



Учебно-методическое пособие по дисциплине «Проектирование информационных систем» предназначено для изучения основных понятий проектирования информационных систем, видов проектирования ИС, а также различных методологий моделирования предметной области. Рассматриваются следующие вопросы: каноническое проектирование ИС, типовое проектирование ИС, полная бизнес-модель компании, шаблоны организационного бизнес-моделирования, построения организационно-функциональной модели компании, инструментальные средства организационного моделирования, процессные потоковые модели, основные элементы процессного подхода, выделение и классификация процессов, структурная модель предметной области, функционально-ориентированные и объектно-ориентированные методологии описания предметной области. Рекомендовано для студентов высших учебных заведений обучающихся по специальностям в области информационных систем и технологий.



Сapieв Азамат Заурбиевич - доцент кафедры информационной безопасности и прикладной информатики Майкопского государственного технологического университета. За время работы в Университете им опубликовано более 90 научных работ по информатике, информационным системам и технологиям, информационным рынкам технической информации и региональной экономики.



FOR AUTHOR USE

Азамат Сapieв

Проектирование информационных систем

Учебно-методическое пособие по дисциплине «Проектирование информационных систем»

Азамат Сапиев

Проектирование информационных систем

FOR AUTHOR USE ONLY

FOR AUTHOR USE ONLY

Азамат Сапиев

Проектирование информационных систем

**Учебно-методическое пособие по дисциплине
«Проектирование информационных систем»**

FOR AUTHOR USE ONLY

LAP LAMBERT Academic Publishing RU

Imprint

Any brand names and product names mentioned in this book are subject to trademark, brand or patent protection and are trademarks or registered trademarks of their respective holders. The use of brand names, product names, common names, trade names, product descriptions etc. even without a particular marking in this work is in no way to be construed to mean that such names may be regarded as unrestricted in respect of trademark and brand protection legislation and could thus be used by anyone.

Cover image: www.ingimage.com

Publisher:

LAP LAMBERT Academic Publishing

is a trademark of

International Book Market Service Ltd., member of OmniScriptum Publishing Group

17 Meldrum Street, Beau Bassin 71504, Mauritius

Printed at: see last page

ISBN: 978-620-2-92128-2

Copyright © Азамат Сапиев

Copyright © 2020 International Book Market Service Ltd., member of OmniScriptum Publishing Group

FOR AUTHOR USE ONLY

СОДЕРЖАНИЕ

1. ОСНОВНЫЕ ПОНЯТИЯ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ	3
2. ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИС	16
3. ОРГАНИЗАЦИЯ РАЗРАБОТКИ ИС	27
3.1. Каноническое проектирование ИС	27
3.2. Типовое проектирование ИС	36
4. АНАЛИЗ И МОДЕЛИРОВАНИЕ ФУНКЦИОНАЛЬНОЙ ОБЛАСТИ ВНЕДРЕНИЯ ИС	41
4.1. Полная бизнес-модель компании	41
4.2. Шаблоны организационного бизнес-моделирования	47
4.3. Построения организационно-функциональной модели компании	55
4.4. Инструментальные средства организационного моделирования	60
5. СПЕЦИФИКАЦИЯ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ К ИС	63
5.1. Процессные потоковые модели	63
5.2. Основные элементы процессного подхода	68
5.3. Выделение и классификация процессов	70
6. МЕТОДОЛОГИИ МОДЕЛИРОВАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ	81
6.1. Структурная модель предметной области	81
6.2. Функционально-ориентированные и объектно-ориентированные методологии описания предметной области	89
СПИСОК ЛИТЕРАТУРЫ	107

FOR AUTHOR USE ONLY

1. ОСНОВНЫЕ ПОНЯТИЯ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Информация в современном мире превратилась в один из наиболее важных ресурсов, а информационные системы (ИС) стали необходимым инструментом практически во всех сферах деятельности.

Разнообразие задач, решаемых с помощью 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 систем. В зависимости от объема решаемых задач, используемых технических средств, организации функционирования, информационные системы делятся на ряд групп (классов) (рис. 1.1.).

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 наименований, 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.



Рис. 1.1. Классификация информационных систем

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

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

Автоматизированные ИС предполагают участие в процессе обработки информации и человека, и технических средств, причем 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 пр.)

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 ИС.

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

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.

ИС управления технологическими процессами (ТП) - 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.

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, считается уже пройденным этапом для многих предприятий.

Существует *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, зарплата, 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. Задача подобных информационных систем - интеграция новых сведений в организацию и помощь в обработке бумажных документов.

Информационные системы уровня менеджмента - используются работниками среднего управленческого звена для мониторинга, контроля, принятия решений и администрирования. Основные функции этих информационных систем:

- 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 - 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 предоставить информацию из многих источников. Некоторые стратегические системы обладают ограниченными аналитическими возможностями.

С точки зрения **программно-аппаратной реализации** можно выделить ряд типовых архитектур 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.

Индустрия разработки автоматизированных информационных систем управления зародилась в 1950-х - 1960-х годах и к концу века приобрела вполне законченные формы.

На первом этапе основным подходом в *проектировании ИС* был метод "снизу-вверх", когда система создавалась как набор приложений, наиболее важных в данный момент для поддержки деятельности предприятия. Основной целью этих проектов было не создание тиражируемых продуктов, а обслуживание текущих потребностей конкретного учреждения. Такой подход отчасти сохраняется и сегодня. 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 проблему.

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 вызвали недовольство руководителей и пользователей системы.

Следующий этап 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. Системы начали проектироваться "сверху-вниз", т.е. в предположении, что одна программа должна удовлетворять потребности многих пользователей.

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 обычно 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 насущная необходимость формирования новой методологии построения информационных систем.

Цель такой методологии 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. Основными задачами, решению которых должна способствовать методология проектирования корпоративных ИС, являются следующие:

- обеспечивать создание корпоративных ИС, отвечающих целям и задачам организации, а также предъявляемым требованиям по автоматизации деловых процессов заказчика;
- гарантировать создание системы с заданным качеством в заданные сроки и в рамках установленного бюджета проекта;
- поддерживать удобную дисциплину сопровождения, модификации и наращивания системы;
- обеспечивать преемственность разработки, т.е. использование в разрабатываемой ИС существующей информационной инфраструктуры организации (задела в области информационных технологий).

Внедрение методологии должно приводить к снижению сложности процесса создания ИС за счет полного и точного описания этого процесса, а также применения современных методов и технологий создания ИС на всем жизненном цикле ИС - от замысла до реализации.

Проектирование ИС охватывает три основные области:

- 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 информационных систем 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:

- 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 репозитории проекта. Создание моделей, их контроль, преобразование и предоставление в коллективное пользование осуществляется с использованием специальных программных инструментов - CASE-средств.

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

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

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. 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 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 и 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.

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 (платформ) и операционной системы (операционных систем). 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 3-уровневая 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 должной производительности использоваться параллельные серверы баз данных (например, Oracle Parallel Server, DB2 UDB и т.п.).

Этап проектирования завершается разработкой технического проекта ИС.

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 системы 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 наработки 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.

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.

FOR AUTHOR USE ONLY

2. ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИС

Методология проектирования информационных систем описывает процесс создания и сопровождения систем в виде *жизненного цикла* (ЖЦ) ИС, представляя его как некоторую последовательность стадий и выполняемых на них процессов. Для каждого этапа определяются состав и последовательность выполняемых работ, получаемые результаты, методы и средства, необходимые для выполнения работ, роли и ответственность участников и т.д. Такое формальное описание ЖЦ ИС позволяет спланировать и организовать процесс коллективной разработки и обеспечить управление этим процессом.

Жизненный цикл ИС 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. *Модель жизненного цикла* - 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*:

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

- **Поэтапная модель с промежуточным контролем.** 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.

• **Спиральная модель.** 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).



Рис. 2.1. Каскадная модель ЖЦ ИС



Рис. 2.2. Поэтапная модель с промежуточным контролем



Рис. 2.3. Спиральная модель ЖЦ ИС

На практике наибольшее распространение получили две основные *модели жизненного цикла*:

- *каскадная модель* (характерна для периода 1970-1985 гг.);
- *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.

Можно выделить следующие положительные стороны применения каскадного подхода:

- на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
- 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. 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. В результате реальный процесс создания ИС оказывается соответствующим *поэтапной модели с промежуточным контролем*.

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.

Спиральная модель ЖЦ была предложена для преодоления перечисленных проблем. На этапах анализа и проектирования реализуемость технических решений и степень удовлетворения потребностей заказчика проверяется путем создания прототипов. Каждый виток спирали соответствует созданию работоспособного фрагмента или версии системы. Это позволяет уточнить требования, цели и характеристики проекта, определить качество разработки, спланировать работы следующего витка спирали. Таким образом углубляются и последовательно конкретизируются детали проекта и в результате выбирается обоснованный вариант, который удовлетворяет действительным требованиям заказчика и доводится до реализации.

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.

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.

Несмотря на настойчивые рекомендации компаний - вендоров и экспертов в области проектирования и разработки ИС, многие компании продолжают использовать *каскадную модель* вместо какого-либо варианта итерационной модели. Основные причины, по которым *каскадная модель* сохраняет свою популярность, следующие:

Привычка - многие ИТ-специалисты получали образование в то время, когда изучалась только *каскадная модель*, поэтому она используется ими и в наши дни.

Иллюзия снижения рисков участников проекта (заказчика и исполнителя). *Каскадная модель* предполагает разработку законченных продуктов на каждом этапе: технического задания, технического проекта, программного продукта и пользовательской документации. Разработанная документация позволяет не только определить требования к продукту следующего этапа, но и определить обязанности сторон, объем работ и сроки, при этом окончательная оценка сроков и стоимости проекта производится на начальных этапах, после завершения обследования. Очевидно, что если требования к информационной системе меняются в ходе реализации проекта, а качество документов оказывается невысоким (требования неполны и/или противоречивы), то в действительности использование *каскадной модели* создает лишь иллюзию определенности и на деле увеличивает риски, уменьшая лишь ответственность участников проекта. При формальном подходе менеджер проекта реализует только те требования, которые содержатся в спецификации, опирается на документ, а не на реальные потребности бизнеса. 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 требует 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.

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".

Каждая из стадий создания системы предусматривает выполнение определенного объема работ, которые представляются в виде *процессов ЖЦ*. *Процесс* определяется как совокупность взаимосвязанных действий, преобразующих входные данные в выходные. Описание каждого процесса включает в себя перечень решаемых задач, исходных данных и результатов.

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, предложенный в BSP, используется сегодня не только в ИТ-проектах, но и проектах по реинжинирингу бизнес-процессов, изменению организационной структуры. Важнейшие шаги процесса BSP, их последовательность (получить поддержку высшего руководства, определить процессы предприятия, определить классы данных, провести интервью, обработать и организовать данные интервью) можно встретить практически во всех формальных методиках, а также в проектах, реализуемых на практике.

Среди наиболее известных стандартов можно выделить следующие:

- ГОСТ 34.601-90 - распространяется на автоматизированные системы и устанавливает стадии и этапы их создания. Кроме того, в стандарте содержится 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, так же 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 (XP). Экстремальное программирование (самая новая среди рассматриваемых методологий) сформировалось в 1996 году. В основе методологии командная работа, эффективная коммуникация между заказчиком и исполнителем в течение всего проекта по разработке ИС, а разработка ведется с использованием последовательно дорабатываемых прототипов.

В соответствии с базовым международным стандартом ISO/IEC 12207 все процессы ЖЦ ПО делятся на три группы:

Основные процессы:

- приобретение;
- поставка;
- разработка;
- эксплуатация;
- сопровождение.

Вспомогательные процессы:

- документирование;
- управление конфигурацией;
- обеспечение качества;
- разрешение проблем;
- аудит;
- аттестация;
- совместная оценка;
- верификация.

Организационные процессы:

- создание инфраструктуры;
- управление;
- обучение;
- усовершенствование.

Вспомогательные процессы предназначены для поддержки выполнения основных процессов, обеспечения качества проекта, организации верификации, проверки и тестирования ПО. Организационные процессы определяют действия и задачи, выполняемые как заказчиком, так и разработчиком проекта для управления своими процессами.

Для поддержки практического применения стандарта ISO/IEC 12207 разработан ряд технологических документов: Руководство для ISO/IEC 12207 (ISO/IEC TR 15271:1998 Information technology - Guide for ISO/IEC 12207) и Руководство по применению ISO/IEC 12207 к управлению проектами (ISO/IEC TR 16326:1999 Software engineering - Guide for the application of ISO/IEC 12207 to project management).

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 правительственных, коммерческих, военных и академических организациях. Стандарт применим для широкого класса систем, но его основное предназначение - поддержка создания компьютеризированных систем.

Согласно стандарту ISO/IEC серии 15288 в структуру ЖЦ следует включать следующие группы процессов:

Договорные процессы:

- приобретение (внутренние решения или решения внешнего поставщика);
- поставка (внутренние решения или решения внешнего поставщика).

Процессы предприятия:

- управление окружающей средой предприятия;
- инвестиционное управление;
- управление ЖЦ ИС;
- управление ресурсами;
- управление качеством.

Проектные процессы:

- планирование проекта;
- оценка проекта;
- контроль проекта;
- управление рисками;
- управление конфигурацией;
- управление информационными потоками;
- принятие решений.

Технические процессы:

- определение требований;
- анализ требований;
- разработка архитектуры;
- внедрение;
- интеграция;

- верификация;
- переход;
- аттестация;
- эксплуатация;
- сопровождение;
- утилизация.

Специальные процессы:

- определение и установка взаимосвязей исходя из задач и целей.

Стадии создания системы, предусмотренные в стандарте ISO/IEC 15288, несколько отличаются от рассмотренных выше.

FOR AUTHOR USE ONLY

3.ОРГАНИЗАЦИЯ РАЗРАБОТКИ ИС

3.1. Каноническое проектирование ИС

Организация канонического проектирования ИС ориентирована на использование главным образом каскадной модели жизненного цикла ИС. Стадии и этапы работы описаны в стандарте ГОСТ 34.601-90.

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 - получить полное и однозначное понимание 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*), где 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 работ). В документе желательно отразить не только затраты, но и выгоду проекта, например время окупаемости проекта, ожидаемый экономический эффект (если его удастся 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;

- требования 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 определяются 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 формализуемости 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 максимально 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 организации создаются в двух видах:

- **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;
- разработки 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 тестирования 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 по e-mail и т.п.). 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.

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;
- функции 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.

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 изготавливаемых 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 автономные 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.

3.2. Типовое проектирование ИС

Типовое проектирование ИС предполагает создание системы из готовых типовых элементов. Основопологающим 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 к многократному использованию) проектное решение.

Принятая классификация *ТПР* основана на уровне декомпозиции системы. Выделяются следующие классы *ТПР*:

- элементные *ТПР* - типовые решения по задаче или по отдельному виду обеспечения задачи (информационному, программному, техническому, математическому, организационному);
- подсистемные *ТПР* - в качестве элементов типизации выступают отдельные подсистемы, разработанные с учетом функциональной полноты и минимизации внешних информационных связей;
- объектные *ТПР* - 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 установки пакета;
- особенности эксплуатации пакета;
- помощь 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. Достаточно полный перечень показателей можно найти в литературе.

Числовые значения показателей для конкретных ППП устанавливаются 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 получают экспертным 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 - репозитории - 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 репозитория, который поставляется вместе с программным продуктом и расширяется по мере накопления опыта проектирования информационных систем для различных отраслей и типов производства.

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

Базовая модель ИС в репозитории 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 метаописания хранится в репозитории 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 самого нижнего уровня модели бизнес-функций. Для отображения процессов используется модель управления событиями (EPC - Event-driven Process Chain). 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 (например, ABAP в SAP, Tools в BAAN).

Реализация типового проекта предусматривает выполнение следующих операций:

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

FOR AUTHOR USE ONLY

4. АНАЛИЗ И МОДЕЛИРОВАНИЕ ФУНКЦИОНАЛЬНОЙ ОБЛАСТИ ВНЕДРЕНИЯ ИС

4.1. Полная бизнес-модель компании

Практика выработала ряд подходов к проведению организационного анализа, но наибольшее распространение получил инжиниринговый 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 рис. 4.1. представлена 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 согласно [ISO-15704] — это деятельность, осуществляемая предприятием для того, чтобы выполнить функцию, для которой оно было учреждено, - предоставления заказчикам продукта или услуги.

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 других 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.

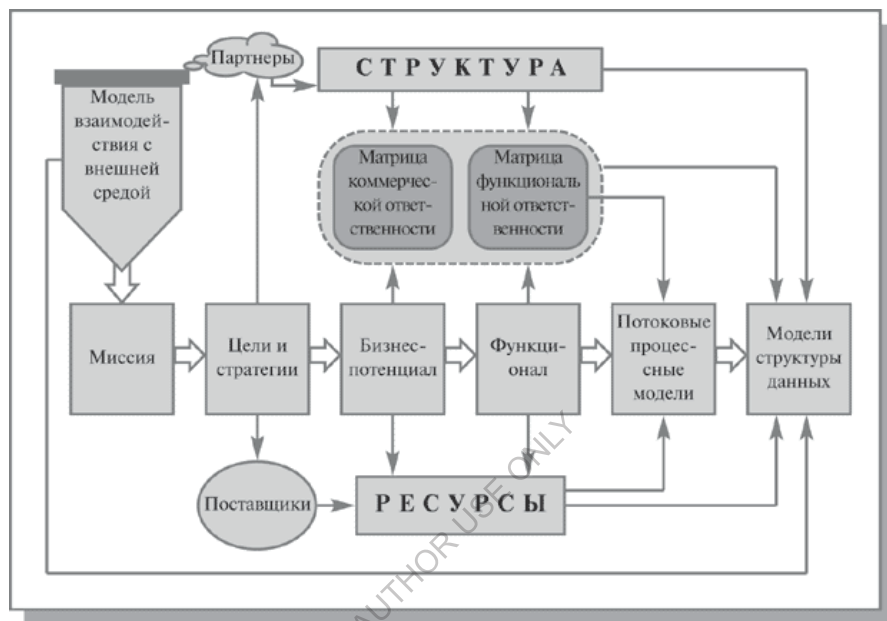


Рис. 4.1. Обобщенная схема организационного бизнес-моделирования

Дерево целей формирует *дерево стратегий* - иерархические списки уточнения и детализации способов 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 сегментации и продвижения. Ресурсные стратегии определяют 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 воспроизводства 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 - 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

деятельности (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 отдельных 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.д.);
- 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) - технология 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 управленческих 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.

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 конкурентоспособной, какие цели и стратегии для этого необходимо реализовать);
- **Организационно-функциональную модель** (отвечает на вопрос кто-что делает в компании, и кто за что отвечает);

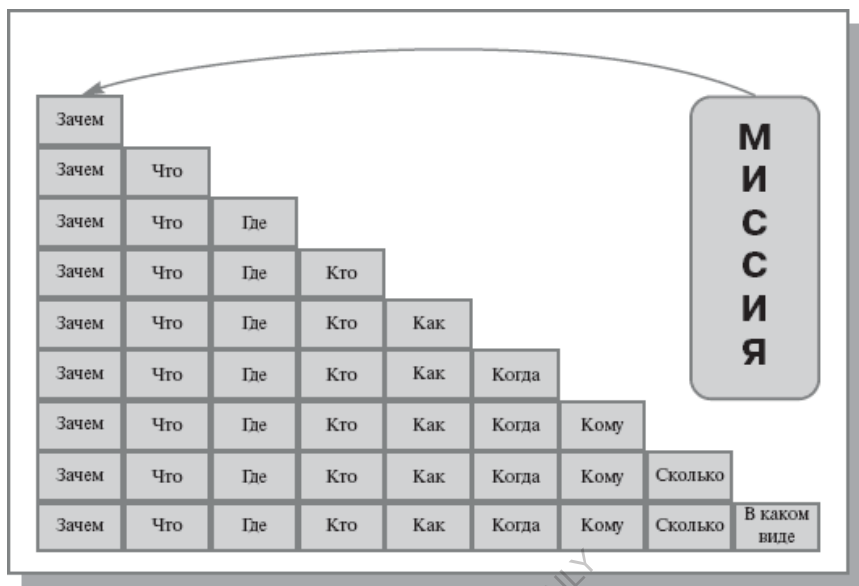


Рис. 4.2. Основные этапы процессно-целевого описания компании

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

• **Функционально-технологическую модель** (отвечает на вопрос что-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).

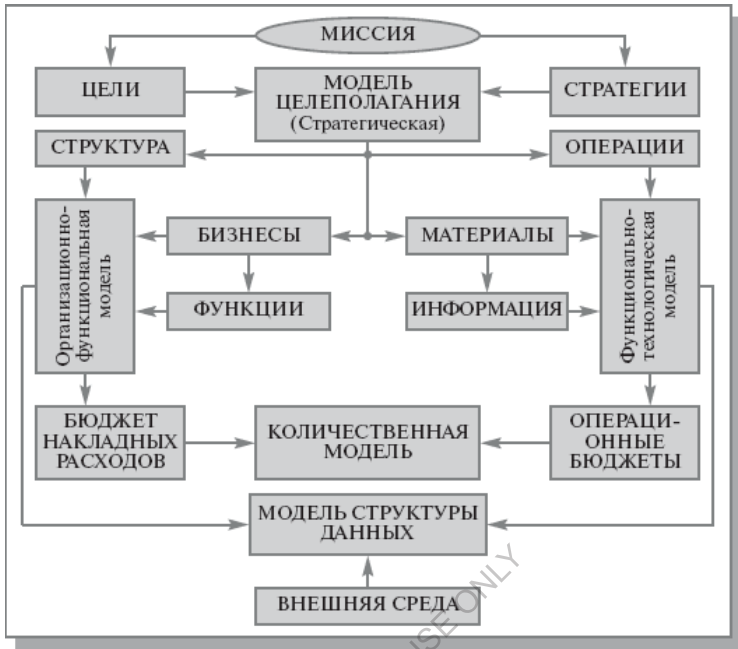


Рис. 4.3. Полная бизнес-модель компании

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

4.2. Шаблоны организационного бизнес-моделирования

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

Шаблон разработки миссии

Как было сказано выше, любая компания с ее микро- и макроокружением представляет собой иерархию вложенных друг в друга открытых, субъектно-

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, и возможностями и желанием компании удовлетворить эти интересы, с другой. Поиск компромисса может быть выполнен по шаблону, представленному на рис. 4.4.

		надо				
		рыночная конъюнк- тура	внешняя среда			
			Политика	Экономика	Социал. сфера	Технология
объект	Уникальность технологий					
	Исключительность ресурсов					
	Знания и умения					
хочу						
	Ценности и ожидания					

МИССИЯ

Рис. 4.4. Шаблон разработки миссии (матрица проекций)

При разработке модели *миссии компании* рекомендуется:

Описать базис конкурентоспособности компании - совокупность характеристик компании 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 позицию "могу".

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".

Выявить наличие способствующих и противодействующих факторов для выбранного вида деятельности со стороны государственных институтов в области политики и экономики.

Оценить перспективу развития технологии в выбранной сфере деятельности.

Оценить возможную поддержку или противодействие общественных организаций.

Сопоставить результаты вышеперечисленных действий с учетом правовых, моральных, этических и др. ограничений со стороны персонала и сформировать позицию "хочу".

Оценить уровень возможных затрат и доходов.

Оценить возможность достижения приемлемого для всех сторон компромисса и сформулировать *Миссию компании* в соответствии с шаблоном, приведенным на рис. 4.5.

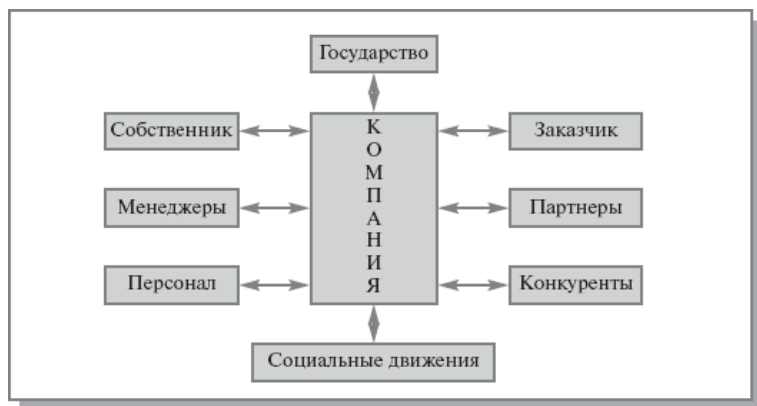


Рис. 4.5. Шаблон разработки миссии

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 получают от бизнеса компании менеджеры;
- что получит от компании персонал;
- в чем может заключаться 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 рис. 4.6.

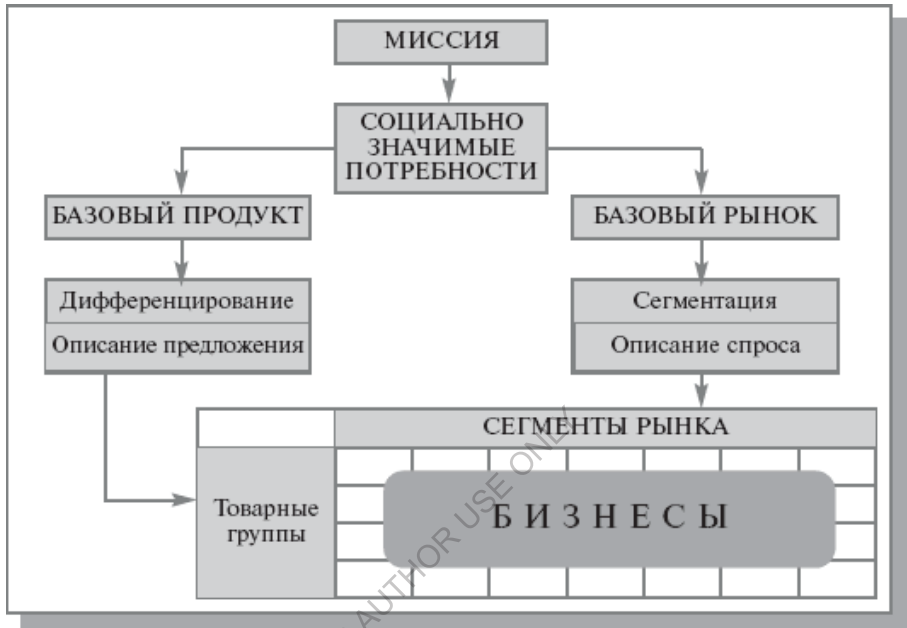


Рис. 4.6. Шаблон формирования бизнесов

В результате формируются 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 (рис. 4.7.) устанавливается соответствие между сформированными товарными группами и сегментами рынка и определяется список бизнесов компании (на пересечении строк и столбцов находятся бизнесы компании).



Рис. 4.7. Шаблон формирования бизнесов (матрица проекций)

Шаблон формирования функционала компании (основных бизнес-функций)

На основании списка бизнесов, с помощью матричной проекции (рис. 4.8.) формируется классификатор бизнес-функций компании.



Рис. 4.8. Шаблон формирования основных бизнес-функций

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 реализуемых менеджерами при организации работ

в любом контуре управления). Далее аналогично, с помощью *матрицы проекций*, формируется список основных функций менеджмента. На рис. 4.9. приведены примеры классификаторов, на основании которых построена матрица - генератор основных функций менеджмента.

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

Рис. 4.9. Шаблон формирования основных функций менеджмента

Представленные матричные проекции 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 компании* выполняется с помощью *матрицы организационных проекций* (рис. 4.10.).

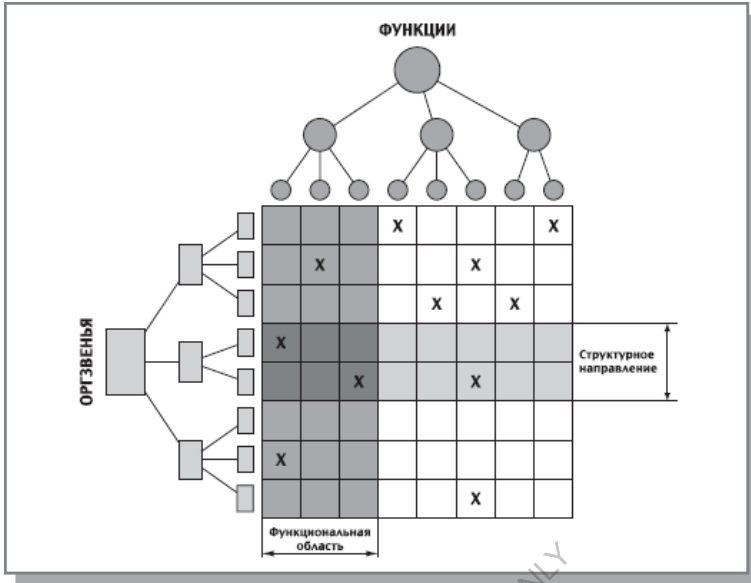


Рис. 4.10. Шаблон распределения функций по организационным звеньям

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 звено, 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 перераспределить 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 5 000 единиц - 50 звеньев на 100 функций.

Аналогично строится *матрица коммерческой ответственности*.

Шаблон потокового процессного описания

Шаблон потокового процессного описания приведен на рис.4. 11. Такое описание дает представление о процессе последовательного преобразования ресурсов в продукты усилиями различных исполнителей на основании соответствующих регламентов.

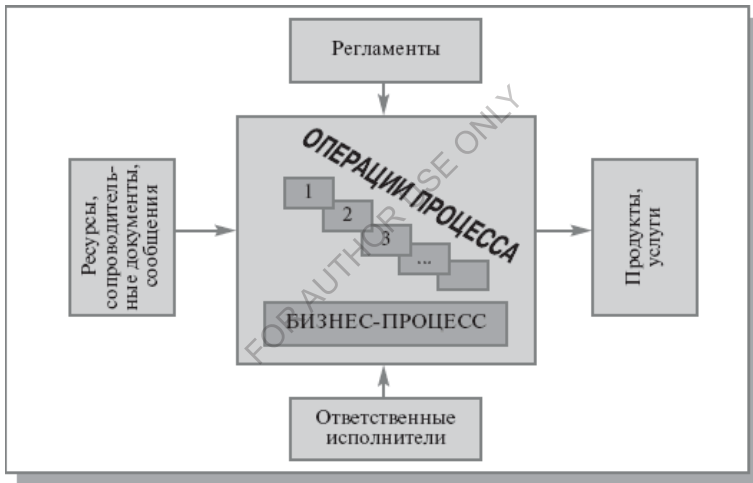


Рис. 4.11. Потоковая процессная модель

4.3. Построения организационно-функциональной модели компании

Организационно-функциональная модель компании строится на основе функциональной схемы деятельности компании.

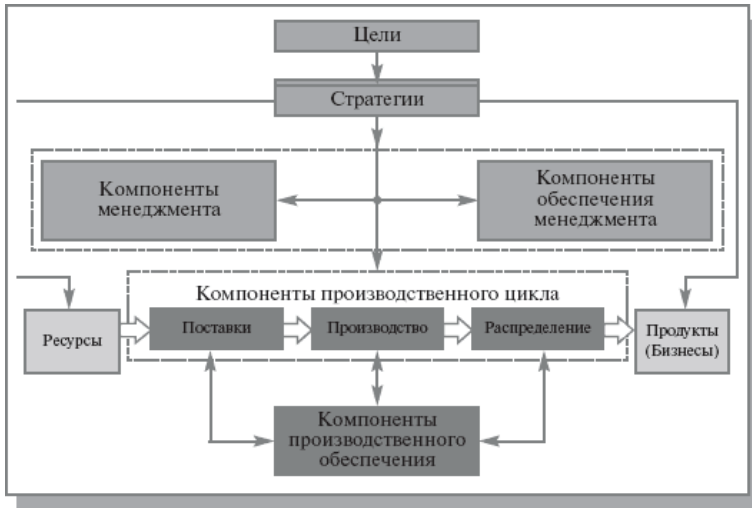


Рис. 4.12. Функциональная схема компании

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. Для управления указанным процессом воспроизводства формируется совокупность компонентов менеджмента, которая порождает набор функций управления. Для поддержания процессов воспроизводства и управления формируются наборы соответствующих функций обеспечения (охраны, технического оснащения, профилактики и ремонта и пр.). Такой подход позволяет описать предприятие с помощью универсального множества управленческих регистров (цели, стратегии, продукты, функции, организационные звенья 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 базовых раздела:

- основные функции - непосредственно связанные с процессом преобразования внешних ресурсов в продукцию и услуги предприятия;
- функции менеджмента - или функции управления предприятием;
- функции обеспечения - поддерживающие производственную, коммерческую и управленческую 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 оргзвеньев важно, чтобы уровень детализации функций соответствовал уровню детализации звеньев. После формирования всех базовых классификаторов с помощью матричных проекций производится их закрепление за оргзвеньями предприятия:

Процесс формирования *матрицы проекций* функций на оргзвенья на практике напоминает игру в крестики-нолики (рис. 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 функции, ставится крестик. Для проекций большой размерности используется механизм расстановки связей между двумя классификаторами, представленных списками.

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

- агрегированную модель;
- детализированную 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 компании. 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 рис. 4.13.



Рис. 4.13. Схема создания Положения об организационно- функциональной структуре компании

4.4. Инструментальные средства организационного моделирования

Применение современных технологий для организационного моделирования позволяет значительно ускорить организационное проектирование. В начале 1990-х годов на Западе появились первые программы для решения задач, связанных с организационными проблемами управления предприятием. Orgware - новый класс программ - был ориентирован на решение задач систематизации, хранения и обработки "неколичественной" информации об организации бизнеса, которые раньше не имели адекватной компьютерной поддержки.

Первый российский продукт - БИГ-Мастер - был создан как компьютерный инструмент для поддержки определенной концепции управления предприятием, получившей название регулярного менеджмента. Главной задачей orgware был переход к строго документированным процедурам и регламентам деятельности. В основу компьютерной парадигмы регулярного менеджмента был положен следующий подход: "Надо создавать не систему взаимосвязанных документов, а систему взаимосвязанных информационных моделей предприятия, которые и будут порождать требуемые документы".

Концептуальной основой БИГ-Мастера стал современный процессный подход к организации деятельности компании. На верхнем уровне система процессов обычно описывается деревом функций - для его обозначения часто используется термин *функционал*. Функции здесь рассматриваются в качестве "свернутых" процессов. Все процессы-функции, как минимум, должны быть определены (т.е. идентифицированы как вид деятельности, имеющий некую цель и результаты) и классифицированы по видам (основные, обеспечивающие, процессы управления). Также должны быть распределены ответственность и полномочия для управления процессами на регулярной основе. На этом уровне

для описания компании в БИГ-Мастере применяются два типа моделей: *древовидные модели (классификаторы)* и *матричные модели (проекции)*.

На нижнем уровне выделенные ("ключевые") процессы могут быть описаны как технологическая последовательность операций (для получения требуемых результатов). Для этого применяются *потокковые модели* бизнес-процессов, назначение которых - описание горизонтальных отношений в организации, связывающих между собой описанные ранее объекты посредством информационных и материальных потоков. Для структурного анализа и проектирования процессов, описываемых *потокковыми моделями*, БИГ-Мастер поддерживает методологию SADT (IDEF). Наличие механизма матричных проекций позволяет определить и описать процессы компании как целостную взаимосвязанную систему.

За счет иерархической структуры классификаторов бизнес-модель одновременно содержит отношения "функция-исполнитель" всех степеней детализации, что позволяет с помощью встроенного генератора отчетов настраивать "разрешение" взгляда на компанию применительно к конкретной управленческой задаче. Система проекций позволяет отразить в отчете любые дополнительные свойства, относящиеся к данному объекту (например, квалификационные требования для персонала, задействованного в процессе). Кроме того, взгляд на компанию может быть связан с любой "координатой отсчета" - например, от документа или сотрудника - в каких процессах и как они участвуют и т.п.

Классификаторы, проекции и *потокковые модели* бизнес-процессов поддерживаются различными способами их визуализации. Для классификаторов - в виде списков и деревьев (оргафов), для проекции - в виде связанных списков и транспонируемых матриц, а для *потокковых моделей* бизнес-процессов - в виде диаграмм IDEF0 (IDEF3) и текстового описания, что облегчает понимание задач участниками процессов. При этом конструирование самих *потокковых моделей* происходит в привычных табличных формах.

В модели возможно формирование неограниченного количества новых классификаторов, проекций и *поточковых моделей*, а следовательно, отчетов и документов для описания и, что особенно важно, создания регламентов деятельности компании.

Наличие в БИГ-Мастере нескольких инструментов моделирования является чрезвычайно полезным. *Матричные модели* поддерживают вертикальную интеграцию - подробное системно-целевое описание компании, выстроенное по иерархии управления и исполняемым функциям. В процессной модели преобладает функционально-технологический подход - горизонтальная интеграция бизнес-операций по процедурам. Все вышеперечисленные возможности БИГ-Мастера делают его удобным инструментальным средством организационного моделирования.

FOR AUTHOR USE ONLY

5. СПЕЦИФИКАЦИЯ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ К ИС

5.1. Процессные потоковые модели

Разработка требований к проектируемой ИС строится на основе статического и динамического описания компании. Статическое описание компании, 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 на управление 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;
- повышенное внимание к вопросам обеспечения качества;
- автоматизация технологий выполнения бизнес-процессов.

Согласно стандарту "Основные Положения и Словарь — ИСО/ОПМС 9000:2000" (п. 2.4) понятие "**Процессный подход**" определяется как:

"Любая деятельность, или комплекс деятельности, в которой используются ресурсы для преобразования входов в выходы, может"

рассматриваться как процесс. 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 процессного 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:

Верхний уровень модели должен отражать только контекст диаграммы – взаимодействие моделируемого единственным контекстным процессом предприятия с внешним миром.

На втором уровне должны быть отражены тематически сгруппированные бизнес-процессы предприятия и их взаимосвязи.

Каждая из деятельностей должна быть детализирована на бизнес-процессы. Детализация бизнес-процессов осуществляется посредством бизнес – функций.

Описание элементарной бизнес-операции осуществляется с помощью миниспецификации.

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 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 (рис.5.1.).

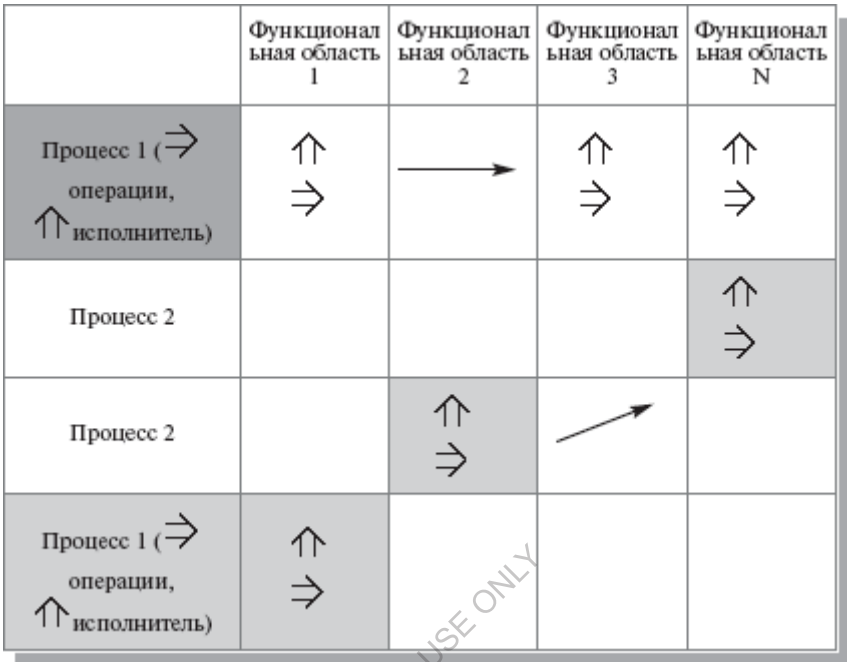


Рис. 5.1. Схема управления деятельностью компании

Фактически основной задачей организационного проектирования является выбор оптимального соотношения между эффективностью использования ресурсов и эффективностью процессов. Жесткая специализация подразделений экономит ресурсы организации, но 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 расстаться). 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 ее "горизонтальным" описанием, как системы процессов.

5.2. 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 (подробнее см. лекции 6, 7) 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 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.

5.3. Выделение и классификация процессов

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

1. Идентификация всей системы "функциональных областей" и процессов компании и 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 и 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 четыре вида бизнес-процессов:

1. Процессы, создающие наибольшую добавленную стоимость (экономическую стоимость, которая определяется издержками компании, относимыми на продукцию).

2. Процессы, создающие наибольшую ценность для клиентов (маркетинговую стоимость за счет дифференциации продукции).

3. Процессы с наиболее интенсивным межзвенным взаимодействием, создающие транзакционные издержки.

4. 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 управления*;
- *процессы обеспечения*;
- сопутствующие;
- вспомогательные;
- процессы развития.

Рассмотрим модель деятельности компании (рис. 5.2.), 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.



Рис. 5.2. Упрощенная модель деятельности компании

Основные процессы образуют "жизненный цикл" продукции компании. Критериями эффективности таких процессов являются обычно качество, точность и своевременность выполнения каждого заказа. Многие потребители рассматривают увеличение качества как нечто более важное, чем уменьшение цены. 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 развитой деятельности компании, может быть много. Все они описываются по производственно-коммерческим цепочкам: "первичное взаимодействие с клиентом и определение его потребностей → реализация запроса (заявки, заказа, контракта 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 (предусмотренных контрактом на поставку или имеющих самостоятельное значение) и т.п.

Эти этапы цепочки также достаточно стандартны (например, в стандарте ИСО редакции 1994 г. 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;
- подготавливает, 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, — идентификация 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, распоряительство, координация, контроль. Самая распространенная ошибка — это смешение этих принципов.

Для реализации процессного описания исключительно важным является то, что любая управленческая деятельность разворачивается 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 (возврат к п.3);
- регулирование на уровне выработки решения (возврат к п.1,2)

Каждый из этих этапов имеет своих характерных для него исполнителей — управленцев, 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 управленческими циклами. Тогда модель деятельности предприятия становится двухуровневой (рис.5.3.)



Рис. 5.3. Двухуровневая модель деятельности предприятия

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

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 отчетность);
- 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;
- 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 (рис.5.4.) и процессов управления (рис. 5.5.).

		КОМПОНЕНТЫ ПРОИЗВОДСТВЕННОГО ОБЕСПЕЧЕНИЯ		
		Технологическое	Техническое	Энергоресурсы
ОСНОВНЫЕ БИЗНЕС-ФУНКЦИИ	№1	БИЗНЕС-ФУНКЦИИ (Обеспечивающие)		
	№2			
	№3			
	...			
	№n			

Рис. 5.4. Упрощенная матрица-генератор обеспечивающих бизнес-функций

		КОМПОНЕНТЫ ПРОИЗВОДСТВЕННОГО ОБЕСПЕЧЕНИЯ						
		Информационные ресурсы	Внутренние коммуникации	Компьютерная обработка	Обработка средствами оргтехники	Защита информации	Телекоммуникация и сеть	Транспорт
Основные функции менеджмента	№1	ФУНКЦИИ МЕНЕДЖМЕНТА (обеспечивающие)						
	№2							
	№3							
	...							
	№n							

Рис. 5.5. 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.

FOR AUTHOR USE ONLY

6. МЕТОДОЛОГИИ МОДЕЛИРОВАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ

6.1. 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* в ИС;

- обеспечение оценки эффективности реализации *модели предметной области* на основе определенных методов и вычисляемых показателей.

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

Структурный аспект предполагает построение:

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

Для отображения структурного аспекта *моделей предметных областей* в основном используются графические методы, которые должны гарантировать представление информации о компонентах системы. Главное требование к графическим методам документирования — простота. Графические методы должны обеспечивать возможность структурной декомпозиции спецификаций системы с максимальной степенью детализации и согласований описаний на смежных уровнях декомпозиции.

С моделированием непосредственно связана проблема **выбора языка** представления проектных решений, позволяющего как можно больше привлекать будущих пользователей системы к ее разработке. **Язык моделирования** – это *нотация*, в основном графическая, которая используется для описания проектов. **Нотация** представляет собой совокупность графических объектов, используемых в модели. *Нотация* является синтаксисом языка моделирования. Язык моделирования, с одной стороны, должен делать

решения проектировщиков понятными пользователю, с другой стороны, предоставлять проектировщикам средства достаточно формализованного и однозначного определения проектных решений, подлежащих реализации в виде программных комплексов, образующих целостную систему программного обеспечения.

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

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

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

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

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

В основе различных методологий моделирования предметной области ИС лежат принципы последовательной детализации абстрактных категорий. Обычно модели строятся на трех уровнях: на внешнем уровне (**определении требований**), на концептуальном уровне (**спецификации требований**) и внутреннем уровне (**реализации требований**). Так, на внешнем уровне модель отвечает на вопрос, что должна делать система, то есть определяется состав

основных компонентов системы: объектов, *функций*, событий, организационных единиц, технических средств. **На концептуальном уровне** модель отвечает на вопрос, как должна функционировать система? Иначе говоря, определяется характер взаимодействия компонентов системы одного и разных типов. На внутреннем уровне модель отвечает на вопрос: с помощью каких программно-технических средств реализуются требования к системе? С позиции жизненного цикла ИС описанные уровни моделей соответственно строятся на этапах анализа требований, логического (технического) и физического (рабочего) проектирования. Рассмотрим особенности построения *моделей предметной области* на трех уровнях детализации.

Объектная структура

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

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

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

Далее концептуальная модель на внутреннем уровне отображается в виде файлов базы данных, входных и выходных документов ЭИС. Причем динамические объекты представляются единицами переменной информации или документами, а статические объекты — единицами условно-постоянной

информации в виде списков, номенклатур, ценников, справочников, классификаторов. Модель базы данных как постоянно поддерживаемого информационного ресурса отображает хранение условно-постоянной и накапливаемой переменной информации, используемой в повторяющихся информационных процессах.

Функциональная структура

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

Функция может быть представлена одним действием или некоторой совокупностью действий. В последнем случае каждой *функции* может соответствовать некоторый процесс, в котором могут существовать свои *подпроцессы*, и т.д., пока каждая из подфункций не будет представлять некоторую недекомпозируемую последовательность действий.

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

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

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

Структура управления

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

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

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

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

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

Организационная структура

Организационная структура представляет собой совокупность организационных единиц, как правило, связанных иерархическими и процессными отношениями. Организационная единица — это подразделение, представляющее собой объединение людей (персонала) для выполнения совокупности общих *функций* или *бизнес-процессов*. В функционально-ориентированной организационной структуре организационная единица выполняет набор *функций*, относящихся к одной *функции* управления и входящих в различные процессы. В процессно-ориентированной структуре организационная единица выполняет набор *функций*, входящих в один тип процесса и относящихся к разным *функциям* управления.

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

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

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

Техническая структура

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

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

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

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

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

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

Структурным анализом принято называть метод исследования системы, которое начинается с ее общего обзора, а затем детализируется, приобретая иерархическую структуру с все большим числом уровней. Для таких методов характерно: разбиение на уровни абстракции с ограниченным числом элементов (от 3 до 7); ограниченный контекст, включающий только существенные детали каждого уровня; использование строгих формальных правил записи; последовательное приближение к результату. *Структурный анализ* основан на двух базовых принципах – "разделяй и властвуй" и принципе иерархической упорядоченности. Решение трудных проблем путем их разбиения на множество меньших независимых задач (так называемых "черных ящиков") и организация этих задач в древовидные иерархические структуры значительно повышают понимание сложных систем. Определим ключевые понятия *структурного анализа*.

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

Функция – совокупность *операций*, сгруппированных по определенному признаку.

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

6.2. 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 ответственностей 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 изменениям 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

Методологию IDEF0 можно считать следующим этапом развития хорошо известного графического языка описания функциональных систем SADT (Structured Analysis and Design Technique). Исторически IDEF0 как стандарт был разработан в 1981 году в рамках обширной программы автоматизации промышленных предприятий, которая носила обозначение ICAM (Integrated Computer Aided Manufacturing). Семейство стандартов IDEF унаследовало свое обозначение от названия этой программы (IDEF=Icam DEFinition), и последняя его редакция была выпущена в декабре 1993 года Национальным Институтом по Стандартам и Технологиям США (NIST).

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 (Activity Box) 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"). На диаграмме функциональный блок изображается прямоугольником. Каждая из четырех сторон функционального блока имеет свое определенное значение (роль), при этом:

- верхняя сторона имеет значение "Управление" (Control);
- левая сторона имеет значение "Вход" (Input);
- правая сторона имеет значение "Выход" (Output);
- нижняя сторона имеет значение "Механизм" (Mechanism).

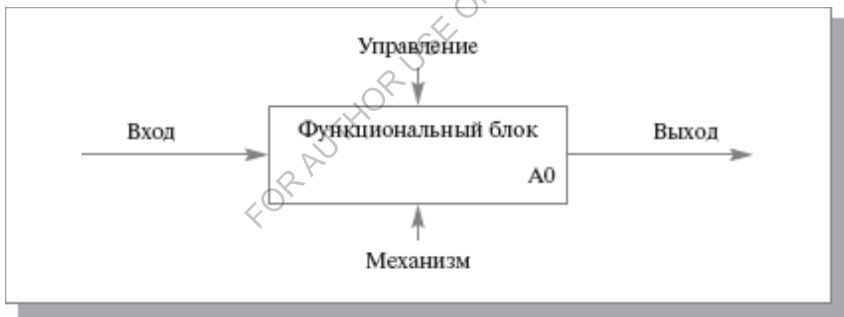


Рис. 6.1. Функциональный блок

Интерфейсная дуга (Arrow) отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на *функцию*, представленную данным функциональным блоком. Интерфейсные дуги часто называют потоками или стрелками.

С помощью интерфейсных дуг отображают различные объекты, в той или иной степени определяющие процессы, происходящие в системе. Такими

объектами могут быть элементы реального мира (детали, вагоны, сотрудники и т.д.) или потоки данных и информации (документы, данные, инструкции и т.д.).

В зависимости от того, к какой из сторон функционального блока подходит данная интерфейсная дуга, она носит название "входящей", "исходящей" или "управляющей".

Необходимо отметить, что любой функциональный блок по требованиям стандарта должен иметь, по крайней мере, одну управляющую интерфейсную дугу и одну исходящую. Это и понятно – каждый процесс должен происходить по каким-то правилам (отображаемым управляющей дугой) и должен выдавать некоторый результат (выходящая дуга), иначе его рассмотрение не имеет никакого смысла.

Обязательное наличие управляющих интерфейсных дуг является одним из главных отличий стандарта IDEF0 от других методологий классов DFD (Data Flow Diagram) и WFD (Work Flow Diagram).

Декомпозиция (Decomposition) является основным понятием стандарта IDEF0. Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его *функции*. При этом уровень детализации процесса определяется непосредственно разработчиком модели.

Декомпозиция позволяет постепенно и структурированно представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко усваиваемой.

Последним из понятий IDEF0 является глоссарий (Glossary). Для каждого из элементов IDEF0 — диаграмм, функциональных блоков, интерфейсных дуг — существующий стандарт подразумевает создание и поддержание набора соответствующих определений, ключевых слов, повествовательных изложений и т.д., которые характеризуют объект, отображенный данным элементом. Этот набор называется глоссарием и является описанием сущности данного элемента. Глоссарий гармонично дополняет наглядный графический язык, снабжая диаграммы необходимой дополнительной информацией.

Модель IDEF0 всегда начинается с представления системы как единого целого – одного функционального блока с интерфейсными дугами, простирающимися за пределы рассматриваемой области. Такая диаграмма с одним функциональным блоком называется **контекстной диаграммой**.

В пояснительном тексте к контекстной диаграмме должна быть указана **цель** (Purpose) построения диаграммы в виде краткого описания и зафиксирована **точка зрения** (Viewpoint).

Определение и формализация цели разработки IDEF0-модели является крайне важным моментом. Фактически цель определяет соответствующие области в исследуемой системе, на которых необходимо фокусироваться в первую очередь.

Точка зрения определяет основное направление развития модели и уровень необходимой детализации. Четкое фиксирование точки зрения позволяет разгрузить модель, отказавшись от детализации и исследования отдельных элементов, не являющихся необходимыми, исходя из выбранной точки зрения на систему. Правильный выбор точки зрения существенно сокращает временные затраты на построение конечной модели.

Выделение подпроцессов. В процессе декомпозиции функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме. Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы, и называется дочерней (Child Diagram) по отношению к нему (каждый из функциональных блоков, принадлежащих дочерней диаграмме, соответственно называется дочерним блоком – Child Box). В свою очередь, функциональный блок — предок называется родительским блоком по отношению к дочерней диаграмме (Parent Box), а диаграмма, к которой он принадлежит – родительской диаграммой (Parent Diagram). Каждая из подфункций дочерней диаграммы может быть далее детализирована путем аналогичной декомпозиции соответствующего ей функционального блока. В каждом случае декомпозиции

функционального блока все интерфейсные дуги, входящие в данный блок или исходящие из него, фиксируются на дочерней диаграмме. Этим достигается структурная целостность IDEF0-модели.

Иногда отдельные интерфейсные дуги высшего уровня не имеет смысла продолжать рассматривать на диаграммах нижнего уровня, или наоборот — отдельные дуги нижнего отражать на диаграммах более высоких уровней — это будет только перегружать диаграммы и делать их сложными для восприятия. Для решения подобных задач в стандарте IDEF0 предусмотрено понятие туннелирования. Обозначение "туннеля" (Arrow Tunnel) в виде двух круглых скобок вокруг начала интерфейсной дуги обозначает, что эта дуга не была унаследована от функционального родительского блока и появилась (из "туннеля") только на этой диаграмме. В свою очередь, такое же обозначение вокруг конца (стрелки) интерфейсной дуги в непосредственной близости от блока-приемника означает тот факт, что в дочерней по отношению к этому блоку диаграмме эта дуга отображаться и рассматриваться не будет. Чаще всего бывает, что отдельные объекты и соответствующие им интерфейсные дуги не рассматриваются на некоторых промежуточных уровнях иерархии, — в таком случае они сначала "погружаются в туннель", а затем при необходимости "возвращаются из туннеля".

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 группой людей, принадлежащих к разным

областям деятельности моделируемой системы. Обычно процесс разработки является итеративным и состоит из следующих условных этапов:

- 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.

- 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.

- 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 диаграммы 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, используемыми 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 продуцировании 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 "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 содержимое 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 распределенный 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 шаге 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 миниспецификация – формальное описание алгоритма преобразования входных данных в выходные.

Миниспецификация 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 отнесем: необходимость искусственного ввода управляющих процессов, поскольку управляющие воздействия (потoki) 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, имеющее четко определенное поведение и обладающие 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 связанного 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 ориентированна 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 существующих методик

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.

При выборе методики моделирования 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*.

Синтетическая методика

Как можно видеть из представленного обзора, каждая из рассмотренных 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 их 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 и 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*).

СПИСОК ЛИТЕРАТУРЫ

1. Федеральный закон Российской Федерации от 27 июля 2006 г. N 149-ФЗ. Об информации, информационных технологиях и о защите информации.
2. ГОСТ 24.202-80 . Требования к содержанию документа «Технико-экономическое обоснование создания АСУ.
3. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
4. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированных систем.
5. ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.
6. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадия создания.
7. ГОСТ 34.603-92. Виды испытаний автоматизированных систем.
8. ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению.
9. ГОСТ Р ИСО/МЭК 12207-99 Информационная технология. Процессы жизненного цикла программных средств.
10. ГОСТ Р В51987-2002. Информационная технология. Комплекс стандартов на автоматизированные системы. Требования и показатели качества функционирования информационных систем (ИС). Общие положения.
11. ГОСТ Р ИСО 15288-2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем.

12. Анисимов В.В. Проектирование информационных систем. Электронный ресурс: <https://sites.google.com/site/anisimovkhv/learning/pris>.
13. Буч Г., Якобсон А., Рамбо Дж., UML. Классика CS. Издание второе, - СПб.: Питер, 2006. - 736 с.
14. Галямина И.Г., Управление процессами, - СПб.: Питер, 2013. - 304 с.
15. Грекул В. И., Денищенко Г. Н., Коровкина Н. Л.— Проектирование информационных систем: учебное пособие / 2-е изд., испр. — М.: Интернет-Университет информационных технологий (ИНТУИТ.РУ): БИНОМ. Лаборатория знаний, 2010 .С 299.
16. Дейт К. Дж., Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом “Вильямс”, 2005. — 1328 с.: ил. — Парал. тит. англ.
17. Диго С.М., Базы данных. Проектирование и создание: Учебно-методический комплекс. – М.: Изд. центр ЕАОИ. 2008. – 171 с.
18. Дубейковский В.И., Эффективное моделирование с СА ERwin Process Modeler (BPwin; AllFusion Process Modeler), - М.: Диалог-МИФИ, 2009, - 384 с.
19. Кириллов, В. В., Введение в реляционные базы данных / В. В. Кириллов, Г. Ю. Громов. — СПб.: БХВ-Петербург, 2009. — 464 с.
20. Ларман К., Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку, - М.: Вильямс, 2013. - 736 с.
21. Леоненков А.В., Самоучитель UML 2, - СПб.:БХВ-Петербург, 2007. - 576 с.
22. Маклаков С.В., Туманов В.Е., Проектирование реляционных хранилищ данных, - М.: Диалог-МИФИ, 2007, - 336 с.
23. Новиков Ф.А., Иванов Д.Ю., Моделирование на UML. Теория, практика, видеокурс, - СПб.: Профессиональная литература, 2010. - 640 с.

- 24.Новиков Ф.А., Иванов Д.Ю., Моделирование на UML [Электронный ресурс]: Интернет книга. -Электронные данные. - 2013. - Режим доступа: <http://book.uml3.ru>.
- 25.Рамбо Дж., Блаха М., UML 2.0. Объектно-ориентированное моделирование и разработка, - СПб.: Питер, 2007. - 544 с.
- 26.Репин В.В., Елиферов В.Г., Процессный к управления. Моделирование бизнес-процессов, - М.: Манн, Иванов и Фербер, 2013. - 544 с.
- 27.Роберт Дж. Мюллер, Проектирование баз данных и UML, - М.: Лори, 2013, - 432 с.
- 28.Советов Б. Я., Базы данных: теория и практика :Учебник для бакалавров / 2-е изд., - М.: Юрайт, 2012. - 464 с.

FOR AUTHOR USE ONLY

FOR AUTHOR USE ONLY

**More
Books!**



yes
I want morebooks!

Buy your books fast and straightforward online - at one of world's fastest growing online book stores! Environmentally sound due to Print-on-Demand technologies.

Buy your books online at
www.morebooks.shop

Покупайте Ваши книги быстро и без посредников он-лайн – в одном из самых быстрорастущих книжных он-лайн магазинов! окружающей среде благодаря технологии Печати-на-Заказ.

Покупайте Ваши книги на
www.morebooks.shop

KS OmniScriptum Publishing
Brivibas gatve 197
LV-1039 Riga, Latvia
Telefax: +371 686 204 55

info@omniscryptum.com
www.omniscryptum.com

OMNIScriptum



FOR AUTHOR USE ONLY