

разработки информационных систем”.

Ростов-на-Дону

2010 г.

Основные термины.

Информационное обеспечение (ИО) - представляет собой совокупность проектных решений по объемам, размещению, форма организации информации, циркулирующей в ИС.

Лингвистическое обеспечение (ЛО) - объединяет совокупность языковых средств для формализации естественного языка, построения и сочетания информационных единиц в ходе общения пользователей со средствами вычислительной техники.

Техническое обеспечение (ТО) - представляет собой комплекс технических средств (технические средства сбора, регистрации, передачи, обработки, отображения, тиражирования информации, оргтехника и др.), обеспечивающих работу ИТ.

Программное обеспечение (ПО) - включает совокупность программ, реализующих функции и задачи ИС и обеспечивающих устойчивую работу комплексов технических средств.

Математическое обеспечение (МО) - совокупность математических методов, моделей и алгоритмов обработки информации, используемых при решении функциональных задач и в процессе автоматизации проектировочных работ.

Организационное обеспечение (ОО) - представляет собой комплекс документов, составленный в процессе проектирования ИС, утвержденный и положенный в основу эксплуатации.

Правовое обеспечение (ПрО) - представляет собой совокупность правовых норм, регламентирующих правоотношения при создании и внедрении ИС и ИТ.

Эргономическое обеспечение (ЭО) - как совокупность методов и средств, используемых на разных этапах разработки и функционирования ИС и ИТ, предназначено для создания оптимальных условий высококачественной, высокоэффективной и безошибочной деятельности человека в ИТ, для ее быстрейшего освоения.

Конечный продукт работы любого менеджера - это решения и действия.

Принятое им решение ведет либо к преуспеванию предприятия, либо к неудачам. **Принятие решения** - это всегда выбор определенного направления деятельности из нескольких возможных. Так как процесс управления любой организацией в экономике реализуется исключительно посредством формирования и реализации управленческих решений, поэтому остановимся на типах решений, которые имеют различные характеристики и требуют различных источников данных.

Оперативные решения - периодические: одна и та же задача возникает периодически. В результате процесс принятия решения становится относительно рутинным и почти беспроблемным. Параметры (характеристики) хозяйственных процессов, используемые в ходе принятия решения, определены, их оценка известна с высокой точностью, а взаимосвязь параметров с принимаемым решением понятна. Принятие оперативных решений ведет к вполне ожидаемым и прогнозируемым результатам. Оперативные решения являются краткосрочными.

Тактические решения обычно принимаются управленцами среднего уровня, ответственными за обеспечение средствами для достижения целей и намерений, поставленных ЛПР верхнего звена. Тактические решения не так рутинны и структурированы, как оперативные решения. Все главные параметры объекта управления, входящие в состав тактических решений, неизвестны; оценки характеристик, определенные как важные, могут быть неизвестны, а взаимосвязь между характеристиками и решениями может быть не ясна.

Стратегические решения принимаются на основе целей компании, определенных в его уставе и уточненных высшим руководством предприятия. Эти цели определяют основу, на которой должно базироваться долгосрочное планирование, а также определение критических факторов деятельности предприятия. Эти решения обеспечивают базу для принятия тактических и оперативных решений.

Стратегический план

После вышеприведенных шагов первый план, который необходимо сформировать, - это стратегический план, являющийся сводом инициатив (хотя еще не конкретных проектов), которые должна выполнить организация для продвижения по направлению к видению. Он также должен содержать числовые результаты, которых необходимо достичь за определенный период.

Центральное планирование

В этом случае имеется центральный отдел, координирующий ИС-стратегию и бизнес-стратегию, а начальник ИС-отдела должен быть частью управленческого аппарата, занятого принятием решений.

Операционный план ИС

После определения инициатив они должны быть представлены в виде проектов с конкретными результатами, приоритетами и т.д., т.е. в виде операционного плана.

Жизненный цикл информационных систем

Под жизненным циклом системы обычно понимается непрерывный процесс, который начинается с момента принятия решения о необходимости создания системы и заканчивается в момент ее полного изъятия из эксплуатации.

Введение

Создание ИС и ИТ представляет собой сложный процесс проектирования. Целью проектирования являются подготовка проектных документов и внедрение человекомашиной системы управления организацией. В процессе проектирования выявляются наиболее существенные характеристики экономического объекта, изучаются его внешние и внутренние информационные потоки, создаются математические и физические аналоги исследуемой системы и ее элементов, устанавливаются условия взаимодействия человека и технических средств управления.

Для того чтобы иметь хорошую информационную систему необходимо планировать ее создание. Именно поэтому вопрос планирования информационных систем рассматривается здесь отдельно. Процесс планирования должен начинаться с оценки текущей ситуации, определения миссии информационной системы, интенсивности использования информации, пользователей, оценки среды организации, места на рынке, ее сильных и слабых сторон, выработки стратегии, которая должна лечь в основу бизнес-плана по созданию информационной системы.

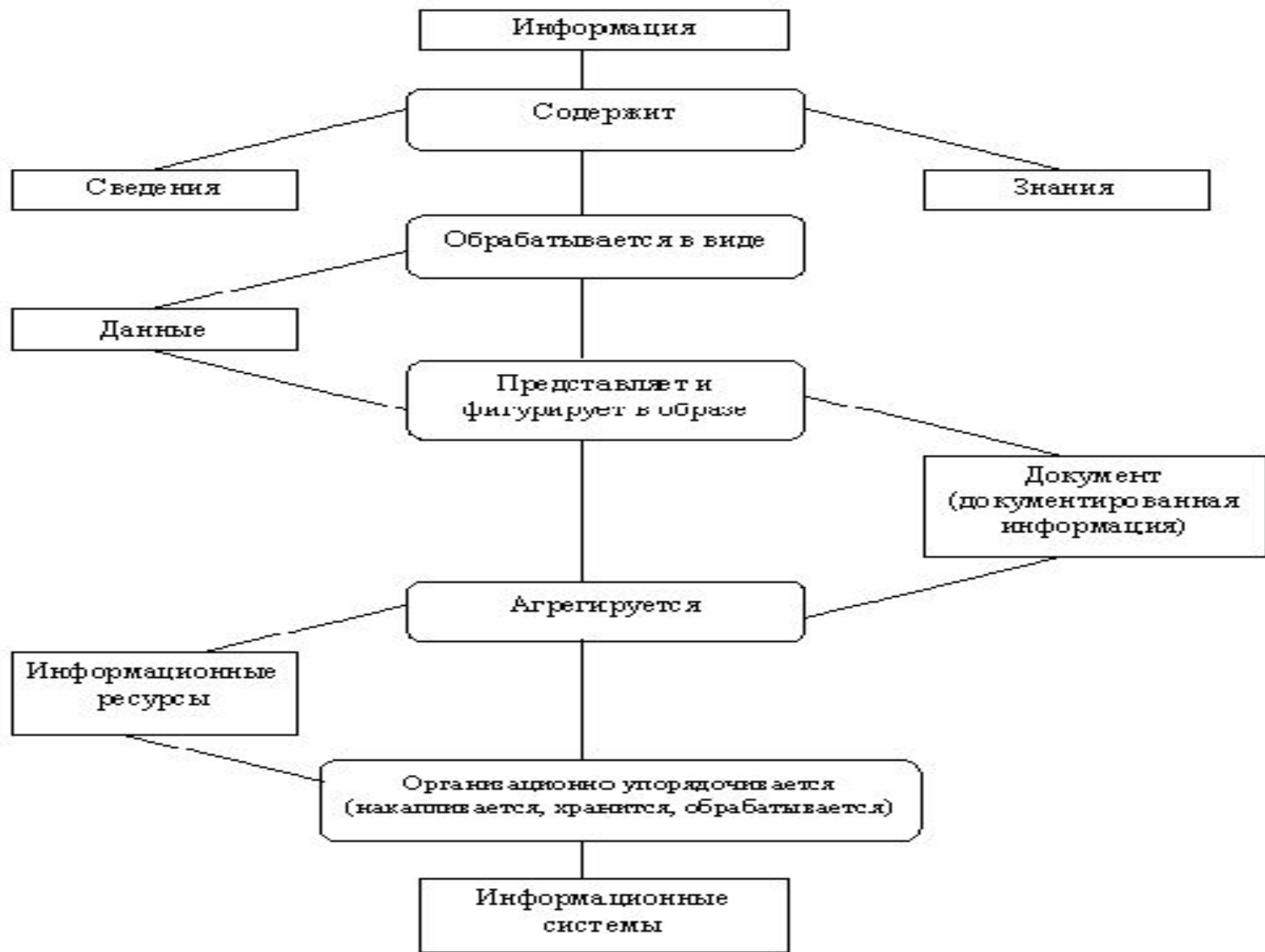


Рис. 1.2. Базовые понятия информационных систем.

История создания идеи ИС.

В настоящее время история развития систем, предназначенных для хранения и обработки информации с использованием ЭВМ, насчитывает уже более полувека. Еще относительно недавно в ходу были перфораторы в качестве устройств ввода данных, листинги в виде рулонов бумаги длиной порою до нескольких метров - в качестве носителя результатов машинной обработки, недельные, либо месячные временные интервалы - в качестве нормативных сроков обработки информации.

В последнее десятилетие ушедшего века ситуация претерпела качественные изменения. Если попытаться сформулировать "портрет" современной информационной системы масштаба предприятия в виде десятка тезисов, то мы увидим, что она имеет:

- в основе - методологию управления, направленную на достижение стратегических целей высшего менеджмента предприятия, выраженной в информационной системе в виде системы управляющих воздействий, регламентирующей деятельность пользователей. Возможность доступа к данным для множества пользователей, объединенных в локальную сеть предприятия, а зачастую - и для пользователей, удаленных от центрального офиса на сотни и тысячи километров. Наличие средств коммуникации и элементов корпоративного решения задач коллективом пользователей;

- развитый, дружелюбный графический интерфейс конечного пользователя;

- режимы обработки оперативной информации, близкие к режиму реального времени. Средства аутентификации и разграничения доступа, позволяющие дозировать информацию в соответствии с должностными обязанностями пользователя; высокий уровень защищенности от несанкционированного доступа. Один или более серверов баз данных, суммарный объем которых измеряется в гига- или терабайтах; возможность обработки тысяч и миллионов записей при составлении отчетности, инвариантность (в определенных пределах) к аппаратным и операционным средам функционирования серверных и клиентских приложений. Использование стандартизованных языков и протоколов для представления и манипулирования данными.

Классификация ИС.

1. Классификация ИС по масштабу.

Информационные системы могут быть классифицированы по различным признакам.

Однопользовательские ИС, как это ясно из названия, предназначены для использования на одном рабочем месте. В настоящее время на мировом и отечественном рынке представлено множество решений, предназначенных для автоматизации деятельности отдельно взятого пользователя. Как правило, это - решения, ориентированные на специалиста в той или иной области, будь то составление спецификаций для сборки изделий из комплектующих, планирование ремонтов оборудования, учет расходов и доходов частного предпринимателя оптовой торговли, либо составление расписаний занятий в деканате. В настоящее время альтернативу таким узкоспециализированным системам составили табличные процессоры, не имеющие проблемной специализации, в первую очередь - MS Excel. Системы этого класса трудно отнести к классу ИС, но зачастую они позволяют непрограммирующему специалисту создать и, что очень важно, самостоятельно развивать собственные решения, заменяющие, а местами и перекрывающие функционал однопользовательских систем образца 90-х годов.

В основе большинства однопользовательских систем лежит стандарт X-Base (Clipper, FoxPro, dBase). Широко используются также решения на базе систем Paradox, Clarion, MS Access. Каждая из перечисленных конкурирующих систем обладает собственной высокоуровневой инструментальной средой, позволяющей спроектировать базу данных, логику обработки, пользовательский интерфейс, отчеты с помощью "помощников"-построителей. На рубеже тысячелетий появились также и однопользовательские решения на базе промышленных реляционных СУБД. В этом случае ПО сервера устанавливается непосредственно на рабочую станцию пользователя. Примером может служить Personal Oracle. Данные решения предъявляют значительные требования к ресурсам рабочей станции, однако несут в себе многие преимущества промышленных СУБД.

Групповые системы предназначены для автоматизации деятельности в рабочей группе (отделе, кластере, группе проекта и т.д.). В отличие от однопользовательских ИС, групповые системы, как правило, представляют специализированные клиентские решения (их часто называют автоматизированными рабочими местами, АРМ) для различных участников группы. Например, для оптовой фирмы, ИС может представлять набор таких АРМ, как "Менеджер по продажам", "Кладовщик", "Снабженец", "Директор". Для учебного планирования - "Преподаватель", "Работник бюро планирования", "Работник учебного отдела", "Специалист по планированию на кафедре", "Работник деканата".

Групповое использование решений на базе табличных процессоров возможно, но имеет существенные ограничения, связанные с разграничением доступа, регламентацией и синхронизацией вносимых изменений. По сути единственный режим их использования, обеспечивающий корректность данных - "файловый сервер, один автор, N читателей".

При создании групповых ИС в целом используются те же средства и инструментальные среды, что и при создании однопользовательских ИС. Следует, однако, отметить, что для использования в группе при выборе между системами с файловым и реляционным сервером следует отдавать предпочтение реляционному серверу, причем целесообразно использование выделенного сервера. Это может быть, например, сервер Oracle, DB2, MS SQL, Sybase, Informix.

Корпоративные ИС (КИС) предназначены для автоматизации деятельности предприятия. В англоязычной литературе понятие "КИС" неразрывно связано с понятием "ERP" (Enterprise Resource Planning). В основе ERP-систем лежит международный стандарт управления предприятием MRP-II (Manufacture Resource Planning), обеспечивающий возможность учета, анализа и планирования основных ресурсов - финансов, человеческих, материальных.

Соответственно, корпоративные ERP-системы - набор интегрированных приложений, которые комплексно, в едином информационном пространстве поддерживают все основные аспекты управленческой деятельности предприятий: планирование ресурсов (финансовых, человеческих, материальных) для производства товаров (услуг), оперативное управление выполнением планов (включая снабжение, сбыт, ведение договоров), все виды учета и анализ результатов хозяйственной деятельности.

Среди требований, предъявляемым к современным КИС:

- централизация данных в единой базе (в основе - всегда промышленная СУБД),
- близкий к реальному времени режим работы,
- сохранение общей модели управления для предприятий разных отраслей,
- поддержка территориально-распределенных структур,
- работа на широком круге аппаратно-программных платформ и СУБД.

Примеры ERP-систем - SAP R3, "Галактика", MS Navision Axapta.

2. Классификация информационных систем по архитектуре.

Архитектура "Файл-сервер". Исторически первая архитектура информационных систем. Как исполняемые модули, так и данные размещаются в отдельных файлах операционной системы. Доступ к данным осуществляется путем указания пути (path) и использования файловых операций (открыть, считать, записать). Для хранения данных используется выделенный сервер (отдельный компьютер), который и является файловым сервером. Исполняемые модули хранятся либо на рабочих станциях, либо на файловом сервере. В последнем случае упрощается процедура их администрирования, но при этом возрастают требования к надежности сети.

Архитектура "Клиент-сервер". Клиент-сервер - это не только архитектура, это - новая парадигма, пришедшая на смену устаревшим концепциям. Суть ее заключается в том, что клиент (исполняемый модуль) запрашивает те или иные сервисы в соответствии с определенным протоколом обмена данными. При этом, в отличие от ситуации с файловым сервером, нет необходимости в использовании прямых путей операционной системы: клиент их "не знает", ему "известны" лишь имя источника данных и другие специальные сведения, используемые для авторизации клиента на сервере. Сервер, который физически может находиться на том же компьютере, а может - на другом конце земного шара, обрабатывает запрос клиента и, произведя соответствующие манипуляции с данными, передает клиенту запрашиваемую порцию данных.

В рамках направления "клиент-сервер" существуют два основных "диалекта": "тонкий" и "толстый" клиент.

В системах на основе тонкого клиента используется мощный сервер баз данных, это - высокопроизводительный компьютер и библиотека так называемых хранимых процедур, позволяющих производить вычисления, реализующие основную логику обработки данных, непосредственно на сервере. Клиентское приложение, соответственно, предъявляет невысокие требования к аппаратному обеспечению рабочей станции. Основное достоинство таких систем - относительная дешевизна клиентских станций.

Системы с толстым клиентом, напротив, реализуют основную логику обработки на клиенте, а сервер представляет собой в чистом виде сервер баз данных, обеспечивающий исполнение только стандартизованных запросов на манипуляцию с данными (как правило - чтение, запись, модификацию данных в таблицах реляционной базы данных). В системах такого класса требования к рабочей станции выше, а к серверу - ниже. Достоинство архитектуры - переносимость серверной компоненты на серверы различных производителей: все промышленные серверы баз данных реляционного типа поддерживают работу со стандартизованным языком манипулирования данными SQL, но внутренний встроенный язык обработки данных, необходимый для реализации логики обработки на сервере у каждого из серверов свой.

Трехслойная архитектура. Базируется на дальнейшей специализации компонент архитектуры: клиент занимается только организацией интерфейса с пользователем, сервер баз данных - только стандартизированной обработкой данных. Для реализации логики обработки данных архитектура предусматривает отдельный слой - слой бизнес-логики. Этот слой может представлять собой либо выделенный сервер (сервер приложений), либо размещаться на клиенте в качестве динамической библиотеки. Данная архитектура позволила соединить достоинства тонкого и толстого клиентов: хорошая переносимость соединяется в ней с невысокими требованиями к клиенту.

С развитием интернет-технологий появилась разновидность трехслойной архитектуры на основании использования web-технологий. В этой разновидности роль сервера приложений играет web-сервер, а в качестве клиента используется стандартный web-браузер. Достоинства - в пониженных требованиях к клиенту и в легкой встраиваемости данной архитектуры в мировые информационные сети. Основной недостаток - известные ограничения, накладываемые на интерфейс пользователя web-браузерами.

С некоторой степенью приближения все ИС можно разделить на 2 класса: информационно-поисковые и управляющие.

Конечные пользователи информационно-поисковых систем (ИПС), как правило, имеют доступ к хранимым данным только "по чтению" и используют данные системы для поиска ответов на те или иные вопросы. Доступ по модификации данных имеет администратор системы, в функции которого входит обеспечение актуальности информации, устранение ошибок.

Классические примеры ИПС - системы поиска в библиотеках, на транспорте (справки о наличии билетов). На современном этапе развития информационных технологий классические ИПС постепенно вытесняются поисковыми серверами Интернет - общего назначения и специализированными.

Альтернатива ИПС - управляющие системы автоматизируют (полностью или частично) деятельность, связанную с принятием решений. Действия конечных пользователей таких систем приводят к модификации информации, что, конечно, не исключает возможности просто получать информацию, как в ИПС.

Примеры управляющих систем - системы бухгалтерского учета, системы планирования производственных ресурсов и т.п.

Разработка ИС.

Проблемы на стадии разработки

Взгляд руководства организации и ее персонала, не говоря уже о разработчиках, на создание информационной системы различен. Здесь мы попытаемся, не вдаваясь в технические проблемы, построить модель процесса создания информационной системы для менеджеров и показать, в чем их задачи. Существует две различных стадии осуществления проекта построения информационных систем и технологий - разработка и внедрение и эксплуатация.

Стадия разработки и внедрения обычно всегда осуществляется полностью. Ей не мешает ни слабое развитие технологии, ни отсутствие компетенции персонала или пользователей, ни отсутствие хороших консультантов.

Если на этой стадии возникают проблемы, то они связаны со следующими тремя основными причинами:

недостаток поддержки основного персонала, особенно когда надо уделить достаточно времени и энергии на критических стадиях;
слишком амбициозные планы вместо пошагового, мудрого подхода;
неудача при получении достаточного количества советов от практиков с настоящим опытом использования похожих систем в похожем бизнесе.

В рамках группового обсуждения в работе R.Hanage (Ma-naging Information for Profit and Growth), были получены следующие ответы на вопрос о том, какие проблемы возникали с проектами информационных технологий? консультанты по информационным технологиям не понимали наших мыслей;

трудно найти нужный совет;

сложно подобрать прикладное обеспечение для деловых процессов;

неподходящее время для установки системы;

плохая техническая и программная поддержка.

Как правило, проект информационных технологий всегда занимает больше времени, чем предполагалось. Необходимо быть готовым к тому, чтобы вложить больше ресурсов, чем требуется, для того чтобы быть уверенным, что он не остановится;

участвующие в осуществлении проекта люди всегда думают, что их работа сделана, когда аппаратура и программы работают успешно. Фактически проект завершен только тогда, когда достигнуты ожидаемые преимущества для бизнеса. Если проект связан с деловыми целями по улучшению отдельных сторон функционирования организации, и все это знают, он более успешен.

Имеется четыре стадии создания информационной системы.

1 Эскиз проекта.

Подробное описание целей и задач проекта, ожидаемой прибыли, временных ресурсов, любых ограничений, доступных ресурсов и т.д. Стоит также определить "менеджера проекта", который отвечает за его осуществление, и ответственного за проект в высшем руководстве, который будет главной персоной в бизнесе и будет поддерживать менеджера проекта, когда это необходимо и в самом конце выполнения проекта.

2 Оценка проекта.

Это самая главная часть проекта. В ней принимаются все важные решения - что будут делать системы, как они будут работать, какая аппаратура и прикладные программы будут использоваться и как они будут обслуживаться. Важнее всего, что здесь анализируются возможные затраты и прибыли от различных действий и производится конечный выбор. В качестве основного правила следует использовать принцип, согласно которому система должна быть настолько простой, насколько возможно. Грандиозные проекты системы могут вылиться в невероятные затраты. Изменения, которые вносятся позже, являются более дорогими. Сначала готовят список требований к системе - детальный перечень того, что система будет делать для бизнеса и как ею управлять.

Изучаются потребности постоянных пользователей (и других заинтересованных лиц), так как только они действительно знают, что им нужно и как это вписать в существующую деятельность.

Список включает в себя данные которые предназначены для ввода, основные результаты и отчеты, количество пользователей, размеры информации, связи с другими существующими системами и т.д. и должен быть достаточно подробным для того, чтобы можно было послать запрос поставщикам аппаратуры и программного обеспечения.

На этой стадии мы не должны, просто компьютеризировать существующие способы работы. Проект информационных технологий - это хорошая возможность еще раз подумать, как лучше сделать информационную систему.

Следующая стадия состоит в том, чтобы посмотреть на требования к аппаратуре и программному обеспечению. Проконсультироваться с потенциальными поставщиками, посмотреть другие деловые решения и посоветоваться со знающими консультантами. Некоторые трудные решения должны подвергнуться тщательной оценке. Следует ответить, например, на такие вопросы: использовать ли уже готовый пакет прикладных программ либо заказать новое программное обеспечение. Ответы будут зависеть от степени риска, к которой Вы готовы, и от отличий Вашего бизнеса от других типичных фирм.

Анализ затрат и прибыли - это финальный шаг перед окончательным решением.

Затраты на прикладные программы и аппаратуру относительно невелики, особенно если Вы используете стандартный пакет. Большими затратами являются время, на установку системы и время на поддержку ее работы.

3 Построение и тестирование.

Одним из самых недооцененных шагов в установке любой системы является ввод всех данных в систему до ее запуска.

Персонал должен убедиться, что с системой легко работать. Ничто не убивает энтузиазм по отношению к новой системе быстрее, чем серия технических проблем.

4 Управление проектом и оценка риска.

Если только проект не совсем тривиален, то необходимо существование менеджера проекта, у которого есть достаточно времени, чтобы работать с проектом и иметь дело с массой проблем, которые могут возникнуть. Проект не завершен до тех пор, пока менеджер проекта не сможет продемонстрировать, что система работает надежно и приносит прибыль. Важная часть его роли состоит в том, чтобы постоянно осознавать риск проекта. Риски должны обсуждаться открыто, несмотря на соблазн спрятать голову в песок и надеяться, что все обойдется. Риск можно спланировать: приняв альтернативные решения, приготовившись к крайним действиям и т.д.

Примером послужит выбор программного обеспечения, при котором различные решения могут быть рискованны в различной степени. Более нет места для подробного обсуждения, но использование следующего перечня вопросов может помочь выделить некоторые пункты.



Рис.1.3. Программное обеспечение информационной системы

Принципы создания.

Согласно приведенным подходам формируются основные принципы создания ИС и ИТ управления:

системность и логичность построения обеспечивающих и функциональных элементов ИС;

широкое применение экономико-математических методов и стандартных программ прогнозно-статистического характера. Задачи управления производственной, финансовой деятельностью организации в большинстве своем ставятся как аналитические, оптимизационные или как задачи планирования.

предполагает декомпозицию системы на ряд комплексов (модулей) задач, каждый из которых моделирует определенную сферу управленческой деятельности.

использование новых методов и включение вновь созданных программных модулей в систему автоматизации управленческих работ. Проектирование ИС должно изначально базироваться на модульных принципах, а компьютерная реализация - допускать расширение за счет совершенствования структуры программного обеспечения.

это принцип адаптации всех элементов и системы в целом. Он должен полностью пронизывать идеологию построения ИС управления - от анализа задач, технико-экономических показателей и их группировок в модули до формулирования целей.

Общий показатель надежности ИС концентрирует в себе ряд важных характеристик:

- частоту возникновения сбоев в техническом обеспечении;
- степень адекватности математических моделей;
- верификационную чистоту программ;
- относительный уровень достоверности информации;
- интегрированный показатель надежности эргономического обеспечения ИС.

Адаптационные свойства системы отражают ее способность приспособливаться к изменениям окружающего внешнего фона внутренней управленческой и производственной среды организации. Важная задача заказчика - сформулировать на этапе проектирования границы допущения отклонений в значениях управляющих и выходных параметров, имеющих принципиальное значение для функционирования всей системы.

В общем виде постановка задачи состоит из четырех принципиально важных компонентов:

- организационно-экономической схемы и ее описания;
- свода применяемых математических моделей;
- описания вычислительных алгоритмов;
- концепции построения информационной модели системы.

Постановка и дальнейшая компьютерная реализация задач требуют усвоения основных понятий, касающихся теоретических основ, информационных технологий. К ним относятся:

- свойства, особенности и структура экономической информации;
- условно-постоянная информация, ее роль и назначение;
- носители информации, макет машинного носителя;
- средства формализованного описания информации;
- алгоритм, его свойства и формы представления;
- назначение и способы контроля входной и результатной информации;
- состав и назначение устройств компьютера;
- состав программных средств, назначение операционных систем, пакетов прикладных программ (ППП), интегрированных пакетов программ типа АРМ менеджера, АРМ руководителя, АРМ (финансиста, АРМ бухгалтера и т.п.)

Жизненные циклы ИС.

Современные сети разрабатываются на основе стандартов, что позволяет обеспечить, во-первых, их высокую эффективность и, во-вторых, возможность их взаимодействия между собой.

Вообще говоря, все стандарты на информационные системы (как и на любые системы вообще) можно разбить на следующие два основных класса:

- Функциональные стандарты, определяющие порядок функционирования системы в интересах достижения цели, поставленной перед нею ее создателями.

- Стандарты жизненного цикла, определяющие то, как создается, развертывается, применяется и ликвидируется система.

Модели, определяемые стандартами этих двух классов, конечно же взаимосвязаны, однако решают совершенно разные задачи и характеризуются принципиально различными подходами к их построению.

Поясним это на примере. Наиболее полной функциональной моделью системы является сама система, однако "биография" самой системы ни в коем случае не может рассматриваться в качестве модели ее жизненного цикла. Куда ближе к модели жизненного цикла информационной системы является описание жизни живого существа, начиная с момента зачатия.

Таким образом, жизненный цикл информационной системы охватывает все стадии и этапы ее создания, сопровождения и развития:

- предпроектный анализ (включая формирование функциональной и информационной моделей объекта, для которого предназначена информационная система);
- проектирование системы (включая разработку технического задания, эскизного и технического проектов);
- разработку системы (в том числе программирование и тестирование прикладных программ на основании проектных спецификаций подсистем, выделенных на стадии проектирования);
- интеграцию и сборку системы, проведение ее испытаний;
- эксплуатацию системы и ее сопровождение;
развитие системы.

Немного специфики

Продолжительность жизненного цикла современных информационных систем составляет около 10 лет, что значительно превышает сроки морального и физического старения технических и системных программных средств, используемых при построении системы. Поэтому в течение жизненного цикла системы проводится модернизация ее технико-программной базы. При этом прикладное программное обеспечение системы должно быть сохранено и перенесено на обновляемые аппаратно-программные платформы.

Эти проблемы привели к тому, что подавляющее большинство проектов информационных систем внедряется с нарушениями качества, сроков или сметы.

Почти треть проектов информационных систем прекращают свое существование, оставшись незавершенными. По данным, публикуемым Standish Group, в 1996 году 84% проектов информационных систем не были завершены в установленные сроки, в 1998 году сократилась до 74%, однако и в 2000-м общий объем "хронической незавершенки" не опустился ниже 50%.

Главной причиной такого положения является то, что уровень технологии анализа и проектирования систем, методов и средств управления проектами не соответствует сложности создаваемых систем, которая постоянно возрастает в связи с усложнением и быстрыми изменениями бизнеса.

Из мировой практики известно, что затраты на сопровождение прикладного программного обеспечения информационных систем составляют не менее 70% его совокупной стоимости на протяжении жизненного цикла. Поэтому крайне важно еще на проектной стадии предусмотреть необходимые методы и средства сопровождения прикладного программного обеспечения, включая методы конфигурационного управления.

В России создание и испытания автоматизированных систем, к которым относятся и информационные системы, регламентированы рядом ГОСТов, прежде всего серии 34. Однако отдельные положения этих ГОСТов уже устарели, а ряд этапов жизненного цикла информационных систем предоставлены недостаточно полно. Поэтому более целесообразно рассматривать в качестве определяющего документа международный стандарт ISO/IEC 12207. Данный стандарт определяет структуру жизненного цикла, содержащую процессы, которые должны быть выполнены во время создания программного обеспечения информационной системы.

Рисунок 1. Каскадная модель жизненного цикла



Эти процессы подразделяются на три группы: основные (приобретение, поставка, разработка, эксплуатация и сопровождение), вспомогательные (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, оценка, аудит и решение проблем) и организационные (управление проектами, создание инфраструктуры проекта, определение, оценка и улучшение самого жизненного цикла, обучение).

Каскадная и спиральная модели

Однако стандарт ISO/IEC 12207 не предлагает конкретной модели жизненного цикла и методов разработки, его рекомендации являются общими для любых моделей жизненного цикла. Под моделью обычно понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении жизненного цикла. Из существующих в настоящее время моделей наиболее распространены две: каскадная и спиральная. Они принципиально различаются самим подходом к информационной системе и ее программному обеспечению. Суть различий в том, что в каскадной модели информационная система является однородной и ее программное обеспечение определяется как единое (с ней) целое. Данный подход характерен для более ранних информационных систем (каскадный метод применяется с 1970 года), а также для систем, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования. При выполнении этих условий каскадный метод позволяет достичь хороших результатов.

Суть каскадного метода (рис. 1) заключается в разбиении всей разработки на этапы, причем переход от предыдущего этапа к последующему осуществляется только после полного завершения работ предыдущего этапа. Соответственно на каждом этапе формируется законченный набор проектной документации, достаточной для того, чтобы разработка могла быть продолжена другой группой разработчиков. Другим положительным моментом каскадной модели является возможность планирования сроков завершения работ и затрат на их выполнение. Однако у каскадной модели есть один существенный недостаток - очень сложно уложить реальный процесс создания программного обеспечения в такую жесткую схему и поэтому постоянно возникает необходимость возврата к предыдущим этапам с целью уточнения и пересмотра ранее принятых решений. Результатом такого конфликта стало появление модели с промежуточным контролем (рис. 2), которую представляют или как самостоятельную модель, или как вариант каскадной модели. Эта модель характеризуется межэтапными корректировками, удлиняющими период разработки изделия, но повышающими надежность.

Однако и каскадная модель, и модель с промежуточным контролем обладают серьезным недостатком - запаздыванием с получением результатов. Данное обстоятельство объясняется тем, что согласование результатов возможно только после завершения каждого этапа работ.

На время же проведения каждого этапа требования жестко задаются в виде технического задания. Так что существует опасность, что из-за неточного изложения требований или их изменения за длительное время создания программного обеспечения конечный продукт окажется не востребуемым. Для преодоления этого недостатка и была создана спиральная модель, ориентированная на активную работу с пользователями и представляющая разрабатываемую информационную систему как постоянно корректируемую во время разработки. В спиральной модели (рис. 3) основной упор делается на этапы анализа и проектирования, на которых реализуемость технических решений проверяется путем создания прототипов. Спиральная модель позволяет начинать работу над следующим этапом, не дожидаясь завершения предыдущего. Спиральная модель имеет целью как можно раньше ознакомить пользователей с работоспособным продуктом, корректируя при необходимости требования к разрабатываемому продукту и каждый "виток" спирали означает создание фрагмента или версии. Основная проблема спирального цикла - определение момента перехода на следующий этап, и возможным ее решением является принудительное ограничение по времени для каждого из этапов жизненного цикла. Наиболее полно достоинства такой модели проявляются при обслуживании программных средств.

Сравнивая эти модели, можно сказать, что каскадная модель более универсальна, т. е. она применима к производству разных изделий, будь то отбойный молоток или графический редактор. Для разных изделий просто будут изменяться количество и название этапов модели. Спиральная же модель более ориентирована именно на информационные системы, особенно на программные продукты, поэтому при разработке информационных систем и их программного обеспечения она предпочтительнее каскадной.

Следующим шагом в вопросе поддержания жизненного цикла информационной системы, как, впрочем, и любого другого изделия, является его автоматизация. Однако автоматизация различных процессов, связанных с разработкой, производством и эксплуатацией как изделий промышленности, так и информационных систем наиболее эффективна в том случае, когда она охватывает все этапы жизненного цикла изделия. При этом необходимо преодоление следующих проблем: наличие множества различных систем, ориентированных на решение конкретных задач, относящихся к разным этапам жизненного цикла, приводит к трудностям обмена данными между смежными системами; участие в поддержке жизненного цикла изделия нескольких предприятий требует эффективного обмена информацией об изделии между партнерами; сложность изделия, наличие множества его модификаций, заимствование, стандартизация, унификация, требуют поддержки многоуровневых многовариантных сборочных моделей. Эти проблемы могут быть преодолены путем реализации концепции CALS.

CALS

Аббревиатура CALS расшифровывается как Continuous Acquisition and Life cycle Support - непрерывная информационная поддержка жизненного цикла продукта. Встречается также другой перевод, менее схожий с исходным названием, но более близкий по смыслу: обеспечение неразрывной связи между производством и прочими этапами жизненного цикла изделия. Данная технология, разработанная в 80-х годах в Министерстве обороны США, распространилась по всему миру и охватила практически все сферы мировой экономики. Она предназначена для повышения эффективности и качества бизнес-процессов, выполняемых на протяжении всего жизненного цикла продукта, за счет применения безбумажных технологий. Началом создания системы CALS-технологий явилась разработка системы стандартов описания процессов на всех этапах жизненного цикла продукции.

В международных стандартах серии ISO 9004 (управление качеством продукции) введено понятие "жизненный цикл изделия". Данное понятие включает в себя следующие этапы жизненного цикла изделия: маркетинг, поиск и изучение рынка; проектирование и/или разработка технических требований к создаваемой продукции; материально-техническое снабжение; подготовка и разработка технологических процессов; производство; контроль, проведение испытаний и обследований; упаковка и хранение; реализация и/или распределение продукции; монтаж, эксплуатация; техническая помощь в обслуживании; утилизация после завершения использования продукции.

Рисунок 2. Модель жизненного цикла с промежуточным контролем



Для развития методологии CALS в США были созданы Управляющая промышленная группа по вопросам CALS (ISG) и ее исполнительный консультативный комитет. В настоящий момент в мире действует более 25 национальных организаций (комитетов или советов по развитию CALS), в том числе в США, Японии, Канаде, Великобритании, Германии, Швеции, Норвегии, Австралии и других странах, а также в НАТО.

Основные усилия этих и подобных организаций были направлены на создание разного уровня нормативной документации. За последние несколько лет разработаны следующие документы: ISO 10303 (Industrial automation systems and integration -- Product data representation and exchange), ISO 13584 (Part Library), Def Stan 00-60 (Integrated Logistic Support), MIL-STD-2549 (Configuration Management. Data Interface), MIL-HDBK-61 (Configuration Management. Guidance), AECMA Specification 2000M (International Specification for Materiel Management Integrated Data Processing for Military Equipment), AECMA Specification 1000D (International Specification for Technical Data Publications, Utilising a Common Source Data Base) и т. д.

Стандарты CALS

Стандарты, разработанные ISO для CALS-технологий, можно разбить на три группы: представление информации о продукте, представление текстовой и графической информации и общего назначения. К первой группе относятся: ISO/IEC 10303 Standard for the Exchange of Product Model Data (STEP) и ISO 13584 Industrial Automation -- Parts Library.

Во вторую группу входят: ISO 8879 Information Processing -- Text and Office System - Standard Generalised Markup Language (SGML); ISO/IEC 10179 Document Style Semantics and Specification Language (DSSSL); ISO/IEC IS 10744 Information Technology -- Hypermedia/Time Based Document Structuring Language (HyTime); ISO/IEC 8632 Information Processing Systems -- Computer Graphics - Metafile; ISO/IEC 10918 Coding of Digital Continuous Tone Still Picture Images (JPEG); ISO 11172 MPEG2 Motion Picture Experts Group (MPEG); Coding of Motion Pictures and associated Audio for Digital Storage Media и ISO/IECS 13522 Information Technology -- Coding of Multimedia and Hypermedia Information (MHEG).

Рисунок 3. Спиральная модель жизненного цикла



Эффективность реализации CALS

Основная задача, решаемая путем применения CALS-технологий, -- экономия времени и средств при одновременном повышении качества. Так, в США применение CALS-технологий сопровождается следующими типовыми показателями.

В процессах проектирования и инженерных расчетах: сокращение времени проектирования на 50%; снижение затрат на изучение выполнимости проектов на 15--40%.

В процессах организации поставок: уменьшение количества ошибок при передаче данных на 98%; сокращение времени поиска и извлечения данных на 40%; сокращение времени планирования на 70%; сокращение стоимости информации на 15--60%.

В производственных процессах: сокращение производственных затрат на 15-60%; повышение показателей качества на 80%;

в процессах эксплуатационной поддержки изделий: сокращение времени на изменения технической документации на 30%; сокращение времени планирования поддержки на 70%; снижение стоимости технической документации на 10--50%.

Отдельные вопросы построения информационных систем и технологий

Здесь мы рассмотрим некоторые технологии создания информационных систем, наиболее часто предлагаемые разработчиками. Знакомство с такими технологиями облегчит процесс понимания заказчиком предложений разработчика.

Автоматизированные системы проектирования

Усложнение информационных систем и расширение областей их применения, повышение требований к ним привели к тому, что даже большие, коллективы разработчиков не в состоянии за приемлемое время, в, условиях ограничений по ресурсам и с заданным качеством разработать информационную систему. В результате развития средств и методов создания информационных систем оформилось направление, связанное с автоматизацией проектирования информационной системы и информационной технологии. Это путь использования готовых решений, обеспечения заданного качества и ускорения работ при создании информационной системы и информационной технологии. (См.: Ойхман Е.Г., Попов Э.В. Реинжиниринг бизнеса: реинжиниринг организации и информационные технологии. - М.: Финансы и статистика, 1997.)

Этап тестирования

После завершения разработки отдельного модуля системы выполняют автономный тест, который преследует две основные цели: обнаружение отказов модуля (жестких сбоев); соответствие модуля спецификации (наличие всех необходимых функций, отсутствие лишних функций).

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

Далее группа модулей тестируется на надежность работы, то есть проходят, во-первых, тесты имитации отказов системы, а во-вторых, тесты наработки на отказ. Первая группа тестов показывает, насколько хорошо система восстанавливается после сбоев программного обеспечения, отказов аппаратного обеспечения. Вторая группа тестов определяет степень устойчивости системы при штатной работе и позволяет оценить время безотказной работы системы. В комплект тестов устойчивости должны входить тесты, имитирующие пиковую нагрузку на систему.

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

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

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

Широкое распространение в этой области получил подход CASE (Computer Aided Software/System Engineering - CASE-технология). CASE-технология совокупность методов анализа, проектирования, разработки и сопровождения информационной системы, поддерживаемых комплексом взаимосвязанных средств автоматизации. Это инструментарий для системных аналитиков, разработчиков и программистов, позволяющий автоматизировать процесс исследования, проектирования и разработки информационной системы (анализ предметной области, спецификации проектов, выпуск документации, тестирование реализаций проектов, планирование и контроль разработок, моделирование и т.п.). Это индустриализация технологии создания информационной системы и информационной технологии, позволяющая отделить и автоматизировать процесс проектирования информационной системы от последующих этапов разработки. Использование CASE-технологий существенно изменяет технологию работ на этапах анализа, проектирования и модернизации информационной системы. В CASE-технологиях применяются специальные методы анализа, проектирования и моделирования. CASE-технологии могут использоваться при создании информационной системы любых типов.

Общий показатель надежности ИС концентрирует в себе ряд важных характеристик:

- частоту возникновения сбоев в техническом обеспечении;
- степень адекватности математических моделей;
- верификационную чистоту программ;
- относительный уровень достоверности информации;
- интегрированный показатель надежности эргономического обеспечения ИС.

Адаптационные свойства системы отражают ее способность приспособливаться к изменениям окружающего внешнего фона внутренней управленческой и производственной среды организации. Важная задача заказчика - сформулировать на этапе проектирования границы допущения отклонений в значениях управляющих и выходных параметров, имеющих принципиальное значение для функционирования всей системы.

Заключение

Сегодня информационные технологии оказывают влияние не только на обработку данных, но и на способ выполнения работы людьми, на продукцию, характер конкуренции. Информация во многих организациях становится ключевым ресурсом, а информационная обработка – делом стратегической важности.

Большинство организаций не сможет успешно конкурировать, пока не предложит своим клиентам такой уровень обслуживания, который возможен лишь при помощи систем, основанных на высоких технологиях.

Информационная система управления – это система, обеспечивающая уполномоченный персонал данными или информацией, имеющими отношение к организации. Информационная система управления, в общем случае, состоит из четырех подсистем: системы обработки транзакций, системы управленческих отчетов, офисной информационной системы и системы поддержки принятия решений, включая информационную систему руководителя, экспертную систему и искусственный интеллект.

Информационные системы используются организациями в разных целях. Они повышают производительность труда, помогая выполнять работу лучше, быстрее и дешевле, функциональную эффективность, помогая принимать наилучшие решения. Информационные системы повышают качество услуг, предоставляемых заказчикам и клиентам, помогают создавать и улучшать продукцию. Они позволяют закрепить клиентов и отдалить конкурентов, сменить основу конкуренции путем изменения таких составляющих, как цена, расходы, качество.

Список литературы.

1. А.Н. Адаменко, А.М. Кучуков. Логическое программирование и Visual Prolog СПб.: БХВ--Петербург, 2003.
2. Братко И. Алгоритмы искусственного интеллекта на языке PROLOG. М.: «Вильямс», 2004.
3. Джексон П. Введение в экспертные системы.-Москва, С. Петербург, Киев: Изд. дом "Вильямс", 2002
4. Дж. Доорс, А. Рейблейн, С. Вадера. Пролог - язык программирования будущего. М.: Финансы и статистика, 1990
5. Дюбуа Д., Прад А. Теория возможностей. Приложения к представлению знаний. -М.: Радио и связь, 1995
6. Корнеев В.В., Гарев А.Ф., Васюшин СВ., Райх В.В. Базы данных. Интеллектуальная обработка информации. - М.: Изд-во "Нолидж",
7. Мендельсон Э. Введение в математическую логику. М., 1976
8. Нечаев В.В., Панченко В.М., Свиридов А.П. Исследование операций и теория систем. Основы статистической динамики знаний. Учебное пособие.-М.: МИРЭА, 2000
9. Новиков П. С. Элементы математической логики. М., 1959
10. Попов Э.В. Экспертные системы реального времени. В: Открытые системы, N2 (10), 1995
11. Хоггер К. Введение в логическое программирование М.: Мир, 1988
12. Черч А. Введение в математическую логику, т. I. М. 1960

13. Интернет источники.