Системный анализ

Лекция 2

С сего начать работу над проектом?

- Выяснить
  - какие задачи должна решать программная система,
  - какими свойствами она должна обладать
- Ответы на эти вопросы должен дать ***этап системного анализа*** фазы разработки ПС

«Проблема заказчика»

- Заказчик
  - формулирует задачу на своем профессиональном языке;
  - имеет достаточно расплывчатое представление о функциях будущей программной системы;
  - не способен оценить возможности реализации тех или иных своих пожеланий

Первый шаг

- Достижение взаимопонимания между заказчиком и разработчиком
- Разработчик должен понять специфику деятельности заказчика и связанные с ней проблемы

АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Анализ предметной области

- **Определение**

  Деятельность, направленная на выявление реальных потребностей заказчика, а также на выяснения смысла высказанных требований, называется ***анализом предметной области*** или ***бизнес-моделированием,*** если речь идет о потребностях коммерческой организации

- Анализ предметной области – это первый шаг ***этапа системного анализа***, с которого начинается разработка программной системы

Что в результате

- В результате
  - разработчики должны научиться понимать язык, на котором говорят заказчики;
  - выявить цели их деятельности;
  - определить набор решаемых ими задач;
  - определить набор сущностей, с которыми приходится иметь дело при решении этих задач

Модель предметной области

- Анализом предметной области занимаются ***системные аналитики*** или ***бизнес-аналитики***
- Результаты этого анализа представляются в виде ***модели предметной области*** – набора графических схем и текстовых документов

Далее приводятся основные понятия, теории моделирования

МОДЕЛИРОВАНИЕ. ОСНОВНЫЕ ПОНЯТИЯ

Системы и модели

- Под ***системой*** подразумевается совокупность взаимодействующих компонентов и взаимосвязей между ними
- ***Моделью*** M некоторой системы S называется информационный объект, который может быть использован для получения ответов на некоторый круг вопросов относительно S

Характеристики модели

- К ним относятся:
  - цель моделирования,
  - объект моделирования,
  - точка зрения модели,
  - средство моделирования
- Модель должна быть адекватна целям и объекту моделирования

Цель моделирования

- Получение ответов на некоторую совокупность вопросов является **целью моделирования**
- Цель моделирования формулируется на самом раннем этапе разработки модели

Объект моделирования

- **Объектом моделирования** является сама система. При этом необходимо точно определить **границы системы**, чтобы избежать включения в модель посторонних объектов

Точка зрения модели

- Круг вопросов, на которые модель должна дать ответ определяется ***точкой зрения*** данной модели
- Точку зрения лучше всего представлять как место позицию человека или объекта, на которую надо встать, чтобы увидеть систему в действии

Итак

- Объект определяет, что включить в модель, а что исключить из нее
- Точка зрения диктует автору модели выбор нужной информации об объекте и форму ее подачи
- Цель становится критерием окончания моделирования

Результат моделирования

- **Результатом моделирования** является набор взаимоувязанных описаний, начиная с описания самого верхнего уровня системы и кончая подробным описанием деталей или операций

Виды моделей

- Формальные модели, используемые на этапе анализа предметной области можно разделить на две группы:
  - модели, зависящие от подхода к разработке (структурного или объектно-ориентированного)
  - модели, не зависящие от подхода к разработке

Структурный подход

- Сущность структурного подхода заключается в декомпозиции программной системы по ***функциональному принципу***
- При структурном подходе первичным считают проектирование обрабатывающих компонентов - ***процедур***
- Проектирование же структур данных выполняют параллельно

Объектный подход

- В основе объектного подхода к разработке программного обеспечения лежит ***объектная декомпозиция***, предполагающая объединение процедур и структур данных

  процедуры + структуры данных = классы

Объектный подход

- При этом разрабатываемое ПО представляется в виде совокупности взаимодействующих ***объектов***, совместно обеспечивающих выполнение требуемых функций

Классификация моделей

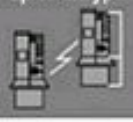
МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Схема Захмана

- При проведении анализа предметной области бывает полезно воспользоваться схемой, предложенной в 1987 году Джоном Захманом (John A. Zachman)
- Схема Захмана определяет цели моделирования, применимые к широкому кругу предметных областей

Основная идея

- Деятельность любой организации можно описать, используя ответы на 6 простых вопросов:
  - «ЧТО делается», или объекты/данные;
  - «КАК делается», или функции/процессы;
  - «ГДЕ делается», —инфраструктура;
  - «КТО делает» — люди;
  - «КОГДА делается» — графики работ;
  - «ЗАЧЕМ делается» — стимулы и стратегии

	Мотивация	Люди	Данные	Функции	Место	Время
Контекст	Цели и стратегия бизнеса 	Важные для бизнеса организации 	Вещи, значимые для бизнеса 	Основные бизнес-процессы 	География бизнеса 	События и периоды, важные для бизнеса 
Модель бизнеса	Бизнес-план, частные цели и стратегии 	Модели потоков работ 	Семантические модели  Бизнес-сущности и их связи	Модели бизнес-процессов 	Система логистики 	Базовый график работ 
Системная модель	Модель бизнес-правил 	Архитектура пользовательского интерфейса 	Концептуальная модель данных 	Архитектура приложений 	Архитектура распределенной системы 	Структура обработки событий 
Технологическая модель	Модель правил обработки событий 	Архитектура представления 	Физическая модель данных 	Архитектура программно-аппаратной системы 	Технологическая архитектура 	Структура циклов управления 
Детальное представление	Спецификации правил работы системы 	Спецификации ролей и прав доступа 	Спецификации форматов данных 	Код программных компонентов 	Спецификации архитектуры сети 	Спецификации обработки событий и прерываний 
Работающая организация	Стратегия и тактика	Структура организации	Данные	Выполняемые функции	Географическое расположение и сети	Планы

Структура матриц Захмана

- Шести вопросам соответствуют шесть столбцов матрицы Захмана
- Шесть строк соответствуют шести **уровням рассмотрения**
- Каждая ячейка матрицы задает свой тип описания (модели) свойств предприятия
- Конкретный вид этих моделей определяется выбором между структурным и объектно-ориентированным подходами

СТРУКТУРНОЕ МОДЕЛИРОВАНИЕ

Методология SADT

- Методология структурного моделирования *SADT (Structured Analysis And Design Technique)* появилась в 1970-х годах и предназначалась для анализа сложных систем различного профиля

Методология SADT

- В основных чертах эта методология сформулирована Дугласом Т.Россом (компания SofTech) в 1973 году
- Методология SADT применяется на ранних этапах процесса создания системы, часто еще до разработки технического задания (ТЗ)

Достоинства SADT

1. Может использоваться для проектирования сложных систем любого назначения
2. Позволяет отражать в модели такие системные характеристики, как управление, обратная связь и исполнители
3. Имеет развитые процедуры поддержки коллективной работы
4. Может быть использована на ранних этапах создания системы
5. Может сочетаться с другими структурными методами проектирования

Основные направления

- Существует два основных направления в SADT-моделировании:
  - *функциональные модели* выделяют события в системе,
  - *модели данных* выделяют объекты (данные) системы, связывающие функции между собой и с их окружением
- Стандартизованный вариант методологии создания функциональных моделей – ***IDEF0***

Наиболее удобной формой представления информации при *анализе предметной области* являются графические диаграммы различного рода

ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ МОДЕЛЕЙ

Проект ICAM

- Методология *IDEF0* появилась в рамках проекта ***ICAM*** *(Integrated Computer-Aided Manufacturing)*, имевшем целью разработку методов анализа процессов взаимодействия в производственных (промышленных) системах и инициированного Министерством обороны США

Цель проекта

- Основная цель – обеспечение возможности эффективного обмена информацией между всеми специалистами - участниками программы *ICAM* (отсюда название методологии *IDEF - Icam DEFinition*)

Методологии IDEF

- В рамках проекта ICAM планировалась разработка семейства методологий моделирования различных аспектов функционирования систем:
  - ***IDEF0*** – методология создания функциональной модели системы (основана на методе SADT Росса);

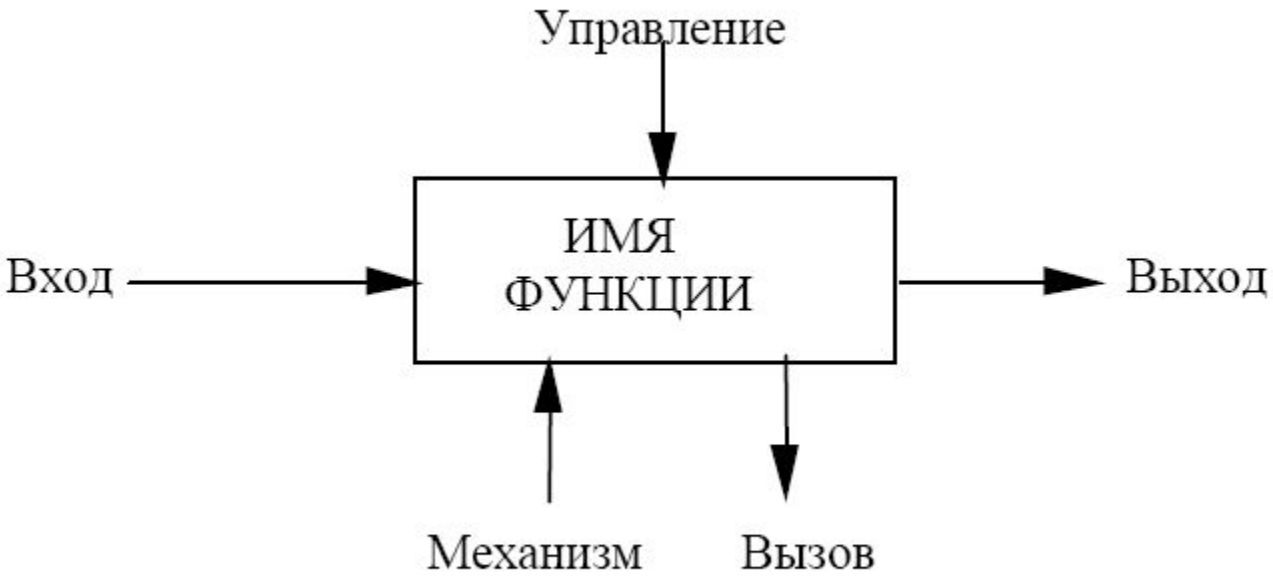
Методологии IDEF

- ***IDEF1*** – методология создания информационной модели системы (основана на реляционной теории Кодда и использовании ER-диаграмм);
- ***IDEF2*** – методология создания динамической модели системы;
- ***IDEF3*** – методология создания модели потоков работ (обычно используется вместе с диаграммами потоков данных DFD Data flow diagram)

Синтаксис IDEF0-моделей

- Основной формой представления IDEF0-модели является ***диаграмма***
- Каждая IDEF0-диаграмма содержит блоки (работы) и дуги (стрелки).
  - Блоки изображают функции моделируемой системы.
  - Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними.
- Функциональные блоки на диаграмме изображаются прямоугольниками, а дуги – стрелками

Блоки и стрелки



Основные правила

1. Каждая сторона функционального блока должна иметь стандартное отношение блок/стрелки:
   a) входные стрелки должны связываться с левой стороной блока;
   b) управляющие стрелки должны связываться с верхней стороной блока;
   c) выходные стрелки должны связываться с правой стороной блока;

Основные правила

d) стрелки механизма (кроме стрелок вызова) должны указывать вверх и подключаться к нижней стороне блока;

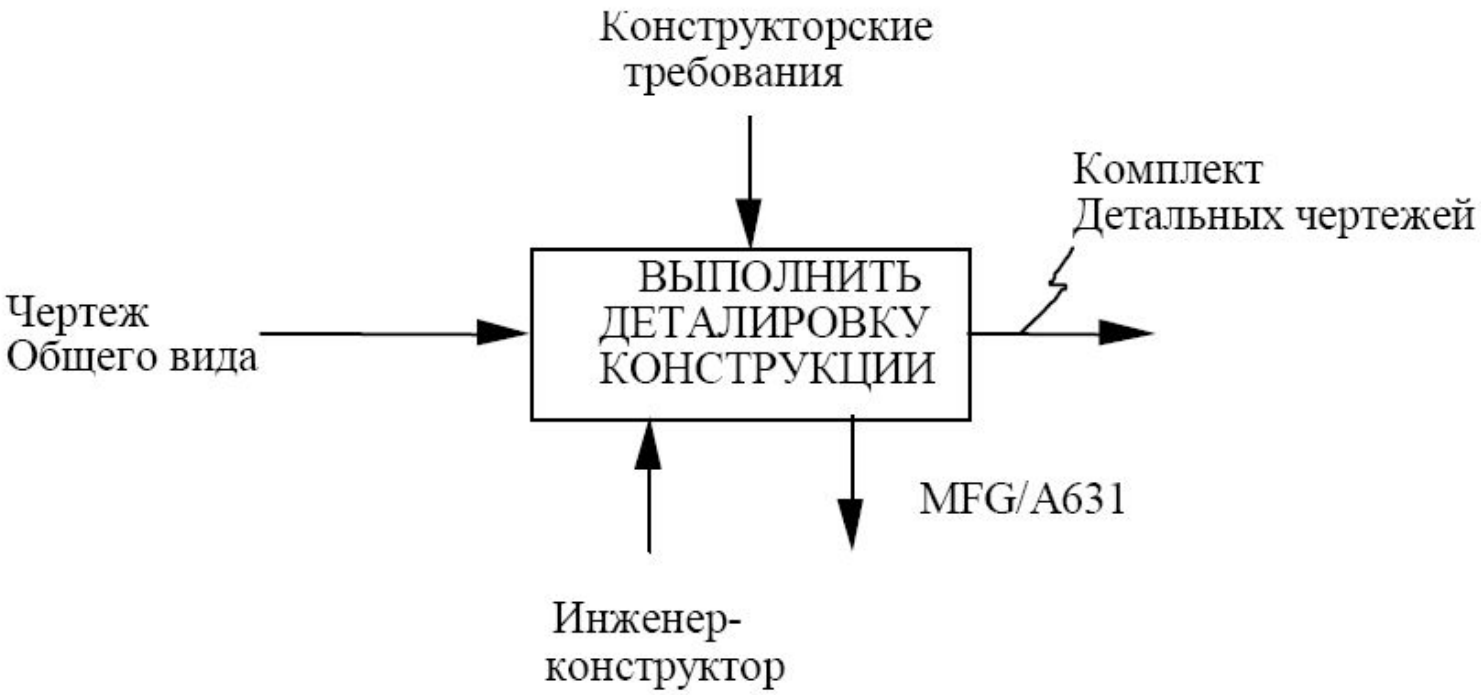
e) стрелки вызова механизма должны указывать вниз, подключаться к нижней стороне блока, и помечаться ссылкой на вызываемый блок

3. В метках стрелок не должны использоваться следующие термины: функция, вход, управление, выход, механизм, вызов

Основные правила

3. Сегменты стрелок, за исключением стрелок вызова, должны помечаться существительным или оборотом существительного
4. Чтобы связать стрелку с меткой, следует использовать "тильду" (~)

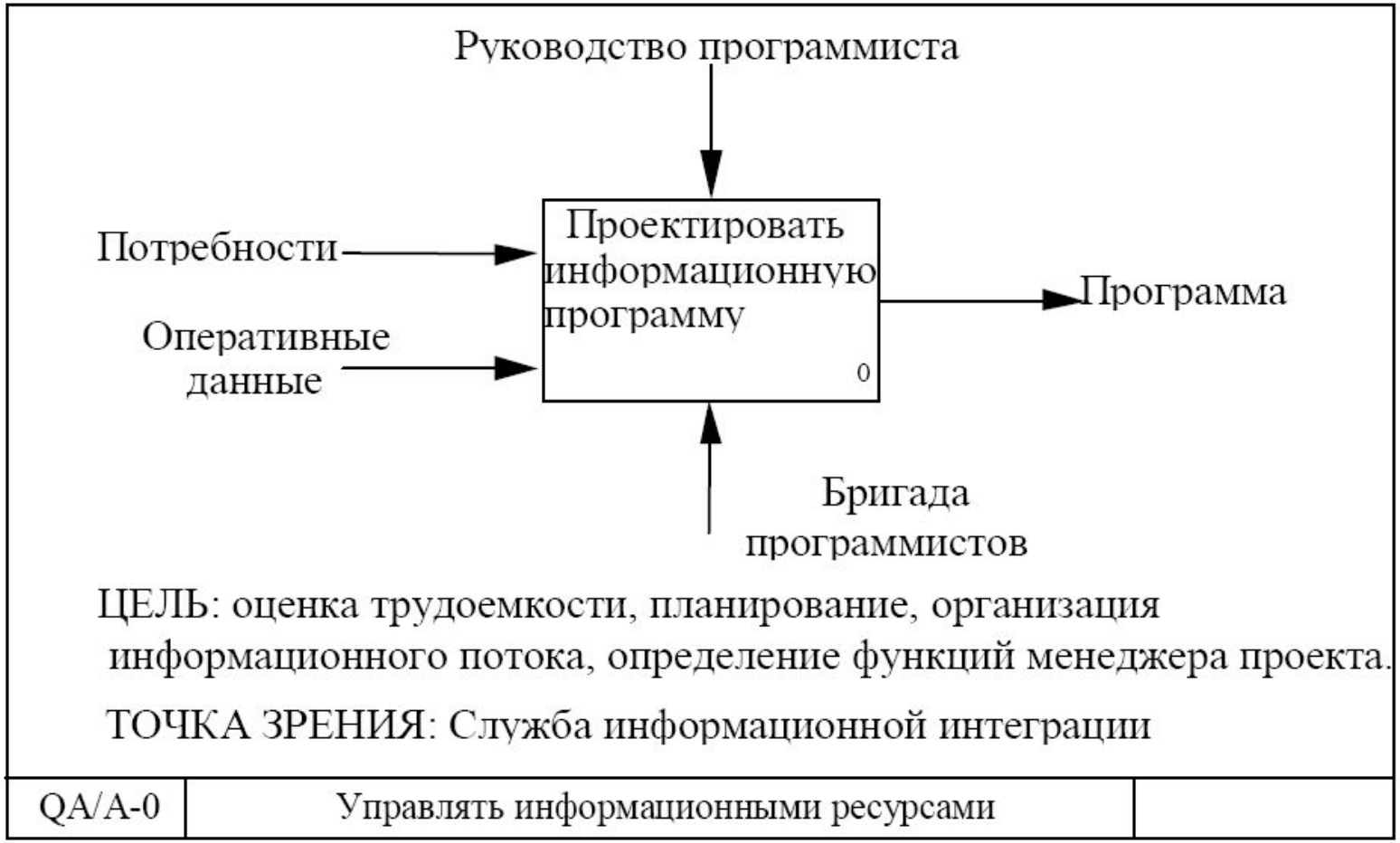
Пример блока и стрелок



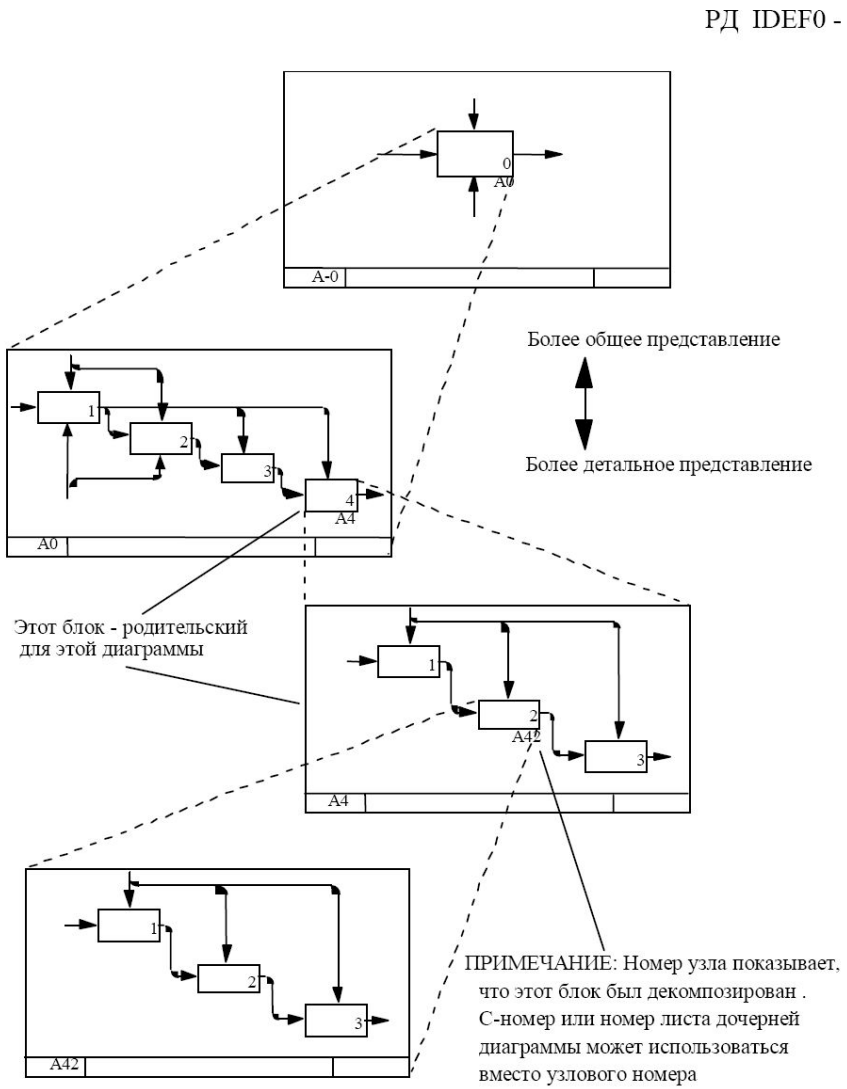
Принцип декомпозиции

- Функции моделируемой системы могут быть разбиты на составные части и представлены в виде более подробных диаграмм (**принцип декомпозиции**)
  - Диаграмма верхнего уровня называется ***контекстной*** и обеспечивает наиболее общее описание объекта моделирования
  - За этой диаграммой следует серия ***дочерних диаграмм***, дающих детальное представление об объекте

Контекстная диаграмма



Декомпозиция диаграмм



Состав IDEFo-модели

- IDEF0-модели состоят из трех типов документов:
  - графических диаграмм,
  - текста,
  - глоссария
- Эти документы имеют перекрестные ссылки друг на друга

Текстовое сопровождение

- **Графическая диаграмма** – главный компонент IDEF0-модели, содержащий блоки, стрелки, соединения блоков и стрелок и ассоциированные с ними отношения
- **Текст** используется для объяснений и уточнений характеристик, потоков, внутриблочных соединений и т.д.

Глоссарий

- **Глоссарий** предназначен для определения аббревиатур, ключевых слов и фраз, используемых в качестве имен и меток на диаграммах
- Глоссарий определяет понятия и термины, которые должны быть одинаково понимаемы всеми участниками разработки и пользователями модели

Семантика стрелок

- Стрелки на диаграмме IDEF0 , представляют данные или материальные объекты
- Входные и управляющие стрелки блока, соединяющие его с другими блоками или с внешней средой, описывают условия, которые необходимо выполнить для реализации функции, записанной в качестве имени блока

Отношения между блоками

- В методологии IDEF0 существует 6 типов отношений между блоками в пределах одной диаграммы:
  - доминирование;
  - управление;
  - выход - вход;
  - обратная связь по управлению;
  - обратная связь по входу;
  - выход – механизм

Отношение доминирования

- Определяется взаимным расположением блоков на диаграмме
- Предполагается, что блоки, расположенные на диаграмме выше и левее, «доминируют» над блоками, расположенными ниже и правее
- **Доминирование** понимается как влияние, которое один блок оказывает на другие блоки диаграммы

Отношения управления и выход-вход

- **Отношение управления** возникает тогда, когда выход одного блока служит управляющим воздействием на блок с меньшим доминированием
- **Отношение выход – вход** возникает при соединении выхода одного блока с входом другого блока с меньшим доминированием

Обратные связи

- **Обратная связь по управлению** возникает тогда, когда выход некоторого блока создает управляющее воздействие на блок с *большим* доминированием
- **Отношение обратной связи** по **входу** имеет место тогда, когда выход блока становиться входом другого блока с *большим* доминированием

Отношение «выход-механизм»

- Обратная связь по управлению и обратная связь по входу представляют итерацию (выход функции влияет на будущее выполнение других функций с большим доминированием)
- **Связи «выход – механизм»** отражают ситуацию, при которой выход одной функции становиться средством достижения цели для другой

Отношение «выход-механизм»

- Связи «выход – механизм» возникают при отображении в модели процедур пополнения и распределения ресурсов, создания или подготовки средств для выполнения функций системы

Дуги диаграмм IDEF0

- Дуги IDEF0, как правило, изображают наборы предметов, поэтому они могут разветвляться и соединяться вместе различным образом

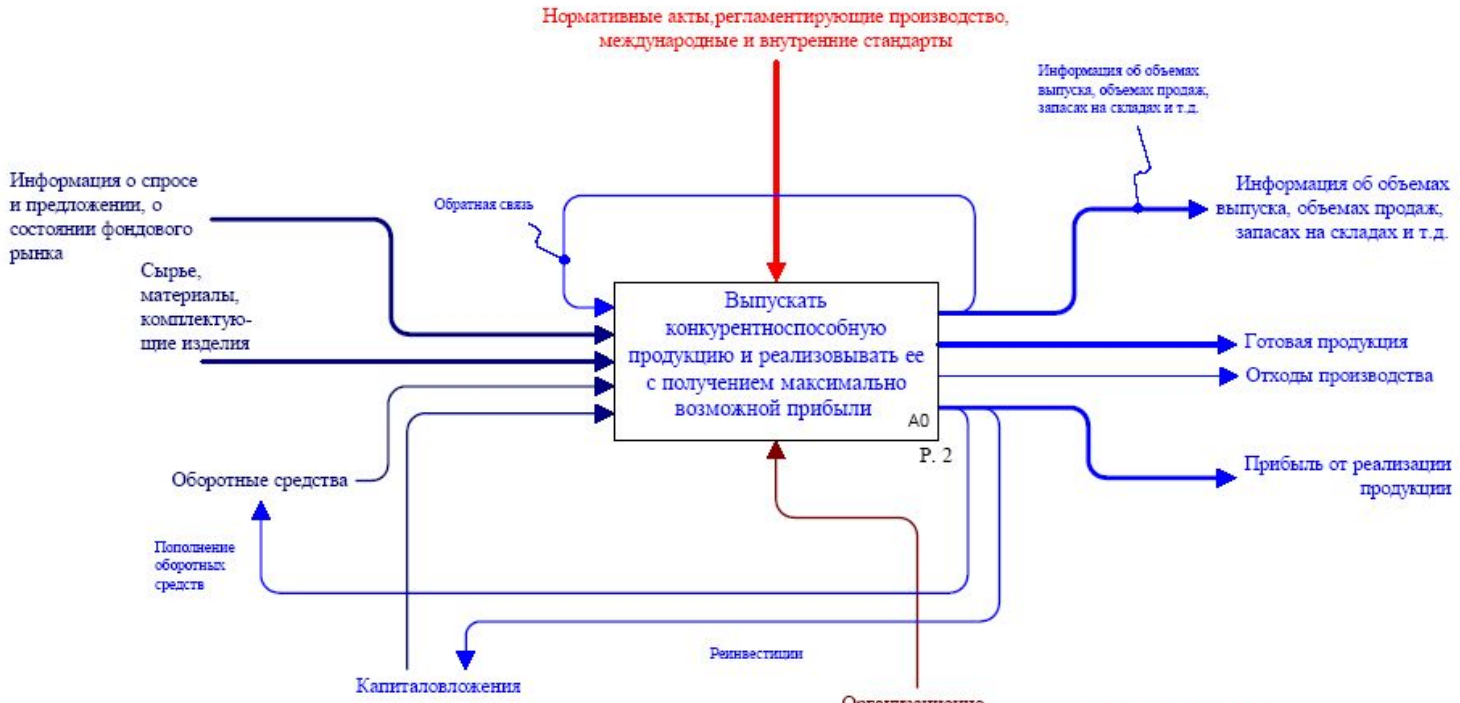
Разветвление дуг

- Разветвления дуги означают, что часть ее содержимого (или весь набор предметов) может появиться в каждом ответвлении дуги
- Дуга всегда помечается до разветвления, чтобы дать название всему набору
- Каждая ветвь дуги может быть помечена в соответствии со следующими правилами:
  - непомеченная ветвь содержит все предметы, указанные в метке перед разветвлением;
  - каждая метка ветви уточняет, что именно содержит эта ветвь.

Слияние дуг

- Слияние дуг указывает, что содержимое каждой ветви участвует в формировании после слияния объединенной дуги
- После слияния дуга всегда помечается для указания нового набора
- Каждая ветвь перед слиянием может помечаться в соответствии со следующими правилами:
  - непомеченные ветви содержат все предметы, указанные в общей метке после слияния;
  - каждая метка ветви уточняет, что именно содержит эта ветвь

Используется в:	Автор: Прикладная Логистика	Дата: 02/11/99	20/12/99	11/01/00	x	Рабочая версия	Читатель	Дата	Контекст:
РД "Методология функционального моделирования IDEF0"	Проект: Типовое Предприятие	Время: 03:34 PM	09:39 PM	03:04 PM		Проект			Top
	Замечания: 1 2 3 4 5 6 7 8 9 10	Версия: 1				Рекомендовано			
						Публикация			

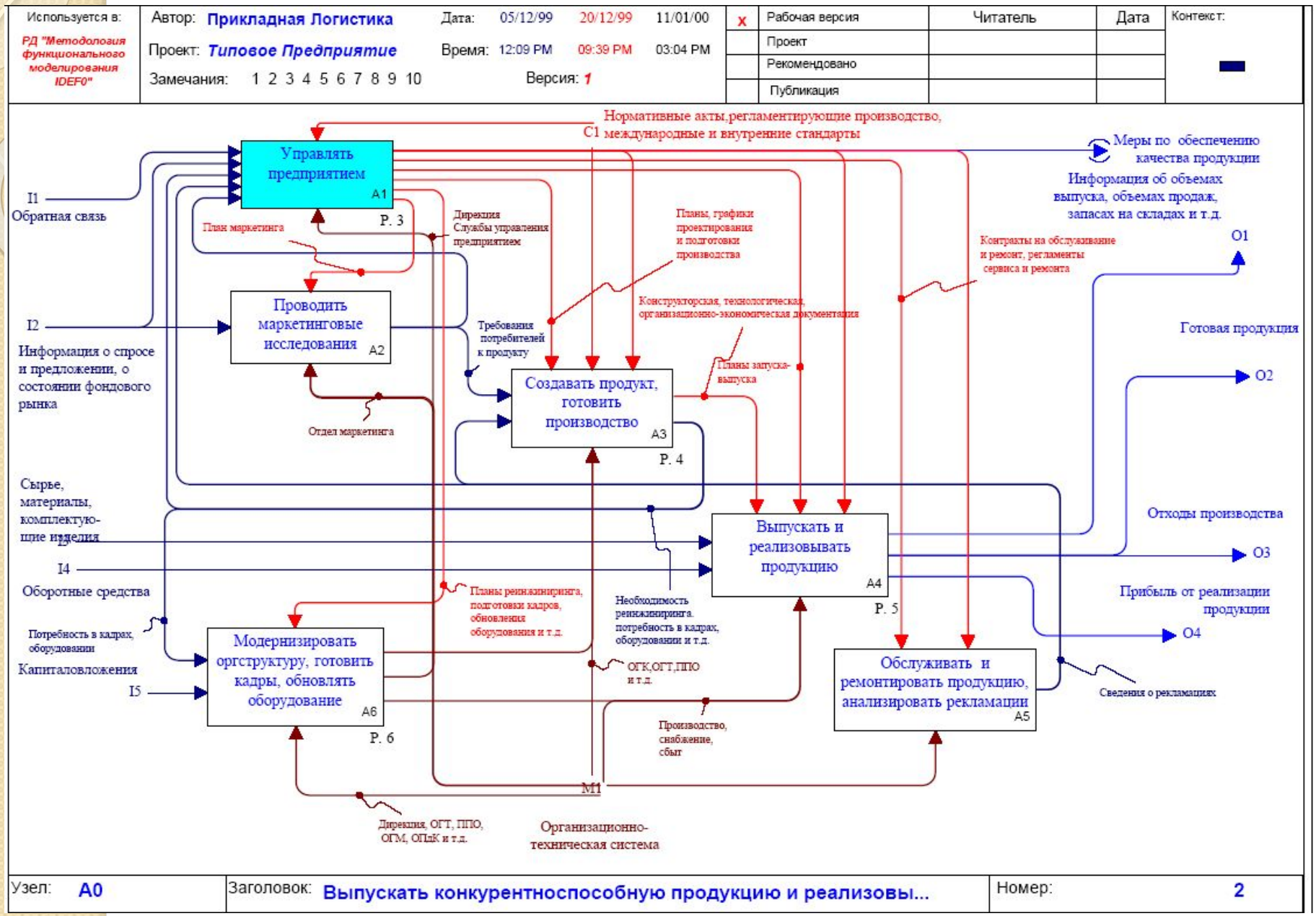


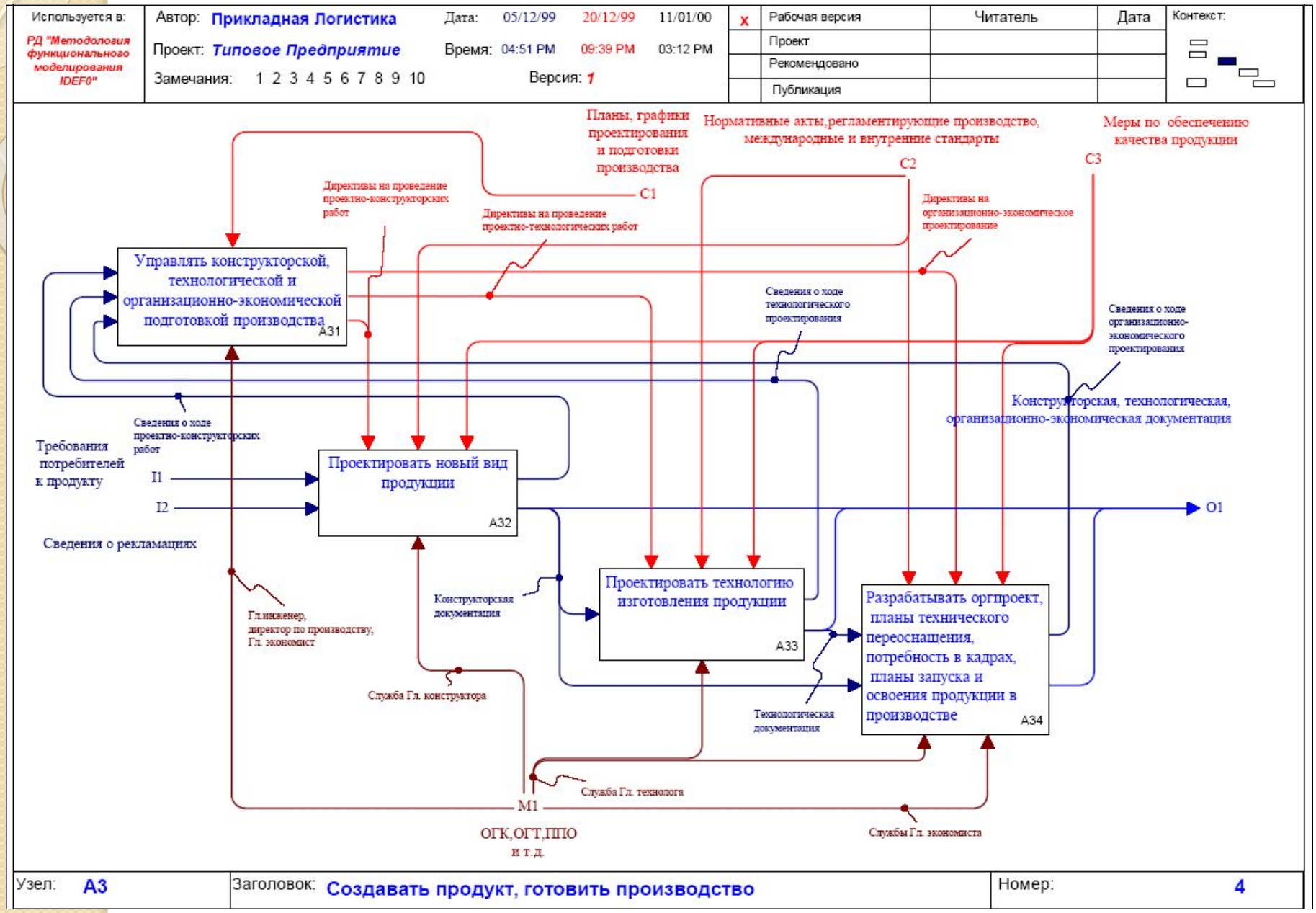
ЦЕЛЬ:

- показать возможность создания типизированной модели;
- выявить основные связи;
- показать связь системы обеспечения качества с остальными компонентами;
- показать структуру элементов управления;
- показать формирование механизмов в процессе деятельности.

Точка зрения: разработчика РД

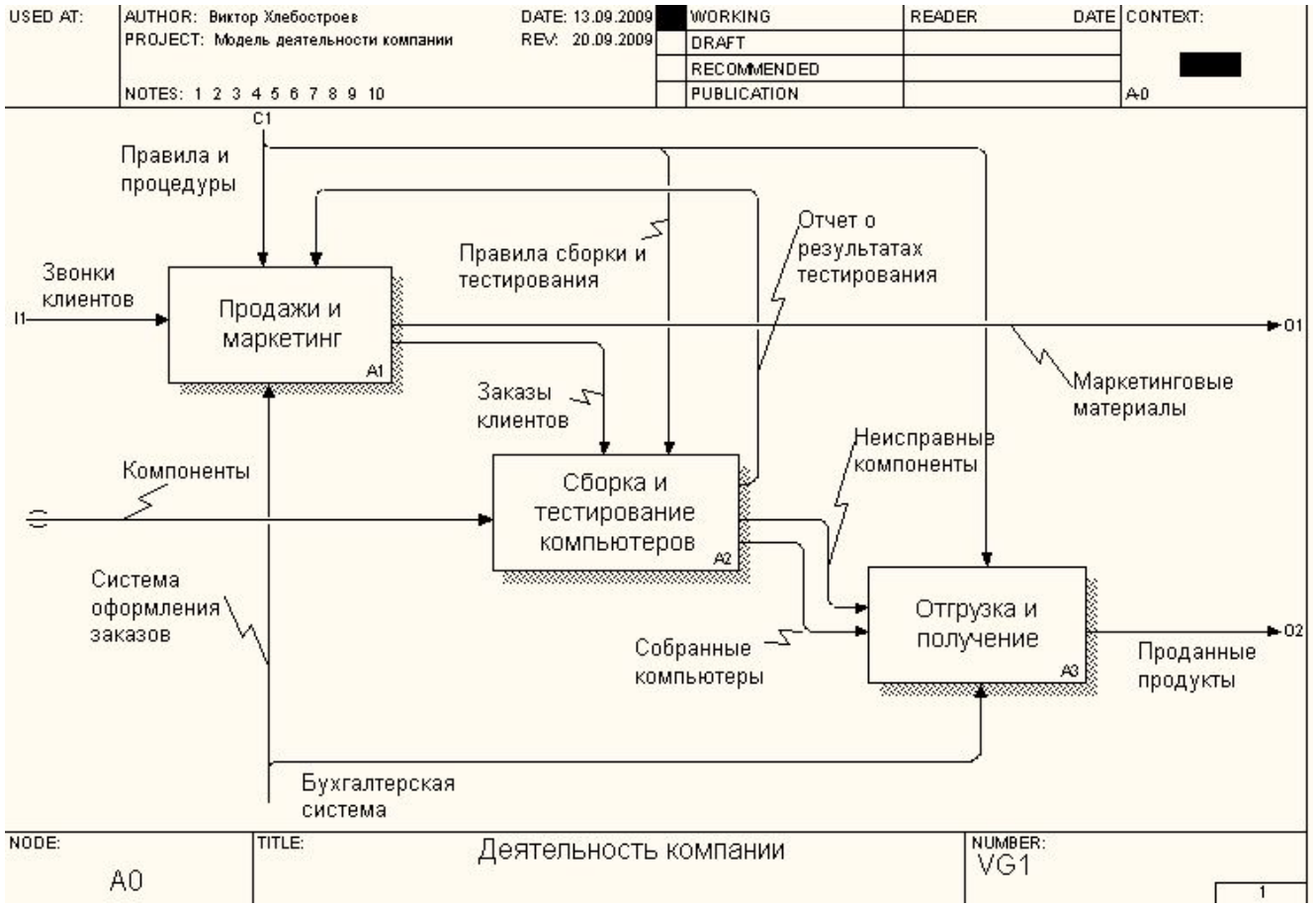
Узел: A-0	Заголовок: ПРИЛОЖЕНИЕ 2	Номер: 1


Используется в:
РД "Методология функционального моделирования IDEF0"
Автор: Прикладная Логистика
Дата: 05/12/99 20/12/99 11/01/00
Проект: Типовое Предприятие
Время: 12:09 PM 09:39 PM 03:04 PM
Замечания: 1 2 3 4 5 6 7 8 9 10
Версия: 1
x
Рабочая версия
Проект
Рекомендовано
Публикация
Читатель
Дата
Контекст:
Нормативные акты,регламентирующие производство,
C1 международные и внутренние стандарты
Управлять предприятием
A1
Меры по обеспечению качества продукции
Информация об объемах выпуска, объемах продаж, запасах на складах и т.д.
I1
Обратная связь
Р. 3
План маркетинга
Дирекция Службы управления предприятием
Планы, графики проектирования и подготовки производства
Контракты на обслуживание и ремонт, регламенты сервиса и ремонта
O1
Проводить маркетинговые исследования
A2
I2
Информация о спросе и предложении, о состоянии фондового рынка
Требования потребителей к продукту
Конструкторская, технологическая, организационно-экономическая документация
Готовая продукция
Планы запуска-выпуска
O2
Создавать продукт, готовить производство
A3
Р. 4
Отдел маркетинга
Сырье, материалы, комплектую-щие изделия
I3
Отходы производства
O3
Выпускать и реализовывать продукцию
A4
I4
Оборотные средства
Р. 5
Прибыль от реализации продукции
O4
Планы реинжиниринга, подготовки кадров, обновления оборудования и т.д.
Необходимость реинжиниринга. потребность в кадрах, оборудовании и т.д.
Потребность в кадрах, оборудовании
Капиталовложения
I5
Модернизировать оргструктуру, готовить кадры, обновлять оборудование
A6
Р. 6
ОГК,ОГТ,ППО и т.д.
Обслуживать и ремонтировать продукцию, анализировать рекламации
A5
Сведения о рекламациях
Производство, снабжение, сбыт
M1
Дирекция, ОГТ, ППО, ОГМ, ОПдК и т.д.
Организационно-техническая система
Узел: A0
Заголовок: Выпускать конкурентноспособную продукцию и реализовы...
Номер: 2

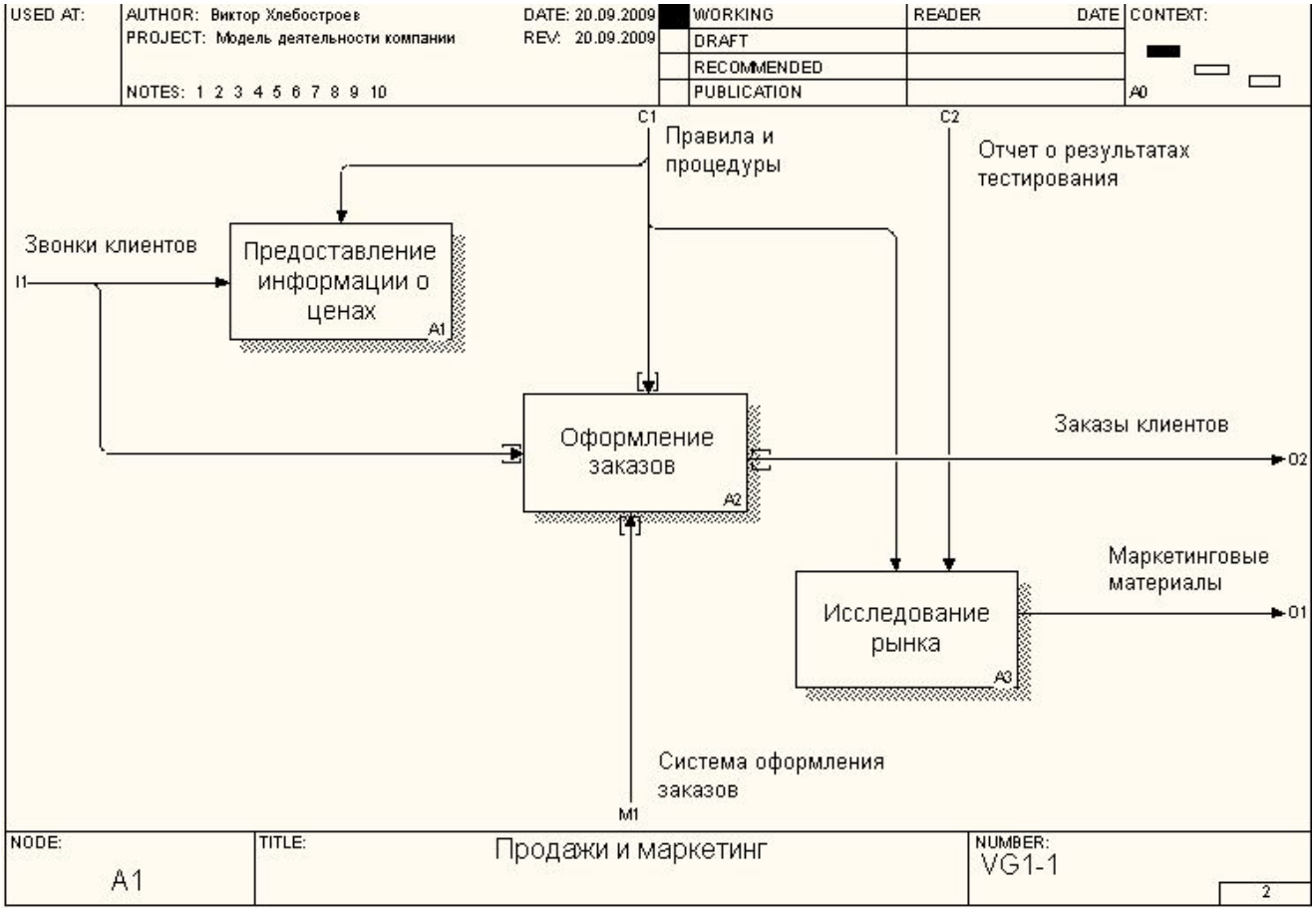

Используется в:
РД "Методология функционального моделирования IDEF0"
Автор: Прикладная Логистика
Дата: 05/12/99 20/12/99 11/01/00
Проект: Типовое Предприятие
Время: 04:51 PM 09:39 PM 03:12 PM
Замечания: 1 2 3 4 5 6 7 8 9 10
Версия: 1
x
Рабочая версия
Проект
Рекомендовано
Публикация
Читатель
Дата
Контекст:
Планы, графики проектирования и подготовки производства
Нормативные акты,регламентирующие производство, международные и внутренние стандарты
Меры по обеспечению качества продукции
C1
C2
C3
Директивы на проведение проектно-конструкторских работ
Директивы на проведение проектно-технологических работ
Директивы на организационно-экономическое проектирование
Управлять конструкторской, технологической и организационно-экономической подготовкой производства
A31
Сведения о ходе технологического проектирования
Сведения о ходе организационно-экономического проектирования
Конструкторская, технологическая, организационно-экономическая документация
Сведения о ходе проектно-конструкторских работ
Требования потребителей к продукту
I1
I2
Проектировать новый вид продукции
A32
O1
Сведения о рекламациях
Гл.инженер, директор по производству, Гл. экономист
Конструкторская документация
Проектировать технологию изготовления продукции
A33
Разрабатывать оргпроект, планы технического переоснащения, потребность в кадрах, планы запуска и освоения продукции в производстве
A34
Технологическая документация
Служба Гл. конструктора
Служба Гл. технолога
M1
ОГК,ОГТ,ППО и т.д.
Службы Гл. экономиста
Узел: A3
Заголовок: Создавать продукт, готовить производство
Номер: 4

Пример IDEF0-модели

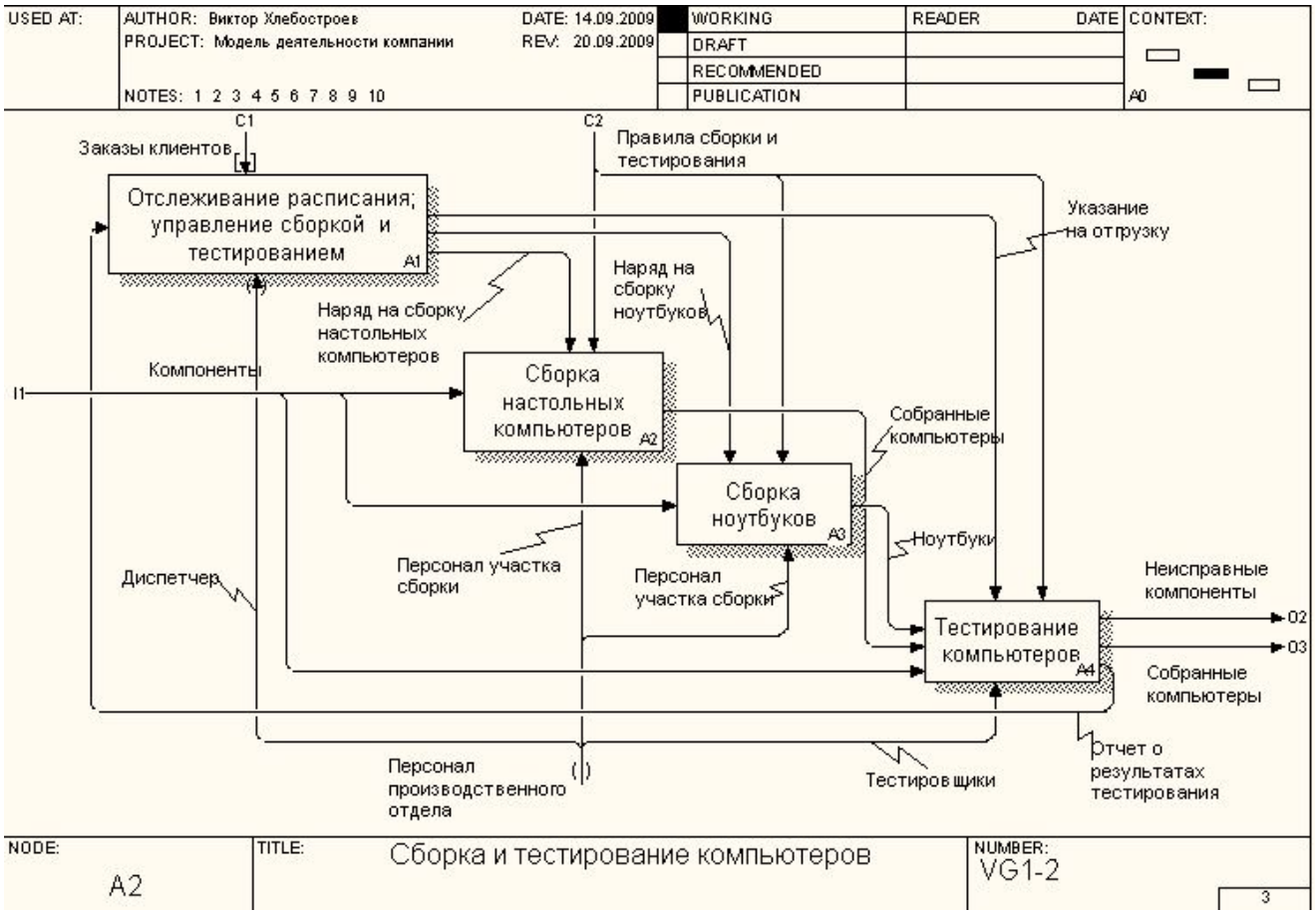
Пример IDEF0-модели



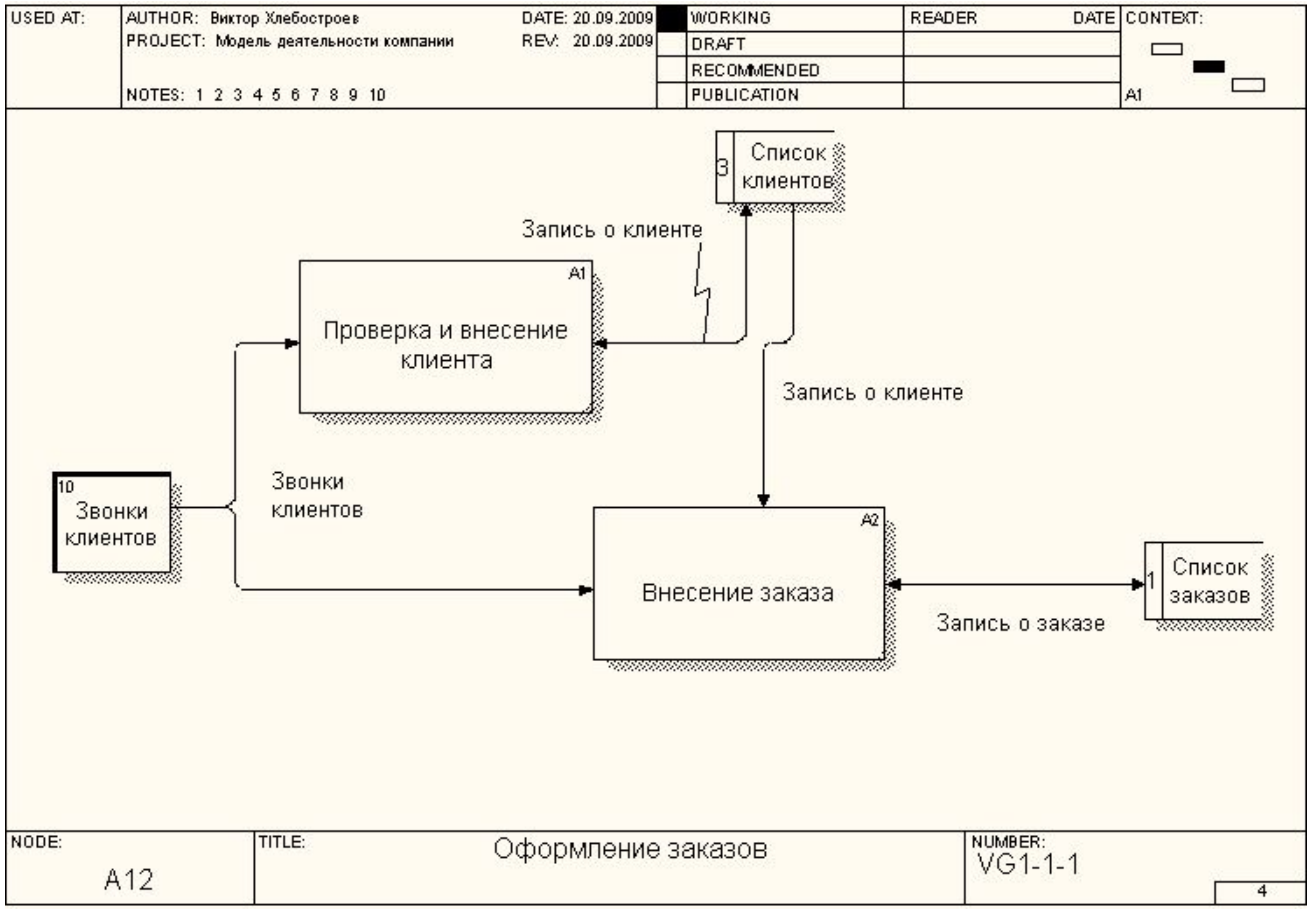
Пример IDEF0-модели



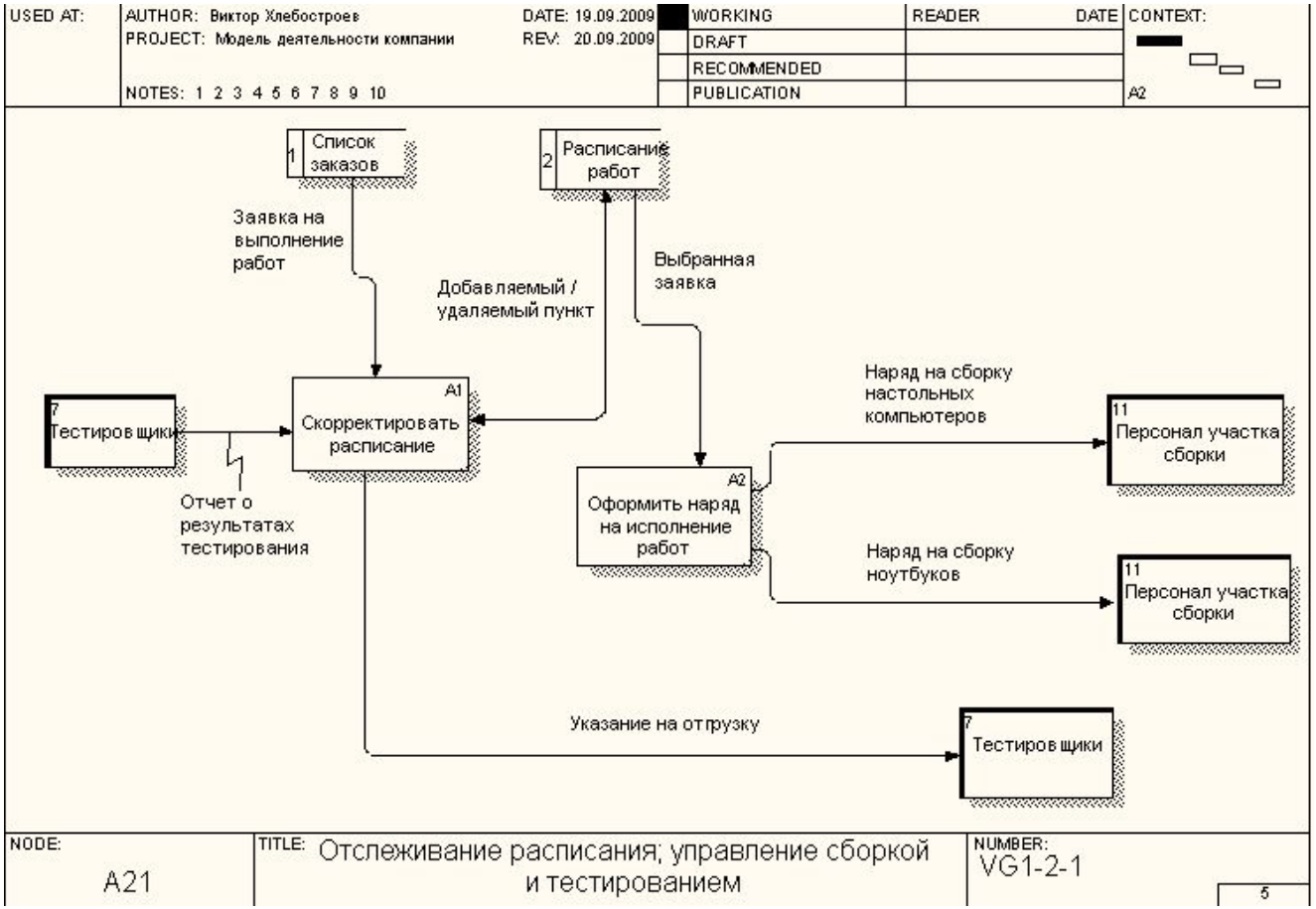
Пример IDEF0-модели



Пример IDEF0-модели



Пример IDEF0-модели



Методология IDEF3

- Предназначена для описания и документирования последовательности технологических процессов (потоков работ) в системе
- Отражает характер взаимоотношений между процессами обработки и объектами, являющимися частью этих процессов и участвующими совместно в одном процессе

Сценарии

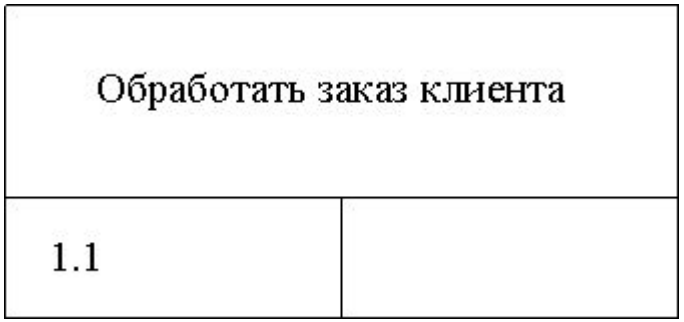
- Основой модели IDEF3 служит **сценарий** бизнес-процесса
- **Сценарием** (Scenario) называется описание последовательности изменений свойств объекта, в рамках рассматриваемого процесса

Исполнение сценария

- Исполнение каждого сценария сопровождается соответствующим документооборотом, который состоит из двух основных потоков:
  - **документов, определяющих структуру** и последовательность процесса (технологических указаний, описаний стандартов и т.д.),
  - **документов, отображающих ход его выполнения** (результатов тестов и экспертиз, отчетов о браке, и т.д.).

Диаграммы IDEF3

- Модель IDEF3, как и другие модели SADT, представляет собой иерархию диаграмм
- Основным элементом модели является **действие**
- Действие изображается прямоугольником, именуется отглагольным существительным и снабжается уникальным номером



Связи

- Взаимоотношения между действиями называются **связями** и обозначаются стрелками
- Существует три вида связей:
  - временн**о**е предшествование,
  - объектный поток,
  - нечеткое отношение

Временное предшествование

- Предыдущее действие должно завершиться прежде, чем начнется последующее
- Изображается одинарной сплошной стрелкой

Объектный поток

- Предшествующее действие завершается до начала последующего и порождает объект, который необходим для выполнения последующего действия
- Изображается двойной сплошной стрелкой

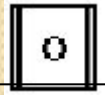
Нечеткое отношение

- Отношение между связями нельзя строго определить как отношение «предшествующий – последующий »
- Изображается одинарной пунктирной стрелкой
- Чаще всего используется для представления параллельно выполняющихся действий или альтернативных вариантов во временн**о**м следовании

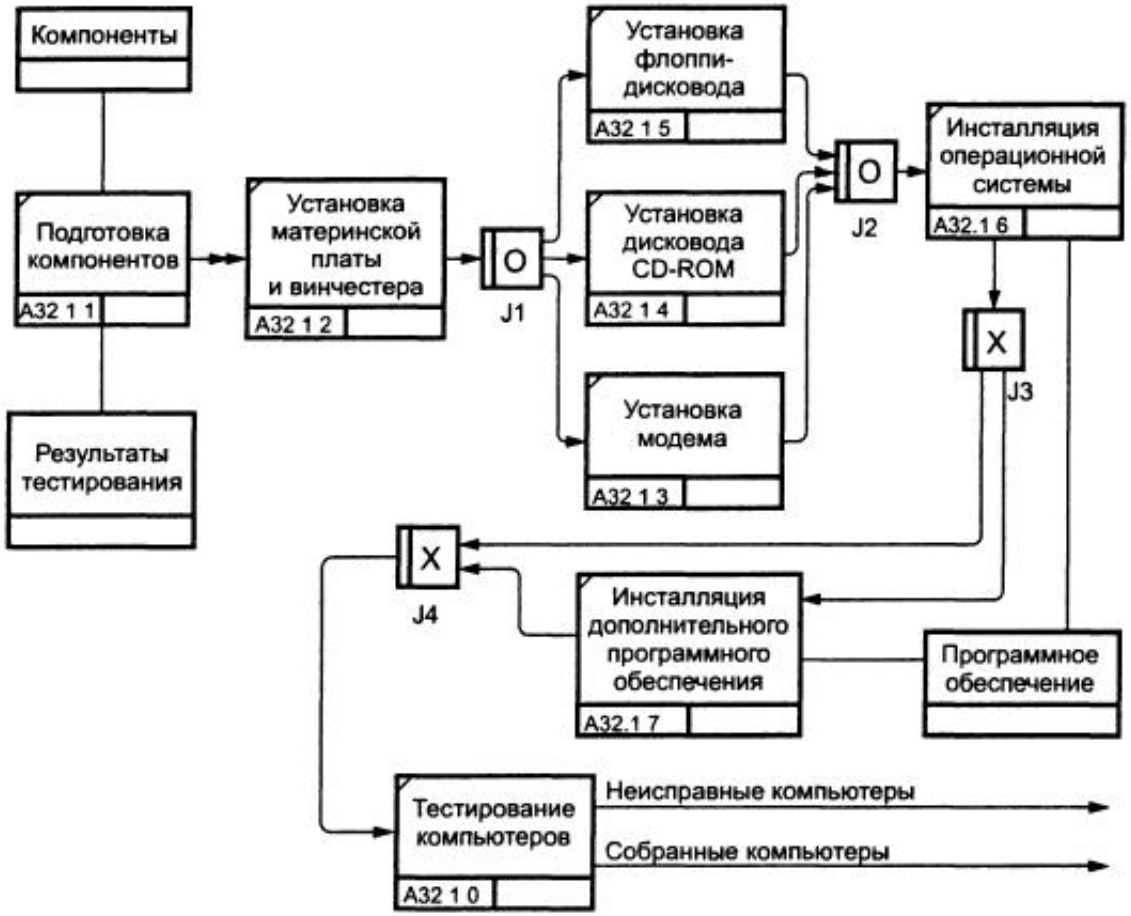
Перекрестки

- Действие может быть связано с несколькими другими действиями по входу или по выходу
- На диаграммах это приводит к необходимости разбивать одну стрелку на несколько или, напротив, объединять несколько стрелок в одну
- Для этой цели служат синтаксические элементы диаграмм, называемые **соединениями** или **перекрестками**

Типы перекрестков

Обозначение	Наименование	Смысл в случае слияния стрелок (Fan-in Junction)	Смысл в случае разветвления стрелок (Fan-out Junction)
	Asynchronous AND	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
	Synchronous AND	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
	Asynchronous OR	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
	Synchronous OR	Один или несколько предшествующих процессов завершаются одновременно	Один или несколько следующих процессов запускаются одновременно
	XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

Пример IDEF3-модели



Диаграммы потоков данных

- Диаграммы потоков данных (*Data flow diagramming*, DFD) хорошо дополняют функциональные диаграммы модели, описывая потоки данных
- Позволяют проследить, каким образом происходит обмен информацией как внутри системы между бизнес-функциями, так и системы в целом с внешней информационной средой
- Используются для описания документооборота, обработки информации

Преимущества DFD-диаграмм

- DFD-диаграммы создавались как средство проектирования программных систем, тогда как IDEF0 - как средство проектирования систем вообще
- DFD имеют более богатый набор элементов, адекватно отражающих специфику программных систем (например, хранилища данных являются прообразами файлов или баз данных)

Преимущества DFD-диаграмм

- С помощью DFD-диаграмм требования к проектируемой ИС разбиваются на функциональные компоненты (процессы) и представляются в виде сети, связанной потоками данных
- Главная цель декомпозиции DFD-функций - продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами

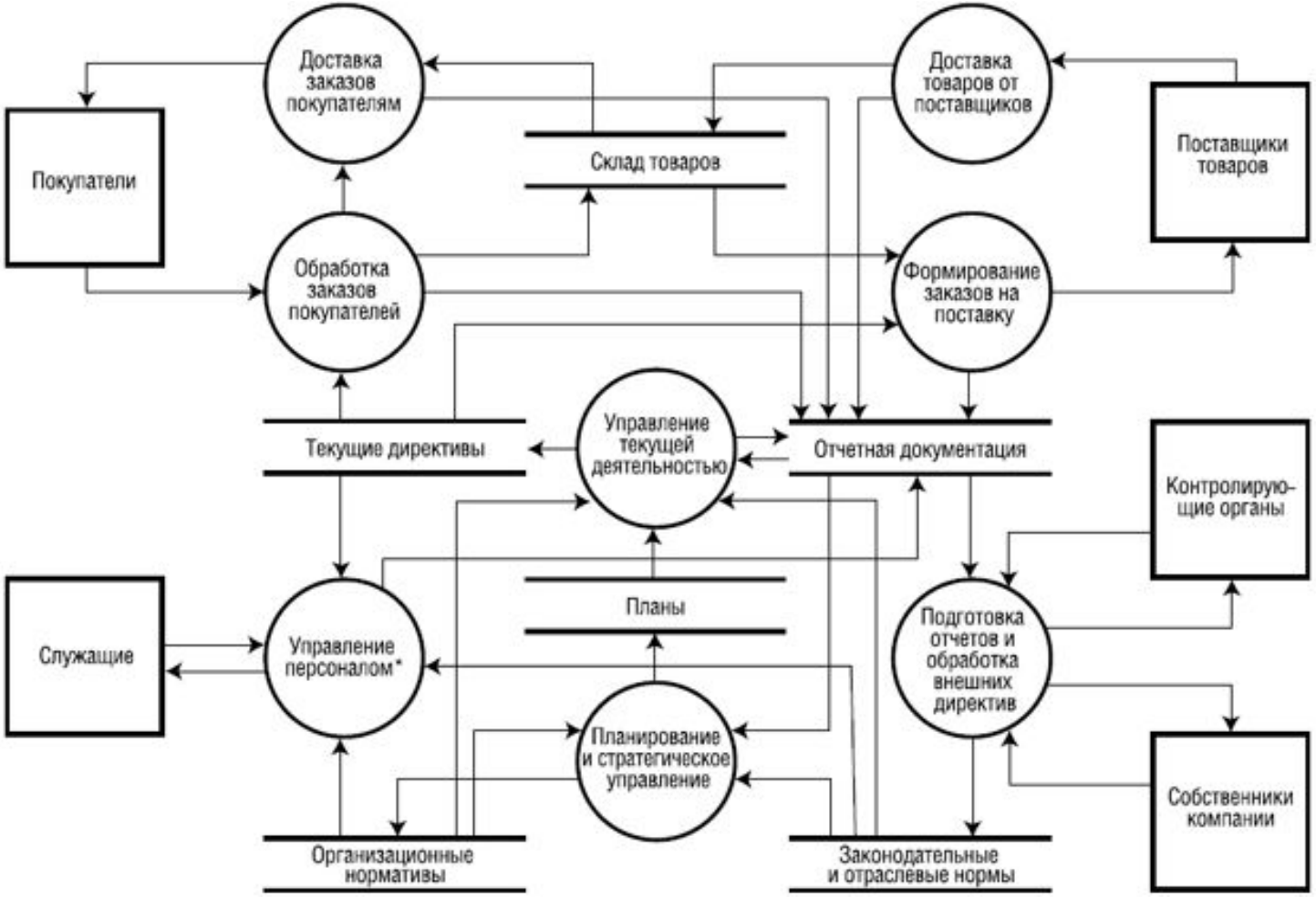
Синтаксические элементы

- На DFD-диаграммах могут присутствовать следующие элементы:
  - функциональные блоки (процессы);
  - стрелки (данные);
  - хранилища данных;
  - внешние ссылки

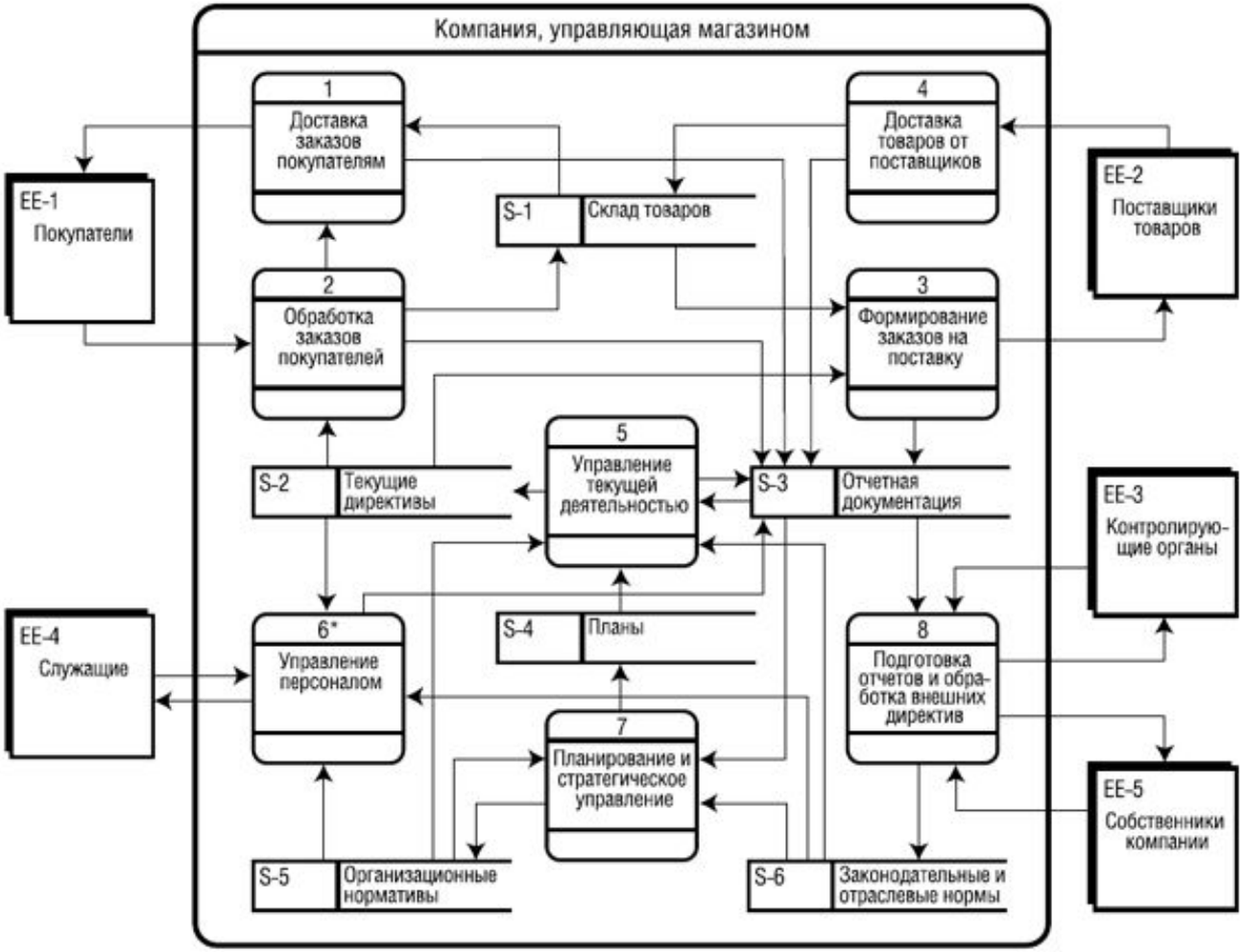
Нотации для DFD

- Используются несколько систем обозначений для перечисленных элементов
- Наиболее известны
  - нотация Йордана-ДеМарко (Yourdon-DeMarco)
  - нотация Гэйна-Сарсона (Gane-Sarson)
- Обе предложены в 1979 году

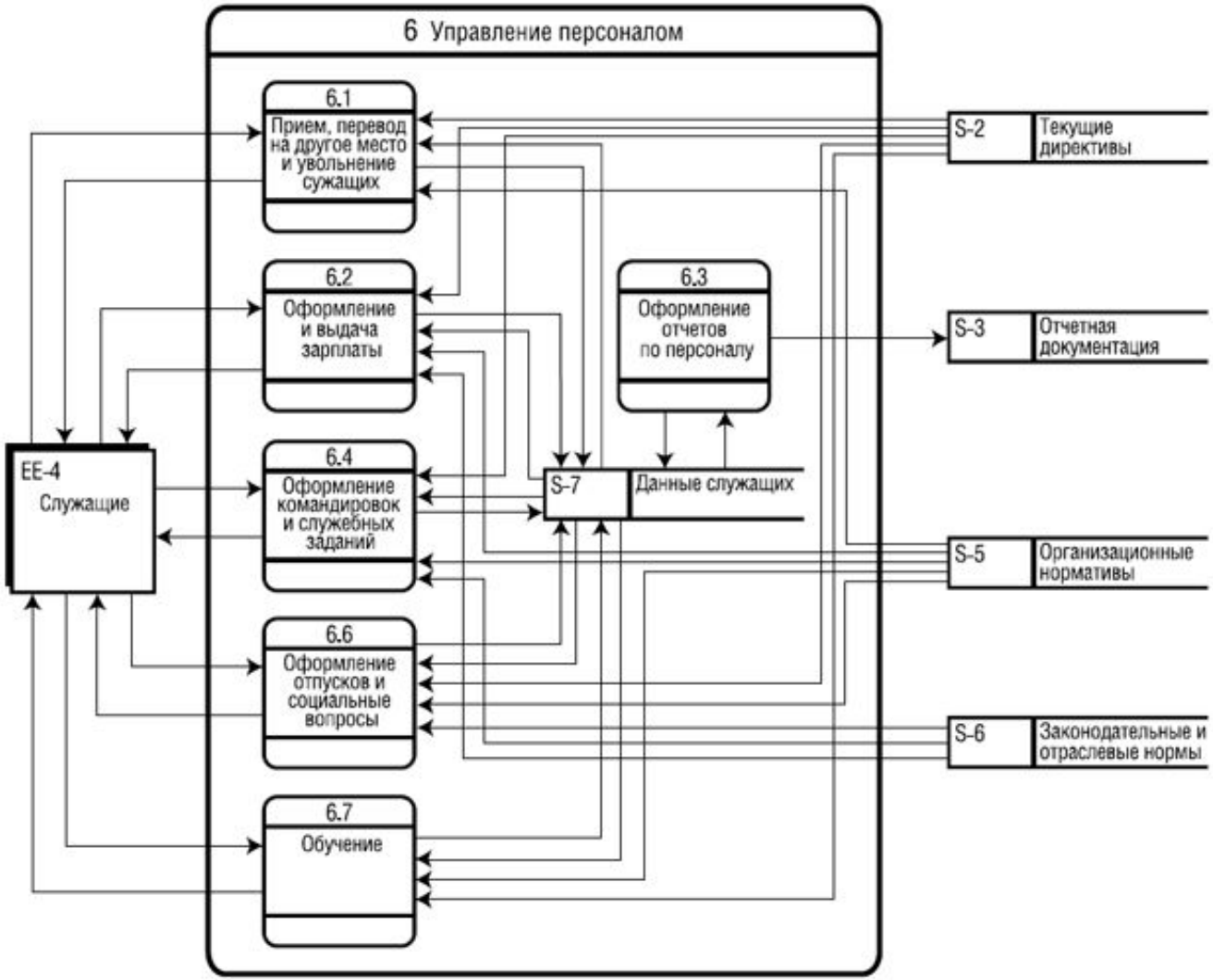
Пример нотации Йордана-ДеМарко



Пример нотации Гейна-Сарсона



Детализация процесса "Управление персоналом"



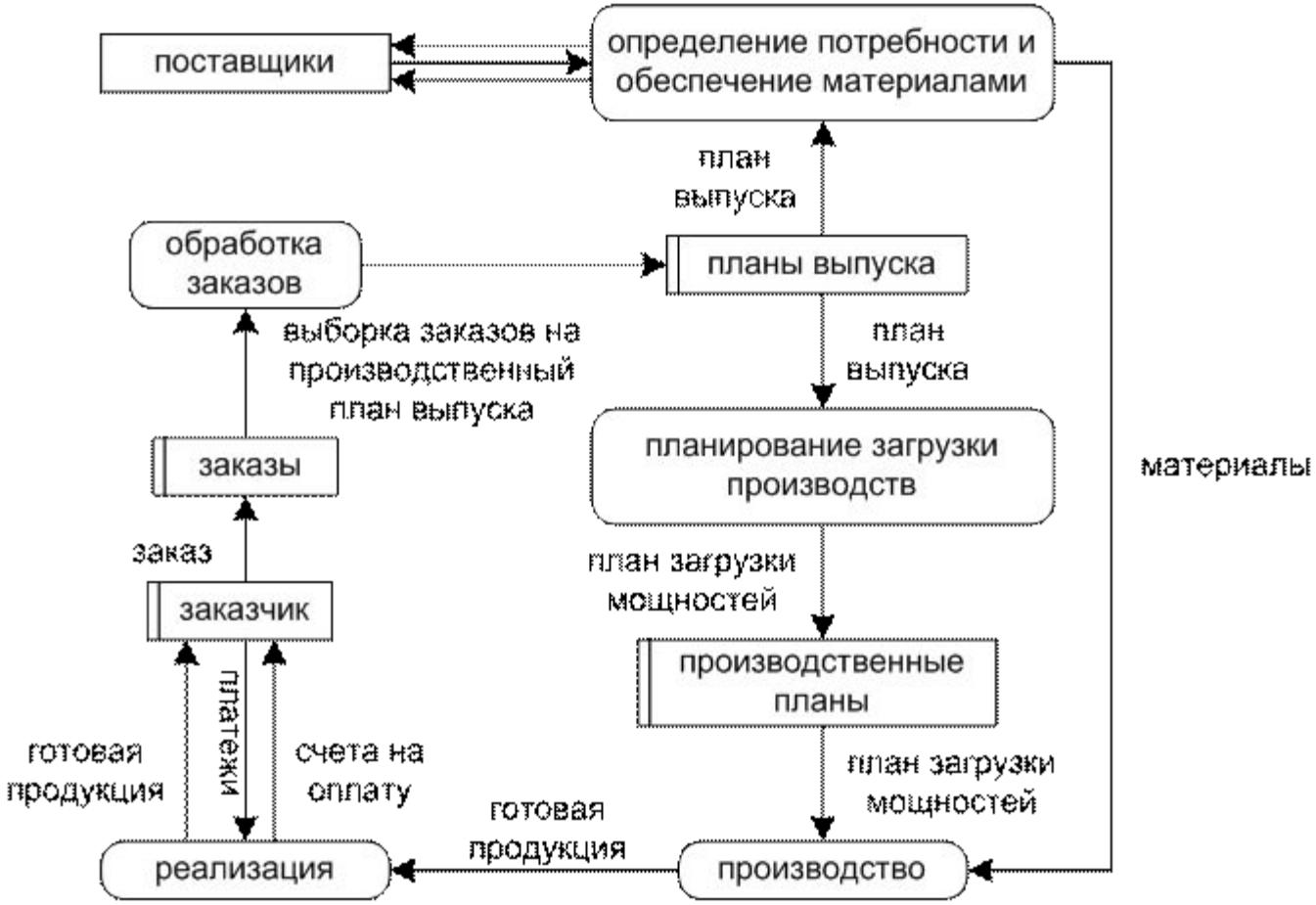
Модель «сущность-связь»

- Модель «сущность-связь» (*entity-relationship model*, ERM) – это еще способ построения концептуальных схем предметной области
- Модель «сущность-связь» была предложена в 1976 году американским профессором компьютерных наук в университете штата Луизиана Питером Пин-Шен Ченом (*Peter Pin-Shen Chen*)

Модель «сущность-связь»

- ER-модель обычно используется при высокоуровневом (концептуальном) проектировании баз данных
- С её помощью можно выделить ключевые сущности предметной области и обозначить связи, которые могут устанавливаться между этими сущностями
- ER-модель имеет графическое представление в виде ER-диаграмм

Пример ER-диаграммы



Методы объектного анализа и моделирования используются при разработке объектно-ориентированного программного обеспечения

ОБЪЕКТНОЕ МОДЕЛИРОВАНИЕ

Графические средства

- В качестве графических моделей в этих методах применяются:
  - **диаграммы вариантов использования** (вместо диаграмм потоков данных)
  - **диаграммы классов** (вместо диаграмм сущностей и связей)

Варианты использования

- **Вариантом использования (use case)** или **прецедентом** называют некоторый сценарий действий системы, обеспечивающий значимый для ее пользователей результат
- Это сценарий, неоднократно возникающий во время работы системы и имеющий определенные условия начала и завершения

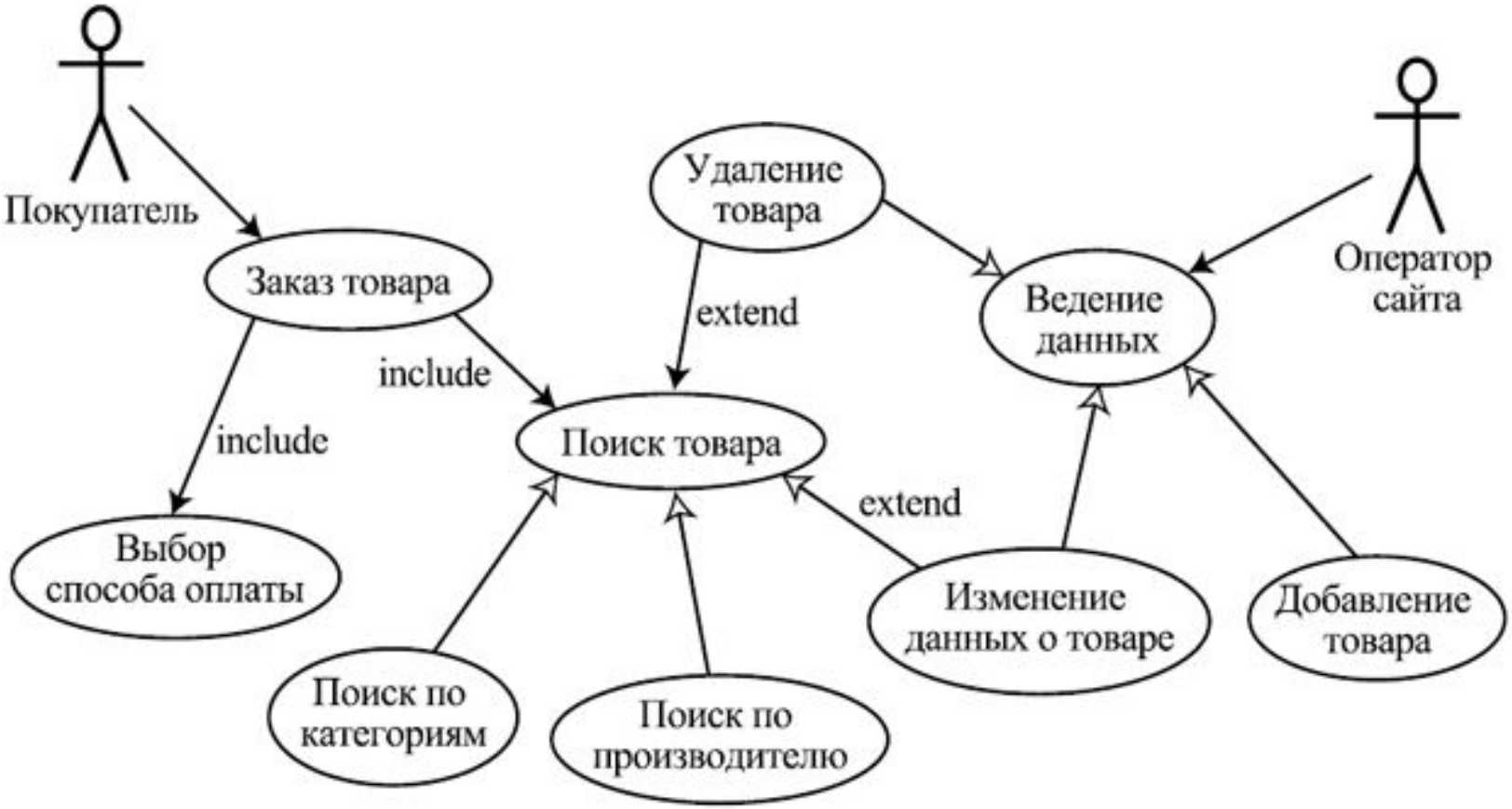
Диаграммы прецедентов

- Диаграммы вариантов использования менее информативны по сравнению с диаграммами потоков данных

  процессы + хранилища данных = варианты использования

- Кроме того, на них указываются связи между прецедентами и действующими лицами – аналогами внешних сущностей

Пример


Покупатель
Заказ товара
include
include
Выбор способа оплаты
Поиск товара
Поиск по категориям
Поиск по производителю
Удаление товара
extend
Ведение данных
Оператор сайта
extend
Изменение данных о товаре
Добавление товара

Отношение расширения

- Вариант использования А **расширяет (extends)** другой вариант использования В, если в ходе сценария работы А при определенных условиях надо включить полный сценарий работы В
- Сценарий «Удаление товара» расширяет сценарий «Поиск товара»

Отношение включения

- Вариант использования А **включает (includes, или использует, uses)** вариант использования В, если А всегда в некоторый момент включает полностью сценарий работы В
- Сценарий «Заказ товара» включает сценарий «Выбор способа оплаты»

Описание прецедента

- Должно содержать:
  - имя, говорящее о назначении прецедента
  - несколько предложений с его описанием
  - частота возникновения прецедента
  - условия его запуска – предусловия
  - условия, которые должны быть выполнены после его успешного завершения – постусловия
  - основной сценарий работы

Описание прецедента

- альтернативные сценарии с указанием условий их запуска
- действующие лица (необязательно)
- расширяемые варианты использования (необязательно)
- включаемые варианты использования (необязательно)
- статус: "в разработке", "готов к проверке", "в процессе проверки", "подтвержден", "отвергнут« (необязательно)

Дополнения

- Для представления остальной информации каждый вариант использования может дополняться набором разнообразных UML-диаграмм (взаимодействий, деятельностей, сценариев, и пр.)

СИСТЕМНЫЙ АНАЛИЗ

Проблемы

- Итогом анализа предметной области является построение ее модели
- Эта модель, в свою очередь, служит основой для выявления ***проблем*** предприятия-заказчика и его ***потребностей***, связанных с этими проблемами

Этапы определения потребностей

- Выделение небольшого числа основных проблем
- Анализ каждой из основных проблем:
  - причины возникновения
  - степень влияния на другие проблемы
- Поиск наиболее существенных проблем, влекущих появление остальных
- Определение ограничений на возможные решения

Область применения

- После выделения основных потребностей нужно решить вопрос об ***области ответственности*** будущей системы, т.е. определить, какие из потребностей надо пытаться удовлетворить в ее рамках, а какие — нет

Функции системы

- На основе выделенных потребностей пользователей формулируются возможные ***функции*** будущей системы
- Формулировка функций должна быть достаточно короткой, ясной для пользователей, без лишних деталей

Функции системы

- Например:
  - Все данные о сделках и клиентах будут сохраняться в базе данных
  - Расписание проведения ремонтных работ будет строиться автоматически
  - Система будет поддерживать до 10000 одновременно работающих пользователей

Функции системы

- Предлагая те или иные функции, нужно уметь оценивать их влияние на структуру и деятельность организаций, в рамках которых будет использоваться ПО:

  «as-is» → «to-be»

- Это можно сделать, имея полученные при анализе предметной области модели их текущей деятельности

Системный анализ

Системная спецификация

- Результаты системного анализа представляются в виде ***системной спецификации***, в которой должны быть описаны:
  - функции разрабатываемой системы,
  - ее характеристики,
  - ограничения разработки,
  - состав входной и выходной информации

Требования к ПО

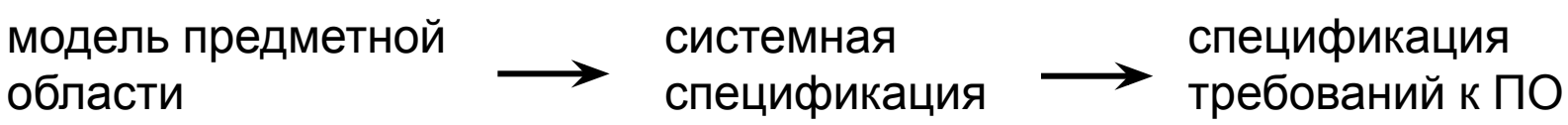
- Системная спецификация служит исходным документом при проведении анализа **требований к программной системе**
- Требования детализируют способ реализации запланированных функций

Анализ требований

- Имеет своей целью:
  - определить функции и характеристики программного продукта
  - обозначить его интерфейс с другими системными элементами
  - определить проектные ограничения программного продукта
  - выбрать формы представления информации в ходе проектирования
  - построить модели режимов функционирования продукта

Спецификация требований

- Результаты анализа требований сводятся в ***спецификацию требований к программному продукту***
- Таким образом, последовательность основных шагов этапа системного анализа выглядит следующим образом:



Конец лекции