

Министерство образования Белгородской области
Областное государственное автономное профессиональное
образовательное учреждение
«Белгородский индустриальный колледж»

Рассмотрено
предметно-цикловой комиссией
Протокол заседания № _____
От « ____ » _____ 2022 г.
Председатель цикловой комиссии
_____ / Третьяк И.Ю.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

**МДК. 05.01. Проектирование и дизайн информационных
систем**
для специальности
09.02.07 «Информационные системы и программирование»

Разработчик:
Шершнева М.А. преподаватель
специальных дисциплин ОГАОУ БИК

Белгород 2022

Тематика лабораторных работ по МДК. 05.01. Проектирование и дизайн информационных систем

Номер лабораторной работы	Тема лабораторной работ	Кол-во часов
1.	«Анализ предметной области различными методами: контент-анализ, вебометрический анализ, анализ ситуаций, моделирование и др.»	2
2.	«Организация проектирования информационных систем»	2
3.	«Изучение устройств автоматизированного сбора информации»	2
4.	«Оценка экономической эффективности информационной системы»	2
5.	«Разработка модели архитектуры информационной системы»	2
6.	«Обоснование выбора средств проектирования информационной системы»	2
7.	«Описание бизнес-процессов заданной предметной области»	2
8.	«Управление проектом информационных систем»	2
9.	Анализ рисков информационной безопасности	2
10.	«Построение модели управления качеством процесса изучения модуля «Проектирование и разработка информационных систем»»	2
11.	«Построение модели управления качеством процесса изучения модуля «Проектирование и разработка информационных систем»»	2
12.	«Реинжиниринг методом интеграции»	2
13.	«Разработка требований безопасности информационной системы»	2
14.	«Внутренние проверки качества информационной системы»	2
15.	«Реинжиниринг бизнес-процессов методом горизонтального и/или вертикального сжатия»	2
16.	«Построение модели бизнес-процессов, подлежащих автоматизации, с использованием диаграммы деятельности (activity diagram), моделирование поведения объектов в проблемной области с использованием диаграммы состояний (statechart diagram) с помощью CASE-средства Rational Rose Enterprise Edition»	2
17.	«Разработка технической документации»	2
18.	«Проектирование спецификации информационной системы индивидуальному заданию»	2
19.	«Разработка общего функционального описания программного средства по индивидуальному заданию»	2
20.	«Разработка руководства по установке программного средства по индивидуальному заданию»	2
21.	«Разработка руководства пользователя программного средства по индивидуальному заданию»	2
22.	«Изучение средств автоматизированного документирования»	2
23.	Разработка функциональной модели информационной системы	2
ИТОГО:		46

Лабораторная работа № 1

Анализ предметной области различными методами: контент-анализ, вебометрический анализ, анализ ситуаций, моделирование и др

Цели и задачи лабораторной работы

Целями выполнения лабораторной работы являются:

1. Закрепление имеющихся знаний о методах анализа и спецификации требований к информационным системам.
2. Приобретение навыков анализа и формализации требований, предъявляемых к ИС.
3. Приобретение навыков разработки технического задания на создание новой информационной системы.

В процессе выполнения лабораторной работы решаются следующие задачи:

1. Выполняется анализ постановки задачи на создание ИС.
2. Выявляются и формулируются концептуальные, функциональные и технические требования к информационной системе.
3. Разрабатывается документ «Техническое задание на создание ИС», описывающий требования к ИС и содержащий другие, необходимые для разработки, сведения.

Краткие теоретические сведения

Требования к программному обеспечению – это совокупность утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации.

Требования могут выражаться в виде текстовых утверждений и графических моделей. Требования функционального характера определяют требуемое поведение программной системы.

Выделяют следующие виды требований функционального характера:

1. Бизнес-требования – определяют назначение ПО, описываются в документе о видении и границах программного проекта.
2. Пользовательские требования – определяют набор пользовательских задач, которые должен решать программный продукт, а также способы их решения. Пользовательские требования могут выражаться в виде фраз утверждений, сценариев использования, сценариев взаимодействия, пользовательских историй.
3. Функциональные требования – охватывают предполагаемое поведение системы, определяя действия, которые система способна выполнять. Процесс разработки требований включает в себя выполнение следующих этапов:

- 1) выявление требований (сбор, понимание, рассмотрение и выяснение потребностей заинтересованных лиц);
- 2) анализ (проверка целостности и законченности);
- 3) спецификация (документирование требований);
- 4) проверка правильности.

Контрольные вопросы для допуска к работе

1. Требования к информационной системе.
2. Методы анализа и спецификации требований.

3. Анализ предметной области.
4. Разработка технического задания на создание информационной системы.
5. Концептуальные требования.
6. Функциональные требования.
7. Технические требования.
8. Технологии и методологии управления требованиями.

Порядок выполнения работы

Вариант индивидуального задания определяет информационную систему, для создания которой необходимо составить документ «Техническое задание» с подробным формализованным описанием требований к системе.

В процессе выполнения лабораторной работы необходимо:

1. Изучить требования к структуре и содержанию документа «Техническое задание на создание ИС». Составить план документа.
2. Сформулировать цели и задачи создания ИС. Охарактеризовать вид ИС, её назначение, используемые в работе системы данные. Сформулировать концептуальные требования к ИС.
3. Дать характеристику типового объекта автоматизации (организации, предприятия) для которого создаётся и на котором будет внедрена ИС. Описать автоматизируемые бизнес-процессы.
4. Сформулировать требования к системе в целом. Описать структуру ИС. Перечислить функциональные подсистемы.
5. Сформулировать функциональные требования. Описать требования к функциям и задачам, выполняемым системой. Описать назначение и состав функций каждой из подсистем.
6. Описать предметную область. Разработать концептуальную модель данных предметной области. Сформулировать требования к информационному обеспечению системы.
7. Сформулировать требования к программному обеспечению системы. Описать требования к пользовательскому интерфейсу. Сформулировать технические требования к реализации и режимам работы ИС.
8. Используя полученные результаты, подготовить документ «Техническое задание на создание ИС», включающий в себя полное описание концептуальных, функциональных и технических требований к создаваемой системе. Варианты индивидуальных заданий

Варианты индивидуальных заданий

1. ИС «Телефонный справочник» (поисковая система).
2. ИС «Библиотека» (информационно-справочная система, поисковая система).
3. ИС «Издательство» (СЭДО, САБП).
4. ИС «Поликлиника» (СЭДО, информационно-справочная система).
5. ИС «Школа» (обучающая система, информационносправочная система).
6. ИС «Ателье» (САБП).
7. ИС «Склад» (САБП).

8. ИС «Торговля» (САБП, СЭДО).
9. ИС «Автосалон» (САБП, СЭДО).
10. ИС «Продажа подержанных автомобилей» (информационно-справочная система, поисковая система).
11. ИС «Автосервис» (САБП).
12. ИС «Пассажирское автопредприятие» (САБП, СЭДО).
13. ИС «Диспетчерская служба такси» (ГИС, СЭДО).
14. ИС «Агентство по продаже авиабилетов» (информационно-справочная система, поисковая система).
15. ИС «Туристическое агентство» (информационносправочная система, поисковая система).
16. ИС «Гостиница» (информационно-справочная система, СЭДО).

Лабораторная работа № 2

Организация проектирования информационных систем Цели и задачи лабораторной работы

Целями выполнения лабораторной работы являются:

1. Закрепление имеющихся знаний о моделях жизненного цикла ИС и современных методологиях разработки программного обеспечения.
2. Приобретение навыков анализа требований, условий и ограничений проекта создания ИС и оценки трудоёмкости его реализации.
3. Приобретение навыков составления планов разработки ИС на основе разных моделей жизненного цикла.

В процессе выполнения лабораторной работы решаются следующие задачи:

1. Выполняется анализ постановки задачи.

Готовятся исходные данные для планирования. Формулируются ограничения и условия разработки.

2. Разрабатываются прототипы документов: «Техническое задание», «Технический проект», «План тестирования», «План ввода в эксплуатацию».

3. Составляется календарный план разработки ИС.

Краткие теоретические сведения

Жизненный цикл (ЖЦ) информационной системы – непрерывный процесс, который начинается с момента принятия решения о необходимости создания системы и заканчивается в момент её полного изъятия из эксплуатации.

Модель жизненного цикла ИС – структура, описывающая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного обеспечения в течение всей жизни ИС, от определения требований до завершения её использования.

К настоящему времени наибольшее распространение получили следующие основные модели ЖЦ:

- 1) каскадная (водопадная) модель и её варианты;
- 2) инкрементная модель;
- 3) спиральная модель.

Каскадная или водопадная модель ЖЦ является классической моделью однократного прохода, которая описывает линейную последовательность этапов создания ИС.

Спиральная модель ЖЦ относится к эволюционным моделям. Каждый виток раскручивающейся спирали соответствует разработке одной (начальной, промежуточной или окончательной) версии ИС и представляет собой полный цикл разработки, начиная с анализа и заканчивая внедрением.

Прототип – версия ИС, предназначенная для демонстрации заказчику некоторых ключевых свойств будущего продукта. Создание прототипа позволяет вовлечь заказчика в разработку информационной системы в самом начале работы.

Порядок выполнения работы

Вариант индивидуального задания определяет информационную систему, для создания которой необходимо составить план разработки на основе каскадной и спиральной моделей жизненного цикла.

В процессе выполнения лабораторной работы необходимо:

1. Подготовить исходные данные. Исходными данными для планирования являются:

1.1. Общее описание некоторой ИС (назначение, область применения, решаемые задачи, технологические особенности реализации и внедрения).

1.2. Ограничения и условия разработки (требования заказчика, возможности команды разработчиков, сроки разработки, бюджет проекта и т.д.).

2. Составить план разработки ИС с применением каскадного подхода:

2.1. Составить эскизный план разработки ИС на основе каскадной модели ЖЦ.

2.2. Для этапа «Анализ требований» составить документ «Техническое задание» с подробным описанием функциональных требований к ИС.

2.3. Для этапа «Проектирование» составить документ «Технический проект» с описанием проектных решений (архитектура системы, логическая структура базы данных, решения по реализации пользовательского интерфейса и т.д.).

2.4. Для этапа «Тестирование» составить документ «План тестирования» с описанием методики тестирования и контрольных тестов.

2.5. Для этапа «Внедрение» составить документ «План ввода ИС в эксплуатацию».

2.6. Уточнить параметры календарного плана разработки ИС, учитывая ограничения и условия разработки.

2.7. Объединить календарный план разработки и составленные документы в единый отчёт «Разработка ИС на основе каскадной модели ЖЦ».

14

3. Составить план разработки ИС с применением итеративного подхода:

3.1. Разделить весь процесс создания и внедрения ИС на несколько итераций.

3.2. На основе имеющихся документов (см. пункты 2.2 – 2.5) для каждой итерации составить отдельный комплект документов.

3.3. Составить календарный план итеративной разработки ИС.

3.4. Объединить план итеративной разработки и составленные документы в единый отчёт «Разработка ИС на основе спиральной модели ЖЦ».

Варианты индивидуальных заданий

1. ИС «Телефонный справочник» (поисковая система).

2. ИС «Библиотека» (информационно-справочная система, поисковая система).

3. ИС «Издательство» (СЭДО, САБП).

4. ИС «Поликлиника» (СЭДО, информационно-справочная система).

5. ИС «Школа» (обучающая система, информационносправочная система).

6. ИС «Ателье» (САБП).
7. ИС «Склад» (САБП).
8. ИС «Торговля» (САБП, СЭДО).
9. ИС «Автосалон» (САБП, СЭДО).
10. ИС «Продажа подержанных автомобилей» (информационно-справочная система, поисковая система).
11. ИС «Автосервис» (САБП).
12. ИС «Пассажирское автопредприятие» (САБП, СЭДО).
13. ИС «Диспетчерская служба такси» (ГИС, СЭДО).
14. ИС «Агентство по продаже авиабилетов» (информационно-справочная система, поисковая система).
15. ИС «Туристическое агентство» (информационносправочная система, поисковая система).
16. ИС «Гостиница» (информационно-справочная система, СЭДО).

Контрольные вопросы:

1. Современные методологии разработки информационных систем.
2. Жизненный цикл информационных систем.
3. Этапы жизненного цикла: анализ, проектирование, программирование, тестирование, эксплуатация.
4. Модели жизненного цикла.
5. Каскадная модель жизненного цикла.
6. Преимущества и недостатки каскадной модели жизненного цикла.
7. Итеративная модель.
8. Спиральная модель.
9. Методология Microsoft Solutions Framework.
10. Методология Rational Unified Process.

Лабораторная работа 3

Изучение устройств автоматизированного сбора информации

Цель: Научиться проводить оценку экономической эффективности информационной системы.

Задание: в среде MS Excel провести оценку экономической эффективности внедрения ИС методом дисконтирования.

Исходные данные:

В государственной картинной галерее внедряется информационная система учета художественных ценностей.

Объект автоматизации - отдел учета государственной картинной галереи.

После внедрения разрабатываемой информационной системы сотрудники музея смогут в более короткие сроки осуществлять отбор предметов для выставки; рукописное оформление учетных документов и карточек на предметы заменится их электронными вариантами; сотрудники музея смогут более рационально использовать свое рабочее время.

Оценка отрицательных денежных потоков

Инвестиции в проект будут складываться из следующих составляющих:

1. затраты на приобретение лицензионного ПО;
2. затраты на разработку проекта.

Работа информационной системы возможна в двух вариантах: файловом и клиент-серверном.

Для файлового варианта потребуется следующее ПО:

- платформа 1С:Предприятие 8.1;

Поскольку технологическая платформа распространяется не отдельно, а только в составе прикладного решения, то пользователю будет необходимо приобрести любое прикладное решение профессиональной версии. Наименьшую стоимость имеет ПП «1С: Бухгалтерия 8» - 9 900 (руб.).

- клиентские лицензии;

Основные поставки продуктов системы «1С:Предприятие 8», содержащие программную часть (технологическую платформу) и прикладные решения для автоматизации различных задач управления и учета (конфигурации), выпускаются в виде однопользовательских продуктов.

В комплект основной поставки входит дистрибутив на CD-ROM, комплект документации, однопользовательский ключ защиты от несанкционированного доступа (ключ защиты) для порта USB, Лицензионное соглашение, разрешающее использование программного продукта на одном компьютере, и другие материалы. Для использования продуктов системы «1С:Предприятие» на двух и более компьютерах в пределах одной локальной вычислительной сети требуется приобретение дополнительных лицензий. Фирмой «1С» выпускаются дополнительные лицензии на 1, 5, 10, 20 и 50 рабочих мест, необходимое пользователю количество рабочих мест складывается из этих лицензий.

Стоимость лицензий следующая:

- на 1 рабочее место - 4 750 (руб.);

- на 5 рабочих мест - 15 950 (руб.);
- на 10 рабочих мест - 31 200 (руб.);
- на 50 рабочих мест - 139 800 (руб.);
- на 100 рабочих мест - 272 000 (руб.).

Исходя из того, что клиентские лицензии можно будет докупить в случае необходимости, то будет достаточно 5 лицензий.

Таким образом, для файлового варианта работы системы затраты могут составить ? (руб.).

С другой стороны, в ассортименте продаж фирмы 1С есть такой продукт как «1С:Бухгалтерия 8. Комплект на 5 пользователей» стоимостью 19 950 руб.

Поскольку комплекты ПО равнозначны, то в качестве затрат на ПО при файловом варианте работы системы примем менее дорогой вариант стоимостью ? руб.

Для клиент-серверного варианта потребуется следующее ПО:

- платформа 1С:Предприятие 8.1 и клиентские лицензии- ? (руб.);
- лицензия на сервер 1С:Предприятие - 37 500 (руб.);
- СУБД PostgreSQL - 0 (руб.)

Таким образом, для клиент-серверного варианта затраты составят ? (руб.).

В затраты на создание ИС входят затраты на выполнение следующих задач:

Стоимость разработки проекта будет складываться из общей стоимости работы ресурсов.

Для ресурсов назначим следующую стоимость работы:

- дипломник - 0 руб./день;
- программист - 600 руб./день;
- представитель заказчика - 0 руб./день.

Таким образом, стоимость разработки составит ? (руб.).

К стоимости разработки проекта следует прибавить величину социальных отчислений, которую заплатит заказчик проекта. Ставки отчислений в социальные фонды (Пенсионный и Фонд обязательного медицинского страхования) с 01.01.2011 составляют 34 %.

Таким образом, суммарные затраты на файловый вариант работы системы составит ? руб.

Затраты на клиент-серверный вариант работы системы составит ? руб.

Поскольку файловый вариант работы системы предполагает возможный переход к клиент-серверному (без какой-либо потери данных), то в качестве отрицательных денежных потоков примем затраты на файловый вариант работы, т.е. ? руб.

Оценка положительных денежных потоков

Положительные денежные потоки предполагается получить за счет увеличения количества дней работы музея. В настоящее время музей работает 4 дня в неделю (со вторника по пятницу) (т.е. в среднем 16 дней в месяц). По понедельникам в музее проводится работа по подготовке новых выставок.

Поскольку внедряемая информационная система позволит сократить время на отбор необходимых предметов для организации выставки, то каждый месяц музей сможет работать на два дня больше. Рассчитаем возможный доход за день.

За предыдущий год музей посетило 22,5 тыс. человек, из них 12,2 тыс. чел. (54%) - индивидуально (в т.ч. по льготным билетам - 2,7 тыс. чел. (22%)) и 10,3 тыс. чел. (46%) - в составе экскурсий (в т.ч. 5,2 тыс. чел. - по льготным билетам (50,5%)). Стоимость билета составляет 50 руб. - для взрослого и 30 руб. - для льготника. Стоимость экскурсий - 200 руб. для группы взрослых людей и 100 руб. - для группы детей (в стоимость экскурсии не входит стоимость билетов).

Среднее количество посетителей за один день работы музея составляет: **? руб.**

Из 117 человек 63 (54%) посетят музей индивидуально, 54 (46%) - в составе группы (экскурсия).

Из 63 человек, которые посетят музей индивидуально, 49 человек (78%) купят взрослый билет, 14 (22%) - льготный. Т.о., доход от индивидуального посещения музея в день составит: **? руб.**

Из 54 человек, которые посетят музей в составе группы (экскурсия), 27 человек (49,5%) купят взрослый билет, и 27 (50,5%) - льготный. Учитывая, что обычно группа состоит из 10-15 человек, то к доходу от продажи билетов прибавим стоимость двух экскурсий для взрослых людей и двух - для детей. Таким образом, получим: **? руб.**

Суммируя доход в день от индивидуального посещения музея и проведения экскурсий, получаем: **? руб.**

За год получим: **? руб.**

Таким образом, положительные потоки от внедрения проекта составят **? руб.** в год.

Поскольку доходы от выставочной деятельности музеев не облагаются налогом на прибыль, то этот налог не учитываем.

Расчет показателей

Использование системы рассчитано на 5 лет.

Ставку рефинансирования примем равной 8,25.

Темп инфляции примем равным 8,8% в год.

Рассчитайте чистый приведенный доход NPV.

NPV = **? руб.**

Принесет ли прибыль внедрение данного проекта?

Определите срок окупаемости инвестиций PP.

PP = **? лет.**

За сколько лет и месяцев окупится проект?

Определим индекс доходности инвестиций PI.

PI = **? руб.**

Каждый вложенный в проект рубль окупит себя (?) и принесет еще **? руб.**

Определите внутреннюю норму доходности проекта IRR.

IRR = ? %.

Запас прочности проекта составляет ? %.

Для того, чтобы проект стал невыгодным учетная ставка должна увеличиться на ? % относительно ее текущего значения.

Теоретические сведения

Под понятием «оценка экономической эффективности ИС» понимается процесс, включающий в себя понимание, определение и измерение того, насколько полезным в экономическом плане является или явилось внедрение ИС для предприятия. При этом экономическая полезность рассматривается обычно как денежный эквивалент того, насколько изменились доходы/расходы предприятия в результате инвестирования в ИС.

Под методом оценки эффективности ИС подразумевается способ или набор средств проведения полной оценки ИС. Они могут состоять как из формальных, так и из неформальных процедур, при этом под неформальными понимаются не основанные на цифровых данных, быстрые, преимущественно субъективные процедуры оценки, а под формальными - более объективные, рациональные, базирующиеся на недвусмысленных данных механизмы оценки.

Как известно, внедрение современных информационных технологий - дело дорогостоящее. Функционирование компаний в рыночной среде требует как минимум анализа экономических последствий, а еще лучше - оценки экономической эффективности того или иного шага преобразования системы управления компанией.

Оценка экономической эффективности ИС - сложная и трудоемкая работа, требующая не только технических, но и экономических навыков. Только сочетание этих двух составляющих может привести к достоверному результату проводимого анализа.

Продвижение на рынке ИС в условиях современной конкуренции невозможно без предоставления результатов оценки ожидаемой эффективности системы. Кроме того, существующая статистическая оценка успешности внедрения систем управления предприятием характеризуется неудачей внедрения от 40 до 70 % случаев.

Специалисты в области разработки, внедрения и сопровождения ИС должны обладать навыками проведения предварительной экспертизы проекта. Они должны уметь вести постоянный мониторинг системы на соответствие внедряемых технологий стратегии развития предприятия. Процесс соизмерения затрат и достигаемого за их счет эффекта должен быть именно «процессом», то есть итерационной процедурой, проводимой на протяжении всего этапа разработки и внедрения проекта, результат которой способен повлиять на дальнейшее продолжение проекта.

Существуют следующие этапы оценки экономической эффективности информационной системы:

- традиционная оценка эффективности как соотношение затрат и результатов;
- расчет совокупной стоимости владения информационной системой;

- оценка внедрения ИС как инвестиционного проекта;
- разработка сбалансированной системы показателей для оценки экономического эффекта;
- оценка эффективности проектов независимо от технических, технологических, финансовых, отраслевых или региональных особенностей осуществляется на основе единых принципов. К ним относятся:
 - рассмотрение проекта на протяжении всего жизненного цикла;
 - моделирование денежных потоков;
 - сопоставимость условий сравнения различных проектов;
 - положительность и максимум эффекта;
 - учет фактора времени;
 - учет только предстоящих в ходе осуществления проекта затрат и поступлений;
 - сравнение «с проектом» и «без проекта»;
 - учет всех наиболее существенных последствий проекта;
 - учет наличия разных участников проекта;
 - многоэтапность оценки;
 - учет влияния на эффективность инвестиционного проекта;
 - учет влияния инфляции;
 - учет влияния неопределенностей и рисков.

Показатели коммерческой эффективности проекта в целом отражают финансовые последствия внедрения информационной системы. В качестве основных показателей для расчета коммерческой эффективности проекта рекомендуется использовать следующие:

- чистый доход;
- чистый дисконтированный доход;
- внутренняя норма доходности;
- индексы доходности затрат и инвестиций;
- срок окупаемости.

Таким образом, исходя из всего выше сказанного, можно сделать вывод, что процесс оценки экономической эффективности информационных систем сложен и неоднозначен. Подходить следует индивидуально в каждом конкретном случае, но опираясь на предложенные схемы и методики, что позволит исключить «человеческий фактор» и снизить погрешности ввиду отсутствия каких либо данных.

Выбор и обоснование методики оценки информационной системы Метод оценки чистого приведенного эффекта

Важнейшим показателем эффективности инвестиционного проекта является чистая текущая стоимость (*Net Present Value, NPV*) - накопленный дисконтированный эффект за расчетный период.

В общем случае методика расчета *NPV* заключается в суммировании современных (пересчитанных на текущий момент) величин чистых эффективных денежных потоков по всем интервалам планирования на всем протяжении периода исследования. При этом, как правило, учитывается и ликвидационная или остаточная стоимость проекта, формирующая

дополнительный денежный поток за пределами горизонта исследования. Для пересчета всех указанных величин используются коэффициенты приведения, основанные на выбранной ставке сравнения (дисконтирования).

Классическая формула для расчета NPV выглядит следующим образом:

где:

r - ставка дисконтирования (в десятичном выражении);

n - номер года;

О степени эффективности вложения средств в данный проект говорит полученная величина NPV .

Очевидно, что если:

- $NPV > 0$, то проект следует принять;
- $NPV < 0$, то проект следует отвергнуть;
- $NPV = 0$, то проект ни прибыльный, ни убыточный.

Определение срока окупаемости инвестиций

Срок окупаемости относится к числу наиболее часто используемых показателей эффективности инвестиций. Достаточно сказать, что именно этот показатель, наряду с внутренней ставкой доходности, выбран в качестве основного в методике оценки инвестиционных проектов, участвующих в конкурсном распределении централизованных инвестиционных ресурсов.

Цель данного метода состоит в определении продолжительности периода, в течение которого проект будет работать, что называется, «на себя». При этом весь объем генерируемых проектом денежных средств, главными составляющими которого являются чистая прибыль и сумма амортизационных отчислений (то есть чистый эффективный денежный поток), засчитывается как возврат на первоначально инвестированный капитал.

Расчет простого срока окупаемости, в силу своей специфической наглядности, часто используется как метод оценки риска, связанного с инвестированием.

Существенным недостатком рассматриваемого показателя является то, что он ни в коей мере не учитывает результаты деятельности за пределами установленного периода исследования проекта и, следовательно, не может применяться при сопоставлении вариантов капиталовложений, различающихся по срокам жизни.

Простой срок окупаемости рассчитывается по формуле:

где:

PP - срок окупаемости, выраженный в интервалах планирования;

PV - полные инвестиционные затраты проекта;

NCF - чистый эффективный денежный поток за один интервал планирования.

Определение индекса доходности инвестиций

Рассматриваемый показатель тесно связан с показателем чистой современной ценности инвестиций, но, в отличие от последнего, позволяет

определить не абсолютную, а относительную характеристику эффективности инвестиций.

Индекс доходности инвестиций PI рассчитывается по следующей формуле:

где PV - полные инвестиционные затраты проекта.

PI характеризует уровень доходов на единицу затрат, т.е. эффективность вложений - чем больше значение этого показателя, тем выше отдача с каждой единицы капитальных вложений, инвестированных в данный проект.

Метод определения внутренней нормы доходности проекта

Под внутренней нормой доходности проекта IRR (*Internal Rate Of Return*) понимают значение коэффициента дисконтирования (r), при котором NPV проекта равен 0.

где IC - величина исходных инвестиций.

Смысл расчета этого коэффициента при анализе эффективности результата инвестиций следующий: IRR показывает максимально допустимый относительный уровень расходов, которые могут быть вложены в данный проект. Например, если проект полностью финансируется за счет ссуды коммерческого банка, то значение IRR показывает верхнюю границу допустимого уровня банковской процентной ставки, превышение которой делает проект убыточным.

Движущие мотивы финансирования проекта существенно зависят от того, реализуется ли инвестиция за свои или привлеченные средства. Однако для большинства инвестиционных проектов в случаях финансирования и за счет собственных средств, и за счет привлеченных финансовых ресурсов в основе лежит показатель цены капитала.

Практическое применение данного метода сводится к последовательной итерации, с помощью которой находится дисконтирующий множитель, обеспечивающий равенство $NPV = 0$.

Ориентируясь на существующие в момент анализа процентные ставки на ссудный капитал, выбираются два значения коэффициента дисконтирования $r_1 < r_2$, таким образом, чтобы в интервале $(r_1; r_2)$ функция $NPV = f(r)$ меняла свое значение с «+» на «-» или наоборот. Далее используем формулу:

где:

r_1 - значение процентной ставки в дисконтном множителе, при котором $f(i_1) < 0$ ($f(i_1) > 0$);

r_2 - значение процентной ставки в дисконтном множителе, при котором $f(i_2) > 0$ ($f(i_2) < 0$);

Лабораторная работа № 5

Разработка модели архитектуры информационной системы

Цели и задачи лабораторной работы

Целями выполнения лабораторной работы являются:

1. Закрепление имеющихся знаний о технологиях и методологиях моделирования информационных систем.
2. Приобретение навыков объектно-ориентированного анализа, моделирования и проектирования ИС.
3. Приобретение навыков разработки моделей ИС в виде диаграмм, построенных с применением унифицированного языка моделирования UML.

В процессе выполнения лабораторной работы решаются следующие задачи:

1. Выполняется разработка концептуальных моделей ИС для описания автоматизируемых бизнес-процессов с помощью диаграмм деятельности и диаграмм последовательности.
2. Выполняется разработка логических моделей ИС для описания требований к системе с помощью диаграмм прецедентов и диаграмм классов.
3. Выполняется разработка физических моделей ИС для описания конкретного способа реализации системы с помощью диаграмм базы данных, диаграмм компонентов и диаграмм развертывания.

Краткие теоретические сведения

Унифицированный язык моделирования UML – это графический язык моделирования общего назначения, предназначенный для спецификации, визуализации, проектирования и документирования всех компонентов, создаваемых при разработке программных систем. Язык UML является объектно-ориентированным языком. Его использование основывается на понимании общих принципов объектно-ориентированного анализа и проектирования:

1. Принцип абстрагирования предписывает включать в модель только те аспекты проектируемой системы, которые имеют непосредственное отношение к выполнению системой своих функций.
2. Принцип многомодельности означает, что никакое единственное представление системы не является достаточным для адекватного выражения всех ее особенностей.
3. Принцип иерархического построения моделей сложных систем предписывает рассматривать процесс построения моделей на разных уровнях абстрагирования или детализации в рамках фиксированных представлений.

Диаграмма UML – это графическое представление набора элементов, изображаемое в виде связанного графа с вершинами (сущностями) и ребрами (отношениями), используемое для визуализации системы с разных точек зрения.

Диаграммы UML используются для описания различных аспектов функционирования и структуры ИС на разных стадиях создания системы и, соответственно, на разных этапах моделирования: концептуального, логического и физического. Контрольные вопросы для допуска к работе

1. Моделирование информационных систем.
2. Виды моделей.
3. Объектно-ориентированный анализ и проектирование.
4. Технологии, языки и средства моделирования.
5. Язык унифицированного моделирования UML.
6. Диаграммы языка UML: структурные диаграммы, диаграммы поведения, диаграммы взаимодействия.
7. Инструментальные средства моделирования ИС.
8. Применение UML при проектировании ИС.

Порядок выполнения работы

Вариант индивидуального задания определяет ИС, для создания которой необходимо разработать совокупность моделей системы в виде комплекта диаграмм UML.

Построенные модели ИС должны описывать различные аспекты проектирования и разработки системы на разных стадиях её жизненного цикла.

В процессе выполнения лабораторной работы необходимо:

1. Разработать модель прецедентов, описывающую бизнеспроцессы организации с точки зрения внешнего пользователя (клиента) и отражающую взгляд на деятельность организации извне. Результатом моделирования являются диаграммы деятельности и диаграммы прецедентов.

2. Разработать модель бизнес-объектов, описывающую выполнение бизнес-процессов организации ее внутренними исполнителями. Основными компонентами модели являются внешние и внутренние исполнители. Результатом моделирования являются диаграммы последовательности.

3. Разработать концептуальную модель данных, описывающую объекты предметной области и связи между ними. Результатом моделирования являются диаграммы классов и диаграммы объектов.

4. Разработать описание требований к системе. Результатом является исчерпывающий перечень функций, которые должны быть реализованы в системе, и подробное описание необходимой реализации этих функций.

5. Разработка моделей базы данных и приложений, представляющих собой детальное описание проекта базы данных и клиентских приложений информационной системы. Результатом моделирования являются диаграммы компонентов и диаграммы базы данных.

6. Разработать проект физической реализации информационной системы. Результатом проектирования являются диаграммы развёртывания и диаграммы компонентов.

Варианты индивидуальных заданий

В качестве списка вариантов индивидуальных заданий используется перечень информационных систем из лабораторной работы № 1.

Лабораторная работа № 6

Обоснование выбора средств проектирования информационной системы

Цели и задачи лабораторной работы

Целями выполнения лабораторной работы являются:

1. Закрепление имеющихся знаний о средствах разработки программного обеспечения информационных систем.
2. Приобретение навыков работы в современных интегрированных средах разработки программного обеспечения.
3. Приобретение навыков разработки клиентского программного обеспечения ИС с применением принципов методологии RAD.

В процессе выполнения лабораторной работы решаются следующие задачи:

1. Проектируется макет интерфейса и разрабатывается прототип клиентского приложения для заданной ИС.
2. Разрабатывается программный код клиентского приложения для реализации функциональных требований к ИС. Выполняется тестирование и отладка разработанного приложения.
3. Разрабатывается документ «Руководство пользователя» с описанием назначения и функциональных возможностей клиентского приложения создаваемой системы.

Краткие теоретические сведения

Быстрая разработка приложений RAD (Rapid Application Development) является одной из современных методологий разработки программного обеспечения. Как и другие методологии (MSF, RUP и др.) RAD описывает итеративный подход к организации процесса разработки ПО и соответствующую модель жизненного цикла.

Методологию RAD также часто связывают с технологией визуального программирования и применением современных интегрированных сред разработки программного обеспечения.

Методология RAD основывается на визуализации процесса создания программного кода приложений и поддерживается инструментальным ПО, которое предоставляет разработчикам средства визуального программирования.

Применение средств визуального программирования позволяет значительно ускорить процесс разработки приложений, а также уменьшить трудоёмкость работы по модификации уже готовой программы, внесению в неё необходимых дополнений или изменений. Средства быстрой разработки приложений, как правило, основываются на объектно-ориентированной компонентной архитектуре.

Процедура разработки интерфейса средствами RAD сводится к набору последовательных операций, включающих:

- 1) размещение компонентов интерфейса в нужном месте;
- 2) задание моментов времени их появления на экране;
- 3) настройку связанных с ними атрибутов и событий.

Интегрированная среда разработки (ИСР) является средством, с помощью которого выполняются проектирование, программирование, тестирование и отладка прикладных программ.

Примерами современных ИСР, поддерживающих методологию RAD и технологию визуального программирования, являются Microsoft Visual Studio, Embarcadero RAD Studio, IntelliJ IDEA, MonoDevelop и др.

Контрольные вопросы для допуска к работе

1. Средства разработки программного обеспечения ИС.
2. Программные платформы, технологии программирования и инструментальные средства разработки.
3. Интегрированные среды разработки.
4. Современные средства разработки ПО.
5. Методология быстрой разработки приложений RAD.
6. Технология визуального программирования.
7. Автоматическая генерация программного кода.
8. Применение RAD и визуального программирования для прототипирования клиентских приложений ИС.

Порядок выполнения работы

Вариант индивидуального задания определяет информационную систему, для которой необходимо разработать клиентское программное обеспечение.

В процессе выполнения лабораторной работы необходимо:

1. Выполнить анализ требований к информационной системе. Составить перечень функциональных требований к клиентскому приложению. Сформулировать общие требования к пользовательскому интерфейсу.
2. Разработать проект пользовательского интерфейса приложения. С помощью интегрированной среды разработки создать макеты экранных форм с размещёнными на них элементами интерфейса.
3. Разработать прототип клиентского приложения, пользуясь средствами визуального программирования интегрированной среды разработки.
4. Реализовать необходимый функционал приложения добавлением программного кода для обработки системных событий и действий пользователя.
5. Выполнить тестирование общей работоспособности и отдельных функциональных возможностей разработанного приложения. Исправить возможные ошибки.
6. Выполнить верификацию функциональных возможностей разработанного приложения, сравнивая их с имеющимся перечнем функциональных требований.
7. Разработать документ «Руководство пользователя» с описанием назначения и функциональных возможностей клиентского приложения создаваемой системы.

Варианты индивидуальных заданий

В качестве списка вариантов индивидуальных заданий используется перечень информационных систем из лабораторной работы № 1.

Лабораторная работа 7.

Описание бизнес-процессов заданной предметной области.

Цель работы. Изучить методику моделирования предметной области на примере выбранного учебного проекта.

Моделирование предметной области

Задание:

- Создать на примере выбранного учебного проекта следующие модели: структурную модель предметной области, объектную структуру, функциональную структуру, структуру управления, организационную структуру.
- Изучить на примере выбранного учебного проекта следующие подходы к описанию предметной области: функционально-ориентированные и объектно-ориентированные методологии, функциональную методику IDEF, функциональную методику потоков данных, объектно-ориентированную методику. Сравнить существующие методики. Изучить синтетическую методику.

Примеры предметных областей для учебного проекта:

1. Учебный центр
2. Супермаркет
3. Транспортная компания
4. Мебельный цех
5. Сервисный центр бытовой техники
6. Центр обслуживания копировальной техники
7. Книжный магазин
8. Софтверная компания
9. Авиакасса
10. Железнодорожная касса
11. Отделение пенсионного фонда
12. Служба занятости города
13. Секретариат коммерческой компании
14. Ресторан быстрого питания
15. Интернет-магазин
16. Агентство недвижимости
17. Департамент жилищно-коммунального хозяйства
18. Управляющая компания
19. Поликлиника
20. Автосалон
21. Автосервис
22. Салон мобильной связи
23. Туристическое агентство
24. Коммерческий банк
25. Производственная компания

Вы можете выбрать другую проблемную область. В этом случае необходимо согласовать ее с преподавателем.

Структурная модель предметной области

В основе проектирования ИС лежит моделирование предметной области. Для того чтобы получить адекватный предметной области проект ИС в виде системы правильно работающих программ, необходимо иметь целостное, системное представление модели, которое отражает все аспекты функционирования будущей информационной системы. При этом под моделью предметной области понимается некоторая система, имитирующая структуру или функционирование исследуемой предметной области и отвечающая основному требованию - быть адекватной этой области.

Предварительное моделирование предметной области позволяет сократить время и сроки проведения проектировочных работ и получить более эффективный и качественный проект. Без проведения моделирования предметной области велика вероятность допущения большого количества ошибок в решении стратегических вопросов, приводящих к экономическим потерям и высоким затратам на последующее перепроектирование системы. Вследствие этого все современные технологии проектирования ИС основываются на использовании методологии моделирования предметной области.

К моделям предметных областей предъявляются следующие требования:

- формализация, обеспечивающая однозначное описание структуры предметной области;
- понятность для заказчиков и разработчиков на основе применения графических средств отображения модели;
- реализуемость, подразумевающая наличие средств физической реализации модели предметной области в ИС;
- обеспечение оценки эффективности реализации модели предметной области на основе определенных методов и вычисляемых показателей.

Для реализации перечисленных требований, как правило, строится **система моделей**, которая отражает структурный и оценочный аспекты функционирования предметной области.

Структурный аспект предполагает построение:

- объектной структуры, отражающей состав взаимодействующих в процессах материальных и информационных объектов предметной области;
- функциональной структуры, отражающей взаимосвязь функций (действий) по преобразованию объектов в процессах;
- структуры управления, отражающей события и бизнес-правила, которые воздействуют на выполнение процессов;
- организационной структуры, отражающей взаимодействие организационных единиц предприятия и персонала в процессах;
- технической структуры, описывающей топологию расположения и способы коммуникации комплекса технических средств.

Для отображения структурного аспекта моделей предметных областей в основном используются графические методы, которые должны гарантировать представление информации о компонентах системы. Главное требование к

графическим методам документирования - простота. Графические методы должны обеспечивать возможность структурной декомпозиции спецификаций системы с максимальной степенью детализации и согласований описаний на смежных уровнях декомпозиции.

С моделированием непосредственно связана проблема **выбора языка** представления проектных решений, позволяющего как можно больше привлекать будущих пользователей системы к ее разработке. Язык моделирования - это нотация, в основном графическая, которая используется для описания проектов. Нотация представляет собой совокупность графических объектов, используемых в модели. Нотация является синтаксисом языка моделирования. Язык моделирования, с одной стороны, должен делать решения проектировщиков понятными пользователю, с другой стороны, предоставлять проектировщикам средства достаточно формализованного и однозначного определения проектных решений, подлежащих реализации в виде программных комплексов, образующих целостную систему программного обеспечения.

Графическое изображение нередко оказывается наиболее емкой формой представления информации. При этом проектировщики должны учитывать, что графические методы документирования не могут полностью обеспечить декомпозицию проектных решений от постановки задачи проектирования до реализации программ ЭВМ. Трудности возникают при переходе от этапа анализа системы к этапу проектирования и в особенности к программированию.

Главный критерий адекватности структурной модели предметной области заключается в функциональной полноте разрабатываемой ИС.

Оценочные аспекты моделирования предметной области связаны с разрабатываемыми показателями эффективности автоматизируемых процессов, к которым относятся:

- время решения задач;
- стоимостные затраты на обработку данных;
- надежность процессов;
- косвенные показатели эффективности, такие, как объемы производства, производительность труда, оборачиваемость капитала, рентабельность и т.д.

Для расчета показателей эффективности, как правило, используются статические методы функционально-стоимостного анализа (АВС) и динамические методы имитационного моделирования.

В основе различных методологий моделирования предметной области ИС лежат принципы последовательной детализации абстрактных категорий. Обычно модели строятся на трех уровнях: на внешнем уровне (**определении требований**), на концептуальном уровне (**спецификации требований**) и внутреннем уровне (**реализации требований**). Так, на внешнем уровне модель отвечает на вопрос, что должна делать система, то есть определяется состав основных компонентов системы: объектов, функций, событий, организационных единиц, технических средств. **На концептуальном уровне** модель отвечает на вопрос, как должна функционировать система? Иначе говоря, определяется характер взаимодействия компонентов системы

одного и разных типов. На внутреннем уровне модель отвечает на вопрос: с помощью каких программно-технических средств реализуются требования к системе? С позиции жизненного цикла ИС описанные уровни моделей соответственно строятся на этапах анализа требований, логического (технического) и физического (рабочего) проектирования. Рассмотрим особенности построения моделей предметной области на трех уровнях детализации.

Объектная структура

Объект - это сущность, которая используется при выполнении некоторой функции или операции (преобразования, обработки, формирования и т.д.). Объекты могут иметь динамическую или статическую природу: динамические объекты используются в одном цикле воспроизводства, например заказы на продукцию, счета на оплату, платежи; статические объекты используются во многих циклах воспроизводства, например оборудование, персонал, запасы материалов.

На внешнем уровне детализации модели выделяются основные виды материальных объектов (например, сырье и материалы, полуфабрикаты, готовые изделия, услуги) и основные виды информационных объектов или документов (например, заказы, накладные, счета и т.д.).

На концептуальном уровне построения модели предметной области уточняется состав классов объектов, определяются их атрибуты и взаимосвязи. Таким образом, строится обобщенное представление структуры предметной области.

Далее концептуальная модель на внутреннем уровне отображается в виде файлов базы данных, входных и выходных документов ЭИС. Причем динамические объекты представляются единицами переменной информации или документами, а статические объекты - единицами условно-постоянной информации в виде списков, номенклатур, ценников, справочников, классификаторов. Модель базы данных как постоянно поддерживаемого информационного ресурса отображает хранение условно-постоянной и накапливаемой переменной информации, используемой в повторяющихся информационных процессах.

Функциональная структура

Функция (операция) представляет собой некоторый преобразователь входных объектов в выходные. Последовательность взаимосвязанных по входам и выходам функций составляет бизнес-процесс. Функция бизнес-процесса может порождать объекты любой природы (материальные, денежные, информационные). Причем бизнес-процессы и информационные процессы, как правило, неразрывны, то есть функции материального процесса не могут осуществляться без информационной поддержки. Например, отгрузка готовой продукции осуществляется на основе документа "Заказ", который, в свою очередь, порождает документ "Накладная", сопровождающий партию отгруженного товара.

Функция может быть представлена одним действием или некоторой совокупностью действий. В последнем случае каждой функции может

соответствовать некоторый процесс, в котором могут существовать свои подпроцессы, и т.д., пока каждая из подфункций не будет представлять некоторую недекомпозируемую последовательность действий.

На внешнем уровне моделирования определяется список основных бизнес-функций или видов бизнес-процессов. Обычно таких функций насчитывается 15- 20.

На концептуальном уровне выделенные функции декомпозируются и строятся иерархии взаимосвязанных функций.

На внутреннем уровне отображается структура информационного процесса в компьютере: определяются иерархические структуры программных модулей, реализующих автоматизируемые функции.

Структура управления

В совокупности функций бизнес-процесса возможны альтернативные или циклические последовательности в зависимости от различных условий протекания процесса. Эти условия связаны с происходящими событиями во внешней среде или в самих процессах и с образованием определенных состояний объектов (например, заказ принят, отвергнут, отправлен на корректировку). **События** вызывают выполнение функций, которые, в свою очередь, изменяют состояния объектов и формируют новые события, и т.д., пока не будет завершен некоторый бизнес-процесс. Тогда последовательность событий составляет конкретную реализацию бизнес-процесса.

Каждое событие описывается с двух точек зрения: **информационной и процедурной**. Информационно событие отражается в виде некоторого сообщения, фиксирующего факт выполнения некоторой функции изменения состояния или появления нового. Процедурно событие вызывает выполнение новой функции, и поэтому для каждого состояния объекта должны быть заданы описания этих вызовов. Таким образом, события выступают в связующей роли для выполнения функций бизнес-процессов.

На внешнем уровне определяются список внешних событий, вызываемых взаимодействием предприятия с внешней средой (платежи налогов, процентов по кредитам, поставки по контрактам и т.д.), и список целевых установок, которым должны соответствовать бизнес-процессы (регламент выполнения процессов, поддержка уровня материальных запасов, уровень качества продукции и т.д.).

На концептуальном уровне устанавливаются бизнес-правила, определяющие условия вызова функций при возникновении событий и достижении состояний объектов.

На внутреннем уровне выполняется формализация бизнес-правил в виде триггеров или вызовов программных модулей.

Организационная структура

Организационная структура представляет собой совокупность организационных единиц, как правило, связанных иерархическими и процессными отношениями. Организационная единица - это подразделение, представляющее собой объединение людей (персонала) для выполнения совокупности общих функций или бизнес-процессов. В функционально-

ориентированной организационной структуре организационная единица выполняет набор функций, относящихся к одной функции управления и входящих в различные процессы. В процессно-ориентированной структуре организационная единица выполняет набор функций, входящих в один тип процесса и относящихся к разным функциям управления.

На внешнем уровне строится структурная модель предприятия в виде иерархии подчинения организационных единиц или списков взаимодействующих подразделений.

На концептуальном уровне для каждого подразделения задается организационно-штатная структура должностей (ролей персонала).

На внутреннем уровне определяются требования к правам доступа персонала к автоматизируемым функциям информационной системы.

Техническая структура

Топология определяет территориальное размещение технических средств по структурным подразделениям предприятия, а коммуникация - технический способ реализации взаимодействия структурных подразделений.

На внешнем уровне модели определяются типы технических средств обработки данных и их размещение по структурным подразделениям.

На концептуальном уровне определяются способы коммуникаций между техническими комплексами структурных подразделений: физическое перемещение документов, машинных носителей, обмен информацией по каналам связи и т.д.

На внутреннем уровне строится модель "клиент-серверной" архитектуры вычислительной сети.

Описанные модели предметной области нацелены на проектирование отдельных компонентов ИС: данных, функциональных программных модулей, управляющих программных модулей, программных модулей интерфейсов пользователей, структуры технического комплекса. Для более качественного проектирования указанных компонентов требуется построение моделей, увязывающих различные компоненты ИС между собой. В простейшем случае в качестве таких моделей взаимодействия могут использоваться матрицы перекрестных ссылок: "объекты-функции", "функции-события", "организационные единицы - функции", "организационные единицы - объекты", "организационные единицы - технические средства" и т.д. Такие матрицы не наглядны и не отражают особенности реализации взаимодействий.

Для правильного отображения взаимодействий компонентов ИС важно осуществлять совместное моделирование таких компонентов, особенно с содержательной точки зрения объектов и функций. Методология структурного системного анализа существенно помогает в решении таких задач.

Структурным анализом принято называть метод исследования системы, который начинается с ее общего обзора, а затем детализируется, приобретая иерархическую структуру с все большим числом уровней. Для таких методов характерно: разбиение на уровни абстракции с ограниченным числом элементов (от 3 до 7); ограниченный контекст, включающий только существенные детали каждого уровня; использование строгих формальных

правил записи; последовательное приближение к результату. Структурный анализ основан на двух базовых принципах - "разделяй и властвуй" и принципе иерархической упорядоченности. Решение трудных проблем путем их разбиения на множество меньших независимых задач (так называемых "черных ящиков") и организация этих задач в древовидные иерархические структуры значительно повышают понимание сложных систем. Определим ключевые понятия структурного анализа.

Операция - элементарное (неделимое) действие, выполняемое на одном рабочем месте.

Функция - совокупность операций, сгруппированных по определенному признаку.

Бизнес-процесс - связанная совокупность функций, в ходе выполнения которой потребляются определенные ресурсы и создается продукт (предмет, услуга, научное открытие, идея), представляющий ценность для потребителя.

Подпроцесс - это бизнес-процесс, являющийся структурным элементом некоторого бизнес-процесса и представляющий ценность для потребителя.

Бизнес-модель - структурированное графическое описание сети процессов и операций, связанных с данными, документами, организационными единицами и прочими объектами, отражающими существующую или предполагаемую деятельность предприятия.

Существуют различные методологии структурного моделирования предметной области, среди которых следует выделить

Лабораторная работа № 9

Анализ рисков информационной безопасности

Задание

1. Загрузите ГОСТ Р ИСО/МЭК ТО 13335-3-2007 «МЕТОДЫ И СРЕДСТВА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ. Часть 3 «Методы менеджмента безопасности информационных технологий»

2. Ознакомьтесь с Приложениями С, D и E ГОСТа.

3. Выберите три различных информационных актива организации (см. вариант).

4. Из Приложения D ГОСТа подберите три конкретных уязвимости системы защиты указанных информационных активов.

5. Пользуясь Приложением С ГОСТа напишите три угрозы, реализация которых возможна пока в системе не устранены названные в пункте 4 уязвимости.

6. Пользуясь одним из методов (см. вариант) предложенных в Приложении E ГОСТа произведите оценку рисков информационной безопасности.

7. Оценку ценности информационного актива производить на основании возможных потерь для организации в случае реализации угрозы.

Обоснование выбора информационных активов организации

1. База сотрудников, хранящаяся на сервере –она нужна для издательства, так как на ней хранится информация о всех сотрудниках издательства.

2. Хранилище на электронном носителе- содержит всю информацию нужную для издательства. Информация доступна только определённым лицам и не должна попасть в руки злоумышленников

3. Бухгалтерская документация –информация затраты издательства, зарплата сотрудникам

Оценка ценности информационных активов

В этом издательстве мы выделили 3 важных актива и поставим ему оценку от 0 до 4.

1. База сотрудников информация о сотрудниках. Оценка этого актива 2.

Возможный ущерб:

нарушение законов и/или подзаконных актов.

1. Хранилище на электронном носителе оценивается тем что находится в нём. Оценка этого актива 3. Возможный ущерб: потеря престижа/негативное воздействие на репутацию

2. Бухгалтерская документация самые цены сведения. Оценка этого актива 4. Возможный ущерб: финансовые потери, нарушение конфиденциальности коммерческой информации, снижение эффективности бизнеса.

Уязвимости системы защиты информации

1. Размещение в зонах возможного затопления (возможна, например, угроза затопления).

2. Отсутствие механизмов идентификации и аутентификации, например, аутентификации пользователей (возможна, например, угроза нелегального проникновения злоумышленников под видом законных пользователей)

3. Отсутствие необходимых знаний по вопросам безопасности (возможна, например, угроза ошибок пользователей).

Угрозы ИБ

1. затопление. Размещение в зонах возможного затопления

В зависимости от того на сколько затоплено помещение, можно сказать высока ли угроза потери важнейшей информации. При затопленности, общего помещения, например, из-за наводнения, потери будут колоссальны. Вся информация будет уничтожена.

2. Угроза нелегального проникновения злоумышленников под видом законных пользователей. Могут быть выкрадены очень важные документы для организации. И приведет к упадку Издательства.

3. Угроза ошибок пользователей. Не значительна ошибка. Вызванная сотрудником Издательства.

Оценка рисков

Вопросы:

База сотрудников:

При угрозе затопление вероятность возникновения угрозы мала, а легкость возникновения угрозы в уязвимых местах имеет высокое значение, то частота будет равна 2.

При угрозе нелегального проникновения злоумышленников под видом законных пользователей вероятность возникновения угрозы высока, а легкость возникновения угрозы в уязвимых местах имеет среднее значение, то частота будет равна 3.

При угрозе ошибок пользователей вероятность возникновения угрозы высока, а легкость возникновения угрозы в уязвимых местах имеет высокое значение, то частота будет равна 4.

Хранилище на электронном носителе:

При угрозе затопление вероятность возникновения угрозы мала, а легкость возникновения угрозы в уязвимых местах имеет высокое значение, то частота будет равна 2.

При угрозе нелегального проникновения злоумышленников под видом законных пользователей вероятность возникновения угрозы высока, а легкость возникновения угрозы в уязвимых местах имеет среднее значение, то частота будет равна 3.

При угрозе ошибок пользователей вероятность возникновения угрозы высока, а легкость возникновения угрозы в уязвимых местах имеет высокое значение, то частота будет равна 4.

Бухгалтерская документация:

При угрозе затопление вероятность возникновения угрозы мала, а легкость возникновения угрозы в уязвимых местах имеет высокое значение, то частота будет равна 2.

При угрозе нелегального проникновения злоумышленников под видом законных пользователей вероятность возникновения угрозы высока, а легкость возникновения угрозы в уязвимых местах имеет среднее значение, то частота будет равна 3.

При угрозе ошибок пользователей вероятность возникновения угрозы высока, а легкость возникновения угрозы в уязвимых местах имеет высокое значение, то частота будет равна 4.

Ценность актива	Уровень угрозы								
	Низкий			Средний			Высокий		
	Уровень уязвимости			Уровень уязвимости			Уровень уязвимости		
	Н	С	В	Н	С	В	Н	С	В
0	0	1	2	1	2	3	2	3	4
1	1	2	3	2	3	4	3	4	5
2	2	3	4	3	4	5	4	5	6
3	3	4	5	4	5	6	5	6	7
4	4	5	6	5	6	7	6	7	8

Обозначение: Н — низкий, С — средний, В — высокий.

Лабораторная работа № 13

Разработка требований безопасности информационной системы

1. Цель работы:

Составить и проанализировать требования к информационной системе, оформить техническое задание на разработку программного обеспечения.

2. Методические указания

Лабораторная работа направлена на ознакомление с процессом разработки требований к информационной системе и составления технического задания на разработку программного обеспечения, получение навыков по использованию основных методов формирования и анализа требований.

Требования к результатам выполнения лабораторного практикума:

1. наличие диаграммы идентификации точек зрения и диаграммы иерархии точек зрения;
2. наличие пользовательских требований, четко описывающих будущий функционал системы;
3. наличие системных требований, включающих требования к структуре, программному интерфейсу, технологиям разработки, общие требования к системе (надежность, масштабируемость, распределённость, модульность, безопасность, открытость, удобство пользования и т.д.);
4. наличие составленного технического задания.

3. Теоретические сведения

Общие сведения о требованиях к информационным системам

Проблемы, которые приходится решать специалистам в процессе создания программного обеспечения, очень сложны. Природа этих проблем не всегда ясна, особенно если разрабатываемая программная система инновационная. В частности, трудно чётко описать те действия, которые должна выполнять система. Описание функциональных возможностей и ограничений, накладываемых на систему, называется требованиями к этой системе, а сам процесс формирования, анализа, документирования и проверки этих функциональных возможностей и ограничений – разработкой требований.

Требования подразделяются на пользовательские и системные. Пользовательские требования – это описание на естественном языке (плюс поясняющие диаграммы) функций, выполняемых системой, и ограничений, накладываемых на неё. Системные требования – это описание особенностей системы (архитектура системы, требования к параметрам оборудования и т.д.), необходимых для эффективной реализации требований пользователя.

Разработка требований

Разработка требований — это процесс, включающий мероприятия, необходимые для создания и утверждения документа, содержащего спецификацию системных требований. Различают четыре основных этапа процесса разработки требований:

- анализ технической осуществимости создания системы,
- формирование и анализ требований,
- специфицирование требований и создание соответствующей документации,

- аттестация этих требований.

На рис. 1 показаны взаимосвязи между этими этапами и результаты, сопровождающие каждый этап процесса разработки системных требований.

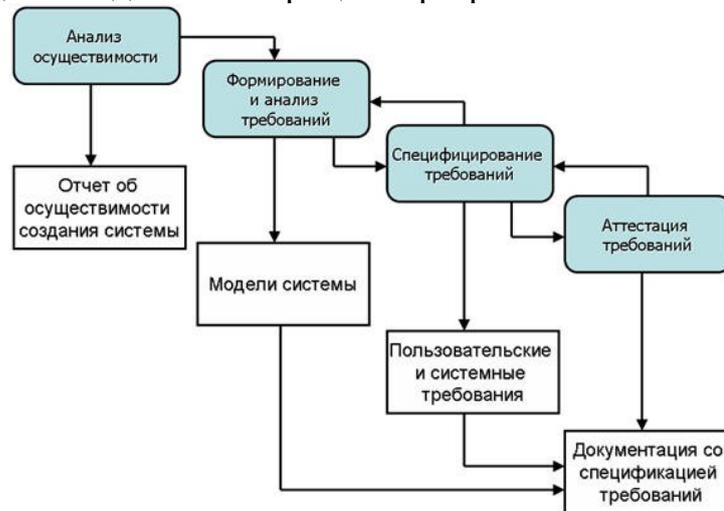


Рис. 1. Процесс разработки требований

Но поскольку в процессе разработки системы в силу разнообразных причин требования могут меняться, управление требованиями, т.е. процесс управления изменениями системных требований, является необходимой составной частью деятельности по их разработке.

Формирование и анализ требований

Следующим этапом процесса разработки требований является формирование (определение) и анализ требований.

Обобщенная модель процесса формирования и анализа требований показана на рис. 2. Каждая организация использует собственный вариант этой модели, зависящий от “местных факторов”: опыта работы коллектива разработчиков, типа разрабатываемой системы, используемых стандартов и т.д.

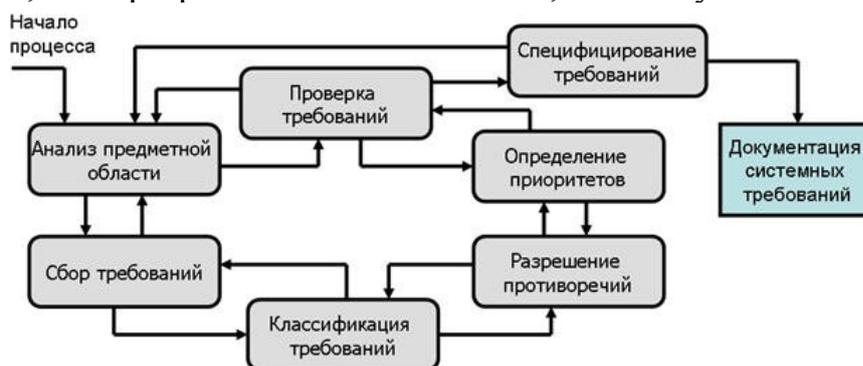


Рис. 2. Процесс формирования и анализа требований

Процесс формирования и анализа требований проходит через ряд этапов.

1. *Анализ предметной области.* Аналитики должны изучить предметную область, где будет эксплуатироваться система.

2. *Сбор требований.* Это процесс взаимодействия с лицами, формирующими требования. Во время этого процесса продолжается анализ предметной области.

3. *Классификация требований.* На этом этапе бесформенный набор требований преобразуется в логически связанные группы требований.

4. *Разрешение противоречий.* Без сомнения, требования многочисленных лиц, занятых в процессе формирования требований, будут противоречивыми. На этом этапе определяются и разрешаются противоречия различного рода.

5. *Назначение приоритетов.* В любом наборе требований одни из них будут более важны, чем другие. На этом этапе совместно с лицами, формирующими требования, определяются наиболее важные требования.

6. *Проверка требований.* На этом этапе определяется их полнота, последовательность и непротиворечивость.

Процесс формирования и анализа требований циклический, с обратной связью от одного этапа к другому. Цикл начинается с анализа предметной области и заканчивается проверкой требований. Понимание требований предметной области увеличивается в каждом цикле процесса формирования требований.

Рассмотрим три основных подхода к формированию требований: метод, основанный на множестве опорных точек зрения, сценарии и этнографический метод.

Опорные точки зрения

Подход с использованием различных опорных точек зрения к разработке требований признает различные (опорные) точки зрения на проблему и использует их в качестве основы построения и организации как процесса формирования требований, так и непосредственно самих требований.

Различные методы предлагают разные трактовки выражения "точка зрения". Точки зрения можно трактовать следующим образом.

1. *Как источник информации о системных данных.* В этом случае на основе опорных точек зрения строится модель создания и использования данных в системе. В процессе формирования требований отбираются все такие точки зрения (и на их основе определяются данные), которые будут созданы или использованы при работе системы, а также способы обработки этих данных.

2. *Как структура представлений.* В этом случае точки зрения рассматриваются как особая часть модели системы. Например, на основе различных точек зрения могут разрабатываться модели "сущность-связь", модели конечного автомата и т.д.

3. *Как получатели системных сервисов.* В этом случае точки зрения являются внешними (относительно системы) получателями системных сервисов. Точки зрения помогают определить данные, необходимые для выполнения системных сервисов или их управления.

Наиболее эффективным подходом к анализу таких систем является использование внешних опорных точек зрения. На основе этого подхода разработан метод VORD (Viewpoint-Oriented Requirements Definition — определение требований на основе точек зрения) для формирования и анализа требований. Основные этапы метода VORD показаны на рис. 3.

1. Идентификация точек зрения, получающих системные сервисы, и идентификация сервисов, соответствующих каждой точке зрения.

2. Структурирование точек зрения — создание иерархии сгруппированных точек зрения. Общесистемные сервисы предоставляются более высоким уровням иерархии и наследуются точками зрения низшего уровня.

3. Документирование опорных точек зрения, которое заключается в точном описании идентифицированных точек зрения и сервисов.

4. Отображение системы точек зрения, которая показывает системные объекты, определенные на основе информации, заключенной в опорных точках зрения.



Рис. 3. Метод VORD

Пример. Рассмотрим использование метода VORD на первых трех шагах анализа требований для системы поддержки заказа и учета товаров в бакалейной лавке. В бакалейной лавке для каждого товара фиксируется место хранения (определенная полка), количество товара и его поставщик. Система поддержки заказа и учета товаров должна обеспечивать добавление информации о новом товаре, изменение или удаление информации об имеющемся товаре, хранение (добавление, изменение и удаление) информации о поставщиках, включающей в себя название фирмы, ее адрес и телефон. При помощи системы составляются заказы поставщикам. Каждый заказ может содержать несколько позиций, в каждой позиции указываются наименование товара и его количество в заказе. Система по требованию пользователя формирует и выдает на печать следующую справочную информацию:

- список всех товаров;
- список товаров, имеющих в наличии;
- список товаров, количество которых необходимо пополнить;
- список товаров, поставляемых данным поставщиком.

Первым шагом в формировании требований является идентификация опорных точек зрения. Во всех методах формирования требований, основанных на использовании точек зрения, начальная идентификация является наиболее трудной задачей. Один из подходов к идентификации точек зрения — метод "мозговой атаки", когда определяются потенциальные системные сервисы и организации, взаимодействующие с системой. Организуется встреча лиц, участвующих в формировании требований, которые предлагают свои точки зрения. Эти точки зрения представляются в виде диаграммы, состоящей из ряда круговых областей, отображающих возможные точки зрения (рис. 4). Во время "мозговой атаки" необходимо идентифицировать потенциальные опорные точки зрения, системные сервисы, входные данные, нефункциональные требования, управляющие события и исключительные ситуации.

Следующей стадией процесса формирования требований будет идентификация опорных точек зрения (на рис. 4 показаны в виде темных

круговых областей) и сервисов (показаны в виде затененных областей). Сервисы должны соответствовать опорным точкам зрения. Но могут быть сервисы, которые не поставлены им в соответствие. Это означает, что на начальном этапе "мозговой атаки" некоторые опорные точки зрения не были идентифицированы.

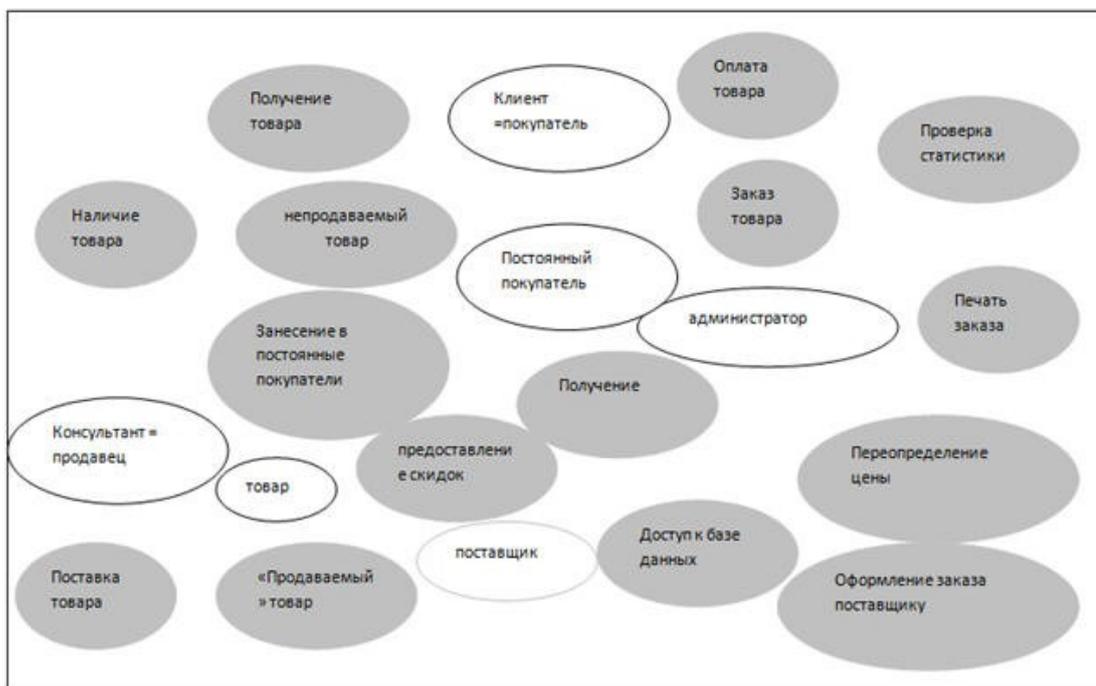


Рис. 4. Диаграмма идентификации точек зрения

В таблице 1 показано распределение сервисов для некоторых идентифицированных на рис. 4 точек зрения. Один и тот же сервис может быть соотнесен с несколькими точками зрения.

Таблица 1 - Сервисы, соотнесенные с точками зрения

клиент	покупатель	постоянный покупатель	товар	поставщик	продавец	администратор
Проверка наличия товара	Занесение в список постоянных клиентов	Получение скидки	Прием товара	Занесение в базу данных (название, адрес, телефон и т.д.)	Продажа товара	Доступ к базе данных
Покупка товара		Получение информацию о новых поступлениях	Занесение в базу данных (данные о поставщике, кол-ве, месте хранения и.д.)		Печать чека	Проверка статистики
Получение чека			Назначение цены		Доступ к каталогу	Переопределение цены
Заказ товара			Переопределение цены		Проверка наличия товара	Оформление заказа поставщику
Занесение покупателя и			«Покупаемый» или		Оформление заказа	Печать заказа

клиент	покупатель	постоянный покупатель	товар	поставщик	продавец	администратор
суммы покупки в базу данных			«непокупаемый» товар		покупателю	

Информация, извлеченная из точек зрения, используется для заполнения форм шаблонов точек зрения и организации точек зрения в иерархию наследования. Это позволяет увидеть общие точки зрения и повторно использовать информацию в иерархии наследования. Сервисы, данные и управляющая информация наследуются подмножеством точек зрения. На рис. 5 показана часть иерархии точек зрения для системы поддержки заказа и учета товаров.



Рис. 5. Иерархия точек зрения

Аттестация 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 после введения ее в эксплуатацию. Стоимость внесения в систему изменений, необходимых для устранения ошибок в требованиях, намного выше, чем исправление ошибок проектирования или кодирования. Причина в том, что изменение требований обычно влечет за собой значительные изменения в системе, после внесения которых она должна пройти повторное тестирование.

Во время процесса аттестации должны быть выполнены различные типы проверок требований.

1. *Проверка правильности требований.* Пользователь может считать, что система необходима для выполнения некоторых определенных функций. Однако дальнейшие размышления и анализ могут привести к необходимости введения дополнительных или новых функций. Системы предназначены для разных пользователей с различными потребностями, и поэтому набор требований будет представлять собой некоторый компромисс между требованиями пользователей системы.

2. *Проверка на непротиворечивость.* Спецификация требований не должна содержать противоречий. Это означает, что в требованиях не должно быть противоречащих друг другу ограничений или различных описаний одной и той же системной функции.

3. *Проверка на полноту.* Спецификация требований должна содержать требования, которые определяют все системные функции и ограничения, налагаемые на систему.

4. *Проверка на выполнимость.* На основе знания существующих технологий требования должны быть проверены на возможность их реального выполнения. Здесь также проверяются возможности финансирования и график разработки системы.

Существует ряд методов аттестации требований, которые можно использовать совместно или каждый в отдельности.

1. *Обзор требований.* Требования системно анализируются рецензентами.

2. *Прототипирование.* На этом этапе прототип системы демонстрируется конечным пользователям и заказчику. Они могут экспериментировать с этим прототипом, чтобы убедиться, что он отвечает их потребностям.

3. *Генерация тестовых сценариев.* В идеале требования должны быть такими, чтобы их реализацию можно было протестировать. Если тесты для требований разрабатываются как часть процесса аттестации, то часто это позволяет обнаружить проблемы в спецификации. Если такие тесты сложно или невозможно разработать, то обычно это означает, что требования трудно выполнить и поэтому необходимо их пересмотреть.

4. *Автоматизированный анализ непротиворечивости.* Если требования представлены в виде структурных или формальных системных моделей, можно использовать инструментальные CASE-средства для проверки непротиворечивости моделей. Для автоматизированной проверки непротиворечивости необходимо построить базу данных требований и затем проверить все требования в этой базе данных. Анализатор требований готовит отчет обо всех обнаруженных противоречиях.

Пользовательские и системные требования

На основании полученных моделей строятся пользовательские требования, т.е. как было сказано в начале описание на естественном языке функции, выполняемых системой, и ограничений, накладываемых на неё.

Пользовательские требования должны описывать внешнее поведение системы, основные функции и сервисы предоставляемые системой, её нефункциональные свойства. Необходимо выделить опорные точки зрения и сгруппировать требования в соответствии с ними. Пользовательские требования можно оформить как простым перечислением, так и используя нотацию вариантов использования.

Далее составляются системные требования. Они включают в себя:

1. Требования к архитектуре системы. Например, число и размещение хранилищ и серверов приложений.

2. Требования к параметрам оборудования. Например, частота процессоров серверов и клиентов, объём хранилищ, размер оперативной и видео памяти, пропускная способность канала и т.д.

3. Требования к параметрам системы. Например, время отклика на действие пользователя, максимальный размер передаваемого файла, максимальная скорость передачи данных, максимальное число одновременно работающих пользователей и т.д.

4. Требования к программному интерфейсу.

5. Требования к структуре системы. Например, Масштабируемость, распределённость, модульность, открытость.

- масштабируемость – возможность распространения системы на большое количество машин, не приводящая к потере работоспособности и эффективности, при этом способность системы наращивать свою мощность должна определяться только мощностью соответствующего аппаратного обеспечения.

- распределенность - система должна поддерживать распределённое хранение данных.

- модульность - система должна состоять из отдельных модулей, интегрированных между собой.

- открытость - наличие открытых интерфейсов для возможной доработки и интеграции с другими системами.

6. Требования по взаимодействию и интеграции с другими системами. Например, использование общей базы данных, возможность получения данных из баз данных определённых систем и т.д.

4. Порядок выполнения работы

1. Изучить предлагаемый теоретический материал.

2. Построить опорные точки зрения на основании метода VORD для формирования и анализа требований. Результатом должны явиться две диаграммы: диаграмма идентификации точек зрения и диаграмма иерархии точек зрения.

3. Составить информационную модель будущей системы, включающую в себя описание основных объектов системы и взаимодействия между ними. На основании полученной информационной модели и диаграмм идентификации точек зрения, диаграмма иерархии точек зрения сформировать требования пользователя и системные требования.

4. Провести аттестацию требований, указать какие типы проверок выбрали.

5. На основании описания системы (лабораторная работа № 1), информационной модели, пользовательских и системных требований составить техническое задание на создание программного обеспечения.

6. Построить отчёт, включающий все полученные уровни модели, описание функциональных блоков, потоков данных, хранилищ и внешних объектов.

5. Содержание отчета

В отчете следует указать:

1. Цель работы

2. Введение

3. Программно-аппаратные средства, используемые при выполнении работы.

4. Основная часть (описание самой работы), выполненная согласно требованиям к результатам выполнения лабораторного практикума (п.2).

5. Заключение (выводы)

6. Список используемой литературы

6. Литература

1. Соммервиль Иан. Инженерия программного обеспечения, 6-е издание. : Пер. с англ. – М.: Издательский дом “Вильямс”, 2002. – 624 с.

2. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. – СПб.:Питер, 2002. – 496 с.

3. Константайн Л., Локвуд Л. Разработка программного обеспечения. – СПб.:Питер, 2004. – 592 с.

4. Иванова Г.С. Технология программирования: Учебник для вузов. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. - 320 с.

5. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы

6. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению

ЛАБОРАТОРНАЯ РАБОТА №14

Внутренние проверки качества информационной системы

Процесс оценки качества программного обеспечения осуществляется для каждой фазы его жизненного цикла и включает:

- выбор совокупности (номенклатуры) показателей качества оцениваемого программного средства;
- определение значений этих показателей;
- сравнение полученных значений с базовыми значениями показателей.

Под жизненным циклом программного обеспечения понимается период времени с момента начала предпроектного обследования до момента полного выхода программы из употребления пользователями.

Весь период жизненного цикла программного обеспечения делится на следующие временные промежутки или фазы.

1. Анализ - этап определения требований к программному обеспечению, спецификация требований и формирования технического задания на проектирование программы.

2. Проектирование - этап разработки технического проекта.

3. Реализация - этап разработки программного обеспечения, средств тестирования и документации.

4. Тестирование - этап испытания программного обеспечения и устранение недостатков.

5. Изготовление - этап преобразования программного обеспечения в форму, готовую для поставки; завершение формирования документации.

6. Внедрение - этап подтверждения стабильной работы программного обеспечения; ввод в стадию активного применения.

7. Эксплуатация - этап применения программного обеспечения по назначению.

8. Сопровождение - этап устранения дефектов в процессе эксплуатации; усовершенствование, оптимизация и модификация используемого программного обеспечения при условии сохранности целостности программного продукта.

Оценка качества программного обеспечения на всех фазах жизненного цикла осуществляется на основе четырёхуровневой системы показателей.

Показатели первого уровня (факторы качества) характеризуют потребительски-ориентированные свойства программных средств, которые соответствуют потребностям пользователей. Факторы качества, собственно, определяют наиболее значимые (с точки зрения использования) свойства программ. Для оценки качества программного обеспечения используют следующие факторы:

- надёжность;
- сопровождаемость;
- удобство применения;
- эффективность;
- универсальность;
- корректность.

Каждый фактор представляет собой интегральную оценку, которой соответствует несколько критериев качества (комплексных показателей второго уровня).

В качестве примера в табл. 6.1 представлен состав фактора универсальность по критериям и метрикам для различных фаз жизненного цикла программных средств. Здесь символом «+» отмечены фазы жизненного цикла, на которых определяются значения указанных метрик для расчета соответствующих критериев и факторов. Пустые ячейки таблицы, соответствующие метрикам и фазам жизненного цикла, означают, что указанные метрики на указанных в таблице фазах не определяются.

Таблица 1. Состав и соответствие показателей качества программных систем на различных фазах их жизненного цикла

Факторы	Критерии	Метрики	Фаза жизненного цикла					
			Анализ	Проектирование	Реализация	Тестирование	Изготовление	Сопровождение
Универсальность	Гибкость	Широта охвата функций	+	+	+	+	+	+
		Простота архитектуры проекта		+	+	+	+	+
		Сложность архитектуры проекта		+	+	+	+	+
		Сложность структуры кода программы			+	+	+	+
		Применение стандартных протоколов связи			+	+	+	+
		Применение стандартных интерфейсных программ			+	+	+	+
Факторы	Критерии	Метрики	Фаза жизненного цикла					
			Анализ	Проектирование	Реализация	Тестирование	Изготовление	Сопровождение
		Сложность архитектуры проекта		+	+	+	+	+
		Сложность структуры кода программы			+	+	+	+
		Применение стандартных протоколов связи			+	+	+	+
		Применение стандартных интерфейсных программ			+	+	+	+
Универсальность	Мобильность	Зависимость от используемого комплекса технических средств	+		+	+	+	+
		Зависимость от базового программного обеспечения	+		+	+	+	+
		Изоляция немобильности	+		+	+	+	+
	Модифицируемость	Простота кодирования			+	+	+	+
		Число комментариев			+	+	+	+
		Качество комментариев			+	+	+	+
		Использование описательных средств языка			+	+	+	+
		Независимость модулей			+	+	+	+

Оценить качество программного обеспечения по фактору универсальность для фазы жизненного цикла сопровождение по результатам оценки программы экспертами. Результаты проведенной экспертизы представлены в табл. 2. При проведении расчетов число экспертов окончательно определено в индивидуальном задании (см. табл. 3). Базовые значения критериев универсальности - 0,5.

Таблица 2. Значения оценочных элементов универсальности

Критерии	Метрики	Оценочные элементы	Оценки				
			1	2	3	4	5
1. Гибкость	1. Широта охвата функций	1. Оценка числа потенциальных пользователей	0,32	0,54	0,89	0,88	0,5
		2. Оценка числа функций программы	0,78	0,14	0,43	0,92	0,87
1. Гибкость	1. Широта охвата функций	3. Насколько набор функций удовлетворяет требованиям пользователя	0,2	0,91	0,14	0,98	0,6
		4. Насколько возможности программ охватывают область решаемых пользователем задач	0,71	0,64	0,31	0,94	0,05
		5. Возможность настройки формата входных данных для конкретных пользователей	0,99	0,04	0,57	0,94	0,31
	2. Простота архитектуры проекта	1. Наличие схемы иерархии модулей программ	0,3	0,4	0,1	0,48	0,97
		2. Оценка независимости модулей	0,25	0,06	0,35	0,8	0,57
		3. Оценка числа уникальных элементов	0,41	0,28	0,77	0,74	0,27
		4. Используется ли в текущем вызове модуля информация, полученная в предыдущем вызове	0,85	0,55	0,91	0,25	0,99
		5. Оценка организации точек входа и выхода модуля	0	0,01	0,21	0,9	0,84
		6. Наличие описания атрибутов модуля	0,56	0,2	0,21	0,12	0,51
	3. Сложность архитектуры проекта	1. Оценка программ по числу переходов и точек ветвления	0,71	0,1	0,9	0,89	0,92
	4. Сложность структуры кода программы	1. Использование метода пошагового уточнения	0,36	0,35	0,69	0,91	0,65
		2. Наличие описания структуры программы	0,02	0,21	0,67	0,73	0,38
		3. Наличие описания связей между элементами структуры программы	0,23	0,4	0,17	0,09	0,47
		4. Наличие в программе повторного выполнения функций (подпрограмм)	0,36	0,04	0,4	0,8	0,6

Критерии	Метрики	Оценочные элементы	Оценки				
			1	2	3	4	5
1. Гибкость	5. Применение стандартных протоколов связи	1. Использование стандартных протоколов связи	0,36	0,19	0,52	0,06	0,64
	6. Применение стандартных интерфейсных программ	1. Использование стандартных интерфейсных подпрограмм	0,86	0,96	0,16	0,54	0,91
2. Мобильность	1. Зависимость от используемого комплекса технических средств	1. Оценка зависимости программ от емкости оперативной памяти	0,14	0,31	0,96	0,19	0,68
		2. Оценка зависимости временных характеристик программы от скорости вычислений ЭВМ	0,53	0,9	0,83	0,98	0,73
		3. Оценка зависимости функционирования программы от числа внешних запоминающих устройств и их общей емкости	0,36	0,54	0,96	0,53	0,85
		4. Оценка зависимости функционирования программы от специальных устройств ввода-вывода	0,75	0,97	0,24	0,33	0,71
		5. Оценка зависимости программ от емкости оперативной памяти	0,49	0,5	0,67	0,81	0,78
	2. Зависимость от базового программного обеспечения	1. Применение специальных языков программирования	0,72	0,07	0,24	0,02	0,79
		2. Оценка зависимости программы от программ операционной системы	0,36	0,41	0,49	0,88	0,91
		3. Зависимость от других программных средств	0,12	0,54	0,43	0,16	0,52
	3. Изоляция немобильности	1. Оценка локализации переносимой части программы	0,63	0,9	0,83	0,41	0,58

Критерии	Метрики	Оценочные элементы	Оценки					
			1	2	3	4	5	
3. Модифицируемость	1. Простота кодирования	1. Оценка использования отрицательных или булевых выражений	0,36	0,91	0,86	0,14	0,98	
		2. Оценка программы по использованию условных переходов	0,52	0,86	0,26	0,31	0,94	
		3. Оценка программы по использованию безусловных переходов	0,86	0,12	0,94	0,57	0,94	
		4. Оформление процедур входа и выхода из циклов	0,99	0,25	0,15	0,1	0,48	
		5. Ограничения на модификацию переменной индексации в цикле	0,5	0,05	0,23	0,35	0,8	
		6. Оценка модулей по направлению потока управления	0,35	0,89	0,03	0,77	0,74	
		7. Оценка программы по использованию локальных переменных	0,8	0,96	0,07	0,91	0,25	
	2. Число комментариев	1. Оценка программы по числу комментариев	0,83	0,42	0,6	0,21	0,9	
	3. Качество комментариев	1. Наличие заголовка в программе	0,01	0,41	0,21	0,21	0,12	
		2. Комментарии к точкам ветвления	0,49	0,72	0,42	0,9	0,89	
		3. Комментарии к машинозависимым частям программы	0,35	0,89	0,84	0,69	0,91	
		4. Комментарии к машинозависимым операторам программы	0,24	0,57	0,3	0,67	0,73	
		5. Комментарии к операторам объявления переменных	0,17	0,82	0,14	0,17	0,09	
		6. Оценка семантики операторов	0,37	0,19	0,17	0,4	0,8	
		7. Наличие соглашений по форме представления комментариев	0,61	0,04	0,16	0,52	0,06	
		8. Наличие общих комментариев к программам	0,33	0,61	0,67	0,16	0,54	
	Критерии	4. Использование описательных средств языка	1. Использование языков высокого уровня	0,2	0,66	0,11	0,96	0,19
			2. Семантика имен используемых переменных	0,81	0,63	0,48	0,83	0,98
3. Использование отступов, сдвигов и пропусков при формировании текста			0,9	0,42	0	0,96	0,53	
4. Размещение операторов по строкам			0,94	0,04	0,55	0,24	0,33	
5. Независимость модулей		1. Передача информации для управления по параметрам	0,16	0,75	0,15	0,67	0,81	
		2. Параметрическая передача входных данных	0,59	0,63	0,87	0,24	0,02	
		3. Наличие передачи результатов работы между модулями	0,64	0,17	0,86	0,49	0,88	
		4. Наличие проверки правильности данных, получаемых модулями от вызываемого модуля	0,64	0,95	0,21	0,43	0,16	
		5. Использование общих областей памяти	0,32	0,54	0,42	0,83	0,41	

Лабораторная работа 15

Реинжиниринг бизнес-процессов методом горизонтального и/или вертикального сжатия

Теоретические сведения

Особенности разработки и внедрения проектов
реинжиниринга в бизнес-процессы

Горизонтальное сжатие процесса означает объединение нескольких рабочих процедур (задач-операций) в одну работу. При этом компания избавляется от «сборочного конвейера», когда отдельные процедуры производственного процесса распределяются по отдельным рабочим местам, а исполнители подчиняются различным подразделениям компании. Весь новый процесс выполняется или одним сотрудником, или командой в несколько человек, что позволяет ускорить его на порядок. Достоинствами данного варианта являются:

— уменьшение числа ошибок и сокращение числа работников, занятых устранением последствий этих ошибок;

— улучшение управляемости производства за счет уменьшения числа людей и четко распределенной ответственности между ними.

Вертикальное сжатие процессов — предоставление исполнителям права принимать самостоятельные решения. При традиционной организации работ в компании для принятия решений исполнители должны обращаться к управленческой иерархии (поскольку считается, что сами они не имеют ни времени, ни склонности, ни глубоких и всесторонних знаний, необходимых для принятия решений). БПР расширяет полномочия и повышает роль каждого из сотрудников в работе компании, что приводит к уменьшению времени задержек, снижению стоимости продукции и услуг, ускорению реакции на запросы клиентов и т.д.

Выполнение шагов каждого процесса в естественном порядке, использование, наряду с линейным упорядочением работ, параллельное их проведение, что приводит к сокращению времени и затрат. Характерным примером здесь является метод одновременной инженерии, использованный в конце 80-х лет фирмой Kodac, который предусматривает при разработке и проектировании новых изделий последовательно-параллельный подход на основе компьютерной технологии CAD/CAM и интегрированной БД. Идея метода состоит в отказе от чертежей на бумаге и проектировании изделий непосредственно на экране ЭВМ, а также использовании БД, хранящей текущее состояние проекта. Ежедневно в БД добавлялись результаты, полученные параллельно работающими группами проектировщиков, после чего эта результаты инспектировались самими проектировщиками на предмет обнаружения взаимных несоответствий или ошибок. Обнаруженные дефекты исправлялись немедленно в отличие от традиционной технологии, где «работу над ошибками» можно было сделать через недели и месяцы, когда параллельно разработанные части изделия могли быть собраны вместе. Срок выпуска изделий был сокращен почти вдвое, а стоимость снизилась на 25% -за счет

того, что новый процесс позволил моделировать сборку изделия до его изготовления и выбирать те конструкции, которые дешевле и проще в производстве.

Исполнение процессов «различных вариантах — процесс начинается с проверочной процедуры, цель которой: определить, какой вариант его реализации наиболее подходит в заданной производственной ситуации. В простых случаях обработку запросов выполняет ЭС, обеспечивающая принятие решений и доступ ко всем необходимым данным и инструментариям, в сложных случаях — сотрудник приглашает экспертов. Таким образом, новые процессы по сравнению с традиционным стали более ясными и простыми, поскольку каждый вариант ориентирован только на одну, соответствующую ему, ситуацию.

Выполнение каждой работы там, где это наиболее целесообразно — БПР распределяет работу между подразделениями, устраняя излишнюю интеграцию, что приводит к повышению эффективности процесса в целом. Специалисты, работающие в компании, обычно группируются в функциональные подразделения: расчетный отдел, отдел заказов, транспортный отдел и т.д., силами которых выполняются соответствующие работы. Однако такой способ функционирования часто оказывается чересчур медленным и расточительным, поскольку специалисты отделов вынуждены по многим вопросам обращаться друг к другу, прежде чем приступить к выполнению заказа ищи заявки клиента. После проведения БПР специалисты получают возможность работать более самостоятельно и инициативно, решая все возникающие вопросы в основном своими силами.

Уменьшение числа проверок: На практике стоимость проверок часто превосходит стоимость потерь, которые имели бы место при отсутствии проверок. БПР предлагает вместо проверки каждого из выполняемых заданий укрупнение этих заданий и осуществление проверок и управляющих воздействий в случае необходимости, что сокращает время и стоимость проверок.

Минимизация числа согласований — БПР минимизирует число согласований путем сокращения внешних точек контакта и, как следствие этого стирания граней между функциональными подразделениями.

«Уполномоченный» менеджер обеспечивает единую точку контакта, — если процесс сложен и интеграция невозможна, то «уполномоченный» менеджер:

- ведет себя с заказчиком (клиентом) так, как если бы он был ответственным за весь процесс;
- отвечает на вопросы заказчика (клиента) и решает его проблемы;
- имеет доступ ко всем ИС, используемым в процессе, и ко всем исполнителям.

Преобладает смешанный централизованно-децентрализованный подход, — применение современных информационных технологий дает возможность компании действовать на уровне подразделений автономно (децентрализовано), сохраняя за ними возможность использования централизованных данных.

Например, продавцы товаров и услуг в регионах через мобильные средства связи имеют постоянный доступ к БД в штаб-квартире компании; получают информацию о ценах, спецификации товаров и т.п.; используют ее при составлении контрактов, в результате, во-первых, отпадает надобность в бюрократических региональных структурах, во-вторых, повышаются оперативность и качество обслуживания заказчиков (клиентов). Наличие централизованной БД не препятствует автономной работе подразделений компании и облегчает их работу, позволяя обходиться без лишнего бюрократического аппарата.

Многие сравнивают реинжиниринг с процессом оптимального уменьшения размера корпорации. Однако это имеет место лишь в том случае, если реинжиниринг бизнес-процессов проведен неправильно. Но когда реинжиниринг осуществляется корректно, то эти два метода совершенно различны. Действительно, находясь на грани экономического краха, компания использует процесс оптимального уменьшения своего размера и разрабатывает схему увольнения персонала для того, чтобы добиться необходимой экономии. Когда реинжиниринг применяется правильно, то он концентрируется на целях, определенных в ходе стратегического планирования. Часто реализация этих целей требует увеличения ресурсов. В реинжиниринге сокращение затрат и кадров осуществляется лишь после достижения целей и реализации перепроектирования. В процессе оптимального уменьшения размеров организации о сокращении затрат и персонала часто сообщается прежде, чем любое перепроектирование будет завершено.

Для реинжиниринга ключевым является понятие бизнес-процессов: именно в этом заключается основное отличие реинжиниринга как менеджмента процессов от функционального менеджмента. Бизнес-процесс можно определить как совокупность различных видов деятельности, в рамках которой «на входе» используются один или более видов ресурсов, и в результате этой деятельности «на выходе» создается продукт, представляющий ценность для потребителя.

Для осуществления такого управления необходима реализация нескольких важных факторов.

Первый — менеджмент процессов имеет не вертикальную (иерархическую), а горизонтальную структуру. Компании необходимо снизить, если не разрушить вообще, организационные барьеры и создать рабочие команды, которые были бы кроссфункциональными. Это способствует более прямому контакту с клиентами (и поставщиками) и взаимодействию с другими рабочими единицами.

Второй фактор заключается в том, что менеджмент процессов ставит на первое место, прежде всего, цикл времени. Понятие «цикл времени» можно определить как реально измеряемое время от начала производства до конца, до завершения производства или выполнения заказа или услуги. Это понятие отличается от понятия «время производства». Например, компании требуется 20 минут, чтобы произвести товар, но нужно ждать 15 дней для улаживания финансовой стороны дела и еще 5 дней, пока приказ о выполнении работы

будет доставлен почтой. В этом случае полный цикл времени составляет более 20 дней.

Третьей критической переменной для менеджмента процессов является выбор. Смысл этого фактора заключается в том, что клиент не обязательно должен ждать целый день, когда ему доставят товар, а может выбрать определенное время в течение 48 или 72 часов, и соответствующая служба гарантирует доставку в назначенное время плюс минус полчаса.

Повышенное внимание к клиенту и его интересам является главной причиной применения управления процессами в организации.

Метод управления процессами имеет важное значение для менеджеров и работников компании. Управление процессами предполагает совершенно другой подход к контролю над производством. Данный метод менеджмента означает, что традиционная работа руководителя главным образом в качестве контролирующего лица не создает новой стоимости. Руководитель в управлении процессами является «хозяином процессов», устанавливающим цели команды, инструктирующим, обучающим работников и облегчающим взаимодействие команды и организационную коммуникацию. Процессные команды сами по себе состоят из работников, которых поощряют и обучают для того, чтобы они работали на цели группы, такие как удовлетворение клиента, снижение затрат, ускорение цикла времени и т. д.

Существует связь между управлением процессами и партисипативным управлением (см. Партисипативное управление). Трудно оспорить тот факт, что наделение полномочиями при принятии решения важно, но управление процессами, особенно осуществляемое посредством реинжиниринга, представляет собой нечто большее, чем постановка новых задач для команды. Фактически кроссфункциональная команда, осуществляющая управление процессами, может быть совсем не похожей на самоуправляемые рабочие группы 1990-х гг. Последние, в большинстве случаев, выполняют свою работу без надзирателей, контролеров. Процессные команды принимают решения, обращаясь непосредственно к источнику проблемы, при этом тщательно продумывая внутреннюю и внешнюю коммуникацию, создавая параллельные, а не последовательные, информационные каналы и метрическую систему оценки внутри группы.

Особое внимание следует уделять развитию технологий. Более передовые технологии изменяют выполняемую работу и возможность направлять и координировать ее, в результате чего существенно повышается способность организации увеличивать размеры подразделений, снижать прямой контроль и развивать внутри- и межорганизационную и кроссфункциональную коммуникацию. Децентрализованное принятие решений с участием самостоятельных рабочих групп и проектных команд при поддержке систем информационного контроля все больше и больше превалирует в организационных структурах. Такая направленность на партисипативное управление уменьшает количество уровней в организации и на смену контролирующим функциям руководства приходят функции координации и поддержки. Для этого требуется, чтобы организация пересмотрела роли

руководителя и работника (посредством изменений культуры, образования или стратегий реинжиниринга). Однако для того, чтобы любой тип рабочей команды действовал эффективно, нужно преодолеть основные препятствия, а для этого необходимо тщательное планирование, обучение и подготовка.

Управление процессами является ключевым звеном в стратегии преобразования бюрократических организаций. Преобразования в рамках реинжиниринга нацелены на уменьшение административного регулирования, снижение иерархичности организации, повышение гибкости системы и ее упрощение — эти цели преследуют большинство реформ. Управление процессами предполагает создание совершенно иной структуры и распределение рабочих заданий, налаживание лучшей коммуникации с клиентами и поставщиками.

Существует несколько вопросов, на которые необходимо ответить, прежде чем начинать процесс реинжиниринга компании.

В первую очередь нужно понять, почему необходим реинжиниринг. Последний важен, когда текущая работа организационных систем устаревает и нуждается в радикальных изменениях для того, чтобы удовлетворять экономические потребности и соответствовать реалиям будущего.

Также в процессе реинжиниринга стараются использовать последние достижения в области информационных и коммуникационных технологий, которые позволяют применять самые различные подходы к выполнению работы.

Далее нужно определить, что именно изменяется в процессе реинжиниринга организации. Реинжиниринг концентрируется на ключевых рабочих процессах компании и перестраивает организационный менеджмент и системы поддержки так, чтобы они соответствовали не внутренним, а внешним требованиям. Особое внимание уделяется сокращению и продолжительности процессов производства товаров и услуг, которые добавляют стоимость основного продукта с точки зрения потребителя, совершенствованию технологий и коммуникаций для налаживания прямых связей между поставщиками, производителями и клиентами, созданию кроссфункциональных рабочих команд, переориентированию административного аппарата на поддержку, а не регулирование.

Также нужно представлять себе, для кого проводится реинжиниринг. Реинжиниринг направлен на клиента, на предоставление ему продукта лучшего качества как основы конкурентоспособности организации.

Здесь также нужно решить вопрос о том, когда следует начинать реинжиниринг. Его нужно начинать, когда компания приходит к выводу, что ее положение является неприемлемым и более желательны инновационные изменения, а не обычное улучшение бизнеса.

Для выполнения реинжиниринга существуют три главных требования:

1. Для того, чтобы реинжиниринг способствовал решению приоритетных задач, необходима заинтересованность в нем руководителей высшего звена.

2. Необходимы качественные методы сравнения существующих проблем и ограничений, накладываемых клиентами с прогнозируемыми потребностями и рыночными ожиданиями. Использование качественной методологии также означает реализацию моделей перепроектирования, которые можно испытать на качество и простоту применения.

3. Привлечение лучших экспертов из различных функциональных областей для перепроектирования. Для нового и радикального решения своих проблем с помощью реинжиниринга компания должна привлечь лучших специалистов.

Для выполнения реинжиниринга необходимо распределение обязанностей внутри фирмы. Обычно главный менеджер назначает руководящий комитет на высшем уровне для того, чтобы координировать процесс реинжиниринга в целом. Задачами этой группы являются координация и коммуникация. Для выполнения непосредственно самого реинжиниринга создается высококвалифицированная группа или реинжиниринговая команда, состоящая из экспертов самой фирмы и привлеченных со стороны. Она занимается анализом и разработкой нового проекта. Помимо этого существует некоторый тип внутренней команды выполнения, которая наблюдает за процессом и результатами преобразований. Зачастую вопрос реализации реинжиниринга настолько труден, что компании вынуждены создавать дополнительно переходную и интеграционную команду. Такая команда планирует необходимые изменения в политике проведения работ обучения, в реинвестировании рабочей силы, в привлечении потребителей, в коммуникации и т. д.

Для проведения реинжиниринга компании обычно требуется от 6 до 12 месяцев на анализ и перепроектирование и от 12 до 18 для переориентирования и полной реализации разработок. Если мы используем понятие полного цикла времени, то весь проект реинжиниринга занимает от 24 до 30 месяцев. Но, так как многие его стадии могут идти параллельно и выполняться одновременно, можно достигнуть значительно быстрее.

Реинжиниринг представляет собой не только процесс, но и методологию. Будучи процессом, он помогает компании выбирать объекты реинжиниринга, создавать и транслировать представления об изменениях, перепроектировать и внедрять новый процесс, переходить от старых способов работы к новым, осуществлять и улучшать новые процессы.

Реинжиниринг бизнес-процессов также является определенной методологией, включающий трехступенчатый подход:

1. Картографирование процессов. Составление блок-схем (также расчет затрат и полного цикла времени производства) того, как компания представляет услуги и производит товары (часто называют анализом текущего состояния) и детальное описание того, какие технологические средства используются.

2. Оценка потребителей или заинтересованных лиц. Определение насущных потребностей клиентов в товарах и услугах и их последующих рыночных ожиданий с помощью фокус-групп, опросов и встреч с клиентами.

3. Прогнозирование процессов. Продумывание того, как должны протекать рабочие процессы и разработка новых моделей инноваций и коренных улучшений (часто называют анализом будущего состояния).

ЛАБОРАТОРНАЯ РАБОТА № 16

Построение модели бизнес-процессов, подлежащих автоматизации, с использованием диаграммы деятельности (activity diagram), моделирование поведения объектов в проблемной области с использованием диаграммы состояний (statechart diagram) с помощью CASE-средства Rational Rose Enterprise Edition»

Цель работы: изучить методологию функционального моделирования IDEF0 и получить практические навыки в моделировании предметной области.

Подготовка к лабораторной работе

Ознакомиться с лекционным материалом по теме «Структурный подход при разработке программного обеспечения. Создание моделей бизнес-процессов предметной области» учебной дисциплины «Разработка и стандартизация ПС и ИТ».

Для выполнения лабораторной работы студент должен обладать навыками работы с пакетом Vрwin, справочная информация по использованию которого представлена в первой части данного пособия.

Теоретическая часть. Моделирование бизнес-процессов предметной области

В случае создания ПО для информационной системы управления предприятием (совокупность средств, методов и персонала для обработки, хранения и выдачи информации) структурный анализ начинается с исследования того, **как организована система** управления предприятием, с обследования **функциональной и информационной структуры** системы управления, чтобы понять, как работает организация, которую собираются автоматизировать. Для описания работы предприятия необходимо построить модель. Такая модель должна быть адекватна предметной области; следовательно, она должна содержать в себе знания всех участников бизнес-процессов организации.

По результатам обследования аналитик строит **модель «как есть»**: обобщенную логическую модель исходной предметной области, отображающую ее функциональную структуру, особенности основной деятельности и информационное пространство, в котором эта деятельность осуществляется. Далее создают **модель «как надо»**: усовершенствованную обобщенную логическую модель, отображающую реорганизованную предметную область или ее часть, которая подлежит автоматизации. Эта стадия анализа содержит элементы проектирования. *Структурные, функциональные* модели созданные на ранних этапах проектирования программной систем, помогут проектировщику выявить основные функции и составные части проектируемой системы и, по возможности, обнаружить и устранить существенные ошибки. Функциональные диаграммы предметной области помогут понять, как выполняются отдельные операции организации, которые собираются автоматизировать. На этом уровне **определяются все функции**, которые выполняет объект, и **процессы**, протекающие в объекте (например, подразделениях предприятия) для выполнения функций, определяются **связи** между функциями, между процессами. **Функция -**

преобразование входных потоков в выходные, осуществляемое в соответствии с некоторыми внутренними правилами. Выполнение функции обеспечивает процесс. **Процесс** - совокупность взаимосвязанных действий (**работ**), преобразующих некоторые входные данные в выходные. Каждый процесс характеризуется определенными задачами и методами их решения, исходными данными, полученными от других процессов, и результатами. Для описания работы организации необходимо построить модель и выделить те процессы, которые должны быть автоматизированы.

Наиболее удобным языком моделирования бизнес-процессов является методология функционального моделирования - IDEF0, предложенный более 20 лет назад Дугласом Россом (SoftTech, Inc.) и называвшийся первоначально SADT - Structured Analysis and Design Technique.

В IDEF0 система представляется как совокупность взаимодействующих работ или функций. Такая чисто функциональная ориентация является принципиальной - функции системы анализируются независимо от объектов, которыми они оперируют. Это позволяет более четко смоделировать логику и взаимодействие процессов организации.

Под моделью в IDEF0 понимают описание системы (текстовое и графическое), которое должно дать ответ на некоторые заранее определенные вопросы.

Моделируемая система рассматривается как произвольное подмножество Вселенной. Произвольное потому, что, во-первых, мы сами умозрительно определяем, будет ли некий объект компонентом системы, или мы будем его рассматривать как внешнее воздействие, и, во-вторых, оно зависит от точки зрения на систему. Система имеет границу, которая отделяет ее от остальной Вселенной. Взаимодействие системы с окружающим миром описывается как вход (нечто, что перерабатывается системой), выход (результат деятельности системы), управление (стратегии и процедуры, под управлением которых производится работа) и механизм (ресурсы, необходимые для проведения работы). Находясь под управлением, система преобразует входы в выходы, используя механизмы.

Процесс моделирования какой-либо системы в IDEF0 начинается с определения контекста, т.е. наиболее абстрактного уровня описания системы в целом. В контекст входит определение субъекта моделирования, цели и точки зрения на модель.

Под субъектом понимается сама система, при этом необходимо точно установить, что входит в систему, а что лежит за ее пределами; другими словами, мы должны определить, что мы будем в дальнейшем рассматривать как компоненты системы, а что как внешнее воздействие. На определение субъекта системы будет существенно влиять позиция, с которой рассматривается система, и цель моделирования - вопросы, на которые построенная модель должна дать ответ; другими словами, первоначально необходимо определить область (Scope) моделирования. Описание области как системы в целом, так и ее компонентов является основой построения модели. Хотя предполагается, что в течение моделирования область может корректироваться,

она должна быть в основном сформулирована изначально, поскольку именно область определяет направление моделирования и когда должна быть закончена модель. При формулировании области необходимо учитывать два компонента - широту и глубину. Широта подразумевает определение границ модели - мы определяем, что будет рассматриваться внутри системы, а что снаружи. Глубина определяет, на каком уровне детализации модель является завершённой. При определении глубины системы необходимо не забывать об ограничениях времени - трудоёмкость построения модели растёт в геометрической прогрессии от глубины декомпозиции.

После определения границ модели предполагается, что новые объекты не должны вноситься в моделируемую систему; поскольку все объекты модели взаимосвязаны, внесение нового объекта может быть не просто арифметической добавкой, но в состоянии изменить существующие взаимосвязи. Внесение таких изменений в готовую модель является, как правило, очень трудоёмким процессом (так называемая проблема "плавающей области").

Модели AS-IS и TO-BE. Целью построения функциональных моделей обычно является выявление наиболее слабых и уязвимых мест деятельности организации, анализ преимуществ новых бизнес-процессов и степени изменения существующей структуры организации бизнеса. Анализ недостатков и "узких мест" начинают с построения модели AS-IS (Как есть), т. е. модели существующей организации работы. Модель AS-IS может строиться на основе изучения документации (должностных инструкций, положений о предприятии, приказов, отчетов и т. п.), анкетирования и опроса служащих предприятия, создания фотографии рабочего дня и других источников. Полученная модель AS-IS служит для выявления неуправляемых работ, работ не обеспеченных ресурсами, ненужных и неэффективных работ, дублирующихся работ и других недостатков в организации деятельности предприятия. Исправление недостатков, перенаправление информационных и материальных потоков приводит к созданию модели TO-BE (Как будет) - модели идеальной организации бизнес-процессов. Как правило, строится несколько моделей TO-BE, среди которых определяют наилучший вариант .

Рисунок 2.10 - Схема построения моделей "TO-BE" как результат анализа модели "AS-IS"

Технология проектирования программного обеспечения ИС подразумевает сначала создание модели AS-IS, ее анализ и улучшение бизнес-процессов, т. е. создание модели TO-BE, и только на основе модели TO-BE строится модель данных, прототип и затем окончательный вариант ПО.

Порядок выполнения работы

1. Построить бизнес-модель предметной области, согласно выбранного варианта (Приложение А) с помощью нотации IDEF0.

Требования к бизнес-модели:

- Модель должна отражать бизнес-процессы предметной области (Приложение А).

- Количество бизнес-процессов в контекстной диаграмме модели должно быть не менее 3.

- При командной работе количество бизнес-процессов должно соответствовать количеству разработчиков в команде. Каждый член команды выполняет моделирование одного из бизнес-процессов самостоятельно.

- Наличие в модели не менее 3 уровней: контекстная диаграмма и 2 уровня декомпозиции.

- Бизнес-модель формируется путем слияния диаграмм отдельных бизнес-процессов.

2. Проанализировать построенную модель и сформулировать требования к разрабатываемой программе

3. Оформить отчет по лабораторной работе.

4. Представить отчет по лабораторной работе для защиты.

Защита отчета по лабораторной работе

Отчет по лабораторной работе должен быть оформлен согласно требований СТО ВГУЭС и состоять из следующих структурных элементов:

1. титульный лист;

2. текстовая часть;

3. приложение: разработанная функциональная модель.

Текстовая часть отчета должна включать пункты:

- условие задачи;

- порядок выполнения

- краткие сведения о составе и компонентах построенной функциональной модели.

- список требований к разрабатываемой программе

Защита отчета по лабораторной работе заключается в предъявлении преподавателю полученных результатов в виде файла и демонстрации полученных навыков при ответах на вопросы преподавателя.

Контрольные вопросы

1. Что такое жизненный цикл программного продукта?

2. Дайте определение модели жизненного цикла ПО.

3. Приведите этапы разработки программного средства.

4. Что представляет собой структурный подход к разработке ПС?

5. Что включает в себя этап предпроектного исследования?

6. В чем преимущество построения модели предметной области при разработке ПС?

7. Перечислите особенности методологии SADT?

8. Для чего строят модели AS-IS и TO-BE?

9. Что такое бизнес-процесс?

10. В каких отношениях находятся заказчик и разработчик при выработке требований к программному средству?

Лабораторная работа №17. Разработка технической документации

Цель работы

Освоение навыков выделения требований к программным продуктам и формирования технического задания по государственным стандартам.

Задание

1. Получите задание у преподавателя.
2. Для заданного варианта необходимо разработать техническое задание по одному из гостов.

При выполнении задания используйте теоретический материал приведенный ниже.

Теоретический материал

3.1 Общие сведения о стандартах

Техническое задание является исходным материалом для создания информационной системы или другого продукта. Поэтому техническое задание (сокращенно ТЗ) в первую очередь должно содержать основные технические требования к продукту и отвечать на вопрос, что данная система должна делать, как работать и при каких условиях.

Как правило, этапу составления технического задания предшествует проведение обследования предметной области, которое завершается созданием аналитического отчета. Именно аналитический отчет (или аналитическая записка) ложится в основу документа Техническое задание.

Если в отчете требования заказчика могут быть изложены в общем виде и проиллюстрированы UML-диаграммами, в техническом задании следует подробно описать все функциональные и пользовательские требования к системе. Чем подробнее будет составлено техническое задание, тем меньше спорных ситуаций возникнет между заказчиком и разработчиком во время приемочных испытаний.

Таким образом, техническое задание является документом, который позволяет как разработчику, так и заказчику представить конечный продукт и впоследствии выполнить проверку на соответствие предъявленным требованиям.

Руководствующими стандартами при написании технического задания являются ГОСТ 34.602.89 «Техническое задание на создание автоматизированной системы» [1] и ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению» [2]. Первый стандарт предназначен для разработчиков автоматизированных систем, второй для программных средств (разница между данными стандартами описана в статье «Что такое ГОСТ» [3]).

Список и описание разделов, которые должны содержаться в техническом задании согласно ГОСТу 19.201-78 Техническое задание. Требования к содержанию и оформлению:

1. Введение

2. Основания для разработки
3. Назначение разработки
4. Требования к программе или программному изделию
 - 4.1. Требования к функциональным характеристикам
 - 4.2. Требования к надежности
 - 4.3. Условия эксплуатации
 - 4.4. Требования к составу и параметрам технических средств
 - 4.5. Требования к информационной и программной совместимости
 - 4.6. Требования к маркировке и упаковке
 - 4.7. Требования к транспортированию и хранению
 - 4.8. Специальные требования
5. Требования к программной документации
6. Техничко-экономические показатели
7. Стадии и этапы разработки
8. Порядок контроля и приемки

Список и описание разделов, которые должны содержаться в техническом задании согласно ГОСТу 34.602.89 Техническое задание на создание автоматизированной системы:

1. Общие сведения
2. Назначение и цели создания системы
3. Характеристика объекта автоматизации
4. Требования к системе
 - 4.1. Требования к системе в целом
 - 4.1.1. Требования к структуре и функционированию системы
 - 4.1.2. Требования к функциям (задачам), выполняемым системой
 - 4.1.3. Показатели назначения
 - 4.1.4. Требования к надежности
 - 4.1.5. Требования к безопасности
 - 4.1.6. Требования к эргономике и технической эстетике
 - 4.1.7. Требования к транспортабельности для подвижных систем
 - 4.1.8. Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы
 - 4.1.9. Требования к защите информации от несанкционированного доступа
 - 4.1.10. Требования по сохранности информации при авариях
 - 4.1.11. Требования к защите от влияния внешних воздействий
 - 4.1.12. Требования к патентной чистоте
 - 4.1.13. Требования по стандартизации и унификации
 - 4.1.14. Дополнительные требования
 - 4.2. Требования к численности и квалификации персонала системы и режиму его работы
 - 4.3. Требования к видам обеспечения
5. Состав и содержание работ по созданию системы
6. Порядок контроля и приемки системы

7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

8. Требования к документированию

9. Источники разработки

Таким образом, документ Техническое задание должен, отражать все требования к проектируемому продукту, выделенные на этапе аналитического исследования объекта автоматизации.

Основными разделами технического задания для обоих ГОСТов являются:

- Общие сведения о системе (программе);
- Назначение, цели и задачи системы (программы);
- Требования к системе (функциональные требования, пользовательские требования, требования к системе в целом и тд);
- Требования к видам обеспечения;
- Требования к документированию;
- Стадии и этапы разработки;
- Порядок контроля и приемки системы (программы).

3.2 Общие сведения

Данный раздел должен содержать полное наименование системы и все варианты сокращений, которые будут использованы при разработке документации.

Пример:

«В данном документе создаваемая информационная система называется «Единое окно доступа к образовательным ресурсам», сокращенно ЕО. Систему Единое окно доступа к образовательным ресурсам далее в настоящем документе допускается именовать Единое окно или Система.»

Также сюда следует включить подразделы, сообщающие реквизиты организаций, участвующих в разработке (Заказчика и Исполнителя).

В подразделе «Основания для разработки» документа Техническое задание перечисляются основные документы, на основании которых выполняются данные работы. Например, для системы, выполняемой по заказу Правительства страны или другого Государственного органа, должны быть указаны законы, указы и постановления Правительства.

Далее следует указать сроки начала и окончания работ и сведения об источнике финансирования. Данная информация может быть указана и в конце технического задания в разделе с указанием стадий и этапов работ.

Неотъемлемой частью документа Техническое задание также должен быть список терминов и сокращений. Термины и сокращения лучше представить в виде таблицы с двумя столбцами «Термин» и «Полная форма».

Термины и сокращения располагаются в алфавитном порядке. В первую очередь принято давать расшифровку русскоязычным терминам и сокращениям, потом англоязычным.

3.3 Назначение и цели создания системы

Данный раздел должен содержать назначение и цели создания системы.

Пример:

«Информационная система «Единое окно доступа к образовательным ресурсам» предназначена для обеспечения пользователей полной, оперативной и удобной информацией, касающейся системы образования Российской Федерации, организаций выполняющих функцию образовательных учреждений.

Основной целью Системы является формирование единой информационной среды и автоматизации бизнес-процессов Образовательных учреждений Российской Федерации.

Создание информационной системы «Единое окно» должно обеспечить:

- предоставление пользователям широкого спектра информационных ресурсов;
- повышение уровня информационной безопасности;
- повышение эффективности работы образовательных учреждений и ведомств за счет оптимизации ряда бизнес-процессов;
- повышение эффективности процесса взаимодействия информационных систем и сервисов внутри ведомства.
- Создание Системы позволит сократить эксплуатационные затраты в результате повышения эффективности работы ведомства.»

3.4 Требования к системе

Данный раздел предназначен для описания основных функциональных требований системы. Это самая важная часть технического задания, так как именно она станет основным аргументом исполнителя при спорах с Заказчиком в процессе сдачи системы в эксплуатацию. Поэтому к его написанию необходимо подойти наиболее тщательно.

В документе Техническое задание должны быть представлены все требования, выявленные на этапе проведения анализа объекта автоматизации. Лучше всего выделить основные бизнес-процессы, которые и должны быть раскрыты посредством описания функциональных требований.

Пример:

«4.1 Бизнес-процесс «Предоставление информации об образовательных учреждениях Российской Федерации. В данном бизнес-процессе выделяются следующие участники:

Модератор – работник ведомства, входящий в состав обслуживающего персонала Системы, ответственный за корректность предоставляемых данных

Автор – сотрудник образовательного учреждения, ответственный за размещение информации о своей организации.

Пользователь – гражданин, нуждающийся в получении информации о работе образовательных учреждений Российской Федерации.

4.1.1 Регистрация образовательного учреждения в Системе

Регистрация образовательного учреждения Российской Федерации осуществляется ответственным сотрудником учреждения («Постановление Правительства ...»).

Процесс регистрации образовательного учреждения включает следующие шаги:

- Автор создает запись об организации;
- Автор заносит данные организации;
- Система проверяет наличие лицензии для данной организации
- Если лицензия существует в базе данных, Система отправляет Автору сообщение об успешной регистрации;
- Если лицензия не найдена в базе данных, Система отправляет сообщение Автору об отсутствии лицензии для данной организации.»

Информацию, приведенную в данном разделе, следует, более полно раскрыть в приложении к документу Техническое задание. В приложении к техническому заданию можно привести экранную форму и ниже описать все события, которые на ней присутствуют (создание, просмотр, редактирование, удаление и т.п.).

Требования к системе в целом включают раскрытие ее архитектуры с описанием всех подсистем. В данной части Технического задания следует описать требования к интеграции системы с другими продуктами (если таковые имеются). Далее в техническое задание должны быть включены:

- требования к режимам функционирования системы
- показатели назначения
- требования к надежности
- требования к безопасности
- требования к численности и квалификации персонала и режиму его работы
- требования к защите информации
- требования по сохранности информации при авариях
- требования к патентной чистоте
- требования по стандартизации и унификации
- и т.д.

3.5 Требованиям к видам обеспечения

В данном разделе должны быть представлены требования к математическому, информационному, лингвистическому, программному, техническому и др. видам обеспечения (если таковые имеются).

3.6 Требования к документированию

Раздел «Требования к документированию» включает перечень проектных и эксплуатационных документов, которые должны быть предоставлены заказчику.

Данный раздел технического задания также важен, как и описание функциональных требований, поэтому не следует ограничиваться фразой «Заказчику должна быть предоставлена вся документация согласно ГОСТ 34». Большинство документов из списка, указанного в ГОСТ 34.201-89 скорее всего не понадобятся ни исполнителю, ни заказчику, поэтому лучше сразу согласовать список на этапе разработки документа Техническое задание.

Минимальный пакет документов обычно включает:

- Техническое задание;
- Ведомость эскизного (технического) проекта;
- Пояснительная записка к Техническому проекту;
- Описание организации информационной базы;
- Руководство пользователя;
- Руководство администратора;
- Программа и методика испытаний;
- Протокол приемочных испытаний;
- Акт выполненных работ

Перечень документов в техническом задании лучше представить в виде таблицы, где указывается наименование документа и стандарт на основании, которого он должен быть разработан.

3.7 Стадии и этапы разработки

В данном разделе документа Техническое задание следует представить информацию обо всех этапах работ, которые должны быть проведены.

Описание этапа должно включать наименование, сроки, описание работ и конечный результат.

3.8 Порядок контроля и приемки системы

В данном разделе документа Техническое задание необходимо указать документ, на основании которого должны быть проведены приемо-сдаточные испытания.

При необходимости техническое задание может быть дополнено другими разделами, или сокращено путем удаления нецелесообразных пунктов.

При изменении структуры технического задания, во избежание конфликтных ситуаций, ее необходимо согласовать с заказчиком до разработки документа.

ЛАБОРАТОРНАЯ РАБОТА 18

Проектирование спецификации информационной системы индивидуальному заданию

Цель работы Для выбранного варианта информационной системы определить набор требований и спецификаций на создание информационной системы.

Содержание работы и методические указания к ее выполнению

Для успешной реализации базы данных объект проектирования должен быть прежде всего адекватно описан, должны быть построены полные и непротиворечивые функциональные модели базы данных. Опыт проектирования информационных систем показывает, что это логически сложная, трудоемкая и длительная по времени работа, требующая высокой квалификации участвующих в ней специалистов.

При проектировании информационной системы необходимо провести анализ целей этой системы и выявить требования к ней отдельных пользователей. Информация для построения модели информационной системы берется на основе проведения всестороннего обследования организации, для которой выполняется разработка информационной системы. Сбор данных начинается с изучения сущностей предметной области, процессов, использующих эти сущности, и связей между ними

Последовательность выполнения лабораторной работы:

1. Ознакомиться с предложенным вариантом описания предметной области. Проанализировать предметную область, уточнив и дополнив ее, руководствуясь собственным опытом, консультациями и другими источниками.

2. Выполнить структурное разбиение предметной области на отдельные подразделения (отделы, службы, подсистемы, группы и пр.) согласно выполняемым ими функциям.

3. Определить задачи и функции системы в целом и функции каждого подразделения (подсистемы).

4. Выполнить словесное описание работы каждого подразделения (подсистемы), алгоритмов и сценариев выполнения ими отдельных работ.

5. Оформить отчет со следующими разделами:

- исходное задание;
- расширенное описание предметной области с учетом сделанных дополнений;
- состав подразделений (подсистем) информационной системы;
- перечень функций и задач системы в целом и каждого подразделения (подсистемы) в отдельности, отношения их между собой;
- подробное описание работы каждого подразделения (подсистемы);
- описание отдельных сценариев работ подразделений (подсистем);
- входная и выходная информация для каждого подразделения (подсистемы);

Лабораторная работа № 19

ЛАБОРАТОРНАЯ РАБОТА №23

Разработка функциональной модели информационной системы

Цель работы: Целью данной лабораторной работы является построение функциональной модели информационной системы с помощью методологии DFD.

Краткие теоретические сведения

В основе данной методологии (методологии Gane/Sarson) лежит построение структурно-функциональной модели анализируемой ИС проектируемой или реально существующей. В соответствии с методологией модель системы определяется как иерархия диаграмм потоков данных (ДПД или DFD), описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю. Диаграммы верхних уровней иерархии (контекстные диаграммы) определяют основные процессы или подсистемы ИС с внешними входами и выходами. Они детализируются при помощи диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут такой уровень декомпозиции, на котором процесс становится элементарными и детализировать их далее невозможно.

Источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или процессам. Те в свою очередь преобразуют информацию и порождают новые потоки, которые переносят информацию к другим процессам или подсистемам, накопителям данных или внешним сущностям - потребителям информации.

Таким образом, основными компонентами диаграмм потоков данных являются:

- внешние сущности;
- системы/подсистемы;
- процессы;
- накопители данных;
- потоки данных.

2.1. Внешние сущности Внешняя сущность представляет собой материальный предмет, другую подсистему или пользователя системы, представляющих собой источник или приемник информации, например, заказчики, персонал, поставщики, клиенты, склад. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой ИС. В процессе анализа некоторые внешние сущности могут быть перенесены внутрь диаграммы анализируемой ИС, если это необходимо, или, наоборот, часть процессов ИС может быть вынесена за пределы диаграммы и представлена как внешняя сущность.

Внешняя сущность обозначается квадратом (рис.

2.1), расположенным как бы "над" диаграммой и бросающим на нее тень, для того, чтобы можно было выделить этот символ среди других обозначений:

Рис. 2.1

2.2. Системы и подсистемы При построении модели сложной ИС она может быть представлена в самом общем виде на так называемой контекстной диаграмме в виде одной системы как единого целого, либо может быть декомпозирована на ряд подсистем.

Подсистема (или система) на контекстной диаграмме изображается следующим образом (рис. 2.2).

Рис.2.2 Поле номера подсистемы служит для ее идентификации. В поле имени вводится наименование подсистемы в виде предложения с подлежащим и соответствующими определениями и дополнениями.

2.3. Процессы Процесс представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. Физически процесс может быть реализован различными способами: это могут быть сотрудники подразделения организации (отдела), выполняющие обработку входных документов и выпуск отчетов, программа, аппаратно реализованное логическое устройство и т.д.

Процесс на диаграмме потоков данных изображается, как показано на рисунке 2.3.

Рис. 2.3 Номер процесса служит для его идентификации.

В поле имени вводится наименование процесса в виде предложения с активным недвусмысленным глаголом в неопределенной форме (вычислить, рассчитать, проверить, определить, создать, получить), за которым следуют существительные в винительном падеже, например:

- "Ввести сведения о клиентах";
- "Выдать информацию о текущих расходах";
- "Проверить кредитоспособность клиента".

Использование таких глаголов, как "обработать", "модернизировать" или "отредактировать" означает, как правило, недостаточно глубокое понимание данного процесса и требует дальнейшего анализа.

Информация в поле физической реализации показывает, какое подразделение организации, программа или аппаратное устройство выполняет данный процесс.

2.4. Накопители данных Накопитель данных (или хранилище данных) представляет собой абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.

Накопитель данных может быть реализован физически в виде ящика в картотеке, таблицы в оперативной памяти, файла на магнитном носителе и т.д. Накопитель данных на диаграмме потоков данных изображается, как показано на рисунке 2.4.

Рис. 2.4 Накопитель данных идентифицируется буквой "D" и произвольным числом. Имя накопителя выбирается из соображения наибольшей информативности для проектировщика.

Накопитель данных в общем случае является прообразом будущей базы данных (или файла базы данных) и описание хранящихся в нем данных должно быть увязано с информационной моделью.

2.5. Потоки данных Поток данных определяет информацию, передаваемую через некоторое соединение от источника к приемнику. Реальный поток данных может быть представлен некоторыми документами, информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными лентами или дискетами, переносимыми с одного компьютера на другой и т.д.

Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока (рис. 2.5).

Каждый поток данных имеет имя, отражающее его содержание.

Рис. 2.5

2.6. Построение иерархии диаграмм потоков данных Первым шагом при построении иерархии ДПД является построение контекстных диаграмм. Обычно при

проектировании относительно простых ИС строится единственная контекстная диаграмма с сетевой топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.

Если же при описании сложной системы ограничиться единственной контекстной диаграммой потоков данных, то она будет содержать слишком большое количество источников и приемников информации, которые трудно расположить на листе бумаги нормального формата, и кроме того, единственный главный процесс не раскрывает структуры распределенной системы.

Для сложных ИС строится иерархия контекстных диаграмм.

При этом контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

Иерархия контекстных диаграмм определяет взаимодействие основных функциональных подсистем проектируемой ИС как между собой, так и с внешними входными и выходными потоками данных и внешними объектами (источниками и приемниками информации), с которыми взаимодействует ИС.

Разработка контекстных диаграмм решает проблему строгого определения функциональной структуры ИС на самой ранней стадии ее проектирования, что особенно важно для сложных многофункциональных систем, в разработке которых участвуют разные организации и коллективы разработчиков.

3. Методология DFD в пакете BPWin Работа с моделями DFD в BPWin почти аналогична построению функциональных моделей методологии IDEF0. Остановимся подробнее на имеющихся отличиях.

3.1. Панель инструментов Панель инструментов (рис. 2.6) изменилась, поскольку мы используем другую методологию.

— — —

Go to sibling diagram – переход к родственной диаграмме;

Go to parent diagram – переход к родительской диаграмме;

Create or go to child diagram – Создать/перейти к дочерней диаграмме.

3.2. Создание словаря сущностей В BPWin словарь информационных сущностей хранится в закладке Arrow Data, а словесное описание блока/стрелки – в закладке Definition. Словарь сущностей будет использоваться нами в лабораторной работе №3.

Для того чтобы перейти в режим редактирования сущностей, выберите стрелку, щелкните правой кнопкой мыши и в появившемся меню выберите пункт Arrow Data. Откроется окно свойств стрелки (рис.

2.7):

Рис. 2.7 В этом окне мы можем присвоить какой-либо стрелке-связи учитываемые нами сущности. Однако, поскольку пока ещё ни одной сущности не создано, необходимо их сначала создать. Для этого нажмите кнопку Ent/Att Editor. Откроется окно редактора сущностей (рис.

2.8):

Рис 2.8 В этом окне мы можем добавлять/изменять/удалять сущности и их атрибуты. Для этого существуют кнопки Add (Добавить), Update (Обновить) и Delete (Удалить) в верхней части экрана (редактирование сущностей), так и в нижней его части (редактирование).

3.3. Пример уровня детализации модели

В процессе работы создаваемые вами уровни будут принципиально схожи с представленным на рисунке 2.9:

— — —

3.4. Экспорт сущностей из BPWin Используя пакеты BPWin и ERWin, вы можете автоматизировать составление полной документации по модели вашей предметной области. Сущности, созданные вами в BPWin могут быть экспортированы в ERWin. Для этого, не закрывая созданную вами модель, выберите пункт меню FileExport-ERWin 4.0 (brx). Появится окно сохранения файла (рис.

2.10)

Рис 2.10 Сохраните файл с сущностями.

4. Задания на лабораторную работу

3.1. Изучить представление методологии DFD в пакете BPWin.

3.2. Построить структурно-функциональную модель выбранной предметной области.

3.3. Составить словарь сущностей и экспортировать его из BPWin.

5. Требования к отчету Отчет должен содержать титульный лист, цель работы, задания на лабораторную работу, диаграммы функциональной модели и ответы на контрольные вопросы

6. Контрольные вопросы

1. Что такое структурно-функциональная модель? Для чего она используется?

2. Назовите основные элементы структурно-функциональной модели.

3. Что такое словарь модели? Для чего он служит?

4. Как осуществляется экспорт из BPWin?