ходом работ на всем жизненном цикле проекта, помогая завершить его в срок, в рамках бюджета и с надлежащим качеством.

Простейшим инструментом планирования и управления проектом является визуальный анализ его графика. Одним из способов визуального представления проектов являются диаграммы Ганта. Они представляют собой исторически один из первых и весьма эффективный метод оперативно-календарного планирования.

В MS Project диаграмма Ганта представляет собой график, на котором по горизонтали размещена шкала времени, а по вертикали расположен список задач. При этом длина отрезков, обозначающих задачи, пропорциональна длительности задач.

Проект представлен в виде графика Ганта на рисунке 4.2.

Слева на диаграмме представлен перечень операций, на которые разбивается проект. Справа на горизонтальной шкале времени откладываются сроки начала и окончания операций. Соответственно, размеры линий графика, отражающих отдельные операции, пропорциональны их продолжительности.

Визуально диаграмма Ганта представляет собой последовательность действий, выполняемых в рамках данного проекта.

Рисунок 4.2 – Диаграмма Ганта. Лист 1

## Распределение ресурсов проекта

Для управления ресурсами проекта в MS Project имеется коллекция представлений проекта, которые специально ориентированы на решение задач данного типа.

Процесс назначения ресурсов задачам проекта называют ресурсным планированием проекта.

Ресурсное планирование позволяет:

* оценить потребность в ресурсах конкретного типа;
* спланировать рациональное распределение потребности в ресурсах во времени;
* определить участки проекта, являющиеся критическими с точки зрения потребностей в ресурсах;
* оценить суммарную стоимость проекта;
* контролировать расходование ресурсов при реализации проекта.

Планирование ресурсов начинается с определения состава ресурсов, то есть составления списка людей и оборудования, необходимого для выполнения проектных работ. Работа со списком ресурсов осуществляется в сводной таблице загрузки ресурсов проекта.

Ресурсы проекта представлены в сводной таблице загрузки ресурсов.

Рисунок 4.4 – Сводная таблица загрузки ресурсов

Представление ресурсов в такой форме позволяет оперативно управлять атрибутами ресурса, и выявлять те ресурсы, по которым допускается превышение ограничения на доступный объем ресурса.

Определение набора ресурсов позволяет построить детальную таблицу загрузки ресурсов. Данное представление позволяет проводить анализ количественного распределения отдельно взятого ресурса по различным этапам проекта.

Рисунок 4.5 – Детальная таблица загрузки ресурсов

График использования ресурсов позволяет получить подробную информацию о том, какой вклад в трудоемкость каждой операции вносит каждый ресурс как в течение всего проекта, так и в течение отдельных периодов его реализации.

Уровень интеграции периодов реализации проекта может определятся пользователем.

Статистика проекта показывает дату начала и окончания проекта, его длительность, количество рабочих часов, стоимость работ.

Рисунок 4.6 – График использования ресурсов специалиста по вариантам использования

Рисунок 4.7 – Статистика проекта

Таким образом, общая стоимость работ составляет 93900 рублей.

## Расписание проектных работ

Сетевая диаграмма представляет совокупность операций проекта в виде логической схемы типа вершинного графа. Фрагмент сетевой диаграммы представлен на рисунке 4.8.

Рисунок 4.8 – Фрагмент сетевой диаграммы

На сетевой диаграмме операции изображаются с помощью прямоугольников, а связи межу ними – с помощью стрелок. В общем виде сетевая диаграмма представляет собой набор узлов и стрелок. На такой диаграмме легко проследить порядок следования действий слева направо и увидеть взаимосвязь между различными последовательностями узлов.

Сетевая диаграмма не достаточно информативна при решении задач, требующих оценки временных характеристик проекта, однако полезна при структурно-логическом анализе.

На фрагменте сетевой диаграммы представлена последовательность этапов анализа требований заказчика, куда входят анализ предметной области, анализ существующих решений, сбор требований, а также не представленные спецификация требований и выбор методологии проектирования.

В сетевой диаграмме содержится следующая информация:

* название действия или идентификатор узла;
* время наискорейшего начала производственного этапа, которое определяется на основании времени завершения предыдущего действия или какого-либо другого ограничения;
* время быстрейшего завершения действия;
* продолжительность этапа;
* максимально возможный срок, когда действие может быть начато, не затронув стадию следующего действия;
* максимально возможный срок, когда действие может быть завершено, не затронув стадию следующего действия.

Недостаток этих диаграмм в том, что в больших проектах с огромным количеством действий они становятся очень громоздкими и неудобочитаемыми.

## Оценка размера и возможности повторного использования ПО

Измерение размера, оценка и составление графика сложным образом переплетаются в процессе планирования проекта. Каждое из этих действий проекта может быть выполнено с помощью множества различных методик.

При выполнении операции определения размеров используются пять весьма полезных и широко известных технологий:

* подсчет сток кода в качестве единиц оценки размера;
* подсчет функциональных точек в качестве единиц оценки размера;
* подсчет точек свойств в качестве единиц оценки размера;
* применение блиц-модели;
* применение модели Wideband Delphi.

Повторное использование существующих компонентов не всегда возможно в полном объеме. Особенности существующих компонентов может не допускать обеспечение качества и возможность повторного использования. Выход состоит в использовании набора весовых множителей, произведенных на основе эмпирических правил, с целью их применения при оценке процесса повторного использования.

В разработанной программной системе использовался архитектурный подход изоляции слоев приложения и технология паттернов проектирования, что закладывает предпосылки повторного использования кода. Так серверная и клиентская части приложения взаимодействуют через интерфейсы IServer и IClient. При проектировании класса ServerImpl использовался паттерн Singleton, что поддерживает режим повторного использования кода. Подключаемые модули поставляются в виде динамически подключаемых библиотек, которые могут использоваться в других проектах. В каждой библиотеке определен один или несколько классов реализующих интерфейс взаимодействия. Все управляющие объекты реализуют интерфейс IManager и доступ к ним осуществляется по названию менеджера через класс ServerImpl.

## Оценка экономической эффективности проекта

Оценка эффективности вложения средств в информационную систему для государственных и муниципальных организаций – одна из самых сложных задач.

Многие специалисты считают, что возможна только качественная оценка эффективности внедрения таких информационных системы, что принципиально невозможно оценить эффективность вложения средств в цифрах и существуют только косвенные показатели, по которым можно судить об эффективности вложений /29/. Если в результате внедрения информационной системы повысилась эффективность планирования бюджетного процесса муниципальной организации, то на момент оценки этого, вполне возможно, еще не видно. В результате внедрения программного комплекса информационной система бюджетного процесса ф**инансовое управление администрации Новоегорлыкского сельского поселения** удастся достичь следующих результатов:

* увеличить эффективность управления бюджетными средствами;
* повысить прозрачность процессов управления;
* упростить и ускорить процесс составления проекта бюджета.

При разработке информационной системы было принято предположение, что её внедрение будет способствовать условному высвобождению трудовых ресурсов, что обеспечит косвенный эффект.

Для оценки проекта используем метод расчета **чистого приведенного дохода**, который предусматривает дисконтирование денежных потоков: доходы и затраты приводятся к одному моменту времени.

Центральным показателем является показатель **NPV** – текущая стоимость денежных потоков за вычетом текущей стоимости денежных оттоков. Это обобщенный конечный результат инвестиционной деятельности в абсолютном измерении.

При разовой инвестиции расчет чистого приведенного дохода можно вычислить по следующей формуле:

Формула расчета чистого приведенного дохода:

где – чистый приведенный доход;

n – месяц;

– величина денежного потока в течение n месяцев, рубли;

– ставка дисконтирования;

 – стартовые инвестиции, рубли;

Формула для расчета величины денежного потока:

где – величина денежного потока, рубли;

– выгоды от реализации проекта;

– суммарные ежемесячные затраты на реализацию проекта;

Затраты на реализацию проекта в первом месяце составляют 20000 рублей, во втором – 25000 рублей, в третьем – 30000 рублей, в четвертом т – 19000 рублей. Ставка дисконтирования равна 10,5%.

В таблице 21 представлен денежный поток за каждый месяц в отдельности.

Таблица 21 – Денежный поток по каждому месяцу

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Месяц** | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| **R**k | **-**20000 | **-**25000 | **-**30000 | **-**10000 | 25000 | 30000 | 35000 |

NPV=2170 рублей.

Поскольку величина чистой текущей стоимости 2170 рублей, то есть *NPV* > 0, то проект можно принять.

Коэффициент возврата инвестиций ROIпозволяет оценить прибыльность инвестиций, вложенных в проект.

Формула для расчета коэффициента возврата инвестиций:

где **–** чистый приведенный доход;

 – стартовые инвестиции, рубли;

При ROI > 100% – инвестиции прибыльны, при ROI < 100% – инвестиции убыточны.

.

Поскольку ROI = 102%, то есть ROI > 100% – инвестиции прибыльны.

По результатам приведенных расчетов проект эффективен, поэтому целесообразно его реализовать.

Расчет экономической эффективности проекта показал прибыльность инвестиций в случае принятия данного проекта. По этим результатам можно сделать вывод, что проект выгоден для организации и окупится в течении четырех месяцев с момента начала эксплуатации программного обеспечения.

Анализ экономической эффективности проекта помогает принять правильное решение относительно проекта, опираясь на конкретные данные о его убыточности или прибыльности.

Построение диаграмм позволяет наглядно представить график работ, изобразить последовательность действий, выполняемых в рамках данного проекта.

Общая картина проекта содержится в плане разработки программного продукта, который описывает способы достижения целей, поставленных перед проектом. Рабочий график является только одним из элементов этого плана и отображает взаимоотношения между всеми производственными процессами, непосредственно связан с реальным календарем и помогает правильно организовать работу проектировщиков. С его помощью можно избежать неопределенности и продуктивно распределить рабочее время.

# Заключение

Управление финансами всегда являлось одной из доминирующих составляющих процесса управления государством. Центральной задачей управления финансами является составление проекта бюджета. Автоматизация этой задачи позволит увеличить эффективность управления бюджетными средствами и повысить прозрачность процессов управления.

Автоматизированная система бюджетного процесса позволяет упростить и ускорить процесс составления проекта бюджета как отдельно взятого поселения, так и всей территории в целом. Архитектура автоматизированной системы бюджетного процесса, предложенная в дипломном проекте, предполагает размещение части программного комплекса непосредственно на компьютерах администраторов и распорядителей бюджетных средств, а также администраторов источников финансирования дефицита бюджета, что позволяет повысить оперативность составления проекта бюджета. Другой отличительной особенностью разработанной системы является ее модульность, т.е. в архитектуре системы изначально заложена возможность расширения ее функциональных возможностей за счет подключаемых модулей.

Результатом дипломного проекта является разработанная автоматизированная информационная система, полностью охватывающая первую часть бюджетного процесса – составление и утверждение проекта бюджета – которая внедрена и успешно используется в Финансовом управлении **администрации Новоегорлыкского сельского поселения**.

В качестве перспективы развития этой системы можно предложить дальнейшее расширение ее функциональных возможностей и постепенный охват оставшихся стадий бюджетного процесса.

Список используемых источников

1. Анализ требований и создание архитектуры решений на основе Microsoft.NET. Учебный курс MSCD/Пер. с англ. – М.: Издательско-торговый дом «Русская редакция», 2004. – 416 с.: ил.
2. Бизнес-правила в среде разработки и моделирования. Проверено 19.01.2008.
3. Богданов, В. Управление проектами в Microsoft Project 2002 / В. Богданов. − СПб.: Питер, 2003. – 604 с.
4. Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++./ Г. Буч. – М.: «Бином», СПб: «Невский диалект», 2001. – 560 с.
5. Бюджетный кодекс Российской Федерации. Постатейный научно-практический комментарий. Второе, дополнительное издание. – М: Агентство «Библиотечка «Российской газеты», 2006 – 288 с.
6. Бюджетные финансовые технологии. Проверено 25.04.2008.
7. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. / А.М. Вендров. – Проверено 05.02.2008.
8. Вигерс, К. Разработка требований к программному обеспечению. / К. Вигерс. – М.: Изд.-торг. Дом «Русская Редакция», 2004. – 576 с.
9. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб: «Питер», 2001. – 368 с.
10. Грехэм, И. Объектно-ориентированные методы. Принципы и практика / И. Грехэм – М.: «Вильямс», 2004. – 1024 с.
11. Коберн, А. Быстрая разработка программного обеспечения.: Пер. с англ. / А. Коберн. – М.: ЛОРИ, 2002. – 462 с.
12. Коберн, А. Современные методы описания функциональных требований к системам.: Пер. с англ. / А. Коберн. – М.: ЛОРИ, 2002. – 364 с.
13. Конноли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика./ Т. Конноли, К., Бегг, А. Страчан. – М.: «Вильямс», 2001. – 632 с.
14. Концептуальное проектирование реляционный баз данных с использованием языка UML. . Проверено 05.02.2008.
15. Ларман, К. Применение UML и шаблонов проектирования. 2-е издание / К. Ларман – М.: «Вильямс», – 2002. – 496 с.
16. Леффингуэлл, Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. / Д. Леффингуэлл, Д. Уидриг. – М.: «Вильямс», 2002. – 462 с.
17. Маклин, С. Microsoft.NET Remoting.: Пер. с англ. / С. Маклин, Дж. Нафтел, К. Ульямс. – М.: Издательско-торговый дом «Русская редакция», 2003. – 384 с.
18. Мацяшек Л.А. Анализ требований к проектированию систем. Разработка информационных систем с использованием UML / Л.А. Мацяшек. М.: Изд. Дом «Вильямс», 2002. – 432 с.
19. Мюллер Р.Дж. Базы данных и UML / Р. Дж. Мюллер. М: «ЛОРИ», 2002. – 420 с.
20. Нейбург, Э. Дж. Проектирование баз данных с помощью UML. / Э.Дж. Нейбург, Р.А. Максимчук. – М.: «Вильямс», 2002. – 420 с.
21. Олифер, В.Г. Основы сетей передачи данных./ В.Г. Олифер, Н.А. Олифер. Проверено 19.01.2008.
22. Описание предметной области с использованием UML при разработке программных систем. . Проверено 19.01.2008.
23. Проектирование и реализация баз данных Microsoft SQL Server 2000. Учебный курс MCAD, MCSE, MCDBA/Пер. с англ. – 2-е изд., испр. – М.: Издательско-торговый дом «Русская редакция», 2003. – 512 стр.: ил.
24. Полякова, Л.Н. Основы SQL БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий. / Л.Н. Полякова. – ИНТУИТ.ру, – 2007. – 462 с.
25. Рамбо, Дж. UML: специальный справочник. / Дж. Рамбо, А. Якобсон, Г. Буч. – СПб: «Питер», 2001. – 632 с.
26. Разработка Windows-приложений на Microsoft Visual Basic. Net и Microsoft Visual C#.Net. Учебный курс MCAD, MCSD/Пер. с англ. – М.: Издательско-торговый дом «Русская редакция», 2003. – 512 стр.: ил.
27. Розенберг, Д. Применение объектного моделирования с использованием UML и анализ прецедентов. / Д. Розенберг, К. Скотт. – М.: «ДМК Пресс», 2002. – 436 с.
28. Скрипкин, К.Г. Экономическая эффективность информационных систем. К.Г. Скрипкин. – М.: ДМК Пресс, 2002. – 420 с.
29. Смирнова, Г.Н. Проектирование экономических информационных систем: Учебник Г.Н. Смирнова, А.А. Сорокин Под ред. Ю.Ф. Тельнова. − М.: Финансы и статистика, 2003. – 512 с.
30. Тамре, Л. Введение в тестирование программного обеспечения: Пер. с англ. / Л. Тамре. – М.: Издательский дом «Вильямс», 2003. – 314 с.
31. Уилсон С. Принципы проектирования и разработки программного обеспечения. Учебный курс MCSD/Пер. с англ. – 2-е изд., испр. С. Уилсон, Б. Мейплс, Т. Лэндгрейв. – М.: Издательско-торговый дом «Русская редакция», 2002. – 736 стр.: ил.