

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«Санкт-Петербургский государственный университет
промышленных технологий и дизайна»**
Высшая школа технологии и энергетики
Кафедра прикладной математики и информатики

ПРОЕКТИРОВАНИЕ И СОПРОВОЖДЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Конспект лекций для студентов всех форм обучения
по направлению подготовки
01.03.02 — Прикладная математика и информатика

Составитель
В. П. Яковлев

Санкт-Петербург
2025

Утверждено
на заседании кафедры ПМИ
20.05.2025 г., протокол № 9

Рецензенты:
Н. В. Орлова, В. И. Сидельников

Конспект лекций соответствует программе и учебному плану дисциплины «Проектирование и сопровождение информационных систем» для студентов, обучающихся по направлению подготовки 01.03.02 «Прикладная математика и информатика».

Текст лекций охватывает весь объем материалов по дисциплине. Издание предназначено для самостоятельной работы студентов всех форм обучения.

Утверждено Редакционно-издательским советом ВШТЭ СПбГУПТД
в качестве текста лекций

Режим доступа: http://publish.sutd.ru/tp_get_file.php?id=202016, по паролю.
- Загл. с экрана.

Дата подписания к использованию 27.08.2025 г. Рег. № 5091/24

Высшая школа технологии и энергетики СПбГУПТД
198095, СПб, ул. Ивана Черных, 4.

© ВШТЭ СПбГУПТД, 2025

ОГЛАВЛЕНИЕ

ЛЕКЦИЯ № 1. ОСНОВНЫЕ ПОНЯТИЯ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ	4
1. Классификация информационных систем	4
2. Методы проектирования информационных систем	8
3. Этапы создания информационных систем	15
ЛЕКЦИЯ № 2. МОДЕЛИ ЖИЗНЕННЫХ ЦИКЛОВ ИНФОРМАЦИОННЫХ СИСТЕМ	18
1. Понятие и модели жизненного цикла ИС	18
2. Стандарты, регламентирующие жизненные циклы ИС	28
ЛЕКЦИЯ № 3. ОРГАНИЗАЦИЯ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ	29
1. Средства проектирования ИС	29
2. Каноническое проектирование	35
3. Типовое проектирование	44
ЛЕКЦИЯ № 4. АНАЛИЗ И МОДЕЛИРОВАНИЕ ФУНКЦИОНАЛЬНОЙ ОБЛАСТИ ВНЕДРЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ	47
1. Полная бизнес-модель компании	47
2. Шаблоны организационного бизнес-моделирования	52
3. Построение организационно-функциональной модели компании	58
ЛЕКЦИЯ № 5. ДИНАМИЧЕСКОЕ ОПИСАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ	62
1. Процессные потоковые модели	62
2. Основные элементы процессного подхода	65
3. Выделение и классификация процессов	67
ЛЕКЦИЯ № 6. МЕТОДОЛОГИЯ МОДЕЛИРОВАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ	74
1. Структурная модель предметной области	74
2. Функционально-ориентированные и объектно-ориентированные методологии описания предметной области	80
3. Синтетическая методика	87
ЛЕКЦИЯ № 7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ	89
1. Внемашинное информационное обеспечение	90
2. Внутримашинное информационное обеспечение	100
ЛЕКЦИЯ № 8. СОПРОВОЖДЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ	105
1. Понятие сопровождения информационных систем	105
2. Варианты сопровождения информационных систем	106
3. Задачи сопровождения информационных систем	107
4. Сценарий и структура ИТ-сопровождения	110
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	114

ЛЕКЦИЯ № 1. ОСНОВНЫЕ ПОНЯТИЯ ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

- 1. Классификация информационных систем.**
- 2. Методы проектирования информационных систем.**
- 3. Этапы создания информационных систем.**

1. Классификация информационных систем

Разнообразие задач, решаемых с помощью информационных систем (ИС), привело к появлению множества разнотипных систем, отличающихся принципами построения и заложенными в них правилами обработки информации.

Информационные системы можно классифицировать по целому ряду различных признаков. В основу рассматриваемой классификации положены наиболее существенные признаки, определяющие функциональные возможности и особенности построения современных информационных систем. В зависимости от *объема решаемых задач, используемых технических средств, организации функционирования* информационные системы делятся на ряд групп (классов).

По типу хранимых данных ИС делятся на фактографические и документальные.

Фактографические системы предназначены для хранения и обработки структурированных данных в виде чисел и текстов. Над такими данными можно выполнять различные операции.

В документальных системах информация представлена в виде документов, состоящих из названий, описаний, рефератов и текстов. Поиск по неструктурированным данным осуществляется с использованием семантических признаков. Отобранные документы предоставляются пользователю, а обработка данных в таких системах практически не производится.

Основываясь на *степени автоматизации информационных процессов* в системе управления фирмой, информационные системы делятся на *ручные, автоматические и автоматизированные*.

Ручные информационные системы характеризуются отсутствием современных технических средств переработки информации и выполнением всех операций человеком.

В автоматических информационных системах все операции по переработке информации выполняются без участия человека.

Автоматизированные информационные системы предполагают участие в процессе обработки информации и человека, и технических средств. При этом главная роль в выполнении рутинных операций обработки данных отводится компьютеру. Именно этот класс систем соответствует современному представлению понятия «информационная система».

В зависимости от *характера обработки данных* ИС делятся на информационно-поисковые и информационно-решающие.

Информационно-поисковые системы производят ввод, систематизацию, хранение, выдачу информации по запросу пользователя без сложных преобразований данных. Например, ИС библиотечного обслуживания, резервирования и продажи билетов на транспорте, бронирования мест в гостиницах и прочее.

Информационно-решающие системы осуществляют, кроме того, операции переработки информации по определенному алгоритму. *По характеру использования выходной информации* такие системы принято делить на *управляющие и советующие*.

Результирующая информация **управляющих ИС** непосредственно трансформируется в принимаемые человеком решения. Для этих систем характерны задачи расчетного характера и обработка больших объемов данных. Например, ИС планирования производства или заказов, бухгалтерского учета.

Советующие ИС вырабатывают информацию, которая принимается человеком к сведению и учитывается при формировании управленческих решений, а не инициирует конкретные действия. Эти системы имитируют интеллектуальные процессы обработки знаний, а не данных. Например, экспертные системы.

В зависимости от *сферы применения* различают следующие классы информационных систем.

Информационные системы организационного управления предназначены для автоматизации функций управленческого персонала как промышленных предприятий, так и непромышленных объектов (гостиниц, банков, магазинов и прочее).

Основными функциями подобных систем являются: оперативный контроль и регулирование, оперативный учет и анализ, перспективное и оперативное планирование, бухгалтерский учет, управление сбытом, снабжением и другие экономические и организационные задачи.

Информационные системы управления технологическими процессами служат для автоматизации функций производственного персонала по контролю и управлению производственными операциями. В таких системах обычно предусматривается наличие развитых средств измерения параметров технологических процессов (температуры, давления, химического состава и т. п.), процедур контроля допустимости значений параметров и регулирования технологических процессов.

ИС автоматизированного проектирования (САПР) предназначены для автоматизации функций инженеров-проектировщиков, конструкторов, архитекторов, дизайнеров при создании новой техники или технологии. Основными функциями подобных систем являются: инженерные расчеты, создание графической документации (чертежей, схем, планов), создание проектной документации, моделирование проектируемых объектов.

Интегрированные (корпоративные) ИС используются для автоматизации всех функций фирмы и охватывают весь цикл работ от планирования деятельности до сбыта продукции. Они включают в себя ряд модулей (подсистем), работающих в едином информационном пространстве и выполняющих функции поддержки соответствующих направлений деятельности. Типовые задачи, решаемые модулями корпоративной системы, приведены в таблице 1.

Таблица 1 – Функциональное назначение модулей корпоративной ИС

Подсистема маркетинга	Производственные подсистемы	Финансовые и учетные подсистемы	Подсистема кадров (человеческих ресурсов)	Прочие подсистемы, (например, ИС руководства)
Исследование рынка и прогнозирование продаж	Планирование объемов работ и разработка календарных планов	Управление портфелем заказов	Анализ и прогнозирование потребности в трудовых ресурсах	Контроль за деятельностью фирмы
Управление продажами	Оперативный контроль и управление производством	Управление кредитной политикой	Ведение архивов записей о персонале	Выявление оперативных проблем
Рекомендации по производству новой продукции	Анализ работы оборудования	Разработка финансового плана	Анализ и планирование подготовки кадров	Анализ управленческих и стратегических ситуаций
Анализ и установление цены	Участие в формировании заказов поставщикам	Финансовый анализ и прогнозирование		Обеспечение процесса выработки стратегических решений
Учет заказов	Управление запасами	Контроль бюджета, бухгалтерский учет и расчет зарплаты		

Анализ современного состояния рынка ИС показывает устойчивую тенденцию роста спроса на информационные системы организационного управления. Причем спрос продолжает расти именно на интегрированные системы управления. Автоматизация отдельной функции, например, бухгалтерского учета или сбыта готовой продукции, является уже пройденным этапом для многих предприятий.

Некоторые популярные в 2024 году программные продукты для реализации ИС организационного управления различных классов:

- **Для управления проектами:** Jira, YouGile, Wrike, Asana, ClickUp, Comindware Project, «Планиро».
- **Для автоматизации управленческого учёта и МСФО:** «Планум», 1С, Optimacros.
- **Для управления бизнес-процессами (BPM-системы):** Directum RX, «Первая форма», Elma365, «Битрикс24», Visary BPM, Pyrus, Naumen BPM, Amber BPM, Neaktor BPM2B.

Выбор программного продукта зависит от конкретных потребностей и задач организации.

Существует также классификация ИС в зависимости от уровня управления, на котором система используется.

Информационная система оперативного уровня поддерживает исполнителей, обрабатывая данные о сделках и событиях (счета, накладные, зарплаты, кредиты, поток сырья и материалов). Информационная система оперативного уровня является связующим звеном между фирмой и внешней средой.

Задачи, цели, источники информации и алгоритмы обработки на оперативном уровне заранее определены и в высокой степени структурированы.

Информационные системы специалистов поддерживают работу с данными и знаниями, повышают продуктивность и производительность работы инженеров и проектировщиков. Задачей подобных информационных систем является интеграция новых сведений в организацию и помочь в обработке бумажных документов.

Информационные системы уровня менеджмента используются работниками среднего управленческого звена для мониторинга, контроля, принятия решений и администрирования. Основными функциями этих систем являются:

- сравнение текущих показателей с прошлыми;
- составление периодических отчетов за определенное время, а не выдача отчетов по текущим событиям, как на оперативном уровне;
- обеспечение доступа к архивной информации и т. д.

Стратегическая информационная система – компьютерная информационная система, обеспечивающая поддержку принятия решений по реализации стратегических перспективных целей развития организации.

Информационные системы стратегического уровня помогают высшему звену управленцев решать неструктурированные задачи, осуществлять долгосрочное планирование. *Основная задача таких систем – сравнение происходящих во внешнем окружении изменений с существующим потенциалом фирмы.* Они призваны создать общую среду компьютерной телекоммуникационной поддержки решений в неожиданно возникающих ситуациях. Используя самые совершенные программы, эти системы способны в любой момент времени предоставить информацию из многих источников. Некоторые стратегические системы обладают ограниченными аналитическими возможностями.

Типовые структуры ИС

С точки зрения программно-аппаратной реализации можно выделить ряд типовых структур ИС.

Традиционные архитектурные решения основаны на *использовании выделенных файл-серверов или серверов баз данных*.

Существуют также варианты архитектур корпоративных информационных систем, *базирующихся на технологии Internet (Intranet-приложения)*.

Следующая разновидность архитектуры информационной системы основывается на концепции «хранилища данных» (DataWarehouse) – *интегрированной информационной среды, включающей разнородные информационные ресурсы*.

И, наконец, для построения глобальных распределенных информационных приложений используется *архитектура интеграции информационно-вычислительных компонентов на основе объектно-ориентированного подхода*.

2. Методы проектирования информационных систем

Общая методология проектирования информационных систем как раздел информатики в терминологии создания ИС разбивается на ряд отдельных **методологий** (или концепций), предлагающих свои методы и средства проектирования ИС. Такими методологиями являются методология SADT, ERD, DFD и прочие, о которых мы будем говорить ниже.

В реальных условиях *проектирование – это поиск способа, который удовлетворяет требованиям функциональности системы средствами имеющихся технологий с учетом заданных ограничений*.

В ходе проектирования ИС используют следующие понятия:

Системный подход: любая система представляет собой совокупность взаимосвязанных элементов, функционирующих совместно для достижения общей цели.

Метод проектирования: организованная совокупность процессов создания ряда моделей, которые описывают различные аспекты создаваемой системы с использованием четко определенной нотации.

Технология проектирования: совокупность технологических операций в их последовательности и взаимосвязи, приводящая к разработке проекта системы.

Помимо названных методов проектирования ИС существует **классификация методов проектирования** по ряду признаков.

По степени использования типовых проектных решений методы проектирования ИС подразделяются на:

– оригинальные (индивидуальные), предполагающие разработку систем без использования готовых проектных решений;

– типовые, основывающиеся на использовании типовых проектных решений с последующей их адаптацией к особенностям конкретной предметной области.

По характеру адаптации проектных решений выделяют следующие методы проектирования ИС:

– методы перепрограммирования, подразумевающие разработку изменяемых программных модулей заново;

– методы параметризации, настраивающие проектные решения путём изменения отдельных параметров в программных модулях;

– модельные методы, основанные на внесении изменений в модель предметной области с последующей генерацией программного кода изменяемого модуля.

По степени автоматизации методы проектирования делятся на:

– методы с универсальной компьютерной поддержкой, использующие универсальные языки программирования, СУБД, табличные процессоры;

– методы со специальной компьютерной поддержкой, использующие специальные программные средства, позволяющие осуществлять генерацию программного кода на основе созданной модели предметной области.

Метод проектирования «снизу-вверх»

На первом этапе развития технологий проектирования основным подходом в проектировании ИС был **метод «снизу-вверх»**, при использовании которого система создается как набор приложений, наиболее важных в данный момент для поддержки деятельности предприятия (организации).

Метод проектирования ИС «**снизу-вверх**» (восходящий метод) основан на том, чтобы создавать информационную систему организации путём автоматизации отдельных рабочих мест на общих принятых принципах: на одном языке программирования, с использованием одних программных средств, по единой концепции создания информационных связей. Недостатком такого подхода является то, что фактически отсутствует проектирование верхнего уровня управления в системе, что при попытке соединения ПО отдельных автоматизированных рабочих мест в единую систему может привести к пересечению, дублированию либо отсутствию некоторых информационных связей.

Основной целью этих проектов является не создание тиражируемых продуктов, а обслуживание текущих потребностей конкретного учреждения. Такой подход отчасти сохраняется и в настоящее время. В рамках «лоскутной автоматизации» достаточно хорошо обеспечивается поддержка отдельных функций, но практически полностью отсутствует стратегия развития комплексной системы автоматизации, а объединение функциональных подсистем превращается в самостоятельную и достаточно сложную задачу.

Создавая свои отделы и управления автоматизации, предприятия пытались «обустроиться» своими силами. Однако периодические изменения технологий работы и должностных инструкций, сложности, связанные с разными представлениями пользователей об одних и тех же данных, приводили к

непрерывным доработкам программных продуктов для удовлетворения все новых и новых пожеланий отдельных работников. Как следствие – и работа программистов, и создаваемые ИС вызывали недовольство руководителей и пользователей системы.

Метод проектирования «сверху-вниз»

Метод «сверху вниз» (нисходящий метод) имеет в основе идею «одной программы для всех», то есть на каждом рабочем месте устанавливается одинаковое программное обеспечение (ПО), и каждый пользователь в соответствии со своими функциональными обязанностями использует те или иные функции ИС. Но при таком подходе очевидное преимущество единого согласованного управления и охвата всех сфер деятельности предметной области часто сокращается за счёт невозможности учесть при проектировании все особенности отдельных функциональных подсистем, что приводит к несоответствию информационной модели реальному объекту моделирования.

Метод связан с осознанием того факта, что существует потребность в достаточно стандартных программных средствах автоматизации деятельности различных учреждений и предприятий. Из всего спектра проблем разработчики выделили наиболее заметные: *автоматизацию ведения бухгалтерского аналитического учета и технологических процессов*.

Сама идея использования универсальной программы накладывает существенные ограничения на возможности разработчиков по формированию структуры базы данных, экранных форм, по выбору алгоритмов расчета. Заложенные «сверху» жесткие рамки не дают возможности гибко адаптировать систему к специфике деятельности конкретного предприятия:

- учесть необходимую глубину аналитического и производственно-технологического учета;
- включить необходимые процедуры обработки данных;
- обеспечить интерфейс каждого рабочего места с учетом функций и технологии работы конкретного пользователя.

Решение этих задач требует серьезных доработок системы. Таким образом, материальные и временные затраты на внедрение системы и ее доводку под требования заказчика значительно превышают запланированные показатели.

В то же время заказчики ИС стали выдвигать все больше требований, направленных на обеспечение возможности комплексного использования корпоративных данных в управлении и планировании своей деятельности.

Таким образом, возникла необходимость формирования новой методологии построения информационных систем.

Метод многокомпонентности

Многокомпонентный метод проектирования ИС подразумевает автоматизацию общих функций управления в системе на верхнем уровне, а также унификацию обеспечивающих подсистем, то есть создание единой информационной среды для функционирования всех модулей ИС, а функции отдельных структурных единиц автоматизируются с помощью специализированного ПО для конкретных рабочих мест. При этом такое

специализированное ПО совместимо с общей информационной системой и обладает всеми возможностями экспорта и импорта данных. Очевидным преимуществом метода является то, что, не пытаясь сразу решить все задачи отдельных функциональных структур, проектирование ИС многокомпонентным методом позволяет быстро создать основу информационной системы и оставляет пространство для доработки и совершенствования её функциональных возможностей.

Этот метод предусматривает адаптацию подсистемы ПО к принятым в организации условиям работы. Проведение модернизации одного из компонентов не затрагивает центральную часть (ядро) и другие ее компоненты, что значительно повышает надежность, продолжительность жизни автоматизированной системы и обеспечивает наиболее полное выполнение требуемых функций.

Другая вариация многокомпонентного метода предусматривает создание «ядра» информационной системы, включающего помимо общего ПО, необходимого для её функционирования (управляющей и обеспечивающих подсистем), наиболее простые стандартные функциональные подсистемы. Продвинутые функции автоматизируются в виде «надстроек» к информационной системе. Такой способ сохраняет возможность постепенного расширения комплекса функций ИС.

Именно многокомпонентный метод является наиболее признанным и широко используемым в настоящее время при создании многофункциональных ИС «с нуля». В его основе лежит широко используемый при проектировании ИС структурный, заключающийся в декомпозиции (разбиении) информационной системы на отдельные автоматизируемые функции, соответствующие функциональным подсистемам, и так далее на всё более низких уровнях вплоть до отдельных информационных процедур.

Цель **метода многокомпонентности** заключается в *регламентации процесса проектирования ИС и обеспечении управления этим процессом* с тем, чтобы гарантировать выполнение требований как к самой ИС, так и к характеристикам процесса разработки.

В проектировании ИС выделяют два подхода: **функционально-ориентированный и объектно-ориентированный**. Они различаются тем, что именно принимается в качестве основного элемента информационной системы: функция, выполняемая системой, или объект в составе системы, например, структурное подразделение.

Функционально-ориентированный (структурный) подход к проектированию ИС основан на структурном анализе информационных систем, осуществляющем декомпозицию системы на некоторое множество иерархически подчинённых функций, выполняемых ею.

При функционально-ориентированном подходе формируются **модель требований** (логическая модель), описывающая, что должна делать проектируемая система (её функции), без указания, как это достигается, и **модель**

реализации (физическая модель), демонстрирующая, как система будет удовлетворять предъявленным к ней требованиям.

Главным преимуществом функционально-ориентированного подхода является то, что он позволяет рассматривать систему без учёта её структуры как инструмент для решения заданного круга задач. Это упрощает моделирование системы и дальнейшее её проектирование, особенно на первых этапах создания новой системы, когда её структура ещё не определена. Кроме того, модульный принцип наиболее широко используемого многокомпонентного метода проектирования основан именно на функциональной декомпозиции системы.

Основным недостатком функционально-ориентированного подхода является то, что функции системы реализуются через процессы обработки данных, но при этом сама структура данных не рассматривается при проектировании, то есть учитывается лишь один срез системы. В то же время данные и их структура обладают более постоянными характеристиками по сравнению с процессами их обработки.

Объектно-ориентированный подход использует объектную декомпозицию, где система описывается в терминах объектов и связей между ними, а поведение системы – как обмен информацией между объектами. Основными понятиями объектно-ориентированного подхода являются объект и класс.

Объект – это предмет или явление, которые имеют чётко определяемое поведение, состояние и индивидуальность.

Класс – это множество объектов, связанных общностью структуры и поведения. Объект является представителем класса объектов.

Объектно-ориентированный подход позволяет решить проблему разделения процессов и данных, так как объект содержит в себе и данные, и операции, выполняемые над ними (так называемые методы объекта). А так как даже при изменении требований к функциям системы структура данных обычно остается неизменной, объектная модель ИС более устойчива.

Другими преимуществами объектно-ориентированного подхода являются:

- сокращение объёма программного кода ПО ИС за счёт того, что объекты одного класса легче описать, в то время как функции не поддаются типизации и дублированию;
- облегчается процесс эволюционного расширения системы путём добавления новых объектов и связывания их с уже существующими;
- объектная модель легче воспринимается человеком, так как содержит описания реальных объектов, а не логических функций.

Однако объектно-ориентированный подход пока поддерживается небольшим числом программных средств проектирования (CASE-средств) и по сравнению со структурным подходом недостаточно обеспечен средствами визуализации (унифицированный язык моделирования UML, используемый для графического описания объектных моделей, ещё развивается, поэтому используется нечасто).

Соответственно существованию двух подходов к проектированию ИС – функционально-ориентированного и объектно-ориентированного, – методы анализа и проектирования информационных систем подразделяются на методы *функционально-ориентированного* (функционального) проектирования и методы *объектно-ориентированного* проектирования.

Функционально-ориентированные методы представляют систему как набор функций, преобразующих поступающие потоки информации в выходной поток, и чаще используются, когда структура системы ещё не полностью определена и будет изменяться в зависимости от функций системы. *Объектно-ориентированные методы* лучше соответствуют уже существующим системам, когда отдельные объекты выделены и имеют свои функции.

Наиболее известным является *семейство методик моделирования и проектирования IDEF*, разработанных на основе международных стандартов, включающее методики моделирования различных срезов систем.

Объектно-ориентированный подход использует единую методологию моделирования систем – *унифицированный язык моделирования UML* (Unified Modeling Language) – язык для определения, представления, проектирования и документирования систем. UML содержит стандартный набор диаграмм и нотаций к ним:

- 1) диаграммы вариантов использования системы – для моделирования требований к системе на основе выполняемых ею функций;
- 2) диаграммы классов – для моделирования статической структуры классов объектов и связей между ними;
- 3) диаграммы поведения системы – для определения правил запуска различных процессов в системе;
- 4) диаграммы взаимодействия – для моделирования процесса обмена сообщениями между объектами, включающие:
 - a) диаграммы последовательности;
 - b) кооперативные диаграммы;
- 5) диаграммы состояний – для моделирования поведения объектов системы при переходе из одного состояния в другое;
- 6) диаграммы деятельности – для моделирования поведения системы в рамках различных вариантов использования или моделирования деятельности;
- 7) диаграммы реализации, включающие:
 - a) диаграммы компонентов – для моделирования иерархии компонентов системы;
 - b) диаграммы размещения – для моделирования физической архитектуры системы.

Наличие единого языка моделирования для всех типов диаграмм системы и, следовательно, единой базы для создания программных средств поддержки моделирования обеспечивают широкие перспективы развития и применения объектно-ориентированного подхода к проектированию ИС.

Выбор того или иного подхода к проектированию ИС в дальнейшем определяет круг используемых программных средств проектирования ИС.

Функционально-ориентированный подход преимущественно используется при проектировании ИС, охватывающих ряд бизнес-процессов, имеющих место в одной функциональной области. Например, по своей структуре функционально-ориентированными обычно являются ИС статистического учёта, налогового учёта, бухгалтерского учёта и аудита, банковской деятельности, страховой деятельности, системы комплексной автоматизации предприятий (типа ERP) и множество других.

Объектно-ориентированный подход шире используется в ИС, основу которых составляют множества объектов, например, записей и таблиц в базах данных. Это информационные системы государственного и муниципального управления реестрового типа (системы регистрации населения, учёта объектов недвижимости), библиотечные системы, информационно-справочные системы.

Методы проектирования ИС



Рисунок 1.1 – Методы проектирования ИС

Проектирование ИС заключается в выполнении работ в трех областях:

- проектирование объектов данных, которые будут реализованы в базе данных;
- проектирование программ, экранных форм, отчетов, которые будут обеспечивать выполнение запросов к данным;
- учет конкретной среды или технологии, а именно: топологии сети, конфигурации аппаратных средств, используемой архитектуры (файл-сервер или клиент-сервер), параллельной обработки, распределенной обработки данных и т. п.

Проектирование информационной системы всегда начинается с определения цели проекта. В общем виде *цель проекта можно определить, как решение ряда взаимосвязанных задач*, включающих в себя обеспечение на момент запуска системы и в течение всего времени ее эксплуатации следующих факторов:

- требуемой функциональности системы и уровня ее адаптивности к изменяющимся условиям функционирования;
- требуемой пропускной способности системы;

- требуемого времени реакции системы на запрос;
- безотказной работы системы;
- необходимого уровня безопасности;
- простоты эксплуатации и поддержки системы.

*Согласно современной методологии, процесс создания ИС представляет собой **процесс построения и последовательного преобразования ряда согласованных моделей** на всех этапах жизненного цикла (ЖЦ) ИС.* На каждом этапе ЖЦ создаются специфичные для него модели – организации, требований к ИС, проекта ИС, требований к приложениям и т. д.

Модели формируются рабочими группами команды проекта, сохраняются и накапливаются в репозитарии проекта. Создание моделей, их контроль, преобразование и представление в коллективное пользование осуществляется с использованием специальных программных инструментов – CASE-средств.

3. Этапы создания информационных систем

Процесс создания ИС делится на ряд этапов, ограниченных некоторыми временными рамками и заканчивающихся выпуском конкретного продукта (моделей, программных продуктов, документации и прочее).

Обычно выделяют следующие *этапы создания информационной системы: формирование требований к системе, проектирование, реализация, тестирование, ввод в действие, эксплуатация и сопровождение.*

Формирование требований к системе

Начальным этапом процесса создания ИС является моделирование бизнес-процессов, протекающих в организации и реализующих ее цели и задачи. Модель организации, описанная в терминах бизнес-процессов и бизнес-функций, позволяет сформулировать основные требования к ИС. Это фундаментальное положение методологии обеспечивает объективность в выработке требований к проектированию системы.

Множество моделей описания требований к ИС затем преобразуется в систему моделей, описывающих концептуальный проект ИС. Формируются модели архитектуры ИС, требований к программному обеспечению (ПО) и информационному обеспечению (ИО).

Затем формируется архитектура ПО и ИО, выделяются корпоративные БД и отдельные приложения, формируются модели требований к приложениям и проводится их разработка, тестирование и интеграция.

Целью начальных этапов создания ИС, выполняемых на стадии анализа деятельности организации, является формирование требований к ИС, корректно и точно отражающих цели и задачи организации-заказчика. Для того чтобы специфицировать процесс создания ИС, отвечающей потребностям организации, нужно выяснить и четко сформулировать, в чем заключаются эти потребности. Для этого необходимо определить требования заказчиков к разработке ИС и отобразить их на языке моделей в требования к разработке проекта ИС так, чтобы обеспечить соответствие целям и задачам организации.

Задача формирования требований к ИС является одной из наиболее ответственных, трудно формализуемых, наиболее дорогих и тяжелых для исправления в случае ошибки. Современные инструментальные средства и программные продукты позволяют достаточно быстро создавать ИС по готовым требованиям. Но зачастую эти системы не удовлетворяют заказчиков, требуют многочисленных доработок, что приводит к резкому удорожанию фактической стоимости ИС. Основной причиной такого положения является неправильное, неточное или неполное определение требований к ИС на этапе анализа.

Проектирование

На этапе проектирования прежде всего формируются модели данных. Проектировщики в качестве исходной информации получают результаты анализа. Построение логической и физической моделей данных является основной частью проектирования базы данных. Полученная в процессе анализа информационная модель сначала преобразуется в логическую, а затем в физическую модель данных.

Параллельно с проектированием схемы базы данных выполняется проектирование процессов, чтобы получить спецификации (описания) всех модулей ИС. Оба эти процесса проектирования тесно связаны, поскольку часть бизнес-логики обычно реализуется в базе данных (ограничения, триггеры, хранимые процедуры).

Главная цель проектирования процессов заключается в отображении функций, полученных на этапе анализа, в модули информационной системы.

При проектировании модулей определяют интерфейсы программ: разметку меню, вид окон, горячие клавиши и связанные с ними вызовы.

Конечными продуктами этапа проектирования являются:

- схема базы данных (на основании ER-модели, разработанной на этапе анализа);
- набор спецификаций модулей системы (они строятся на базе моделей функций).

Кроме того, на этапе проектирования осуществляется также разработка архитектуры ИС, включающая в себя выбор платформы и операционной системы.

В неоднородной ИС могут работать несколько компьютеров на разных аппаратных платформах и под управлением различных операционных систем. Кроме выбора платформы, на этапе проектирования определяются следующие характеристики архитектуры:

- будет ли это архитектура «файл-сервер» или «клиент-сервер»;
- будет ли это трехуровневая архитектура со следующими слоями: сервер, ПО промежуточного слоя (сервер приложений), клиентское ПО;
- будет ли база данных централизованной или распределенной. Если база данных будет распределенной, то какие механизмы поддержки согласованности и актуальности данных будут использоваться;
- будет ли база данных однородной, то есть, будут ли все серверы баз данных продуктами одного и того же производителя (например, все

- серверы только Oracle или все серверы только DB2 UDB). Если база данных не будет однородной, то какое ПО будет использовано для обмена данными между СУБД разных производителей (уже существующее или разработанное специально как часть проекта);
- будут ли для достижения данной производительности использоваться параллельные серверы баз данных (например, Oracle Parallel Server, DB2 UDB и т. п.).

Этап проектирования завершается разработкой технического проекта информационной системы.

Реализация

На этапе реализации осуществляется создание программного обеспечения системы, установка технических средств, разработка эксплуатационной документации.

Этап тестирования обычно оказывается распределенным во времени.

После завершения разработки отдельного модуля системы выполняют ***автономный тест***, который *преследует две основные цели*:

- обнаружение отказов модуля (жестких сбоев);
- соответствие модуля спецификации (наличие всех необходимых функций, отсутствие лишних функций).

После того как автономный тест успешно пройдет, модуль включается в состав разработанной части, и группа сгенерированных модулей проходит ***тесты связей***, которые должны отследить их взаимное влияние.

Далее группа модулей *тестируется на надежность работы*, то есть проходят, во-первых, тесты имитации отказов системы, а во-вторых, тесты наработка на отказ.

Первая группа тестов показывает, насколько хорошо система восстанавливается после сбоев программного обеспечения, отказов аппаратного обеспечения. Вторая группа тестов определяет степень устойчивости системы при штатной работе и позволяет оценить время безотказной работы системы. В комплект тестов устойчивости должны входить тесты, имитирующие пиковую нагрузку на систему.

Затем весь комплект модулей проходит ***системный тест*** – тест внутренней приемки продукта, показывающий уровень его качества. Сюда входят тесты функциональности и тесты надежности системы.

Последний тест информационной системы – приемо-сдаточные испытания. Такой тест предусматривает показ информационной системы заказчику и должен содержать группу тестов, моделирующих реальные бизнес-процессы, чтобы показать соответствие реализации требованиям заказчика.

Оформление документации. На основе технического задания (и эскизного проекта) разрабатывается технический проект ИС. ***Технический проект системы*** – это техническая документация, содержащая общесистемные проектные решения, алгоритмы решения задач, а также оценку экономической эффективности автоматизированной системы управления и перечень мероприятий по подготовке объекта к внедрению. На этом этапе

осуществляются комплекс научно-исследовательских и экспериментальных работ для выбора основных проектных решений и расчет экономической эффективности системы.

Ввод в действие. Эксплуатация. Сопровождение

Эти этапы идут параллельно. Во время ввода в действие возможен процесс обучения работе с системой. Возможны доработка продукта, исправление ошибок и устранение неточностей.

Необходимость контролировать процесс создания ИС, гарантировать достижение целей разработки и соблюдение различных ограничений (бюджетных, временных и пр.) привело к широкому использованию в этой сфере методов и средств программной инженерии: структурного анализа, объектно-ориентированного моделирования, CASE-систем.

ЛЕКЦИЯ № 2. МОДЕЛИ ЖИЗНЕННЫХ ЦИКЛОВ ИНФОРМАЦИОННЫХ СИСТЕМ

- 1. Понятие и модели жизненного цикла ИС.**
- 2. Стандарты, регламентирующие жизненные циклы ИС.**

1. Понятие и модели жизненного цикла ИС

Жизненный цикл информационной системы можно представить как ряд событий, происходящих с системой в процессе ее создания и использования. Модель жизненного цикла отражает различные состояния системы, начиная с момента возникновения необходимости в данной ИС и заканчивая моментом ее полного выхода из употребления.

Жизненный цикл традиционно разделяют на следующие основные этапы.

1. Анализ требований.
2. Проектирование.
3. Кодирование (программирование).
4. Тестирование и отладка.
5. Эксплуатация и сопровождение.

Модель жизненного цикла – это структура, содержащая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного продукта в течение всей жизни системы, от определения требований до завершения ее использования. В настоящее время известны и используются следующие модели жизненного цикла:

- 1) каскадная модель;
- 2) спиральная модель.

Каскадная модель – модель процесса разработки программного обеспечения, жизненный цикл которой выглядит как поток, последовательно проходящий фазы анализа требований, проектирования, реализации, тестирования, интеграции и поддержки.

Она предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе.

На первом этапе проводится исследование проблемы, которая должна быть решена, четко формулируются все требования заказчика. Результатом, получаемым на данном этапе, является техническое задание (разработка требований), согласованное со всеми заинтересованными сторонами.

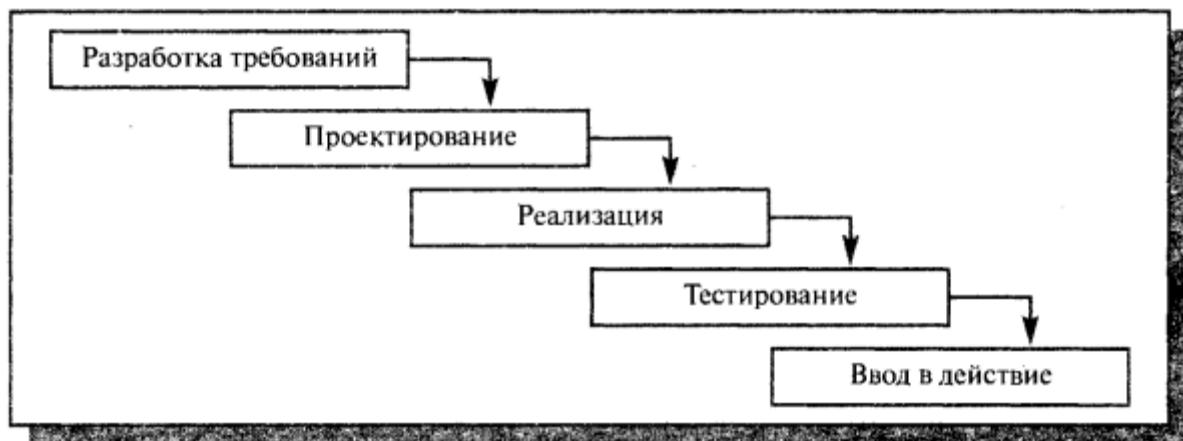


Рисунок 2.1 – Каскадная модель ЖЦ ИС

На втором этапе разрабатывают проектные решения, удовлетворяющие всем требованиями, сформулированным в техническом задании. Результатом данного этапа является комплект проектной документации, содержащей все необходимые данные для реализации проекта.

Третий этап – реализация проекта. Здесь осуществляется разработка программного обеспечения (кодирование) в соответствии с проектными решениями, полученными на предыдущем этапе. Методы, используемые для реализации, не имеют принципиального значения. Результатом выполнения данного этапа является готовый программный продукт.

На четвертом этапе проводится проверка (тестирование) полученного программного обеспечения на предмет соответствия требованиям, заявленным в техническом задании. Опытная эксплуатация позволяет выявить и исправить различного рода скрытые недостатки, проявляющиеся в реальных условиях работы информационной системы. Результат – готовая к эксплуатации система.

Последний этап – сдача готового проекта, ввод его в действие. Главная задача этого этапа – документально подтвердить, что все требования заказчика выполнены в полной мере. Результат – работающая ИС с полным комплектом сопроводительной документации, утвержденные заказчиком документы о принятии системы в эксплуатацию.

Достоинства каскадной модели:

- 1) стабильность требований в течение всего жизненного цикла разработки;
- 2) на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
- 3) выполняемые в логической последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.

Каскадный подход хорошо зарекомендовал себя при построении относительно простых ИС, когда в самом начале разработки можно достаточно точно и полно сформулировать все требования к системе.

Недостатки каскадной модели. Перечень недостатков каскадной модели при ее использовании для разработки информационных систем достаточно обширен:

- существенная задержка в получении результатов;
- ошибки и недоработки на любом из этапов проявляются, как правило, на последующих этапах работ, что приводит к необходимости возврата назад;
- сложность параллельного ведения работ по проекту;
- чрезмерная информационная перенасыщенность каждого из этапов;
- сложность управления проектом;
- высокий уровень риска и ненадежность инвестиций.

Основным недостатком этого подхода является то, что реальный процесс создания системы никогда полностью не укладывается в такую жесткую схему, постоянно возникает потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений. Это приводит к существенной задержке в получении результатов.

В результате реальный процесс создания ИС оказывается соответствующим поэтапной модели с промежуточным контролем.

Здесь разработка ИС ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют учитывать реально существующее взаимовлияние результатов разработки на различных этапах; время жизни каждого из этапов растягивается на весь период разработки.

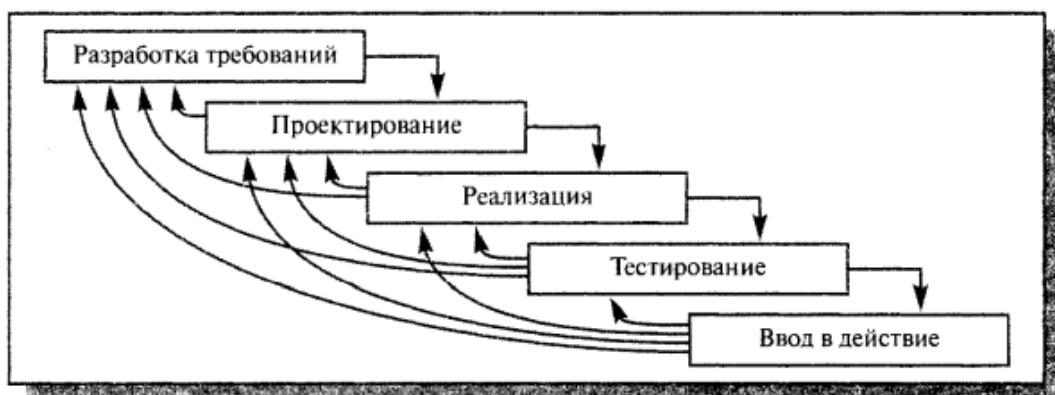


Рисунок 2.2 – Поэтапная модель с промежуточным контролем

Сpirальная модель ЖЦ была предложена для преодоления перечисленных проблем каскадной модели. На этапах анализа и проектирования реализуемость технических решений и степень удовлетворения потребностей заказчика проверяется путем создания прототипов.

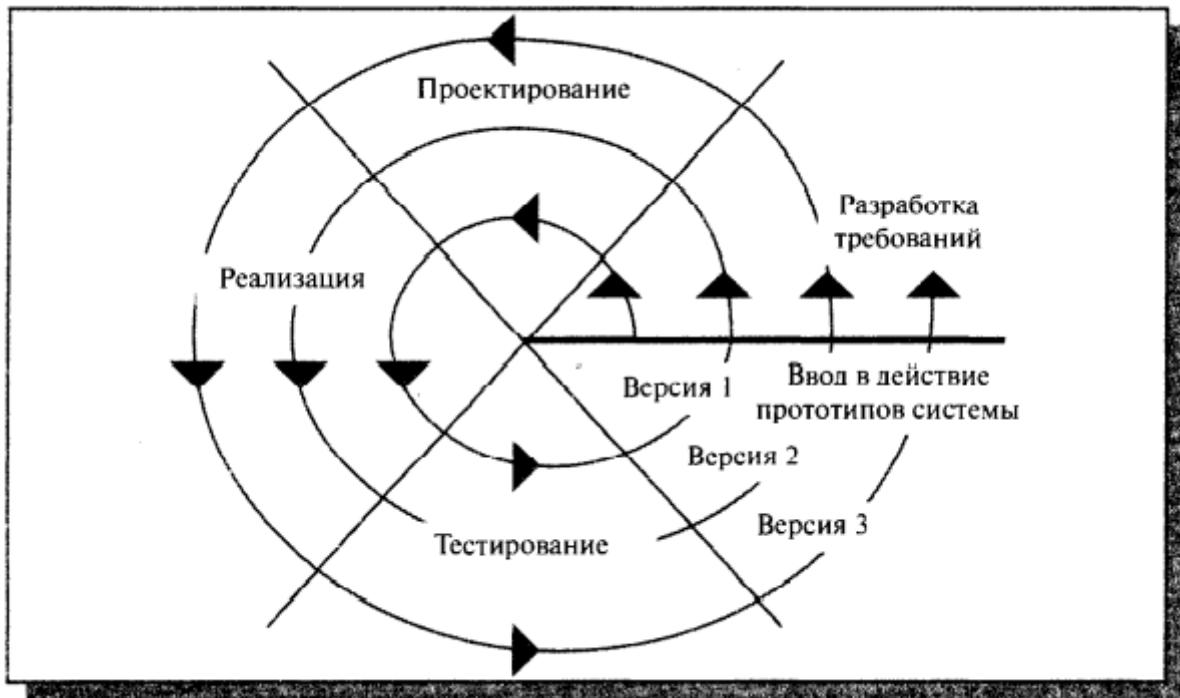


Рисунок 2.3 – Сpirальная модель ЖЦ ИС

На каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество и планируются работы следующего витка. Особое внимание уделяется начальным этапам разработки – анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов (макетирования).

Достоинства спиральной модели:

- позволяет быстрее показать пользователям системы работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований;
- допускает изменение требований при разработке программного обеспечения, что характерно для большинства разработок, в том числе типовых;
- в модели предусмотрена возможность гибкого проектирования, поскольку в ней воплощены преимущества каскадной модели и в то же время разрешены итерации по всем фазам этой же модели;
- позволяет получить более надежную и устойчивую систему, по мере развития программного обеспечения ошибки и слабые места обнаруживаются и исправляются на каждой итерации;
- эта модель разрешает пользователям активно принимать участие при планировании, анализе рисков, разработке, а также при выполнении оценочных действий;

- уменьшаются риски заказчика, который может с минимальными для себя финансовыми потерями завершить развитие неперспективного проекта;
- обратная связь по направлению от пользователей к разработчикам выполняется с высокой частотой и на ранних этапах модели, что обеспечивает создание нужного продукта высокого качества.

Недостатки спиральной модели:

- если проект имеет низкую степень риска или небольшие размеры, модель может оказаться дорогостоящей, оценка рисков после прохождения каждой спирали связана с большими затратами;
- жизненный цикл модели имеет усложненную структуру, поэтому может быть затруднено ее применение разработчиками, менеджерами и заказчиками;
- спираль может продолжаться до бесконечности, поскольку каждая ответная реакция заказчика на созданную версию может порождать новый цикл, что отдаляет окончание работы над проектом;
- большое количество промежуточных циклов может привести к необходимости обработки дополнительной документации;
- использование модели может оказаться дорогостоящим, так как время, затраченное на планирование, повторное определение целей, выполнение анализа рисков и прототипирование, может быть чрезмерным.

На практике наибольшее распространение получили две основные модели жизненного цикла:

- 1) каскадная модель (характерна для периода 1970-1985 гг.);
- 2) спиральная модель (характерна для периода после 1986 г.).

В ранних проектах достаточно простых ИС каждое приложение представляло собой единый функционально и информационно независимый блок. Для разработки такого типа приложений эффективным оказался каскадный способ. Каждый этап завершался после полного выполнения и документального оформления всех предусмотренных работ.

Однако и эта схема не позволяет оперативно учитывать возникающие изменения и уточнения требований к системе. Согласование результатов разработки с пользователями производится только в точках, планируемых после завершения каждого этапа работ, а общие требования к ИС зафиксированы в виде технического задания на все время ее создания. Таким образом, пользователи зачастую получают систему, не удовлетворяющую их реальным потребностям.

Итеративная разработка отражает объективно существующий спиральный цикл создания сложных систем. Она позволяет переходить на следующий этап, не дожидаясь полного завершения работы на текущем и решить главную задачу – как можно быстрее показать пользователям системы работоспособный продукт, тем самым активизируя процесс уточнения и дополнения требований.

Рассмотрим преимущества итерационного подхода, применяемого в спиральной модели, более подробно.

1. Существенно упрощается внесение изменений в проект при изменении требований заказчика.

2. При использовании спиральной модели отдельные элементы информационной системы интегрируются в единое целое постепенно. Поскольку интеграция начинается с меньшего количества элементов, то возникает гораздо меньше проблем при ее проведении (по некоторым оценкам, при использовании каскадной модели разработки интеграция занимает до 40 % всех затрат в конце проекта).

3. Уменьшение уровня рисков. Данное преимущество является следствием предыдущего, так как риски обнаруживаются именно во время интеграции. Поэтому уровень рисков максимальен в начале разработки проекта. По мере продвижения разработки ожидаемый уровень рисков снижается. Данное утверждение справедливо при любой модели разработки, однако при использовании спиральной модели снижение уровня рисков происходит с наибольшей скоростью. Это связано с тем, что при данном подходе интеграция выполняется уже на первой итерации, и на начальных витках спирали выявляются многие аспекты проекта, такие как пригодность используемых инструментальных средств и программного обеспечения, квалификация разработчиков и т. п.

4. Итерационная разработка обеспечивает большую гибкость в управлении проектом, давая возможность внесения тактических изменений в разрабатываемое изделие. Например, можно сократить сроки разработки за счет снижения функциональности системы или использовать в качестве составных частей системы продукцию сторонних фирм вместо собственных разработок.

5. Итерационный подход спиральной модели ЖЦ упрощает повторное использование компонентов (реализует компонентный подход к программированию). Это обусловлено тем, что гораздо проще выявить (идентифицировать) общие части проекта, когда они уже частично разработаны, чем пытаться выделить их в самом начале проекта. Анализ проекта после проведения нескольких начальных итераций позволяет выявить общие многократно используемые компоненты, которые на последующих итерациях будут совершенствоваться.

Основная проблема спирального цикла – определение момента перехода на следующий этап. Для ее решения вводятся временные ограничения на каждый из этапов жизненного цикла, и переход осуществляется в соответствии с планом, даже если не вся запланированная работа закончена. Планирование производится на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков.

6. Итерационный подход дает возможность совершенствовать процесс разработки – анализ, проводимый в конце каждой итерации, позволяет проводить оценку того, что должно быть изменено в организации разработки, и улучшить ее на следующей итерации.

Несмотря на настойчивые рекомендации экспертов в области проектирования и разработки информационных систем, некоторые компании

продолжают использовать каскадную модель вместо других вариантов. Основными причинами, по которой каскадная модель сохраняет свою популярность, являются:

1. **Привычка** – многие ИТ-специалисты получили образование в то время, когда изучалась и использовалась только каскадная модель, поэтому она используется ими и в наши дни.
2. **Иллюзия снижения рисков** участников проекта (заказчика и исполнителя).

Есть два основных типа контрактов на разработку ПО. *Первый тип* предполагает выполнение определенного объема работ за определенную сумму в определенные сроки (fixed price). *Второй тип* предполагает повременную оплату работы (time work). Выбор того или иного типа контракта зависит от степени определенности задачи.

Каскадная модель с определенными этапами и их результатами лучше приспособлена для заключения контракта с оплатой по результатам работы, а именно этот тип контрактов позволяет получить полную оценку стоимости проекта до его завершения. Более вероятно заключение контракта с повременной оплатой на небольшую систему, с относительно небольшим весом в структуре затрат предприятия.

Разработка и внедрение интегрированной информационной системы требуют существенных финансовых затрат, поэтому используются контракты с фиксированной ценой, и, следовательно, каскадная модель разработки и внедрения. Спиральная модель чаще применяется при разработке информационной системы силами собственного отдела ИТ предприятия.

3. **Проблемы внедрения** при использовании итерационной модели. В некоторых областях спиральная модель не может применяться, поскольку невозможно использование/тестирование продукта, обладающего неполной функциональностью (например, военные разработки, атомная энергетика и т. д.). Поэтапное итерационное внедрение информационной системы для бизнеса возможно, но сопряжено с организационными сложностями (перенос данных, интеграция систем, изменение бизнес-процессов, учетной политики, обучение пользователей). Трудозатраты при поэтапном итерационном внедрении оказываются значительно выше, а управление проектом требует настоящего искусства.

Предвидя указанные сложности, заказчики выбирают каскадную модель, чтобы «внедрять систему один раз».

В последнее время достаточно популярной также стала **модель непрерывной разработки (продолжающейся разработки)** (рисунок 2.4).



Рисунок 2.4 – Модель непрерывной разработки

Особенностью этого подхода является непрерывный процесс разработки и развития больших информационных систем с планируемыми точками передачи в эксплуатацию новых версий и новых функциональных блоков (подсистем, задач).

Кроме того, модель продолжающейся разработки с целью еще более ускоренного внедрения трансформировалась в несколько моделей жизненного цикла, получивших название «*быстрого прототипирования*»: *fast track*, *lite* и другие. Они содержат небольшое количество этапов, например, в модели *fast track* (Oracle CDM) их всего три: моделирование требований; проектирование и создание системы; внедрение.

Модели быстрого прототипирования применимы для ИС средней и малой сложности, наряду с быстрым эффектом дают снижение управляемости проекта в целом и приводят к появлению проблем при стыковке различных фрагментов ИС.

Инкрементная модель жизненного цикла (рисунок 2.5) представляет собой пример итеративного подхода к разработке программного обеспечения ИС, который предполагает разбиение жизненного цикла проекта на последовательность итераций, каждая из которых напоминает «мини-проект», включающий все фазы жизненного цикла в применении к созданию отдельных версий системы, обладающих меньшей функциональностью по сравнению с проектом в целом.

При этом на каждой итерации получается работающая версия программной системы, обладающая функциональностью всех предыдущих плюс текущей итерации. В результате финальной итерации получается конечный продукт, обеспечивающий реализацию всех требований.

С точки зрения структуры жизненного цикла модель называют *итеративной* (говоря о процессе). С точки зрения развития продукта – *инкрементной* (имеется ввиду наращивание функциональности продукта). *Инкремент – приращение, увеличение (например, в языке программирования – увеличение значения переменной на 1).*



Рисунок 2.5 – Инкрементная модель жизненного цикла

Для каждого инкремента выполняется:

- анализ, на котором мы собираем требования, анализируем и планируем сам инкремент;
- проектирование, на котором происходит проектирование архитектуры, допроектирование тех вещей, которые не были сделаны на предыдущем инкременте;
- разработка;
- тестирование.

Важно, что каждый инкремент заканчивается работающим продуктом.

Пусть он ограниченной функциональности, пусть у него не все реализовано, но это отдельный продукт, который можно показать, который можно отдать заказчику на тестирование.

Сам подход является однократным, т. е. мы весь наш проект делаем за один большой этап, имея в виду требования, но весь проект делится на инкременты, это небольшие версии продуктов, для которых заранее определена функциональность. То есть мы говорим, что у нас вся длительная разработка состоит, например, из 5 фаз. На первом этапе получим однопользовательскую версию, на втором инкременте (фазе) мы получим версию, которая поддерживает много пользователей, на третьем инкременте у нас появится поддержка веб-интерфейсов, а на четвертом какое-нибудь протоколирование и т. д. (рисунок 2.6). То есть хоть требования у нас все равно задаются только один раз, но на выходе у нас появляется несколько инкрементов.

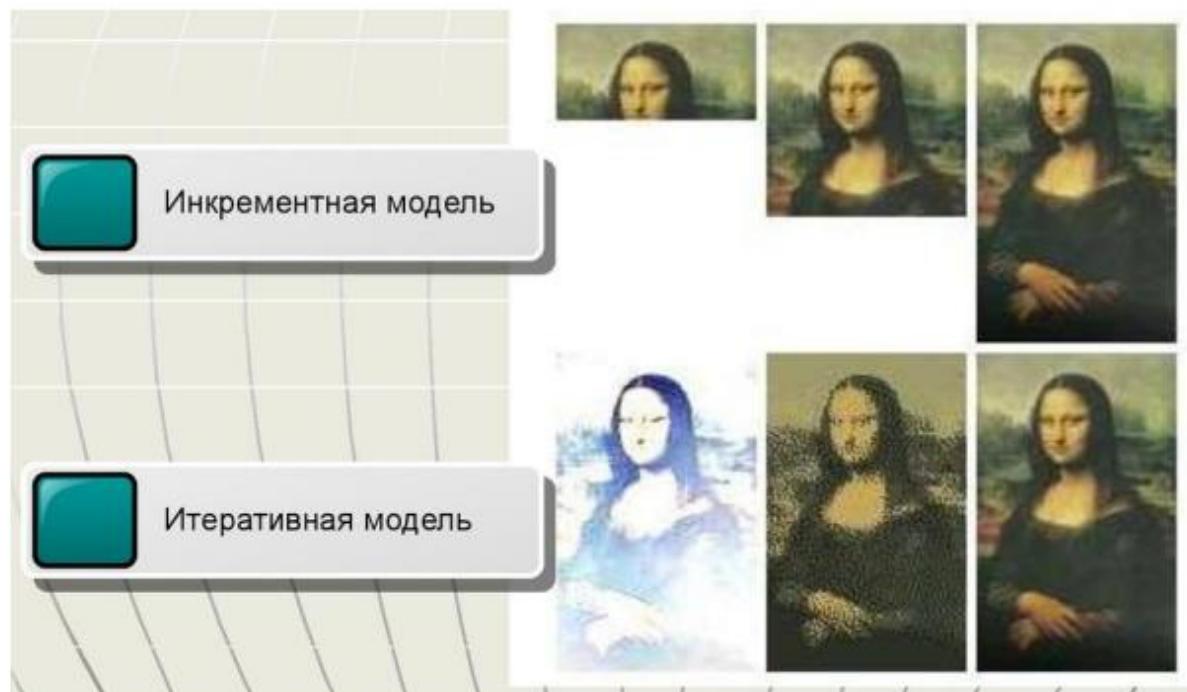


Рисунок 2.6 – Иллюстрация различий между инкрементной и итеративной (спиральной) моделями

Преимущества инкрементной модели:

- на момент создания определенного инкремента требования стабилизируются, поскольку не являющиеся особо важными изменения отодвигаются на момент создания последующих инкрементов;
- не требуется заранее тратить средства, необходимые для разработки всего проекта, поскольку сначала выполняется разработка и реализация основной функции или функции из группы высокого риска;
- в результате выполнения каждого инкремента получается функциональный продукт;
- ускоряется начальный график поставки (что позволяет соответствовать возросшим требованиям рынка);
- риск распределяется на несколько меньших по размеру инкрементов (не сосредоточен в одном большом проекте разработки);
- в конце каждой инкрементной поставки существует возможность пересмотреть риски, связанные с затратами и соблюдением установленного графика.

Недостатки инкрементной модели:

- в модели не предусмотрены итерации в рамках каждого инкремента;
- определение полной функциональности системы должно осуществляться в начале жизненного цикла, чтобы обеспечить определение инкрементов;
- формальный критический анализ и проверку намного труднее выполнить для инкрементов, чем для системы в целом;
- поскольку создание некоторых модулей будет завершено значительно раньше других, возникает необходимость в четко определенных интерфейсах;
- для модели необходимо хорошее планирование и проектирование;
- может возникнуть тенденция к оттягиванию решений трудных проблем на будущее с целью продемонстрировать руководству успех, достигнутый на ранних этапах разработки.

Для правильного выбора модели ЖЦ ИС обычно руководствуются четырьмя критериями: **стоимость, риск, качество, скорость.**

В ходе рассмотрения жизненных циклов ИС используют кроме всего прочего такие понятия, как:

Стадии ЖЦ – отражают состояния ИС и их изменения. Каждая из стадий создания системы предусматривает выполнение определенного объема работ, которые представляются в виде **процессов ЖЦ**.

Процессы ЖЦ – отражают те действия, которые должны обязательно выполняться для эффективного проектирования ИС.

Процесс определяется как *совокупность взаимосвязанных действий, преобразующих входные данные в выходные*. Описание каждого процесса включает в себя перечень решаемых задач, исходных данных и результатов.

Этапы жизненного цикла – входят в состав стадий; предполагают выполнение определенного объема работ в течение ограниченного времени; определяются как совокупность взаимосвязанных действий, преобразующих входные данные в выходные. Одни и те же процессы могут выполняться на различных стадиях (этапах) ЖЦ.

2. Стандарты, регламентирующие жизненные циклы ИС

Существует целый ряд стандартов, регламентирующих ЖЦ ПО, а в некоторых случаях и процессы разработки.

Значительный вклад в теорию проектирования и разработки информационных систем внесла компания IBM, предложив еще в середине 1970-х годов методологию BSP (Business System Planning – методология организационного планирования).

Метод структурирования информации с использованием матриц пересечения бизнес-процессов, функциональных подразделений, функций систем обработки данных (информационных систем), информационных объектов, документов и баз данных, предложенных в BSP, используются сегодня не только в ИТ-проектах, но и в проектах по реинжинирингу бизнес-процессов, изменению организационной структуры. Важнейшие шаги процесса BSP, их последовательность можно встретить практически во всех формальных методиках, а также в проектах, реализуемых на практике.

Среди наиболее известных стандартов можно выделить следующие:

ГОСТР 59793-2021 – распространяется на автоматизированные системы и устанавливает стадии и этапы их создания. Кроме того, в стандарте есть описание содержания работ на каждом этапе. Стадии и этапы работы, закрепленные в стандарте, в большей степени соответствуют каскадной модели жизненного цикла.

ISO/IEC 12207:1995 – стандарт на процессы и организацию жизненного цикла. Распространяется на все виды заказного ПО. Стандарт не содержит описания фаз, стадий и этапов.

Custom Development Method (методика Oracle) по разработке прикладных информационных систем – технологический материал, детализированный до уровня заготовок проектных документов, рассчитанных на использование в проектах с применением Oracle. Применяется CDM для классической модели ЖЦ (предусмотрены все работы/задачи и этапы), а также для технологий «быстрой разработки» (Fast Track) или «облегченного подхода», рекомендуемых в случае малых проектов.

Rational Unified Process (RUP) предлагает итеративную модель разработки, включающую четыре фазы: начало, исследование, построение и внедрение. Каждая фаза может быть разбита на этапы (итерации), в результате которых выпускается версия для внутреннего или внешнего использования. Прохождение через четыре основные фазы называется циклом разработки, каждый цикл завершается генерацией версии системы. Если после этого работа над проектом не прекращается, то полученный продукт продолжает развиваться и снова минует те же фазы. Суть работы в рамках RUP – это создание и сопровождение моделей на базе UML.

Microsoft Solution Framework (MSF) сходна с RUP, также включает четыре фазы: *анализ, проектирование, разработка, стабилизация*, является итерационной, предполагает использование объектно-ориентированного

моделирования. MSF в сравнении с RUP в большей степени ориентирована на разработку бизнес-приложений.

Extreme Programming (EP). Экстремальное программирование (самая новая среди рассмотренных методологий) сформировалось в 1996 году. В основе методологии – командная работа, эффективная коммуникация между заказчиком и исполнителем в течение всего проекта по разработке ИС; разработка ведется с использованием последовательно дорабатываемых прототипов.

Для поддержки практического применения стандарта ISO/IEC 12207 был разработан ряд технологических документов: Руководство для ISO/IEC 12207 (ISO/IEC TR 15271:1998), руководство по применению ISO/IEC 12207 и др.

Позднее был разработан и в 2002 г. опубликован ***стандарт на процессы жизненного цикла систем (ISO/IEC 15288 System life cycle processes).***

К разработке стандарта были привлечены специалисты различных областей: системной инженерии, программирования, управления качеством, человеческими ресурсами, безопасностью и пр. Был учтен практический опыт создания систем в правительственные, коммерческих, военных и академических организациях. Стандарт применим для широкого класса систем, но его основное предназначение – поддержка создания компьютеризированных систем.

ЛЕКЦИЯ № 3. ОРГАНИЗАЦИЯ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ

- 1. Средства проектирования ИС.**
- 2. Каноническое проектирование.**
- 3. Типовое проектирование.**

1. Средства проектирования ИС

Технологии индустриального проектирования информационных систем, включающие *типовыe и автоматизированныe* технологии проектирования, отличаются от канонической технологии применением специальных программных средств для повышения качества и скорости проектирования. Такие средства называются ***CASE-средствами*** (Computer Aided System/Software Engineering – создание систем и ПО с помощью компьютера). CASE-средства классифицируются по ряду признаков.

По функциональному назначению выделяют следующие типы:

A. Основные типы:

- 1) *средства анализа* – формируют и анализируют модели объектов автоматизации (например, BPwin, Designer/IDEF);
- 2) *средства анализа и проектирования* – поддерживают наиболее распространенные методологии проектирования для создания проектных спецификаций: проектных компонентов интерфейсов системы,

архитектуры системы, алгоритмов, структур данных (например, Silverrun, семейство программ ARIS);

- 3) *средства проектирования баз данных* – выполняют моделирование данных и создание схем баз данных в формате распространенных СУБД (например, ERwin, Database Designer);
- 4) *средства разработки приложений и генераторы кода* – обычно входят в состав сред программирования (например, Delphi);
- 5) *средства реинжиниринга* – производят анализ и отладку программных кодов и схем баз данных и формируют на их основе проектные спецификации (например, PRO-4, ERwin, Silverrun).

Б. Вспомогательные типы:

- 1) *средства конфигурирования* (конфигурационного управления) – обеспечивают управление и контроль процесса разработки и сопровождения ПО посредством идентификации состояния компонент системы (например, комплекс программ PVCS);
- 2) *средства тестирования* – исполняют программу в тестирующем режиме с целью обнаружения ошибок (например, программа QA (Quality Works));
- 3) *средства документирования* – представляют собой средства формирования отчётов и компоненты издательских систем (например, SoDA);
- 4) *средства планирования и управления проектом* (например, Microsoft Office Project).

По степени интегрированности выделяют следующие категории:

- 1) отдельные локальные CASE-средства, решющие небольшие автономные задачи;
- 2) частично интегрированные средства, поддерживающие большинство этапов жизненного цикла ИС;
- 3) полностью интегрированные средства, охватывающие весь жизненный цикл ИС.

По применяемым методологиям и моделям систем и баз данных различают:

- 1) малые CASE-средства, используемые для создания небольших ИС и поддерживающие до 5 моделей и методов (Silverrun, ERwin, BPwin);
- 2) средние CASE-средства, поддерживающие до 15 моделей и методов (например, Rational Rose);
- 3) крупные CASE-средства, поддерживающие свыше 15 типов моделей и методов и используемые для создания полнофункциональных сложных информационных систем (например, ПК ARIS).

Существуют также классификации *по степени интегрированности с СУБД* и *по доступным платформам*.

Автоматизированное проектирование ИС с применением средств компьютерной поддержки иначе называется ***CASE-технологией проектирования***. Её основные принципы:

- непрерывная компьютерная поддержка всего процесса проектирования ИС;

- применение функционально-ориентированного или объектно-ориентированного подхода к моделированию и проектированию ИС;
- иерархическое представление модели предметной области на основе декомпозиции системы в соответствии с методом проектирования «сверху вниз»;
- обеспечение наглядности представления модели с помощью визуальных средств проектирования (с применением нотации методологии, реализованной в конкретном CASE-средстве);
- использование централизованного хранилища данных – репозитория;
- уделение наибольшего внимания не программированию, а анализу и проектированию ИС;
- разбиение процесса автоматизированного проектирования ИС с применением CASE-средств на этапы.

Этапами **технологии автоматизированного проектирования** с применением CASE-средств являются:

1. Анализ предметной области:

1. Предпроектное обследование объекта автоматизации.
2. Разработка CASE-модели структуры реального объекта, анализ разработанной CASE-модели, разработка предложений по устранению недостатков.
3. Разработка вариантов усовершенствования CASE-модели, выбор оптимального варианта в качестве ТЗ на создание ИС.

2. Проектирование ИС:

1. Декомпозиция иерархической модели ИС на основе функционально-ориентированного или объектно-ориентированного подхода.
2. Разработка детализирующих моделей и диаграмм.

3. Программирование ИС.

4. Внедрение ИС, её сопровождение на основе CASE-модели.

Её реализация основана на применении CASE-средств. Критериями отнесения программного продукта к группе CASE-средств являются следующие особенности:

- наличие графических средств для визуального описания и документирования ИС (реализованных нотаций каких-либо методологий проектирования);
- интеграция отдельных функциональных компонент поддержки проектирования ИС в одном средстве, обеспечивающем компьютерной поддержкой весь процесс проектирования;
- наличие единого хранилища проектных метаданных – репозитария.

В состав полнофункционального интегрированного CASE-средства входят следующие компоненты:

- репозитарий – основа CASE-средства – специальная база данных проекта, единое хранилище метаданных для всех компонент CASE-средства, содержащее версии проекта ИС, информацию о проектировщиках и их правах доступа, диаграммах и их компонентах, связях между ними, о

- программных модулях и прочее, синхронизирующее информацию из различных компонент программы и обеспечивающее полноту и непротиворечивость метаданных;
- графические средства анализа и проектирования, с помощью которых осуществляется визуализация моделирования, построение диаграмм модели (SADT, DFD, ERD);
 - средства контроля и сбора статистики, выделяющие на диаграммах ошибочные элементы и собирающие статистику ошибок;
 - средства администрирования проекта, предназначенные для управления доступом к репозитарию и для контроля хода выполнения проекта;
 - браузер для просмотра диаграмм моделей ИС;
 - средства разработки приложений (программирования);
 - средства конфигурационного управления;
 - средства документирования;
 - средства тестирования;
 - средства реинжиниринга;
 - средства управления проектом.

Все современные методологии проектирования ИС и реализующие их программные продукты поставляются в электронном виде вместе с CASE-средствами, включающими библиотеки стандартных моделей, шаблонов, процессов и методов, а также специальный интерфейс для адаптации типовых решений и модули контроля непротиворечивости вновь создаваемых элементов системы.

Путём настройки типовых элементов из библиотек, добавления или удаления процессов и объектов в типовой модели системы можно быстро создать новую модель проектируемой системы. На этом основано **типовое проектирование ИС** – создание системы из готовых типовых элементов (ТПР – типовых (тиражируемых) проектных решений).

В зависимости от того, насколько крупные части системы формируются из типовых решений, взятых в специальной электронной библиотеке, выделяют три уровня ТПР.

Элементные ТПР воспроизводят отдельные действия, выполняемые процессы, решаемые системой задачи. Это обеспечивает реализацию модульного подхода к проектированию и документированию ИС, однако приводит к большим времененным затратам на сопряжение отдельных элементов в единую систему.

Подсистемные ТПР, в которых в качестве элементов типизации выступают отдельные подсистемы, разработанные с учётом функциональной полноты и минимизации внешних информационных связей, обеспечивают сокращение затрат на проектирование и программирование взаимосвязанных компонентов и высокую степень интеграции элементов системы, но снижают степень адаптивности системы к изменению отдельных процессов в моделируемой системе.

Объектные ТПР – это типовые отраслевые проекты, которые включают полный набор стандартных функциональных и обеспечивающих подсистем ИС. Такие проекты обладают свойствами конфигурируемости (возможности выбирать круг включаемых в систему модулей) и масштабируемости (настройки под конкретное количество пользователей), но при этом, воспроизводя типовую систему, могут значительно отличаться по структуре и функционалу от реальной моделируемой системы.

По типовому принципу строятся обычно небольшие информационные системы для автоматизации деятельности стандартных объектов управления: промышленных предприятий со стандартным операционным циклом, предприятий обслуживания. Для них легче всего подобрать состав компонентов ИС из перечня доступных ТПР.

В рамках типового проектирования применяются два подхода: **параметрически-ориентированное** и **модельно-ориентированное** проектирование, – отличающиеся способом, которым структура ТПР адаптируется к структуре конкретной проектируемой системы.

В **параметрически-ориентированном проектировании** изменяются параметры ТПР (количественные и качественные характеристики элементов системы и их связей).

Технология параметрически-ориентированного проектирования заключается в следующем:

- 1) выбор критериев оценки ТПР на пригодность для использования в конкретной моделируемой системе (например, по назначению, свойствам, техническим требованиям, документации, возможности адаптации, стоимости приобретения и доработки);
- 2) сравнение различных ТПР по выбранным критериям и выбор пригодных;
- 3) адаптация выбранного ТПР к решаемой задаче путём настройки его параметров.

Модельно-ориентированное проектирование подразумевает адаптацию самой структуры модели ТПР, её состава и характеристик. Основой его является репозитарий применяемого CASE-средства – специальная база метаданных («данных о данных»). В репозитарии хранятся:

- базовая (ссылочная) модель ИС – общая модель любой организации, включающая набор общих функций, процессов, объектов, правил, организационную структуру, которые могут быть использованы программными модулями типовых ИС;
- типовые (референтные) модели ИС, описывающие некоторые классовые (например, среднеотраслевые) структуры объектов, и использующие специфические наборы функций, процессов и объектов из базовой модели;
- модели конкретных организаций.

Очевидно, что модель реального автоматизируемого объекта может отличаться от любой из имеющихся в репозитарии, поэтому в модельно-ориентированном проектировании применяют специальные CASE-средства (такие, как BAAN Enterprise Modeler, SAP Business Engineering Workbench), с

помощью которых на основе модели автоматизируемого объекта осуществляется конфигурирование программного обеспечения ИС этого объекта. При этом в разных CASE-средствах адаптация осуществляется по-разному: в одних конкретная модель ИС собирается из готовых типовых элементов (например, в BAAN Enterprise Modeler), в других уже готовая типовая модель изменяется по заданному алгоритму адаптации (как в SAP Business Engineering Workbench).

Примечательно, что в репозитарии хранится не описание самой информационной системы, а описание её модели – метаописание, – на основе которой CASE-средство само конфигурирует и настраивает информационную систему.

Технология модельно-ориентированного проектирования имеет следующий вид:

- 1) устанавливаются глобальные параметры системы;
- 2) определяется организационная и функциональная структура объекта автоматизации;
- 3) определяется структура баз данных;
- 4) формируется перечень выполняемых системой функций и процессов;
- 5) описываются интерфейсы различных модулей;
- 6) задаётся структура отчётов, содержащих выходные данные;
- 7) настраиваются системы контроля доступа и архивации данных.

Обычно CASE-средства ориентированы на какой-либо один подход к проектированию ИС – функционально-ориентированный или объектно-ориентированный, и поддерживают соответствующие подходу методы. Но в настоящее время существуют программные продукты – так называемые мосты (или интерфейсы), связывающие какие-либо два CASE-средства: одно – функционально-ориентированное, другое – объектно-ориентированное. Например, есть мосты для связи функционально-ориентированного CASE-средства ERwin и объектно-ориентированных CASE-средств Rational Rose или Paradigm Plus, или функционально-ориентированного Silverrun и объектно-ориентированного Rational Rose.

В заключение отметим, что в современных условиях создание сложных информационных систем обязательно осуществляется с применением программных средств поддержки процесса создания ИС. В настоящее время разработаны CASE-средства поддержки всех этапов жизненного цикла ПО ИС. Многие из них совместимы либо с помощью внутренних функций экспорта/импорта данных, либо посредством специальных программных средств – мостов. Таким образом, можно подобрать группу CASE-средств для полной реализации проекта создания, внедрения и развития ИС.

2. Каноническое проектирование

Организация канонического проектирования ИС ориентирована на использование главным образом каскадной модели жизненного цикла ИС. Стадии и этапы работы описаны в стандарте ГОСТ 5793-2021.

В зависимости от сложности объекта автоматизации и набора задач, требующих решения при создании конкретной ИС, стадии и этапы работ могут иметь различную трудоемкость. Допускается объединять последовательные этапы и даже исключать некоторые из них на любой стадии проекта. Допускается также начинать выполнение работ следующей стадии до окончания предыдущей.

Стадии и этапы создания ИС, выполняемые организациями – участниками, прописываются в договорах и технических заданиях на выполнение работ:

Стадия 1. Формирование требований к ИС

На начальной стадии проектирования выделяют следующие этапы работ:

- обследование объекта и обоснование необходимости создания ИС;
- формирование требований пользователей к ИС;
- оформление отчета о выполненной работе и тактико-технического задания на разработку.

Стадия 2. Разработка концепции ИС

- изучение объекта автоматизации;
- проведение необходимых научно-исследовательских работ;
- разработка вариантов концепции ИС, удовлетворяющих требованиям пользователей;
- оформление отчета и утверждение концепции.

Стадия 3. Техническое задание

- разработка и утверждение технического задания на создание ИС.

Стадия 4. Эскизный проект

- разработка предварительных проектных решений по системе и ее частям;
- разработка эскизной документации на ИС и ее части.

Стадия 5. Технический проект

- разработка проектных решений по системе и ее частям;
- разработка документации на ИС и ее части;
- разработка и оформление документации на поставку комплектующих изделий;
- разработка заданий на проектирование в смежных частях проекта.

Стадия 6. Рабочая документация

- разработка рабочей документации на ИС и ее части;
- разработка и адаптация программ.

Стадия 7. Ввод в действие

- подготовка объекта автоматизации;
- подготовка персонала;

- комплектация ИС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями);
- строительно-монтажные работы;
- пусконаладочные работы;
- проведение предварительных испытаний;
- проведение опытной эксплуатации;
- проведение приемочных испытаний.
- Стадия 8. Сопровождение ИС
- выполнение работ в соответствии с гарантийными обязательствами;
- послегарантийное обслуживание.

Обследование – это изучение и диагностический анализ организационной структуры предприятия, его деятельности и существующей системы обработки информации.

Материалы, полученные в результате обследования, используются для:

- обоснования разработки и поэтапного внедрения систем;
- составления технического задания на разработку систем;
- разработки технического и рабочего проектов систем.

На этапе обследования целесообразно выделить *две составляющие: определение стратегии внедрения ИС и детальный анализ деятельности организации.*

Основная задача первого этапа обследования – оценка реального объема проекта, его целей и задач на основе выявленных функций и информационных элементов автоматизируемого объекта высокого уровня.

Эти задачи могут быть реализованы или заказчиком ИС самостоятельно, или с привлечением консалтинговых организаций. Этап предполагает тесное взаимодействие с основными потенциальными пользователями системы и бизнес-экспертами. **Основная задача взаимодействия** – получение полного и однозначного понимания требований заказчика. Как правило, нужная информация может быть получена в результате интервью, бесед или семинаров с руководством, экспертами и пользователями.

По завершении этой стадии обследования появляется возможность определить вероятные технические подходы к созданию системы и оценить затраты на ее реализацию (затраты на аппаратное обеспечение, закупаемое программное обеспечение и разработку нового программного обеспечения).

Результатом этапа определения стратегии является документ (**технико-экономическое обоснование проекта**), в котором четко сформулировано, что получит заказчик, если согласится финансировать проект, когда он получит готовый продукт (график выполнения работ) и сколько это будет стоить (для крупных проектов должен быть составлен график финансирования на разных этапах работы). В документе желательно отразить не только затраты, но и выгоду проекта, например, время окупаемости проекта, ожидаемый экономический эффект (если его удается оценить).

Ориентировочное содержание этого документа:

- ограничения, риски, критические факторы, которые могут повлиять на успешность проекта;
- совокупность условий, при которых предполагается эксплуатировать будущую систему: архитектура системы, аппаратные и программные ресурсы, условия функционирования, обслуживающий персонал и пользователи системы;
- сроки завершения отдельных этапов, форма приемки/сдачи работ, привлекаемые ресурсы, меры по защите информации;
- описание выполняемых системой функций;
- возможности развития системы;
- информационные объекты системы;
- интерфейсы и распределение функций между человеком и системой;
- требование к программным и информационным компонентам ПО, требования к СУБД;
- что не будет реализовано в рамках проекта.

На этапе детального анализа деятельности организации изучаются задачи, обеспечивающие реализацию функций управления, организационная структура, штаты и содержание работ по управлению предприятием, а также характер подчиненности вышестоящим органам управления. На этом этапе должны быть выявлены:

- инструктивно-методические и директивные материалы, на основании которых определяется состав подсистем и перечень задач;
- возможности применения новых методов решения задач.

Аналитики собирают и фиксируют информацию в двух взаимосвязанных формах:

- функции – информация о событиях и процессах, которые происходят в бизнесе;
- сущности – информация о вещах, имеющих значение для организации и о которых что-то известно.

При изучении каждой функциональной задачи управления определяются:

- наименование задачи; сроки и периодичность ее решения;
- степень формализации задачи;
- источники информации, необходимые для решения задачи;
- показатели и их количественные характеристики;
- порядок корректировки информации;
- действующие алгоритмы расчета показателей и возможные методы контроля;
- действующие средства сбора, передачи и обработки информации;
- действующие средства связи;
- принятая точность решения задачи;
- трудоемкость решения задачи;

- действующие формы представления исходных данных и результатов их обработки в виде документов;
- потребители результатной информации по задаче.

Одной из наиболее трудоемких, хотя и хорошо формализуемых задач этого этапа является описание документооборота организации. При обследовании документооборота составляется схема маршрута движения документов, которая должна отразить:

- количество документов;
- место формирования показателей документа;
- взаимосвязь документов при их формировании;
- маршрут и длительность движения документа;
- место использования и хранения данного документа;
- внутренние и внешние информационные связи;
- объем документа в знаках.

По результатам обследования устанавливается перечень задач управления, решение которых целесообразно автоматизировать, и очередность их разработки.

На этапе обследования следует классифицировать планируемые функции системы по степени важности. Один из возможных форматов представления такой классификации – MuSCoW.

Эта аббревиатура расшифровывается так: *Must have* – необходимые функции; *Should have* – желательные функции; *Could have* – возможные функции; *Won't have* – отсутствующие функции.

Функции первой категории обеспечивают критичные для успешной работы системы возможности.

Реализация функций второй и третьей категорий ограничивается временными и финансовыми рамками: разрабатывается то, что необходимо, а также возможное в порядке приоритета число функций второй и третьей категорий.

Последняя категория функций особенно важна, поскольку необходимо четко представлять границы проекта и набор функций, которые будут отсутствовать в системе.

Модели деятельности организаций создаются двух видов:

Модель «как есть» («as-is») – *отражает существующие в организации бизнес-процессы*;

Модель «как должно быть» («to be») – *отражает необходимые изменения бизнес-процессов, с учетом внедрения ИС*.

На этапе анализа необходимо привлекать к работе группы тестирования для решения следующих задач:

- получения сравнительных характеристик предполагаемых к использованию аппаратных платформ, операционных систем, СУБД, иного окружения;
- разработка плана работ по обеспечению надежности информационной системы и ее тестирования.

Привлечение тестировщиков на ранних этапах разработки является целесообразным для любых проектов. Если проектное решение оказалось неудачным и это обнаружено слишком поздно (на этапе разработки или, что еще хуже, на этапе внедрения в эксплуатацию), то исправление ошибки проектирования обходится очень дорого. Чем раньше группы тестирования выявляют ошибки в информационной системе, тем ниже стоимость сопровождения системы. Время на тестирование системы и на исправление обнаруженных ошибок следует предусматривать не только на этапе разработки, но и на этапе проектирования.

Для автоматизации проектирования следует использовать системы отслеживания ошибок (bug tracking). Это позволяет иметь единое хранилище ошибок, отслеживать их повторное появление, контролировать скорость и эффективность исправления ошибок, видеть наиболее нестабильные компоненты системы, а также поддерживать связь между группой разработчиков и группой тестирования (уведомления об изменениях). Чем больше проект, тем сильнее потребность в bug tracking.

Результаты обследования представляют объективную основу для формирования технического задания на информационную систему.

Техническое задание – это документ, определяющий цели, требования и основные исходные данные, необходимые для разработки автоматизированной системы управления.

При разработке технического задания необходимо решить следующие задачи:

- установить общую цель создания ИС, определить состав подсистем и функциональных задач;
- разработать и обосновать требования, предъявляемые к подсистемам;
- разработать и обосновать требования, предъявляемые к информационной базе, математическому и программному обеспечению, комплексу технических средств (включая средства связи и передачи данных);
- установить общие требования к проектируемой системе;
- определить перечень задач создания системы и исполнителей;
- определить этапы создания системы и сроки их выполнения;
- провести предварительный расчет затрат на создание системы и определить уровень экономической эффективности ее внедрения.

ТЗ на АС содержит следующие обязательные разделы:

- общие сведения;
- цели и назначение создания автоматизированной системы;
- характеристика объектов автоматизации;
- требования к автоматизированной системе;
- состав и содержание работ по созданию автоматизированной системы;
- порядок разработки автоматизированной системы;
- порядок контроля и приемки автоматизированной системы;

- требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие;
- требования к документированию;
- источники разработки.

В ТЗ на АС могут быть включены приложения.

В зависимости от вида, назначения, специфических особенностей объекта автоматизации и условий функционирования АС допускается оформлять разделы ТЗ в виде приложений, вводить дополнительные разделы ТЗ. Разделы ТЗ могут быть разделены на подразделы. Допускается вводить дополнительные, исключать или объединять подразделы ТЗ.

Эскизный проект предусматривает *разработку предварительных проектных решений по системе и ее частям*.

Выполнение стадии эскизного проектирования не является строго обязательной. Если основные проектные решения определены ранее или достаточно очевидны для конкретной ИС и объекта автоматизации, то эта стадия может быть исключена из общей последовательности работ.

Содержание эскизного проекта задается в ТЗ на систему. **Как правило, на этапе эскизного проектирования определяются:**

- функции ИС;
- функции подсистем, их цели и ожидаемый эффект от внедрения;
- состав комплексов задач и отдельных задач;
- концепция информационной базы и ее укрупненная структура;
- функция системы управления базой данных;
- состав вычислительной системы и других технических средств;
- функции и параметры основных программных средств.

По результатам проделанной работы оформляется, согласовывается и утверждается документация в объеме, необходимом для описания полной совокупности принятых проектных решений и достаточном для дальнейшего выполнения работ по созданию системы. На основе технического задания (и эскизного проекта) разрабатывается технический проект ИС.

Технический проект системы – это техническая документация, содержащая общесистемные проектные решения, алгоритмы решения задач, а также оценку экономической эффективности автоматизированной системы управления и перечень мероприятий по подготовке объекта к внедрению.

На этом этапе осуществляется комплекс научно-исследовательских и экспериментальных работ для выбора основных проектных решений и расчет экономической эффективности системы. Состав и содержание технического проекта приведены в таблице 3.2.

Таблица 3.2 – Содержание технического проекта

№ п\в	Раздел	Содержание
1	Пояснительная записка	<ul style="list-style-type: none"> • основания для разработки системы • перечень организаций разработчиков • краткая характеристика объекта с указанием основных технико-экономических показателей его функционирования и связей с другими объектами • краткие сведения об основных проектных решениях по функциональной и обеспечивающим частям системы
2	Функциональная и организационная структура системы	<ul style="list-style-type: none"> • обоснование выделяемых подсистем, их перечень и назначение • перечень задач, решаемых в каждой подсистеме, с краткой характеристикой их содержания • схема информационных связей между подсистемами и между задачами в рамках каждой подсистемы
3	Постановка задач и алгоритмы решения	<ul style="list-style-type: none"> • организационно-экономическая сущность задачи (наименование, цель решения, краткое содержание, метод, периодичность и время решения задачи, способы сбора и передачи данных, связь задачи с другими задачами, характер использования результатов решения, в которых они используются) • экономико-математическая модель задачи (структурная и развернутая форма представления) • входная оперативная информация (характеристика показателей, диапазон изменения, формы представления) • нормативно-справочная информация (НСИ) (содержание и формы представления) • информация, хранимая для связи с другими задачами • информация, накапливаемая для последующих решений данной задачи

Продолжение табл. 3.2

		<ul style="list-style-type: none"> • информация по внесению изменений (система внесения изменений и перечень информации, подвергающейся изменениям) • алгоритм решения задачи (последовательность этапов расчета, схема, расчетные формулы) • контрольный пример (набор заполненных данными форм входных документов, условные документы с накапливающейся и хранимой информацией, формы выходных документов, заполненные по результатам решения экономико-технической задачи и в соответствии с разработанным алгоритмом расчета)
4	Организация информационной базы	<ul style="list-style-type: none"> • источники поступления информации и способы ее передачи • совокупность показателей, используемых в системе • состав документов, сроки и периодичность их поступления • основные проектные решения по организации фонда НСИ • состав НСИ, включая перечень реквизитов, их определение, диапазон изменения и перечень документов НСИ • перечень массивов НСИ, их объем, порядок и частота корректировки информации • структура фонда НСИ с описанием связи между его элементами; требования к технологии создания и ведения фонда • методы хранения, поиска, внесения изменений и контроля • определение объемов и потоков информации НСИ • контрольный пример по внесению изменений в НСИ • предложения по унификации документации

Окончание табл. 3.2

5	Альбом форм документов	
6	Система математического обеспечения	<ul style="list-style-type: none"> • обоснование структуры математического обеспечения • обоснование выбора системы программирования • перечень стандартных программ
7	Принцип построения комплекса технических средств	<ul style="list-style-type: none"> • описание и обоснование схемы технологического процесса обработки данных • обоснование и выбор структуры комплекса технических средств и его функциональных групп • обоснование требований к разработке нестандартного оборудования • комплекс мероприятий по обеспечению надежности функционирования технических средств
8	Расчет экономической эффективности системы	<ul style="list-style-type: none"> • сводная смета затрат, связанных с эксплуатацией систем • расчет годовой экономической эффективности, источниками которой являются оптимизация производственной структуры хозяйства (объединения), снижение себестоимости продукции за счет рационального использования производственных ресурсов и уменьшения потерь, улучшения принимаемых управленческих решений
9	Мероприятия по подготовке объекта к внедрению системы	<ul style="list-style-type: none"> • перечень организационных мероприятий по совершенствованию бизнес-процессов • перечень работ по внедрению системы, которые необходимо выполнить на стадии рабочего проектирования, с указанием сроков и ответственных лиц
10	Ведомость документов	

В *завершение стадии технического проектирования* производится разработка документации на поставку серийно выпускаемых изделий для комплектования ИС, а также определяются технические требования и составляются ТЗ на разработку изделий, не изготавляемых серийно.

На стадии «рабочая документация» осуществляется создание программного продукта и разработка всей сопровождающей документации.

Документация должна содержать все необходимые и достаточные сведения для обеспечения выполнения работ по вводу ИС в действие и ее эксплуатации, а также для поддержания уровня эксплуатационных характеристик (качества) системы. Разработанная документация должна быть соответствующим образом оформлена, согласована и утверждена.

Для ИС, которые являются разновидностью автоматизированных систем, устанавливают следующие *основные виды испытаний*: *предварительные, опытная эксплуатация и приемочные*. При необходимости допускается дополнительно проведение других видов испытаний системы и ее частей.

В зависимости от взаимосвязей частей ИС и объекта автоматизации испытания могут быть автономными или комплексные. Автономные испытания охватывают части системы. Их проводят по мере готовности частей системы к сдаче в опытную эксплуатацию. Комплексные испытания проводят для групп взаимосвязанных частей или для системы в целом.

Для планирования проведения всех видов испытаний разрабатывается документ «Программа и методика испытаний». Разработчик документа устанавливается в договоре или ТЗ. В качестве приложения в документ могут включаться тесты или контрольные примеры.

Предварительные испытания проводят для определения работоспособности системы и решения вопроса о возможности ее приемки в опытную эксплуатацию.

Предварительные испытания следует выполнять после проведения разработчиком отладки и тестирования поставляемых программных и технических средств системы и представления им соответствующих документов об их готовности к испытаниям, а также после ознакомления персонала ИС с эксплуатационной документацией.

Опытную эксплуатацию системы проводят с целью определения фактических значений количественных и качественных характеристик системы и готовности персонала к работе в условиях ее функционирования, а также определения фактической эффективности и корректировки, при необходимости, документации.

Приемочные испытания проводят для определения соответствия системы техническому заданию, оценки качества опытной эксплуатации и решения вопроса о возможности приемки системы в постоянную эксплуатацию.

3. Типовое проектирование

Типовое проектирование ИС предполагает создание системы из готовых типовых элементов. Основополагающим требованием для применения методов типового проектирования является возможность декомпозиции, проектируемой ИС на множество составляющих компонентов (подсистем, комплексов задач, программных модулей и т. д.). Для реализации выделенных компонентов выбираются имеющиеся на рынке типовые проектные решения, которые настраиваются на особенности конкретного предприятия.

Типовое проектное решение (ТПР) – это тиражируемое (пригодное для многократного использования) проектное решение. Принятая классификация ТПР основана на уровне декомпозиции системы. Выделяются следующие классы ТПР:

- **элементные ТПР** – это типовые решения по задаче или по отдельному виду обеспечения задачи (информационному, программному, техническому, математическому, организационному);
- **подсистемные ТПР** – в которых в качестве элементов типизации выступают отдельные подсистемы, разработанные с учетом функциональной полноты и минимизации внешних информационных связей;
- **объектные ТПР** – это типовые отраслевые проекты, которые включают полный набор функциональных и обеспечивающих подсистем ИС.

Каждое типовое решение предполагает наличие, кроме собственно функциональных элементов (программных или аппаратных), документации с детальным описанием ТПР и процедур настройки в соответствии с требованиями разрабатываемой системы.

Для реализации типового проектирования используются два подхода: параметрически-ориентированное и модельно-ориентированное проектирование.

Параметрически-ориентированное проектирование включает следующие этапы: определение критериев оценки пригодности пакетов прикладных программ (ППП) для решения поставленных задач, анализ и оценка доступных ППП по сформулированным критериям, выбор и закупка наиболее подходящего пакета, настройка параметров (доработка) закупленного ППП.

Критерии оценки ППП делятся на следующие группы:

- назначение и возможности пакета;
- отличительные признаки и свойства пакета;
- требования к техническим и программным средствам;
- документация пакета;
- факторы финансового порядка;
- особенности эксплуатационного пакета;
- помошь поставщика по внедрению и поддержанию пакета;
- оценка качества пакета и опыт его использования;
- перспективы развития пакета.

Внутри каждой группы критериев выделяется некоторое подмножество частных показателей, детализирующих каждый из десяти выделенных аспектов анализа выбираемых ППП.

Числовые значения показателей для конкретных ППП устанавливается экспертами по выбранной шкале оценок (например, 10-балльной). На их основе формируются групповые оценки и комплексная оценка пакета (путем вычисления средневзвешенных значений). Нормированные взвешивающие коэффициенты также получаются эксперты путем.

Модельно-ориентированное проектирование заключается в адаптации состава и характеристик типовой ИС в соответствии с моделью объекта автоматизации. Технология проектирования в этом случае должна обеспечивать единые средства для работы как с моделью типовой ИС, так и с моделью конкретного предприятия.

Типовая ИС в специальной базе метаинформации – репозитарии – содержит модель объекта автоматизации, на основе которой осуществляется конфигурирование программного обеспечения. Таким образом, модельно-ориентированное проектирование ИС предполагает, прежде всего, построение модели объекта автоматизации с использованием специального программного инструментария (например, SAP Business Engineering Workbench (BEW), BAAN Enterprise Modeler). Возможно также создание системы на базе типовой модели ИС из репозитария, который поставляется вместе с программным продуктом и расширяется по мере накопления опыта проектирования информационных систем для различных отраслей и типов производства.

Репозитарий содержит базовую (ссылочную) модель ИС, типовые (референтные) модели определенных классов ИС, модели конкретных ИС предприятий.

Базовая модель ИС в репозитарии содержит описание бизнес-функций, бизнес-процессов, бизнес-объектов, бизнес-правил, организационной структуры, которые поддерживаются программными модулями типовой ИС. Типовые модели описывают конфигурации информационной системы для определенных отраслей или типов производства.

Модель конкретного предприятия строится либо путем выбора фрагментов основной или типовой модели в соответствии со специфическими особенностями предприятия (BAAN Enterprise Modeler), либо путем автоматизированной адаптации этих моделей в результате экспертного опроса (SAP Business Engineering Workbench).

Построенная модель предприятия в виде мета описания хранится в репозитарии и при необходимости может быть откорректирована. На основе этой модели автоматически осуществляется конфигурирование и настройка информационной системы.

Бизнес-правила определяют условия корректности совместного применения различных компонентов ИС и используются для поддержания целостности создаваемой системы.

Модель бизнес-функций представляет собой иерархическую декомпозицию функциональной деятельности предприятия.

Модель бизнес-процессов отражает выполнение работ для функций. Для отображения процессов используется модель управления событиями.

Именно модель бизнес-процессов позволяет выполнить настройку программных модулей – приложений информационной системы в соответствии с характерными особенностями конкретного предприятия.

Модели бизнес-объектов используются для интеграции приложений, поддерживающих исполнение различных бизнес-процессов.

Модель организационной структуры предприятия представляет собой традиционную иерархическую структуру подчинения подразделений и персонала.

Внедрение типовой информационной системы начинается с анализа требований к конкретной ИС, которые выявляются на основе результатов предпроектного обследования объекта автоматизации.

Для оценки соответствия этим требованиям программных продуктов может использоваться описанная выше методика оценки ППП. После выбора программного продукта на базе имеющихся в нем референтных моделей строится предварительная модель ИС, в которой отражаются все особенности реализации ИС для конкретного предприятия. Предварительная модель является основой для выбора типовой модели системы и определения перечня компонентов, которые будут реализованы с использованием других программных средств или потребуют разработки с помощью имеющихся в составе типовой ИС инструментальных средств.

Реализация типового проекта предусматривает выполнение следующих операций:

- установка глобальных параметров системы;
- задание структуры объекта автоматизации;
- определение структуры основных данных;
- задание перечня реализуемых функций и процессов;
- описание интерфейсов;
- описание отчетов;
- настройка авторизации доступа;
- настройка системы архивирования.

ЛЕКЦИЯ № 4. АНАЛИЗ И МОДЕЛИРОВАНИЕ ФУНКЦИОНАЛЬНОЙ ОБЛАСТИ ВНЕДРЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

- 1. Полная бизнес-модель компании.**
- 2. Шаблоны организационного бизнес-моделирования.**
- 3. Построение организационно-функциональной модели компании.**

1. Полная бизнес-модель компании

На практике выработан ряд подходов к проведению организационного анализа, но наибольшее распространение получил **инженерный подход**. Организационный анализ компании при таком подходе проводится по определенной схеме с помощью полной бизнес-модели компании. Компания рассматривается как целевая, открытая, социально-экономическая система, принадлежащая иерархической совокупности открытых внешних надсистем (рынок, государственные учреждения, и пр.) и внутренних подсистем (отделы,

цеха, бригады и пр.). Возможности компании определяются характеристиками ее структурных подразделений и организацией их взаимодействия.

На рисунке 4.1 представлена обобщенная схема организационного бизнес-моделирования.

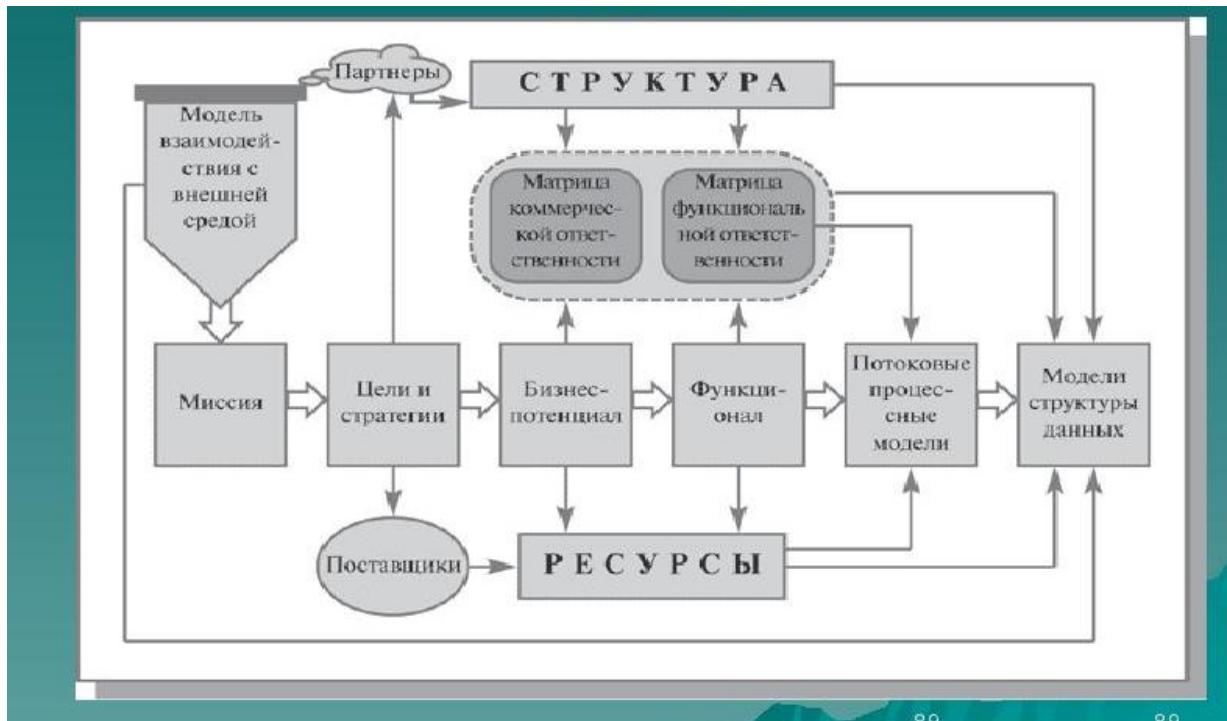


Рисунок 4.1 – Обобщенная схема организационного бизнес-моделирования

Построение бизнес-модели компании начинается с описания модели взаимодействия с внешней средой по закону единства и борьбы противоположностей, то есть с определения миссии компании.

Миссия компании, согласно стандарту ISO-15704 – это:

1) деятельность, осуществляемая предприятием для того, чтобы выполнить функцию, для которой оно было учреждено, – предоставление заказчику продукта или услуги;

2) механизм, с помощью которого предприятие реализует свои цели и задачи.

Миссия компании по удовлетворению социально-значимых потребностей рынка определяется как компромисс интересов рынка и компании. При этом миссия как атрибут открытой системы разрабатывается, с одной стороны, исходя из рыночной конъюнктуры и позиционирования компании относительно остальных участников внешней среды, а с другой – исходя из объективных возможностей компании и ее субъективных ценностей, ожиданий и принципов. Миссия является своеобразной мерой устремлений компании и в частности определяет рыночные претензии компании (предмет конкурентной борьбы). Определение миссии позволяет сформировать **дерево целей компании** – иерархические списки уточнения и детализации миссии.

Дерево целей формирует дерево стратегий – *иерархические списки уточнения и детализации достижения целей*. При этом на корпоративном уровне разрабатываются стратегии роста, интеграции и инвестиции бизнесов. Блок бизнес-стратегий определяет продуктовые и конкурентные стратегии, а также стратегии привлечения материальных, финансовых, человеческих и информационных ресурсов. Функциональные стратегии определяют стратегии в организации компонентов управления и этапов жизненного цикла продукции. Одновременно выясняется потребность и предмет партнерских отношений (субподряд, сервисные услуги, продвижение и пр.).

Это позволяет обеспечить заказчикам необходимый продукт требуемого качества, в нужном количестве, в нужном месте, в нужное время и по приемлемой цене. При этом компания может занять в партнерской цепочке создаваемых ценностей оптимальное место, где ее возможности и потенциал будут использоваться наилучшим образом.

Это дает возможность сформировать *бизнес-потенциал компании – набор видов коммерческой деятельности, направленный на удовлетворение потребностей конкретных сегментов рынка*.

Далее, исходя из специфики каналов сбыта, формируется первоначальное представление об организационной структуре (определяются центры коммерческой ответственности). Возникает понимание основных ресурсов, необходимых для воспроизведения товарной номенклатуры.

Бизнес-потенциал, в свою очередь, определяет *функционал компании – перечень бизнес-функций, функций менеджмента и функций обеспечения, требуемых для поддержания на регулярной основе указанных видов коммерческой деятельности*. Кроме того, уточняются необходимые для этого ресурсы (материальные, человеческие, информационные) и структура компании.

Построение бизнес-потенциала и функционала компании позволяет с помощью матрицы проекций определить зоны ответственности менеджмента.

Матрица проекций – это модель, представленная в виде матрицы, задающей систему отношений между классификаторами в любой их комбинации.

Матрица коммерческой ответственности закрепляет ответственность структурных подразделений за получение дохода в компании от реализации коммерческой деятельности. Ее дальнейшая детализация (путем выделения центров финансовой ответственности) обеспечивает построение финансовой модели компании, что, в свою очередь, позволяет внедрить систему бюджетного управления.

Матрица функциональной ответственности закрепляет ответственность структурных звеньев (и отдельных специалистов) за выполнение бизнес-функций при реализации процессов коммерческой (закупка, производство, сбыт и пр.), а также функций менеджмента, связанных с управлением этими процессами (планирование, учет, контроль в области маркетинга, финансов, управления персоналом и пр.). Дальнейшая детализация матрицы (до уровня ответственности сотрудников) позволит получить

функциональные обязанности персонала, что в совокупности с описанием прав, обязанностей, полномочий обеспечит разработку пакета должностных инструкций.

Описание бизнес-потенциала, функционала и соответствующих матриц ответственности представляет собой статическое описание компании. При этом процессы, протекающие в компании пока в свернутом виде (как функции), идентифицируются, классифицируются и, что особенно важно, закрепляются за исполнителями (будущими хозяевами этих процессов).

На этом этапе бизнес-моделирования формируется общепризнанный набор основополагающих внутрифирменных регламентов:

- базовое положение об организационно-функциональной структуре компании;
- пакет положений об отдельных видах деятельности (финансовой, маркетинговой, и т. п.);
- пакет положений о структурных подразделениях (цехах, отделах, секторах, группах и т. п.);
- должностные инструкции.

Это вносит прозрачность в деятельность компании за счет четкого разграничения и документального закрепления зон ответственности менеджеров.

Дальнейшее развитие (детализация) бизнес-модели происходит на этапе динамического описания компании на уровне процессных потоковых моделей.

Процессные потоковые модели – это модели, описывающие процесс последовательного во времени преобразования материальных и информационных потоков компании в ходе реализации какой-либо бизнес-функции или функции менеджмента. Сначала (на верхнем уровне) описывается логика взаимодействия участников процесса, а затем (на нижнем уровне) – технологии работы отдельных специалистов на своих рабочих местах.

Завершается организационное бизнес-моделирование *разработкой модели структур данных*, которая определяет перечень и форматы документов, сопровождающих процессы в компании, а также задает форматы описания объектов внешней среды, компонентов и регламентов самой компании. При этом создается система справочников, на основании которых получают пакеты необходимых документов и отчетов.

Такой подход позволяет описать деятельность компании с помощью универсального множества управлеченческих регистров (цели, стратегии, продукты, функции, организационные звенья и др.).

Управленческие регистры по своей структуре представляют собой иерархические классификаторы. Объединяя классификаторы в функциональные группы и закрепляя между собой элементы различных классификаторов с помощью матричных проекций, можно получить полную бизнес-модель компании.

При этом происходит процессно-целевое описание компании, позволяющее получить взаимосвязанные ответы на следующие вопросы: **зачем-что-где-кто-как-когда-кому-сколько** (рисунок 4.2).

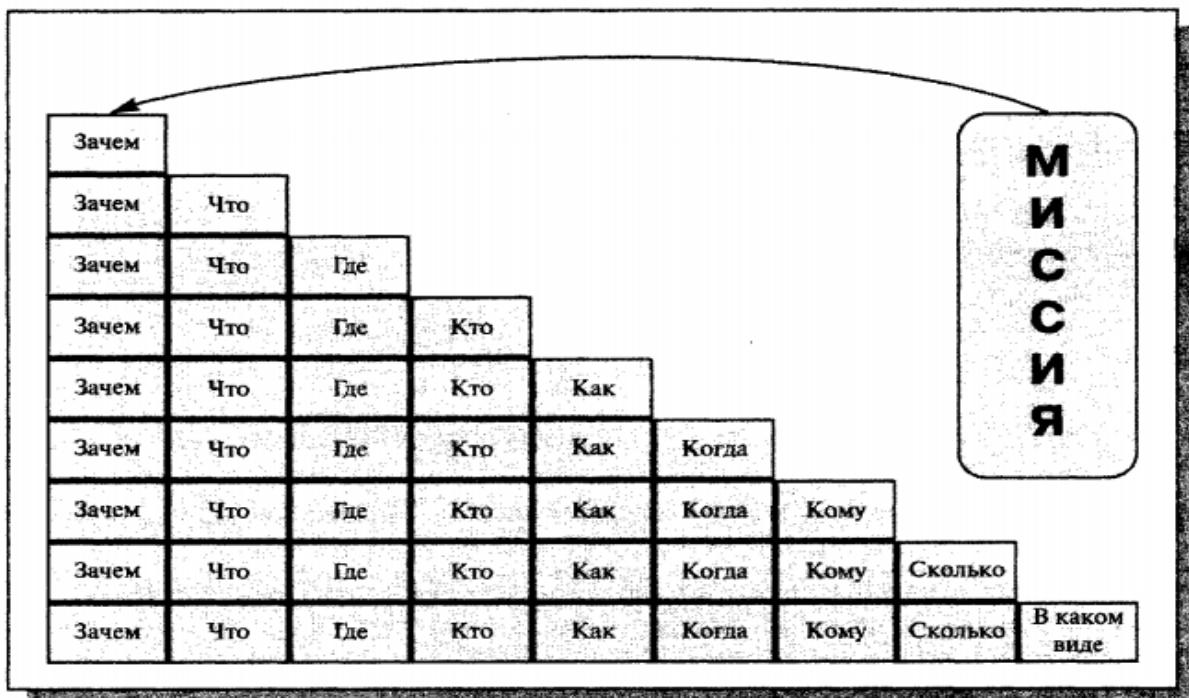


Рисунок 4.2 – Основные этапы процессно-целевого описания компании

Следовательно, **полная бизнес-модель компании** – это совокупность функционально ориентированных информационных моделей, обеспечивающая взаимосвязанные ответы на следующие вопросы: **зачем-что-где-кто-как-когда-кому-сколько** (рисунок 4.3).

Таким образом, **организационный анализ** предполагает построение комплекса взаимосвязанных информационных моделей компании, который включает:

- **стратегическую модель целеполагания** (отвечает на вопросы: зачем компания занимается именно этим бизнесом, почему предполагает быть конкурентно-способной, какие цели и стратегии для этого необходимо реализовать);
- **организационно-функциональную модель** (отвечает на вопросы *кто-что делает в компании и кто за что отвечает*);
- **функционально-технологическую модель** (отвечает на вопрос *что и как реализуется в компании*);
- **процессно-ролевую модель** (отвечает на вопрос *кто-что-как-кому*);
- **количественную модель** (отвечает на вопрос *сколько необходимо ресурсов*);
- **модель структуры данных** (отвечает на вопрос *в каком виде описываются регламенты компании и объекты внешнего окружения*).

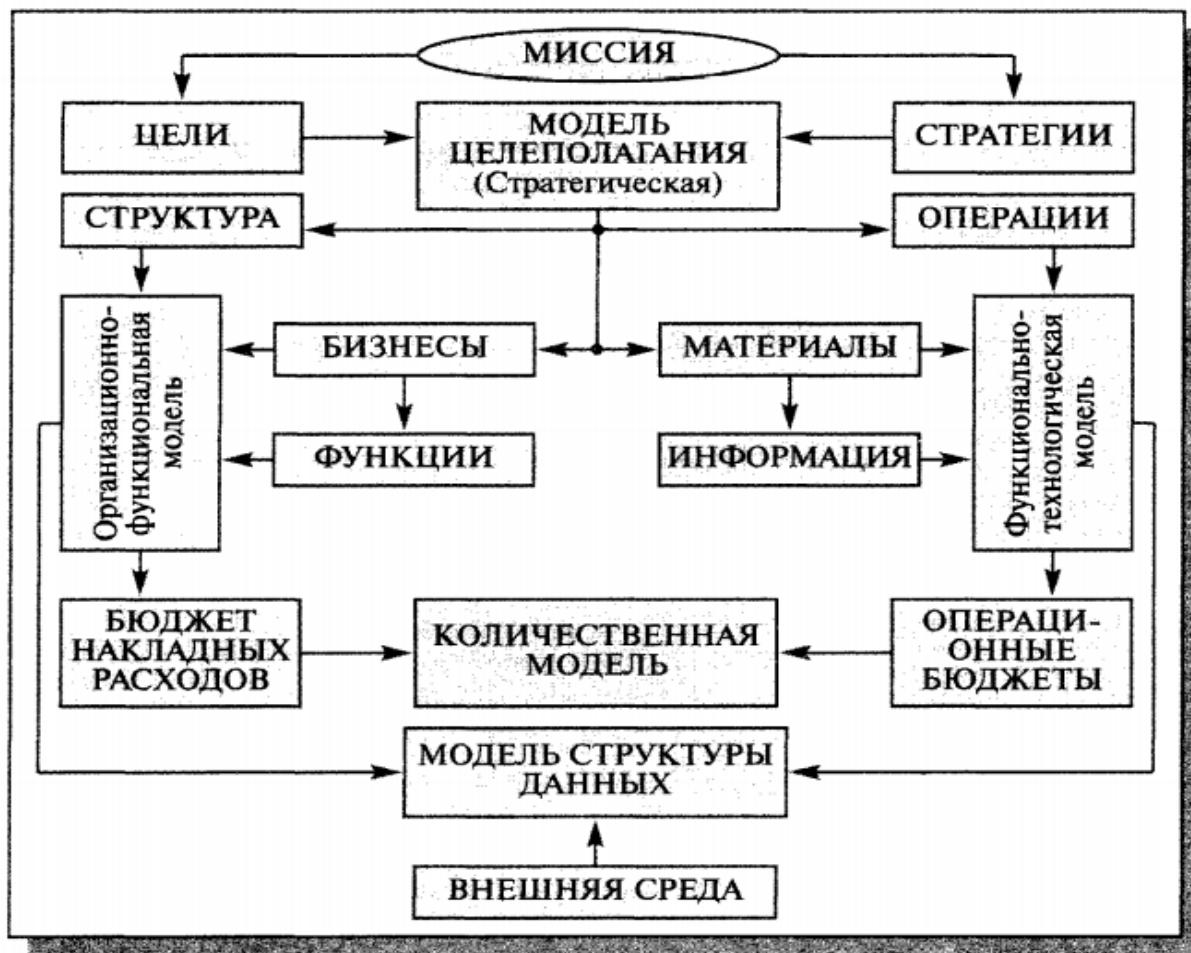


Рисунок 4.3 – Полная бизнес-модель компании

Представленная совокупность моделей обеспечивает необходимую полноту и точность описания компании и позволяет вырабатывать понятные требования к проектируемой информационной системе.

2. Шаблоны организационного бизнес-моделирования

Технология организационного бизнес-моделирования предполагает использование типовых шаблонных техник описания компаний.

Шаблон разработки миссии. Как отмечалось ранее, любая компания с ее микро- и макроокружением представляет собой иерархию вложенных друг в друга открытых субъективно-ориентированных систем. Компания, с одной стороны, является частью рынка, а с другой, отстаивает в конкурентной борьбе собственные интересы. Миссия представляет собой результат позиционирования компании среди других участников рынка. Поэтому миссию компании нельзя описывать путем анализа ее внутреннего устройства. Для построения модели взаимодействия компании с внешней средой (определение миссии компании на рынке) необходимо:

- идентифицировать рынок (надсистему), частью которого является компания;

- определить свойства (потребности) рынка;
- определить предназначение (миссию) компании, исходя из ее роли на рынке.

Кроме этого, миссия, как было сказано выше, это компромисс между потребностями рынка, с одной стороны, возможностями и желанием компании удовлетворить эти интересы, с другой. Поиск компромиссов может быть выполнен по шаблону, представленному на рис. 4.4.

		надо				
		рыночная конъюнктура	внешняя среда			
			Политика	Экономика	Социал. сфера	Технология
объект	Уникальность технологий					
	Исключительность ресурсов					
	Знания и умения					
субъект	Ценности и ожидания					

МИССИЯ

Рисунок 4.4 – Шаблон разработки миссии (матрица проекций)

При разработке модели миссии компании рекомендуется:

1. **Описать базис конкурентоспособности компании** – совокупность характеристик как социально-экономической системы.

Например:

- для объекта – уникальность освоенных технологий и исключительность имеющихся в компании ресурсов (финансовых, материальных, информационных и др.);
- для субъекта – знания и умения персонала и опыт менеджеров.

Это определяет уникальность ресурсов и навыков компании и формирует позиции «могу».

2. **Выяснить конъюнктуру рынка**, т. е. определить наличие платежеспособного спроса на предлагаемые товары или услуги и степень удовлетворения рынка конкурентами. Это позволяет понять потребности рынка и сформировать позицию «надо».
3. **Выявить наличие способствующих и противодействующих факторов для выбранного вида деятельности со стороны государственных институтов в области политики и экономики.**
4. **Оценить перспективу развития технологии в выбранной сфере деятельности.**

5. *Оценить возможную поддержку или противодействие общественных организаций.*
6. *Сопоставить результаты вышеперечисленных действий с учетом правовых, моральных, этических и других ограничений со стороны персонала и сформировать позицию «хочу».*
7. *Оценить уровень возможных затрат и доходов.*
8. *Оценить возможность достижения приемлемого для всех сторон компромисса и сформулировать Миссию компании в соответствии с шаблоном (рисунок 4.5).*

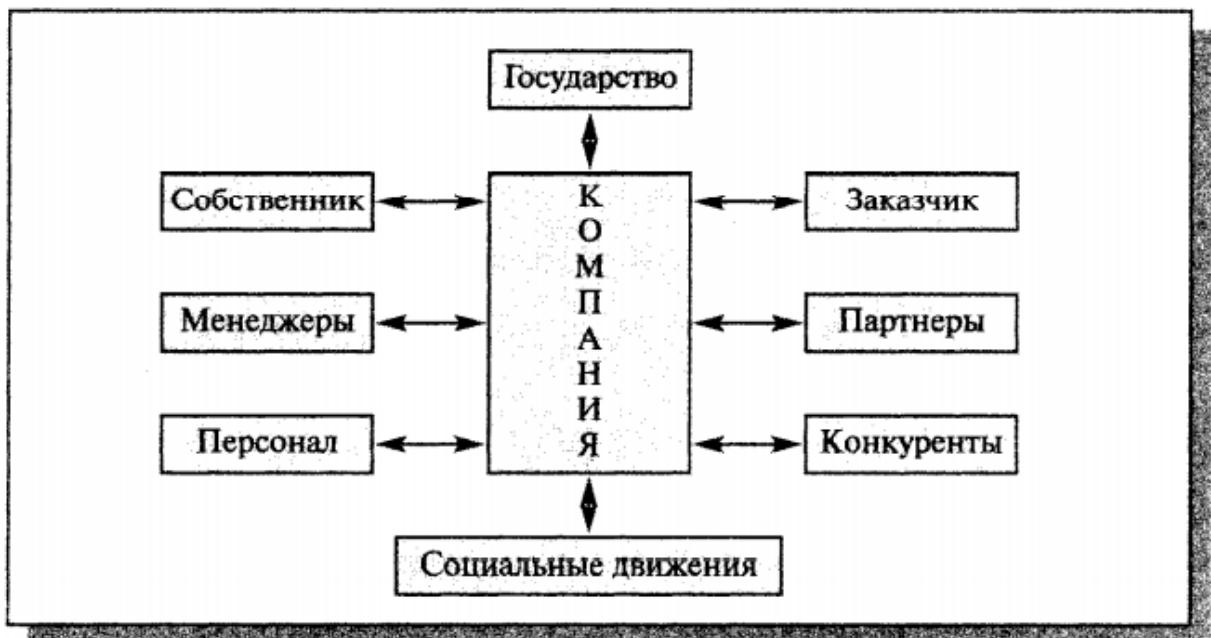


Рисунок 4.5 – Шаблон разработки миссии

Миссия в широком понимании представляет собой основную деловую концепцию компании, изложенную в виде восьми положений, определяющих взаимоотношения компании с другими субъектами:

- что получит заказчик в части удовлетворения своих потребностей;
- кто, для чего и как может выступать в качестве партнера компании;
- на какой основе предполагается строить отношения с конкурентами (какова, в частности готовность пойти на временные компромиссы);
- что получит собственник и акционеры от бизнеса;
- что получат от бизнеса компании менеджеры;
- что получит от компании персонал;
- в чем заключается сотрудничество с общественными организациями;
- как будут строиться отношения компании с государством (в частности, возможное участие в поддержке государственных программ).

Шаблон формирования бизнесов. В соответствии с разработанной Миссией компании определяются социально значимые потребности, на удовлетворение которых направлен бизнес компании. Разработка бизнес-потенциала компании может быть выполнена по шаблону формирования бизнесов, представленному на рисунке 4.6.

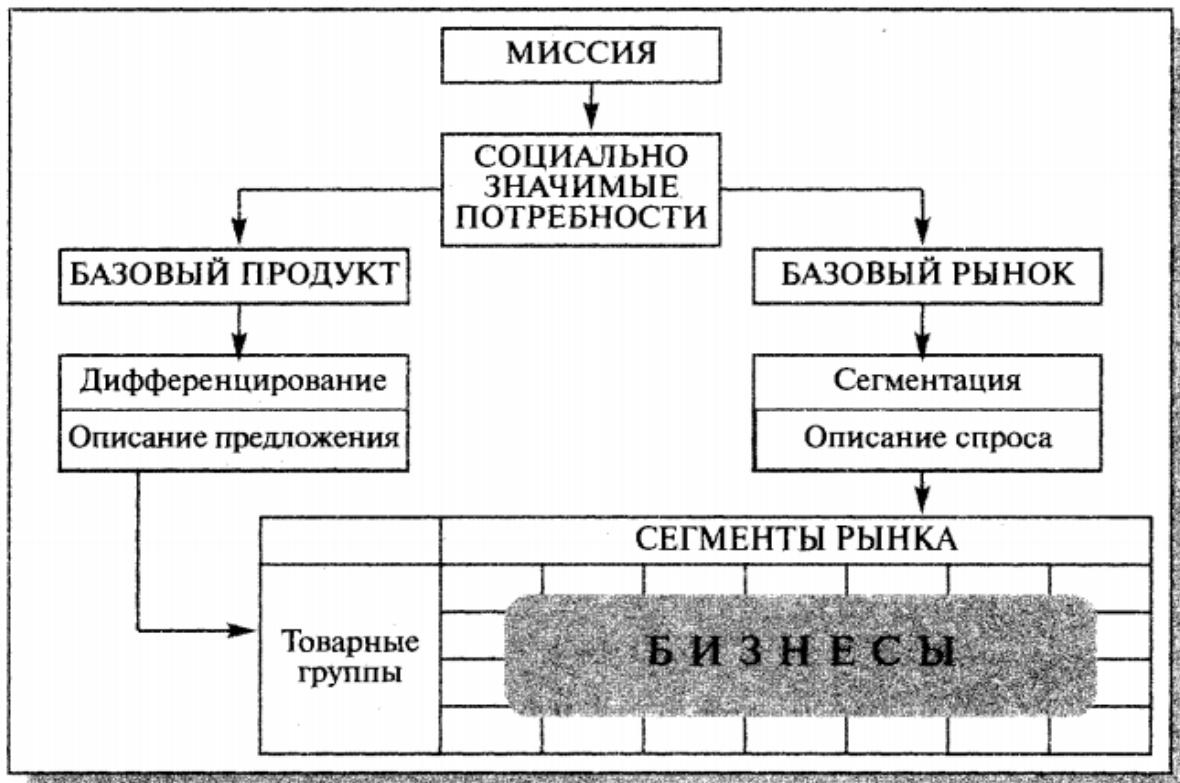


Рисунок 4.6 – Шаблон формирования бизнесов

В результате формируется базовый рынок и базовый продукт, детализация которых определяет предложения компании глазами покупателей (товарные группы) и однородные по отношению к продуктам компании группы покупателей (сегменты рынка). С помощью матричной проекции (рисунок 4.7) устанавливается соответствие между сформированными товарными группами и сегментами рынка и определяется список бизнесов компании (на пересечении строк и столбцов находятся бизнесы компании).

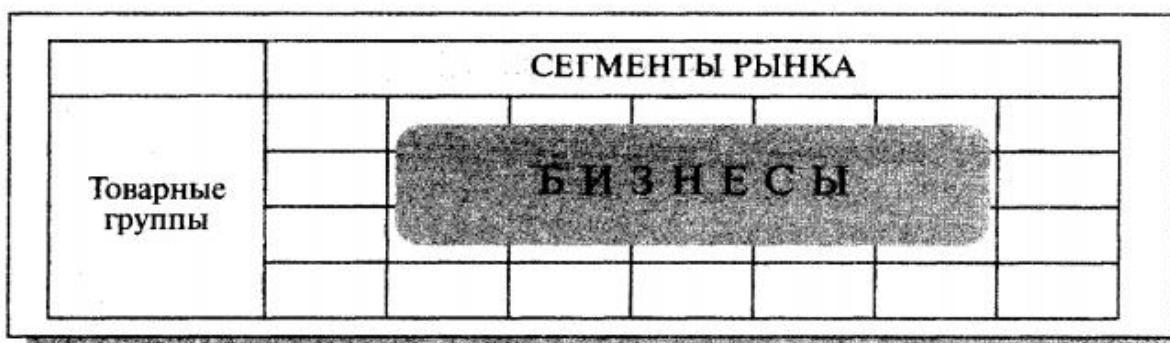


Рисунок 4.7 – Шаблон формирования бизнесов (матриц проекции)

Шаблон формирования функционала компании (основных бизнес-функций). На основании списка бизнесов, с помощью матричной проекции (рисунок 4.8) формируется классификатор бизнес-функций компании.

ЭТАПЫ ПРОИЗВОДСТВЕННОГО ЦИКЛА	БИЗНЕСЫ		
	№1	№2	№3
	Проектирование		
Закупки			
Производство			
Распределение			
Сбыт			
Сопровождение			

БИЗНЕС-ФУНКЦИИ (ОСНОВНЫЕ)

Рисунок 4.8 – Шаблон формирования основных бизнес-функций

Далее аналогично, с помощью матрицы проекций, формируется список основных функций менеджмента. На рисунке 4.9 приведены примеры классификаторов, на основании которых построена матрица – генератор основных функций менеджмента.

Этапы управленческого цикла	Компоненты менеджмента	Структуры	Логистика	Финансы	Экономика	Учет	Маркетинг	Персонал
	Сбор информации							
Выработка решений								
Реализация								
Учет								
Контроль								
Анализ								
Регулирование								

ФУНКЦИИ МЕНЕДЖМЕНТА (основные)

Рисунок 4.9 – Шаблон формирования основных функций менеджмента

Для формирования основных функций менеджмента компании сначала разрабатываются и утверждаются два базовых классификатора – «Компоненты

менеджмента» (перечень используемых на предприятии инструментов/ контуров управления) и «Этапы управленческого цикла» (технологическая цепочка операций, последовательно реализуемая менеджерами при организации работ в любом контуре управления).

Представленные матричные проекции (рисунки 4.8, 4.9) позволяют формировать функции любой степени детализации путем более подробного описания как строк, так и столбцов матрицы.

Шаблон формирования зон ответственности за функционал компании. Формирование зон ответственности за функционал компаний выполняется с помощью матрицы организационных проекций (рисунок 4.10).

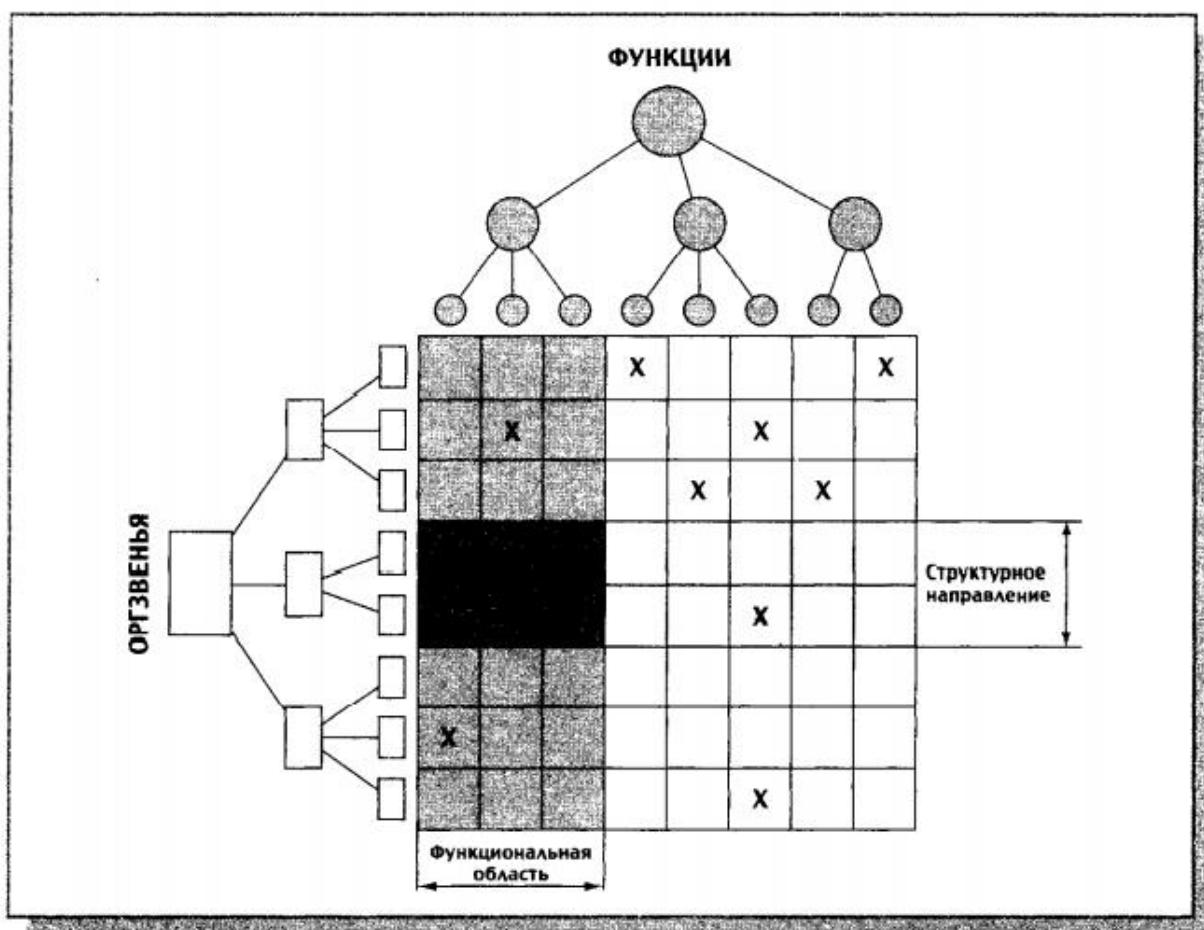


Рисунок 4.10 – Шаблон распределения функций по организационным звеньям

Матрица организационных проекций представляет собой таблицу, в строках которой расположен список исполнительных звеньев, в столбцах – список функций, выполняемых в компании. Для каждой функции определяется исполнительное лицо, отвечающее за эту функцию.

Заполнение такой таблицы позволяет по каждой функции найти исполняющие ее подразделения или сотрудника. Анализ заполненной таблицы позволяет увидеть «пробелы» как в исполнении функций, так и в загруженности сотрудников, а также рационально перераспределять все задачи между

исполнителями и закрепить как систему в документе «Положение об организационной структуре».

Положение об организационной структуре – это внутрифирменный документ, фиксирующий продукты и услуги компании, функции, выполняемые в компании, исполнительные звенья, реализующие функции, распределение функций по звеньям.

Таблица проекций функций на исполнительные звенья может иметь весьма большую размерность. В средних компаниях это, например, 500 единиц (20 звеньев на 25 функций). В больших компаниях это может быть 5000 единиц (50 звеньев на 100 функций).

Аналогично строится **матрица коммерческой ответственности**.

Шаблон потокового процессного описания. Шаблон потокового процессного описания приведен на рисунке 4.11. Такое описание дает представление о процессе последовательного преобразования ресурсов в продукты усилиями различных исполнителей на основании соответствующих регламентов.

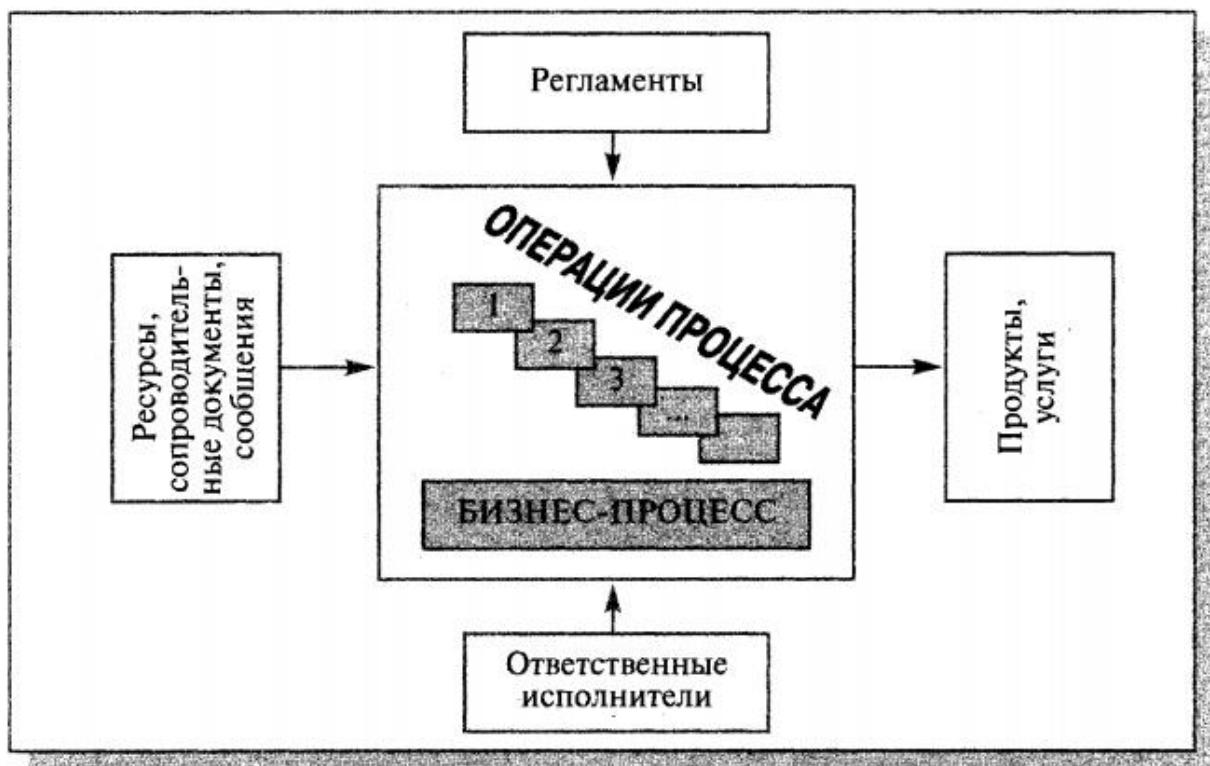


Рисунок 4.11 – Потоковая процессная модель

3. Построение организационно-функциональной модели компании

Организационно-функциональная модель компании строится на основе функциональной схемы деятельности компании (рисунок 4.12).

На основании миссии формируются цели и стратегии компании. С их помощью определяется необходимый набор продуктов и как следствие, требуемые ресурсы. Воспроизводство продукции происходит за счет

переработки ресурсов в основном производственном цикле. Его компоненты формируют необходимые бизнес-функции для поставки ресурсов, производства продуктов и их распределения в места реализации.

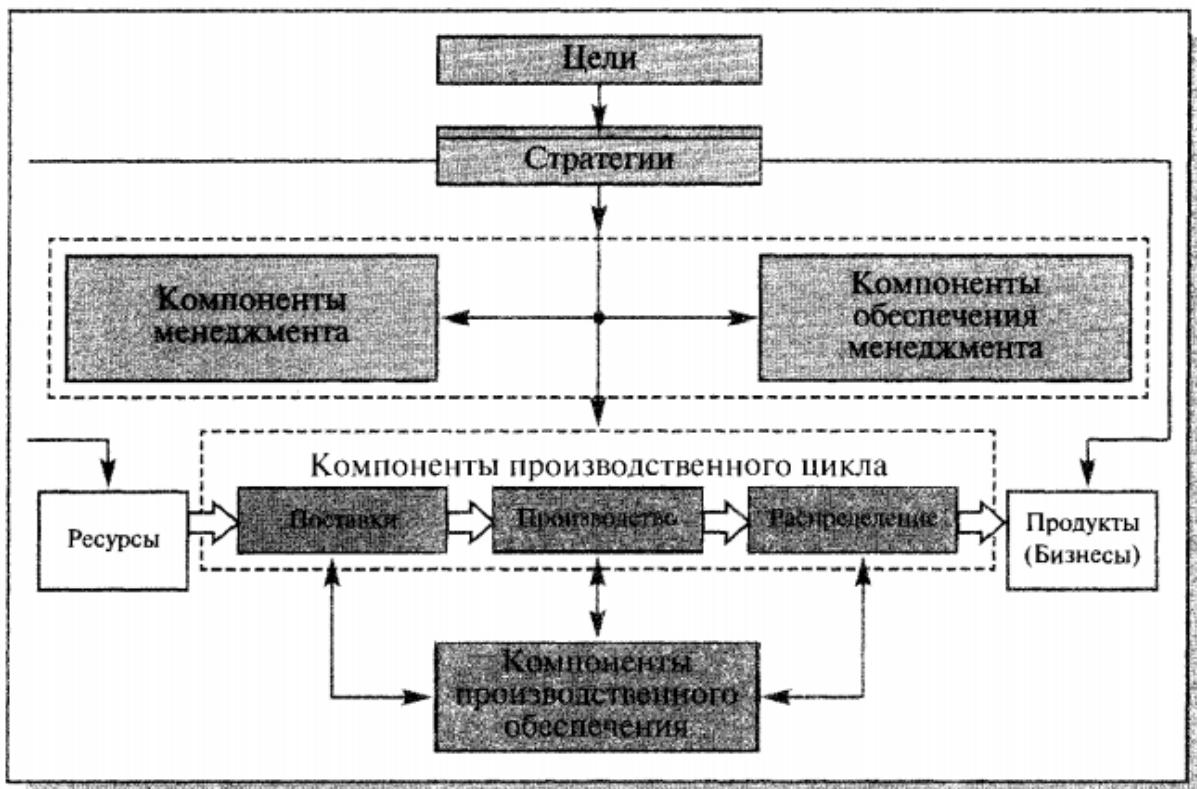


Рисунок 4.12 – Функциональная схема компании

Для управления указанным процессом воспроизведения формируется совокупность компонентов менеджмента, которая порождает набор функций управления. Для поддержания процессов воспроизведения и управления формируются наборы соответствующих функций обеспечения (охраны, технического оснащения, профилактики и ремонта и пр.).

Управленческие регистры представляют собой иерархические классификаторы. Объединяя классификаторы в функциональные группы и закрепляя между собой элементы различных классификаторов с помощью матричных проекций, можно получить модель организационной структуры компании.

Для построения организационно-функциональной модели используется всего два типа элементарных моделей.

Древовидные модели (классификаторы) – точные иерархические списки выделенных объектов управления (организационных звеньев, функций, ресурсов, в том числе исполнительных механизмов для бизнес-процессов, документов и их структуры, и т. п.). Каждый элемент классификатора может быть дополнительно охарактеризован рядом атрибутов: тип, шкала, комментарий и т. п. Фактически классификаторы представляют собой набор управленческих регистров, содержащих, в основном, неколичественную

информацию, совокупность которых задает систему координат для описания деятельности компании. Количество таких списков-классификаторов определяется целью построения модели.

Матричные модели – это проекции, задающие систему отношений между классификаторами в любой их комбинации. Связи могут иметь дополнительные атрибуты (направление, название, индекс, шкала и вес).

В начальной модели применяется всего несколько классификаторов предметной области:

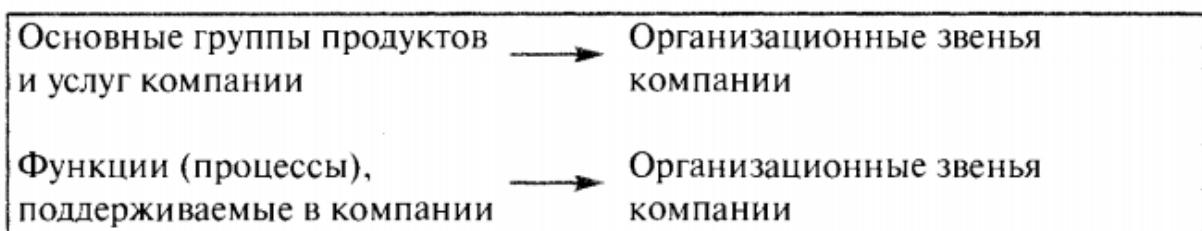
- основные группы продуктов и услуг компании;
- ресурсы, потребляемые компанией в ходе своей деятельности;
- функции (процессы), поддерживаемые в компании;
- организационные звенья компании.

В классификаторе функций обычно выделяют три базовые раздела:

- основные функции – непосредственно связанные с процессом преобразования внешних ресурсов в продукцию и услуги предприятием;
- функции менеджмента – или функции управления предприятием;
- функции обеспечения – поддерживающие производственную, коммерческую и управлеченческую деятельность.

Главной функцией компании является предоставление продуктов и услуг, поэтому сначала производится формальное описание, согласование и утверждение руководством предприятия перечня его бизнесов (направлений коммерческой деятельности), продукции и услуг. Из этого классификатора внешним контрагентам должно быть понятно, чем предприятие интересно рынку, а для внутренних целей – для чего нужен тот или иной функционал компании.

В результате этих операций производится идентификация функционала и создается единая терминология описания функций предприятия, которая должна быть согласована всеми ведущими менеджерами. При составлении классификатора организаций важно, чтобы уровень детализации функций соответствовал уровню детализации звеньев. После формирования всех базовых классификаторов с помощью матричных проекций производится их закрепление за организационными единицами предприятия:



Процесс формирования матрицы проекций функций на организацию на практике напоминает игру в крестики-нолики (рисунок 4.10).

По строчкам таблицы указываются подразделения, по столбцам – функции, составляющие содержание процесса управления или бизнес-процесса в данной компании. На пересечениях функций и подразделений, которые ответственны за выполнение функций, ставится крестик. Для проекций большой

размерности используется механизм расстановки связей между двумя классификаторами, представленных списками.

Стандартная практика построения моделей организационно-функциональной структуры компании поддерживает два уровня детализации:

- 1) агрегированная модель;
- 2) детализированная модель.

Агрегированная модель – модель организационной структуры, учетные регистры которой имеют ограничение по степени детализации до 2-3 уровней.

Целью построения данной модели является предоставление информации об организационной структуре высшим руководителям компании для проведения стратегического анализа, анализа соответствия данной структуры стратегии и внешнему окружению банка.

Модель может также предоставляться внешним пользователям (например, потенциальным инвесторам как иллюстрация к бизнес-плану, крупным клиентам и др.).

Детализированная модель – модель организационной структуры, детализация учетных регистров которой производится на более глубоких уровнях, чем в агрегированной модели. Степень детализации в модели обусловлена конкретными потребностями компании (создание определенных организационных регламентов).

Целью построения данной модели является предоставление информации о распределении функциональных обязанностей между подразделениями компании, а также об организации бизнес-процессов в компании. Построение детализированной модели позволяет создавать различные внутрифирменные регламенты: Положения об организационной структуре (рисунок 4.13).

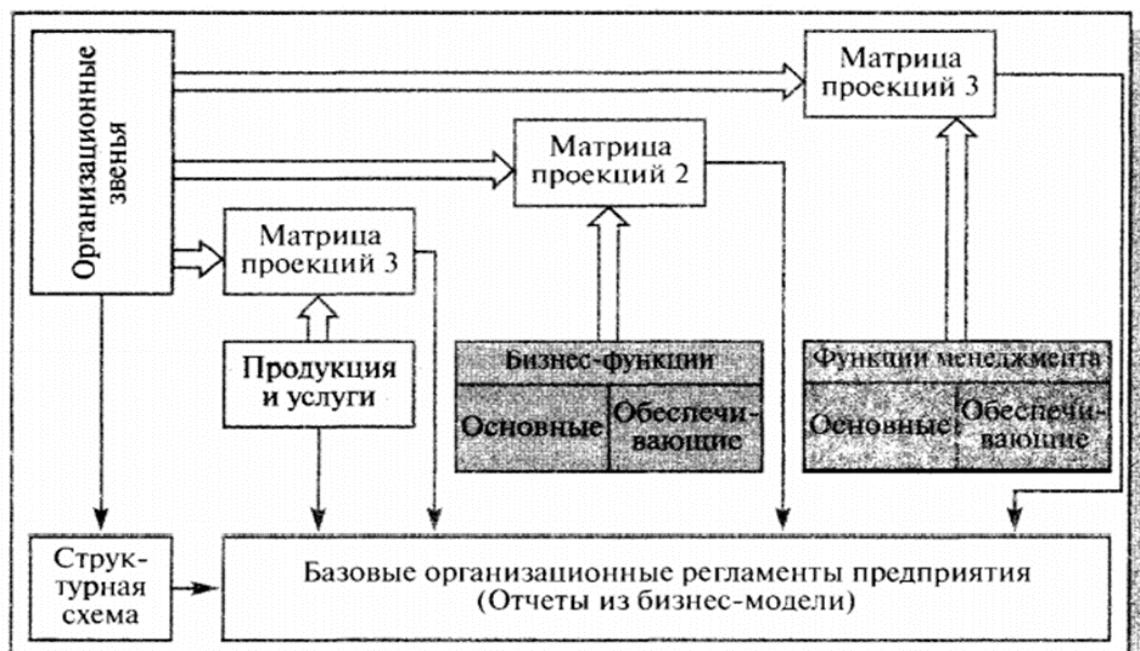


Рисунок 4.13 – Схема создания Положения об организационно-функциональной структуре компании

ЛЕКЦИЯ № 5. ДИНАМИЧЕСКОЕ ОПИСАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

1. Процессные потоковые модели.
2. Основные элементы процессного подхода.
3. Выделение и классификация процессов.

1. Процессные потоковые модели

Разработка требований к проектируемой ИС строится на основе статического и динамического описания компании. Статическое описание компании, рассмотренное в лекции 4, проводится на уровне функциональных моделей и включает описание бизнес-потенциала, функционала и соответствующих матриц ответственности.

Дальнейшее развитие (детализация) бизнес-модели происходит на этапе динамического описания компании на уровне процессных потоковых моделей.

Процессные потоковые модели – это модели, *описывающие процесс последовательного во времени преобразования материальных и информационных потоков компании в ходе реализации какой-либо бизнес-функции или функции менеджмента*. На верхнем уровне описывается логика взаимодействия участников процесса, на нижнем – технология работы отдельных специалистов на своих рабочих местах. Процессные потоковые модели отвечают на вопросы *кто-что-как-кому*.

Современное состояние экономики характеризуется переходом от традиционной функциональной модели деятельности компании, построенной на принципах разделения труда, узкой специализации и жестких иерархических структурах, к модели процессной, основанной на интеграции работ вокруг бизнес-процессов.

Главными недостатками функционального подхода являются:

- разбиение технологий выполнения работы на отдельные фрагменты, иногда между собой несвязанные, которые выполняются различными структурными подразделениями;
- отсутствие целостного описания технологий выполнения работы;
- сложность увязывания простейших задач в технологию, производящую реальный товар или услугу;
- отсутствие ответственности за конечный результат;
- высокие затраты на согласование, налаживание взаимодействия, контроль и т. д.;
- отсутствие ориентации на клиента.

Процессный подход предполагает смещение акцентов от управления отдельными структурными элементами к управлению сквозными бизнес-процессами, связывающими деятельность всех структурных элементов. Каждый деловой процесс проходит через ряд подразделений, т. е. в его выполнении участвуют специалисты различных отделов компании. Чаще всего

приходится сталкиваться с ситуацией, когда собственно процессами никто не управляет, а управляют лишь подразделениями. Более того, структура компаний строится без учета возможностей оптимизации деловых процессов, обеспечивающих необходимые функции. Процессный подход позволяет устранить фрагментарность в работе, организационные и информационные разрывы, дублирование, нерациональное использование финансовых, материальных и кадровых ресурсов.

Процессный подход к организации деятельности предприятия предполагает:

- широкое делегирование полномочий и ответственности исполнителям;
- сокращение уровней принятия решений;
- сочетание принципа целевого управления с групповой организацией труда;
- повышение внимания к вопросам обеспечения качества;
- автоматизацию технологий выполнения бизнес-процессов.

Согласно стандарту «Основные положения и словарь – ИСО/ОПМС 9000:2000» (п. 2.4) понятие **«Процессный подход»** определяется следующим образом: **любая деятельность, в которой используются ресурсы для преобразования входов в выходы, может рассматриваться как процесс.**

Для того, чтобы результативно функционировать, организации должны определять и управлять многочисленными взаимосвязанными и взаимодействующими процессами. Часто выход одного процесса образует непосредственно вход следующего. **Систематическая идентификация и менеджмент применяемых организацией процессов, и особенно взаимодействия таких процессов, могут считаться «процессным подходом».**

Основной принцип системного подхода определяет *структурирование бизнес-системы в соответствии с деятельностью и бизнес-процессами предприятия, а не в соответствии с его организационно-штатной структурой.* Именно бизнес-процессы, обеспечивающие значимый для потребителя результат, представляют ценность и для специалистов, проектирующих ИС.

Процессная модель компании должна строиться с учетом следующих положений:

1. Верхний уровень модели должен отражать только контекст диаграммы – взаимодействие моделируемого единственным контекстным процессом предприятия с внешним миром.
2. На втором уровне должны быть отражены тематически сгруппированные бизнес-процессы предприятия и их взаимосвязи.
3. Каждая из деятельности должна быть детализирована на бизнес-процессы.
4. Детализация бизнес-процессов осуществляется посредством бизнес-функций.
5. Описание элементарной бизнес-операции осуществляется с помощью мини-спецификации.

Процессный подход требует комплексного изучения различных сторон жизни организации – правовых основ и правил деятельности, организационной структуры, функций и показателей результатов их исполнения, интерфейсов ресурсного обеспечения, организационной культуры. В результате анализа создается модель деятельности «как есть». Обработка этой модели с помощью различных аналитических методов позволяет проверить, насколько деловые процессы рациональны, а также определить, является ли та или иная операция ориентированной на общественно значимый конечный результат или излишней бюрократической процедурой.

В ходе анализа деловых процессов детально исследуются сферы ответственности подразделений ведомства, его руководителей и сотрудников. Это позволяет установить адреса владельцев деловых процессов, в результате чего процессы перестают быть бесхозными, создаются условия для разработки и внедрения систем стимулирования и ответственности за конечные результаты, определяются моменты и процедуры передачи ответственности. Анализ и оценка деловых процессов позволяют подойти к обоснованию стандартов их выполнения, допустимых рисков и диапазонов свободы принятия решений исполнителями, предельных нормативов затрат ресурсов на единицу измерения эффекта.

Однако чисто «процессная компания» является скорее иллюстрацией правильной организации работ. В действительности все бизнес-процессы компании протекают в рамках организационной структуры предприятия, описывающей функциональные компетентности и отношения.

Управление всей текущей деятельностью компании ведется по двум направлениям – *управление функциональными областями*, которые поддерживают множество унифицированных бизнес-процессов, разделенных на операции, и *управление интегрированными бизнес-процессами*, задачей которого является маршрутизация и координация унифицированных процессов для выполнения как оперативных заказов потребителей, так и глобальных проектов самой организации.

Фактически *основной задачей организационного проектирования является выбор оптимального соотношения между эффективностью использования ресурсов и эффективностью процессов.*

Жесткая специализация подразделений экономит ресурсы организации, но при этом снижает качество реализации процессов. Создание «процессных» команд, включающих собственных специалистов по всем ключевым операциям, обходится достаточно дорого, но при этом значительно сокращается время и повышается точность выполнения процесса. Иногда организации могут позволить себе выбрать этот путь, особенно в тех случаях, когда создается высокая ценность процесса, за которую потребитель согласен платить. Но, как правило, ищется какой-то компромисс на основе процессно-матричных структур.

Когда компания начинает ориентироваться на процессы, исключительно важной становится роль владельцев интегрированных межфункциональных

процессов, касающихся многих функциональных областей. Кроме того, новая парадигма деятельности предприятия вызывает появление большого числа процессов управления, распределенных по всему предприятию, а не сосредоточенных в специализированных организационных единицах, это: системы качества, бюджетирования, маркетинга и т. п.

Поэтому постановка бюджетирования как организационной, а не только финансовой задачи предполагает делегирование полномочий, т. е. власти (с которой нелегко расстаются). На более низкие уровни делегируется ответственность за принятие финансовых решений: о заключении сделки-договора, об оплате, о закупке, о скидках и отпуске в кредит и т. п. Это позволяет упростить связи между подразделениями и снизить количество уровней вертикального прохождения документов, т. е. является необходимым условием реализации классической схемы реинжиниринга.

Таким образом, процессная ориентация ведет к перестройке организационной структуры, делает организационную структуру компании более «плоской», что иллюстрирует тесную связь между «вертикальным» описанием организации (как структуры распределения ответственности, полномочий и взаимоотношений) и её «горизонтальным» описанием, как системы процессов.

2. Основные элементы процессного подхода

В рамках процессного подхода любое предприятие рассматривается как бизнес-система – система, которая представляет собой связанное множество бизнес-процессов, конечными целями которых является выпуск продукции или услуг.

Под бизнес-процессом понимают совокупность различных видов деятельности, которые создают результат, имеющий ценность для потребителя. Бизнес-процесс – это цепочка работ (функций), результатом которой является какой-либо продукт или услуга.

Каждый бизнес-процесс имеет свои границы и роли. В процессном подходе используются следующие ключевые роли.

Владелец процесса – человек, отвечающий за ход и результаты процесса в целом. Он должен знать бизнес-процесс, следить за его выполнением и совершенствовать его эффективность. Владельцу бизнес-процесса необходимо обладать коммуникативностью, энтузиазмом, способностью влиять на людей и производить изменения.

Лидер команды – работник, обладающий знаниями о бизнес-процессе и имеющий позитивные личные качества.

Коммуникатор – работник, обучающий команду различным методам работы, подготавливающий совместно с лидером совещания и анализирующий их результат.

Координатор процесса – работник, отвечающий за согласованную работу всех частей бизнеса и обеспечивающий связь с другими бизнес-процессами.

Координатор должен обладать административными способностями и пониманием стратегических целей предприятия.

Участники команды – специалисты различных уровней иерархии. Участники команды получают поддержку и методическое обеспечение от консультанта и коммуникатора, вместе с лидером проводят моделирование, анализ и оценку бизнес-процесса.

Одним из основных элементов процессного подхода является команда. Существует несколько типов процессных команд.

Ситуационная команда – обычно работает на постоянной основе и выполняет периодически повторяющуюся работу.

Виртуальная команда – создается для разработки нового продукта или услуги.

Ситуационный менеджер – высококвалифицированный специалист, способный самостоятельно выполнить до 90 % объема работ.

Важной задачей процессного подхода является формирование процессных команд.

Подготовка и формирование команды включает:

- учебные курсы;
- практический тренинг по освоению методов, методик и др.;
- психологическое тестирование;
- тестирование рабочих навыков.

Достижение определенной совокупности целей за счет выполнения бизнес-процессов называется деревом целей. Дерево целей имеет, как правило, иерархический вид. Каждая цель имеет свой вес и критерий (количественный или качественный) достижимости.

Бизнес-процессы реализуют бизнес-функции предприятия. Под бизнес-функцией понимают вид деятельности предприятия. Множество бизнес-функций представляет иерархическую декомпозицию функциональной деятельности и называется **деревом функций**.

Бизнес-функции связаны с показателями деятельности предприятия, образующими **дерево показателей**. На основании этих показателей строится система показателей оценки эффективности выполнения процессов. Владельцы процессов контролируют свои бизнес-процессы с помощью данной системы показателей.

Наиболее **общими показателями оценки эффективности бизнес-процессов являются:**

- количество производимой продукции заданного качества за определенный интервал времени;
- количество потребляемой продукции;
- длительность выполнения типовых операций и др.

3. Выделение и классификация процессов

При процессном описании должны решаться, как минимум, две задачи:

1. Идентификация всей системы «функциональных областей» и процессов компании, их взаимосвязей.

2. Выделение «ключевых» интегрированных процессов и их описание на потоковом уровне.

Каждая деятельность компании реализуется как процесс, который имеет своего потребителя: внешнего – клиента или внутреннего – сотрудников или подразделения компании, реализующих другие процессы. На стадии системного описания процессов выявляется значимость каждого процесса – в том числе происходит очищение от малопонятной деятельности. На этом этапе выбираются *ключевые процессы* для потокового описания, которое необходимо, например, для создания информационной системы предприятия.

Наиболее распространены следующие виды бизнес-процессов:

- процессы, создающие наибольшую добавочную стоимость (экономическую стоимость, которая определяется издержками компании, относимыми на продукцию);
- процессы, создающие наибольшую ценность для клиентов (маркетинговую стоимость за счет дифференциации продукции);
- процессы с наиболее интенсивным межзвенным взаимодействием, создающие транзакционные издержки;
- процессы, определенные стандартами ИСО-9000, как обязательные к описанию при постановке системы менеджмента качества.

Важнейшим шагом при структуризации любой компании является выделение и классификация бизнес-процессов. Целесообразно основываться на следующих *классах процессов*:

- основные процессы;
- процессы управления;
- процессы обеспечения;
- сопутствующие процессы;
- вспомогательные процессы;
- процессы развития.

Рассмотрим модель деятельности компании (рисунок 5.1), при описании которой используют процессы управления, основные бизнес-процессы и процессы обеспечения.

Основные бизнес-процессы – это процессы, ориентированные на производство товаров и услуг, представляющие ценность для клиента и обеспечивающие получение дохода.

Основные процессы образуют «жизненный цикл» продукции компании. Критериями эффективности таких процессов являются обычно качество, точность и своевременность выполнения каждого заказа. Многие потребители рассматривают улучшение качества как нечто более важное, чем уменьшение цены.



Рисунок 5.1 – Упрощенная модель деятельности компании

Искусный продавец может получить заказ на выполнение работ в условиях конкуренции с другими фирмами, однако только качество товара или услуги определяет в большей степени, повторит ли потребитель свой заказ у этого продавца еще раз. Таких процессов при развитии деятельности компании может быть много. Все они описываются по производственно-коммерческим цепочкам: «первоначальное взаимодействие с клиентом и определение его потребностей – реализация запроса (заявка, заказ, контракт и т. п.) – послепродажное сопровождение и мониторинг удовлетворения потребностей».

Процесс «реализации запроса клиента» может быть декомпозирован на следующие подпроцессы (процессы более низкого уровня):

- разработка (проектирование) продукции;
- закупка (товаров, материалов, комплектующих изделий);
- транспортировка (закупленного);
- разгрузка, приемка на склад и хранение (закупленного);
- производство (со своим технологическим циклом и внутренней логистикой);
- приемка на склад и хранение (готовой продукции);
- отгрузка (консервация и упаковка, погрузка, доставка);
- пуско-наладка;
- оказание услуг (предусмотренные контрактом на поставку или имеющие самостоятельное значение) и т. п.

Эти цепочки также достаточно стандартны (например, в стандарте ИСО приведены многие из этих процессов в качестве обязательных и подлежащих сертификации). Проверить, какие бизнес-цепочки существуют на предприятии, можно с помощью проекции каждого из выделенных «бизнесов, продукции и услуг» на вышеуказанный (стандартный) библиотечный классификатор жизненного или уже производственного цикла.

Для оценки этапов работы с любым документом можно использовать также анализ «жизненного цикла документа», который может выглядеть следующим образом:

- предоставляет исходные данные;
- готовит, разрабатывает;
- заполняет;

- корректирует;
- оформляет;
- подписывает;
- контролирует соответствие установленным требованиям;
- визирует;
- согласует;
- утверждает;
- акцентирует (принимает к сведению, использует);
- хранит;
- снимает копию.

Здесь тоже может быть применена своя матрица-генератор, как средство проверки полноты цикла.

Можно также воспользоваться референтными моделями деятельности аналогичных компаний – они могут сопоставляться с процессами конкурентов, лидеров отрасли, а также совершенствоваться.

Процессы управления – это процессы, охватывающие весь комплекс функций управления на уровне каждого бизнес-процесса и бизнес-системы в целом. Процессы управления имеют своей целью выработку и принятие управленческого решения. Данные управленческие решения могут приниматься относительно всей организации в целом, отдельной функциональной области или отдельных процессов, например:

- стратегическое управление;
- организационное проектирование (структуризация);
- маркетинг;
- финансово-экономическое управление;
- логистика и организация процессов;
- менеджмент качества;
- персонал.

Другая возможная систематизация функций управления связана с понятием управленческого цикла и базируется на пяти исходных функциях управления: *планирование, организация, распорядительство, координация, контроль*. Самая распространенная ошибка – это смешение этих принципов.

Для реализации процессного описания исключительно важным является то, что любая управленческая деятельность развертывается по так называемому «управленческому циклу», который включает:

- сбор информации;
- выработку решения;
- реализацию;
- учет;
- контроль;
- анализ;
- регулирование.

Например, наиболее часто встречающиеся варианты детализации:

- сбор информации;
- определение состава собираемой информации;
- определение форм отчетности;
- выработка решения;
- анализ альтернатив;
- подготовка вариантов решения;
- принятие решения;
- выработка критериев оценки;
- реализация;
- планирование;
- организация;
- мотивация;
- координация;
- контроль исполнения**
- учет результатов;
- сравнение по принятым критериям;
- анализ**
- анализ дополнительной информации;
- диагностика возможных причин отклонений;
- регулирование**
- регулирование на уровне реализации;
- регулирование на уровне выработки решения.

Каждый из этих этапов имеет своих характерных для него исполнителей – управленицев, которых можно отнести к трем основным категориям:

- руководитель (ответственный за принятие и организацию выполнения решений);
- специалист-аналитик (ответственный за подготовку решения и анализ отклонений);
- технические исполнители (сбор информации, учет, коммуникации).

Согласно некоторым подходам, в процессах управления выделяются два типа процессов, относящихся, соответственно, к двум типам менеджмента, условно обозначаемым как «менеджмент ресурсов» и «менеджмент организации», которые отличаются по объекту управления, базовым моделям и, что важно для описания процессов, – своими управлеченческими циклами. Тогда модель деятельности предприятия становится двухуровневой (рисунок 5.2).

Из модели следует, что сами циклы ресурсного планирования нуждаются в регламентации – то есть ресурсное управление может осуществляться только по специально разработанным организационным регламентам.



Рисунок 5.2 – Двухуровневая модель деятельности предприятия

В основе **цикла управления ресурсами** лежит расчет или имитационное моделирование и контроль результатов:

- выбор (или получение от системы верхнего уровня) целевого критерия оценки качества решения;
- сбор информации о ресурсах предприятия или возможностях внешней среды;
- просчет вариантов (с различными предположениями о возможных значениях параметров);
- выбор оптимального варианта – принятие решения (ресурсного плана);
- учет результатов (и точность);
- сравнение с принятым критерием оценки (контроль результатов);
- анализ причин отклонений и регулирование (возврат к 1, 2 или 3).

В основе **цикла организационного менеджмента** лежит структурное или процессное моделирование и процедурный контроль:

- определение состава задач (обособленных функций, операций);
- выбор исполнителей (распределение зон и степени ответственности);
- проектирование процедур (последовательности и порядка исполнения);
- согласование и утверждение регламента исполнения (процесса, плана мероприятий);
- отчетность об исполнении;
- контроль исполнения (процедурный контроль);
- анализ причин отклонений и регулирование (возврат к 1, 2 или 3).

Таким образом, на определенных шагах декомпозиции предприятию надо определить, какие стадии управленческого цикла реализуются по каждой из ранее выделенных задач управления. Это можно проверить с помощью матрицы-генератора, которая раскладывает компоненты менеджмента по этапам управленческого цикла.

Процессы обеспечения – это процессы, предназначенные для жизнеобеспечения основных и сопутствующих процессов и ориентированные на

поддержку их универсальных средств. Например, процесс финансового обеспечения, процесс обеспечения кадрами, процесс юридического обеспечения – это вторичные процессы. Они создают и поддерживают необходимые условия для выполнения основных функций и функций менеджмента. Клиенты обеспечивающих процессов находятся внутри компании.

На верхнем уровне детализации можно выделить примерно следующие стандартные процессы обеспечения:

- обеспечение производства;
- техобслуживание и ремонт оборудования;
- обеспечение теплоэнергоресурсами;
- обслуживание и ремонт зданий и сооружений;
- технологическое обеспечение;
- метрологическое обеспечение;
- техника безопасности;
- экологический контроль и т.п.
- обеспечение управления;
- информационное обеспечение;
- обеспечение документооборота;
- коммуникационное обеспечение;
- юридическое обеспечение;
- обеспечение безопасности;
- материально-техническое обеспечение управления;
- хозяйственное обеспечение;
- обеспечение коммунальными услугами;
- транспортное обслуживание и т. п.

Для каждого из выделенных выше подпроцессов также следует определить, какой основной или управленческий процесс является потребителем этих «внутренних услуг». Для этого существуют свои матрицы-генераторы. Их можно построить отдельно для основных процессов (рисунок 5.3).

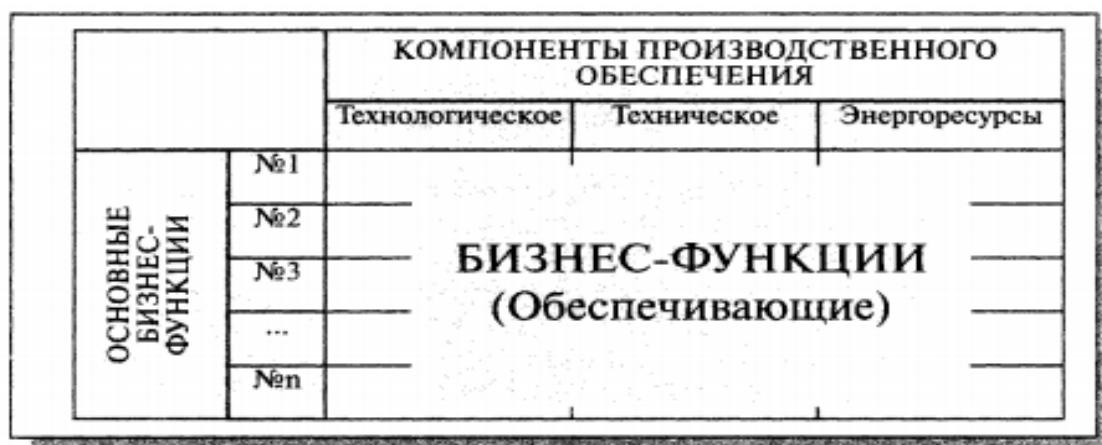


Рисунок 5.3 – Упрощенная матрица-генератор обеспечивающих бизнес-функций

И для процессов управления (рисунок 5.4).



Рисунок 5.4 – Матрица-генератор обеспечивающих бизнес-функций

Разбиение данных процессов производится по индивидуальным технологическим цепочкам. Многие из обеспечивающих процессов стандартны для всех компаний или определенных видов деятельности: промышленность, торговля, предоставление услуг и т. п. Однако, как правило, данный класс функций в меньшей степени «подвергается» потоковому процессному описанию. Большинство из них достаточно хорошо регламентируются должностными и специальными инструкциями.

Референтная модель бизнес-процесса

В качестве основного каркаса, объединяющего и систематизирующего все знания по бизнес-модели, можно использовать референтную модель.

Референтная модель – это модель эффективного бизнес-процесса, созданная для предприятия конкретной области, внедренная на практике и предназначенная для использования при разработке/реорганизации бизнес-процессов на других предприятиях.

По сути, референтные модели представляют собой эталонные схемы организации бизнеса, разработанные для конкретных бизнес-процессов на основе реального опыта внедрения в различных компаниях по всему миру. Они включают в себя проверенные на практике процедуры и методы организации управления. Референтные модели позволяют предприятиям начать разработку собственных моделей на базе уже готового набора функций и процессов.

Референтная модель бизнес-процесса представляет собой совокупность логически взаимосвязанных функций. Для каждой функции указывается исполнитель, входные и выходные документы или информационные объекты. Элементы (функции и документы) референтной модели бизнес-процесса содержат ссылки на соответствующие объекты ИС, а также документы и другую информацию (пользовательские инструкции и т. п.), расположенную в репозитарии проекта. Отсюда и название – референтная модель (в переводе с английского – ссылочная модель).

ЛЕКЦИЯ № 6. МЕТОДОЛОГИЯ МОДЕЛИРОВАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ

1. Структурная модель предметной области.
2. Функционально-ориентированные и объектно-ориентированные методологии описания предметной области.
3. Синтетическая методика.

1. 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;
- 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 взаимодействие события и бизнес-правила, которые воздействует 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 согласования 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 complete, 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 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 (*определение требований*), на концептуальном уровне (*спецификация требований*) и внутреннем уровне (*реализация 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 ИС. 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 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 не декомпозируемую последовательность действий.

На внешнем уровне моделирования определяется список основных бизнес-функций или видов бизнес-процессов. Обычно таких функций насчитывается от 15 до 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 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*. Информационное 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 функции 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 вид 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. п.

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 со 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), представляющая 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 **функционально-ориентированные и объектно-ориентированные методологии**.

2. Функционально-ориентированные и объектно-ориентированные методологии описания предметной области

Процесс бизнес-моделирования может быть реализован в рамках различных методик, отличающихся прежде всего своим подходом к тому, что представляет собой моделируемая организация. В соответствии с различными представлениями об организации методики принято делить на объектные и функциональные (структурные).

Объектные методики рассматривают моделируемую организацию как набор взаимодействующих объектов – производственных единиц. Объект определяется как осязаемая реальность – предмет или явление, имеющие четко определяемое поведение. Целью применения данной методики является выделение объектов, составляющих организацию, и распределение между ними ответственостей за выполняемые действия.

Функциональные методики, наиболее известной из которых является методика IDEF, *рассматривают организацию как набор функций, преобразующий, поступающий поток информации в выходной поток.* Процесс преобразования информации потребляет определенные ресурсы. Основное отличие от объектной методики заключается в четком отделении функций (методов обработки данных) от самих данных.

С точки зрения бизнес-моделирования каждый из представленных подходов обладает своими преимуществами. Объектный подход позволяет построить более устойчивую к изменению систему, лучше соответствует существующим структурам организации. Функциональное моделирование хорошо показывает себя в тех случаях, когда организационная структура находится в процессе изменения или вообще слабо оформлена. Подход от выполняемых функций интуитивно лучше понимается исполнителями при получении от них информации об их текущей работе.

Функциональная методика IDEF0

Методология IDEF0 считается следующим этапом развития хорошо известного графического языка описания функциональных систем SADT (Structured Analysis and Design Technique).

Целью методики является построение функциональной схемы исследуемой системы, описывающей все необходимые процессы с точностью, достаточной для однозначного моделирования деятельности системы.

В основе методологии лежат четыре основных понятия: *функциональный блок, интерфейсная дуга, декомпозиция, глоссарий.*

Функциональный блок представляет собой некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольном наклонении (например, производить услуги).

Подробная информация о методологии была рассмотрена нами на практических занятиях.

Обычно IDEF0-модели несут в себе сложную и концентрированную информацию, и для того, чтобы ограничить их перегруженность и сделать удобочитаемыми, в стандарте приняты соответствующие ограничения сложности.

Рекомендуется представлять на диаграмме от трех до шести функциональных блоков, при этом количество подходящих к одному функциональному блоку (выходящих из одного функционального блока) интерфейсных дуг предполагается не более четырех.

Стандарт IDEF0 содержит набор процедур, позволяющих разрабатывать и согласовывать модель большой группы людей, принадлежащих к разным областям деятельности моделируемой системы. Обычно процесс разработки является итеративным и состоит из следующих условных этапов:

1. *Создание модели группой специалистов*, относящихся к различным сферам деятельности предприятия. Эта группа в терминах IDEF0 называется авторами (Authors). Построение первоначальной модели является динамическим процессом, в течение которого авторы опрашивают компетентных лиц о структуре различных процессов, создавая модели деятельности подразделений. При этом их интересуют ответы на следующие вопросы:

- Что поступает в подразделение «на входе»?
- Какие функции и в какой последовательности выполняются в рамках подразделения?
- Кто является ответственным за выполнение каждой из функций?
- Чем руководствуется исполнитель при выполнении каждой из функций?
- Что является результатом работы подразделения (на выходе)?

На основе имеющихся положений, документов и результатов опросов создается черновик (Model Draft) модели.

2. *Распространение черновика для рассмотрения*, согласований и комментариев. На этой стадии происходит обсуждение черновика модели с широким кругом компетентных лиц (в терминах IDEF0 – читателей) на предприятии.

При этом каждая из диаграмм черновой модели письменно критикуется и комментируется, а затем передается автору. Автор, в свою очередь, также письменно соглашается с критикой или отвергает ее с изложением логики принятия решения и вновь возвращает откорректированный черновик для дальнейшего рассмотрения. Этот цикл продолжается до тех пор, пока авторы и читатели не придут к единому мнению.

3. *Официальное утверждение модели*. Утверждение согласованной модели происходит руководителем рабочей группы в том случае, если у авторов модели и читателей отсутствуют разногласия по поводу ее адекватности. Окончательная модель представляет собой согласованное представление о предприятии (системе) с заданной точки зрения и для заданной цели.

Наглядность графического языка IDEF0 делает модель вполне читаемой и для лиц, которые не принимали участия в проекте ее создания, а также эффективной для проведения показов и презентаций. В дальнейшем на базе построенной модели могут быть организованы новые проекты, нацеленные на производство изменений в модели.

Функциональная методика потоков данных

Целью методики является построение модели рассматриваемой системы в виде диаграммы потоков данных (Data Flow Diagram – DFD), обеспечивающей правильное описание выходов (отклика системы в виде данных) при заданном воздействии на вход системы (подаче сигналов через внешние интерфейсы). Диаграммы потоков данных являются основным средством моделирования функциональных требований к проектируемой системе.

При создании диаграмм потоков данных используются четыре основных понятия: *потоки данных, процессы (работы) преобразования входных потоков данных в выходные, внешние сущности, накопители данных (хранилища)*.

Потоки данных являются абстракциями, использующимися для моделирования передачи информации (или физических компонент) из одной части системы в другую. Потоки на диаграммах изображаются именованными стрелками, ориентация которых указывает направление движения информации.

Назначение процесса (работы) состоит в производстве выходных потоков из входных в соответствии с действием, задаваемым именем процесса. Имя процесса должно содержать глагол в неопределенной форме с последующим дополнением (например, «получить документы по отгрузке продукции»). Каждый процесс имеет уникальный номер для ссылок на него внутри диаграммы, который может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели.

Хранилище (накопитель) данных позволяет на указанных участках определять данные, которые будут сохраняться в памяти между процессами. Фактически хранилище представляет собой «срезы» потоков данных во времени. Информация, которую оно содержит, может использоваться в любое время после ее получения, при этом данные могут выбираться в любом порядке. Имя хранилища должно определять его содержание и быть существительным.

Внешняя сущность представляет собой материальный объект вне контекста системы, являющейся источником или приемником системных данных. Ее имя должно содержать существительное, например, «склад товаров». Предполагается, что объекты, представленные как внешние сущности, не должны участвовать ни в какой обработке.

Кроме основных элементов, в состав DFD входят словари данных и миниспецификации.

Словари данных являются каталогами всех элементов данных, присутствующих в DFD, включая групповые и индивидуальные потоки данных, хранилища и процессы, а также все их атрибуты.

Миниспецификации обработки описывают DFD-процессы нижнего уровня. Фактически миниспецификации представляют собой алгоритмы

описания задач, выполняемых процессами: множество всех миниспецификаций является полной спецификацией системы. Процесс построения DFD начинается с создания так называемой основной диаграммы типа «звезда», на которой представлен моделируемый процесс и все внешние сущности, с которыми он взаимодействует. В случае сложного основного процесса он сразу представляется в виде декомпозиции на ряд взаимодействующих процессов. Критериями сложности в данном случае являются: наличие большого числа внешних сущностей, многофункциональность системы, ее распределительный характер.

Внешние сущности выделяются по отношению к основному процессу. Для их определения необходимо выделить поставщиков и потребителей основного процесса, т. е. все объекты, которые взаимодействуют с основным процессом. На этом этапе описание взаимодействия заключается в выборе глагола, дающего представление о том, как внешняя сущность использует основной процесс или используется им.

Например, основной процесс – «учет обращений граждан», внешняя сущность – «граждане», описание взаимодействия – «подает заявления и получает ответы». Этот этап является принципиально важным, поскольку именно он определяет границы моделируемой системы.

Для всех внешних сущностей строится таблица событий, описывающая их взаимодействие с основным потоком. Таблица событий включает в себя наименование внешней сущности, событие, его тип (типичный для системы или исключительный, реализующийся при определенных условиях) и реакцию системы.

На следующем этапе происходит декомпозиция основного процесса на набор взаимосвязанных процессов, обменивающихся потоками данных. Сами потоки не конкретизируются, определяется лишь характер взаимодействия. Декомпозиция завершается, когда процесс становится простым, т. е.:

- 1) процесс имеет два-три входных и выходных потока;
- 2) процесс может быть описан в виде преобразования входных данных в выходные;
- 3) процесс может быть описан в виде последовательного алгоритма.

Для простых процессов строится мини-спецификация – формальное описание алгоритма преобразования входных данных в выходные.

Мини-спецификация удовлетворяет следующим требованиям: для каждого процесса строится одна спецификация; спецификация однозначно определяет входные и выходные потоки для данного процесса; спецификация не определяет способ преобразования входных потоков в выходные; спецификация ссылается на имеющиеся элементы, не вводя новые; спецификация по возможности использует стандартные подходы и операции.

После декомпозиции основного процесса для каждого подпроцесса строится аналогичная таблица внутренних событий.

Следующим шагом после определения полной таблицы событий выделяются потоки данных, которыми обмениваются процессы и внешние

сущности. Простейший способ их выделения заключается в анализе таблиц событий. События преобразуются в потоки данных от инициатора события к запрашиваемому процессу, а реакции – в обратный поток событий. После построения входных и выходных потоков аналогичным образом строятся внутренние потоки. Для их выделения для каждого из внутренних процессов выделяются поставщики и потребители информации. Если поставщик или потребитель информации представляет процесс сохранения или запроса информации, то вводится хранилище данных, для которого данный процесс является интерфейсом.

После построения потоков данных диаграмма должна быть проверена на полноту и непротиворечивость. Полнота диаграммы обеспечивается, если в системе нет «повисших» процессов, не используемых в процессе преобразования входных потоков в выходные. Непротиворечивость системы обеспечивается выполнением наборов формальных правил о возможных типах процессов: *на диаграмме не может быть потока, связывающего две внешние сущности – это взаимодействие удаляется из рассмотрения; ни одна сущность не может непосредственно получать или отдавать информацию в хранилище данных – хранилище данных является пассивным элементом, управляемым с помощью интерфейсного процесса; два хранилища данных не могут непосредственно обмениваться информацией – эти хранилища должны быть объединены.*

К преимуществам методики DFD относятся:

- возможность однозначно определить внешние сущности, анализируя потоки информации внутри и вне системы;
- возможность проектирования сверху вниз, что облегчает построение модели «как должно быть»;
- наличие спецификаций процессов нижнего уровня, что позволяет преодолеть логическую незавершенность функциональной модели и построить полную функциональную спецификацию разрабатываемой системы.

К недостаткам модели относятся: *необходимость искусственного ввода управляющих процессов, поскольку управляющие воздействия (потоки) и управляющие процессы с точки зрения DFD ничем не отличаются от обычных; отсутствие понятия времени, т. е. отсутствие анализа временных промежутков при преобразовании данных (все ограничения по времени должны быть введены в спецификациях процессов).*

Объектно-ориентированная методика

Принципиальное отличие между функциональным и объектным подходом заключается в способе декомпозиции системы. Объектно-ориентированный подход использует объектную декомпозицию, при этом статическая структура описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами. Целью методики является построение бизнес-модели организации, позволяющей перейти от модели сценариев использования к модели, определяющей отдельные объекты, участвующие в реализации бизнес-функций.

Концептуальной основой объектно-ориентированного подхода является объектная модель, которая строится с учетом следующих принципов:

- абстрагирование;
- инкапсуляция;
- модульность;
- иерархия;
- типизация;
- параллелизм;
- устойчивость.

Основными понятиями объектно-ориентированного подхода являются объект и класс.

Объект – предмет или явление, имеющие четко определенное поведение и обладает состоянием, поведением и индивидуальностью. Структура и поведение схожих объектов определяют общий для них класс. *Класс* – это множество объектов, связанных общностью структуры и поведения. Следующую группу важных понятий объектного подхода составляют наследование и полиморфизм. Понятие *полиморфизм* может быть интерпретировано как способность класса принадлежать более чем одному типу. *Наследование* означает построение новых классов на основе существующих с возможностью добавления или переопределения данных и методов.

Важным качеством объектного подхода является согласованность моделей деятельности организации и моделей проектируемой информационной системы от стадии формирования требований до стадии реализации. По объектным моделям может быть прослежено отображение реальных сущностей моделируемой предметной области (организации) в объекты и классы информационной системы.

Большинство существующих методов объектно-ориентированного подхода включают язык моделирования и описание процесса моделирования. Процесс – это описание шагов, которые необходимо выполнить при разработке проекта. В качестве языка моделирования объектного подхода используется унифицированный язык моделирования UML, который содержит стандартный набор диаграмм для моделирования.

Диаграмма (Diagram) – это графическое представление множества элементов, чаще всего она изображается в виде связного графа с вершинами (сущностями) и ребрами (отношениями) и представляет собой некоторую проекцию системы.

Объектно-ориентированный подход обладает следующими преимуществами:

- Объектная декомпозиция дает возможность создавать модели меньшего размера путем использования общих механизмов, обеспечивающих необходимую экономию выразительных средств. Использование объектного подхода существенно повышает уровень унификации разработки и пригодность для повторного использования, что ведет к созданию среды разработки и переходу к сборочному созданию моделей.

- Объектная декомпозиция позволяет избежать создания сложных моделей, так как она предполагает эволюционный путь развития модели на базе относительно небольших подсистем.
- Объектная модель естественна, поскольку ориентирована на человеческое восприятие мира.

К недостаткам объектно-ориентированного подхода относятся высокие начальные затраты. Этот подход не дает немедленной отдачи. Эффект от его применения сказывается после разработки двух – трех проектов и накопления повторно используемых компонентов. Диаграммы, отражающие специфику объектного подхода, менее наглядны.

Сравнение методик моделирования

В функциональных моделях (DFD-диаграммах потоков данных, SADT-диаграммах) главными структурными компонентами являются функции (операции, действия, работы), которые на диаграммах связываются между собой потоками объектов.

Несомненным достоинством функциональных моделей является реализация структурного подхода к проектированию ИС по принципу «сверху-вниз», когда каждый функциональный блок может быть декомпозирован на множество подфункций и т. д., выполняя, таким образом, модульное проектирование ИС. Для функциональных моделей характерны процедурная строгость декомпозиции ИС и наглядность представления.

При функциональном подходе объектные модели данных в виде ER-диаграмм «объект – свойство – связь» разрабатываются отдельно. Для проверки корректности моделирования предметной области между функциональными и объектными моделями устанавливаются взаимно однозначные связи.

Главный недостаток функциональных моделей заключается в том, что процессы и данные существуют отдельно друг от друга – помимо функциональной декомпозиции существует структура данных, находящаяся на втором плане. Кроме того, не ясны условия выполнения процессов обработки информации, которые динамически могут изменяться.

Перечисленные недостатки функциональных моделей снимаются в объектно-ориентированных моделях, в которых главным структурообразующим компонентом выступает класс объектов с набором функций, которые могут обращаться к атрибутам этого класса.

Для классов объектов характерна иерархия обобщения, позволяющая осуществлять наследование не только атрибутов (свойств) объектов от вышестоящего класса объектов к нижестоящему классу, но и функций (методов).

В случае наследования функций можно абстрагироваться от конкретной реализации процедур (абстрактные типы данных), которые отличаются для определенных подклассов ситуаций. Это дает возможность обращаться к подобным программным модулям по общим именам (полиморфизм) и осуществлять повторное использование программного кода при модификации программного обеспечения. Таким образом, адаптивность объектно-

ориентированных систем к изменению предметной области по сравнению с функциональным подходом значительно выше.

При объектно-ориентированном подходе изменяется и принцип проектирования ИС. Сначала выделяются классы объектов, а далее в зависимости от возможных состояний объектов (жизненного цикла объектов) определяются методы обработки (функциональные процедуры), что обеспечивает наилучшую реализацию динамического поведения информационной системы.

Для объектно-ориентированного подхода разработаны графические методы моделирования предметной области, обобщенные в языке унифицированного моделирования UML. Однако по наглядности представления модели пользователю-заказчику объектно-ориентированные модели явно уступают функциональным моделям.

Для модели предметной области обычно в качестве критерия выступает степень ее динамичности. Для более регламентированных задач больше подходят функциональные модели, для более адаптивных бизнес-процессов (управления рабочими потоками, реализации динамических запросов к информационным хранилищам) – объектно-ориентированные модели. Однако в рамках одной и той же ИС для различных классов задач могут требоваться различные виды моделей, описывающих одну и ту же проблемную область. В таком случае должны использоваться комбинированные модели предметной области.

3. Синтетическая методика

Из рассмотренного материала видно, что каждая из представленных методик позволяет решить задачу построения формального описания рабочих процедур исследуемой системы. Все методики позволяют построить модель «как есть» и «как должно быть». С другой стороны, каждая из этих методик обладает существенными недостатками. Их можно суммировать следующим образом: недостатки применения отдельной методики лежат не в области описания реальных процессов, а в неполноте методического подхода.

Функциональные методики в целом лучше дают представление о существующих функциях в организации, о методах ее реализации, причем чем выше степень детализации исследуемого процесса, тем лучше они позволяют описать систему. Под лучшим описанием в данном случае понимается наименьшая ошибка при попытке по полученной модели предсказать поведение реальной системы. На уровне отдельных рабочих процедур их описание практически однозначно совпадает с фактической реализацией в потоке работ.

На уровне общего описания системы функциональные методики допускают значительную степень произвола в выборе общих интерфейсов системы, ее механизмов и т. д., то есть в определении границ системы. Хорошо описать систему на этом уровне позволяет **объектный подход**, основанный на понятии сценария использования. Ключевым является понятие о сценарии

использования как о сеансе взаимодействия действующего лица с системой, в результате которого действующее лицо получает нечто, имеющее для него ценность. Использование критерия ценности для пользователя дает возможность отбросить не имеющие значения детали потоков работ и сосредоточиться на тех функциях системы, которые оправдывают ее существование. Однако в этом случае задача определения границ системы, выделения внешних пользователей является сложной.

Технология потоков данных, исторически возникшая первой, легко решает проблему границ системы, поскольку позволяет за счет анализа информационных потоков выделить внешние сущности и определить основной внутренний процесс. Однако отсутствие выделенных управляющих процессов, потоков и событийной ориентированности не позволяет предложить эту методику в качестве единственной.

Наилучшим способом преодоления недостатков рассмотренных методик является формирование *синергетической методики*, объединяющей различные этапы отдельных методик. При этом из каждой методики необходимо взять часть методологии, наиболее полно и формально изложенную, и обеспечить возможность обмена результатами на различных этапах применения синергетической методики. В бизнес-моделировании неявным образом идет формирование подобной синергетической методики.

Идея *синтетической методики* заключается в последовательном применении функционального и объектного подхода с учетом возможности реинжиниринга существующей ситуации.

Рассмотрим применение синтетической методики на примере разработки административного регламента.

При построении административных регламентов выделяются следующие стадии.

1. Определение границ системы. На этой стадии при помощи анализа *потоков данных* выделяют внешние сущности и собственно моделируемую систему.
2. Выделение сценариев использования системы. На этой стадии при помощи *критерия полезности* строят для каждой внешней сущности *набор сценариев использования системы*.
3. Добавление системных сценариев использования. На этой стадии определяют сценарии, необходимые для реализации целей системы, отличных от целей пользователей.
4. Построение диаграммы активностей по сценариям использования. На этой стадии строят набор действий системы, приводящих к реализации сценариев использования.
5. Функциональная декомпозиция диаграмм активностей как контекстных диаграмм методики IDEF0.
6. Формальное описание отдельных функциональных активностей в виде административного регламента (с применением различных нотаций).

ЛЕКЦИЯ № 7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

1. Внемашинное информационное обеспечение.

2. Внутримашинное информационное обеспечение.

Информационное обеспечение ИС является средством для решения следующих задач:

- однозначного и экономичного представления информации в системе (на основе кодирования объектов);
- организации процедур анализа и обработки информации с учетом характера связей между объектами (на основе *классификации* объектов);
- организации взаимодействия пользователей с системой (на основе *экранных форм* ввода-вывода данных);
- обеспечения эффективного использования информации в контуре управления деятельностью объекта автоматизации (на основе унифицированной *системы документации*).

Информационное обеспечение ИС включает два комплекса: **внемашинное информационное обеспечение** (классификаторы технико-экономической информации, документы, методические инструктивные материалы) и **внутримашинное информационное обеспечение** (макеты/экранные формы для ввода первичных данных в компьютер или вывода результатной информации, структуры *информационной базы*: входных, выходных файлов, *базы данных*). К информационному обеспечению предъявляются следующие общие требования:

- информационное обеспечение должно быть достаточным для поддержания всех автоматизируемых функций объекта;
- для кодирования информации должны использоваться принятые у заказчика классификаторы;
- для кодирования входной и выходной информации, которая используется на высшем уровне управления, должны быть использованы классификаторы этого уровня;
- должна быть обеспечена совместимость с информационным обеспечением систем, взаимодействующих с разрабатываемой системой;
- формы документов должны отвечать требованиям корпоративных стандартов заказчика (или унифицированной системы документации);
- структура документов и экранных форм должна соответствовать характеристикам терминалов на рабочих местах конечных пользователей;
- графики формирования и содержание информационных сообщений, а также используемые аbbревиатуры должны быть общеприняты в этой предметной области и согласованы с заказчиком;
- в ИС должны быть предусмотрены средства контроля входной и результатной информации, обновления данных в информационных

массивах, контроля целостности информационной базы, защиты от несанкционированного доступа.

Информационное обеспечение ИС можно определить как *совокупность единой системы классификации, унифицированной системы документации и информационной базы*.

1. Внемашинное информационное обеспечение

1.1. Основные понятия классификации технико-экономической информации

Для того чтобы обеспечить эффективный поиск, обработку на ЭВМ и передачу по каналам связи технико-экономической информации, ее необходимо представить в цифровом виде. С этой целью ее нужно сначала упорядочить (классифицировать), а затем формализовать (закодировать) с использованием *классификатора*.

Классификация – это *разделение множества объектов на подмножества по их сходству или различию в соответствии с принятыми методами*. *Классификация* фиксирует закономерные связи между классами объектов. Под объектом понимается любой предмет, процесс, явление материального или нематериального свойства. *Система классификации* позволяет сгруппировать объекты и выделить определенные классы, которые будут характеризоваться рядом общих свойств. Таким образом, совокупность правил распределения объектов множества на подмножества называется *системой классификации*.

Свойство или характеристика объекта *классификации*, которое позволяет установить его сходство или различие с другими объектами *классификации*, называется *признаком классификации*. Например, признак «роль предприятия-партнера в отношении деятельности объекта автоматизации» позволяет разделить все предприятия на две группы (на два подмножества): «поставщики» и «потребители». Множество или подмножество, объединяющее часть объектов *классификации* по одному или нескольким признакам, носит название *классификационной группировки*.

Классификатор – это документ, с помощью которого осуществляется формализованное описание информации в ИС, содержащей наименования объектов, наименования классификационных группировок и их кодовые обозначения.

По сфере действия выделяют следующие виды *классификаторов*: международные, общегосударственные (общесистемные), отраслевые и локальные *классификаторы*.

Международные классификаторы входят в состав Системы международных экономических стандартов (СМЭС) и обязательны для передачи информации между организациями разных стран мирового сообщества.

Общегосударственные (общесистемные) классификаторы обязательны для организации процессов передачи и обработки информации между экономическими системами государственного уровня внутри страны.

Отраслевые классификаторы используют для выполнения процедур обработки информации и передачи ее между организациями внутри отрасли.

Локальные классификаторы используют в пределах отдельных предприятий.

Каждая *система классификации* характеризуется следующими свойствами:

- гибкостью системы;
- емкостью системы;
- степенью заполненности системы.

Гибкость системы – это способность допускать включение новых признаков, объектов без разрушения структуры *классификатора*. Необходимая гибкость определяется временем жизни системы.

Емкость системы – это наибольшее количество классификационных группировок, допускаемое в данной *системе классификации*.

Степень заполненности системы определяется как частное от деления фактического количества группировок на величину емкости системы.

В настоящее время чаще всего применяются два типа систем классификации: иерархическая и многоаспектная.

При использовании иерархического метода *классификации* происходит «последовательное разделение множества объектов на подчиненные, зависимые классификационные группировки». Получаемая на основе этого процесса классификационная схема имеет иерархическую структуру. В ней первоначальный объем классифицируемых объектов разбивается на подмножества по какому-либо признаку и детализируется на каждой следующей ступени *классификации*. Обобщенное изображение иерархической *классификационной схемы* представлено на рисунке 7.1.

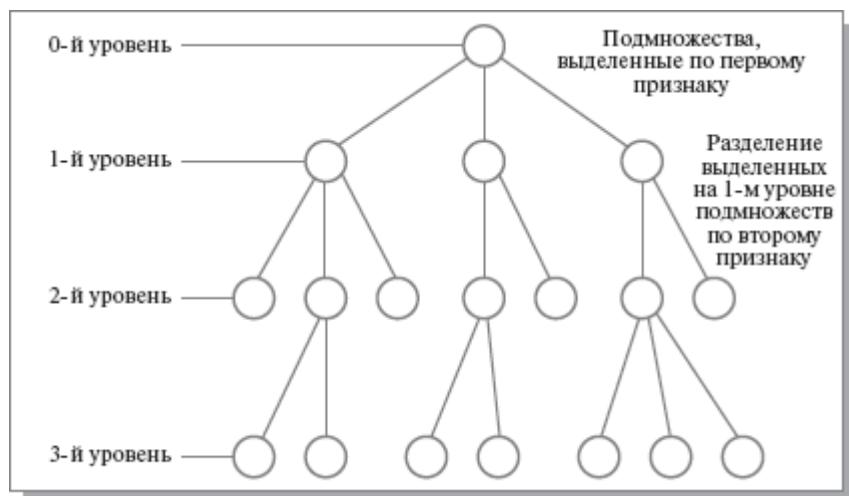


Рисунок 7.1 – Иерархическая классификационная схема

Характерными особенностями иерархической системы являются:

- возможность использования неограниченного количества признаков *классификации*;

- соподчиненность признаков *классификации*, что выражается разбиением каждой классификационной группировки, образованной по одному признаку, на множество классификационных группировок по нижестоящему (подчиненному) признаку.

Таким образом, классификационные схемы, построенные на основе иерархического принципа, имеют неограниченную емкость, величина которой зависит от глубины *классификации* (числа ступеней деления) и количества объектов *классификации*, которое можно расположить на каждой ступени. Количество же объектов на каждой ступени *классификации* определяется основанием кода, то есть числом знаков в выбранном алфавите кода (например, если алфавит – двузначные десятичные числа, то можно на одном уровне разместить 100 объектов). Выбор необходимой глубины *классификации* и структуры кода зависит от характера объектов *классификации* и характера задач, для решения которых предназначен *классификатор*.

При построении иерархической *системы классификации* сначала выделяется некоторое множество объектов, подлежащее классифицированию, для которого определяются полное множество признаков *классификации* и их соподчиненность друг другу, затем производится разбиение исходного множества объектов на классификационные группировки на каждой ступени *классификации*.

К положительным сторонам данной системы следует отнести логичность, простоту ее построения и удобство логической и арифметической обработки.

Серьезным недостатком иерархического метода *классификации* является жесткость *классификационной схемы*. Она обусловлена заранее установленным выбором признаков *классификации* и порядком их использования по ступеням *классификации*. Это ведет к тому, что при изменении состава объектов *классификации*, их характеристик или характера решаемых при помощи *классификатора* задач требуется коренная переработка *классификационной схемы*. Гибкость этой системы обеспечивается только за счет ввода большой избыточности в ветвях, что приводит к слабой заполненности структуры *классификатора*. Поэтому при разработке *классификаторов* следует учитывать, что иерархический метод классификации более предпочтителен для объектов с относительно стабильными признаками и для решения стабильного комплекса задач.

Примеры применения иерархической *классификации* объектов в корпоративной ИС приведены на рисунках 7.2 и 7.3.

Использование приведенных моделей позволяет выполнить *кодирование информации* о соответствующих объектах, а также использовать процедуры обобщения при обработке данных (при анализе затрат на заработную плату – по принадлежности работника к определенной службе, при анализе затрат на производство – по группам материалов: по металлу, по покупным комплектующим и пр.).

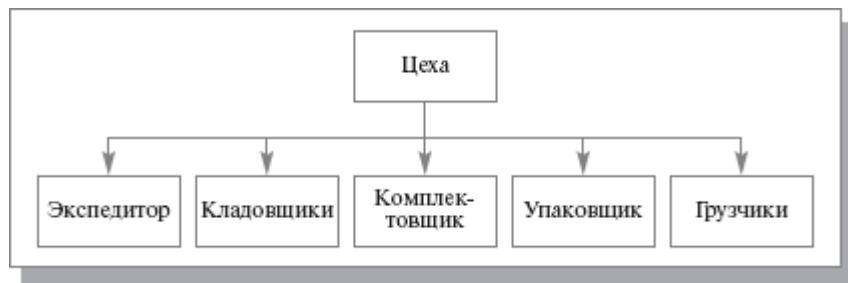


Рисунок 7.2 – Организационная структура подразделения предприятия – цеха отгрузки

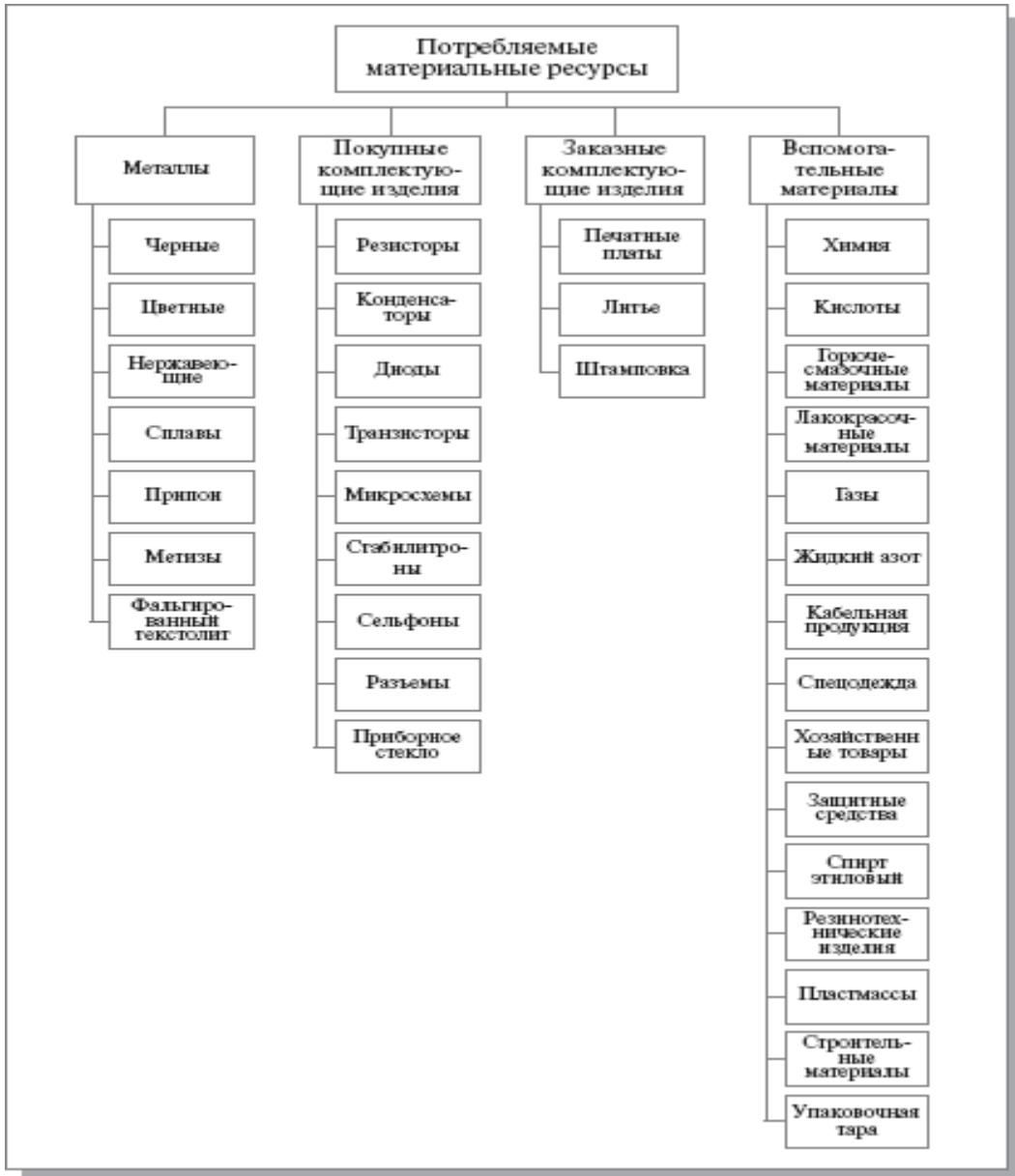


Рисунок 7.3 – Классификатор материальных ресурсов для обеспечения производства

1.2. Многоаспектные системы классификации

Недостатки, отмеченные в иерархической системе, отсутствуют в других системах, которые относятся к классу многоаспектных систем классификации.

Аспект – точка зрения на объект классификации, который характеризуется одним или несколькими признаками. **Многоаспектная система** – это система классификации, которая использует параллельно несколько независимых признаков (аспектов) в качестве основания классификации. Существуют два типа многоаспектных систем: фасетная и дескрипторная.

Фасет – это аспект классификации, который используется для образования независимых классификационных группировок.

Дескриптор – ключевое слово, определяющее некоторое понятие, которое формирует описание объекта и дает принадлежность этого объекта к классу, группе и т. д.

Под фасетным методом классификации понимается «параллельное разделение множества объектов на независимые классификационные группировки». При этом методе классификации заранее жесткой классификационной схемы и конечных группировок не создается. Разрабатывается лишь система таблиц признаков объектов классификации, называемых фасетами. При необходимости создания классификационной группировки для решения конкретной задачи осуществляется выборка необходимых признаков из фасетов и их объединение в определенной последовательности. Общий вид фасетной классификационной схемы представлен на рисунке 7.4.

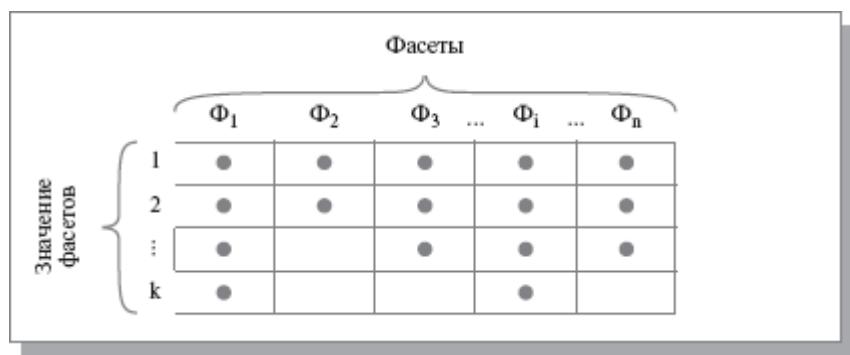


Рисунок 7.4 – Схема признаков фасетной классификации

Внутри фасета значения признаков могут просто перечисляться по некоторому порядку или образовывать сложную иерархическую структуру, если существует соподчиненность выделенных признаков.

К преимуществам данной системы следует отнести большую емкость системы и высокую степень гибкости, поскольку при необходимости можно вводить дополнительные фасеты и изменять их место в формуле. При изменении характера задач или характеристик объектов классификации разрабатываются новые фасеты или дополняются новыми признаками уже существующие фасеты без коренной перестройки структуры всего классификатора.

К недостаткам, характерным для данной системы, можно отнести сложность структуры и низкую степень заполненности системы.

В современных классификационных схемах часто одновременно используются оба метода классификации. Это снижает влияние недостатков методов классификации и расширяет возможность использования классификаторов в информационном обеспечении управления.

1.3. Правила классификации продукции

Принята *классификация* выпускаемой продукции по следующему ряду уровней (иерархическая *классификация*):

- семейство продуктов;
 - группа продуктов;
 - серия продуктов.

Однако эта *система классификации* не обеспечивает идентификацию любого выпускаемого изделия. Для каждой единицы продукта должны указываться следующие атрибуты (фасеты):

- код серии продукта;
 - *конфигурационные параметры*;
 - свойства.

Код серии продукта – алфавитно-цифровой код, однозначно идентифицирующий отдельный продукт. *Конфигурационные параметры* – свойства, значения которых могут быть различными в зависимости от *потребностей пользователей*. Свойства – предопределенные характеристики отдельных продуктов, которые не могут меняться для одного и того же продукта.

Допустимые варианты записи кода серии для различных продуктов показаны на рисунке 7.5.

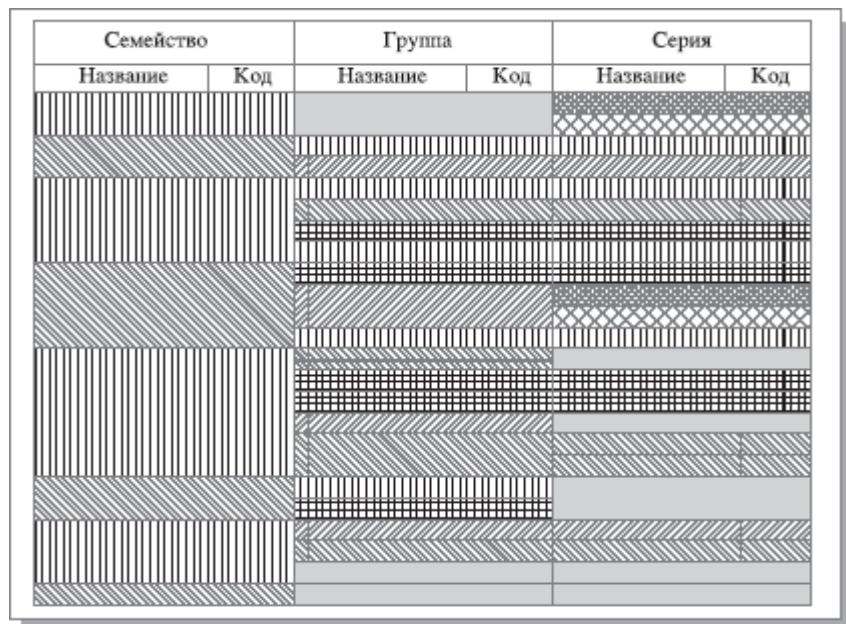


Рисунок 7.5 – Варианты записи кода серии продукта

Признаки фасета «Конфигурационные параметры» для одного семейства продуктов приведены в таблице 7.1.

Продукты и модификации	Характеристики	
	Общие для семейства	Специальные для отдельных моделей
Датчики разности давлений	Искробезопасное исполнение Взрывозащищенное исполнение	Предельно допустимое рабочее избыточное давление
Датчики абсолютного давления, избыточного давления, разрежения, давления-разрежения	Исполнение по материалам Климатическое исполнение Предел допускаемой основной погрешности Верхний предел измерений Код выходного сигнала Состав комплекта монтажных частей	Измеряемый параметр

Рассмотренные выше *системы классификации* хорошо приспособлены для организации поиска с целью последующей логической и арифметической обработки информации на компьютере, но лишь частично решают проблему содержательного поиска информации при принятии управленческих решений.

Для поиска показателей и документов по набору содержательных признаков используется информационный язык дескрипторного типа, который характеризуется совокупностью терминов (дескрипторов) и набором отношений между терминами.

Содержание документов или показателей можно достаточно полно и точно отразить с помощью списка ключевых слов – дескрипторов.

Дескриптор – это термин естественного языка (слово или словосочетание), используемый при описании документов или показателей, который имеет самостоятельный смысл и неделим без изменения своего значения.

Для того чтобы обеспечить точность и однозначность поиска с помощью дескрипторного языка, необходимо предварительно определить все постоянные отношения между терминами.

Все выделенные отношения явно описываются в систематическом словаре понятий – *тезаурусе*, который разрабатывается с целью проведения индексирования документов, показателей и информационных запросов.

1.4. Кодирование технико-экономической информации

Для полной формализации информации недостаточно простой *классификации*, поэтому проводят следующую процедуру – кодирование.

Кодирование – это процесс присвоения условных обозначений объектам и классификационным группам по соответствующей системе кодирования. Кодирование реализует перевод информации, выраженной одной системой знаков, в другую систему, то есть перевод записи на естественном языке в запись с помощью кодов.

Система кодирования – это совокупность правил обозначения объектов и группировок с использованием кодов.

Код – это условное обозначение объектов или группировок в виде знака или группы знаков в соответствии с принятой системой. Код базируется на определенном алфавите (некоторое множество знаков). Число знаков этого множества называется основанием кода. Различают следующие типы алфавитов: цифровой, буквенный и смешанный.

Код характеризуется следующими параметрами:

- длиной;
 - основанием кодирования;
 - структурой кода, под которой понимают распределение знаков по признакам и объектам *классификации*;
 - степенью информативности, рассчитываемой как частное от деления общего количества признаков на длину кода;
 - коэффициентом избыточности, который определяется как отношение максимального количества объектов к фактическому количеству объектов.
- К методам кодирования предъявляются определенные требования:
- код должен осуществлять идентификацию объекта в пределах заданного множества объектов *классификации*;
 - желательно предусматривать использование в качестве алфавита кода десятичных цифр и букв;
 - необходимо обеспечивать по возможности минимальную длину кода и достаточный резерв незанятых позиций для кодирования новых объектов без нарушения структуры *классификатора*.

Методы кодирования могут носить самостоятельный характер – *регистрационные методы кодирования*, или быть основанными на предварительной *классификации* объектов – *классификационные методы кодирования*.

Регистрационные методы кодирования бывают двух видов: *порядковый* и *серийно-порядковый*. В первом случае кодами служат числа *натурального ряда*. Каждый из объектов классифицируемого множества кодируется путем присвоения ему текущего порядкового номера. Данный метод кодирования обеспечивает довольно большую долговечность *классификатора* при незначительной избыточности кода. Этот метод обладает наибольшей простотой, использует наиболее короткие коды и лучше обеспечивает

однозначность каждого объекта *классификации*. Кроме того, он обеспечивает наиболее простое присвоение кодов новым объектам, появляющимся в процессе ведения *классификатора*. Существенным недостатком порядкового метода кодирования является отсутствие в коде какой-либо конкретной информации о свойствах объекта, а также сложность машинной обработки информации при получении итогов по группе объектов *классификации* с одинаковыми признаками.

В *серийно-порядковом методе кодирования* кодами служат числа *натурального ряда* с закреплением отдельных серий этих чисел (интервалов *натурального ряда*) за объектами *классификации* с одинаковыми признаками. В каждой серии, кроме кодов имеющихся объектов *классификации*, предусматривается определенное количество кодов для резерва.

Классификационные коды используют для отражения классификационных взаимосвязей объектов и группировок и применяются в основном для сложной логической обработки экономической информации. Группу классификационных систем кодирования можно разделить на две *подгруппы* в зависимости от того, какую *систему классификации* используют для упорядочения объектов: системы последовательного кодирования и параллельного кодирования.

Последовательные системы кодирования характеризуются тем, что они базируются на предварительной *классификации* по иерархической системе. Код объекта *классификации* образуется с использованием кодов последовательно расположенных подчиненных группировок, полученных при иерархическом методе кодирования. В этом случае код нижестоящей группировки образуется путем добавления соответствующего количества разрядов к коду вышестоящей группировки.

Параллельные системы кодирования характеризуются тем, что они строятся на основе использования фасетной *системы классификации* и коды группировок по фасетам формируются независимо друг от друга.

В параллельной системе кодирования возможны два варианта записи кодов объекта:

1. Каждый фасет и признак внутри фасета имеют свои коды, которые включаются в состав кода объекта. Такой способ записи удобно применять тогда, когда объекты характеризуются неодинаковым набором признаков. При формировании кода какого-либо объекта берутся только необходимые признаки.
2. Для определения групп объектов выделяется фиксированный набор признаков и устанавливается стабильный порядок их следования, то есть устанавливается фасетная формула. В этом случае не надо каждый раз указывать, значение какого из признаков приведено в определенных разрядах кода объекта.

Параллельный метод кодирования имеет ряд преимуществ. К достоинствам рассматриваемого метода следует отнести гибкость структуры кода, обусловленную независимостью признаков, из кодов которых строится код

объекта *классификации*. Метод позволяет использовать при решении конкретных технико-экономических и социальных задач коды только тех признаков объектов, которые необходимы, что дает возможность работать в каждом отдельном случае с кодами небольшой длины. При этом методе кодирования можно осуществлять группировку объектов по любому сочетанию признаков. Параллельный метод кодирования хорошо приспособлен для машинной обработки информации. По конкретной кодовой комбинации легко узнать, набором каких характеристик обладает рассматриваемый объект. При этом из небольшого числа признаков можно образовать большое число кодовых комбинаций. Набор признаков при необходимости может легко пополняться присоединением кода нового признака. Это свойство параллельного метода кодирования особенно важно при решении технико-экономических задач, состав которых часто меняется.

Наиболее сложными вопросами, которые приходится решать при разработке *классификатора*, являются выбор методов *классификации* и кодирования и выбор системы признаков *классификации*. Основой *классификатора* должны быть наиболее существенные признаки *классификации*, соответствующие характеру решаемых с помощью *классификатора* задач.

Важным вопросом является также правильный выбор последовательности использования признаков *классификации* по ступеням *классификации* при иерархическом методе *классификации*. Критерием при этом является статистика запросов к *классификатору*. В соответствии с этим критерием на верхних ступенях *классификации* в *классификаторе* должны использоваться признаки, к которым будут наиболее частые запросы. По этой же причине на верхних ступенях *классификации* выбирают наименьшее основание кода.

1.5. Понятие унифицированной системы документации

Основной компонентой внемашинного информационного обеспечения ИС является *система документации*, применяемая в процессе управления экономическим объектом. Под документом понимается определенная совокупность сведений, используемая при решении технико-экономических задач, расположенная на материальном носителе в соответствии с установленной формой.

Система документации – это совокупность взаимосвязанных форм документов, регулярно используемых в процессе управления экономическим объектом. Отличительной особенностью системы экономической документации является большое разнообразие видов документов.

Существующие *системы документации*, характерные для неавтоматизированных ИС, отличаются большим количеством разных типов форм документов, большим объемом потоков документов и их запутанностью, дублированием информации в документах и работ по их обработке и, как следствие, низкой достоверностью получаемых результатов. Для того чтобы упростить *систему документации*, используют следующие два подхода:

- проведение унификации и стандартизации документов;
- введение безбумажной технологии, основанной на использовании *электронных документов* и новых информационных технологий их обработки.

Унификация документов выполняется путем введения единых форм документов. Таким образом, вводится единообразие в наименования показателей, единиц измерения и терминов, в результате чего получается унифицированная *система документации*.

Унифицированная система документации (УСД) – это рационально организованный комплекс взаимосвязанных документов, который отвечает единым правилам и требованиям и содержит информацию, необходимую для управления некоторым экономическим объектом. По уровням управления они делятся на межотраслевые *системы документации*, отраслевые и *системы документации* локального уровня, т. е. обязательные для использования в рамках предприятий или организаций.

Любой тип УСД должен удовлетворять следующим требованиям:

- документы, входящие в состав УСД, должны разрабатываться с учетом их использования в системе взаимосвязанных ИС;
- УСД должна содержать полную информацию, необходимую для оптимального управления тем объектом, для которого разрабатывается эта система;
- УСД должна быть ориентирована на использование средств вычислительной техники для сбора, обработки и передачи информации;
- УСД должна обеспечить информационную совместимость ИС различных уровней;
- все документы, входящие в состав разрабатываемой УСД, и все реквизиты-признаки в них должны быть закодированы с использованием международных, общесистемных или локальных *классификаторов*.

2. Внутримашинное информационное обеспечение

Внутримашинное информационное обеспечение включает макеты (экранные формы) для ввода первичных данных в компьютер или вывода результатной информации, и структуры *информационной базы*: входных, выходных файлов, *базы данных*.

2.1. Проектирование экранных форм электронных документов

Под *электронными формами документов* понимается не изображение бумажного документа, а изначально электронная (безбумажная) технология работы; она предполагает появление бумажной формы только в качестве твердой копии документа.

Электронная форма документа (ЭД) – это страница с пустыми полями, оставленными для заполнения пользователем. Формы могут допускать

различный тип *входной информации* и содержать *командные кнопки*, переключатели, выпадающие меню или списки для выбора.

Создание форм *электронных документов* требует использования специального программного обеспечения. На рисунке 7.6 приведены основные типовые элементы *электронного документа*, использование которых предусмотрено в большинстве программ автоматизации проектирования *электронных документов*.

Технология обработки *электронных документов* требует использования специализированного программного обеспечения – программ управления документооборотом, которые зачастую встраиваются в ИС.

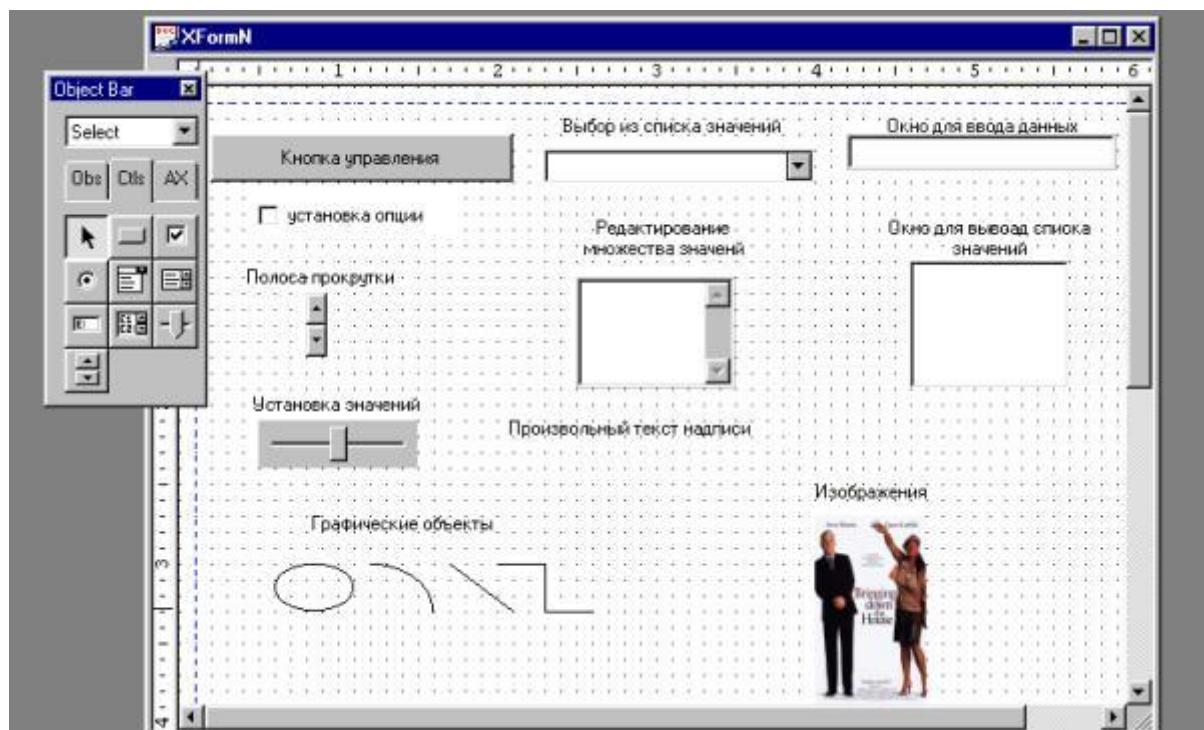


Рисунок 7.6 – Элементы электронного документа

Проектирование форм электронных документов, т. е. создание шаблона формы с помощью программного обеспечения проектирования форм, обычно включает в себя выполнение следующих шагов:

- **создание структуры ЭД** – подготовка внешнего вида с помощью графических средств проектирования;
- **определение содержания формы ЭД**, т. е. выбор способов, которыми будут заполняться поля. Поля могут быть заполнены вручную или посредством выбора значений из какого-либо списка, меню, базы данных;
- **определения перечня макетов экранных форм** – по каждой задаче проектировщик анализирует «постановку» каждой задачи, в которой приводятся перечни используемых входных документов с оперативной и постоянной информацией и документов с результатной информацией;

- **определение содержания макетов** – выполняется на основе анализа состава реквизитов первичных документов с постоянной и оперативной информацией и результатных документов.
- Работа заканчивается программированием разработанных макетов *экранных форм* и их апробацией.

2.2. Информационная база и способы ее организации

Основной частью внутримашинного информационного обеспечения является *информационная база*.

Информационная база (ИБ) – это совокупность данных, организованная определенным способом и хранимая в памяти компьютера в виде файлов, с помощью которых удовлетворяются информационные потребности управлческих процессов и решаемых задач.

Все файлы ИБ можно классифицировать по следующим признакам:

- по этапам обработки (входные, базовые, результатные);
- по типу носителя (на промежуточных носителях – гибких магнитных дисках и на основных носителях – жестких магнитных дисках, магнитооптических дисках и др.);
- по составу информации (файлы с оперативной информацией и файлы с постоянной информацией);
- по назначению (по типу функциональных подсистем);
- по типу логической организации (файлы с линейной и иерархической структурой записи, реляционные, табличные);
- по способу физической организации (файлы с последовательным, индексным и прямым способом доступа).

Входные файлы создаются с первичных документов для ввода данных или обновления базовых файлов.

Файлы с *результатной информацией* предназначаются для вывода ее на печать или передачи по каналам связи и не подлежат долговременному хранению.

К числу *базовых файлов*, хранящихся в *информационной базе*, относят основные, рабочие, промежуточные, служебные и архивные файлы.

Основные файлы должны иметь однородную структуру записей и могут содержать записи с оперативной и условно-постоянной информацией. *Оперативные файлы* могут создаваться на базе одного или нескольких входных файлов и отражать информацию одного или нескольких первичных документов. *Файлы с условно-постоянной информацией* могут содержать справочную, расценочную, табличную и другие виды информации, изменяющейся в течение года не более чем на 40 %, а, следовательно, имеющие коэффициент стабильности не менее 0,6.

Файлы *со справочной информацией* должны отражать все характеристики элементов материального производства (материалы, сырье, основные фонды, трудовые ресурсы и т. п.). Как правило, справочники содержат информацию *классификаторов* и дополнительные сведения об элементах

материальной сферы, например, о ценах. Нормативно-расценочные файлы должны содержать данные о нормах расхода и расценках на выполнение операций и услуг. Табличные файлы содержат сведения об экономических показателях, считающихся постоянными в течение длительного времени (например, процент удержания, отчисления и пр.). Плановые файлы содержат плановые показатели, хранящиеся весь плановый период.

Рабочие файлы создаются для решения конкретных задач на базе основных файлов путем выборки части информации из нескольких основных файлов с целью сокращения времени обработки данных.

Промежуточные файлы отличаются от рабочих файлов тем, что они образуются в результате решения экономических задач, подвергаются хранению с целью дальнейшего использования для решения других задач. Эти файлы, также, как и рабочие файлы, при высокой частоте обращений могут быть также переведены в категорию основных файлов.

Служебные файлы предназначаются для ускорения поиска информации в основных файлах и включают в себя справочники, индексные файлы и каталоги.

Архивные файлы содержат ретроспективные данные из основных файлов, которые используются для решения аналитических, например, прогнозных, задач. Архивные данные могут также использоваться для восстановления *информационной базы* при разрушениях.

Организация хранения файлов в *информационной базе* должна отвечать следующим требованиям:

- полнота хранимой информации для выполнения всех функций управления и решения экономических задач;
- целостность хранимой информации, т. е. обеспечение непротиворечивости данных при вводе информации в ИБ;
- своевременность и одновременность обновления данных во всех копиях данных;
- гибкость системы, т. е. адаптируемость ИБ к изменяющимся информационным потребностям;
- реализуемость системы, обеспечивающая требуемую степень сложности структуры ИБ;
- релевантность ИБ, под которой подразумевается способность системы осуществлять поиск и выдавать информацию, точно соответствующую запросам пользователей;
- удобство языкового интерфейса, позволяющее быстро формулировать запрос к ИБ;
- разграничение прав доступа, т. е. определение для каждого пользователя доступных типов записей, полей, файлов и видов операций над ними.

Существуют следующие способы организации ИБ: совокупность локальных файлов, поддерживаемых функциональными пакетами прикладных программ, и интегрированная база данных, основывающаяся на использовании универсальных программных средств загрузки, хранения, поиска и ведения данных, т. е. системы управления базами данных (СУБД).

Локальные файлы вследствие специализации структуры данных под задачи обеспечивают, как правило, более быстрое время обработки данных. Однако недостатки организации локальных файлов, связанные с большим дублированием данных в информационной системе и, как следствие, несогласованностью данных в разных приложениях, а также негибкостью доступа к информации, перекрывают указанные преимущества. Поэтому организация локальных файлов может применяться только в специализированных приложениях, требующих очень высокой скорости реакции при импорте необходимых данных.

Интегрированная ИБ, т. е. база данных (БД) – это совокупность взаимосвязанных, хранящихся вместе данных при такой минимальной избыточности, которая допускает их использование оптимальным образом для множества приложений.

Централизация управления данными с помощью СУБД обеспечивает совместимость этих данных, уменьшение синтаксической и семантической избыточности, соответствие данных реальному состоянию объекта, разделение хранения данных между пользователями и возможность подключения новых пользователей. Но централизация управления и интеграция данных приводят к проблемам другого характера: необходимости усиления контроля вводимых данных, необходимости обеспечения соглашения между пользователями по поводу состава и структуры данных, разграничения доступа и секретности данных.

Основными способами организации БД являются создание централизованных и распределенных БД. Основным критерием выбора способа организации ИБ является достижение минимальных трудовых и стоимостных затрат на проектирование структуры ИБ, программного обеспечения системы ведения файлов, а также на перепроектирование ИБ при возникновении новых задач.

ЛЕКЦИЯ № 8. СОПРОВОЖДЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

1. Понятие сопровождения информационных систем.
2. Варианты сопровождения информационных систем.
3. Задачи сопровождения информационных систем.
4. Сценарий и структура ИТ-сопровождения.

1. Понятие сопровождения информационных систем

ИС является одним из наиболее гибких видов продукции, который часто подвергается изменениям в течение всего времени его использования. Для сохранения и повышения качества работы системы нужно правильно регламентировать процесс сопровождения его соответствующим тестированием и контролем качества программного обеспечения (ПО).

Сопровождение (поддержка) программного обеспечения – процесс улучшения, оптимизации и устранения дефектов программного обеспечения (ПО) после передачи в эксплуатацию.

Сопровождение ПО – это одна из фаз жизненного цикла программного обеспечения, следующая за фазой передачи ПО в эксплуатацию. В ходе сопровождения в программу вносятся изменения для того, чтобы исправить обнаруженные в процессе использования дефекты и недоработки, а также для добавления новой функциональности, с целью повышения удобства использования и применимости ПО.

Сопровождение программного обеспечения стандартизовано, имеются национальные стандарты Российской Федерации, идентичные международным:

ISO/IEC 12207:2008 System and software engineering – Software life cycle processes, ГОСТ Р ИСО/МЭК 12207-2010 «Национальный стандарт Российской Федерации. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»;

ISO/IEC 14764:99 Information technology – Software maintenance, ГОСТ Р ИСО/МЭК 14764-2002 «Государственный стандарт Российской Федерации. Информационная технология. Сопровождение программных средств».

Сопровождение включает в себя комплекс следующих мероприятий:

- *техническая поддержка* (восстановление работоспособности системы в фиксированной конфигурации);
- *изменение конфигурации ИТ-комплекса* (инсталляция нового ПО и оборудования, его конфигурирование и настройка);
- *проведение технических консультаций для персонала*;
- *оптимизация работы ИТ-системы* (например, повышение общей производительности, достижение оптимального использования дискового пространства систем хранения, сокращение окна резервного копирования и послеаварийного восстановления данных);
- *ведение технической документации* (паспортов комплекса и регламентов работ по восстановлению его работоспособности и профилактике);

- *профилактические мероприятия*, включая определение соответствия, используемого ПО, целям и задачам заказчика на данный момент и, при необходимости, установка новых версий ПО и заплат.

Кроме того, в сфере сопровождения лежит вопрос обеспечения информационной безопасности. Комплексный подход в данном случае состоит в предупреждении неожиданных критических ситуаций, проведении анализа политики безопасности, проверки на соответствие стандартам, выработки рекомендаций.

В случае отсутствия непрерывного сопровождения система за небольшой промежуток времени заваливается, происходят постоянные аппаратные сбои, что может привести к потере данных или происходит потеря актуальности данных.

Даже самая совершенная система защиты не устраниет полностью риск отказа. Даже непродолжительный простой информационной системы способен нанести серьезный урон бизнесу тех организаций, чья деятельность во многом зависит от ИТ.

2. Варианты сопровождения информационных систем

1. По заявкам

Обслуживание ИС производится структурным подразделением заказчика в случае возникновения аварийных ситуаций (отказ оборудования, отказ программного обеспечения) и заключается в разовом обращении к специалистам. При этом заказчику необходимо самостоятельно выявить неисправный элемент и принять решение об организации его восстановления, что может привести к длительным срокам восстановления. Сюда же накладывается время реакции на обращение к специалистам, так как вызов неплановый.

2. Локальное обслуживание по договору

Установление договорных отношений с компанией на обслуживание отдельных компонент системы, например, технических и/или программных. Наиболее распространенной услугой является обслуживание ККМ в различных временных режимах со сроками реакции, например, до 4-х, 8-и, 24-х часов с момента регистрации вызова.

Качество услуг обеспечивается созданием в компании так называемого «страхового фонда»: мобильной инженерной группы и материальных ресурсов (подменного оборудования, запчастей, расходных материалов), ориентированных на устранение экстренных ситуаций. Таким образом, заказчик, не производя полного объема затрат на создание такой инфраструктуры, фактически в состоянии воспользоваться ею в нужный момент.

3. Регламентированное сопровождение

Данный уровень сопровождения характеризуется передачей компании ряда активных функций по обеспечению жизнедеятельности ИС. Здесь объем услуг определяется основной целью – обеспечение бесперебойного

функционирования ИС – и включает комплекс профилактических мероприятий, направленных на предупреждение возникновения сбоев: мониторинг состояния ресурсов и параметров компонент ИС; планирование штатных административных процедур; администрирование (в том числе удаленное) серверов, БД, пользователей, сетей и сетевых ресурсов; организация систем резервного копирования и систем безопасности.

4. Аудит/консультирование

Обеспечение жизнедеятельности ИС не сводится исключительно к сопровождению ее текущего состояния. Как любой живой бизнес, ИС должна отражать его состояние в изменении и развитии. Процесс развития должен соответствовать развитию направлений основного бизнеса, а меры, его обеспечивающие, должны быть реализованы в рамках единой стратегии с обязательным условием сохранения преемственности реализованных решений.

Активные меры по планированию развития компонент ИС заключаются в обеспечении достаточности ресурсов и прогнозирования их изменения на основании аудита информационных решений. В рамках аудита выполняются планирование мероприятий по модернизации аппаратных и программных компонент ИС; разработка поэтапных планов и методов развития и модернизации компьютерных систем; разработка временных графиков и оценка стоимости приобретения оборудования и т. д.

Проектам развития предшествует стадия технической и коммерческой экспертизы, которая уже традиционно выполняется с привлечением консультантов специализированных и консалтинговых компаний.

5. Полное сопровождение.

Передача функций по обеспечению жизненного цикла ИС как единого технологического комплекса компании на принципах аутсорсинга, что включает в себя передачу исполнителю функций обеспечения ресурсами, создания компонент ИС, внедрения прикладных задач, проведения обучения персонала, выполнения информационных услуг и т. д.

В результате, ИС актуализируется и по функциональным возможностям, и по качеству решения отдельных задач.

3. Задачи сопровождения информационных систем

Сопровождение информационных систем (ИС) состоит из двух больших и разноплановых задач.

Первая задача – эксплуатация информационной системы. Решение этой задачи начинается с установки прикладного программного обеспечения (ПО) в определенном программно-аппаратном окружении и настройкой ПО в соответствии с документацией разработчика таким образом, чтобы обеспечить максимальную надежность и производительность работы приложения.

Вторая задача – внесение изменений в информационную систему. Изменения могут включать донастройки тиражируемого ПО или доработки заказного ПО. Донастройки и доработки, как правило, требуют привлечения

консультантов по бизнес-процессам, а также программистов, обладающих необходимыми компетенциями. То есть, когда возникает потребность в «подгонке» ИС под новые условия и требования бизнес-процессов.

Задачи сопровождения рассматриваются с точки зрения потребителя применительно к программной среде, которая включает в себя данные и механизмы их обработки. Все задачи сопровождения разбиты на три класса: *развитие ИС, корректирующее сопровождение и сопровождение данных* (рисунок 8.1).

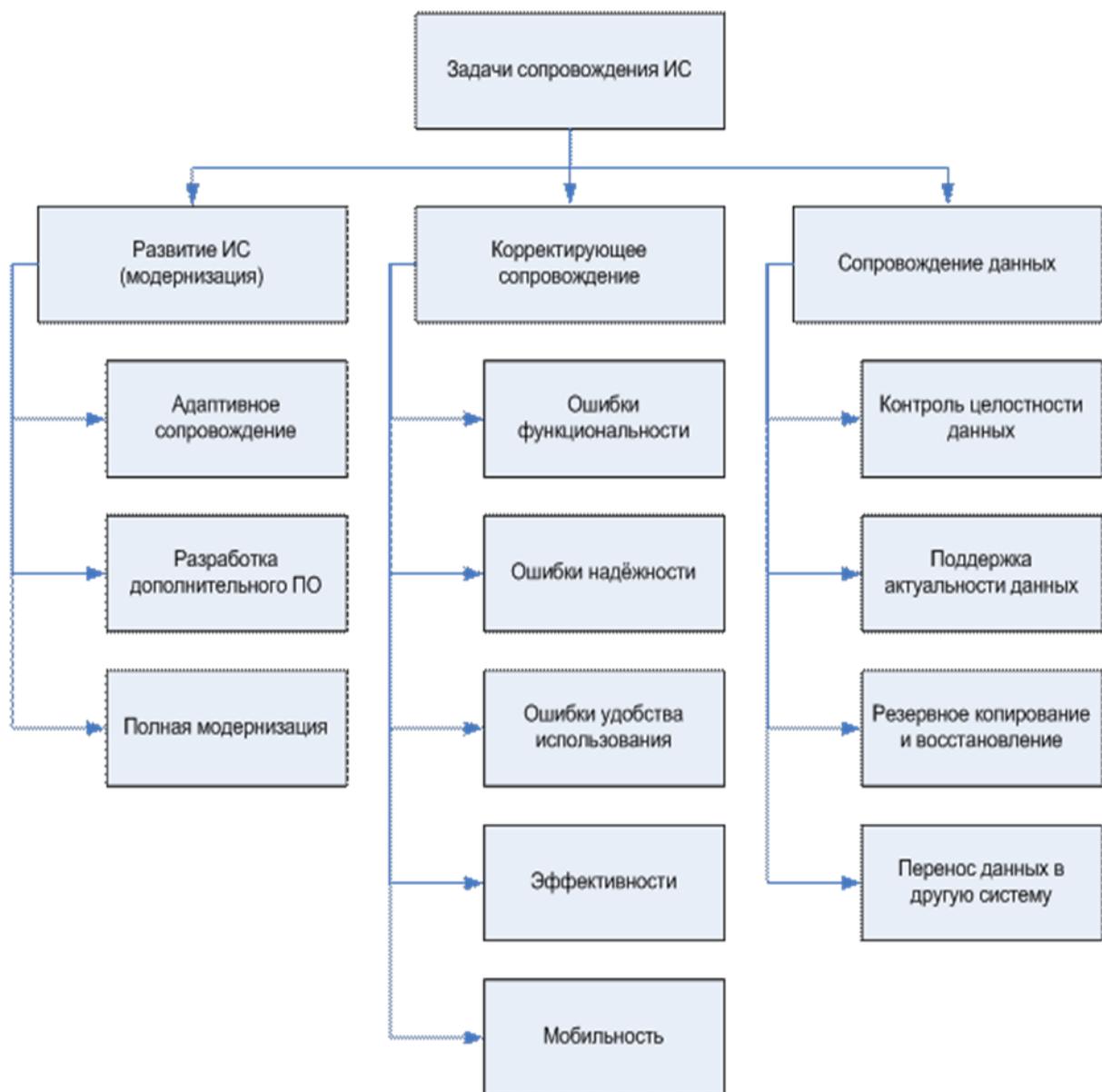


Рисунок 8.1 – Классификации задач сопровождения ПО ИС

Развитие ИС предполагает частичную или полную модернизацию. В связи с этим, развитие ИС можно разделить на доработку ИС, разработку дополнительного ПО и замену ИС на более современную и функциональную.

Адаптивное сопровождение – это доработка программного продукта после поставки, позволяющее адаптировать его к новым условиям эксплуатации.

Полная модернизация является наиболее дорогостоящим этапом в развитии ИС. Основным фактором, объясняющим стремление сменить функционирующую ИС, является моральное устаревание системы.

В случае разработки дополнительного ПО очевидны немалые преимущества: функционал дорабатывается под нужды персонала, необходимые требования будут квалифицированно оценены и выполнены, интерфейс и принципы работы останутся неизменными, доработка будет производиться с использованием существующих средств и методов.

Корректирующее сопровождение направлено на выявление и устранение несоответствий и ошибок после поставки программного продукта.

Информационная система, как и любое ПО, не всегда даёт желаемый результат работы.

Применительно к сопровождению ИС ошибка – это искажение кода программы или искажение данных, которые в ходе функционирования этой программы могут вызвать отказ или снижение эффективности функционирования. Под отказом ИС в общем случае понимают событие, заключающееся в нарушении работоспособности объекта. В качестве показателя степени тяжести ошибки, позволяющего дать количественную оценку тяжести проявления последствий ошибки, можно использовать условную вероятность отказа программного обеспечения при проявлении ошибки. Оценку степени тяжести ошибки как условной вероятности возникновения отказа можно производить согласно ГОСТ 28195-89 «Оценка качества программных средств. Общие положения», используя метрики и оценочные элементы, характеризующие устойчивость программного обеспечения. При этом оценку необходимо производить для каждой ошибки в отдельности, а не для всей ИС.

Возникающие ошибки ИС предлагается разделить на ошибки функциональности, надёжности, удобства использования, эффективности, мобильности. При этом сопровождение каждой из них может быть, как реактивным – в виде реакция на выявленные ошибки, так и профилактическим, которое применяется в особо ответственных модулях и системах во избежание возможных (ещё не возникших) проблем.

Сопровождение данных. Важной спецификой корпоративных ИС является значительное превышение продолжительности жизненного цикла (ЖЦ) данных над продолжительностью ЖЦ программной среды, технологий обработки, бизнес-логики и т. д. Поэтому поддержка данных выделена в отдельный класс задач, состоящий из контроля целостности данных, поддержки актуальности данных и резервного копирования и восстановления, а также переноса данных из одной системы в другую.

Целостность информации определяется как состояние информации, при котором её изменение осуществляется только преднамеренно субъектами, имеющими на него право, либо таковое отсутствует.

Актуальность – это свойство данных в указанный момент времени адекватно отображать состояние объектов предметной области.

Если целостность данных поддерживается с помощью СУБД, то наиболее популярным средством поддержания актуальности является механизм создания резервных копий.

На сегодняшний день существует коммерческое и открытое ПО, позволяющее контролировать точность и неизменность версий файлов. Большинство инструментов мониторинга за актуальностью системы основаны на сравнении текущего состояния системы и состояния, которому можно доверять. Это состояние часто называют базовой целостностью, получаемой с заведомо исправной системы.

В отдельную группу задач сопровождения данных вынесен перенос данных в другую систему. Проблема связана с сохранением существующих связей и семантики. Процесс переноса данных не должен порождать дублирование записей, равно как и их потерю.

4. Сценарий и структура ИТ-сопровождения

Сценарий – некоторая последовательность действий, иллюстрирующая поведение системы.

В разработке программного обеспечения и системном проектировании сценарии использования представляют собой описание поведения системы, которым она отвечает на внешние запросы. Сценарии использования описывают то, «кто» и «что» может сделать с рассматриваемой системой. Методика сценариев использования применяется для выявления требований к поведению системы.

Сценарии использования сосредотачиваются на том, как достигнуть конкретной цели или задачи, необходимой пользователю, как внешнему агенту, от данной системы. Для большинства разрабатываемых программ это означает необходимость разработки множества сценариев для определения общего набора решаемых задач. Степень детализации сценариев использования зависит, прежде всего, от сложности и текущей стадии разрабатываемого проекта. В процессе разработки сценарии могут усложняться и претерпевать значительные изменения.

Сценарии использования описывают, что и как конкретно должна делать система в ответ на запросы пользователей, не касаясь при этом непосредственно дизайнов интерфейсов.

Договор на сопровождение. Информационная поддержка, осуществляемая для обслуживания системы автоматизации заказчика, становится все более популярной. Для небольших и средних фирм это – разумное с точки зрения экономии решение. Оставаясь в информационном поле, заказчик не думает, как решить проблему своими местными кадровыми ресурсами. Эту задачу можно доверить сторонним специалистам. А договор, заключенный между клиентом и исполнителем, будет гарантией получения качественной и

своевременной услуги. Что касается оперативности, это тоже прописывается в договоре.

Подписанный договор, с каждым пунктом которого согласен исполнитель и заказчик, предусматривает:

- передачу сведений и консультации по телефону;
- в круглосуточном режиме получение информации и методических советов с помощью интернет-форума;
- фиксированное время удаленной поддержки через интернет с поправкой на то, что клиент не может получить подключение сразу же за обращением: исполнитель оговаривает оптимальную временную вилку, гарантируя соблюдение ее параметров.

Заказчик такой услуги может рассчитывать на скидку, фиксированный процент которой будет в договоре. Скидка распространяется на оборудование, услуги и программное обеспечение, заявленные в прайс-листе исполнителя. При этом следует иметь в виду, что исключение составят те позиции и распродажи, которые не заявлены в прайсе.

Принято выделять ***несколько линий сопровождения*** (структура приведена на примере внешнего сопровождения ПО):

0 линия (call-center, информационный центр, горячая линия) – обработка телефонных обращений от клиентов, передача обращений техническим специалистам (***1-я линия сопровождения***)

1 линия (инженер по сопровождению, инженер технической поддержки, support engineer) – консультация/настройка/устранение ошибок в работе ПО/наполнение базы знаний, составление мануалов, т. е. работа консультантов:

- прием обращений от конечных пользователей по телефону и электронной почте;
- регистрация, классификация и расстановка приоритетов запросов на основании принятых обращений;
- поиск известного решения в базе данных типовых вопросов и ответов, в случае если решение найдено;
- определение и назначение исполнителей запроса;
- ведение базы знаний по инцидентам и типовым вопросам и ответам.

2 линия (инженер по сопровождению, инженер технической поддержки, support engineer) функциональное сопровождение/проектная деятельность на этапе запуска ПО на машинах заказчика, т. е. работа специалистов по предметной области:

- интервьюирование инициатора запроса с целью более четкого понимания внештатной ситуации и предпосылок к ее возникновению;
- восстановление работоспособности системы с использованием удаленного доступа, при помощи консультации по телефону или электронной почте;
- консультирование пользователей с целью устранения внештатной ситуации и недопущения повторения подобных ситуаций в будущем (в случае, если причиной внештатной ситуации послужили некорректные, с точки зрения системы, действия пользователей);

- выработка и реализация перечня мероприятий для устранения последствий внештатной ситуации;
- оказание помощи пользователям при необходимости реализации нетиповых операций и бизнес-процессов с целью получения заданного результата.

3 линия (инженер по сопровождению, инженер технической поддержки, support engineer) – системное сопровождение/проектная деятельность на этапе запуска ПО на оборудовании заказчика, т. е. работа разработчиков информационной системы:

- интервьюирование инициатора запроса с целью более четкого понимания требований (возникшей ошибки);
- подготовка краткого описания сути предлагаемых изменений и его согласование с инициатором запроса;
- при необходимости проведения серьезной модернизации, составление технических заданий и согласование стоимости доработки системы;
- реализация изменений (устранение ошибки);
- тестирование;
- установка обновлений у Заказчика.

Заказчик страдает, когда версия приложения с доработками или измененными настройками не работает должным образом, и сотрудники заказчика не могут выполнять свою работу. А исполнитель, заинтересованный сдать доработанное ПО, пытается поправить ситуацию и дорабатывает «на ходу» вместо того, чтобы «откатить» систему на предыдущую версию. И это приводит к еще большим простоям и потерям. В связи с этим, для эффективного сопровождения следует внедрять центр компетенции на стороне клиента.

Центр компетенции должен содержать следующие функциональные блоки:

Обязательные функциональные блоки

Блок функциональной поддержки. Поддержка конечных пользователей по телефону, обработка сообщений пользователей, работа с базой решений компании-поставщика, анализ возникновения проблем, обработка запросов-рекламаций в адрес компании-поставщика, администрирование пользователей ИС.

Блок технической поддержки системы. Решение технических проблем на уровне ландшафта системы (комплекс технических средств и правил, имеющих место быть в компании), анализ возникновения технических проблем, обработка запросов-рекламаций в адрес компании-поставщика, инсталляция и апгрейд компонентов ИС, администрирование системного ландшафта.

Блок управления договорами и лицензиями. Заключение договоров на новые компоненты ИС, новые лицензии, а также на поддержку со стороны компании-поставщика.

Блок аудита лицензий и масштабируемости системы. Определение необходимого объёма лицензий, аудит текущего объёма используемых лицензий, оценка результатов измерения системного ландшафта,

взаимодействие с компанией-поставщиком по результатам аудита системного ландшафта.

Рекомендуемые функциональные блоки

Блок управления информационным обеспечением. Распространение информационных материалов среди пользователей ИС, обзор информационных источников компаний-поставщиков, организация доступа к ним для заинтересованных лиц, подготовка специфической информации по запросам, организация информационных мероприятий.

Блок управления дополнительными разработками. Управление запросами пользователей на проведение дополнительных разработок для ИС, разработка новых отчётов и программ, модификация стандартных объектов ИС, контроль проведённых модификаций, разработка интерфейсов между подсистемами ИС.

Блок управления внутренним маркетингом системы. Формирование содержания для материалов разъяснительного характера относительно проведённых внедрений и функционирующих решений, проведение презентаций, организация информационных мероприятий для будущих пользователей ИС, организация референт визитов.

Блок консалтинга в части развития и поддержки системы. Проведение управленческого консалтинга, разработка стандартов и шаблонов для ИС, консультирование по управлению проектами, консультирование по прикладным компонентам ИС.

Блок обучения. Разработка специальных программ обучения, проведение стандартного обучения групп пользователей по учебным материалам компаний-поставщиков, обучение проектных команд.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Грекул, В. И. Проектирование информационных систем: курс лекций / В. И. Грекул. – Москва: Интuit НОУ, 2016. – 570 с. – ISBN 978-5-9556-0033-8.
2. Сухомлинов, А. И. Анализ и проектирование информационных систем: учебное пособие для студентов, обучающихся по направлению 09.03.03 «Прикладная информатика» / А. И. Сухомлинов. – 2-е издание, испр. и доп. – Владивосток: Издательство Дальневосточного федерального университета, 2021. – 1 CD-ROM; [360 с.]. – ISBN 978-5-7444-5003-8.
3. Проектирование информационных систем (на примере методов структурного системного анализа): учебное пособие / О. Г. Инюшкина. – Екатеринбург: «Форт-Диалог Исеть», 2014. – 240 с. – ISBN 978-5-91128-072-7.