



# DFD- МЕТОДОЛОГИЯ

Выполнил:  
Студент группы 606-42, 4 курса  
Кафедры АСОИУ  
Чеботков  
Даниил  
Валерьевич

# Назначение и состав DFD

**DFD** (*data flow diagrams*) — диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

Как и в IDEF0, основу методологии DFD составляет графический язык описания процессов. Авторами одной из первых графических нотаций DFD (1979 г.) стали Эд Йордан (Yourdon) и Том де Марко (DeMarko).

В настоящее время наиболее распространенной является нотация Гейна-Сарсона (Gane-Sarson).

# Назначение и состав DFD (продолжение)

Модель системы в нотации DFD представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе. Модель системы содержит контекстную диаграмму и диаграммы декомпозиции.

Принципы построения функциональной модели с помощью DFD аналогичны принципам методологии IDEF0. Вначале строится контекстная диаграмма, где отображаются связи системы с внешним окружением. В дальнейшем выполняется декомпозиция основных процессов и подсистем с построением иерархии диаграмм.

# Элементы графической нотации DFD

Согласно DFD источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или процессам. Те в свою очередь преобразуют информацию и порождают новые потоки, которые переносят информацию к другим процессам или подсистемам, накопителям данных или внешним сущностям – потребителям информации.

Далее в примерах будет использоваться нотация Гейна-Сарсона.

# Элементы графической нотации DFD

Наименование	Нотация Йордана	Нотация Гейна-Сарсона
Поток данных		
Процесс (система, подсистема)		
Накопитель данных		
Внешняя сущность		

Таблица 1. Элементы графической нотации DFD

# Поток данных

*Поток данных* определяет информацию (материальный объект), передаваемую через некоторое соединение от источника к приемнику. Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными лентами или дискетами, переносимыми с одного компьютера на другой и т. д.

Каждый поток данных имеет имя, отражающее его содержание. Направление стрелки показывает направление потока данных. Иногда информация может двигаться в одном направлении, обрабатываться и возвращаться назад в ее источник. Такая ситуация может моделироваться либо двумя различными потоками, либо одним – двунаправленным.

На диаграммах IDEF0 потоки данных соответствуют входам и выходам, но в отличие от IDEF0 стрелки потоков на DFD могут отображаться входящими и выходящими из любой грани внешней сущности, процесса или накопителя данных.

# Процесс

**Процесс** (в IDEF0 – функция, работа) представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом.

Каждый процесс должен иметь имя в виде предложения с глаголом в неопределенной форме (вычислить, рассчитать, проверить, определить, создать, получить), за которым следуют существительные в винительном падеже, например:

- «Ввести сведения о клиентах»;
- «Рассчитать допустимую скорость»;
- «Сформировать ведомость допустимых скоростей»

Преобразование информации может показываться как с точки зрения процессов, так и с точки зрения **систем** и **подсистем**. Если вместо имени процесса «Рассчитать допустимую скорость» написать «Подсистема расчета допустимых скоростей», тогда этот блок на диаграмме стоит рассматривать, как подсистему.

# Накопитель (хранилище) данных

*Накопитель (хранилище) данных* представляет собой абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.

Накопитель данных может быть реализован физически в виде ящика в картотеке, области в оперативной памяти, файла на магнитном носителе и т.д.

Накопителю обязательно должно даваться уникальное имя и номер в пределах всей модели (всего набора диаграмм). Имя накопителя выбирается из соображения наибольшей информативности для разработчика. Например, если в качестве накопителей выступают таблицы проектируемой базы данных, тогда в качестве имен накопителей рекомендуется использовать имена таблиц. Таким образом, накопитель данных может представлять собой всю базу данных целиком, совокупность таблиц или отдельную таблицу. Такое представление накопителей в дальнейшем облегчит построение информационной модели системы.

# Внешняя сущность

**Внешняя сущность** (терминатор) представляет собой материальный объект или физическое лицо, выступающие как источник или приемник информации (например, заказчики, персонал, программа, склад, инструкция).

Определение некоторого объекта, субъекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ проектируемой информационной системы. В связи с этим внешние сущности, как правило, отображаются только на контекстной диаграмме DFD. В процессе анализа и проектирования некоторые внешние сущности могут быть перенесены на диаграммы декомпозиции, если это необходимо, или, наоборот, часть процессов (подсистем) может быть представлена как внешняя сущность.

# Пример построения модели DFD

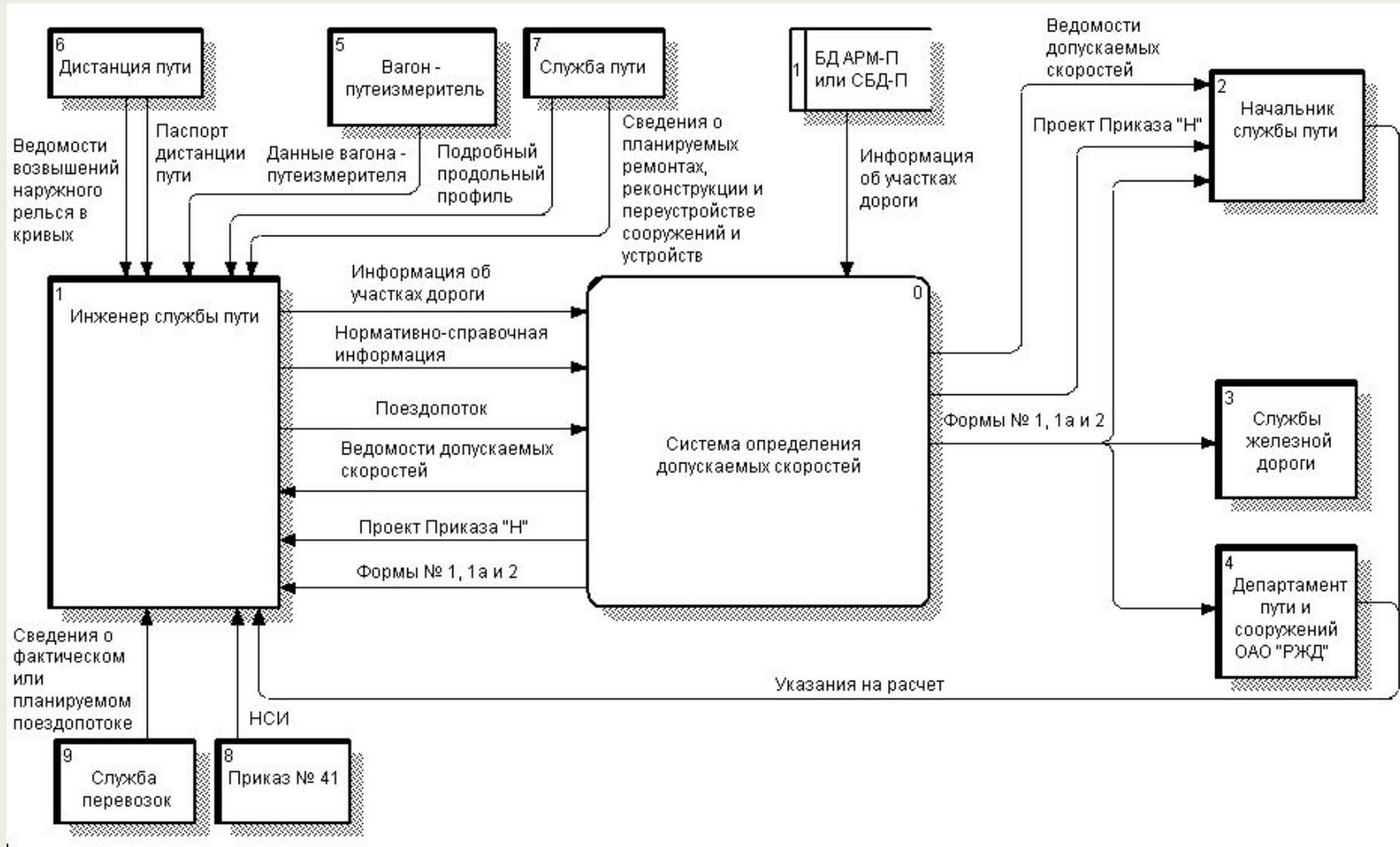


Рис. 1. DFD для системы определения допускаемых скоростей 10

# Пример нотации Йордона - Де Марко

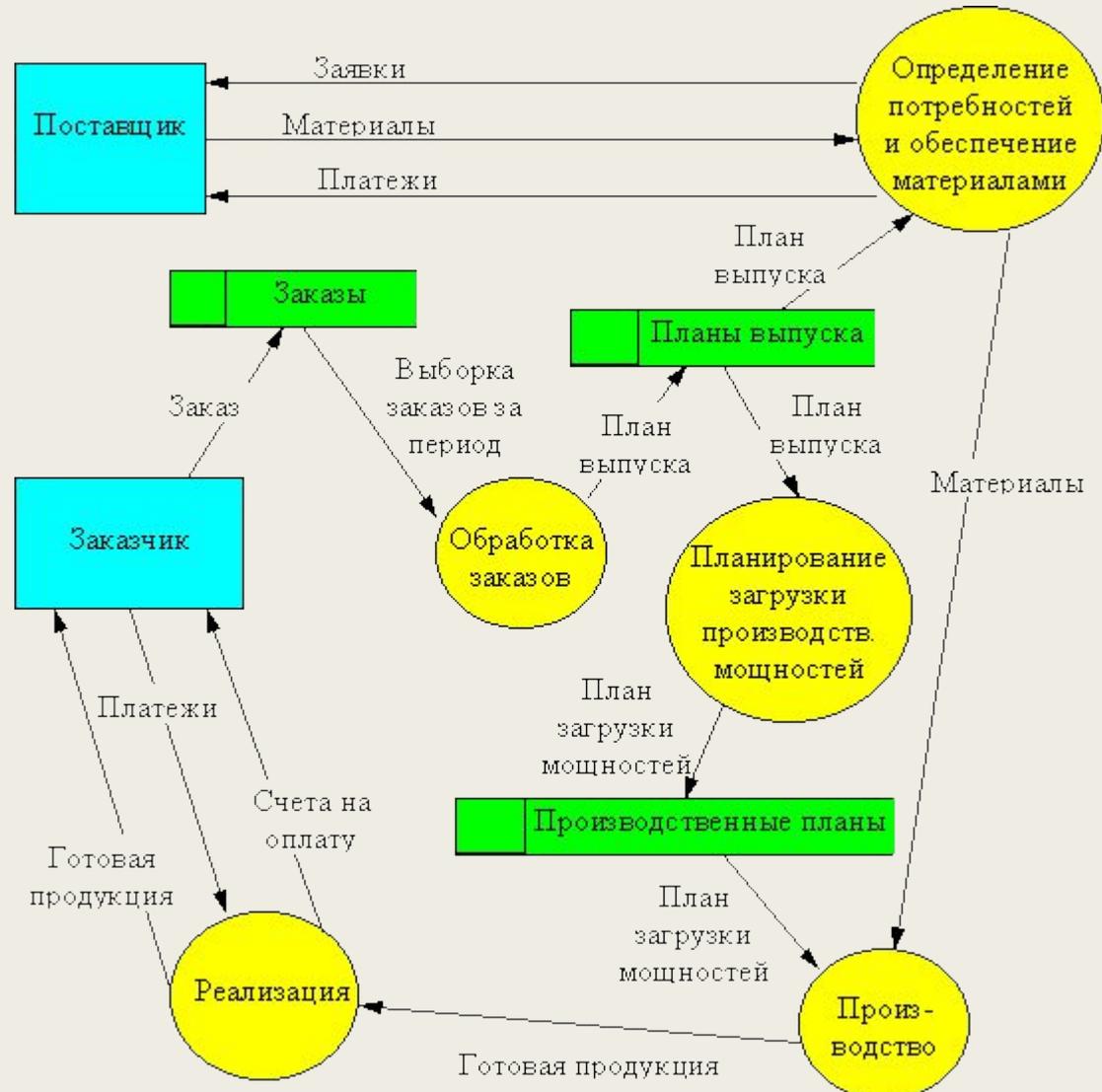


Рис. 2. DFD для предприятия, по принципу "изготовление на заказ"

# Правила и рекомендации построения DFD

По аналогии с IDEFO у каждого процесса (подсистемы) на диаграмме потоков данных должен быть как минимум один входящий и один выходящий поток. Процесс должен запускаться на выполнение либо через обрабатываемый, либо через управляющий поток данных. Работа каждого процесса должна завершаться конкретным результатом.

Каждый накопитель данных также должен иметь как минимум один входящий и один выходящий поток. Наличие только входящих потоков в накопитель означает, что информация накапливается, но не используется.

Наличие только выходящих потоков из накопителя также является ошибкой. Прежде чем использовать данные из накопителя, они должны там появиться в результате работы какого-либо процесса (подсистемы, внешней сущности). Исключением из правил считается случай, когда накопитель является внешней сущностью. Тогда допускается наличие либо только входящих стрелок, либо только выходящих стрелок.

# Средства проектирования

Инструментальные средства проектирования (CASE - системы), как правило, поддерживают несколько нотаций представления DFD-диаграмм. Одной из таких систем является Power Designer компании Sybase, который включает следующие модули:

- Process Analyst - построение диаграмм потоков данных с использованием любой из вышеупомянутых нотаций;
- Data Analyst - построение диаграмм "сущность-связь" и преобразование ее в реляционную модель;
- Application Modeller - средство для генерации приложений.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дэвид А. Марка, Клемент Л. МакГоуэн Методология структурного анализа и проектирования SADT. - М. : 1993.
2. Калянов Г.Н. CASE: структурный системный анализ (автоматизация и применение). М. : ЛОРИ. 1996.
3. Современные методологии описания бизнес-процессов – просто о сложном [Электронный ресурс]. URL: <http://www.betec.ru/index.php?id=6&sid=29> (дата обращения: 05.11.2017).
4. Методологии функционального моделирования [Электронный ресурс]. URL: [http://www.mstu.edu.ru/study/materials/zelenkov/ch\\_5\\_3.html](http://www.mstu.edu.ru/study/materials/zelenkov/ch_5_3.html) (дата обращения: 05.11.2017).
5. Методология DFD [Электронный ресурс]. URL: [https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6\\_3](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3) (дата обращения: 05.11.2017).