

# Программная инженерия

## Лекция 5-6

Составитель: Эверстов В.В.

Дата составления: 28/02/2011

Дата модификации: 01/02/2014

# Стадии разработки ПО

- **Предпроектные исследования;**
- **Техническое задание;**
- Техническое проектирование;
- Рабочее проектирование. Испытания;
- Ввод в эксплуатацию (внедрение);

# Действия на стадии проектирования

- Исследование предметной области,
- Определение объектов и их атрибутов,
- Установление всех структурных связей между объектами и процессами,
- Определение технологии работы ИС.

# Исследование предметной области

Проводятся для определения

- границы предметной области и возможности ее расширения,
- перечня объектов предметной области,
- информационных потребностей пользователей,
- необходимые процессы обработки данных с указанием их периодичности.

# Исследование предметной области

- На данном этапе можно строить различные модели:
  - Функциональная модель,
  - Объектная модель,
  - Модель потоков данных и т.д.

# Исследование предметной области

- При обследовании предметной области удобно пользоваться различными языками нотаций:
  - IDEF
  - DFD
  - UML

# IDEF

- IDEF - методологии семейства ICAM (Integrated Computer-Aided Manufacturing) для решения задач моделирования систем, позволяет отображать и анализировать модели деятельности широкого спектра систем в различных разрезах. При этом широта и глубина обследования процессов в системе определяется самим разработчиком, что позволяет не перегружать создаваемую модель излишними данными.

# Стандарты IDEF

- В настоящий момент к семейству IDEF можно отнести следующие стандарты:
  - IDEF0 — Function Modeling — методология функционального моделирования. С помощью наглядного графического языка IDEF0 изучаемая система предстает перед разработчиками и аналитиками в виде набора взаимосвязанных функций (функциональных блоков — в терминах IDEF0). Как правило, моделирование средствами IDEF0 является первым этапом изучения любой системы. Методологию IDEF0 можно считать следующим этапом развития хорошо известного графического языка описания функциональных систем [SADT](#) (Structured Analysis and Design Technique);
  - IDEF1 — Information Modeling — методология моделирования информационных потоков внутри системы, позволяющая отображать и анализировать их структуру и взаимосвязи;
  - IDEF1X (IDEF1 Extended) — Data Modeling — методология построения реляционных структур (баз данных), относится к типу методологий «Сущность-взаимосвязь» (ER — Entity-Relationship) и, как правило, используется для моделирования реляционных баз данных, имеющих отношение к рассматриваемой системе.

# Концепции IDEF

- Методология IDEF0 основана на следующих концептуальных положениях:
  - Модель,
  - Блочное моделирование и его графическое представление,
  - Передача информации,
  - Строгость и формализм.
  - Итеративное моделирование.
  - Отделение «организации» от «функций».

# Концепции IDEF

- **Модель** – искусственный объект, представляющий собой отображение (образ) системы и ее компонентов. **Система** представляет собой совокупность взаимосвязанных и взаимодействующих частей, выполняющих некоторую полезную работу. Частью (элементами) системы могут быть любые комбинации разнообразных сущностей, включающие людей, информацию, программное обеспечение, оборудование, изделия, сырье или энергию (энергоносители). Модель описывает, что происходит в системе, как ею управляют, какие сущности она преобразует, какие средства использует для выполнения своих функций и что производит.

# Концепции IDEF

- **Блочное моделирование и его графическое представление**
  - Основной концептуальный принцип методологии IDEF – представление любой изучаемой системы в виде набора взаимодействующих и взаимосвязанных блоков, отображающих процессы, операции, действия (определения – см. ниже), происходящие в изучаемой системе. В IDEF0 все, что происходит в системе и ее элементах, принято называть **функциями**. Каждой функции ставится в соответствие **блок**.
  - На **IDEF0 – диаграмме**, основном документе при анализе и проектировании систем, блок представляет собой прямоугольник. Интерфейсы, посредством которых блок взаимодействует с другими блоками или с внешней по отношению к моделируемой системе средой, представляются **стрелками** ), входящими в блок или выходящими из него. Входящие стрелки показывают, какие условия должны быть одновременно выполнены, чтобы функция, описываемая блоком, осуществилась.

# Концепции IDEF

- **Передача информации.** Средства IDEF0 облегчают передачу информации от одного участника разработки модели (отдельного разработчика или рабочей группы) к другому. К числу таких средств относятся:
  - диаграммы, основанные на простой графике блоков и стрелок, легко читаемые и понимаемые;
  - метки на естественном языке для описания блоков и стрелок, а также глоссарий и сопроводительный текст для уточнения смысла элементов диаграммы;
  - последовательная декомпозиция диаграмм, строящаяся по иерархическому принципу, при котором на верхнем уровне отображаются основные функции, а затем происходит их детализация и уточнение;
  - древовидные схемы иерархии диаграмм и блоков, обеспечивающие обзорность модели в целом и входящих в нее деталей.

# Концепции IDEF

- *Строгость и формализм.*
- *Итеративное моделирование.*
- *Отделение «организации» от «функций».*

# Синтаксис IDEF0

- Набор структурных компонентов языка, их характеристики и правила, определяющие связи между компонентами, представляют собой синтаксис языка. Компоненты синтаксиса IDEF0 – блоки, стрелки, диаграммы и правила.

# Блок

- Блок описывает функцию. 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

# Правила рисования блоков

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

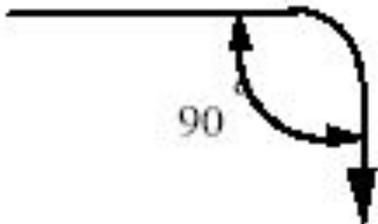
# Стрелка

- Стрелка формируется из одного или более отрезков прямых и наконечника на одном конце. Стрелки не представляют поток или последовательность событий, как в традиционных блок-схемах потоков или процессов. Они лишь показывают, какие данные или материальные объекты должны поступить на вход функции для того, чтобы эта функция могла выполняться.

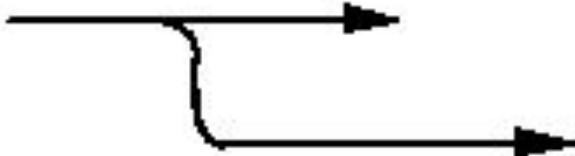
# Стрелки



Прямолинейный отрезок стрелки



- Ломаный сегмент стрелки. Дуга сопряжения  $-90$  град.



Ветвление стрелок



- Слияние стрелок

# Правила рисования стрелок

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

# Семантика блоков и стрелок

- Чтобы гарантировать точность модели, следует использовать стандартную терминологию. Блоки именуются глаголами или глагольными оборотами и эти имена сохраняются при декомпозиции Стрелки и их сегменты, как отдельные, так и связанные в «пучок», помечаются существительными или оборотами существительного. Метки сегментов позволяют конкретизировать данные или материальные объекты, передаваемые этими сегментами, с соблюдением синтаксиса ветвлений и слияний.

# Семантика блоков и стрелок



# Имена и метки

- Как указывалось, имена функций – глаголы или глагольные обороты. Примеры таких имен:

производить детали

планировать ресурсы

наблюдать

наблюдать за выполнением

проектировать систему

эксплуатировать

разработать детальные чертежи

изготовить компонент

проверить деталь

- Стрелки идентифицируют данные или материальные объекты, необходимые для выполнения функции или производимые ею. Каждая стрелка должна быть помечена существительным или оборотом существительного, например:

Спецификации

отчет об испытаниях

бюджет

Конструкторские требования

конструкция детали

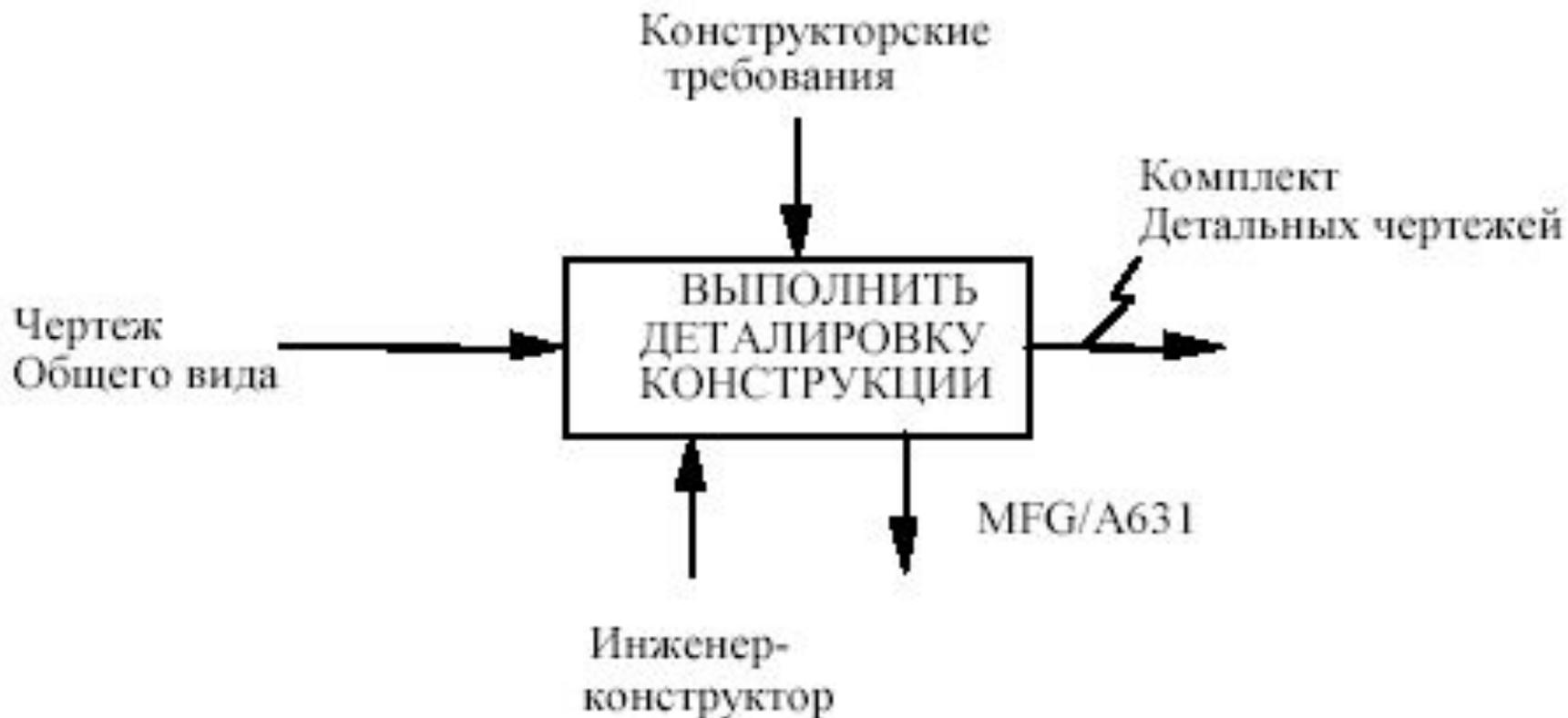
директива

Инженер-конструктор

плата в сборе

требования

# Семантические правила блоков и стрелок



# Дочерние диаграммы

...

A-1	Дополнительная контекстная диаграмма
A-0	Обязательная контекстная диаграмма верхнего уровня (содержащая высший блок A0)
A0	Верхняя дочерняя диаграмма
A1, A2, ..., A6	Дочерние диаграммы
A11, A12, ..., A16, ..., A61, ..., A66	Дочерние диаграммы
A111, A112, ..., A161, ..., A611, ..., A666	Дочерние диаграммы

# Перечень узлов

A0 Производить продукт

A1 Планировать производство

A11 Выбрать технологию производства

A12 Оценить требуемое время и затраты на производство

A13 Разработать производственные планы

A14 Разработать план вспомогательных действий

A2 Разрабатывать и управлять графиком выпуска и ресурсами

A21 Разработать основной график

A22 Разработать график координации работ

A23 Оценивать затраты и приобретать ресурсы

A24 Следить за выполнением графика и расходом ресурсов

A3 Планировать выпуск продукции

