

Методология функционального моделирования DFD

DFD-модель

DFD - Data Flow Diagrams - диаграммы потоков данных

Диаграмма наглядно изображает разные виды компонентов ИС, а также обмен информацией между ними и с внешней средой.

Модель системы определяется как иерархия диаграмм потоков данных, описывающих асинхронный процесс преобразования информации от ее входа в систему до выдачи пользователю.



Главная цель такого представления – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.



DFD-модели могут быть использованы в дополнение к модели IDEFO для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации.



Диаграммы потоков данных являются основным средством моделирования функциональных требований к проектируемой системе.

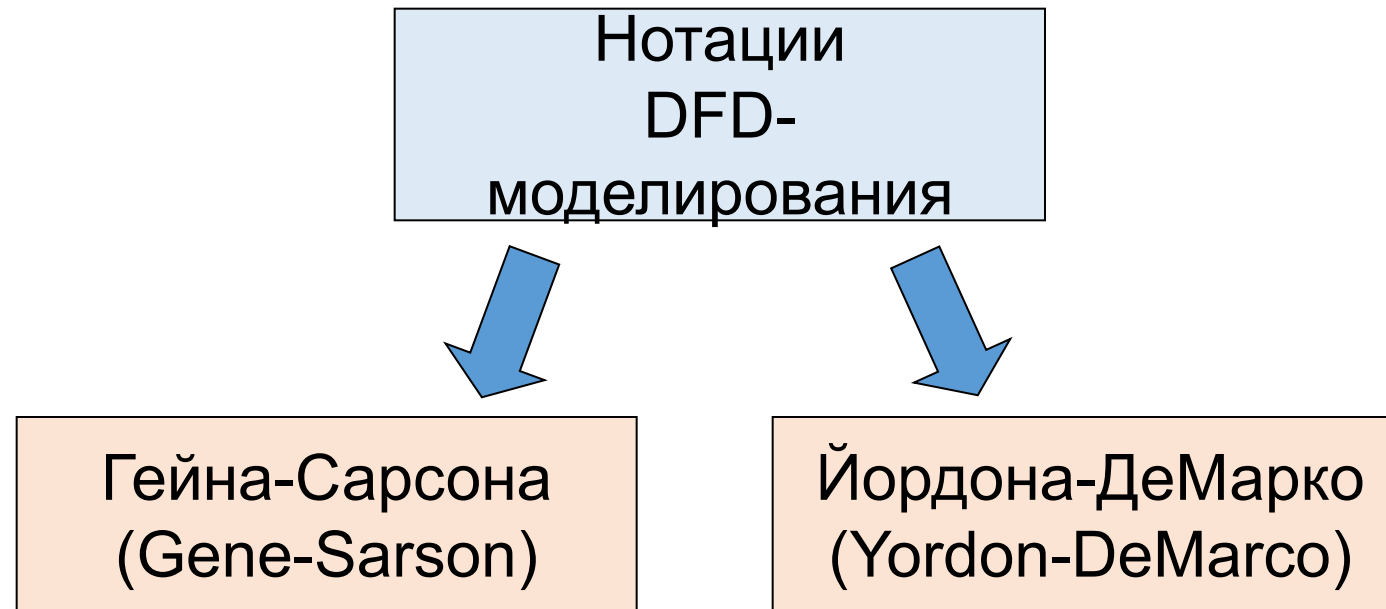
Преимущества методики DFD

- ✓ возможность однозначно определить внешние сущности, анализируя потоки информации внутри и вне системы
- ✓ возможность проектирования сверху вниз, что облегчает построение модели "как должно быть"
- ✓ наличие спецификаций процессов нижнего уровня, что позволяет преодолеть логическую незавершенность функциональной модели и построить полную функциональную спецификацию разрабатываемой системы.

Основные компоненты диаграмм потоков данных

- ✓ внешние сущности
- ✓ системы и подсистемы
- ✓ процессы
- ✓ накопители данных
- ✓ потоки данных.

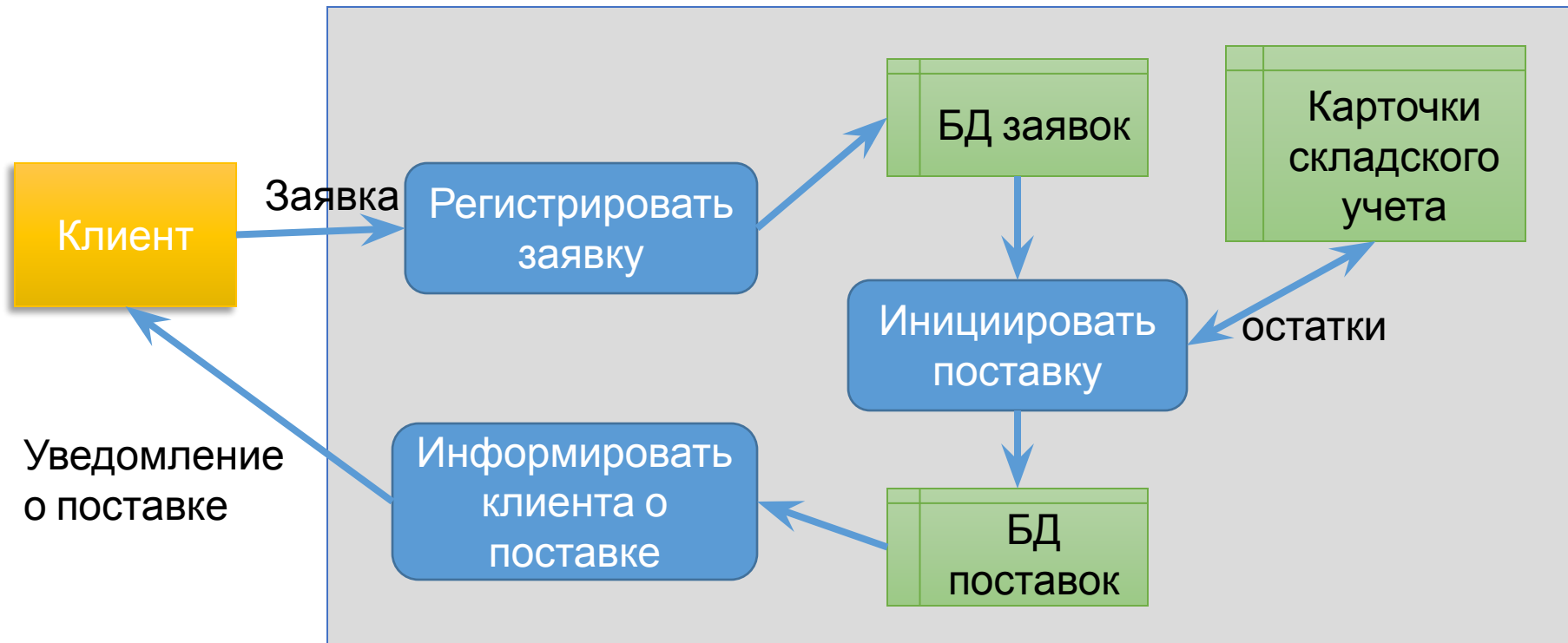
Нотации, используемые в DFD-моделировании



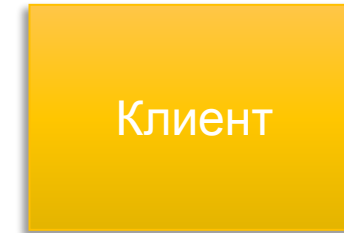
В зависимости от используемой нотации графическое представление элементов диаграмм будет различным.

Пример диаграммы

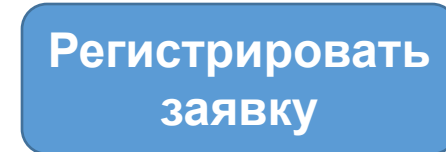
ИС принимает извне потоки данных, преобразует их с помощью процессов обработки, порождая новые потоки, которые могут поступать на вход к другим процессам, сохраняться в накопителях и передаваться из системы во внешнюю среду



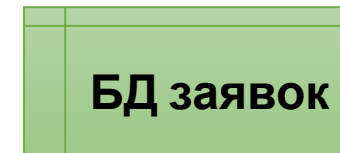
□ **Компоненты диаграммы**
□ **Внешний объект** – это предмет, организация или лицо, являющийся приемником или источником информации и не входящий в состав системы



□ **Процесс** – это некоторая последовательность операций обработки данных, которая преобразует входной поток данных в выходной в соответствии с заданным алгоритмом



□ **Накопитель данных** – это некоторое место или устройство для хранения информации



□ **Поток данных** – это информация, передаваемая от одного блока диаграммы к другому



Внешняя сущность

- ☑ Представляет собой материальный объект или физическое лицо, являющееся источником или приемником информации (например, заказчики, клиенты, поставщики, склад, персонал, банк).
- ☑ Внешняя сущность находится за пределами границ анализируемой системы.
- ☑ Одна и та же внешняя сущность может быть использована многократно на одной или нескольких диаграммах.

Имя

Внешняя сущность в
нотации Йордона-
ДеМарко

Внешняя сущность в
нотации Гейна-Сарсона

Система и подсистема

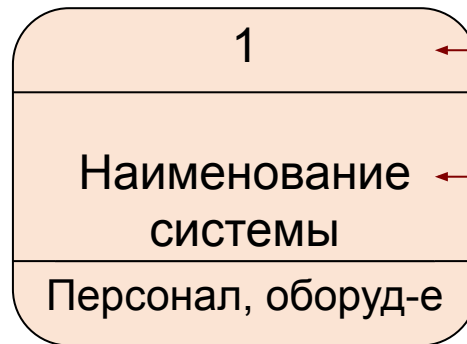


При построении модели сложной системы она может быть представлена в самом общем виде на контекстной диаграмме в виде одной системы, либо в виде ряда подсистем.



Наименование системы/подсистемы представляется в виде **словосочетания с отглагольным существительным** (рассмотрение повестки дня, решение задачи, получение денег и т.п.).

Система/подсистема
в нотации Гейна-
Сарсона



← Поле идентификации

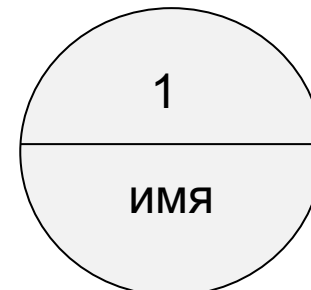
← Поле имени

← Поле физической реализации

Система/подсистема в
нотации Йордона-
ДеМарко



или



Процесс



Представляет собой преобразование
ВХОДНЫХ ПОТОКОВ В ВЫХОДНЫЕ В
соответствии с определенным алгоритмом.

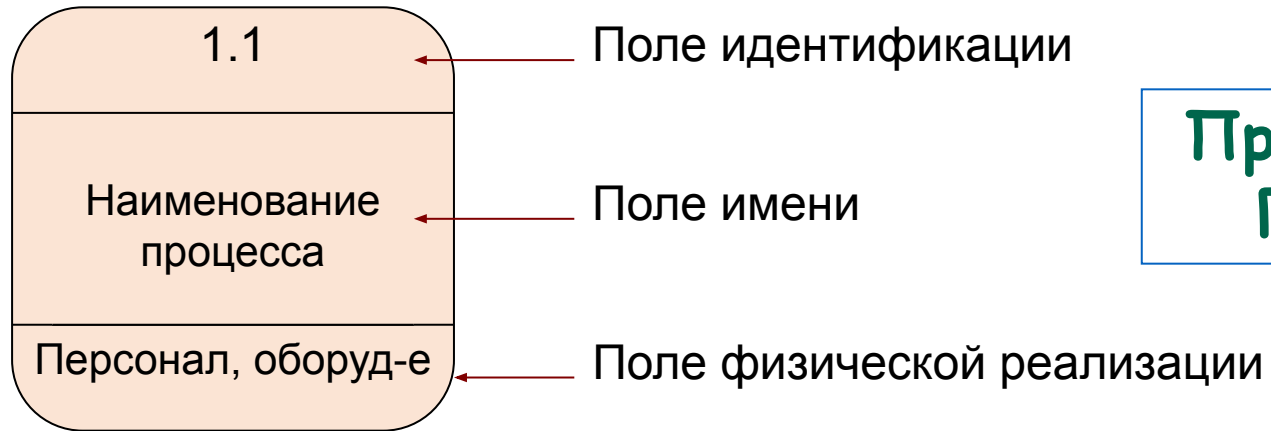


Примеры: обработка входных документов и
выпуск отчетности определенным
подразделением, процессы физически
реализованного устройства.

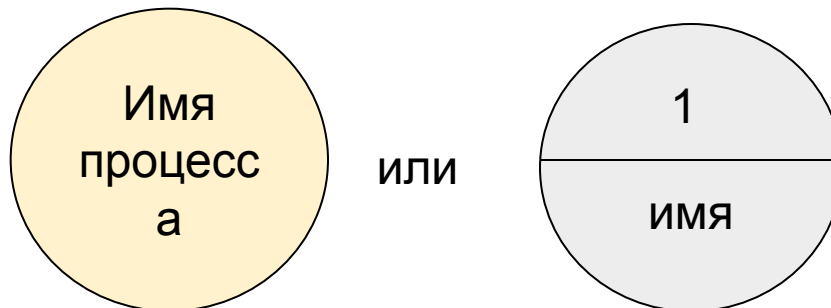


Процесс именуется в виде словосочетания
с активным глаголом в неопределенной
форме, за которым следует
существительное в винительном падеже.

Процесс



Процесс в нотации
Гейна-Сарсона



Процесс в нотации
Йордона-ДеМарко

Процесс отличается от системы/подсистемы по
полю наименования

Накопитель данных



Это абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь.

5	Имя
---	-----

3	Список заказов
---	----------------



Примеры: ящик в картотеке, таблицы в ОЗУ, файл на электронном носителе

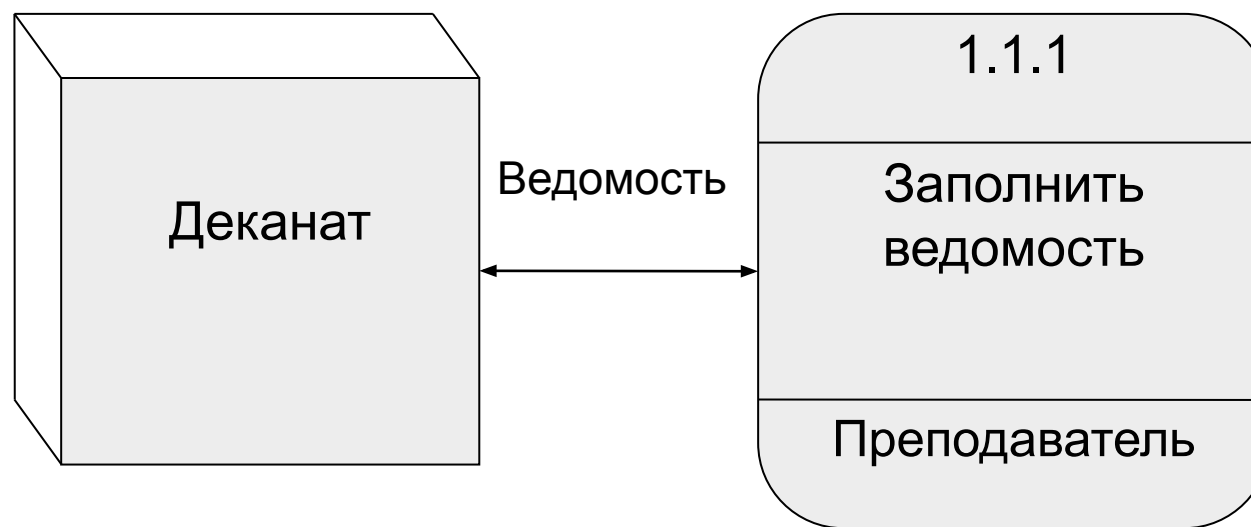


Примечание: В нотациях Гейна-Сарсона и Йордона-ДеМарко графическое представление данного элемента аналогичное

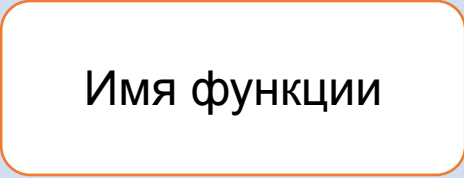
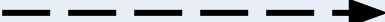
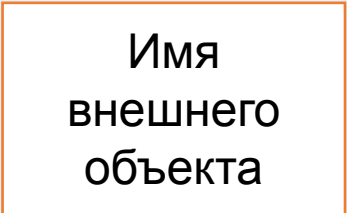
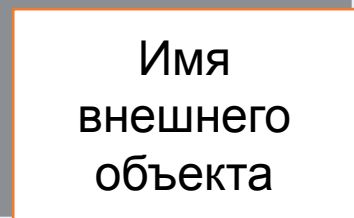
Поток данных



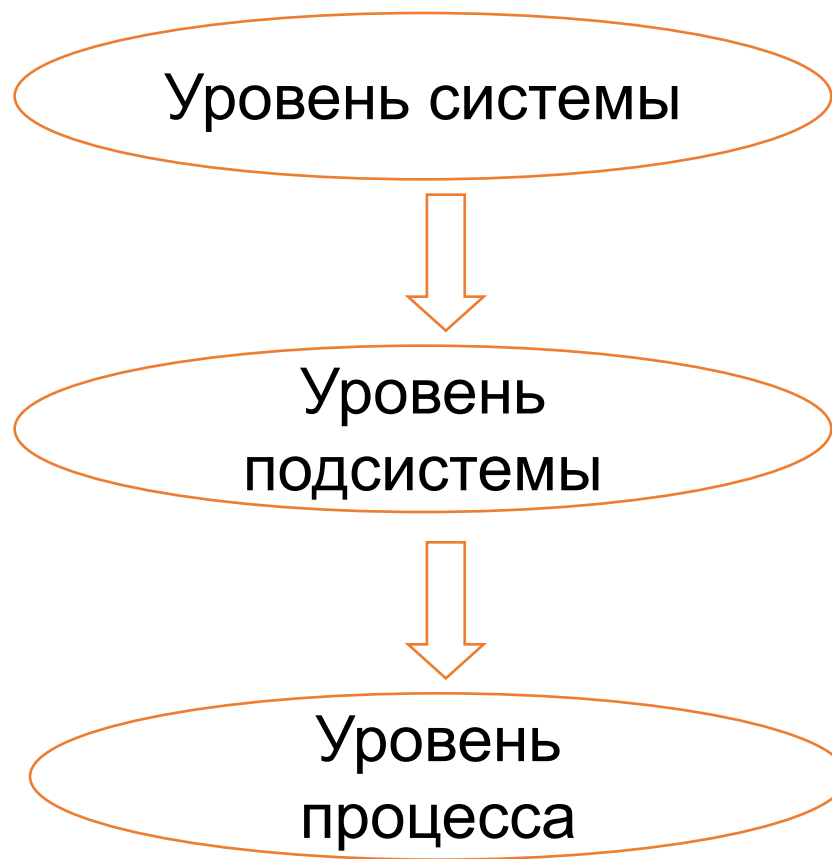
Определяет информацию, передаваемую через некоторые соединения от источника к приемнику. Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами и т.п.



Нотации, используемые в DFD-моделировании

Элемент	Описание	Нотация Йордана-ДеМарко	Нотация Гейна-Сарсона
Функция	Работа	 Имя функции	 Имя функции
Поток данных	Объект, над которым выполняется работа. Может быть логическим или управляющим. (Управляющие потоки обозначаются пунктирной линией со стрелкой)	 	 Понятие управляющего потока отсутствует
Хранилище данных	Структура для хранения информационных объектов	 Имя	 Имя
Внешняя сущность	Внешний по отношению к системе объект, обменивающийся с ней потоками.	 Имя внешнего объекта	 Имя внешнего объекта

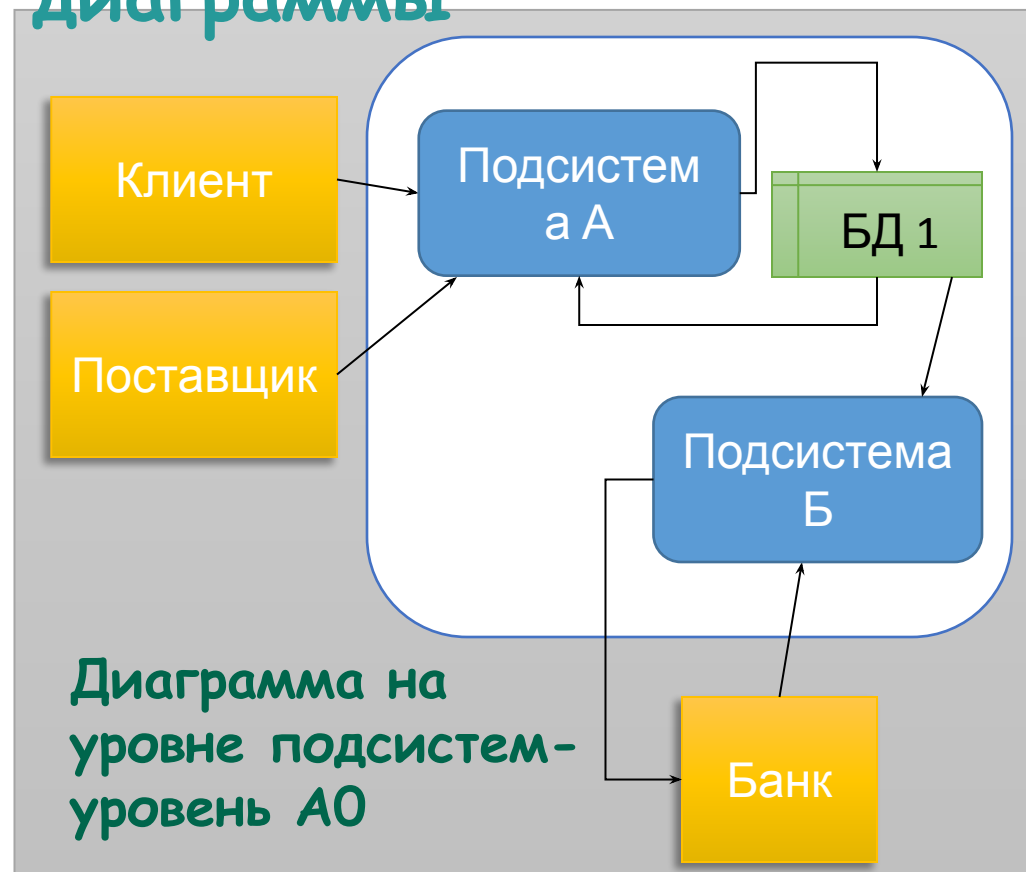
Уровни DFD-модели





Иерархия диаграмм

Сложные системы изображаются на нескольких иерархически связанных диаграммах: каждая диаграмма соответствует одному из уровней функциональной диаграммы



Правила декомпозиции



Уровни диаграммы соответствуют уровням иерархической функциональной диаграммы



Уровни нумеруются, блоки имеют номера в соответствии с уровнем, например: А0 (контекстная), А1, А2 (1 уровень), А1.1, А3.5 (второй уровень), А1.1.3 (третий уровень)



Каждая диаграмма изображается на отдельном листе, между листом и родительским блоком устанавливается гиперссылка

Варианты диаграмм



Диаграмма «**as is**» показывает систему обработки информации в текущий момент времени (функции пользователей, отделов, ручные операции, имеющиеся программные средства). Служит для анализа и выводов о «**слабых местах**» в системе



Диаграмма «**to be**» моделирует будущую ИС . Является частью программной документации.

Построение DFD можно свести к следующим шагам:

- ☑ Идентификация внешних объектов (по отношению к системе)
- ☑ Идентификация информации, которая передается между процессами
- ☑ Разработка контекстной диаграммы
- ☑ Контроль контекстной диаграммы и уточнение, если это нужно
- ☑ Формирование DFD первого уровня, где отражены основные функции системы
- ☑ Дальнейшая декомпозиция каждого процесса до тех пор, пока процесс самого нижнего уровня можно будет представить в виде некоторой спецификации (алгоритма)
- ☑ Ревизия всех уровней с целью выяснения некорректности, устранение обнаруженных некорректностей.

Проверка полноты и непротиворечивости

- ☑ Все процессы должны быть описаны
- ☑ Не допускается обрыва стрелок
- ☑ Не допускаются потоки между внешним объектом и накопителем
- ☑ Не допускаются процессы и накопители только с входящими стрелками («черная дыра»)
- ☑ Не допускаются процессы и накопители только с выходящими потоками («генератор»)

Построение иерархии DFD

1. Построение диаграмм уровня системы и подсистемы

Построение иерархии DFD

2. Построение диаграмм уровня процесса