# **ЛК-1. Жизненный цикл ПО**

Одним из базовых понятий методологии проектирования ИС является понятие **жизненного цикла** ее программного обеспечения (ЖЦ ПО). Методология проектирования *информационных системы* (ИС) описывает процесс создания и сопровождения систем в виде их **жизненного цикла** (ЖЦ), представляя его как некоторую последовательность стадий и выполняемых на них процессов.

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

**Этапы ЖЦ ПО:**

1. Формирование требований к системе (планирование) = сочинение
2. Проектирование
3. Реализация (кодирование)
4. Тестирование
5. Ввод в действие
6. Эксплуатация и сопровождение (2/3 ЖЦ)

\* если последний этап не включают в цикл, то тогда говорят только о периоде создания ПО

Для **каждого** этапа **определяется:**

* 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.

**Члены:**

* руководитель проекта (ответственный за качество)
* разработчики
* проектировщики
  + разраб. архитектуру
  + специалист по анализу чел.ф.
  + программисты UI и т.д.

**Стандарты, регламентирующие ЖЦ ПО** (а в некоторых случаях процесс разработки):

* ГОСТ 34.601-90
* ISO-IEC 12207:1995

определяет структуру ЖЦ, содержащую процессы, действия и задачи, которые должны быть выполнены во время создания ПО.

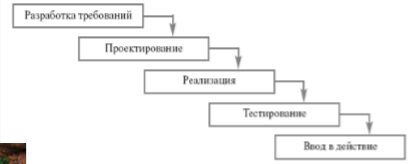
* Custom Development Method (CMD – методика Oracle)
* Rational Unified Process (RUP)
* Microsoft Solution Framework (MSF)
* Extreme Programming (XP)

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

Модель ЖЦ зависит от специфики ИС и условий, в которых последняя создается и функционирует.

**Классические модели ЖЦ:**

* Каскадная – результаты первого блока влияют на процесс следующего
  + Однотактный подход –этап, документ; этап, документ;
  + Прогнозирующая методология – планируешь время и деньги;



* Поэтапная модель с промежуточным контролем



* Спиральная – каждый виток спирали соответствует созданию версии продукта
  + +: позволяет уточнить требования, цели
  + -: высокие требования к заказчику (четкие)
  + -: трудно контролировать время разработки



* Итерационная

**Методология проектирования ПО**

1. **Agile Model**

**Agile –** семейство процессов разработки, а не единственный подход в разработке ПО. Не включает практик, а определяет ценности и принципы, которыми руководствуется команда.

* Люди и взаимодействие важнее процессов и инструментов
* Работающий продукт важнее исчерпывающей документации
* Сотрудничество с заказчиком важнее соблюдения контракта
* Готовность к изменениям важнее первоначального плана

1. **SCRUM**

**Scrum –** набор принципов, на которых строится процесс разработки, позволяющий в жестко фиксированные и небольшие по времени итерации, называемые **спринтами**, предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определен наибольший приоритет.

Возможности ПО к реализации в очередном спринте определяются в начале спринта на этапе планирования и не могут изменяться на всем его протяжении. При этом строго фиксированная небольшая длительность спринта придется процессу разработки предсказуемость и гибкость.

**Подходы к выбору методологии:**

*Принцип 1*. Большая по размерам методология нужна тогда, когда в проекте занято большое число разработчиков.

*Принцип 2*. Большая корректность методологии (видимая со стороны) или, другими словами, «большая плотность» нужна в тех случаях, когда скрытые ошибки в программном продукте 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.

# **ЛК-2. Основные понятия**

**Система** – комплекс взаимодействующих элементов;

**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 | Система отделена от ОКРС границами |
| Целостность | Ее свойство целого принципиально не сводится к сумме свойств составляющих элементов |
| Структурность | Поведение системы обусловлено не только особенностями отдельных элементов, сколько свойствами ее структуры |
| Взаимозависимость со средой | Система формирует и проявляет свойства в процессе взаимодействия со средой |
| Иерархичность | Соподчиненность элементов в системе |
| Множественность описаний | По причине сложности познание системы требует множественности ее описании |

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

3\*3 – 3 штуки решают 3 задачи.

**Особенности проектов:**

* Сложность описания
* Наличие совокупности тесно связанных компонент
* Ограниченная возможность использования типовых решений
* Необходимость интеграции
* Функционирование в неоднородной среде на нескольких аппаратных платформах
* Разобщенность и разнородность отдельных групп разработки
* Существенная временная протяженность проекта

**Модель** – формальное описание особенностей системы, которые существенны для ее исследования.

**Различают три вида моделей:**

* *вербальные*

словесные, описательные

* *натурные*

воссоздание с помощью физических свойств

макетирование, физическое моделирование, масштабированные модели, модели части свойств и т.д.

* *знаковые*

обозначения, схемы, диаграммы, математические уравнения

**Математическая модель –** описание протекания процессов, опис.состояния.

**Информационный поток** – совокупность циркулирующих в системе, между системой и внешней средой сообщений, необходимых для управления, анализа и контроля операций.

**По структуре** информационные потоки бывают:

* *однородные*

характеризуются единым видом носителя, единой функциональной принадлежностью

* *неоднородные*

разнородность носителей и функциональной принадлежности

**По периодичности** информационные потоки бывают:

* *регулярные*

соответствующие регламентированной во времени и передаче данных

* *оперативные*

обеспечивающие связь по требованию

**Виды** информационных потоков характеризуются такими показателями:

* источник возникновения
* направление движения потока
* скорость передачи и приема
* интенсивность потока и т.д.

Формирование ИС невозможно без исследования потоков в разрезе определенных показателей.

Формирование общего представления о системе можно разбить на **стадии:**

* стадии, которые описывают общее, цельное «изучение системы»
* стадии, которые образуют группу «формирование углубленных представлений о системе»
* стадия «моделирования системы»
* стадии, которые образуют группу «сопровождение системы»

Структура ИС = совокупность обеспечивающих подсистем.

**Подсистема** – часть системы, выделенная по какому-либо признаку.

**Типы обеспечивающих подсистем:**

* информационное обеспечение
* техническое обеспечение
* математические и программное обеспечение
* организационное обеспечение
* правовое обеспечение

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

*Цель:* устранение недостатков унифицированной системы документации.

*Назначение:* своевременное формирование и выдача достоверной информации для принятия управленческих решений.

**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.

**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.

***Состав:*** законы, указы, постановления гос.органов власти, приказы, инструкции и т.д.

**Программное обеспечение:**

* общесистемные продукты
* специальные программные продукты
* техническая документация

**Классификация ИС:**

* по типу хранимых данных
  + *фактографические* – не структурируемая, необработанная инфа
  + *документальные* – документы, структурированная информация и методы ее обработки
* по степени автоматизации информационных процессов
  + *ручные* – абсолютно полностью управляет человек
  + *автоматизированные* – скрипты, но их должен кто-то запустить (50/50)
  + *автоматические* – полностью бесчеловечное управление
* по сфере применения
  + *интегрированные* – часть какого-то глобального комплекса
  + *организационного управления* – служат для манипуляции другими подсистемами, системы управления предприятиями (бухгалтерия, управление кадрами)
  + *управления ТП* – информационные системы управления производственными процессами, система, управляющая сборками автомобиля
  + *САПР* – системы автоматического проектирования, которые позволяют что-то создавать
* по характеру обработки данных
  + *информационно-поисковые* – это системы, которые где-то что-то взяли и куда-то что-то передали
  + *информационно-решающие* – системы, которые вникают в суть данных, могут не отвечать за транспорт, они должны сказать некий вердикт
    - *управляющие* – вмешиваются в процесс и могут корректировать
    - *советующие* – простое уведомление
* по уровню управления
  + *стратегические* – системы, которые предназначены для того, чтобы не было привязки ко времени, имеется доступ к большому количеству ресурсов у системы
  + *функциональные* – обычные системы, которые решают узкую задачу, четко заданная задача – система ее решает
  + *операционные* – интегрированные системы, обработка транзакций. Конкретного рода операции

**Интегрированные** (корпоративные) ИС – КИС используются для автоматизации работ от планирования деятельности до сбыта продукции.

Подсистемы могут включать:

* Подсистему маркетинга
* Производственные подсистемы
* Финансовые и учетные подсистемы
* Подсистему кадров (человеческие ресурсы)
* Прочие подсистемы

# **ЛК-3. Основы проектирования ИС**

**Основными документами**, содержащими требования на разработку информационной системы, являются **календарный план** выполнения работ и **техническое задание.**

Календарный план регламентирует состав, сроки и финансирование работ.

Техническое задание регламентирует основные требования к системе.

**Техническое задание** является основным документом, определяющим требования и порядок создания (развития и модернизации) системы.

В **ТЗ** указываются процессы контроля, контрольные точки, процессы приема-сдачи, описание документации, описание дальнейшего процесса модернизации. ТЗ – очень важный документ для исполнителей.

*Вариант 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 систем (проектной документации).

Когда мы переходим к разработке проектной документации, есть несколько подходов к этому процессу. Все подходы используют определенную группу признаков.

**Основные принципы проектирования:**

1. Принцип декомпозиции («разделяй и властвуй») – любые сложные проблемы можно разбить на задачи

2. Принцип иерархического упорядочивания – есть некоторый уровень взгляда на систему, наследование

3. Принцип концептуальной общности – единый подход для всех этапов

4. Принцип абстрагирования – на каждом этапе системы нужно понимать, что важно

5. Принцип формализации – некое методические описание, все по полочкам, по шагам описано

6. Принцип унификации – стараться максимально использовать общепринятые формации

7. Принцип логической независимости – в данном аспекте он относится к синтезу информационной системы, мы продумываем бизнес логики без конкретной реализации

8. Принцип многомодельности – мы можем под разными ракурсами взглянуть на проблему

9. Принцип непротиворечивости (согласованности) – те же модели должны согласовываться между собой

10. Принцип информационной закрытости (инкапсуляции) – для некоторых моделей нужно скрывать внутреннюю сердцевину

11. Принцип полиморфизма – должны проектировать так, чтобы внешний вид нашего объекта можно было легко менять, не затрагивая бэк. Разделять сущность и интерфейс.

**Модель как проекция «системы»**

При моделировании систем реального мира и нетривиальных программных систем выбор проекции неочевиден

Проекция (точка зрения) определяет, на какие вопросы может ответить модель.

**Визуальное представление моделей:**

* блок-схема
* граф переходов
* конечный автомат

**Концептуальная модель** – то, что написали в ТЗ. Из чего состоит система, что может

**Логическая модель** – все эти объекты – как взаимодействую между собой, что из себя представляют

**Физическая модель** – реализация

**Классификация моделей ИС:**

1. по строгости описания
   1. *неформальное* – приближенное, эскизы
   2. *формальной* – дотошное описание
2. по степени физической реализации (логической независимости)
   1. *логические* – концепты с точки зрения теоретического алгоритма
   2. *физические* – конкретные типы данных, структуры
3. по степени отображения динамики происходящих процессов
   1. *статические* – блок-схемы
   2. *динамические* – конечные автоматы
4. по отображаемому аспекту
   1. *функциональные* – основной аспект, должны показать какие функции есть у системы
   2. *информационные* – показывают, какие информационные потоки внутри или снаружи системы
   3. *поведенческие* – чаще всего показывает, как можно пользоваться системой
   4. *компонентные* – структурная схема: структура файлов, классов, библиотек
   5. *смешанные* – и те, и иные модели

**Подходы к проектированию ИС**:

* 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.

**Структурный анализ** – 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 функция, которые они выполняют.

***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 (САПР) – позволяет изучить предметную область, задачи, типовые решение и в итоге создастся конструктор, которые позволит создавать проект;
* 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;

нотация – строгое графическое определение того или иного графического элемента, которая позволяет описать все действия.

**Схема применения структурного подхода**

1. Разработка функционально модели (что умеет делать или должно делать ПС)
2. Разработка информационной модели (какие структуры данных поступают на вход ПС, что возвращает ПС и как они изменяются во время работы)
3. Разработка поведенческих моделей (модели использования системы (какой персонал как взаимодействует с системой)
4. Разработка моделей компонентов и развертывания (как взаимодействуют компоненты на физическом уровне)

**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 – integrates computer-aided manufacturing (конец 70х)
  + Интегрированная компьютеризация производства США
  + Начало разработки методологии IDEF (ICAM Definition)

SADT => ICAM DEFinition => Integrated DEFinition => IDEF0;

**Методология** структурного анализа и поведения системы

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

**IDEF –** сокращение от Integration Definition Methodology (объединение методологических понятий).

**Семейство IDEF:**

* IDEF0 – функциональное моделирование – используется для построения функциональной модели, описывая бизнес-функции и контекст поведения
* IDEF1 – информационное моделирование – представляет собой информационную и модель и применятся для поддержки функций производственной среды, описываю что куда двигается
* IDEF2 – модель симуляции – содержит нотации, которые позволяют смоделировать динамические системы (меняющиеся во времени функции, поведение системы)
* IDEF3 – методология описания процессов моделируемой системы– состоит их PFD и OSTD
  + Process Flow Description (PFD)
  + Object State Transition Description (OSTD)
* IDEF4 – объектно-ориентированное планирование – для выделения каких-то объектов (его заменил UML)
* IDEF5 – описание антологий – позволяет описать антологию проектируемой системы
* 6 – использование рационального опыта, т.е. проектирование типовыми решениями
* 7 – система аудита4
* 8 – моделирование пользовательского интерфейса
* 9 – изучение бизнес процессов с т.зр. ограничений и дополнительных инструкций, архитектурных решений, организационное планирование
* 10 – 13
* 14 – моделирование компьютерных сетей, предназначена для описания требований к надежности, компонентам, концепциям и т.д.

# **ЛК-4. IDEF 0. Описание, назначение и применение**

**IDEF0** - методология функционального моделирования и графическая нотация, предназначенная для формализации и описание бизнес-процессов системы.

С помощью графических элементов мы описываем все что важно для реализации. Главная особенность – соподчиненность объектов. В IDEF0 рассматриваются только логические отношения между объектами и абсолютно нет временной шкалы.

Основная структурная единица – **диаграмма**. На ней обозначается некоторая модель предметной области или ее части. *Диаграмма* – условно говоря некоторый лист, на котором действующие нотации отображают отдельные части предметной области.

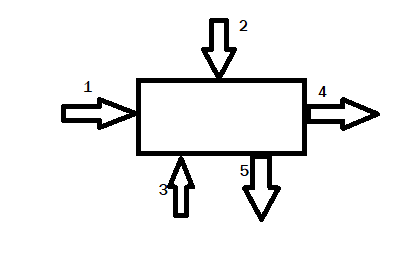
Главные компоненты диаграммы – **блоки**. Блок отображается в виде некоторого прямоугольника и отражает описание некоторой работы, процесса, функции, которые выполняются в некотором контексте задачи в течении времени, которое для нас не важно. Мы просто говорим «вот это функция, она есть, и она выполняется». Внутри блока описывается ***имя функции***, т.е. это должен быть глагол (разработать модель, загрузить файл и т.д.). Вторая неотъемлемая часть – ***номер*** (в правом нижнем углу (*обязательно!*) – идентификационный номер). Нумерация сквозная по всей модели, т.е. блоки уникальны не только в рамках 1 диаграммы, но и во всей модели. Это позволяет легко коммуницировать и искать блоки просто по номеру.

Требований к размерам блоков не предъявляется. Они не должны выходить за поля контекста, а также должны быть таким, чтобы можно было прочитать текст. Также блок должен быть нарисован сплошными линиями и с прямыми углами. Никакие элементы не должны накладываться друг на друга.

Кроме блоков, в диаграмме существуют **стрелки**. Они описывают потоки данных или материальные объекты, которые связаны с определенной функцией. В классическом виде -> есть 4 вида стрелок (влево/вправо/вверх/вниз).

**Стрелка –** сплошная линия, у которой есть наконечник и может быть искривлена (желательно под прямым углом). Толщина не имеет значения. Отрезки только параллельно осям (горизонтально и вертикально). Все стрелки должны соприкасаться с какими-то частями блока. Не должна входить непосредственно в угол блока.

У каждого блока должна быть минимум 1 стрелка управления.



* 1 – **вход функции** – входящая стрелка всегда примыкает слева и направлена в сторону функции, обозначает некоторый материал или инфу, которая используется и/или преобразуется функцией для получения результата (ее *может не быть* и функция может сама генерировать данные)
* 2 – **управление функцией** – некоторые условия/правила/стандарты/концепции/стратегии выполнения функции, которые влияют на ее выполнение (функция загрузки файлов – под 2 правила взаимодействия с системой, ограничения и т.д.)
* 3 – **механизмы** – некоторые ресурсы (мб человеческие), с помощью которых выполняется функций (это мб денежные средства, персонал, инструментарий) либо какие ресурсы используются для ее выполнения
* 4 – **выход функции** – результат выполнения функции (некоторая инфа или материал) – *всегда должен быть!!как минимум 1*
* 5 – **вызов функции / запрос** – специализированная дуга, которая указывает на некоторую другую проблемную (предметную) область (редко применяется) – под проблемной (предметной) областью подразумевается к примеру администрирование пользователей, процесс коммуникации, процесс обеспечения сохранности данных (это 3 отдельные предметные области, которые связаны друг с другом и образуют целостную систему)

В лабе рассматриваем только 1 контекст!

В IDEF0 допустимы слияния стрелок и разветвления.

Существует **5 типов взаимодействия** между блоками. Связь это не всегда вход-выход. Рассмотрим:

* Связь выход – вход => выход одной функции – вход другой функции
* Связь выход – управление => первая функция генерирует управляющие конструкции для другой функции
* Обратная связь выход – управление => стрелка со 2ой функции к 1ой (do..while)
* Обратная связь выход-вход => инфа с первого блока на второй, а со второго обратно на первый (заполнение формы – проверка на корректность – если неправильно, опять на заполнение формы для того, чтобы дозаполнить данные)
* Связь выход – механизм => в качестве выхода может быть информация или материал; к примеру выход – деталь, которая выполняет действие в другом механизме и поступает как механизм для второго.

# **ЛК-5. Описание и назначение методологии IDEF0**

В IDEF0 реализ **три базовых принципа** **моделирования** бизнес-процессов:

1. ***Принцип функциональной декомпозиции***

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

1. ***Принцип ограничения сложности***

Согласно данному принципу количество функциональных блоков на одной диаграмме должно быть не менее 2 (за исключение контекстной диаграммы) и не более 6.

Таким образом обеспечивается разборчивость и удобочитаемость диаграмм IDEF0. Практика показывает, что соблюдение этого принципа в большинстве случаев приводит к кому, что процессы, представленные в виде модели IDEF), становятся лучше структурированы, более понятны и легче поддаются анализу.

1. ***Принцип контекста***

Состоит в том, что моделирование бизнес-процесса начинается с построение контекстной диаграммы. На этой диаграмме отображается только один блок – главная бизнес-функция моделируемой модели.

При определение главной бизнес-функции необходимо всегда иметь в виду цель моделирования и точку зрения на модель.

**Нотации IDEF0**

Основной структурной единицей является *диаграмма*, представляющая собой графическое описание модели предметной области или ее части. Главными компонентами являются *блоки*.

**Синтаксис IDEF0**

Диаграмма содержит:

* Блоки – функции, определяемые как деятельность, процесс, операция, действие или преобразование;
* Стрелки – данные или мат объекты, связанные с функциями;
* Правила определяют, как следует применять компоненты;

На самом раннем этапе моделирования перед началом разработки модели необходимо **определить** ее направленность:

* ***Контекст*** определяет объект модели как часть целого.

очерчивает границы модели с ее внешним окружением посредством описания внешних интерфейсов (то, что надо здесь и сейчас)

* ***точка зрения*** специфицирует, что можно «увидеть» в контексте и под каким «углом»

определяет позицию автора как наблюдателя системы или ее элемента и выбирается таким образом, чтобы получить максимально полезную информацию из разрабатываемой модели

* ***цель*** определяет назначение модели или обеспечиваемых ею взаимодействий

воплощает причину, по которой модель создана

Каждая модель представляет одну точку зрения и одну цель. *Выбор цели* осуществляется с учетом тех вопросов, на которые должна ответить модель*. Выбор точки зрения* – в соответствии с выбранной позицией ответа на поставленные вопросы.

Обычно модель содержит **4 типа диаграмм**:

* ***контекстная*** (1 на всю модель)
* ***декомпозиции***
* дерева узлов – иерархия;
* экспозиции (FEO) – вербальная, описание словами;

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

Например, функции пользователя и функции администратора – это 2 отдельных контекста.

Номер главной контекстной функции имеет номер **А0.**

Диаграммы IDEF0 имеют **двойную нумерацию**. Диаграммы имеют номера по узду, который они раскрывают и сквозную нумерацию.

Контекстная диаграмма всегда имеет номер А0.

Декомпозиция контекстной диаграммы – номер А0, остальные диаграммы декомпозиции – по номера узлов.

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

Дуга, помещенная в туннель в месте присоединения к блоку, означает, что данные, выраженные ею, не будут рассматриваться на диаграмме-потомке.

Дуга, помещенная в туннель на свободном конце, означает, что данные, выраженные ею, не относятся к родительскому блоку и на ней не описываются.

**Соглашения по построению диаграмм**

* блоки располагаются на диаграмме от левого верхнего угла до правого нижнего и нумеруются в том же порядке
* при декомпозиции блока (особенно при коллективной работе) снаружи правого нижнего блока записывается С-номер диаграммы
* стремитесь максимально увеличить расстояние между блоками и поворотами дуг, а также между пересечениями последних
* объединение дуг, источники которых не выделены на диаграмме позволяют графически подчеркнуть единый источник исходных данных
* при соединении блоков избегайте пересечения дуг
* минимизируйте число петель и поворотов каждой дуги. Это также упростит диаграмму и повысит ее читабельность

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 окончательным вариант ИС.

# **ЛК-6. Пример построения IDEF0 и IDEF3**

Пример: приложение, которое предназначено для сбора данных из соц.сетей, анализа информации и выводы отчетов.

Было сформировано ТЗ **функциональных возможностей**:

* предоставление веб-интерфейса с аутентификацией пользователей
* построение запроса для поиска данных
* выполнение запроса к социальным сетям для выборки
* сохранение данных из соц. сетей в базу данных
* анализ текстов на предмет эмоциональной нагрузки
* построение и отображение аналитических отчетов
* ведение истории запросов
* выгрузка данных из системы во внешние документы

**Контекст** использования для моделирования: анализ данных.

РИС контекстной диаграммы

Т.к. будет выполнятся декомпозиция, то за пределам блока пишем А0, а внутри блока 0. Если не допускает редактор, то в правом нижнем углу пишем А0 – это значит что данный блок будет декомпозирован.

РИС

# **ЛК-7. Описание и назначение методологии IDEF3**

**IDEF3** – предназначен для построение поведенческой модели.

**IDEF3** - методология описания процессов, проходящих в системе;

IDEF3 является технологией, хорошо приспособленной **для сбора данных**, требующихся для проведения структурного анализа системы.

IDEF3 предоставляет **инструмент моделирования** сценариев действий сотрудников организации, отделов и т.д.

**IDEF3** - это метод, имеющий основной целью дать возможность аналитикам описать ситуацию, когда процессы выполняются в определенной последовательности.

Любое IDEF0 мб представлен в виде последовательности процессов или операций способами IDEF3.

Может использовать самостоятельно либо вместе с IDEF0.

IDEF3 состоит из **2 методов**:

* process flow description – описание технологических процессов, с указанием того, что происходит на каждом этапе
* object state transition description – описание переходов состояний объектов, с указанием того, какие существуют промежуточные состояния у объектов в моделируемой системе.

РИС

Основная единица описания – **диаграмма.**

IDEF3 еще называют Workflow diagramming.

**Единица работы** (процесс) – Unit of work(UOW) – также называемые работами, являются центральными компонентами модели – тоже самое что блоки в IDEF0.

Каждая работа описывает некоторый сценарий процесса и может являться составляющей некоторой другой работы.

**Компоненты диаграммы** описания процесса:

* работы
* связи
* перекрестки
* объекты ссылок
* единица поведения
* разложение
* разработка

# **ЛК-8. UML диаграммы**

Пример:  
 Диаграмма должна обеспечивать

**Пример построения диаграммы Вариантов использования**

Выделим следующие *роли*:

* пользователь
* менеджер
* ИС «агрегатор соц.сетей»

Выделим следующие «внешние» *функции системы* (прецеденты):

Прецеденты = то, что мы можем сделать с системой

* Ввод запроса на анализ соц.сети
* Получение и просмотр результатов анализа
* Выгрузка отчета в документ
* Ведение статистика запросов (обращений пользователей)
* Аутентификация

**Иерархия актеров**

ИС «агрегатор» - независимая роль

Роль менеджер – расширение роли пользователь, т.е. он берет возможности пользователя и расширяет их своими.

Картинка

**Прецеденты**

Диаграмма классов описывает взаимодействие абстрактных объектов. Показывает, что именно участвует в программе

На диаграмме есть нотации. В данном случае диаграмма отделяется отдельным пространством.

На диаграмме классов отображаются сущности классов, которые представлены в виде 3 частей: имя класса, его атрибуты и его операции.

Атрибут описывает ….

На примере 7 классов : краткие и полные нотации (в кратких есть только имя).

**Связи на диаграмме.**

1. Ассоциации –

Ассоциации обычно используются для проектирования БД и показывает зависимости между классами. Осуществляется по ключу (обще поле в 2 классах)

Все связи имеют кратности. Кратность обозначает сколько объектов одного класса участвуют в других классах. Чаще всего встречаются: 1, 0..1, \*, n..m (n..\*).

1. Обобщение –

По сути это аналог замещаемости, наследования. Отображается в виде стрелочки с треугольником в конце, т.е. закрытый наконечник. Идет от детей к родителю.

От наконечника зависит какой тип связи (открытый наконечник или закрытый (ассоциации и обобщение соответственно)).

Часто бывает, что нельзя в нотациях класса отобразить все, что мы хотим сказать о данной сущности, поэтому есть примечания и связывается с той сущностью к которой относится. Обозначается просто пунктирной линией (БЕЗ СТРЕЛКИ). Примечание = комментарий на диаграмме. Можно просто в углу вставлять примечание, чтобы знать где что происходит. А если надо связать с сущностью – рядом с помощью пунктирной линией. В редких случаях комментарий можно оставлять в самом прямоугольнике с описанием атрибутов – просто двойные - (--).

1. Зависимость - . Зависимость между 2 элементами существует, если при изменении 1 элемента (сервера) изменяется 2ой элемент (клиент). Зависимость может разветвляться.
2. Агрегация и композиция.

Это стрелочка, в которой есть 2 наконечника: стрелочка и ромб (белый – агрегация, черный – композиция). На агрегациях и композициях кратность с числом 1 зачастую не указывают. Агрегации лучше совмещать в одну, также как и композиции.

Черный ромбик – машина включает в себя элемент, который рядом со стрелкой. Элемент, который рядом со стрелкой не существует, значит и элемент рядом с ромбиком тоже не существует. Т.е. показывает, что 1 класс является неотъемлемой частью 2го класса.

Агрегация – отношение часть-целое. Должно быть 4 колеса, чтобы была машина. Композиция показывает, что одна составная часть может быть составной частью другого элемента. Например, двери могут быть как частью машины, так и частью дома (естественно это не в глобальном масштабе, а в рамках данной предметной области).

Пример: агрегация: цветок <>–> лепесток, стебель и т.д.; композиция – букет <>–> цветок.