

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ
РЕСПУБЛИКИ
КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им.И.АРАБАЕВА
ОСПО ИНСТИТУТА НОВЫХ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

«УТВЕРЖДАЮ»

Директор ИНИТ
КГУ им.И.Арабаева
К.Ф.Н., и.о.доц. У.Керимов



_____ 2023г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

по дисциплине: Проектирование информационных систем

для студентов специальности: ПИ, АСУ

форма обучения: очная

Курс 1,2 Семестр 1,2,3,4

Часов: всего _____, лекций 74, практ. зан. 50

СРС _____

Учебно-методический комплекс разработал(а) _____

и.о.доц.Турдакунова А.С.

Рассмотрена и утверждена на заседании ОСПО ИНИТ КГУ им.И.Арабаева
Протокол № 1 от «07» сентября 2023 г.

Зав. ОСПО ИНИТ: Н.С.Сейткадиева _____

Одобрено учебно-методическим советом ИНИТ КГУ им.И. Арабаева
Протокол № 1 от «08» 09 2023 г.

Председатель УМС: _____

Бишкек 2023г.

Введение

Курс предусматривает изучение: состава и структуры различных классов ИС как объектов проектирования; современных технологий проектирования ИС и методик обоснования эффективности их применения; содержания стадий и этапов проектирования ИС и их особенностей при использовании различных технологий проектирования; целей и задач проведения предпроектного обследования объектов информатизации; методов моделирования информационных процессов предметной области; классификацию и общие характеристики современных технологий.

Цели и задачи дисциплины, ее значение в учебном процессе

Дисциплина является важной составной частью подготовки специалиста в области информационных технологий.

Основой курса являются методологии системного анализа и моделирования, позволяющие на этапе создания информационной системы решить следующие основные задачи: обеспечение требуемой функциональности системы и адаптивности к изменяющимся условиям ее функционирования; проектирование реализуемых в системе объектов данных; проектирование программ и средств интерфейса (экранных форм, отчетов), которые будут обеспечивать выполнение запросов к данным.

Межпредметные связи. Перечень дисциплин и их разделов, усвоение которых необходимо при изучении данной дисциплины.

Программой курса предусматривается изучение инструментов поддержки проектирования информационных систем. Практикум дисциплины включает в себя задания для освоения учащимися инструментальных средств разработки и анализа функциональных и информационных моделей деятельности экономических объектов (предприятий и учреждений), являющихся основой проектирования информационных систем. Предварительно требуются знания по ряду дисциплин, предусмотренных учебным планом направления "Прикладная информатика", таких как "Информатика и программирование", "Базы данных", "Информационные системы и технологии". Знания, полученные в ходе изучения дисциплины

Компетенции по Госстандарту.

ПК-4 (профессиональные компетенции) способен ставить и решать прикладные задачи с использованием современных информационно-коммуникационных технологий;

ПК-5 (профессиональные компетенции) способен осуществлять и обосновывать выбор проектных решений по видам обеспечения информационных систем;

ПК-6 (профессиональные компетенции) способен документировать процессы создания информационных систем на всех стадиях жизненного цикла;

ПК-7 (профессиональные компетенции) способен использовать технологические и функциональные стандарты, современные модели и методы оценки качества и надежности при проектировании, конструировании и отладке программных средств;

ПК-8 (профессиональные компетенции) способен проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе, участвовать в реинжиниринге прикладных и информационных процессов;

ПК-11 (профессиональные компетенции) способен принимать участие в создании и управлении ИС на всех этапах жизненного цикла;

ПК-13(профессиональные компетенции) способен принимать участие во внедрении, адаптации и настройке прикладных ИС;

ПК-15(профессиональные компетенции) способен принимать участие в реализации профессиональных - способен проводить оценку экономических затрат на проекты по информатизации и автоматизации решения прикладных задач.

В результате освоения дисциплины студент:

должен знать: методы анализа прикладной области, информационных потребностей, формирования требований к ИС; методологии и технологии проектирования ИС, проектирование обеспечивающих подсистем ИС; методы и средства организации и управления проектом ИС на всех стадиях жизненного цикла, оценка затрат проекта и экономической эффективности ИС; основы менеджмента качества ИС; методы управления IT - проектами.

должен уметь: проводить анализ предметной области, выявлять информационные потребности и разрабатывать требования к ИС; проводить сравнительный анализ и выбор ИКТ для решения прикладных задач и создания ИС; разрабатывать концептуальную модель прикладной области, выбирать инструментальные средства и технологии проектирования ИС; проводить формализацию и реализацию решения прикладных задач выполнять работы на всех стадиях жизненного цикла проекта ИС.

должен владеть: навыками работы с инструментальными средствами моделирования предметной области, прикладных и информационных процессов; навыками разработки технологической документации, использования функциональных и технологических стандартов ИС; методы анализа сложных экономических систем;

методы планирования и управления ресурсами при проведении проектных работ; перспективные информационные технологии проектирования, создания, анализа и сопровождения профессионально-но-ориентированных информационных систем; перспективы развития информационных технологий и информационных систем в экономике; методы научных исследований по теории, технологии разработки и эксплуатации профессионально-ориентированных информационных систем; принципы обеспечения информационной безопасности; требования к надежности и эффективности информационных систем в области применения.

Уметь: формулировать и решать задачи проектирования профессионально-ориентированных информационных систем с использованием различных методов и решений; ставить задачу системного проектирования и комплексирования локальных и глобальных сетей обслуживания пользователей информационных систем; ставить и решать задачи, связанные

с организацией диалога между человеком и информационной системой; проводить выбор интерфейсных средств при построении сложных профессионально-ориентированных информационных систем; формулировать основные технико-экономические требования к проектируемым профессионально-ориентированным информационным системам; создавать и внедрять профессионально-ориентированные информационные системы в экономике.

1. Структура дисциплины

№	Название темы	Количество часов	
		Лек	прак
1.	Введение.	2	
2.	Основные понятия технологии проектирования информационных систем	2	
3.	Понятия информационных систем, классы информационных систем.	2	
4.	Программное обеспечение информационных систем.	2	2
5.	Сложность программного обеспечения. Проектирование сложных систем.	4	2
6.	Этапы проектирования информационных систем	2	
7.	Модели процесса проектирования информационной системы.	4	2
8.	Каскадная проектирование информационных систем	2	2
9.	Спиральная модель проектирование ИС	2	2
10.	Спецификация функциональных требований к ИС	2	2
II модуль			
11.	Информационное обеспечение ИС. Моделирование информационного обеспечения.	4	2
12.	Тестирование и отладка ИС	2	2
13.	Типовое проектирование ИС. Внедрение типовых информационных систем.	4	2
14.	Область применение ИС	2	2
15.	Экономические ИС	2	2
16.	Финансовые ИС	2	2
17.	Образовательные ИС	2	2
18.	Медицинские ИС	2	2
Итого:		44	28

№	Название темы	Кол-во часов	
		Лек	прак
1.	Введение	2	
2.	Начальные навыки работы с типовым решением <i>Пользователи конфигурации, Справочник «Пользователи», Начальное заполнение информационной базы.</i>	2	
3.	Основные сведения об организации <i>Справочник «Организации», Справочник «Подразделение организаций».</i>	2	
4.	Справочник «Физические лица» Физические лица, основные сведения о физических лицах, контактная информация, Работа со справочниками.	2	2
5.	Ответственные лица организаций Типовой план счетов, ввод начальных остатков по счетам, справочник «статьи затрат».	2	
6.	Учет банковских и кассовых операций Приходной кассовый ордер, расходный кассовый ордер, кассовая книга, авансовый отчет.	2	2
7.	Банковские счета организаций Банки, платежное поручение исходящее, платежный ордер (списание денежных средств), конвертация, выписка из банка	2	2
8.	Учет товаров и услуг Счета учета номенклатуры, цены номенклатуры	2	2
9.	Склады Поступление товаров и услуг, Доверенность, возврат товаров и услуг поставщика, реализация товаров и услуг.	2	2
10.	Возврат товаров и услуг от покупателя Перемещение товаров, инвентаризация, оприходование, списание товаров отчет «материальная ведомость»	2	2
11.	Учет основных средств Основные средства, ввод начальных остатков по ОС, поступление ОС, принятие к учету ОС, перемещение ОС, выбытие ОС, инвентаризация ОС.	2	2
12.	Параметры амортизации Выработка для расчета амортизации ОС и НМА, модернизация ОС, способы отражения расходов по амортизации, налоговый учет ОС.	2	2
13.	Кадровый учет Должности, приказы на работу, кадровые перемещение, увольнение, неявки.	2	2
14.	Учет заработной платы работников Расчет заработной платы, начисление зарплаты, расчет налогов, выплаты заработной платы.	2	2
15.	Отчетность по заработной плате Расчетная ведомость, расчетные листки, свод начислений и удержаний.	2	1
16.	Отчетность Стандартные отчеты, главная книга.	2	1
Итого:		32	22

Методическая разработка лекций по дисциплине (Краткий курс лекций, презентации)

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)

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 тестов. Поиск по неструктурированным данным осуществляется с использованием семантических признаков. Отобранные документы предоставляются пользователю, а обработка данных в таких системах практически не производится.

Основываясь на степени автоматизации **информационных процессов** в системе управления фирмой, *информационные системы* делятся на ручные, автоматические и автоматизированные. Ручные 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 представления понятия "информационная система" 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

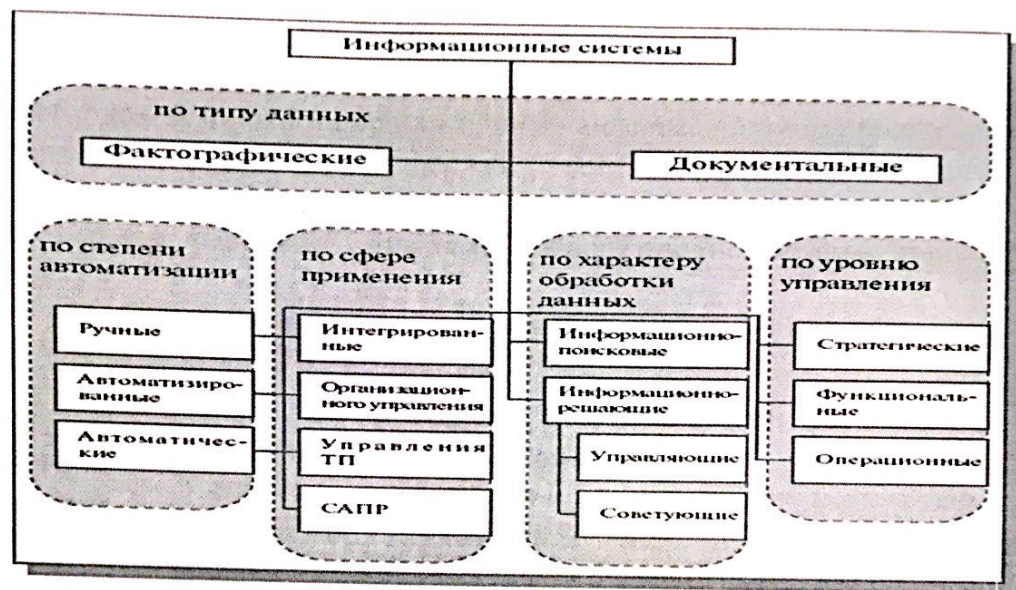


Рис. 1.1. Классификация информационных систем

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.

Типовые задачи, решаемые модулями корпоративной системы, приведены в таблице 1.1.

Таблица 1.1. Функциональное назначение модулей корпоративной ИС.

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

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 операционного уровня - поддерживает исполнителей, обрабатывая данные о сделках и событиях (счета, накладные, зарплата, кредиты, *поток* сырья и материалов). Информационная система операционного 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" (Data Warehouse) - интегрированной *информационной среды*, включающей разнородные информационные ресурсы. И, наконец, для построения глобальных распределенных информационных приложений исполь-

зуется *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 вызывали 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.

Согласно статистическим данным, собранным Standish Group (США), из 8380 проектов, обследованных в США в 1994 году, неудачными оказались более 30% проектов, общая *стоимость* которых превышала 80 миллиардов

долларов. При этом оказались выполненными в срок лишь 16% от общего числа проектов, а перерасход средств составил 189% от запланированного бюджета. 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 преимущество 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 (стадий)*, 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. (Последние два этапа далее не рассматриваются, поскольку выходят за рамки тематики курса.) 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 пройден, *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 *надёжность* 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 надёжности 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*.

Лекция 2. Понятие информационной системы

Система (system – целое, составленное из частей; греч.) – это совокупность элементов, взаимодействующих друг с другом, образующих определенную целостность, единство.

Архитектура системы – совокупность свойств системы, существенных для пользователя.

Элемент системы – часть системы, имеющая определенное функциональное назначение. Элементы, состоящие из простых взаимосвязанных элементов, часто называют *подсистемами*.

Организация системы – внутренняя упорядоченность, согласованность взаимодействия элементов системы, проявляющаяся, в частности, в ограничении разнообразия состояния элементов в рамках системы.

Структура системы – состав, порядок и принципы взаимодействия элементов системы, определяющие основные свойства системы. Если отдельные элементы системы разнесены по разным уровням и характеризуются внутренними связями, то говорят об иерархической структуре системы.

Добавление к понятию *система* слова *информационная* отражает цель ее создания и функционирования. Информационные системы обеспечивают сбор, хранение, обработку, поиск, выдачу информации, необходимой в процессе принятия решений задач из любой области. Они помогают анализировать проблемы и создавать новые информационные продукты.

Информационная система— это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Современное понимание информационной системы предполагает использование в качестве основного технического средства переработки информации компьютера. Кроме того, техническое воплощение информационной системы само по себе ничего не будет значить, если не учтена роль человека, для которого предназначена производимая информация и без которого невозможно ее получение и представление.

Необходимо понимать разницу между компьютерами и информационными системами. Компьютеры, оснащенные специализированными программными средствами, являются технической базой и инструментом для информационных систем. Информационная система немыслима без персонала, взаимодействующего с компьютерами и телекоммуникациями.

В нормативно-правовом смысле *информационная система* определяется как «организационно упорядоченная совокупность документов (массив документов) и информационных технологий, в том числе и с использованием средств вычислительной техники и связи, реализующих информационные процессы»

Процессы, протекающие в информационных системах

Информационный процесс – «процесс создания, сбора, обработки, накопления, хранения, поиска, распространения и потребления информации»

Информационный ресурс – это отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других видах информационных систем)

В нормативно-правовом аспекте **документ** определяется как зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать. Процесс *документирования* превращает информацию в информационные ресурсы. Процессы, обеспечивающие работу информационной системы любого назначения, условно можно представить состоящими из следующих блоков:

- ввод информации из внешних или внутренних источников;
- обработка входной информации и представление ее в удобном виде;

- вывод информации для представления потребителям или передачи в другую систему;
- обратная связь — это информация, переработанная людьми данной организации для коррекции входной информации.

Информационные процессы реализуются с помощью *информационных процедур*, реализующих тот или иной механизм переработки входной информации в конкретный результат.

Различают следующие типы информационных процедур:

1. Полностью *формализуемые*, при выполнении которых алгоритм переработки информации остается неизменным и полностью определен (поиск, учет, хранение, передача информации, печать документов, расчет на моделях).
2. *Неформализуемые* информационные процедуры, при выполнении которых создается новая уникальная информация, причем алгоритм переработки исходной информации неизвестен (формирование множества альтернатив выбора, выбор одного варианта из полученного множества).
3. *Плохо формализованные* информационные процедуры, при выполнении которых алгоритм переработки информации может изменяться и полностью не определен (задача планирования, оценка эффективности вариантов экономической политики).

Функции информационных подразделений, создающих и поддерживающих информационные системы (служба администратора): оповещение и обработка запросов; поддержание целостности и сохранности информации; периодическая ревизия информации; автоматизация индексирования информации.

В целом информационные системы определяются следующими свойствами:

- любая информационная система может быть подвергнута анализу, построена и управляема на основе общих принципов построения систем;
- информационная система является динамичной и развивающейся;
- при построении информационной системы необходимо использовать системный подход;
- выходной продукцией информационной системы является информация, на основе которой принимаются решения;
- информационную систему следует воспринимать как человеко-машинную систему обработки информации.

Внедрение информационных систем может способствовать:

- получению более рациональных вариантов решения управленческих задач за счет внедрения математических методов;
- освобождению работников от рутинной работы за счет ее автоматизации;
- обеспечению достоверности информации;
- совершенствованию структуры информационных потоков (включая систему документооборота);
- предоставлению потребителям уникальных услуг;

• уменьшению затрат на производство продуктов и услуг (включая информационные).

Этапы развития информационных систем

Этапы развития информационных систем и цели их использования представлены в таблице 1.1.

Первые информационные системы появились в пятидесятых годах. Они были предназначены для обработки счетов и расчета зарплаты, а реализовывались на электромеханических бухгалтерских счетных машинах. Это приводило к некоторому сокращению затрат и времени на подготовку бумажных документов.

Таблица 1.1. Этапы развития информационных систем

Период времени	Концепция использования информации	Вид информационных систем	Цель использования
1950-1960 годы	Бумажный поток расчетных документов	Электромеханические бухгалтерские машины	Упрощение процедуры обработки счетов и расчета зарплаты
1960-1970 годы	Помощь в подготовке отчетов	Управленческие информационные системы для производственной информации	Ускорение процесса подготовки отчетности
1970-1980 годы	Управленческий контроль процессов	Системы поддержки принятия решений	Выработка рациональных решений
с 1980 года по н/в	Информация стратегический ресурс, обеспечивающий конкурентное преимущество	Стратегические информационные системы. Автоматизированные офисы	Выживание и процветание организации

Шестидесятые годы знаменуются изменением отношения к информационным системам. Информация, полученная из них, стала применяться для периодической отчетности по многим параметрам. Для этого организациям требовалось компьютерное оборудование широкого назначения, способное обслуживать множество функций, а не только обрабатывать счета и считать зарплату.

В семидесятых – начале восьмидесятых годов информационные системы начинают широко использоваться в качестве средства управленческого контроля, поддерживающего и ускоряющего процесс принятия решений.

К концу восьмидесятых годов концепция использования информационных систем вновь изменяется. Они становятся стратегическим источником информации и используются на всех уровнях организации любого профиля. Информационные системы этого периода, предоставляя вовремя нужную информацию, помогают организации достичь успеха в своей деятельности, создавать новые товары и услуги, находить новые рынки сбыта, обеспечивать себе достойных партнеров, организовывать выпуск продукции по низкой цене и многое другое.

Типы информационных систем. Фактографические и документированные информационные системы

Тип информационной системы зависит от того, чьи интересы она обслуживает и на каком уровне управления. По характеру представления и логической организации хранимой информации информационные системы подразделяются на фактографические, документальные и геоинформационные.

Фактографические информационные системы накапливают и хранят данные в виде множества экземпляров одного или нескольких типов структурных элементов (информационных объектов). Каждый из таких экземпляров или некоторая их совокупность отражают сведения по какому-либо факту, событию отдельно от всех прочих сведений и фактов.

Структура каждого типа информационного объекта состоит из конечного набора реквизитов, отражающих основные аспекты и характеристики объектов данной предметной области. Комплектование информационной базы в фактографических информационных системах включает, как правило, обязательный процесс структуризации входной информации.

Фактографические информационные системы предполагают удовлетворение информационных потребностей непосредственно, т.е. путем представления потребителям самих сведений (данных, фактов, концепций).

В документальных (документированных) информационных системах единичным элементом информации является нерасчлененный на более мелкие элементы документ и информация при вводе (входной документ), как правило, не структурируется, или структурируется в ограниченном виде. Для вводимого документа могут устанавливаться некоторые формализованные позиции (дата изготовления, исполнитель, тематика). Некоторые виды документальных информационных систем обеспечивают установление логической взаимосвязи вводимых документов — соподчиненность по смысловому содержанию, взаимные отсылки по каким-либо критериям и т.д.

Определение и установление такой взаимосвязи представляет собой сложную многокритериальную и многоаспектную аналитическую задачу, которая не может быть формализована в полной мере.

В геоинформационных системах данные организованы в виде отдельных информационных объектов (с определенным набором реквизитов), привязанных к общей электронной топографической основе (электронной

карте). Геоинформационные системы применяются для информационного обеспечения в тех предметных областях, структура информационных объектов и процессов в которых имеет пространственно-географический компонент (маршруты транспорта, коммунальное хозяйство).

Классификация информационных систем по функциональному признаку

Функциональный признак определяет назначение подсистемы, а также ее основные цели, задачи и функции. На рис. 1.1 представлена классификация информационных систем по характеристике их функциональных подсистем.



Рис. 1.1. Классификация информационных систем по функциональному признаку

В хозяйственной практике производственных и коммерческих объектов типовыми видами деятельности, которые определяют функциональный признак классификации информационных систем, являются производственная, маркетинговая, финансовая, кадровая деятельность.

Классификация информационных систем по уровням управления

Выделяют:

- информационные системы оперативного (операционного) уровня – бухгалтерская, банковских депозитов, обработки заказов, регистрации билетов, выплаты зарплаты;
- информационная система специалистов – офисная автоматизация, обработка знаний (включая экспертные системы);
- информационные системы тактического уровня (среднее звено) – мониторинг, администрирование, контроль, принятие решений;
- стратегические информационные системы – формулирование целей, стратегическое планирование.

Информационные системы оперативного (операционного) уровня

Информационная система оперативного уровня поддерживает специалистов-исполнителей, обрабатывая данные о сделках и событиях (счета, накладные, зарплата, кредиты, поток сырья и материалов). Назначение информационной системы на этом уровне — отвечать на запросы о текущем состоянии и отслеживать поток сделок в фирме, что соответствует оперативному управлению. Чтобы с этим справиться, информационная система должна быть легко доступной, непрерывно действующей и предоставлять точную информацию.

Задачи, цели и источники информации на оперативном уровне заранее определены и в высокой степени структурированы. Решение запрограммировано в соответствии с заданным алгоритмом.

Информационная система оперативного уровня является связующим звеном между фирмой и внешней средой. Если система работает плохо, то организация либо не получает информации извне, либо не выдает информацию. Кроме того, система — это основной поставщик информации для остальных типов информационных систем в организации, т.к. содержит и оперативную, и архивную информацию.

Информационные системы специалистов.

Информационные системы этого уровня помогают специалистам, работающим с данными, повышают продуктивность и производительность работы инженеров и проектировщиков. Задача подобных информационных систем — интеграция новых сведений в организацию и помощь в обработке бумажных документов.

По мере того как индустриальное общество трансформируется в информационное, производительность экономики все больше будет зависеть от уровня развития этих систем. Такие системы, особенно в виде рабочих станций и офисных систем, наиболее быстро развиваются сегодня в бизнесе.

Информационные системы офисной автоматизации в следствие своей простоты и много профильности активно используются работниками любого организационного уровня. Наиболее часто их применяют работники средней квалификации: бухгалтеры, секретари, клерки. Основная цель — обработка данных, повышение эффективности их работы и упрощение канцелярского труда. Информационные системы офисной автоматизации связывают воедино работников информационной сферы в разных регионах и помогают поддерживать связь с покупателями, заказчиками и другими организациями. Их деятельность в основном охватывает управление документацией, коммуникации, составление расписаний и т.д.

Эти системы выполняют следующие функции:

- обработка текстов на компьютерах с помощью различных текстовых процессоров;
- производство высококачественной печатной продукции;
- архивация документов;
- электронные календари и записные книжки для ведения деловой информации;
- электронная и аудиопочта;

- видео- и телеконференции.

Информационные системы обработки знаний, в том числе и экспертные системы, вбирают в себя знания, необходимые инженерам, юристам, ученым при разработке или создании нового продукта. Их работа заключается в создании новой информации и нового знания. Так, например, существующие специализированные рабочие станции по инженерному и научному проектированию позволяют обеспечить высокий уровень технических разработок.

Информационные системы тактического уровня (среднее звено)

Основные функции этих информационных систем:

- сравнение текущих показателей с прошлыми показателями;
- 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.

Системы поддержки принятия решений обслуживают частично структурированные задачи, результаты которых трудно спрогнозировать заранее (имеют более мощный аналитический аппарат с несколькими моделями). Информацию получают из управленческих и операционных информационных систем. Используют эти системы все, кому необходимо принимать решение: менеджеры, специалисты, аналитики. Например, их рекомендации могут пригодиться при принятии решения покупать или взять оборудование в аренду.

Характеристика систем поддержки принятия решений:

- обеспечивают решение проблем, развитие которых трудно прогнозировать;
- оснащены сложными инструментальными средствами моделирования и анализа;
- позволяют легко менять постановки решаемых задач и входные данные;
- отличаются гибкостью и легко адаптируются к изменению условий несколько раз в день;
- имеют технологию, максимально ориентированную на пользователя.

Стратегические информационные системы

Развитие и успех любой организации (фирмы) во многом определяются принятой в ней стратегией. Под *стратегией* понимается набор методов и средств решения перспективных долгосрочных задач. В этом контексте можно воспринимать и понятия *стратегический метод*, *стратегическое средство*, *стратегическая система*.

В настоящее время в связи с переходом к рыночным отношениям вопросу стратегии развития и поведения фирмы стали уделять большое внимание, что способствовало коренному изменению во взглядах на информационные системы. Они стали расцениваться как стратегически важные системы, которые влияют на изменение выбора целей фирмы, ее задач, методов, продуктов, услуг, позволяя опередить конкурентов, а также наладить более тесное взаимодействие потребителей с поставщиками. Появился новый тип информационных систем — стратегический.

Стратегическая информационная система — компьютерная информационная система, обеспечивающая поддержку принятия решений по реализации перспективных стратегических целей развития организации. Известны ситуации, когда новое качество информационных систем заставляло изменять не только структуру, но и профиль фирм, содействуя их процветанию. Однако при этом возможно возникновение нежелательной психологической обстановки, связанное с автоматизацией некоторых функций и видов работ, так как это может поставить некоторую часть работающих в затруднительное положение.

Прочие классификации информационных систем

Классификация по степени автоматизации

В зависимости от степени автоматизации информационных процессов в системе управления фирмой информационные системы определяются как ручные, автоматические, автоматизированные.

Ручные информационные системы характеризуются отсутствием современных технических средств переработки информации и выполнением всех операций человеком. Например, о деятельности менеджера в фирме, где отсутствуют компьютеры, можно говорить, что он работает с ручной информационной системой.

Автоматические информационные системы выполняют все операции по переработке информации без участия человека.

Автоматизированные информационные системы предполагают участие в процессе обработки информации и человека, и технических средств, причем главная роль отводится компьютеру. В современном толковании в термин "информационная система" обязательно вкладывается понятие автоматизируемой системы. Автоматизированные информационные системы, учитывая их широкое использование в организации процессов управления, имеют различные модификации и могут быть классифицированы, например, по характеру использования информации и по сфере применения.

Пример 1.1. Роль бухгалтера в информационной системе по расчету заработной платы заключается в задании исходных данных. Информационная система обрабатывает их по заранее известному алгоритму с выдачей результатной информации в виде ведомости, напечатанной на принтере.

Классификация по характеру использования информации

Информационно-поисковые системы производят ввод, систематизацию, хранение, выдачу информации по запросу пользователя без сложных преобразований данных (информационно-поисковая система в библиотеке, в железнодорожных и авиакассах).

Информационно-решающие системы осуществляют все операции переработки информации по определенному алгоритму. Среди них можно провести классификацию по степени воздействия выработанной результатной информации на процесс принятия решений и выделить два класса — управляющие и советующие системы. *Управляющие информационные системы* вырабатывают информацию, на основании которой человек принимает решение. Для этих систем характерен тип задач расчетного

характера и обработка больших объемов данных. Примером могут служить система оперативного планирования выпуска продукции, система бухгалтерского учета.

Советующие информационные системы вырабатывают информацию, которая принимается человеком к сведению и не превращается немедленно в серию конкретных действий. Эти системы обладают более высокой степенью интеллекта, так как для них характерна обработка знаний, а не данных.

Пример 1.2. Существуют медицинские информационные системы для постановки диагноза больному и определения предполагаемой процедуры лечения. Врач может принять к сведению полученную информацию, но и предложить иное решение по сравнению с рекомендуемым системой.

Классификация по сфере применения

Информационные системы организационного управления предназначены для автоматизации функций управленческого персонала. Учитывая наиболее широкое применение и разнообразие этого класса систем, часто любые информационные системы понимают именно в данном толковании. К этому классу относятся информационные системы управления как промышленными фирмами, так и непромышленными объектами: гостиницами, банками, торговыми фирмами и др.

Информационные системы управления технологическими процессами служат для автоматизации функций производственного персонала. Они широко используются при организации поточных линий, изготовлении микросхем, на сборке, для поддержания технологического процесса в металлургической и машиностроительной промышленности.

Информационные системы автоматизированного проектирования 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) информационные системы используются для автоматизации всех функций фирмы и охватывают весь цикл работ от проектирования до сбыта продукции. Создание таких систем весьма затруднительно, поскольку требует системного подхода с позиций главной цели, например получения прибыли, завоевания рынка сбыта и т.д. Такой подход может привести к существенным изменениям в самой структуре фирмы, на что может решиться не каждый управляющий.

Классификация по способу организации

По способу организации групповые и корпоративные информационные системы подразделяются на следующие классы: системы на основе архитектуры файл-сервер; системы на основе архитектуры клиент-сервер; системы на основе многоуровневой архитектуры; системы на основе интернет-технологий.

Лекция №4. Программное обеспечение информационных систем

Классификация программного обеспечения

Под программным обеспечением информационных систем понимается совокупность программных и документальных средств для создания и эксплуатации систем обработки данных средствами вычислительной техники.

В зависимости от функций, выполняемых программным обеспечением, его можно разделить на 2 группы: базовое (системное) программное обеспечение (рис. 1) и прикладное программное обеспечение (рис. 2).

Базовое (системное) ПО организует процесс обработки информации в компьютере и обеспечивает нормальную рабочую среду для прикладных программ. Базовое ПО настолько тесно связано с аппаратными средствами, что его иногда считают частью компьютера.

Прикладное программное обеспечение предназначено для решения конкретных задач пользователя и организации вычислительного процесса информационной системы в целом.

В состав базового (системного) ПО входят:

- операционные системы;
- сервисные программы;
- трансляторы языков программирования;
- программы технического обслуживания.

Операционные системы (ОС) обеспечивают управление процессом обработки информации и взаимодействие между аппаратными средствами и пользователем. Одной из важнейших функций ОС является автоматизация процессов ввода-вывода информации, управления выполнением прикладных задач, решаемых пользователем. ОС загружает нужную программу и память ЭВМ и следит за ходом ее выполнения; анализирует ситуации, препятствующие нормальным вычислениям, и дает указания о том, что необходимо сделать, если возникли затруднения. Исходя из выполняемых функции, ОС можно разбить на три группы (см. рис. 1): однозадачные (однопользовательские); многозадачные (многопользовательские); сетевые.

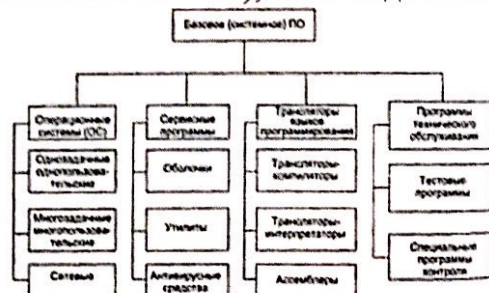


Рис. 1. Базовое (системное) программное обеспечение

Однозадачные ОС предназначены для работы одного пользователя в каждый конкретный момент одной конкретной задачей. Типичным представителем таких операционных систем является MS-DOS (разработанная фирмой Microsoft). Многозадачные ОС обеспечивают коллективное использование ЭВМ в мультипрограммном режиме разделения времени (в памяти ЭВМ находится несколько программ -- задач, -- и

процессор распределяет ресурсы компьютера между задачами). Типичными представителями подобного класса ОС являются: UNIX, OS 2 корпорации IBM, Microsoft Windows 95, Microsoft Windows NT и некоторые другие.

Сетевые операционные системы связаны с появлением локальных и глобальных сетей 11 предназначены для обеспечения доступа пользователя ко всем ресурсам вычислительной сети. Типичными представителями сетевых ОС являются:

Novell NetWare, Microsoft Windows NT, Banyan Vines, IBM LAN, UNIX, Solaris фирмы Sun.

Сервисное программное обеспечение -- это совокупность программных продуктов, предоставляющих пользователю дополнительные услуги в работе с компьютером и расширяющих возможности операционных систем.

По функциональным возможностям сервисные средства можно подразделить на средства:

- улучшающие пользовательский интерфейс;
- защищающие данные от разрушения и несанкционированного доступа;
- восстанавливающие данные;
- ускоряющие обмен данными между диском и ОЗУ;
- архивации-разархивации;
- антивирусные средства.

По способу организации и реализации сервисные средства могут быть представлены: оболочками, утилитами и автономными программами. Разница между оболочками и утилитами зачастую выражается лишь в универсальности первых и специализации вторых.

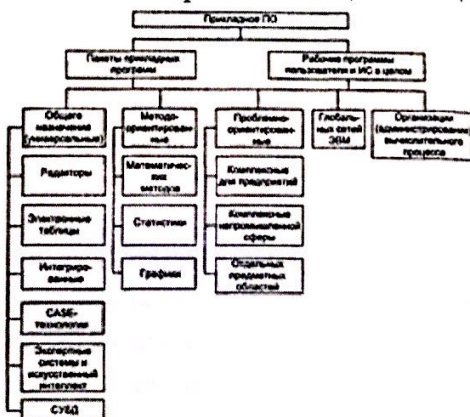


Рис. 2. Прикладное программное обеспечение

Оболочки, являющиеся надстройкой над ОС, называются операционными оболочками. Оболочки являются как бы настройками над операционной системой. Утилиты и автономные программы имеют узкоспециализированное назначение и выполняют каждая свою функцию. Но утилиты, в отличие от автономных программ, выполняются в среде соответствующих оболочек. При этом они конкурируют в своих функциях с программами ОС и другими утилитами. Поэтому классификация сервисных средств по их функциям и способам реализации является достаточно размытой и весьма условной.

Прикладное программное обеспечение и тенденции его развития
ПО общего назначения или типовому прикладному ПО относят программы, предназначенные для любых пользователей ПК независимо от области их профессиональных интересов. Это следующие программы:

- текстовые процессоры,
- табличные процессоры,
- системы иллюстративной и деловой графики (графические процессоры),
- системы управления базами данных,
- экспертные системы,
- программы математических расчетов, моделирования и анализа экспериментальных данных.

Все эти программы имеют широкое применение. Однако специалисты разных областей используют и специальные, только им необходимые программы, относящиеся к специальному программному обеспечению. Так юристы широко используют справочные информационные системы такие как "Гарант", "Юрисконсульт" или "Консультант - плюс". Прикладное программное обеспечение (рис. 2) предназначено для разработки и выполнения конкретных задач (приложений) пользователя. Прикладное программное обеспечение работает под управлением базового ПО, в частности операционных систем.

Редакторы документов - это наиболее широко используемый вид прикладных программ. Они позволяют подготавливать документы гораздо быстрее и удобнее, чем с помощью пишущей машинки. Редакторы документов позволяют использовать различные шрифты символов, абзацы произвольной формы, автоматически переносят слова на новую строку, позволяют делать сноски, включать рисунки, автоматически нумеруют страницы и сноски и т.д. Представители редакторов документов - программы Microsoft Word, Wordpad.

Табличные процессоры. При работе с табличным процессором на экран выводится прямоугольная таблица, в клетках которой могут находиться числа, пояснительные тексты и формулы для расчета значения в клетке по именуемому данным. Все распространенные табличные процессоры позволяют вычислять значения элементов таблиц по заданным формулам, строить по данным в таблицах различные графики и т.д. Представители семейства табличных процессоров Microsoft Excel, Quatro Pro.

Графические редакторы позволяют создавать и редактировать рисунки. В простейших редакторах предоставляются возможности рисования линий, кривых, раскраски областей экрана, создание надписей различными шрифтами и т.д. Большинство редакторов позволяют обрабатывать изображения, полученные с помощью сканеров. Представители графических редакторов - программы Adobe Photoshop, Corel Draw.

Правовые базы данных содержат тексты нормативных документов и предоставляют возможности справки, контекстного поиска, распечатки и т.д. Представители правовых баз данных - пакеты Гарант и Консультант+.

Системы автоматизированного проектирования (САПР) позволяют осуществлять черчение и конструирование различных предметов и

механизмов с помощью компьютера. Среди систем малого и среднего класса в мире наиболее популярна система AutoCad фирмы AutoDesk. Отечественный пакет с аналогичными функциями - Компас.

Системы управления базами данных (СУБД) позволяют управлять большими информационными массивами - базами данных. Программные системы этого вида позволяют обрабатывать на компьютере массивы информации, обеспечивают ввод, поиск, сортировку выборку записей, составление отчетов и т.д. Представители данного класса программ - Microsoft Access, Clipper, Paradox.

Интегрированные системы сочетают в себе возможность системы управления базами данных, табличного процессора, текстового редактора, системы деловой графики, а иногда и другие возможности. Как правило, все компоненты интегрированной системы имеют схожий интерфейс, что облегчает обучение работе с ними. Представители интегрированных систем - пакет Microsoft Office и его бесплатный аналог Open Office.

Бухгалтерские программы предназначены для ведения бухгалтерского учета, подготовки финансовой отчетности и финансового анализа деятельности предприятий. Из-за несовместимости отечественного бухгалтерского учета с зарубежным в нашей стране используются почти исключительно отечественные бухгалтерские программы. Наиболее распространены системы 1С: Предприятие и Инфо-бухгалтер. Основные тенденции развития прикладного программного обеспечения тесно связаны с созданием и переходом на информационные системы четвертого поколения, основанные на иерархической структуре, в которых Центр тяжести перенесен с локальных сетей конечных пользователей на сеть локальных серверов. В основу ИС четвертого поколения закладывается требование сокращения эксплуатационных ресурсов ИС при увеличении масштабируемости системы и расширения круга ее функциональных обязанностей. В ближайшие пять лет ожидается резкое увеличение сложности программного обеспечения, предназначенного для информационных систем различного класса. Следствием этого станет ужесточение требований к характеристикам компьютеров, сетевого оборудования, пропускной способности каналов связи, а также определение оптимального распределения нагрузки в узлах ИС, в которых ресурсы закрепляются за конечным пользователем по принципу «ровно столько, сколько нужно». Поэтому для всех подразделений компаний необходимо подобрать наиболее удачную конфигурацию сервера и состав программного обеспечения и сбалансировать распределение нагрузки между центральным сервером, локальными серверами и рабочими станциями конечных пользователей в каждом подразделении предприятия. В конечном счете, от этого зависит адекватный выбор аппаратных и программных средств для системы, причем для каждой конкретной ИС эта проблема требует индивидуального подхода. Однако некоторые общие принципы балансировки системы можно привести.

Лекция № 5 . Сложность программного обеспечения

Сложность программного обеспечения представляет неотъемлемое свойство программирования и программ, которое проявляется во времени и стоимости создания программ, в объеме или длине текста программ, характеристиках ее логической структуры, задаваемой операторами передачи управления (ветвления, циклы, вызовы подпрограмм и т.д.).

Можно выделить 5 следующих источников сложности программирования: решаемая задача; язык программирования; среда выполнения программы; технологический процесс коллективной разработки; стремление к универсальности и эффективности алгоритмов и типов данных.

От сложности нельзя избавиться, но можно изменить характеристики ее проявления.

Объектно-ориентированное программирование

ООП является третьим крупным этапом (после структурного и модульного программирования) в процессе развития структурного подхода. Создаваемые в середине 70-х годов большие программные системы продемонстрировали, что в рамках процедурно ориентированного стиля использование структурного подхода не дает желаемого результата. С увеличением числа компонентов в создаваемых программных системах число ошибок, связанных с неправильным использованием процедур, некорректным использованием связей между компонентами, стало нелинейно расти. Сроки ввода в эксплуатацию этих систем постоянно срывались. Уменьшить число подобных ошибок и упростить их обнаружение могла позволить алгоритмическая декомпозиция, ориентирующаяся на естественные элементы пространства решаемой задачи. В этом случае на этапе кодирования и отладки упрощалось сопоставление программистских конструкций с моделируемыми объектами. Такую декомпозицию станем называть объектно-ориентированным анализом пространства решаемой задачи или предметной области. Для описания результатов объектно-ориентированного анализа и последующего программного синтеза необходимы адекватные программные средства, построенные на определенных принципах. Введение типов данных обозначило еще одно направление развития технологии программирования. Типизация данных предназначена для облегчения составления программ и для автоматизации выявления ошибок использования данных в виде фактических параметров при вызове функций. Результатом обобщения понятия «тип данных» являются классы объектов (C++) или объектные типы (Object Pascal), которые могут содержать в качестве элементов не только данные определенного типа, но и методы их обработки – функции и процедуры.

Классы

Классами называются специальные типы, которые содержат поля, методы и свойства. Конкретный экземпляр класса называется объектом. В основе классов лежат три фундаментальных принципа: инкапсуляция, наследование и полиморфизм.

Инкапсуляция

Класс представляет собой единство трех сущностей – полей, методов и свойств. Объединение этих сущностей в единое целое и называется инкапсуляцией. Инкапсуляция позволяет во многом изолировать класс от остальных частей программы, сделать его самодостаточным для решения конкретной задачи. Например, в объектно-ориентированном языке программирования Object Pascal класс TForm содержит все необходимое для создания Windows-окна, класс TМemo представляет собой полнофункциональный текстовый редактор, класс обеспечивает работу программы с таймером и т. д. Инкапсуляция представляет собой мощное средство обмена готовыми к работе программными заготовками.

Наследование

Любой класс может быть порожден от другого класса. Для этого при его объявлении указывается имя класса родителя. Порожденный класс автоматически наследует поля, методы и свойства своего родителя и может дополнять их новыми.

Полиморфизм

Полиморфизм – это свойство классов решать схожие по смыслу задачи разными способами. Изменяя алгоритм того или иного метода в потомках класса, программист может придать этим потомкам отсутствующие у родителя специфические свойства. Для изменения метода необходимо перекрыть его в потомке, т. е. объявить в потомке одноименный метод и реализовать в нем нужные действия.

Поля

Полями называются инкапсулированные в классе данные. Поля могут быть любого типа, в том числе – классами. Каждый объект получает уникальный набор полей, но общий для всех объектов данного класса набор методов и свойств. Фундаментальный принцип инкапсуляции требует обращаться к полям только с помощью методов и свойств класса.

Методы

Инкапсулированные в классе процедуры и функции называются методами. В состав любого класса входят два специальных метода конструктор и деструктор, Конструктор распределяет объект в динамической памяти и помещает адрес этой памяти в специальную переменную, которая автоматически объявляется в классе. Деструктор удаляет объект из динамической памяти.

Свойства

Свойства – это специальный механизм классов, регулирующий доступ к полям. Обычно свойство связано с некоторым полем и указывает те методы класса, которые должны использоваться при записи в это поле или при чтении из него.

Компоненты и события

В визуальных системах программирования каркас программы, во всяком случае – ее видимые на экране внешние проявления: окна, кнопки, списки выбора и т.д. создаются программистом из компонентов. Под компонентом понимается некий функциональный элемент, содержащий определенные

свойства и размещаемый программистом в специальном окне (окне формы). Способность программы выполнять полезную работу определяется совокупностью ее реакций на те или иные события. В связи с этим каждый компонент помимо свойств характеризуется также набором событий, на которые он может реагировать. Например, компонент формы «кнопка» может по-разному реагировать на однократный щелчок мыши и двойной щелчок мыши.

ПРОЕКТИРОВАНИЯ СЛОЖНЫХ СИСТЕМ

Технология проектирования ИС. Современные информационные технологии предоставляют широкий выбор способов реализации ИС, выбор которых осуществляется на основе требований со стороны предполагаемых пользователей, которые, как правило, изменяются в процессе разработки. Для теории принятия решений процесс проектирования ИС - это процесс принятия проектно-конструкторских решений, направленных на получение проекта системы, удовлетворяющего требованиям заказчика. Под проектом ИС будет понимать проектно-конструкторскую и технологическую документацию, в которой представлено описание проектных решений по созданию и эксплуатации ИС в конкретной программно-технической среде. Под проектированием ИС понимается процесс преобразования входной информации об объекте проектирования, о методах проектирования и об опыте проектирования объектов аналогичного назначения в соответствии со стандартами в проект ИС. С этой точки зрения проектирование ИС сводится к последовательной формализации проектных решений на различных стадиях жизненного цикла ИС. Объектами проектирования ИС являются отдельные элементы или их компоненты функциональных и обеспечивающих частей. Так, функциональными элементами в соответствии с традиционной декомпозицией выступают задачи, комплексы задач и функции управления. В составе обеспечивающей части ИС объектами проектирования служат элементы и их компоненты информационного, программного и технического обеспечения системы. В качестве субъекта проектирования ИС выступают коллективы специалистов, которые осуществляют проектную деятельность, как правило, в составе специализированной проектной организации, и организация-заказчик, для которой необходимо разработать ИС. При большом объеме и жестких сроках выполнения проектных работ в разработке системы может принимать участие несколько проектных коллективов. В этом случае выделяется головная организация, которая координирует деятельность всех организаций-соисполнителей. Осуществление проектирования ИС предполагает использование проектировщиками определенной технологии проектирования, соответствующей масштабу и особенностям разрабатываемого проекта. Технология проектирования - это совокупность методологии и средств проектирования ИС, а также методов и средств организации проектирования. Методология (концепция + метод) Инструментальные средства проектирования Организация проектирования. В основе технологии проектирования лежит технологический процесс, который определяет действия, их последовательность, состав исполнителей, средства и

(единая система технологической документации), ЕСПД (единая система программной документации). Этот комплект документации предназначен для использования в процессе изготовления и эксплуатации объекта проектирования. Важное значение в этих описаниях имеют математические модели объектов проектирования, так как выполнение проектных процедур при автоматизированном проектировании основано на оперировании математическими моделями. Математическая модель (ММ) технического объекта - система математических объектов (чисел, переменных, матриц, множеств и т.п.) и отношений между ними, отражающих некоторые свойства технического объекта. При проектировании используют математические модели, отражающие свойства объекта, существенные с позиции проектировщика. Среди свойств объекта, отражаемых в описаниях на определенном иерархическом уровне, в том числе в ММ, различают свойства: систем; элементов систем и внешней среды, в которой должен функционировать объект. Количественное выражение этих свойств осуществляется с помощью величин, называемых параметрами. Величины, характеризующие свойства системы, элементов системы и внешней среды, называют соответственно входными, внутренними и внешними параметрами. Однако существование ММ не означает, что она известна разработчику и может быть представлена в явном функциональном виде. Типичной является ситуация, когда математическое описание процессов в проектируемом объекте задается моделью в форме системы уравнений, в которой фигурирует вектор фазовых переменных. Фазовые переменные характеризуют физическое или информационное состояние объекта, а их изменения во времени выражают переходные процессы в объекте. Например, состояние некоторой фирмы можно определить такими фазовыми переменными: сырье, материалы, финансовые и трудовые ресурсы. Выделим следующие особенности параметров в моделях проектируемых объектов: 1. Внутренние параметры в моделях k -го иерархического уровня становятся выходными параметрами в моделях более низкого ($A_k + 1$)-го иерархического уровня. Так, например, трудовые ресурсы являются внутренними при проектировании производственной фирмы и в то же время выходными при проектировании отдела кадров этой фирмы. 2. Выходные параметры или фазовые переменные, фигурирующие в модели одной из подсистем (в одном из аспектов описаний), часто оказываются внешними параметрами в описании других подсистем (других аспектов). Так, например, выходные параметры подсистемы планирования выпуска продукции некоторой компании являются внешними параметрами подсистемы материально-технического снабжения этой компании. 3. Большинство выходных параметров объекта являются функционалами. 4. В техническом задании на проектирование должны фигурировать величины, называемые техническими требованиями к выходным параметрам (нормами выходных параметров). Данные нормы представляют собой границы допустимых диапазонов изменения выходных параметров.

проектирование называют нисходящим. Если раньше выполняются этапы, связанные с низшими иерархическими уровнями, то проектирование называют восходящим. У каждого из этих двух видов проектирования имеются преимущества и недостатки. Е[ри нисходящем проектировании система разрабатывается в условиях, когда ее элементы еще не определены и, следовательно, сведения о их возможностях и свойствах носят предположительный характер. Е[ри восходящем проектировании, наоборот, элементы проектируются раньше системы, и, следовательно, предположительный характер имеют требования к системе. В обоих случаях из-за отсутствия исчерпывающей исходной информации имеют место отклонения от возможных оптимальных технических результатов. Е[оскольку принимаемые предположения могут не оправдаться, часто требуется повторное выполнение проектных процедур предыдущих этапов после выполнения проектных процедур последующих этапов. Такие повторения обеспечивают последовательное приближение к оптимальным результатам и обуславливают итерационный характер проектирования. Следовательно, итерационность нужно относить к важным принципам проектирования сложных объектов. На практике обычно сочетают восходящее и нисходящее проектирование. Например, восходящее проектирование имеет место на всех тех иерархических уровнях, на которых используются унифицированные (стандартные) элементы. Очевидно, что унифицированные элементы, ориентированные на применение в ряде различных систем определенного класса, разрабатываются раньше, чем та или иная конкретная система из этого класса. Обычно унификация объектов имеет целью улучшение техникоэкономических показателей производства и эксплуатации изделий. Использование типовых и унифицированных проектных решений приводит так же к упрощению и ускорению проектирования. Однако, унификация целесообразна только в таких классах объектов, в которых из сравнительно небольшого числа разновидностей элементов предстоит проектирование и изготовление большого числа систем. Именно эти разновидности элементов подлежат унификации. Для сложных систем и для элементов, реализующих новые физические принципы или технологические возможности, в каждом конкретном случае приходится заново выполнять многоуровневое иерархическое проектирование. В этих условиях целесообразно ставить вопрос не об унификации изделий, а об унификации средств их проектирования и изготовления, в частности об унификации проектных процедур в рамках систем автоматизированного проектирования (САПР). Наличие средств автоматизированного выполнения типовых проектных процедур позволяет оперативно создавать проекты новых изделий, а в сочетании со средствами изготовления в условиях гибких автоматизированных производств осуществлять оперативное изготовление новых оригинальных изделий. Окончательное описание проектируемого объекта представляет собой полный комплект схемной, конструкторской и технологической документации, оформленной в соответствии с требованиями ГОСТов: ЕСКД (единая система конструкторской документации), ЕСТД

При проектировании сложных объектов используются следующие принципы : декомпозиция и иерархичность построения описаний объектов проектирования; многоэтапность и итерационность процесса проектирования; - типизация и унификация проектных решений. Описания технических объектов должны быть по сложности согласованы: с возможностями восприятия человеком; с возможностями оперирования описаниями в процессе их преобразования с помощью имеющихся средств проектирования. Выполнить это требование в рамках единого описания удастся лишь для простых изделий. Как правило, требуется структурирование описаний и соответствующее разбиение представлений об объекте на иерархические уровни и аспекты. Это позволяет распределить работы по проектированию сложных объектов между подразделениями проектировщиков, что способствует повышению эффективности и производительности труда. Разделение описаний по степени детализации отображаемых свойств и характеристик объекта лежит в основе блочно-иерархического подхода к проектированию и приводит к появлению иерархических уровней (уровней абстрагирования) в представлениях об объекте. Подобное разбиение продолжается вплоть до получения на некотором уровне элементов, описания которых дальнейшему делению не подлежат, то есть до элементов, описание которых уже известно. Такие элементы по отношению к объекту S называются базовыми элементами. Принцип иерархичности означает структурирование представлений об объекте проектирования по степени детальности описаний (уровни описаний - по вертикали) Принцип декомпозиции (блочности) означает разбиение представлений каждого уровня на ряд составных частей (блоков) с возможностью раздельного (поблочного) проектирования объектов S_i на уровне 1, объектов S_j на уровне 2 и т.д. Кроме разбиения описаний по степени подробности отражения свойств объектов используют декомпозицию описаний по характеру отображаемых свойств объекта. Такая декомпозиция приводит к появлению ряда аспектов описаний. Наиболее крупные аспекты описаний объектов: функциональный; конструкторский; технологический. Решение задач, связанных с преобразованием или получением описаний, относящихся к этим аспектам, называют соответственно функциональным, конструкторским и технологическим проектированием. Функциональный аспект связан с отображением основных принципов функционирования, характера физических и информационных процессов, протекающих в объекте. Функциональный аспект отображается в принципиальных, функциональных, структурных и других схемах и сопровождающих их документах. Конструкторский аспект связан с реализацией результатов функционального проектирования, то есть с определением геометрических форм объектов и их взаиморасположением в пространстве. Технологический аспект относится к реализации результатов функционального и конструкторского проектирования, т.е. связан с описанием методов и средств изготовления объектов. Внутри каждого аспекта возможно свое специфическое выделение иерархических уровней. Если решение задач высоких иерархических уровней предшествует решению задач более низких иерархических уровней, то

ресурсы, требуемые для выполнения этих действий. Так, технологический процесс проектирования ИС в целом делится на совокупность последовательно-параллельных, связанных и соподчиненных цепочек действий. Действия, которые выполняются при проектировании ИС, могут быть определены как неделимые технологические операции или как подпроцессы технологических операций. Все действия могут быть собственно проектировочными, которые формируют или модифицируют результаты проектирования, и оценочными действиями, которые вырабатывают по установленным критериям оценки результатов проектирования. Таким образом, технология проектирования задается регламентированной последовательностью технологических операций, выполняемых в процессе создания проекта на основе того или иного метода, в результате чего стало бы ясно, не только что должно быть сделано для создания проекта, но и как и в каком порядке это должно быть сделано. К основным требованиям, предъявляемым к выбираемой технологии проектирования, относятся следующие: созданный с помощью этой технологии проект должен отвечать требованиям заказчика; выбранная технология должна максимально отражать все этапы цикла жизни проекта; выбираемая технология должна обеспечивать минимальные трудовые и стоимостные затраты на проектирование и сопровождение проекта; технология должна быть основой связи между проектированием и сопровождением проекта; технология должна способствовать росту производительности труда проектировщика; - технология должна обеспечивать надежность процесса проектирования и эксплуатации проекта; технология должна способствовать простому ведению проектной документации. Основу технологии проектирования ИС составляет методология, которая определяет сущность, основные отличительные технологические особенности. Методология проектирования предполагает наличие некоторой концепции, принципов проектирования, реализуемых наборов методов проектирования, которые, в свою очередь, должны поддерживаться некоторыми средствами проектирования. Организация проектирования предполагает определение методов взаимодействия проектировщиков между собой и с заказчиком в процессе создания проекта ИС, которые могут также поддерживаться набором специфических средств. Для конкретных видов технологий проектирования свойственно применение определенных средств разработки ИС, которые поддерживают выполнение как отдельных проектных работ, этапов, так и их совокупностей. Поэтому перед разработчиками ИС, как правило, стоит задача выбора средств проектирования, которые по своим характеристикам в наибольшей степени соответствуют требованиям конкретного предприятия. Средства проектирования должны быть: в своем классе инвариантными к объекту проектирования; охватывать в совокупности все этапы жизненного цикла ИС; технически, программно и информационно совместимыми; простыми в освоении и применении; экономически целесообразными.

Принципы проектирования сложных объектов.

Лекция № 6. Этапы процесса проектирования ИС

Процесс создания ИС делится на ряд этапов (стадий), ограниченных некоторыми временными рамками и заканчивающихся выпуском конкретного продукта (моделей, программных продуктов, документации и пр.). Выделяют следующие этапы создания ИС: формирование требований к системе, проектирование, реализация, тестирование, ввод в действие, эксплуатация и сопровождение. Суть содержания жизненного цикла разработки системы в различных подходах одинакова и сводится к выполнению следующих стадий.

1. Планирование и анализ требований (предпроектная стадия) - системный анализ. Исследование и анализ существующей системы, определение требований к создаваемой системе, оформление техникоэкономического обоснования и технического задания на разработку системы.
2. Проектирование (техническое проектирование, логическое проектирование). Разработка в соответствии со сформулированными требованиями состава автоматизируемых функций и состава обеспечивающих подсистем, оформление технического проекта системы.
3. Реализация проекта (рабочее проектирование, физическое проектирование, программирование). Разработка и настройка программ, наполнение БД, создание рабочих инструкций для персонала, оформление рабочего проекта.
4. Внедрение (тестирование, опытная эксплуатация). Комплексная отладка подсистем, обучение персонала, поэтапное внедрение системы в эксплуатацию, оформление акта о приемо-сдаточных испытаниях системы.

5. Эксплуатация системы (сопровождение, модернизация). Сбор рекламаций и статистики о функционировании системы, исправление ошибок и недоработок, оформление требований к модернизации системы и ее выполнение. 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 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 БД. Полученная в процессе анализа информационная модель сначала преобразуется в логическую, а затем в физическую модель данных. Параллельно с проектированием схемы БД 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 БД (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 БД централизованной или распределенной. Если БД будет распределенной, то какие механизмы поддержки согласованности и актуальности данных будут использоваться; будет ли БД однородной, т. е. будут ли все серверы БД продуктами одного и того же производителя (например, все серверы только Oracle или все серверы только DB2 UDB). Если БД 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 БД (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 пройден, 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, т. е. 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 — приемосдаточные 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.*

При проектировании автоматизированной ИС необходимо спланировать комплекс работ по созданию системы в соответствии с типовыми этапами разработки ИС, краткая характеристика которых приведена в табл 4.1.

Этапы проектирования ИС и их характеристики

Таблица 4.1.

Наименование этапа	Основные характеристики
--------------------	-------------------------

<p>Разработка модели и анализ бизнес-модели</p>	<p>Определяются основные задачи ИС, проводится декомпозиция задач по модулям и определяются функции с помощью которых решаются эти задачи. Описание функций осуществляется на языке производственных (описание процессов предметной области), функциональных (описание форм обрабатываемых документов) и технических требований (аппаратное, программное, лингвистическое обеспечение ИС).</p> <p>Результат:</p> <ol style="list-style-type: none"> 1. Концептуальная модель ИС, состоящая из описания предметной области, ресурсов и потоков данных, перечень требований и ограничений к технической реализации ИС. 2. Аппаратно-технический состав создаваемой ИС.
<p>Формализация бизнес-модели, разработка логической модели бизнес-процессов.</p>	<p>Разработанная концептуальная модель формализуется, т.е. воплощается в виде логической модели ИС.</p> <p>Результат: разработанное информационное обеспечение ИС: схемы и структуры данных для всех уровней модульности ИС, документация по логической структуре ИС, сгенерированные скрипты для создания объектов БД.</p>
<p>Выбор лингвистического обеспечения, разработка программного обеспечения ИС.</p>	<p>Разработка ИС: выбирается лингвистическое обеспечение (среда разработки - инструментарий), проводится разработка программного и методического обеспечения. Разработанная на втором этапе логическая схема воплощается в реальные объекты, при этом логические схемы реализуются в виде объектов БД, а функциональные схемы - в пользовательские формы и приложения.</p> <p>Метод решения: разработка программного кода с использованием выбранного инструментария.</p> <p>Результат: работоспособная ИС.</p>
<p>Тестирование и отладка ИС</p>	<p>На данном этапе осуществляется корректировка информационного, аппаратного, программного обеспечения, проводится</p>

	<p>разработка методического обеспечения (документации разработчика, пользователя) и т.п.</p> <p>Результат: оптимальный состав и эффективное функционирование ИС.</p> <p>Комплект документации: разработчика, администратора, пользователя.</p>
Эксплуатация и контроль версий	<p>Особенность ИС созданных по архитектуре клиент сервер является их многоуровневость и многомодульность, поэтому при их эксплуатации и развитии на первое место выходят вопросы контроля версий, т.е. добавление новых и развитие старых модулей с выводом из эксплуатации старых. Результат: наращиваемость и безызбыточный состав гибкой, масштабируемой ИС.</p>

Лекция №7. Модели процесса проектирования ИС.

На этапе проектирования информационной системы в соответствии с описанными выше принципами и выбранным подходом создаются различные модели, которые описывают ИС. Эти модели могут быть классифицированы по разным признакам.

По отображаемому аспекту.

Функциональная модель, которая описывает варианты использования ИС, ее функциональность, объекты и субъекты взаимодействия и т.п. Функциональные модели бывают как статическими, так и динамическими.

Информационная модель, которая описывает состав и структуру данных (реляционные БД, базы классов и пр.). Информационные модели относятся к статическим.

Поведенческие модели применяются для описания состояний системы и переходов между ними, алгоритмов взаимодействия и обработки информации. Поведенческие модели относят к динамическим.

Компонентные модели, при помощи которых описывается состав и структура программных и аппаратных средств. Такие модели являются статическими.

По степени отображения динамики процессов. Статические модели описывают состав и структуру модели. Динамические модели описывают поведение системы или ее элементов. Главная особенность динамических моделей — явное или косвенное использование понятия «время». По степени физической реализации. Логические модели, которые описывают структуру ИС, ее состав, состояние, поведение, но без привязки к языкам программирования, СУБД, аппаратному обеспечению и пр. Физические модели, описывающие элементы ИС в контексте их конкретной физической реализации. По строгости описания. Неформальные модели, которые не имеют

общепринятого структурированного вида и могут дать только наиболее общее представление об ИС или ее элементе. Как правило, неформальные модели не очень информативны и не могут использоваться для дальнейшего анализа, но иногда их использование необходимо — например, для взаимодействия с бизнес-экспертами.

Формальные модели имеют три подвида. *Описательные* модели основаны на представлении при помощи специальных документов (бланков, отчетов, таблиц и т.п.). *Графические* модели используют схемы, диаграммы, чертежи и пр. *Математические* модели иллюстрируют ИС в виде математических зависимостей, систем уравнений, логических выражений и пр.

Как правило, проектирование ИС начинается с неформальных моделей, и только в итоге происходит их постепенная формализация. Логическое проектирование в соответствии с изложенными ранее принципами также происходит раньше, чем наступает время физического проектирования. Динамические и статические модели могут создаваться параллельно в одно и то же время, по гораздо чаще первыми создаются статические модели.

Информационная модель и моделирование информационных процессов

Модель — одна из основных категорий теории познания. В широком смысле модель — любой образ (изображение, карта, описание, схема, чертёж, график, план и другое) какого-либо объекта, процесса или явления, используемый в качестве их “заместителя” или “представителя”. Модель (лат. “modulus” — мера) — это объект-заместитель объекта-оригинала, обеспечивающий изучение некоторых свойств последнего; упрощённое представление системы для её анализа и предсказания, для получения качественных и количественных результатов, необходимых для принятия правильного управленческого решения. Модель — это создаваемое человеком подобие изучаемых объектов: макеты, изображения, схемы, словесные описания, математические формулы, карты и т.д. Моделирование — это представление объекта моделью для получения информации о нём путём проведения экспериментов с его моделью. Моделирование облегчает изучение объекта с целью его создания, дальнейшего преобразования и развития. Существует два основных вида моделирования: аналитическое и имитационное. Для управления бизнес процессами (англ. “Business Process Management”, BPM) в современных системах используют методы имитационного моделирования. На идею модели по существу базируется любой метод научного исследования, как теоретический (при котором используются различного рода знаковые, абстрактные модели), так и экспериментальный, использующий предметные модели. Модели предметной области — это совокупность описаний, обеспечивающих взаимопонимание между пользователями: специалистами организации и разработчиками. Модели всегда проще реальных объектов, но они позволяют выделить главное, не отвлекаясь на детали. Различают математические, физические, ситуационные, электрические, информационные модели. Так, например, математические модели используют для описания объектов и процессов живой и неживой природы и технологии, в том числе — в физике, биологии,

экономике. Информационная модель - это модель объекта, процесса или явления, включающая информацию в качестве основной составляющей моделируемого объекта, процесса или явления. Наиболее очевидными с точки зрения применения методов моделирования, несомненно, являются процессы управления, где на основе полученной информации необходимо принимать соответствующие решения. Обычно моделирование используется для исследования существующей системы, когда реальный эксперимент проводить нецелесообразно из-за значительных финансовых и трудовых затрат, а также при необходимости проведения анализа проектируемой системы, т.е. которая ещё физически не существует в данной организации. Для человека информационная модель является источником информации, на основе которой он формирует образ реальной обстановки. Однозначного понятия системы нет. В общем виде под системой понимают совокупность взаимосвязанных элементов, образующих определённую целостность, единство. Процесс построения модели является творческой процедурой, трудно поддающейся формализации. Модельные представления являются абстрактными образами элементов системы (объектов, технических средств, программного обеспечения и др.). Вместе они позволяют получить достаточно полное представление о создаваемой системе. Количество групп элементов информационной модели определяется степенью детализации описания состояний и условий функционирования объекта управления. Как правило, элемент информационной модели связан с каким-либо параметром объекта управления. Модель данных является способом отображения самих данных и их связей. Выделяют модели иерархических, сетевых и реляционных данных, как правило, входящих в состав систем управления базами данных (СУБД). В СУБД реализуются модели процессов накопления и применения информации и знаний. В качестве инструментальных многофункциональных информационных моделей применяют, например, модели VIEW (англ. "Virtual Instruments Engineering Workbench"). Для формирования модели используются: длежащего автоматизации; структурно-функциональная схема автоматизируемого объекта; Главная цель проведения моделирования любой системы - изыскание вариантов решений, которые позволяют улучшить основные показатели её деятельности. Необходимым элементом моделирования является анализ потоков данных. Спрос на средства аналитической обработки данных постоянно растёт. При этом пользователи заинтересованы в получении средств, позволяющих автоматически искать не только заданные данные, но неочевидные правила и неизвестные закономерности. Для реализации подобных систем используют методы интеллектуального анализа данных, позволяющие на основе накопленной информации принимать нетривиальные решения и генерировать качественно новые знания, способствующие повышению эффективности решений и деятельности людей, предприятий, организаций и т.п. Логика интеллектуально решаемых аналитических задач заключается в том, что первичные документы, отчёты и сводные таблицы анализируются с целью выявления полученных показателей. Исследование произошедших событий и

полученных результатов (Что произошло?) происходит с целью ответа на вопрос "Почему?". В результате проведённого анализа формируются прогностические (прогнозные) модели, в которых даются варианты развития ситуации. Сбор, обработка и анализ реальных данных функционирования системы или объекта моделирования даёт требуемые количественные оценки для разработки вариантов программно-технического обеспечения автоматизированных систем. При моделировании сложных объектов нельзя разобщать решаемые задачи. В противном случае получатся значительные затраты ресурсов и потери при реализации модели на конкретном объекте. Использование моделирования применительно к таким объектам требует одновременного исследования их взаимосвязей с внешней средой и другими элементами метасистемы. Под сложными системами понимаются системы, обладающие большим числом элементов, свойства которых не могут быть предсказаны, опираясь на знания свойств отдельных частей системы и способы их соединения.

2. Системный подход к решению функциональных задач

Наиболее эффективно моделирование осуществляется при использовании общих принципов системного подхода, лежащего в основе теории систем. Он возник в процессе изучения различных объектов и явлений. К этому методу обычно прибегают при анализе и синтезе больших (сложных) систем. Он предполагает рассмотрение системы путём последовательного перехода от общего к частному, когда исследуемый объект выделяется из окружающей среды. Системный подход - это направление научных исследований, основанных на рассмотрении сложного объекта как целого множества элементов, включающего отношения и связи между ними; это общеметодический принцип, используемый в различных предметных областях.

Сущность системного подхода - это интеграция интеллектуальных усилий общества. Системный подход базируется на первоначальном исследовании системы, выделении элементов, внутренних и внешних связей, существенным образом влияющих на исследуемые объекты и их функционирование в различных условиях и средах. Он начинается с изучения и детализации её составляющих - разбиения системы на функциональные подсистемы, которые тоже могут делиться. Процесс деления осуществляется до выявления конкретных процедур. При этом исследуемая система сохраняет целостность. Таким образом, разработка системы ведётся сверху вниз, последовательно приближаясь к конечному результату. Системный подход позволяет рассматривать различные объекты (системы) с единой точки зрения, выявляя важнейшие черты их функционирования и учитывая наиболее существенные факторы. Он эффективно используется при проектировании и эксплуатации информационных систем, особенно автоматизированных систем управления (АСУ), которые по существу являются человеко-машинными системами. Системный подход подразумевает построение моделей. Обычно эти модели имеют иерархическую структуру, отражающую различные особенности поведения такого сложного объекта, как предприятие, например модель процессов, представление предприятия как разновидности систем массового обслуживания, модель данных, используемых на

предприятия и т.д. При применении системного подхода учитывают все факторы проектируемой системы: функциональные, психологические, социальные, эстетические. Вне системного подхода, в отсутствие необходимой координации функционирования элементов никакая система не может достичь своей цели. В этой связи актуализируется роль координатора автоматизированного производства, способного организовать деятельность всех его участников. Использование системного подхода позволяет принять во внимание множество факторов самого различного характера, выделить из них те, которые оказывают самое большое влияние с точки зрения имеющихся общесистемных целей и критериев, и найти пути и методы эффективного воздействия на них. Системный подход позволяет рассматривать анализ и синтез различных по своей природе и сложности объектов с единой точки зрения, выявляя при этом важнейшие характерные черты функционирования системы и учитывая наиболее существенные для всей системы факторы. Важным для системного подхода является определение структуры системы, понимаемой как совокупность связей между элементами, отражающими их взаимодействие. Наиболее общее – топологическое описание структуры – позволяет в самом общем виде определить составные части системы и формализуется с помощью теории графов. Менее общим считается функциональное описание, предназначенное для рассмотрения поведения отдельных функций (алгоритмов) системы. Оно реализуется с помощью функционального подхода. Системный подход позволяет выявить общие (типовые) процессы, элементы функционирования системы. На этой основе разрабатываются адекватные решения (рекомендации) по использованию информационных технологий с целью улучшения качества функционирования информационных систем и обслуживания с их помощью различных категорий пользователей. Важным элементом системного подхода можно считать качественный анализ, основанный на определении функций, их характеристик и возможностей использования в том или ином процессе. Например, в результате качественного анализа можно определить затраты на организацию системы, которые могут превысить доход от её использования. Однако фактор экономической целесообразности (окупаемость вложений) не всегда может превалировать при решении вопросов использования информационных технологий, например в библиотеках. При разработке моделей и формировании информационных систем, ориентированных на создание информационных продуктов и услуг разБессмертная классика Waterfall. Среди вопросов, неизбежно возникающих в каждом проекте, выделяется один: как целесообразнее управлять процессом разработки продукта? Один из вариантов ответа проверенных годами — Waterfall (или каскадная, водопадная модель управления разработкой ПО).

Правда, сейчас эта методика нещадно критикуется, но так ли все плохо на самом деле или мы в очередной раз придем к тому, что все новое — хорошо забытое старое?

Жизненный цикл разработки программного обеспечения

Практически каждая команда разработчиков может создавать свою модель жизненного цикла ПО или использовать что-то общепризнанное. Один из вариантов — Waterfall. «Родителем» такой модели считается американец В. У. Ройс, который, по слухам, многое позаимствовал у коллег, присвоив себе лавры. Случилось это в 1970 году. До сегодняшнего дня во многих проектах используется описанный им подход: в первоначальном варианте или с доработками.

Хотя некоторые участники ИТ говорят о том, что такой методологии «отродясь-то не бывало»:

Как ИТ-профессионал и преподаватель я в течение более чем 40 лет слышал много мифов об ИТ-индустрии. Но что продолжает удивлять меня, так это то, почему слово «Waterfall» до сих пор используется для описания методологии, которая не существует и почему создатели методов разработки систем используют его в качестве источника сравнения.

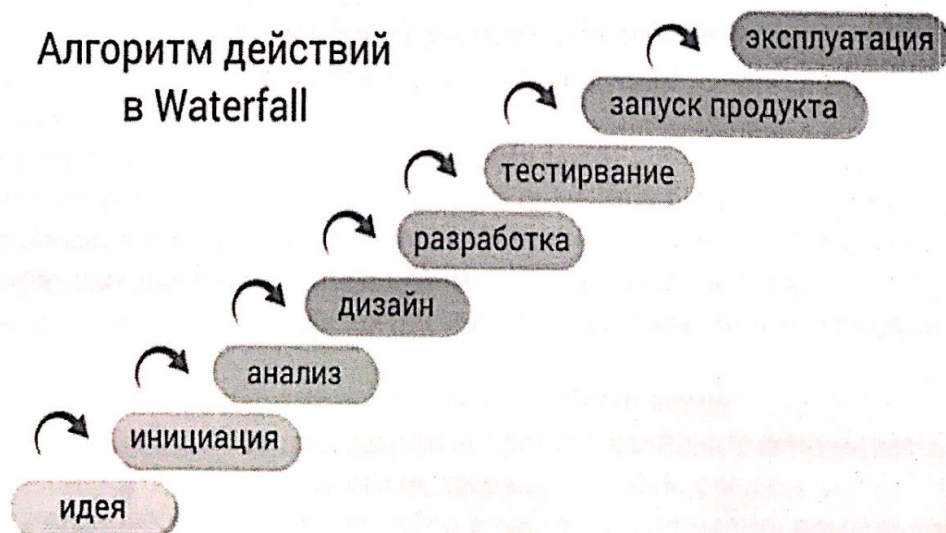
David Dischave,

профессор школы информационных технологий университета Сиракузы, США

Удивительно слышать такое о методологии, которая не один десяток лет используется в создании ПО для самых разных сфер деятельности, в том числе для автомобилестроения, строительства зданий и сооружений, финансового сектора, медицины и электроники.

Waterfall в сфере ИТ

Основной постулат Waterfall модели разработки ПО заключается в том, что следующий этап не может быть начат, пока не завершён предыдущий. При этом произвольные переходы вперед или назад не допускаются, а этапы не перекрывают друг друга. В этом и заключается основное отличие каскадной методологии от гибких собратьев (или конкурентов): Agile, DSDM, Scrum, FDD.



Чтобы понять мысли Ройса, заложенные в основу модели, можно изучить его труд в оригинале: Royce, Winston (1970), *Managing the Development of Large Software Systems*.

Процесс работы, основанный на каскаде

Возникновение идеи и ее обсуждение

На этом этапе о разработке как таковой речи не идет, просто рассматривается некая появившаяся идея, интересная одному или нескольким людям.

Анализ требований

Этап, на котором требования заказчика к проекту описываются в мельчайших деталях, также решается, какими способами будет достигнута цель, обозначаются сроки завершения работ и финансовая составляющая. При этом обычно закладывается некий запас времени и денег для каждого звена работы.

По окончании анализа требований в наличии имеется ТЗ для программистов и бюджет.

Проектирование программного обеспечения

На этом этапе делаются конкретные шаги:

1. выбирается платформа программирования (Python, PHP, JS и пр.)
2. уточняются технические детали (например, как будут взаимодействовать сервис или продукт с серверами, станут ли использовать API, какой будет логика внешнего и внутреннего интерфейса и т.д.)
3. решаются вопросы безопасности проекта (например, будет ли использоваться HTTPS, SSL-шифрование и пр.)
4. описываются роли пользователей программного продукта (администратор, клиент, менеджер и пр.)
5. финализируются вопросы надежности, производительности и дальнейшей техподдержки целевого продукта
6. формируется конкретная команда.

Разработка программного обеспечения

Этап, на котором пишется код, соответствующий документации, разработанной ранее.

Тестирование программного обеспечения

Готовая версия продукта тестируется специалистами в условиях, приближенных к боевым, выявляются и фиксируются баги. Наиболее катастрофичные для работы ПО в целом — исправляются, менее критичные — могут быть не исправлены, если нет времени или исчерпан бюджет.

Техническая поддержка программного обеспечения

Пригодный к работе программный продукт начинают использовать по назначению и осуществлять его поддержку. То есть: следить за работоспособностью, устранять сбои в работе, планировать расширение функционала на основе фидбека от пользователей.

Все перечисленные выше этапы выполняются строго последовательно, полученные результаты документируются.

Чтобы понять эволюцию классической водопадной методологии, описанной выше, можно изучить РМВОК. Между 3-ей и 4-й версиями есть ряд различий, которые помогут понять путь «каскада».

Плюсы и минусы каскадной модели разработки программного обеспечения

Ничего идеального в нашем мире, к сожалению, не существует, потому у каскадной методологии тоже есть сильные и слабые стороны.

Сильные стороны каскадной модели разработки ПО	Слабые стороны каскадной модели разработки ПО
предельная детализация каждого шага работы, сопровождающаяся документированием	затраты времени на ведение подробной документации, которая, к тому же, может быть не всегда понятной заказчику, и вызывать у него вопросы
требования максимально внятно и четко изложены, не могут противоречить друг другу или меняться в середине работы	необходимы квалифицированные бизнес-аналитики, способные сформулировать приемлемое для продуктивной работы ТЗ отсутствует возможность для маневра, если в процессе разработки выяснилось, что продукт не отвечает требованиям рынка
возможность заранее знать, сколько времени и денег будет потрачено на проект	затраты времени и денег достаточно высоки
легкость понимания методики как таковой даже для не самых опытных разработчиков <u>легкость контроля</u> и, при необходимости, передачи проекта другой команде, благодаря строгой системе отчетности.	высокая вероятность выявления критических проблем уже на завершающем этапе разработки, причем их устранение на этапе готового продукта обходится чрезмерно дорого.

Как и когда использовать каскадную модель разработки?
 Как показывает практика, Waterfall модель разработки ПО вполне уместна в следующих случаях: заказчик участвует в проекте только на первом этапе и принимает готовый продукт; изменять требования к продукту не планируется; проект отличается высокой сложностью, длительностью и дороговизной; основной приоритет — качество, даже в ущерб времени; отсутствие команды разработчиков экстра-класса; допускается возможность выполнения проекта на аутсорсе.

Для понимания же мотивации отказа от каскадной методологии можно прочесть книгу «Scrum. Революционный метод управления проектами» Джеффа Сазерленда.

Примеры использования Waterfall

Каскадная модель в чистом виде в современной разработке не так уж распространена и, зачастую, то, что не подходит под определение Agile, нарекают Waterfall, поэтому достаточно сложно определить, где используется именно эта методология.

По оценкам экспертов, значительная часть ERP-систем, программ, предназначенных для строительства, медицины, работы с государственными контрактами, промышленностью и подобных фундаментальных целей разрабатывается при помощи той или иной модификации «водопада».

Понимание особенностей работы с такими проектами улучшает книга Сергея Зыкова «Основы проектирования корпоративных систем».

И это логично. Об этом говорит и Чак Кобб, автор книг, посвященных Agile-методам в проектном менеджменте, наставник, инструктор:

Если бы вы строили мост через реку, было бы смешно сказать: «Мы построим первый пролет, посмотрим, как это выходит, а затем решим, как закончить оставшиеся пролеты!»

Среди компаний, в которых использовали или используют Waterfall можно отметить:

Название компании	Для чего использовалась модель Waterfall	Используется ли сейчас методология	Комментарий представителя компании
Wüstenrot & Württembergische (W&W)	Разработка ERP-системы для финансового сектора	Нет данных	—

Cisco	Разработка систем безопасности	Да	—
EPAM	Разнообразные продукты и решения или их части	Да	Алексей Ионов: «...не обязательно весь большой проект делать по Agile — можно использовать гибкую разработку в рамках отдельных фаз или потоков работ...»
IBM	Разнообразные продукты и решения или их части	Да	Розалинда Рэдклифф: «Настало время, когда команды разработчиков, занимающиеся водопадом, не в силах выполнить требования бизнеса, поэтому эти проекты и продукты получают больше работ по техническому обслуживанию. .. Водопад будет постепенно заменен

			новыми технологиями и новыми командами, заинтересованными во внедрении новых бизнес-практик».	
IT	Microsoft	Разнообразные продукты и решения или их части	Нет	«За последние несколько лет... все наши команды приняли гибкую методологию. Мы обнаружили, что она решила многие проблемы, связанные с традиционной моделью водопада, где проекты планируются заблаговременно и могут занять месяцы или даже годы. В рамках этой модели продукт может быть устаревшим, как только мы его выпустили».
	AT Consulting	Разнообразные продукты и решения или их части	Да	Василий Кораблев: «Для разработки систем с нуля

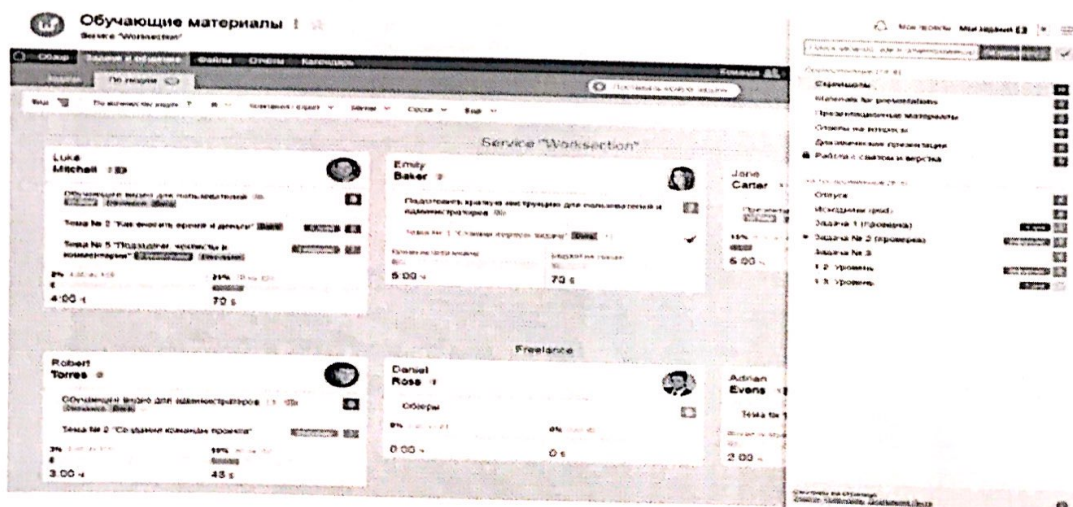
			мы используем один из подходов гибкой (Agile) или водопадной (waterfall) разработки, либо их комбинации».
Parallels	Разнообразные продукты и решения или их части	Да	Николай Добровольский : «В разных проектах у нас используются различные подходы — где-то Agile с одно-двухнедельным и спринтами, а где-то — почти что Waterfall с майлстоунами по несколько месяцев. За многие годы работы у нас сложилось ощущение, что чем больше проект и команда, работающая над ним, тем сложнее и менее эффективно стараться записать разработку

SAP	Разнообразные продукты и решения или их части	Да	в agile-процессе». Евгений Арнаутов: «На этапе создания продукта часто можно видеть варианты Agile, причем иногда совмещенные с Waterfall-подходом».
Toyota	Разнообразные продукты и решения или их части	Нет	Сатоши Ишии: "...мы пытаемся научиться применять TPS (Lean на западе) к разработке программного обеспечения».

Приложения и программы для управления разработкой по каскадной модели

Для работы с «водопадом» можно использовать ряд сервисов для постановки задач. Ключевые критерии — наличие тайм-трекера, канбан-досок и диаграммы Ганта.

Worksection



Привлекательный украинский saas-сервис с удобной мобильной версией.

Подходит для четкого планирования, потому что есть:

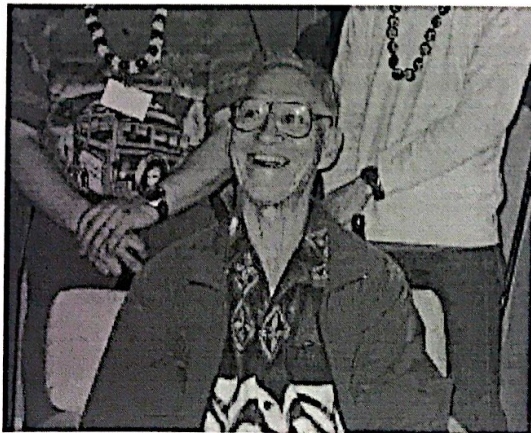
- диаграмма Ганта со связями между задачами и дедлайнами
- распределение обязанностей по исполнителям с разными правами
- система разных типов отчетов
- комментарии, эмоции и вся история действий по проекту сохраняется
- ограниченный доступ заказчика/клиента для прозрачности процесса разработки
- возможность внесения бюджета и расходов
- чек-листы для мелких этапов задач, что облегчает исполнение в точности по инструкции.

Waterfall — методология, используемая достаточно давно и, что бы не говорили критики, в ряде случаев — эффективно.

При этом для управления проектами, которые требуют быстрого реагирования на меняющиеся потребности рынка, данный подход не уместен в чистом виде. Исходя из комментариев представителей самых разных компаний, можно сделать вывод о том, что «водопад» имеет право на жизнь в современных проектах. Но его использование оправдано там, где требования точно не изменятся к моменту готовности проекта.

Лекция №9. Спиральная модель и архитектура разработки программного обеспечения

Спиральная модель — это модель процесса разработки программного обеспечения с учетом рисков. Это комбинация модели водопада и итеративной модели. Spiral Model помогает внедрить элементы разработки программного обеспечения из нескольких моделей процессов для программного проекта на основе уникальных шаблонов рисков, обеспечивая эффективный процесс разработки. Каждая фаза спиральной модели в разработке программного обеспечения начинается с определения цели проектирования и заканчивается тем, что клиент просматривает прогресс. История создания спиральная модель и архитектура разработки программного обеспечения



Барри Бозэм

Спиральная модель была впервые описана **Барри Бозм** в его статье 1986 г. «Спиральная модель разработки и улучшения программного обеспечения».

Барри Бозм родился в 1935 году. Бозм получил степень бакалавра по математике в Гарвардском университете, степень магистра в и степень доктора философии.

Область научных интересов Барри Бозма включает моделирование процесс разработки программного обеспечения архитектуры, метрики программного обеспечения и модели затрат.

В 1988 году **Барри Бозм** опубликовал аналогичную статью для более широкой аудитории. В этих статьях представлена диаграмма, которая была воспроизведена во многих последующих публикациях, посвященных спиральной модели. В этих ранних статьях используется термин «**модель процесса**» для обозначения спиральной модели.

Спиральная модель представляет собой процесс разработки программного обеспечения, сочетающий в себе как итеративность, так и этапность.

Барри Бозм в его статье *Spiral Development: Experience, Principles, and Refinements*, так пишет о спиральной модели

Спиральная разработка — это семейство процессов разработки программного обеспечения, для которых характерно повторение набора элементарных процессов разработки и управление рисками, с целью уменьшения рисков.

Отличительной особенностью этой модели является **специальное внимание рискам, влияющим на организацию жизненного цикла**. Бозм формулирует десять наиболее распространённых (по приоритетам) рисков:

1. Дефицит специалистов.
2. Нереалистичные сроки и бюджет.
3. Реализация несоответствующей функциональности.
4. Разработка неправильного пользовательского интерфейса.
5. «Золотая сервировка», перфекционизм, ненужная оптимизация и оттачивание деталей.
6. Непрекращающийся поток изменений.
7. Нехватка информации о внешних компонентах, определяющих окружение системы или вовлечённых в интеграцию.
8. Недостатки в работах, выполняемых внешними (по отношению к проекту) ресурсами.
9. Недостаточная производительность получаемой системы.
10. Разрыв между квалификацией специалистов и требованиями проекта.

Когда использовать спиральную модель?

Когда требуется частое изменения.

Когда проект большой.

Когда требования нечеткие и сложные.

Когда изменения могут потребоваться в любое время.

Крупные и высокобюджетные проекты.

Например, подобная модель используется Агентстве перспективных оборонных исследовательских проектов (DARPA) США.

И так, что такое спиральная модель

Спиральная модель — одна из наиболее важных моделей жизненного цикла разработки программного обеспечения, которая обеспечивает поддержку **управления рисками**. В схематическом представлении она выглядит как спираль с множеством петель. Точное количество витков спирали неизвестно и может варьироваться от проекта к проекту. **Каждый цикл спирали называется фазой процесса разработки программного обеспечения.** Точное количество этапов, необходимых для разработки продукта, может варьироваться менеджером проекта в зависимости от рисков проекта. Поскольку менеджер проекта динамически определяет количество этапов, он играет важную роль в разработке продукта с использованием спиральной модели. Радиус спирали в любой точке представляет собой затраты (стоимость) проекта на данный момент, а угловой размер представляет прогресс, достигнутый на текущий момент.

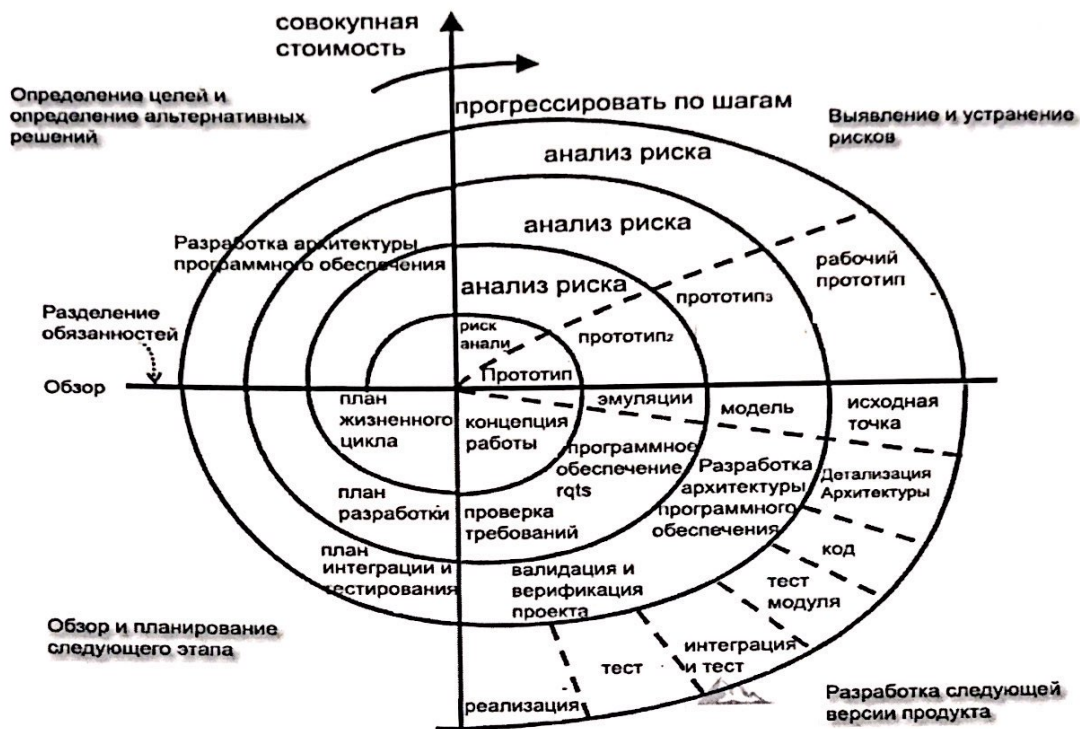
Spiral Model Спиральная модель и архитектура разработки программного обеспечения



Spiral Model Спиральная модель и архитектура разработки программного обеспечения

Фазы спиральной модели

Фазы спиральной модели	Действия, выполненные на этапе
Определение целей и определение альтернативных решений	требования собираются от клиентов, а цели определяются, разрабатываются и анализируются в начале каждого этапа. Затем в этом квадранте предлагаются альтернативные решения, возможные для фазы.
Выявление и устранение рисков	во втором квадранте оцениваются все возможные решения, чтобы выбрать наилучшее из возможных. Затем выявляются риски, связанные с этим решением, и риски устраняются с использованием наилучшей стратегии. В конце этого квадранта создается прототип для наилучшего возможного решения.
Разработка следующей версии продукта	в третьем квадранте выявленные функции разрабатываются и проверяются посредством тестирования. В конце третьего квадранта доступна следующая версия программного обеспечения.
Обзор и планирование следующего этапа	В четвертом квадранте заказчики оценивают уже разработанную версию программного обеспечения. В конце концов, начинается планирование следующего этапа.



Spiral Model Спиральная модель и архитектура разработки программного обеспечения от Барри Бозма

Риск — это любая неблагоприятная ситуация, которая может повлиять на успешное завершение программного проекта. Наиболее важной особенностью спиральной модели является управление этими неизвестными рисками после начала проекта. Такое решение рисков легче осуществить, разработав прототип. Спиральная модель помогает справляться с рисками, предоставляя возможности для создания прототипа на каждом этапе разработки программного обеспечения.

Преимущества Spiral Model

Обработка рисков: проекты с множеством неизвестных рисков, возникающих в процессе разработки, в этом случае спиральная модель является лучшей моделью разработки, которой следует следовать, благодаря анализу рисков и управлению рисками на каждом этапе.

Подходит для больших проектов: рекомендуется использовать спиральную модель в больших и сложных проектах.

Гибкость в требованиях: запросы на изменение в требованиях на более позднем этапе могут быть точно включены с помощью этой модели.

Удовлетворенность клиентов: заказчик может наблюдать за развитием продукта на ранней стадии разработки программного обеспечения и, таким образом, привык к системе, используя ее до завершения работы над продуктом.

Недостатки Spiral Model

Сложность: спиральная модель намного сложнее других моделей SDLC.

Дорого: спиральная модель не подходит для небольших проектов, так как она дорогая.

Слишком сильно зависит от анализа рисков: успешное завершение проекта во многом зависит от анализа рисков. Без очень большого опыта невозможно разработать проект с использованием этой модели.

Сложность в управлении временем: поскольку количество этапов неизвестно в начале проекта, поэтому оценка времени очень сложна.

Спецификация функциональных требований к ИС

Процессные потоковые модели

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

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

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

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

Главными недостатками *функционального подхода* являются:

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

Процессный подход предполагает смещение акцентов от управления отдельными структурными элементами 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.

" *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:

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

	Функциональная область 1	Функциональная область 2	Функциональная область 3	Функциональная область N
Процесс 1 (→) операции, ↑ исполнитель	↑ ⇒	→	↑ ⇒	↑ ⇒
Процесс 2				↑ ⇒
Процесс 2		↑ ⇒	↗	
Процесс 1 (→) операции, ↑ исполнитель	↑ ⇒			

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 расстаются). На более низкие уровни делегируется ответственность за принятие финансовых решений: о заключении сделки-договора, об оплате, о закупке, о скидках и отпуске в *кредит* и т.п. Это позволяет упростить связи между подразделениями и снизить количество уровней вертикального прохождения документов, т.е. является необходимым условием реализации классической схемы *реинжиниринга*. Таким образом, процессная ориентация ведет к перестройке организационной структуры, делает организационную

структуру компании более "плоской", что иллюстрирует тесную *связь* между "вертикальным" описанием организации (как структуры распределения ответственности, полномочий и взаимоотношений) и ее "горизонтальным" описанием, как системы процессов.

Основные элементы процессного подхода

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

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

Каждый *бизнес-процесс* имеет свои границы и роли.

В *процессном подходе* используются следующие ключевые роли:

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

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

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

Координатор процесса – работник, отвечающий за согласованную работу всех частей бизнеса и обеспечивающий связь с другими бизнес-процессами. *Координатор* должен обладать административными способностями и пониманием стратегических целей предприятия.

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

Одним из основных элементов *процессного подхода* является команда.

Существует несколько типов процессных команд:

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

Виртуальная команда – создается для разработки нового продукта или услуги.

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

Важной задачей *процессного подхода* является формирование процессных команд. Подготовка и формирование команды включает:

- учебные курсы;
- практический тренинг по освоению методов, методик и др.;

- психологическое тестирование;
- тестирование рабочих навыков.

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

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

Бизнес-функции связаны с показателями деятельности предприятия, образующими *дерево показателей*. На основании 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.

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.

3. 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.

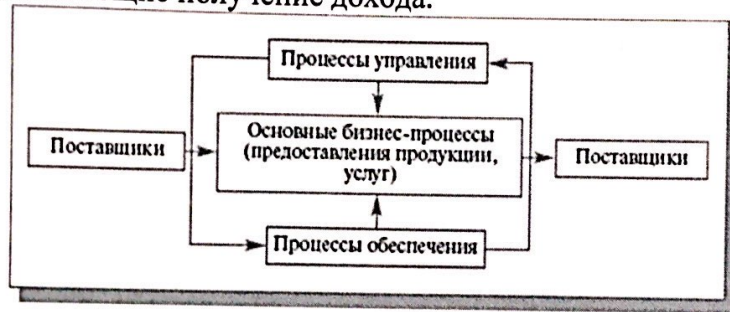


Рис. 5.2. Упрощенная модель деятельности компании

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);