



Жизненный цикл ПО

лектор: Парамонов А.И.



БГТУ



Software Life Cycle

В словаре программной инженерии **IEEE Std 610.12-90** «IEEE Standard Glossary of Software Engineering Terminology» записано:

Жизненный цикл ПО определяется как период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.



Выделяют следующие этапы

ЖЦ ПО:

1. формирование требований к системе (планирование)
2. проектирование
3. реализация (кодирование)
4. тестирование
5. ввод в действие
6. эксплуатация и сопровождение

* Если последний этап не включают в цикл, то принято говорить о **периоде создания ПО**



Для каждого этапа определяются:

- *состав и последовательность выполняемых работ,*
- *получаемые результаты,*
- *методы и средства, необходимые для выполнения работ,*
- *роли и ответственность участников и т.д.*

На каждом этапе ЖЦ создаются специфичные для него **модели**.

Модели формируются рабочими группами команды проекта.



СТАНДАРТЫ

регламентирующие ЖЦ ПО

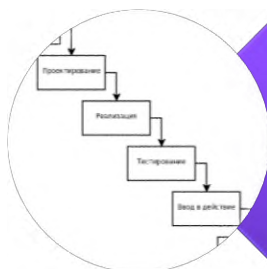
- ГОСТ 34.601-90
- ISO/IEC 12207:1995
- Custom Development Method (**CDM** – методика Oracle)
- Rational Unified Process (**RUP**)
- Microsoft Solution Framework (**MSF**)
- Extreme Programming (**XP**).

Модель ЖЦ

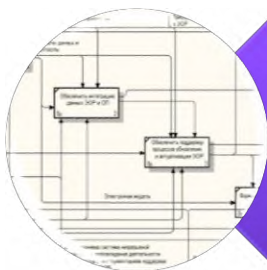
– структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, выполняемых на протяжении всего ЖЦ.



Классические модели ЖЦ



Каскадная модель



Поэтапная модель с промежуточным контролем



Спиральная модель

Водопадная модель (классическая)



Каскадная модель ЖЦ ИС

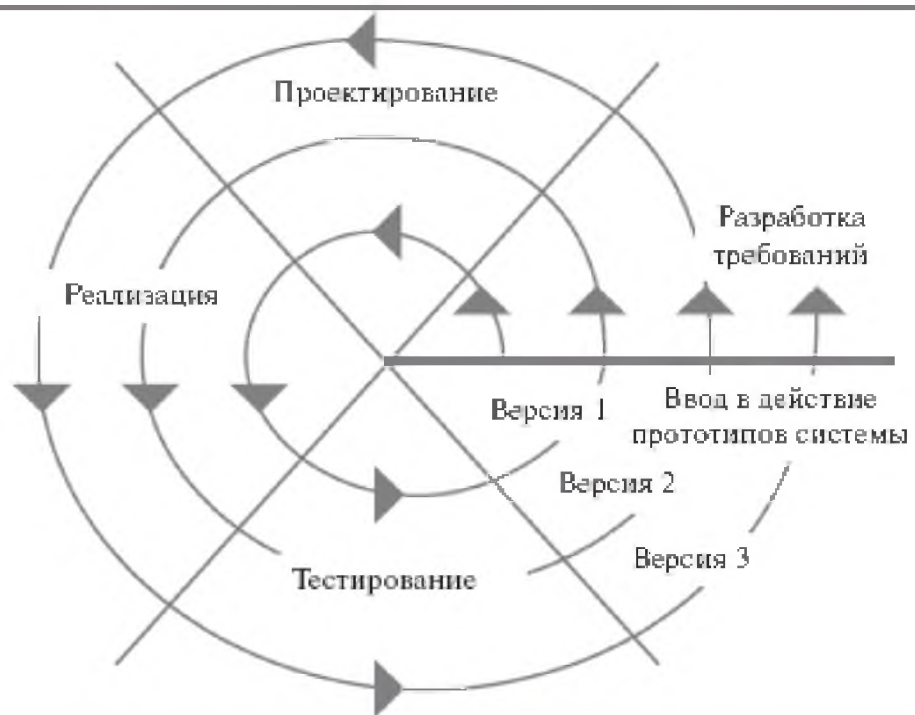
Поэтапная модель с промежуточным контролем



Итерационная модель ЖЦ ИС



Спиральная модель ЖЦ ПО



Методологии проектирования ПО

- Agile software development
- Agile Unified Process (AUP)
- Behavior Driven Development (BDD)
- Big Design Up Front (BDUF)
- Cleanroom
- Design-driven development (D3)
- Design Driven Testing (DDT)
- Domain-Driven Design (DDD)
- Dynamic Systems Development Method (DSDM)
- Evolutionary Model
- Extreme Programming (XP)
- Feature Driven Development
- Iterative and incremental development
- Kaizen
- Microsoft Solutions Framework (MSF)
- Model-driven architecture (MDA)
- Open Unified Process
- Rapid application development (RAD)
- Rational Unified Process (RUP)
- Scrum
- Software Craftsmanship
- Spiral model
- Structured Systems Analysis and Design Method (SSADM)
- Test-driven development (TDD)
- Unified Process (UP)
- V-Model
- Waterfall model
- Wheel and spoke model
- ...

Что и Как выбрать ???

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ методологий проектирования ПО

- 1) Стратегия конструирования
- 2) Адаптивность процесса
- 3) Состав этапов модели ЖЦ и связей между ними
- 4) Формулировка требований

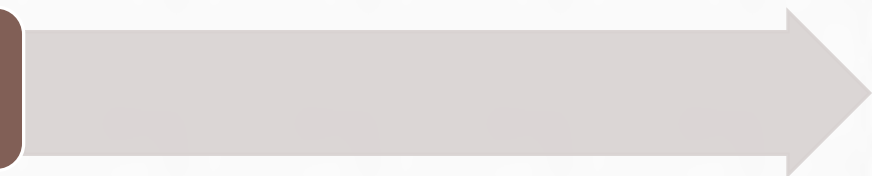
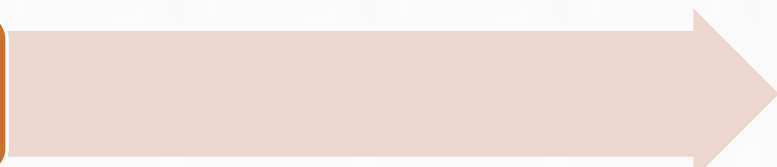
СТРАТЕГИЯ КОНСТРУИРОВАНИЯ

(в соответствии с IEEE/EIA 12207.2)

Однотактные

Инкрементные
(-итеративные)

Эволюционные





Agile Model

— семейство процессов разработки, а не единственный подход в разработке программного обеспечения, и определяется *Agile Manifesto*.



Agile не включает практик, а определяет ценности и принципы, которыми руководствуются команды.



The Agile: Scrum Framework at a glance

Inputs from Executives,
Team, Stakeholders,
Customers, Users



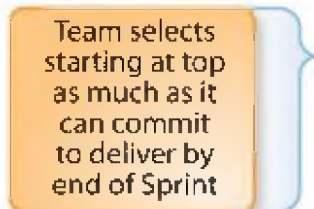
Product Owner



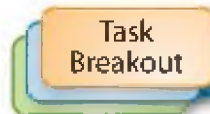
The Team



Product Backlog



Sprint Planning Meeting



Sprint Backlog



1-4 Week Sprint



Scrum Master



Burndown/up Charts



Daily Scrum Meeting

Every 24 Hours



Sprint Review



Finished Work



Sprint Retrospective

Sprint end date and team deliverable do not change



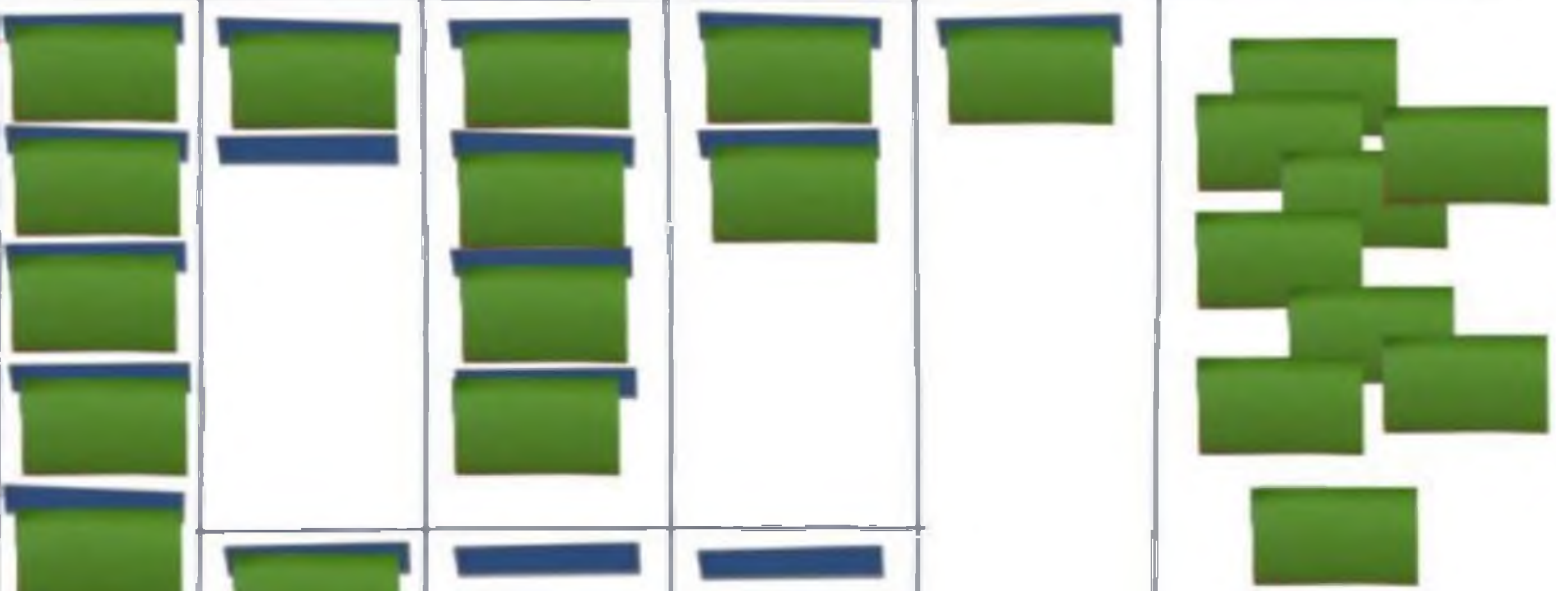
Kanban

EXPEDITE →



GOALS STORY QUEUE ELABORATION & ACCEPTANCE DEVELOPMENT TEST DEPLOYMENT DONE!

YOUR WAIT TIME FROM HERE IS ABOUT 14 DAYS



YOUR WAIT TIME FROM HERE IS ABOUT 18 DAYS

5 3 5 3 1

Подходы к выбору методологии

Принцип 1.

Большая по размерам методология нужна тогда, когда в проекте занято большое число разработчиков.

Принцип 2.

Большая корректность методологии (или "большая плотность») нужна в тех случаях, когда скрытые ошибки в ПО могут повлечь за собой значительный ущерб (большая критичность разрабатываемой системы).

Принцип 3.

Незначительное увеличение "размеров" или "плотности" методологии ведет к существенному увеличению стоимости проекта.

Принцип 4.

Наиболее эффективная форма коммуникации (для передачи идей) — непосредственное взаимодействие, лицом к лицу, как при рисовании у доски.

«Истинное знание — это знание о невежестве» ® Конфуций



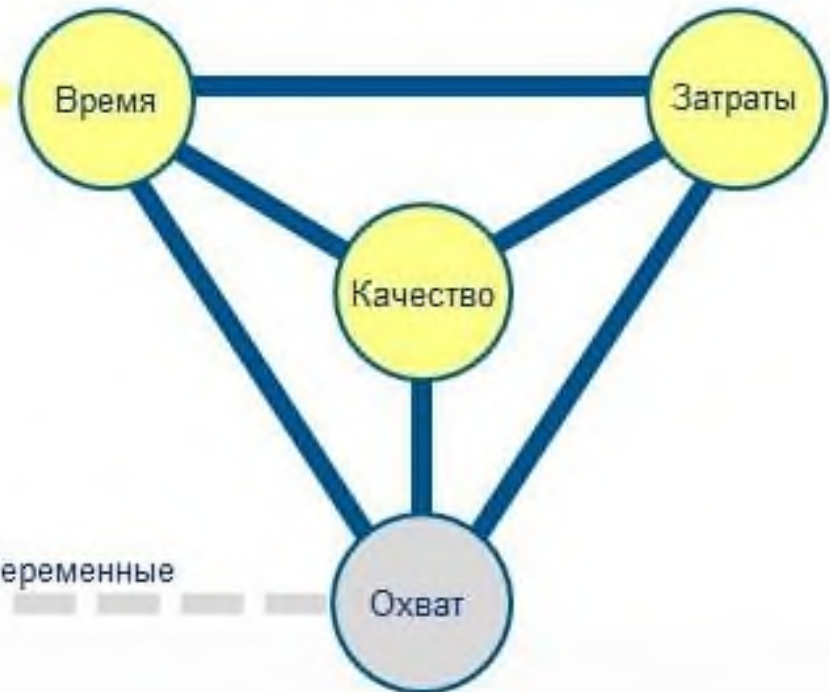
Эффект Даннинга-Крюгера

Dynamic System Development Method

Традиционный подход



Подход DSDM



Постоянные

Переменные

Принципы DSDM

- Пользователи или заказчик (или и те, и другие) участвуют в процесс разработки.
- Команда может самостоятельно принимать решения по проекту.
- Работа делится на спринты, после каждого спринта команда показывает результат или кусок готового продукта.
- Результат всегда должен быть в мире клиента или пользователя, он полезен бизнесу.
- Используют итеративный и инкрементный подход к разработке.
- Любые действия можно отменить, откатиться назад.
- Продукт постоянно и непрерывно тестируют, обратную связь используют для улучшения.

Модель SCORE





SCORE — аббревиатура:

S — symptoms (симптомы)

C — causes (причины)

O — outcomes (результаты)

R — resources (ресурсы)

E — effects (эффекты)



Модель SCORE —
универсальная методика сбора
информации для решения
проблем и достижения целей.

Помогает разложить ситуацию по
полочкам и понять, как перейти из
текущего состояния в желаемое.

Модель разработали в 1987 году
Роберт Дилтс и Тодд Эпштейн,
*специалистами по
нейролингвистическому
программированию.*



Nearly Universal Principles of Projects

СОВМЕСТИМ СО ВСЕМИ ОСНОВНЫМИ МЕТОДАМИ,
СИСТЕМАМИ, ПОДХОДАМИ И ФРЕЙМВОРКАМИ,
ТАКИМИ КАК PRINCE2®, PMBOK® Guide,
P3.express, PM², DSDM®, XP, and Scrum.

Передовые практики важны только тогда, когда они могут эффективно обеспечивать руководство работами и определять рамки для принятия решений.





Информационные системы

Основные понятия

лектор: Парамонов А.И.



БГТУ

Система –

отграниченное, взаимно связанное
множество, отражающее
объективное существование
конкретных отдельных
взаимосвязанных совокупностей
объектов и не содержащее
специфических ограничений,
присущих частным системам



Основные свойства системы

Свойство	Характеристика
Ограниченность	<i>отделена от окружающей среды границами</i>
Целостность	<i>свойство целого принципиально не сводится к сумме свойств составляющих элементов</i>
Структурность	<i>поведение обусловлено не только особенностями отдельных элементов, сколько свойствами ее структуры</i>
Взаимозависимость со средой	<i>формирует и проявляет свойства в процессе взаимодействия со средой</i>
Иерархичность	<i>соподчиненность элементов</i>
Множественность описаний	<i>по причине сложности познание требует множественности взглядов</i>





Информационная система

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

A vertical strip of four images on the left side of the slide. From top to bottom: a blue clock face on a blue background, a red clock face on a red background, a green open book on a green background, and a blue open book on a blue background.

Модель –

**формальное описание
особенностей системы,
которые существенны
для целей её
исследования.**

Виды *моделей*:


- **вербальные**
- **натурные**
- **знаковые**





Математическая модель –

описание протекания процессов
(в том числе функционирования,
движения), описание состояния,
изменения системы на языке
алгоритмических действий с
математическими формулами и
логических переходов.



Формирование общего представления о системе :

- Стадия «изучение системы».
- Стадия «*формирование углубленных представлений о системе*».
- Стадия «моделирования системы».
- Стадия «*сопровождение системы*».



Изучение системы

- 1. Выявление главных функций** (свойств, целей, предназначения) **системы.**
- 2. Выявление основных частей** (модулей) **в системе** и их функций...
- 3. Выявление основных процессов в системе,** их роли, условий осуществления...
- 4. Выявление основных элементов «не-системы»,**
с которыми связана изучаемая система.
Выявление характера этих связей.
- 5. Выявление неопределенностей и случайностей** в ситуации их определяющего влияния на систему и выбор способа их математической формализации.



Анализ системы

6. **Выявление** разветвленной **структуры**, иерархии, формирование представлений о системе как о совокупности модулей, связанных входами-выходами.
7. **Выявление всех элементов и связей**, важных для целей рассмотрения...
8. **Учет изменений** и неопределенностей в системе, входов и постоянных параметров.
9. **Исследование функций и процессов** в системе с целью управления ими...



Сопровождение системы

11. **Накопление опыта** работы с системой и ее моделью.
12. **Оценка предельных возможностей** системы.
13. **Расширение функций** (свойств) системы, изменение требований к ней, новый круг задач, новые условия работы.



ИНФОРМАЦИОННЫЙ ПОТОК

— совокупность циркулирующих в системе, между системой и внешней средой сообщений, необходимых для управления, анализа и контроля операций.



Информационные потоки (по структуре)

- Однородные
- Неоднородные

Информационные потоки (по периодичности)

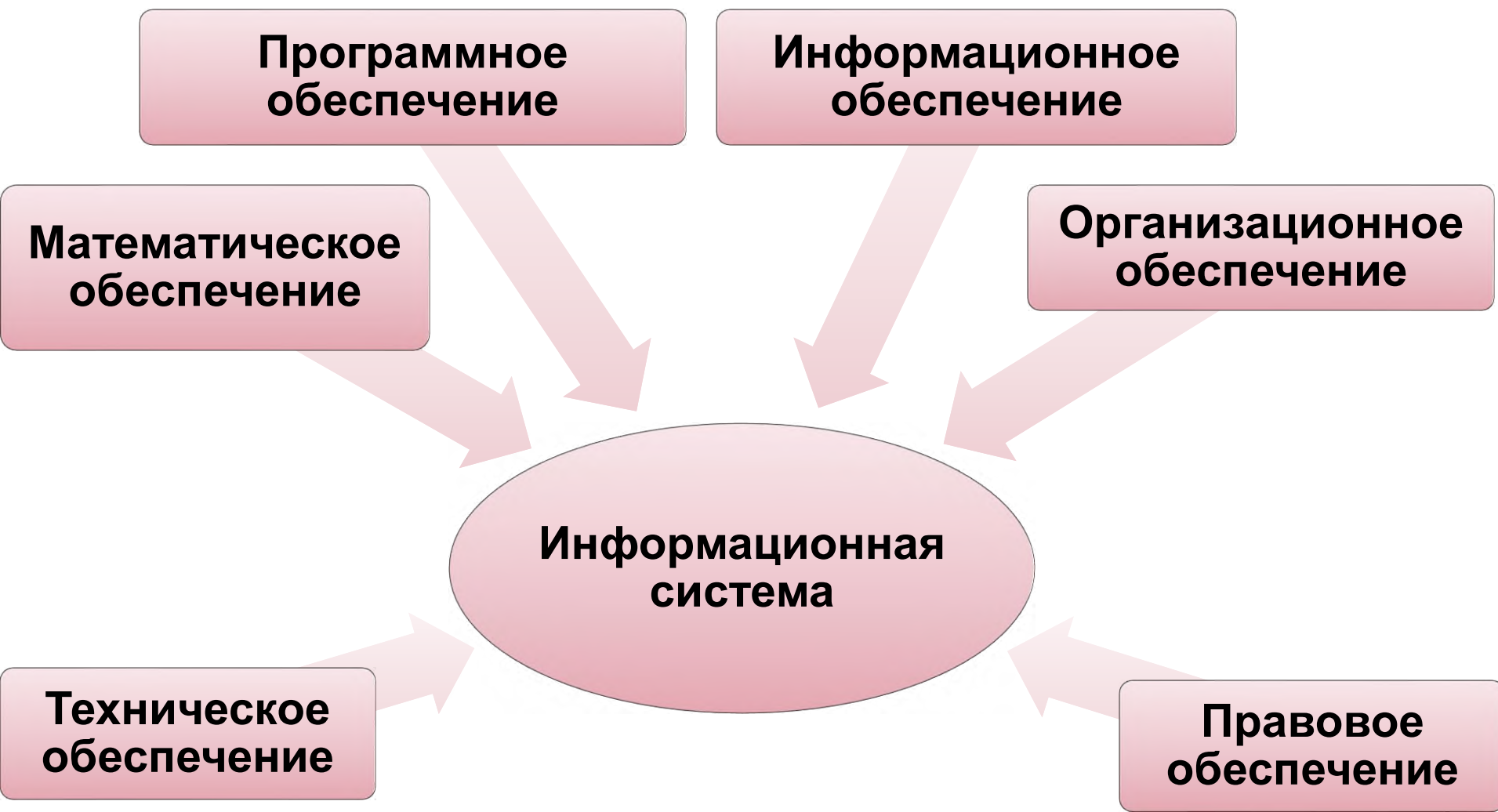
- Регулярные
- Оперативные



Характеристики информационных потоков :

- ИСТОЧНИК ВОЗНИКНОВЕНИЯ;
- направление движения потока;
- скорость передачи и приема;
- интенсивность потока и др.

Типы обеспечивающих подсистем



Информационное обеспечение –

- совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения баз данных.



Техническое обеспечение –

- комплекс технических средств, предназначенных для работы ИС, а также соответствующая документация на эти средства и технологические процессы.



Математическое и программное обеспечение —

- совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач ИС.



Программное обеспечение:

- общесистемные продукты;
- специальные программные продукты;
- техническая документация



Организационное обеспечение –

- совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.



Правовое обеспечение –

- совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем, регламентирующих порядок получения, преобразования и использования информации.



Классификация ИС





ОСНОВЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ





Результаты планирования

Основными документами, содержащими требования на разработку информационной системы, являются :

- **календарный план** выполнения работ – регламентирует состав, сроки и финансирование работ
- **техническое задание** – отражает основные требования к системе.

Техническое задание (ТЗ)

- является основным документом, определяющим требования и порядок создания (развития или модернизации) системы.



Вариант содержания ТЗ

- общие сведения о системе;
- назначение и цели создания (развития) ИС;
- характеристика объектов автоматизации;
- **требования к ИС в целом, к функциям и обеспечению;**
- состав и содержание работ по созданию системы;
- порядок контроля и приемки системы;
- требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;
- требования к документированию;
- источники разработки.





ПРОЕКТИРОВАНИЕ

процесс перехода от одной модели в виде первичного описания системы (ТЗ) к ее описанию в виде набора стандартных документов, достаточных для создания ИС (*проектной документации*)



ОСНОВНЫЕ ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ



1. Принцип декомпозиции
("разделяй и властвуй")



2. Принцип иерархического
упорядочения



3. Принцип концептуальной
общности



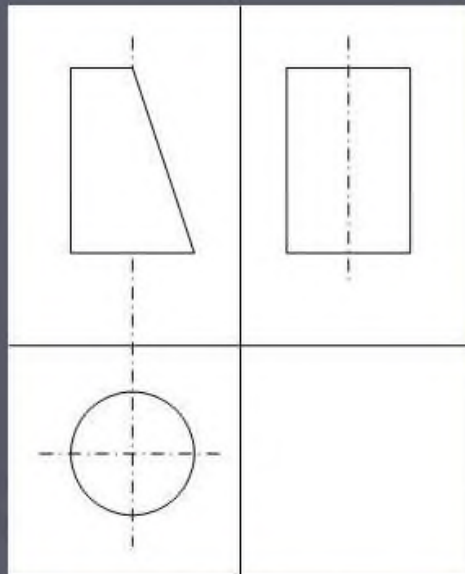
4. Принцип абстрагирования

5. Принцип формализации



6. Принцип унификации
7. Принцип логической независимости
8. Принцип многомодельности
9. Принцип непротиворечивости (согласованности)
10. Принцип информационной закрытости (инкапсуляции)
11. Принцип полиморфизма.

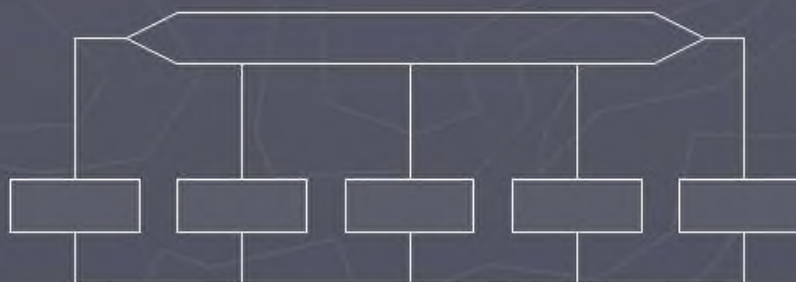
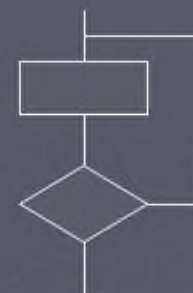
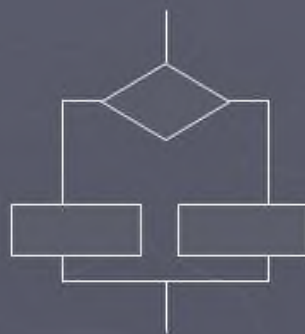
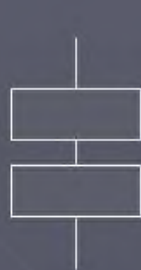
Модель как проекция «системы» - точка зрения на ИС



- При моделировании систем реального мира и нетривиальных программных систем выбор проекций неочевиден.
- Проекция (**точка зрения**) определяет, на какие вопросы может ответить модель.

Визуальное представление моделей

■ Блок-схема



Визуальное представление моделей

■ Граф переходов



Визуальное представление моделей

■ Конечный автомат

Конечный автомат может быть задан параметрами:

- Q – конечное множество состояний
- $q_0 \in Q$ – начальное состояние
- $F \in Q$ – множество заключительных состояний
- Σ – конечное множество входных символов
- δ – заданное отображение $Q \times \Sigma \rightarrow Q$ (функция переходов)

$$\delta = \delta(q_i, b), \text{ где } b \in \Sigma$$

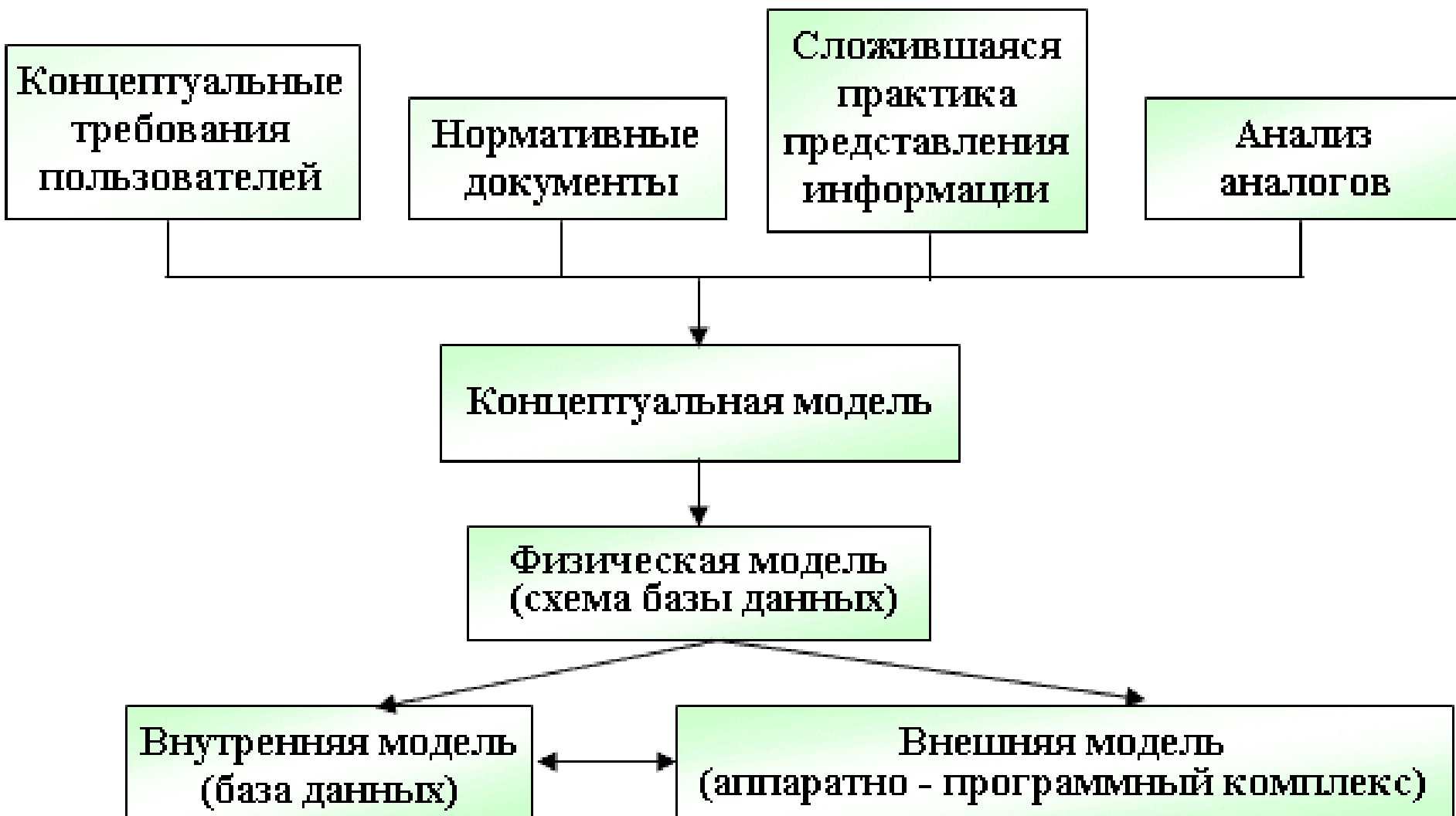
Классификация МОДЕЛЕЙ ИС

- **По строгости описания:**
 - *Неформальные*
 - *Формальные*
- **По степени физической реализации**
(логической независимости):
 - *Логические*
 - *Физические*

Классификация МОДЕЛЕЙ ИС

- По степени отображения динамики происходящих процессов:
 - *статические*
 - *динамические*
- По отображаемому аспекту:
 - *функциональные*
 - *информационные*
 - *поведенческие*
 - *компонентные*
 - *смешанные*

Схема формирования информационной модели



Создание информационной системы ведется в несколько этапов, на каждом из которых конкретизируются и уточняются элементы разрабатываемой системы.

- **Концептуальная модель** - отображает информационные объекты, их свойства и связи между ними без указания способов физического хранения информации.

Модель предметной области, иногда ее также называют **информационно-логической** или **инфологической** моделью.

- Информационными объектами обычно являются **сущности** (обособленные объекты или события, информацию о которых необходимо сохранять), имеющие определенные наборы свойств (**атрибутов**).

- **Физическая модель** - отражает все свойства (атрибуты) информационных объектов базы и связи между ними с учетом способа их хранения (например, используемой СУБД).
- **Внутренняя модель** - *база данных*, соответствующая определенной физической модели.
- **Внешняя модель** - комплекс программных и аппаратных средств для работы с базой данных, обеспечивающий процессы создания, хранения, редактирования, удаления и поиска информации, а также решающий задачи выполнения необходимых расчетов и создания выходных печатных форм.



CASE-технология –

это методология проектирования ИС,
а также набор методов, нотаций и
инструментальных средств,

которые позволяют в наглядной
форме *моделировать* предметную
область, *анализировать* модель ИС
на всех этапах её ЖЦ и

разрабатывать приложения в
соответствии с информационными
потребностями пользователей.



Основные цели использования CASE-технологий :

- **максимальная автоматизация** стадий анализа и проектирования ИС с целью построения формальных и непротиворечивых моделей системы.
- **вынесение части деятельности** (*чем больше, тем лучше*) из стадии кодирования **в стадию проектирования.**

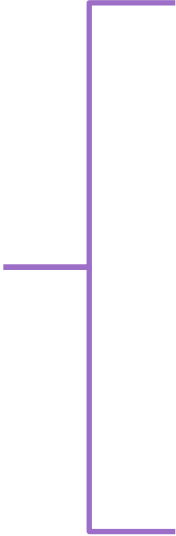


Большинство современных CASE-
средств поддерживает методологии
разных подходов к проектированию ИС.

Выбор той или иной парадигмы
подразумевает следование одному
взгляду на ИС на всех стадиях (согласно
принципу концептуальной общности).

- **НОТАЦИЯ** – установленные способы отображения элементов системы, *т.е.* графы, таблицы, блок-схемы, формальные и естественные языки.
- **ПАРАДИГМА** – исходная концептуальная схема (модель) постановки проблемы и ее решения.

Подходы к проектированию ИС



```
graph LR; A[Подходы к проектированию ИС] --- B[Структурное моделирование]; A --- C[Объектно-ориентированное моделирование];
```

**Структурное
моделирование**

**Объектно-
ориентированное
моделирование**



СТРУКТУРНЫЙ ПОДХОД

- Сущность подхода к разработке модели состоит в расчленении анализируемой системы на части («черные ящики») и их иерархической организации.

Структурным анализом принято называть метод исследования статических характеристик системы путем выделения в ней подсистем и элементов различного уровня иерархии, определения отношений и связей между ними.

- Преимущество работы с «черными ящиками»: нет необходимости знать как они работают – достаточно иметь информацию об их входах и выходах, а также функциях, которые они выполняют.



Структурный анализ

Сфера применения: проектирование производственно-экономических и инженерно-технических систем.


- анализ информационных потоков на предприятии,
- ре-инжиниринг бизнес-процессов,
- компьютеризация деятельности предприятия,
- разработка систем автоматизированного проектирования,
- разработка баз данных,
- разработка программных приложений, реализующих управление информационными потоками (например, системы электронного документооборота)

В моделировании бизнес-процессов структурный подход базируется на **3 основных положениях:**



1. разбиение исследуемого процесса на функциональные блоки — **подпроцессы**;
2. возможность детализации любых процессов путем иерархической **декомпозиции**;
3. использование для описания процесса **графических нотаций** с возможностью текстового разъясняющего дополнения.

Схема применения структурного подхода



Разработка функциональной модели

Разработка информационной модели


Разработка поведенческих моделей

Разработка моделей компонентов и
развертывания



Методологии структурного анализа и проектирования

Методология	Тип разрабатываемой модели
<u>SADT</u> (Structured Analysis and Design Technique, методология структурного анализа и проектирования)	Функциональная
<u>DFD</u> (Data Flow Diagrams, диаграммы потоков данных)	Функциональная или компонентная
<u>ERD</u> (Entity-Relationship Diagrams, диаграммы "сущность-связь")	Информационная
<u>Flowcharts</u> (блок-схемы)	Поведенческая
<u>EPC</u> (Event-driven Process Chain, событийная цепочка процессов)	Функциональная или поведенческая
<u>BPMN</u> (Business Process Model and Notation, модель и нотация бизнес-процессов)	Функциональная или поведенческая



SADT (Structured Analysis and Design Technique)

- методология структурного анализа и проектирования.

Суть методологии:

анализируемый процесс представляется в виде совокупности взаимосвязанных действий, которые имеют четко определенные вход и выход и взаимодействуют между собой на основе определенных правил и с учетом потребляемых информационных, человеческих и производственных ресурсов



История развития структурного подхода

- SA – Structured Analysis (1960-е – середина 1970-х)
 - системы автоматизированного проектирования,
 - структурный анализ при создании алгоритмических языков.
- SADT – Structural Analysis and Design Technique (1974)
 - методология структурного проектирования
- Программа ICAM - Integrated Computer-Aided Manufacturing (конец 1970-х)
 - интегрированная компьютеризация производства США
 - начало разработки методологии IDEF (ICAM Definition)



IDEF

Integration Definition Metodology (Объединение Методологических Понятий)

*Семейство совместно
используемых методов для
процесса моделирования.*




Описание и назначение методологии *IDEF0*

лектор: Парамонов А.И.

IDEFO

— методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов.



В IDEFo реализованы **три базовых принципа** моделирования бизнес-процессов:

- 1.** принцип функциональной декомпозиции;
- 2.** принцип ограничения сложности;
- 3.** принцип контекста.

Принцип 1

В соответствии с *принципом функциональной декомпозиции* сложная бизнес-функция может быть представлена в виде совокупности составляющих ее более простых функций, которые сами в свою очередь могут быть подвергнуты декомпозиции.

Принцип 2

- Согласно *принципу ограничения сложности* количество функциональных блоков на одной диаграмме **должно быть не менее двух (за исключением контекстной диаграммы) и не более шести**.
- Таким образом обеспечивается разборчивость и удобочитаемость диаграмм IDEFo. Практика показывает, что соблюдение этого принципа в большинстве случаев приводит к тому, что процессы, представленные в виде модели IDEFo, становятся лучше структурированы, более понятны и легче поддаются анализу.

Принцип 3

- **Принцип контекста** состоит в том, что моделирование бизнес-процесса начинается с построения *контекстной диаграммы*.
- На этой диаграмме отображается только один блок — главная бизнес-функция моделируемой системы.
- При определении главной бизнес-функции необходимо всегда иметь в виду цель моделирования и точку зрения на модель.

Диаграмма IDEF0

- Основной структурной единицей IDEF0-модели является **диаграмма**, представляющая собой графическое описание модели предметной области или ее части.

диаграммы IDEFo представлены в «рамках»

ИСПОЛЬЗУЮТСЯ:	АВТОР:	ДАТА: 28 мар 1997	РАБОЧАЯ ВЕРСИЯ	ЧИТАТЕЛЬ	ДАТА	КОНТЕКСТ
	ПРОЕКТ:	ПЕРЕСМОТР	ЭСКИЗ			
			РЕКОМЕНДОВ			
	ЗАМЕЧАНИЯ: 1 2 3 4 5 6 7 8 9	ПУБЛИКАЦИ				
УЗЕЛ:	НАЗВАНИЕ:				НОМЕР	
A0					2	
						Стр.:

Синтаксис IDEFo

Диаграммы содержат **блоки, стрелки и правила**.

- **Блоки** представляют функции, определяемые как деятельность, процесс, операция, действие или преобразование.
- **Стрелки** представляют данные или материальные объекты, связанные с функциями.
- **Правила** определяют, как следует применять компоненты; диаграммы обеспечивают формат графического и словесного описания моделей.

Блок

Блоки изображаются в виде *прямоугольников*.

Внутри каждого блока помещается его **имя** и **номер**.

Имя должно быть активным глаголом или глагольным оборотом, описывающим функцию.

Номер блока размещается в правом нижнем углу. Номера блоков используются для их идентификации на диаграмме и в соответствующем тексте.

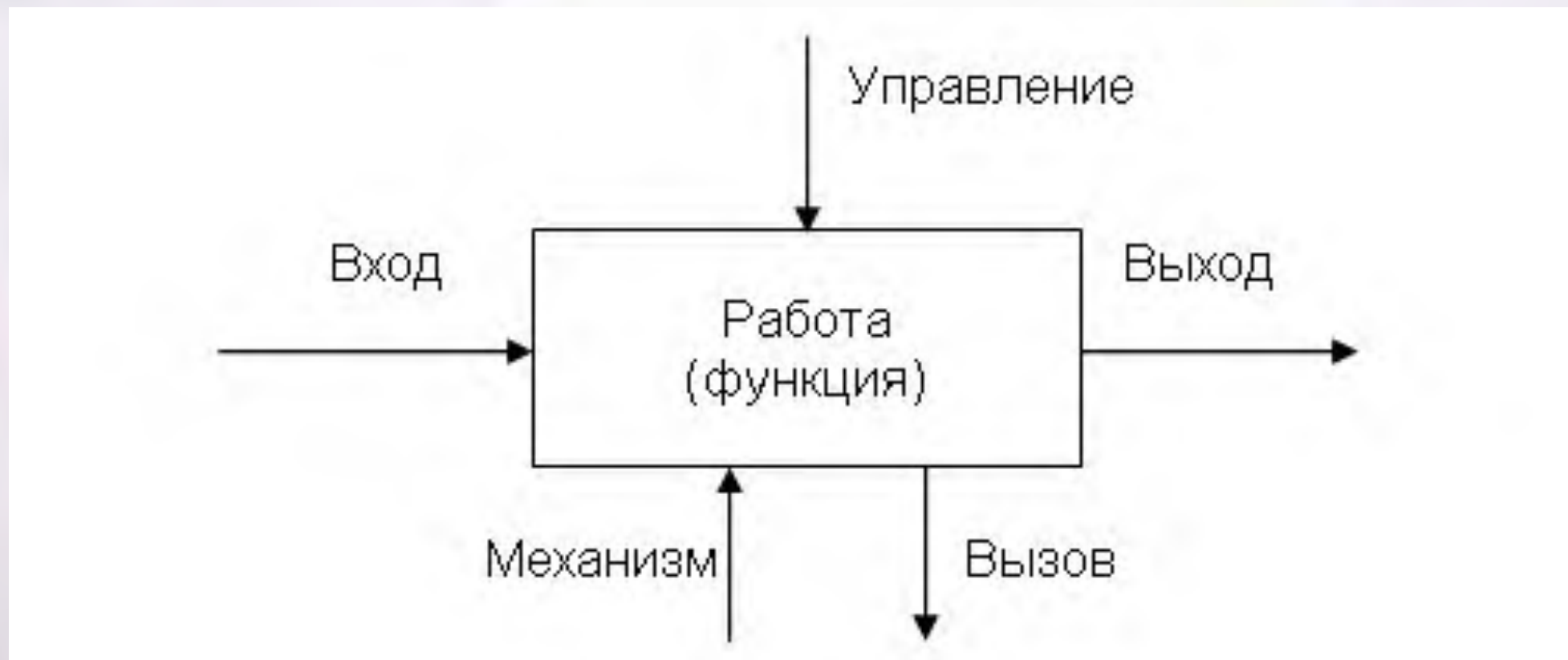


- Имя функции –глагол или глагольный оборот
- Показан номер блока

Синтаксис блоков

1. Размеры блоков должны быть достаточными для того, чтобы включить имя блока.
2. Блоки должны быть прямоугольными, с прямыми углами.
3. Блоки должны быть нарисованы сплошными линиями.

Функциональный блок в IDEFo модели





*В IDEFo различают 5 видов стрелок:
(по аналогии со сторонами блока)*

- ВХОД
- управление
- ВЫХОД
- МЕХАНИЗМ
- (ВЫЗОВ)

Синтаксис стрелок

1. Ломанные стрелки изменяют направление только под углом 90 град.
2. Стрелки должны быть нарисованы сплошными линиями различной толщины.
3. Стрелки могут состоять только из вертикальных или горизонтальных отрезков; отрезки, направленные по диагонали, не допускаются.
4. Концы стрелок должны касаться внешней границы функционального блока, но не должны пересекать ее.
5. Стрелки должны присоединяться к блоку на его сторонах. Присоединение в углах не допускается

ВХОД

- Материал или информация, которые используются и преобразуются работой для получения результата (выхода).

Блок может не иметь ни одной входной дуги.

- Данный вид дуги поступает на левую сторону блока.

УПРАВЛЕНИЕ

- Управление – условия, правила, стратегии, стандарты, которые влияют на выполнение функции.
Каждый блок должен иметь хотя бы одну дугу управления.
- Данный вид дуг поступает на верхнюю сторону блока;

ВЫХОД

- Выход – результат выполнения функции (материал или информация).
Каждая функция должна иметь хотя бы одну выходную дугу.
- Данный вид дуг выходит из правой стороны блока;

МЕХАНИЗМ

- Механизм – ресурсы, с помощью которых выполняется работа.
Это могут быть, например, денежные средства, персонал предприятия, станки.
- Данный вид дуг поступает на нижнюю сторону блока;

ВЫЗОВ

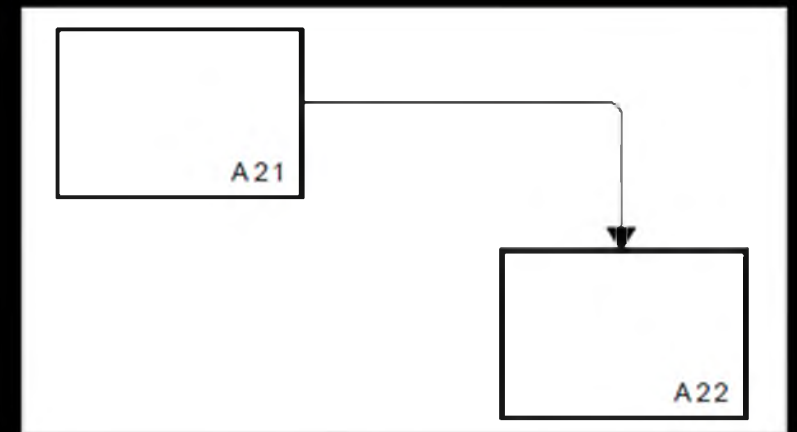
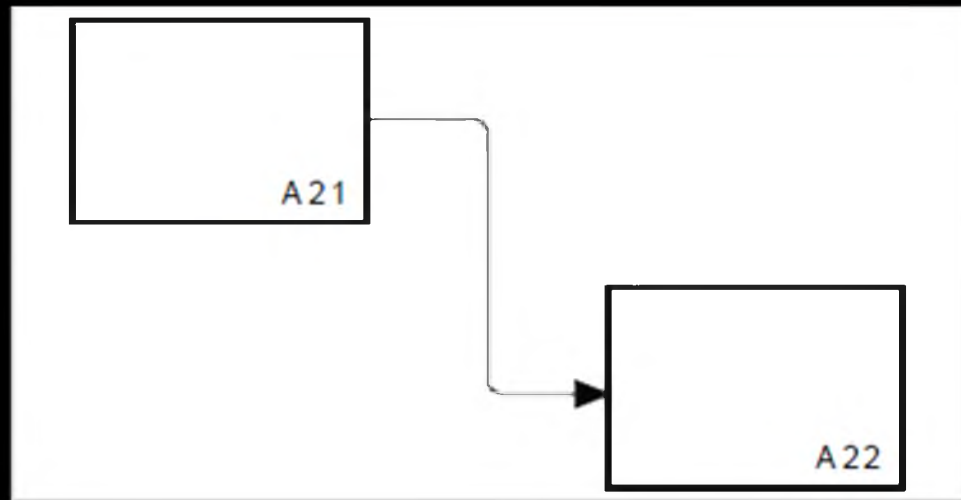
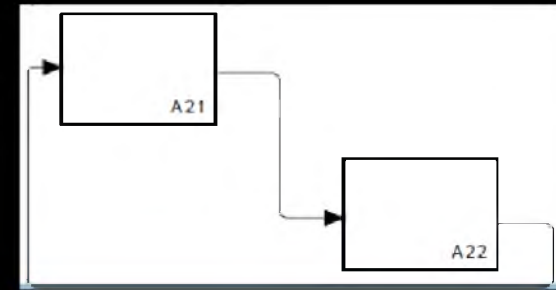
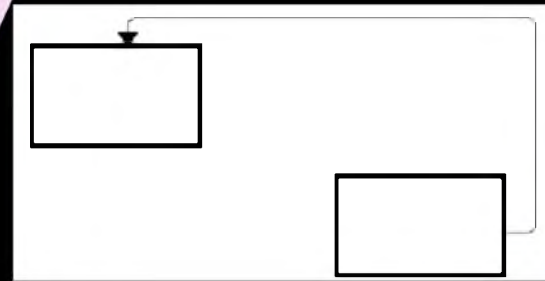
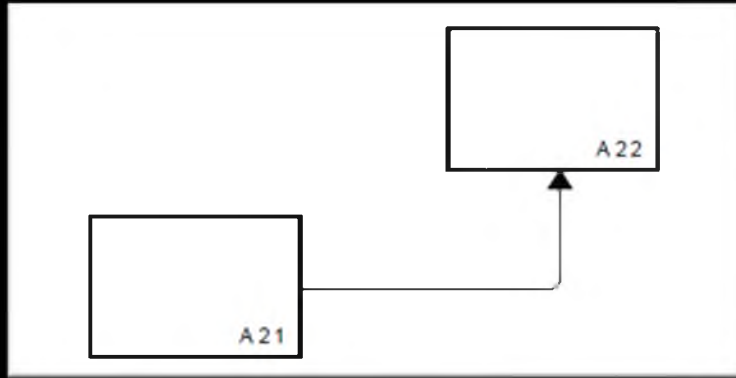
- Специальная дуга, указывающая на другую модель предметной области.

Дуга вызова является расширением IDEF0-методологии и предназначена для организации коллективной работы над моделью, разделения модели на независимые модели и объединения различных моделей предметной области в одну модель.

- Данный вид дуги выходит из нижней стороны блока.

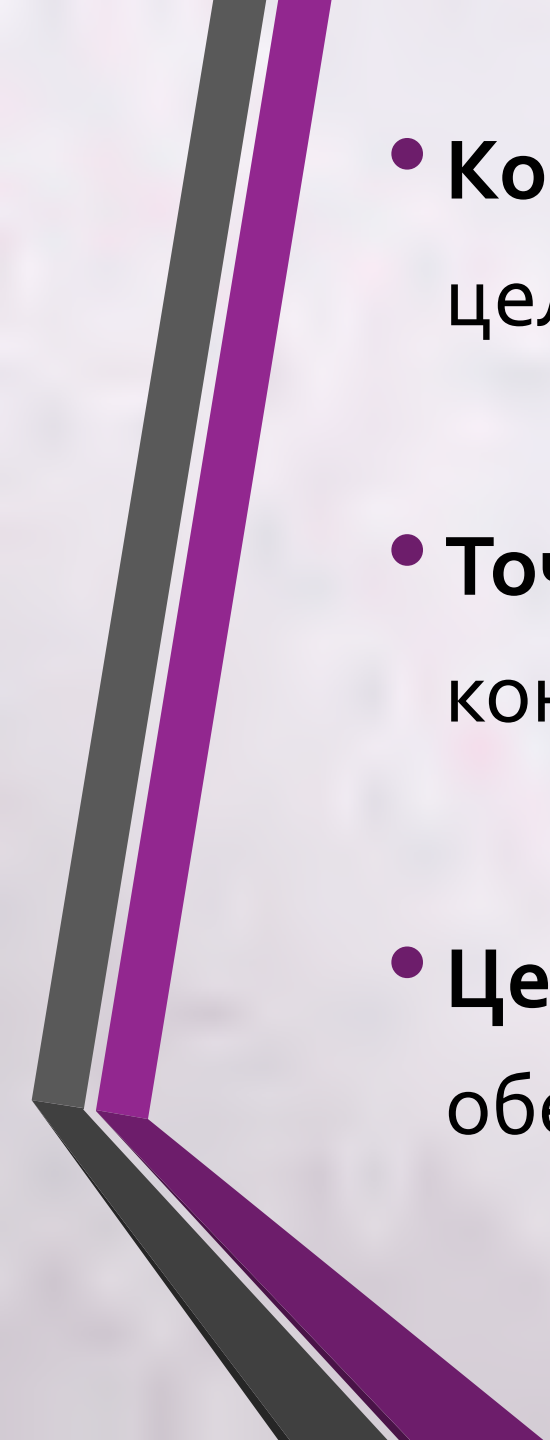
Существует *пять* типов взаимосвязей между блоками


- *Взаимосвязь по управлению* – когда выход одного блока влияет на выполнение функции в другом блоке.
- *Взаимосвязь по входу* – когда выход одного Блока является входом для другого.
- *Обратная связь по управлению* – когда выходы из одной функции влияют на выполнение
- *Обратная связь по входу* – когда выход из одной функции является входом для другой функции, выход которой является для него входом
- *Взаимосвязь "выход-механизм"* – когда выход одной функции является механизмом для другой.



Подготовка к моделированию

На самом раннем этапе моделирования перед началом разработки модели необходимо определить ее направленность: **контекст, точку зрения и цель.**

- 
- **Контекст** определяет объект модели как часть целого.
 - **Точка зрения** специфицирует, что можно "увидеть" в контексте и под каким "углом".
 - **Цель** определяет назначение модели или обеспечиваемых ею взаимодействий.



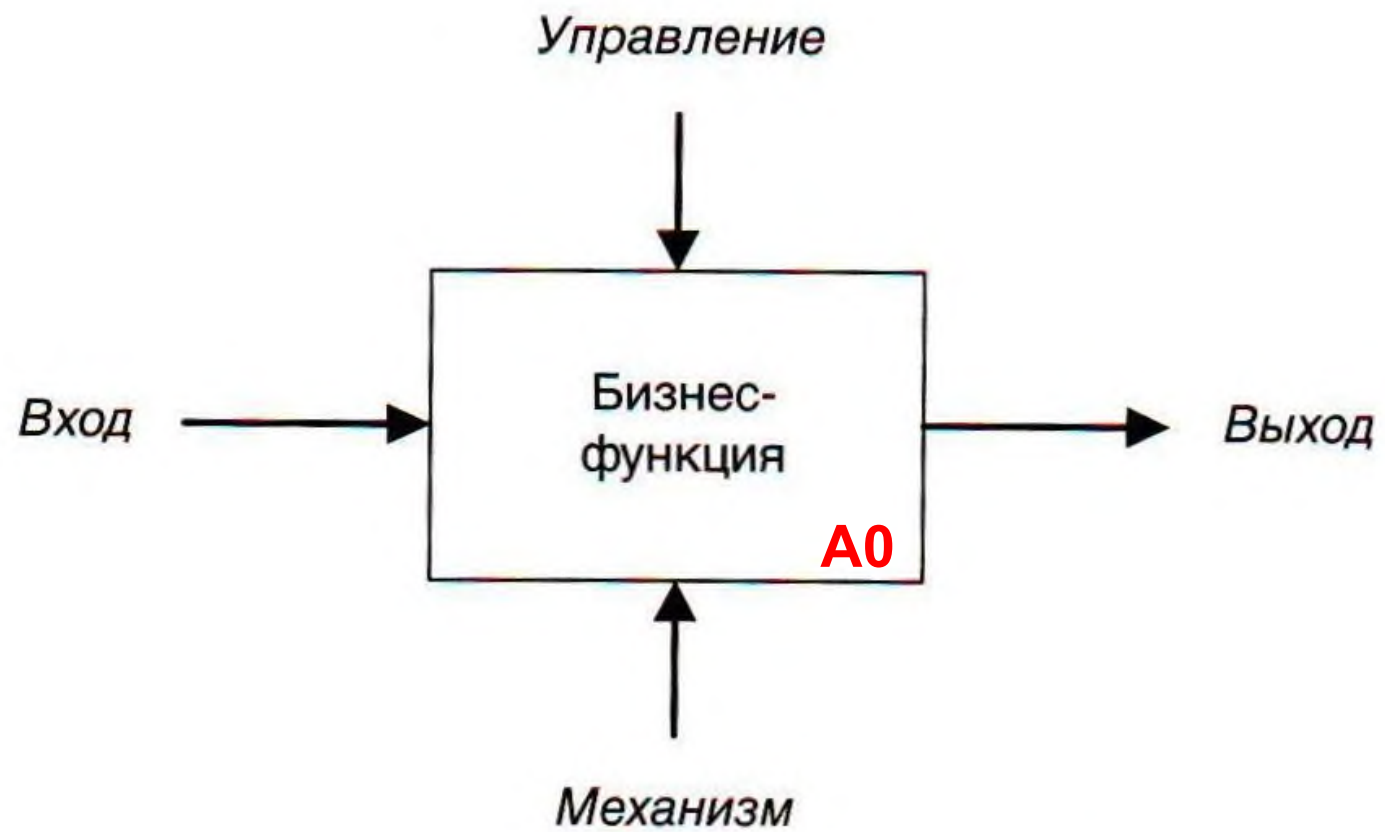
*Каждая модель представляет
только одну точку зрения.*

*Одна модель преследует только одну цель
с определенной точки зрения.*

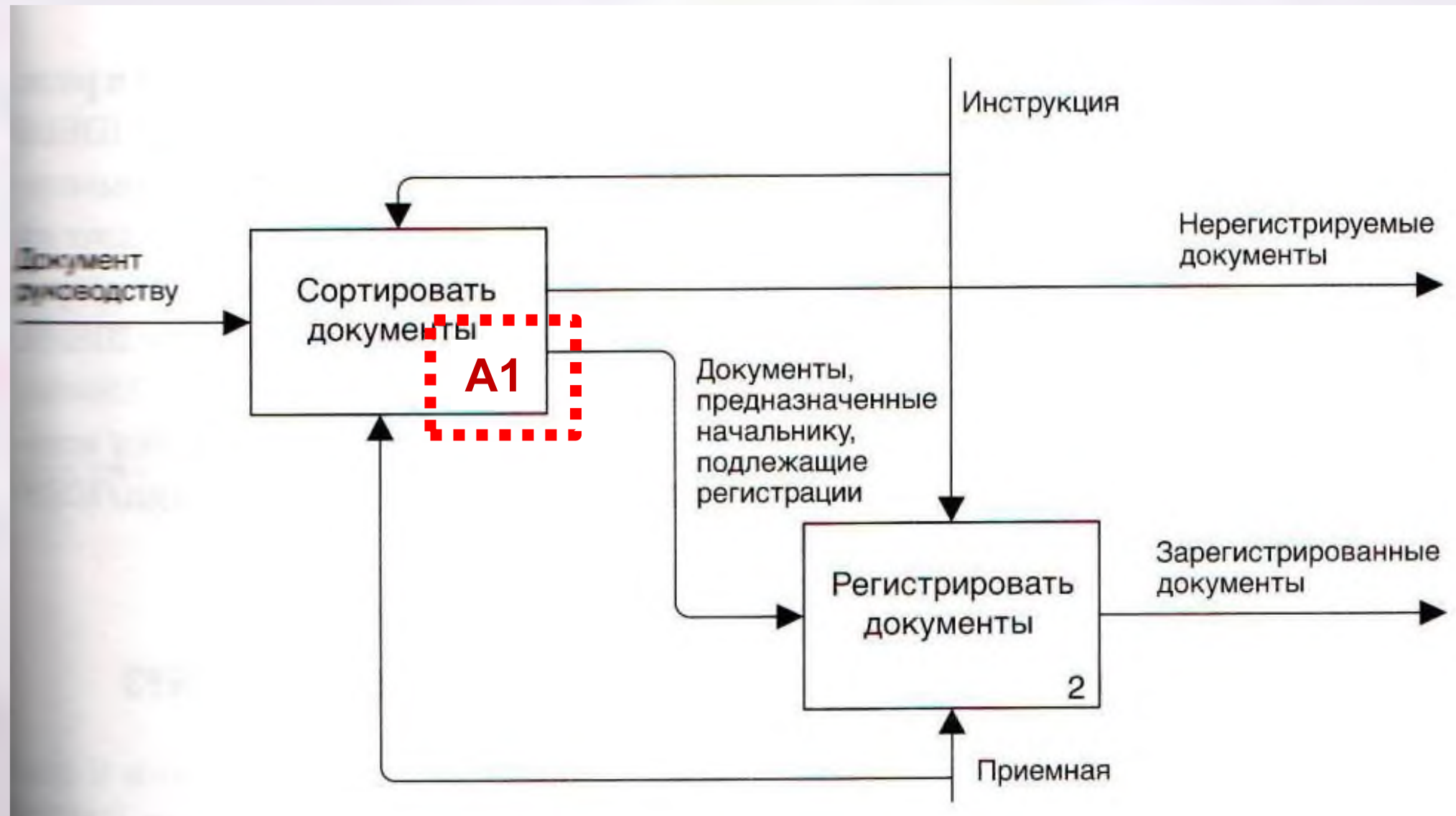
КОНТЕКСТНАЯ ДИАГРАММА

- диаграмма наиболее абстрактного уровня описания системы в целом, содержащей определение субъекта моделирования, цели и точки зрения на модель.

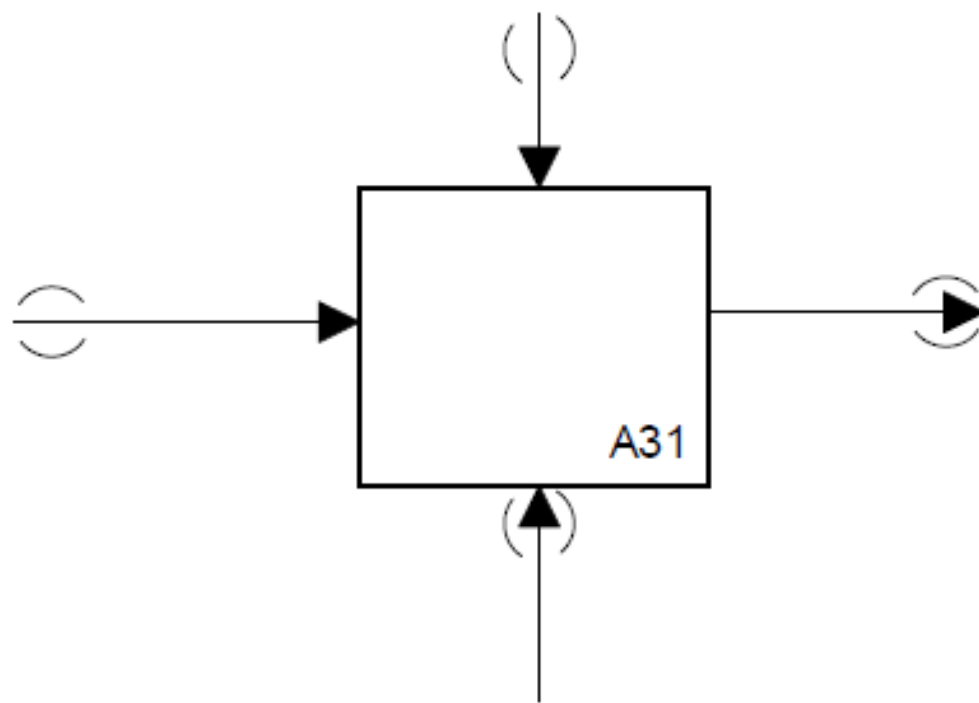
Контекстная (главная) *функция* имеет номер А0.



Иерархия диаграмм



Туннельные дуги – означают, что данные, выраженные этими дугами не рассматриваются на соответствующем уровне детализации



Технология проектирования ИС подразумевает сначала создание модели **AS-IS**, ее анализ и улучшение бизнес-процессов, то есть создание модели **TO-BE**, и только на основе модели **TO-BE** строится модель данных, прототип и затем окончательный вариант ИС.

ИТОГИ:

- методология IDEFo представляет собой четко формализованный подход к созданию функциональных моделей – структурных схем изучаемой системы.
- Схемы строятся по иерархическому принципу с необходимой степенью подробности.
- Совокупность схем (диаграмм) образует модель системы, которая носит качественный, описательный, декларативный характер.

МЕТОДОЛОГИИ СТРУКТУРНОГО АНАЛИЗА И ПРОЕКТИРОВАНИЯ

Семейство IDEFx.

Integration Definition Metodology, или сокращенно IDEF (объединение методологических понятий) – это семейство совместно используемых методов для процесса моделирования. IDEF технология используется, начиная с конца 1980-х годов. Министерство обороны США (Department of Defense USA) является основным пользователем данной технологии. Ей, также, пользуются некоторые крупные корпорации в США.

IDEF0 (Function Modeling) – метод используется для создания функциональной модели, которая является структурированным отображением функций производственной системы или среды, а также информации и объектов, связывающих эти функции.

IDEF1 (Information Modeling) – метод применяется для построения информационной модели, которая представляет собой структурированную информацию, необходимую для поддержки функций производственной системы или среды.

IDEF2 (Simulation Model Design) – данный метод позволяет построить динамическую модель меняющегося во времени поведения функций, информации и ресурсов производственной системы или среды. Данная модель используется редко. В основном востребована на предприятиях, где необходимо описать непрерывную деятельность на конвейерах или аналогичные функции.

IDEF3 (Process Description Capture) – метод используется для сбора информации о состоянии моделируемой системы. Это структурный метод, показывающий причинно-следственные связи и события. Он также показывает, как организована работа, и какие пользователи работают с моделируемой системой.

IDEF3 состоит из двух методов:

- **Process Flow Description (PFD)** – описание процессов, с описанием того, как организована работа между различными элементами моделируемой системы.
- **Object State Transition Description (OSTD)** – описание переходов состояний объектов, с описанием того, какие существуют промежуточные состояния у объектов в моделируемой системе.

IDEF4 (Object-Oriented Design) – метод объектно-ориентированного планирования был разработан для поддержки объектно-ориентированной идеологии.

IDEF5 (Ontology Description Capture) – метод позволяет разрабатывать, изучать и поддерживать онтологию моделируемой системы. Термин «онтология» включает в себя каталог терминов области знаний; правила, объясняющие, как термины могут комбинироваться, создавая при этом корректные ситуации в области знаний и согласованные выводы, используемые в моделируемой системе.

IDEF6 (Design Rational Capture Method) – метод позволяет использовать рациональный опыт проектирования.

IDEF7 (Information System Auditing) – метод описывает проведение методологии аудита информационной системы.

IDEF8 (User Interface Modeling) – метод позволяет разрабатывать необходимые модели Графического Интерфейса Пользователя (Human-System Interaction Design). Метод предназначен для проектирования взаимодействия человека и технической системы.

IDEF9 (Business Constraint Discovery) – модель для анализа имеющихся условий и ограничений (в том числе физических, юридических или любых других) и их влияния на принимаемые решения в процессе реинжиниринга.

IDEF10 - Implementation Architecture Modeling

IDEF11 - Information Artifact Modeling

IDEF12 - Organization Modeling

IDEF13 - Three Schema Mapping Design

IDEF14 (Network Design) – метод позволяет моделировать вычислительные сети. Модель предназначена для представления и анализа данных при проектировании вычислительных сетей на графическом языке с описанием конфигураций, очередей, сетевых компонентов, требований к надежности.