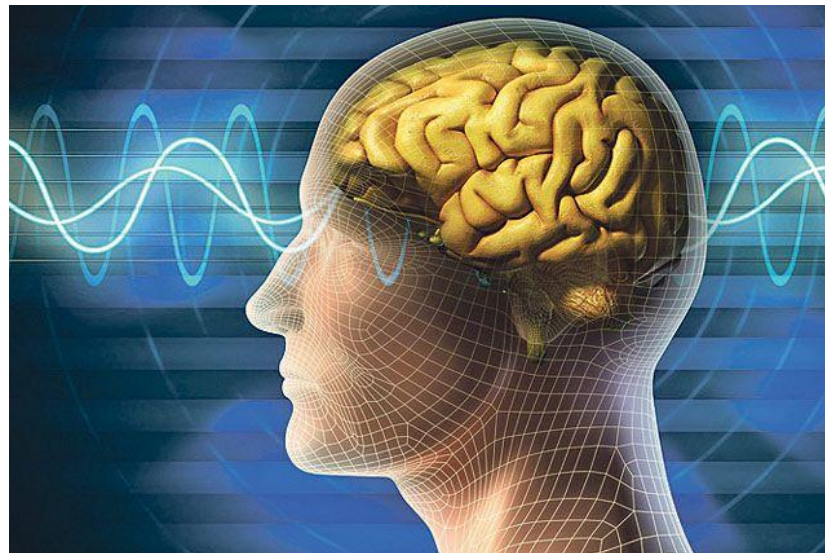


# Лекция № 7

## ФУНКЦИОНАЛЬНОЕ ПРЕДСТАВЛЕНИЕ



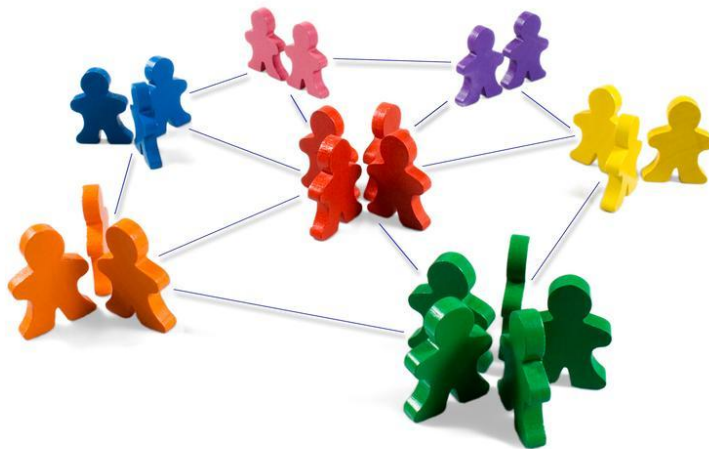
# План лекции

- Способы функционального представления систем
- Временные диаграммы функционирования
- Графические способы функционального представления:
  - дерево функций системы;
  - стандарт функционального моделирования IDEF0.
- Основные понятия IDEF0-методологии
- Принципы ограничения сложности и дисциплина групповой работы в IDEF0-моделях
- Программные продукты, реализующие IDEF0-методологию

# Функциональное представление

Функция смерти заключается в том, чтобы  
сделать жизнь напряжённее.

Джон Роберт Фаулз,  
английский писатель (1926 – 2005)



# Функциональное представление

- Функциональное представление означает рассмотрение функций системы, ее подсистем и отдельных элементов.
- Объектами функционального описания являются системы, которые что-то выполняют или созданы с некоторой целью.
- Функциональное представление иерархично и в его основе обычно лежит функциональная декомпозиция.

# Способы функционального представления

- алгоритмический,
- аналитический,
- посредством временных диаграмм функционирования,
- графический,
- табличный,
- вербальный (словесный).

# Алгоритмический и аналитический способы

- При **алгоритмическом** способе функционирование системы описывается с помощью детерминированного алгоритма. Результат этого алгоритма (или получаемые в процессе работы данные) интерпретируется как основная функция системы (или набор функций).
- При **аналитическом** способе функционирование системы может описываться
  - числовым функционалом, зависящем от функций, описывающих внутренние процессы системы
  - качественным функционалом (упорядочение в терминах «лучше», «хуже», «больше», «меньше» и т.д.)

# Временные диаграммы функционирования системы

- Этот способ обычно используется для технических, в частности человеко-машинных систем. Регламентируется следующий порядок работы:
  - Выбирается некоторый интервал: длительность временного цикла.
  - Определяются параметры функционирования этих компонентов.
  - Составляются диаграммы функционирования их во времени.
  - Уточняются места переработки информации активными компонентами модели, называемыми в дальнейшем процессами системы.
  - Для каждого процесса большой системы определяется состав входной и выходной информации.
  - Уточняется состав управляющих параметров, влияющих на алгоритм функционирования процессов системы.
  - При необходимости выводятся уравнения и расчетные формулы.

# Временные диаграммы функционирования системы

- В рамках каждого временного цикла проводится опрос параметров. Первичная обработка и контроль достоверности опрошенного параметра проводятся сразу же после приема значения параметра, в промежутке между запросом на измерение и приходом замеренного значения следующего параметра.
- Форма такого представления определяется используемым языком формализации. Важно составить схемы информационной связи между процессами системы. Для этой цели на временную диаграмму функционирования каждого процесса следует наложить сведения о местах использования информации, общей для всех процессов.
- Эти диаграммы используются при описании АСУ, атомных реакторов, военных автоматизированных систем и др.



# Необходимая гибкость функционального представления

- Функциональное представление должно соответствовать концепции развития систем во времени и удовлетворять некоторым требованиям:
  - должно быть открытым и допускать возможность расширения (сужения) спектра функций, реализуемых системой;
  - предусматривать возможность перехода от одного уровня рассмотрения к другому, т.е. обеспечивать построение виртуальных моделей систем любого уровня.

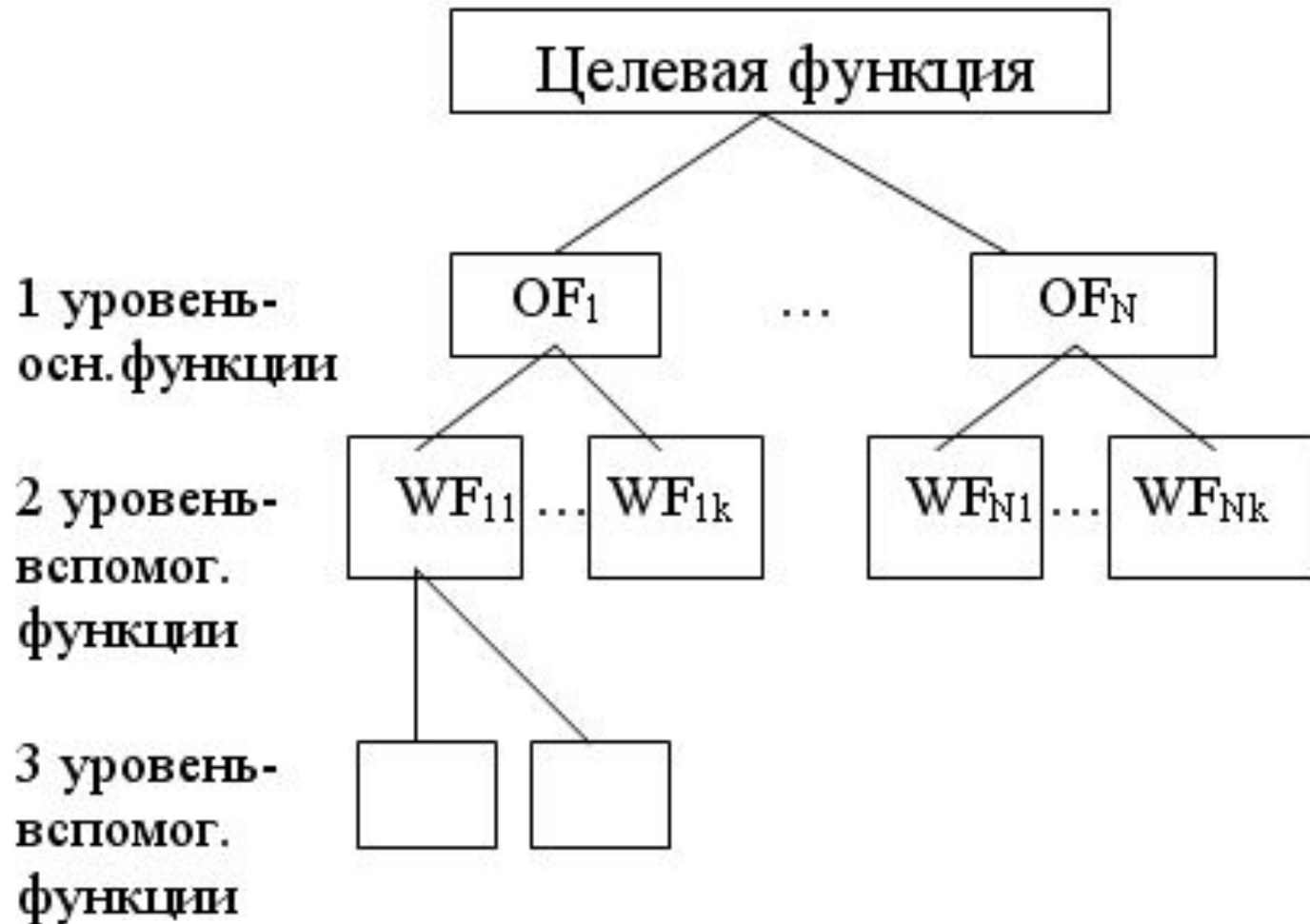
# Графические способы представления

- С точки зрения необходимой гибкости наиболее удобные – графические способы представления.
- Главное преимущество графического способа – его наглядность и легкость иерархических построений.
- При анализе и синтезе технических и социально-экономических систем особенно часто используется графическое представление, разновидностями которого являются:
  - **дерево функций системы,**
  - **стандарт (методология) функционального моделирования IDEFO**

# Дерево функций

- Все функции, реализуемые системой, могут быть условно разделены на три группы: целевая функция; основные функции системы; вспомогательные функции системы.
- **Целевая функция** отражает назначение, сущность и смысл существования системы.
- **Основные функции** представляют собой совокупность макрофункций, которые обеспечивают условия выполнения целевой функции.
- **Вспомогательные функции** расширяют функциональные возможности системы, сферу их применения и способствуют улучшению показателей качества системы. Они обеспечивают условия выполнения основных функций.

# Дерево функций



# Использование дерева функций

- Дерево функций системы представляет декомпозицию функций системы и формируется с целью исследования функциональных возможностей системы.
- Производится анализ совокупности функций, реализуемых на различных уровнях иерархии системы.
- На базе дерева функций системы осуществляется формирование структуры системы на основе функциональных модулей.
- В дальнейшем структура на основе таких модулей покрывается конструктивными модулями (для технических систем) или организационными модулями (для организационно-технических систем).

# Важность построения дерева функций

- Таким образом, этап формирования дерева функций является одним из наиболее ответственных не только при анализе, но и при синтезе структуры системы.
- Ошибки на этом этапе приводят к созданию «систем-инвалидов», не способных к полной функциональной адаптации с другими системами, пользователем и окружающей средой.

# Возможные проблемы с построением дерева функций

- Трудно построить взаимно непересекающееся множество основных функций, а для каждой из основных – множество вспомогательных.
- Трудно заранее учесть все необходимые функции одного уровня и точно «развести» их по отдельным ветвям дерева.
- Иерархическая структура функций может плохо соответствовать реальному положению дел на рассматриваемом объекте и может потребовать слишком глубокой перестройки.

# Методология IDEF0

- **IDEF0** — методология функционального моделирования и графическая нотация, изначально предназначенная для формализации и представления бизнес-процессов.
- Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов.
- В IDEF0 рассматриваются логические отношения между работами. Это отличает IDEF0 от временной последовательности, применяемой в методологии Workflow.
- Методология IDEF0 является одной из самых прогрессивных моделей и используется при организации не только бизнес-проектов, но и других функциональных процессов.



# Основные элементы и понятия IDEF0

В основе методологии IDEF0 лежат четыре основных понятия:

- функциональный блок (Activity Box);
- интерфейсная дуга (Arrow);
- декомпозиция (Decomposition);
- глоссарий (Glossary).

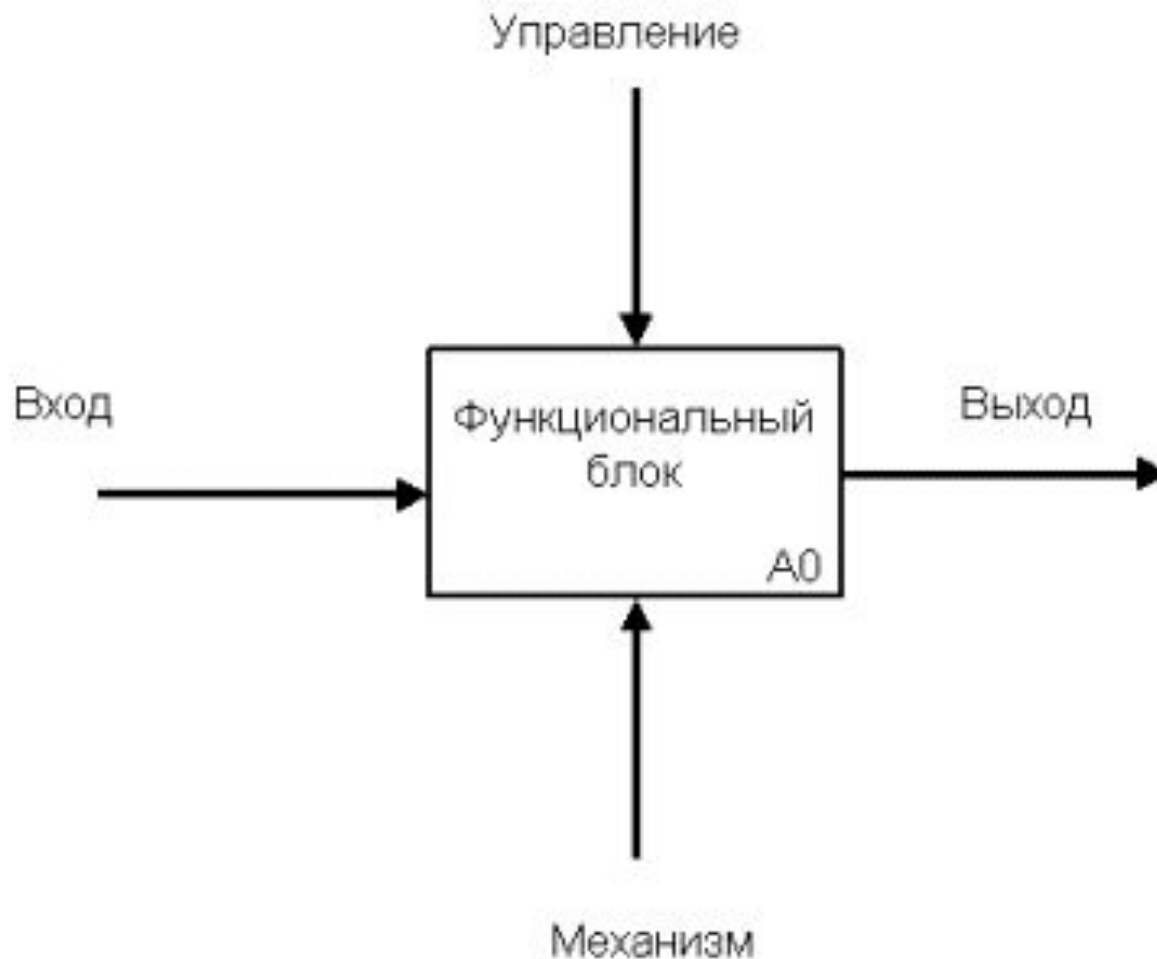
# Функциональный блок

- Функциональный блок графически изображается в виде прямоугольника и олицетворяет собой некоторую конкретную функцию в рамках рассматриваемой системы.
- По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольном наклонении.
- Например: обрабатывать деталь на станке, передать документы в отдел, разработать план-график проведения анализа, опубликовать материалы и др.

# Функциональный блок

- Каждая из четырех сторон функционального блока имеет своё определенное значение:
  - Верхняя сторона имеет значение «Управление» (Control);
  - Левая сторона имеет значение «Вход» (Input);
  - Правая сторона имеет значение «Выход» (Output);
  - Нижняя сторона имеет значение «Механизм» (Mechanism).
- Каждый функциональный блок в рамках единой рассматриваемой системы должен иметь свой уникальный идентификационный номер.

# Функциональный блок



# Интерфейсная дуга

- Интерфейсная дуга отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на соответствующую функцию.
- Графическим отображением интерфейсной дуги является однонаправленная стрелка.
- Каждая интерфейсная дуга должна иметь свое уникальное наименование (Arrow Label).
- По требованию стандарта, наименование интерфейсной дуги должно быть оборотом существительного.

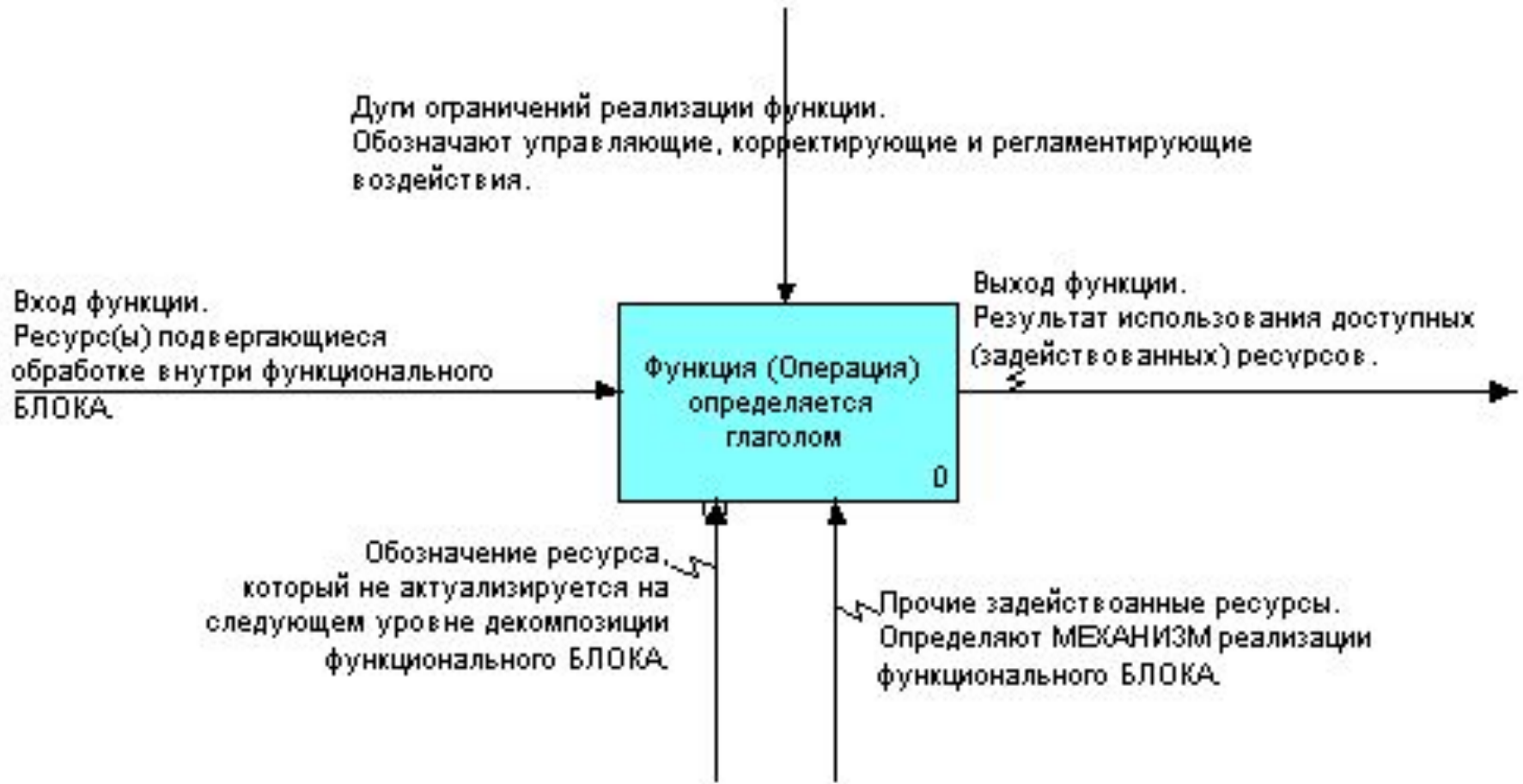
# Интерфейсная дуга

- С помощью интерфейсных дуг отображают различные объекты, в той или иной степени определяющие процессы, происходящие в системе. Такими объектами могут быть элементы реального мира (детали, вагоны, сотрудники и т.д.) или потоки данных и информации (документы, данные, инструкции и т.д.).
- В зависимости от того, к какой из сторон подходит данная интерфейсная дуга, она носит название “входящей”, “исходящей” или “управляющей”. Кроме того, началом и концом каждой функциональной дуги могут быть только функциональные блоки. При этом “источником” может быть только выходная сторона блока, а “приемником” любая из трех оставшихся.

# Интерфейсная дуга

- Любой функциональный блок по требованиям стандарта должен иметь, по крайней мере, одну управляющую интерфейсную дугу и одну исходящую. Это и понятно – каждый процесс должен происходить по каким-то правилам (отображаемым управляющей дугой) и должен выдавать некоторый результат (выходящая дуга), иначе его рассмотрение не имеет никакого смысла.
- За каждой дугой закрепляется специальное замечание, которое отображает суть объекта. Замечание формулируется в виде оборота существительного, отвечающего на вопрос: «Что?».

# Дуги как уточняющие факторы блока





# Пример функционального блока с интерфейсными дугами

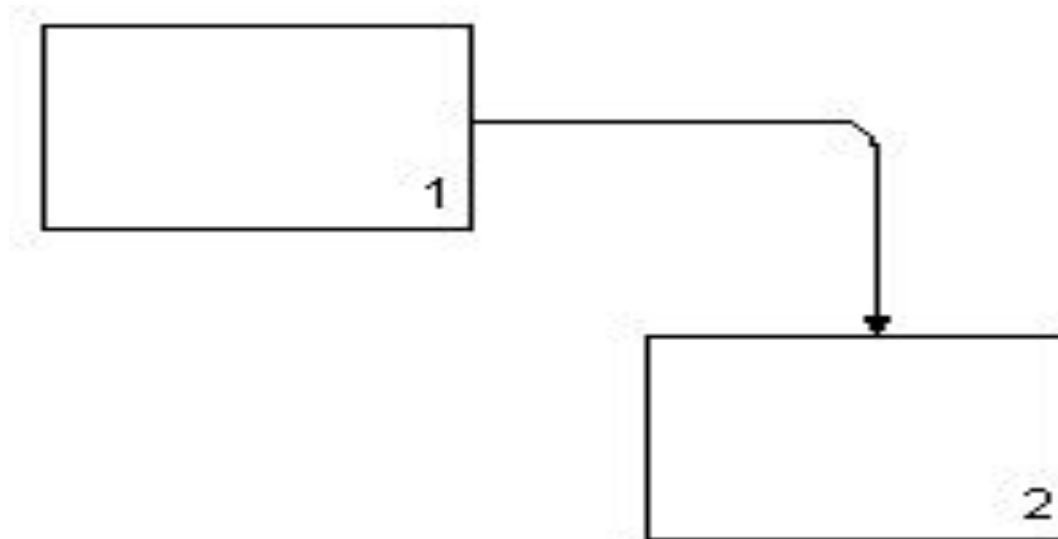


# Типы взаимосвязей между блоками

- При IDEF0 моделировании используются пять типов взаимосвязей между блоками, для представления их отношений.
  - Взаимосвязь по управлению;
  - Взаимосвязь по входу;
  - Обратная связь по управлению;
  - Обратная связь по входу;
  - Взаимосвязь «выход-механизм».

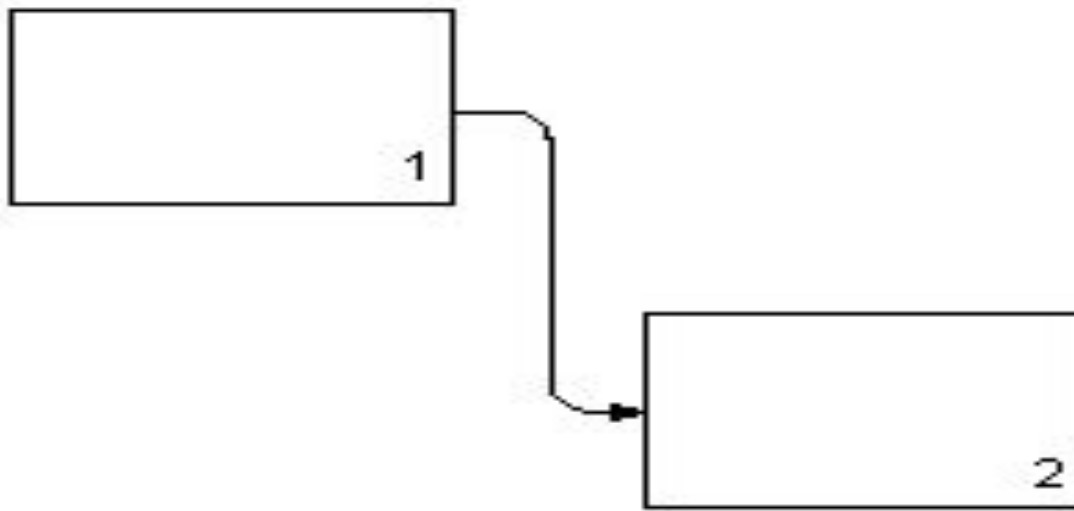
# Взаимосвязь по управлению

- Выход одного Блока влияет на выполнение функции в другом Блоке (разработать инструкцию -> собрать).



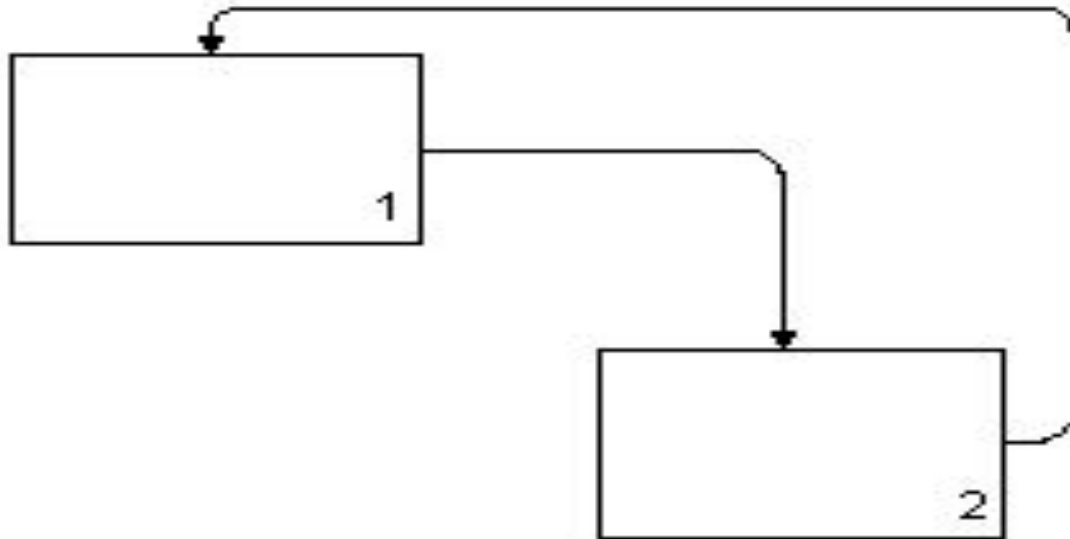
# Взаимосвязь по входу

- Выход одного Блока является входом для другого (изготовить подшипник -> собрать машину).



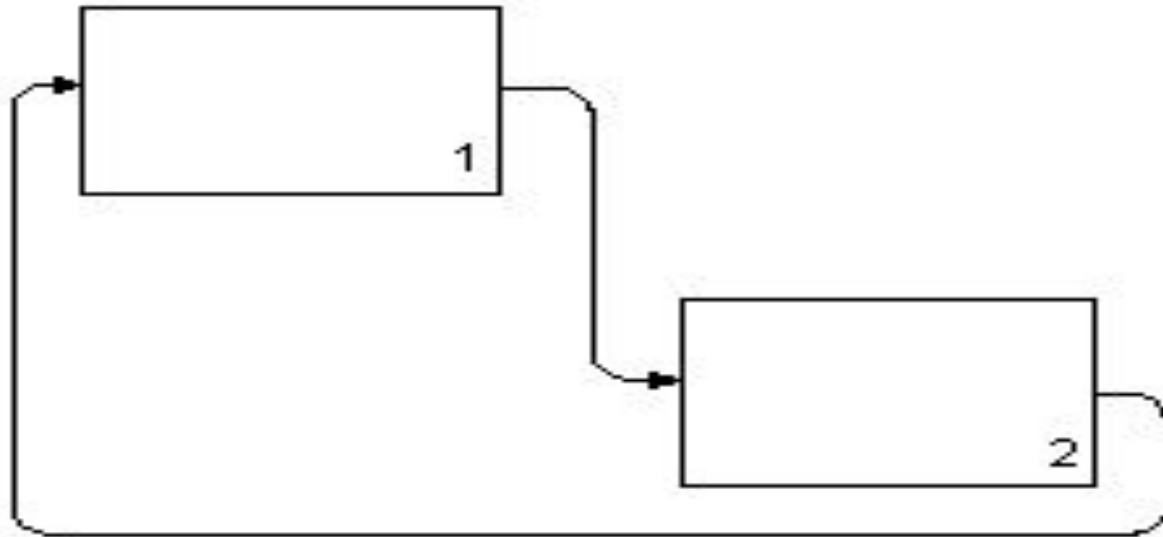
# Обратная связь по управлению

- Выходы из одной функции влияют на выполнение других функций, выполнение которых в свою очередь влияет на выполнение исходной функции (изменение инструкции).



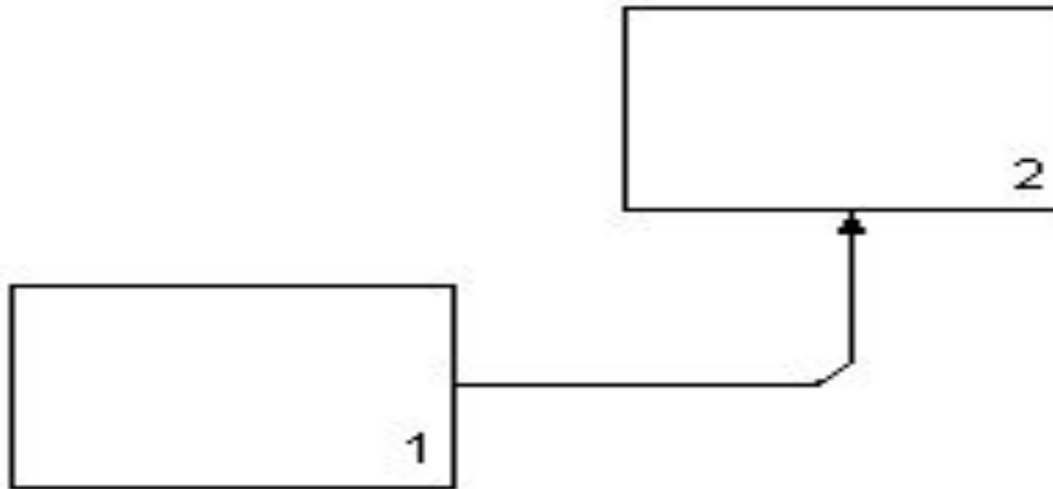
# Обратная связь по входу

- Выход из одной функции является входом для другой функции и наоборот (безотходное производство).



# Взаимосвязь «выход-механизм»

- Выходная дуга одного блока является дугой механизма для другого. Такой тип связи встречается редко и относится чаще всего к подготовительным операциям (когда неясно влияние одного на другое, но оно есть).



# Слияние и разветвление дуг

- Содержание IDEF0-диаграмм уточняется в ходе моделирования постепенно, поэтому дуги на диаграммах редко изображают один объект.
- Чаще всего они отображают определенный набор объектов и могут иметь множество начальных точек (источников) и определенное количество конечных точек (приемников).
- В ходе разработки графической диаграммы для отражения этой особенности используют механизм разветвления/слияния дуг.
- Это позволяет более точно описать из каких наборов объектов состоит входящая в функциональный блок дуга, если она получена путем слияния.



# Декомпозиция

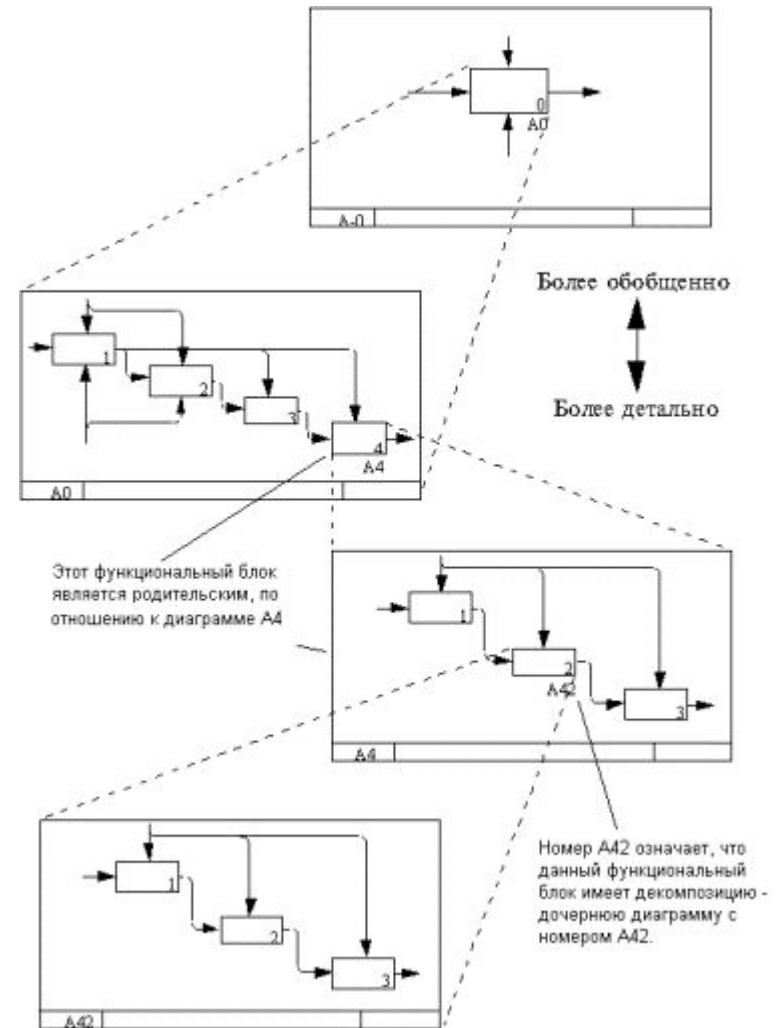
- Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции.
- При этом уровень детализации процесса определяется непосредственно разработчиком модели.
- Декомпозиция позволяет постепенно и структурированно представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко усваиваемой.

# Контекстная диаграмма IDEF0

- Модель IDEF0 всегда начинается с представления системы как единого целого – одного функционального блока с интерфейсными дугами, простирающимися за пределы рассматриваемой области.
- Такая диаграмма с одним функциональным блоком называется контекстной диаграммой, и обозначается идентификатором “А-0”.
- В пояснительном тексте к контекстной диаграмме должна быть указана цель (Purpose) построения диаграммы в виде краткого представления и зафиксирована точка зрения (Viewpoint).

# Декомпозиция в IDEF0

- Функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме.
- Таким образом при декомпозиции возникает множество взаимосвязанных диаграмм.



# Дочерние и родительские диаграммы

- Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы и называется дочерней (Child diagram) по отношению к нему.
- Каждый из функциональных блоков, принадлежащих дочерней диаграмме соответственно называется дочерним блоком (Child Box).
- В свою очередь, функциональный блок – предок называется родительским блоком по отношению к дочерней диаграмме (Parent Box), а диаграмма, к которой он принадлежит – родительской диаграммой (Parent Diagram).

# Целостность и нумерация в IDEF0-модели

- В каждом случае декомпозиции функционального блока все интерфейсные дуги, входящие в данный блок, или исходящие из него фиксируются на дочерней диаграмме. Этим достигается структурная целостность IDEF0-модели.
- Следует обратить внимание на взаимосвязь нумерации функциональных блоков и диаграмм: каждый блок имеет свой уникальный порядковый номер на диаграмме (цифра в правом нижнем углу прямоугольника), а обозначение под правым углом указывает на номер дочерней для этого блока диаграммы. Отсутствие этого обозначения говорит о том, что декомпозиции для данного блока не существует.

# Глоссарий

- Для каждого из элементов IDEF0 (диаграмм, функциональных блоков, интерфейсных дуг) подразумевается создание и поддержание набора соответствующих определений, ключевых слов, повествовательных изложений и т.д., которые характеризуют объект, отображенный данным элементом.
- Этот набор называется глоссарием и является представлением сущности данного элемента.
- Например, для управляющей интерфейсной дуги “распоряжение об оплате” глоссарий может содержать перечень полей соответствующего дуге документа.
- Глоссарий дополняет наглядный графический язык, снабжая диаграммы необходимой дополнительной информацией.

# Разработка IDEF0-диаграмм

- Разработка модели начинается с построения самого верхнего уровня иерархии (A-0) — одного блока и интерфейсных дуг, описывающих внешние связи рассматриваемой системы. Имя функции, записываемое в Блоке 0, является целевой функцией системы с принятой точки зрения и цели построения модели.
- При дальнейшем моделировании Блок 0 декомпозируется на Диаграмме A0, где целевая функция уточняется с помощью нескольких блоков, взаимодействие между которыми описывается с помощью дуг.
- В результате, имена функциональных блоков и интерфейсные дуги, описывающие взаимодействие всех блоков образуют иерархическую взаимосогласованную модель.

# Принципы ограничения сложности IDEF0-диаграмм

- Представление IDEF0 модели построено в виде иерархической пирамиды, в вершине которой представляется самое общее представление системы, а основание представляет собой множество более детальных описаний.
- Обычно IDEF0-модели несут в себе сложную и концентрированную информацию, и для того, чтобы ограничить их перегруженность и сделать удобочитаемыми, в соответствующем стандарте приняты некоторые ограничения сложности.



# Принципы ограничения сложности IDEF0-диаграмм

- Ограничение количества функциональных блоков на диаграмме тремя-шестью. Верхний предел (шесть) заставляет разработчика использовать иерархии при описании сложных предметов, а нижний предел (три) гарантирует, что на соответствующей диаграмме достаточно деталей, чтобы оправдать ее создание.
- Ограничение количества подходящих к одному функциональному блоку (выходящих из одного функционального блока) интерфейсных дуг четырьмя.
- Разумеется, строго следовать этим ограничениям вовсе не обязательно, однако, как показывает опыт, они являются весьма практичными в реальной

# Дисциплина групповой работы над разработкой IDEF0-модели

- Стандарт IDEF0 содержит набор процедур, позволяющих разрабатывать и согласовывать модель большой группой людей, принадлежащих к разным областям деятельности моделируемой системы.
- Обычно процесс разработки является итеративным и состоит из нескольких условных этапов.

# 1. Предварительное создание модели

- Первый этап – предварительное создание модели группой специалистов, относящихся к различным сферам деятельности предприятия.
- Эта группа в терминах IDEF0 называется авторами (Authors).
- Построение первоначальной модели является динамическим процессом, в течение которого авторы опрашивают компетентных лиц о структуре различных процессов.
- На основе имеющихся положений, документов и результатов опросов создается черновик (Model Draft) модели.

## 2. Распространение черновика

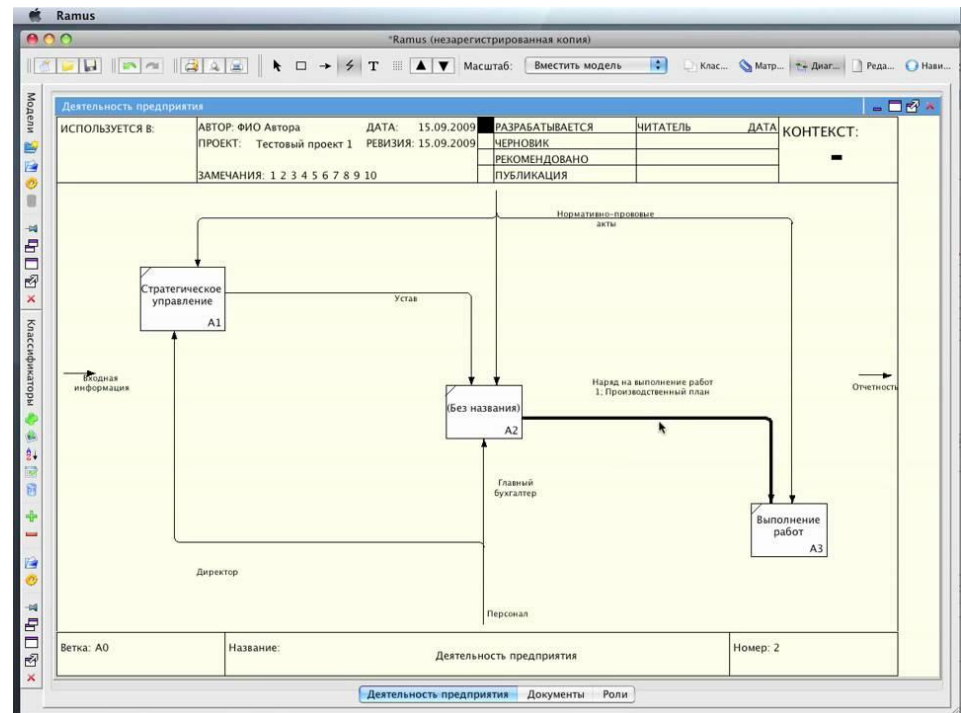
- Второй этап – распространение черновика для рассмотрения, согласований и комментариев.
- На этой стадии происходит обсуждение черновика модели с широким спектром компетентных лиц (в терминах IDEF0 — читателей).
- Каждая из диаграмм черновой модели письменно критикуется и комментируется, а затем передается автору.
- Автор, в свою очередь, также письменно соглашается с критикой или отвергает её с подробной аргументацией и вновь возвращает откорректированный черновик для дальнейшего рассмотрения.
- Этот цикл продолжается до тех пор, пока авторы и читатели не придут к единому мнению.

# 3. Официальное утверждение модели

- Утверждение согласованной модели происходит руководителем рабочей группы в том случае, если у авторов модели и читателей отсутствуют разногласия по поводу ее адекватности.
- Окончательная модель представляет собой согласованное представление о предприятии (системе) с заданной точки зрения и для заданной цели.
- Наглядность графического языка IDEF0 делает модель вполне читаемой и для лиц, которые не принимали участия в проекте ее создания, а также эффективной для проведения показов и презентаций.
- В дальнейшем, на базе построенной модели могут быть организованы новые проекты, нацеленные на производство изменений на предприятии (в

# Программные продукты, реализующие методологию IDEF0

- [CA AllFusion Process Modeler](#)  
(ранее [BPwin](#))
- [Business Studio](#)
- [Ramus](#)
- [MS Visio](#)
- [Corel iGrafx \(IDEF0\)](#)



# Дальнейшая программа

- Информационное представление
- Процессуальное представление и синергетика

