

Нижегородский государственный университет им. Н.И. Лобачевского
Факультет вычислительной математики и кибернетики ННГУ

**Учебно-исследовательская лаборатория
«Математические и программные технологии для современных
компьютерных систем (Информационные технологии)»**

Инструментальные средства поддержки жизненного цикла программного обеспечения

Куратор мини-проекта:
Сысоев А.В.
Составители:
Стариков В.
Зеленцова М.

Содержание

| | | |
|-------|--|-----------|
| 1 | CASE-средства. Общая характеристика и классификация..... | 3 |
| 2 | Методологии и технологии проектирования ИС | 5 |
| 2.1 | Общие требования к методологии и технологии | 5 |
| 3 | Методология RAD..... | 6 |
| 4 | Сущность структурного подхода..... | 9 |
| 4.1 | Методология функционального моделирования SADT | 10 |
| 4.2 | Моделирование потоков данных (процессов) | 16 |
| 4.3 | Case-метод Баркера | 20 |
| 5 | Технология внедрения CASE-средств..... | 25 |
| 5.1 | Определение потребностей в CASE-средствах | 25 |
| 5.1.1 | <i>Анализ возможностей организации</i> | <i>26</i> |
| 5.1.2 | <i>Определение организационных потребностей.....</i> | <i>27</i> |
| 5.1.3 | <i>Анализ рынка CASE-средств</i> | <i>28</i> |
| 5.1.4 | <i>Определение критериев успешного внедрения</i> | <i>28</i> |
| 5.1.5 | <i>Разработка стратегии внедрения CASE-средств</i> | <i>28</i> |
| 5.2 | Оценка и выбор CASE-средств | 30 |
| 5.2.1 | <i>Общие сведения.....</i> | <i>30</i> |
| 5.2.2 | <i>Процесс оценки</i> | <i>31</i> |
| 5.2.3 | <i>Процесс выбора</i> | <i>31</i> |
| 5.2.4 | <i>Критерии оценки и выбора.....</i> | <i>32</i> |
| 5.3 | Выполнение пилотного проекта..... | 33 |
| 5.3.1 | Определение характеристик пилотного проекта..... | 34 |
| 5.3.2 | Планирование пилотного проекта | 35 |
| 5.3.3 | Выполнение пилотного проекта..... | 36 |
| 5.3.4 | Принятие решения о внедрении..... | 36 |
| 5.4 | Переход к практическому использованию CASE-средств | 37 |
| | Литература: | 39 |

1 CASE-средства. Общая характеристика и классификация

Современные CASE-средства охватывают обширную область поддержки многочисленных технологий проектирования ИС: от простых средств анализа и документирования до полномасштабных средств автоматизации, покрывающих весь жизненный цикл ПО.

Наиболее трудоемкими этапами разработки ИС являются этапы анализа и проектирования, в процессе которых CASE-средства обеспечивают качество принимаемых технических решений и подготовку проектной документации. При этом большую роль играют методы визуального представления информации. Это предполагает построение структурных или иных диаграмм в реальном масштабе времени, использование многообразной цветовой палитры, сквозную проверку синтаксических правил. Графические средства моделирования предметной области позволяют разработчикам в наглядном виде изучать существующую ИС, перестраивать ее в соответствии с поставленными целями и имеющимися ограничениями.

Обычно к CASE-средствам относят любое программное средство, автоматизирующее ту или иную совокупность процессов жизненного цикла ПО и обладающее следующими основными характерными особенностями:

- мощные графические средства для описания и документирования ИС, обеспечивающие удобный интерфейс с разработчиком и развивающие его творческие возможности;
- интеграция отдельных компонент CASE-средств, обеспечивающая управляемость процессом разработки ИС;
- использование специальным образом организованного хранилища проектных метаданных (репозитория).

Интегрированное CASE-средство (или комплекс средств, поддерживающих полный ЖЦ ПО) содержит следующие компоненты;

- репозиторий, являющийся основой CASE-средства. Он должен обеспечивать хранение версий проекта и его отдельных компонентов, синхронизацию поступления информации от различных разработчиков при групповой разработке, контроль метаданных на полноту и непротиворечивость;
- графические средства анализа и проектирования, обеспечивающие создание и редактирование иерархически связанных диаграмм (DFD, ERD и др.), образующих модели ИС;
- средства разработки приложений, включая языки 4GL и генераторы кодов;
- средства конфигурационного управления;
- средства документирования;
- средства тестирования;
- средства управления проектом;
- средства реинжиниринга.

Все современные CASE-средства могут быть классифицированы в основном по типам и категориям. Классификация по типам отражает функциональную ориентацию CASE-средств на те или иные процессы ЖЦ. Классификация по категориям определяет степень интегрированности по выполняемым функциям и включает отдельные локальные средства,

решающие небольшие автономные задачи, набор частично интегрированных средств, охватывающих большинство этапов жизненного цикла ИС и полностью интегрированные средства, поддерживающие весь ЖЦ ИС и связанные общим репозиторием. Помимо этого, CASE-средства можно классифицировать по следующим признакам:

- применяемым методологиям и моделям систем и БД;
- степени интегрированности с СУБД;
- доступным платформам.

Классификация по типам в основном совпадает с компонентным составом CASE-средств и включает следующие основные типы:

- **средства анализа (Upper CASE)**, предназначенные для построения и анализа моделей предметной области (Design/IDEF (Meta Software), VPwin (Logic Works));
- **средства анализа и проектирования (Middle CASE)**, поддерживающие наиболее распространенные методологии проектирования и используемые для создания проектных спецификаций (Vantage Team Builder (Cayenne), Designer/2000 (ORACLE), Silverrun (CSA), PRO-IV (McDonnell Douglas), CASE.Аналитик (МакроПроджект)). Выходом таких средств являются спецификации компонентов и интерфейсов системы, архитектуры системы, алгоритмов и структур данных;
- **средства проектирования баз данных**, обеспечивающие моделирование данных и генерацию схем баз данных (как правило, на языке SQL) для наиболее распространенных СУБД. К ним относятся ERwin (Logic Works), S-Designor (SDP) и DataBase Designer (ORACLE). Средства проектирования баз данных имеются также в составе CASE-средств Vantage Team Builder, Designer/2000, Silverrun и PRO-IV;
- **средства разработки приложений**. К ним относятся средства 4GL (Uniface (Compuware), JAM (JYACC), PowerBuilder (Sybase), Developer/2000 (ORACLE), New Era (Informix), SQL Windows (Gupta), Delphi (Borland)) и генераторы кодов, входящие в состав Vantage Team Builder, PRO-IV и частично - в Silverrun;
- **средства реинжиниринга**, обеспечивающие анализ программных кодов и схем баз данных и формирование на их основе различных моделей и проектных спецификаций. Средства анализа схем БД и формирования ERD входят в состав Vantage Team Builder, PRO-IV, Silverrun, Designer/2000, ERwin и S-Designor. В области анализа программных кодов наибольшее распространение получают объектно-ориентированные CASE-средства, обеспечивающие реинжиниринг программ на языке C++ (Rational Rose (Rational Software), Object Team (Cayenne)).

Вспомогательные типы включают:

- **средства планирования и управления проектом** (SE Companion, Microsoft Project);
- **средства конфигурационного управления** (PVCS (Intersolv));
- **средства тестирования** (Quality Works (Segue Software));
- **средства документирования** (SoDA (Rational Software)).

2 Методологии и технологии проектирования ИС

2.1 Общие требования к методологии и технологии

Методологии, технологии и инструментальные средства проектирования (CASE-средства) составляют основу проекта любой ИС. Методология реализуется через конкретные технологии и поддерживающие их стандарты, методики и инструментальные средства, которые обеспечивают выполнение процессов ЖЦ.

Технология проектирования определяется как совокупность трех составляющих:

- **пошаговой процедуры**, определяющей последовательность технологических операций проектирования;
- **критериев и правил**, используемых для оценки результатов выполнения технологических операций;
- **нотаций** (графических и текстовых средств), используемых для описания проектируемой системы.

Технологические инструкции, составляющие основное содержание технологии, должны состоять из описания последовательности технологических операций, условий, в зависимости от которых выполняется та или иная операция, и описаний самих операций.

Технология проектирования, разработки и сопровождения ИС должна удовлетворять следующим общим требованиям:

- технология должна поддерживать полный ЖЦ ПО;
- технология должна обеспечивать гарантированное достижение целей разработки ИС с заданным качеством и в установленное время;
- технология должна обеспечивать возможность выполнения крупных проектов в виде подсистем (т.е. возможность декомпозиции проекта на составные части, разрабатываемые группами исполнителей ограниченной численности с последующей интеграцией составных частей);
- технология должна обеспечивать возможность ведения работ по проектированию отдельных подсистем небольшими группами (3-7 человек);
- технология должна обеспечивать минимальное время получения работоспособной ИС (т.е. сроки реализации отдельных подсистем ИС должны быть минимальными);
- технология должна предусматривать возможность управления конфигурацией проекта, ведения версий проекта и его составляющих, возможность автоматического выпуска проектной документации и синхронизацию ее версий с версиями проекта;
- технология должна обеспечивать независимость выполняемых проектных решений от средств реализации ИС (систем управления базами данных (СУБД), операционных систем, языков и систем программирования);
- технология должна быть поддержана комплексом согласованных CASE-средств, обеспечивающих автоматизацию процессов, выполняемых на всех стадиях ЖЦ.

Реальное применение любой технологии проектирования, разработки и сопровождения ИС в конкретной организации и конкретном проекте невозможно без выработки ряда стандартов (правил, соглашений), которые должны соблюдаться всеми участниками проекта. К таким стандартам относятся:

- **стандарт проектирования.** Он должен устанавливать: набор необходимых моделей (диаграмм) на каждой стадии проектирования и степень их детализации; правила фиксации проектных решений на диаграммах (правила именования объектов, набор атрибутов для всех объектов и правила их заполнения на каждой стадии, правила оформления диаграмм); правила фиксации проектных решений на диаграммах (правила именования объектов, набор атрибутов для всех объектов и правила их заполнения на каждой стадии, правила оформления диаграмм); требования к конфигурации рабочих мест разработчиков; механизм обеспечения совместной работы над проектом (правила интеграции подсистем проекта, правила поддержания проекта в одинаковом для всех разработчиков состоянии, правила проверки проектных решений на непротиворечивость).
- **стандарт оформления проектной документации.** Стандарт должен устанавливать: комплектность, состав и структуру документации на каждой стадии проектирования и требования к ее оформлению; правила подготовки, рассмотрения, согласования и утверждения документации с указанием предельных сроков для каждой стадии; требования к настройке издательской системы, используемой в качестве встроенного средства подготовки документации; требования к настройке CASE-средств для обеспечения подготовки документации в соответствии с установленными требованиями.
- **стандарт пользовательского интерфейса.** Стандарт должен устанавливать: правила оформления экранов (шрифты и цветовая палитра), состав и расположение окон и элементов управления; правила использования клавиатуры и мыши; правила оформления текстов помощи; перечень стандартных сообщений; правила обработки реакции пользователя.

3 Методология RAD

Одним из возможных подходов к разработке ПО в рамках спиральной модели ЖЦ является получившая в последнее время широкое распространение методология быстрой разработки приложений RAD (Rapid Application Development). Под этим термином обычно понимается процесс разработки ПО, содержащий 3 элемента:

- **небольшую команду программистов** (от 2 до 10 человек);
- **короткий**, но тщательно проработанный **производственный график** (от 2 до 6 мес.);
- **повторяющийся цикл**, при котором разработчики, по мере того, как приложение начинает обретать форму, запрашивают и реализуют в продукте требования, полученные через взаимодействие с заказчиком.

Команда разработчиков должна представлять из себя группу профессионалов, имеющих опыт в анализе, проектировании, генерации кода и тестировании ПО с использованием CASE-средств. Члены коллектива должны также уметь трансформировать в рабочие прототипы предложения конечных пользователей.

Жизненный цикл ПО по методологии RAD состоит из четырех фаз:

- **фаза анализа и планирования требований.** На этой фазе пользователи системы определяют функции, которые она должна выполнять, выделяют наиболее

приоритетные из них, требующие проработки в первую очередь, описывают информационные потребности. Определение требований выполняется в основном силами пользователей под руководством специалистов-разработчиков. Ограничивается масштаб проекта, определяются временные рамки для каждой из последующих фаз. Кроме того, определяется сама возможность реализации данного проекта в установленных рамках финансирования, на данных аппаратных средствах и т.п. Результатом данной фазы должны быть список и приоритетность функций будущей ИС, предварительные функциональные и информационные модели ИС.

- **фаза проектирования.** Часть пользователей принимает участие в техническом проектировании системы под руководством специалистов-разработчиков. CASE-средства используются для быстрого получения работающих прототипов приложений. Пользователи, непосредственно взаимодействуя с ними, уточняют и дополняют требования к системе, которые не были выявлены на предыдущей фазе. Более подробно рассматриваются процессы системы. Анализируется и, при необходимости, корректируется функциональная модель. Каждый процесс рассматривается детально. При необходимости для каждого элементарного процесса создается частичный прототип: экран, диалог, отчет, устраняющий неясности или неоднозначности. Определяются требования разграничения доступа к данным. На этой же фазе происходит определение набора необходимой документации. После детального определения состава процессов оценивается количество функциональных элементов разрабатываемой системы и принимается решение о разделении ИС на подсистемы, поддающиеся реализации одной командой разработчиков за приемлемое для RAD-проектов время - порядка 60 - 90 дней. С использованием CASE-средств проект распределяется между различными командами (делится функциональная модель). Результатом данной фазы должны быть: общая информационная модель системы, функциональные модели системы в целом и подсистем, реализуемых отдельными командами разработчиков, точно определенные с помощью CASE-средства интерфейсы между автономно разрабатываемыми подсистемами, построенные прототипы экранов, отчетов, диалогов. Все модели и прототипы должны быть получены с применением тех CASE-средств, которые будут использоваться в дальнейшем при построении системы. Данное требование вызвано тем, что в традиционном подходе при передаче информации о проекте с этапа на этап может произойти фактически неконтролируемое искажение данных. Применение единой среды хранения информации о проекте позволяет избежать этой опасности. В отличие от традиционного подхода, при котором использовались специфические средства прототипирования, не предназначенные для построения реальных приложений, а прототипы выбрасывались после того, как выполняли задачу устранения неясностей в проекте, в подходе RAD каждый прототип развивается в часть будущей системы. Таким образом, на следующую фазу передается более полная и полезная информация.
- **фаза построения.** На фазе построения выполняется непосредственно сама быстрая разработка приложения. На данной фазе разработчики производят итеративное построение реальной системы на основе полученных в предыдущей фазе моделей, а также требований нефункционального характера. Программный код частично формируется при помощи автоматических генераторов, получающих информацию непосредственно из репозитория CASE-средств. Конечные пользователи на этой фазе оценивают получаемые результаты и вносят коррективы, если в процессе разработки система перестает удовлетворять определенным ранее требованиям. Тестирование системы осуществляется непосредственно в процессе разработки. После окончания работ каждой отдельной командой разработчиков производится постепенная интеграция данной части системы с остальными, формируется полный программный код, выполняется тестирование совместной работы данной части

приложения с остальными, а затем тестирование системы в целом. Завершается физическое проектирование системы: определяется необходимость распределения данных, производится анализ использования данных, производится физическое проектирование базы данных, определяются требования к аппаратным ресурсам, определяются способы увеличения производительности, завершается разработка документации проекта. Результатом фазы является готовая система, удовлетворяющая всем согласованным требованиям.

- **фаза внедрения.** На фазе внедрения производится обучение пользователей, организационные изменения и параллельно с внедрением новой системы осуществляется работа с существующей системой (до полного внедрения новой). Так как фаза построения достаточно непродолжительна, планирование и подготовка к внедрению должны начинаться заранее, как правило, на этапе проектирования системы. Приведенная схема разработки ИС не является абсолютной. Возможны различные варианты, зависящие, например, от начальных условий, в которых ведется разработка: разрабатывается совершенно новая система; уже было проведено обследование предприятия и существует модель его деятельности; на предприятии уже существует некоторая ИС, которая может быть использована в качестве начального прототипа или должна быть интегрирована с разрабатываемой.

Методология RAD, как и любая другая, не может претендовать на универсальность, она хороша в первую очередь для относительно небольших проектов, разрабатываемых для конкретного заказчика. Если же разрабатывается типовая система, которая не является законченным продуктом, а представляет собой комплекс типовых компонент (централизованно сопровождаемых, адаптируемых к программно-техническим платформам, СУБД, средствам телекоммуникации, организационно-экономическим особенностям объектов внедрения и интегрируемых с существующими разработками) на первый план выступают такие показатели проекта, как управляемость и качество, которые могут войти в противоречие с простотой и скоростью разработки. Для таких проектов необходимы высокий уровень планирования и жесткая дисциплина проектирования, строгое следование заранее разработанным протоколам и интерфейсам, что снижает скорость разработки.

Методология RAD неприменима для построения сложных расчетных программ, операционных систем или программ управления космическими кораблями, т.е. программ, требующих написания большого объема (сотни тысяч строк) уникального кода.

Не подходят для разработки по методологии RAD приложения, в которых отсутствует ярко выраженная интерфейсная часть, наглядно определяющая логику работы системы (например, приложения реального времени). А также приложения, от которых зависит безопасность людей (например, управление самолетом или атомной электростанцией), так как итеративный подход предполагает, что первые несколько версий наверняка не будут полностью работоспособны, что в данном случае исключается.

Оценка размера приложений производится на основе так называемых функциональных элементов (экраны, сообщения, отчеты, файлы и т.п.) Подобная метрика не зависит от языка программирования, на котором ведется разработка. Размер приложения, которое может быть выполнено по методологии RAD, для хорошо отлаженной среды разработки ИС с максимальным повторным использованием программных компонентов, определяется следующим образом:

- менее 1000 функциональных элементов - один человек;
- 1000-4000 функциональных элементов - одна команда разработчиков;
- более 4000 функциональных элементов – несколько команд из расчета 4000 функциональных элементов на одну команду разработчиков.

Основные принципы методологии RAD:

- разработка приложений итерациями;
- необязательность полного завершения работ на каждом из этапов жизненного цикла;
- обязательное вовлечение пользователей в процесс разработки ИС;
- необходимое применение CASE-средств, обеспечивающих целостность проекта;
- применение средств управления конфигурацией, облегчающих внесение изменений в проект и сопровождение готовой системы;
- необходимое использование генераторов кода;
- использование прототипирования, позволяющее полнее выяснить и удовлетворить потребности конечного пользователя;
- тестирование и развитие проекта, осуществляемые одновременно с разработкой;
- ведение разработки немногочисленной хорошо управляемой командой профессионалов;
- грамотное руководство разработкой системы, четкое планирование и контроль выполнения работ.

4 Сущность структурного подхода

Сущность структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. При разработке системы "снизу-вверх" от отдельных задач ко всей системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов.

Все наиболее распространенные методологии структурного подхода базируются на ряде общих принципов. В качестве двух базовых принципов используются:

- **принцип "разделяй и властвуй"** - принцип решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения;
- **принцип иерархического упорядочивания** - принцип организации составных частей проблемы в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.

Выделение двух базовых принципов не означает, что остальные принципы являются второстепенными, поскольку игнорирование любого из них может привести к непредсказуемым последствиям (в том числе и к провалу всего проекта). Основными из этих принципов являются:

- **принцип абстрагирования** - заключается в выделении существенных аспектов системы и отвлечения от несущественных;
- **принцип формализации** - заключается в необходимости строгого методического подхода к решению проблемы;
- **принцип непротиворечивости** - заключается в обоснованности и согласованности элементов;
- **принцип структурирования данных** - заключается в том, что данные должны быть структурированы и иерархически организованы.

В структурном анализе используются в основном две группы средств, иллюстрирующих функции, выполняемые системой и отношения между данными. Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными, среди которых являются следующие:

- **SADT** (*Structured Analysis and Design Technique*) модели и соответствующие функциональные диаграммы;
- **DFD** (*Data Flow Diagrams*) диаграммы потоков данных;
- **ERD** (*Entity-Relationship Diagrams*) диаграммы "сущность-связь".

На стадии проектирования ИС модели расширяются, уточняются и дополняются диаграммами, отражающими структуру программного обеспечения: архитектуру ПО, структурные схемы программ и диаграммы экранных форм.

Перечисленные модели в совокупности дают полное описание ИС независимо от того, является ли она существующей или вновь разрабатываемой. Состав диаграмм в каждом конкретном случае зависит от необходимой полноты описания системы.

4.1 Методология функционального моделирования SADT

Методология **SADT** разработана *Дугласом Россом*. На ее основе разработана, в частности, известная методология **IDEF0** (*Icam DEFinition*), которая является основной частью программы **ICAM** (Интеграция компьютерных и промышленных технологий), проводимой по инициативе **BBC США**.

Методология **SADT** представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель **SADT** отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями. Основные элементы этой методологии основываются на следующих концепциях:

- **графическое представление блочного моделирования.** Графика блоков и дуг **SADT**-диаграммы отображает функцию в виде блока, а интерфейсы входа/выхода представляются дугами, соответственно входящими в блок и выходящими из него. Взаимодействие блоков друг с другом описываются посредством интерфейсных дуг, выражающих "ограничения", которые в свою очередь определяют, когда и каким образом функции выполняются и управляются;
- **строгость и точность.** Выполнение правил **SADT** требует достаточной строгости и точности, не накладывая в то же время чрезмерных ограничений на действия аналитика. Правила SADT включают:
- **ограничение количества блоков на каждом уровне декомпозиции** (правило 3-6 блоков);
- **связность диаграмм** (номера блоков);
- **уникальность меток и наименований** (отсутствие повторяющихся имен);
- **синтаксические правила для графики** (блоков и дуг);
- **разделение входов и управлений** (правило определения роли данных);
- **отделение организации от функций**, т.е. исключение влияния организационной структуры на функциональную модель.

Методология **SADT** может использоваться для моделирования широкого круга систем и определения требований и функций, а затем для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции. Для уже существующих систем **SADT** может быть использована для анализа функций, выполняемых системой, а также для указания механизмов, посредством которых они осуществляются.

Результатом применения методологии *SADT* является модель, которая состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга. Диаграммы - главные компоненты модели, все функции ИС и интерфейсы на них представлены как блоки и дуги. Место соединения дуги с блоком определяет тип интерфейса. Управляющая информация входит в блок сверху, в то время как информация, которая подвергается обработке, показана с левой стороны блока, а результаты выхода показаны с правой стороны. Механизм (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей в блок снизу (рисунок 1).

Одной из наиболее важных особенностей методологии *SADT* является постепенное введение все больших уровней детализации по мере создания диаграмм, отображающих модель.

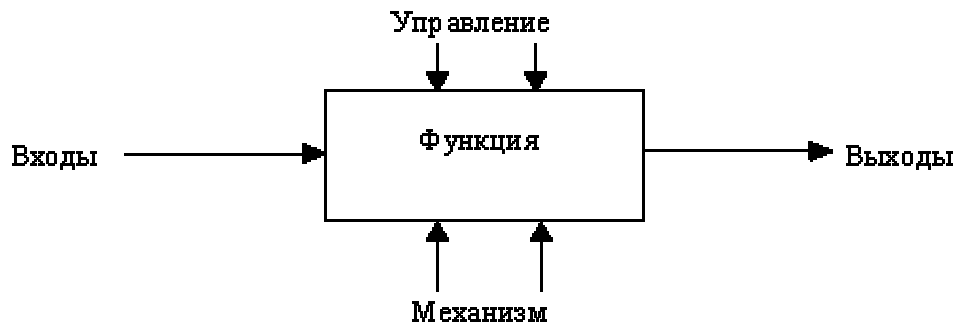


Рис. 1. Функциональный блок и интерфейсные дуги

На рисунке 2 показана структура *SADT*-модели. Каждый компонент модели может быть декомпозирован на другой диаграмме. Каждая диаграмма иллюстрирует "внутреннее строение" блока на родительской диаграмме.

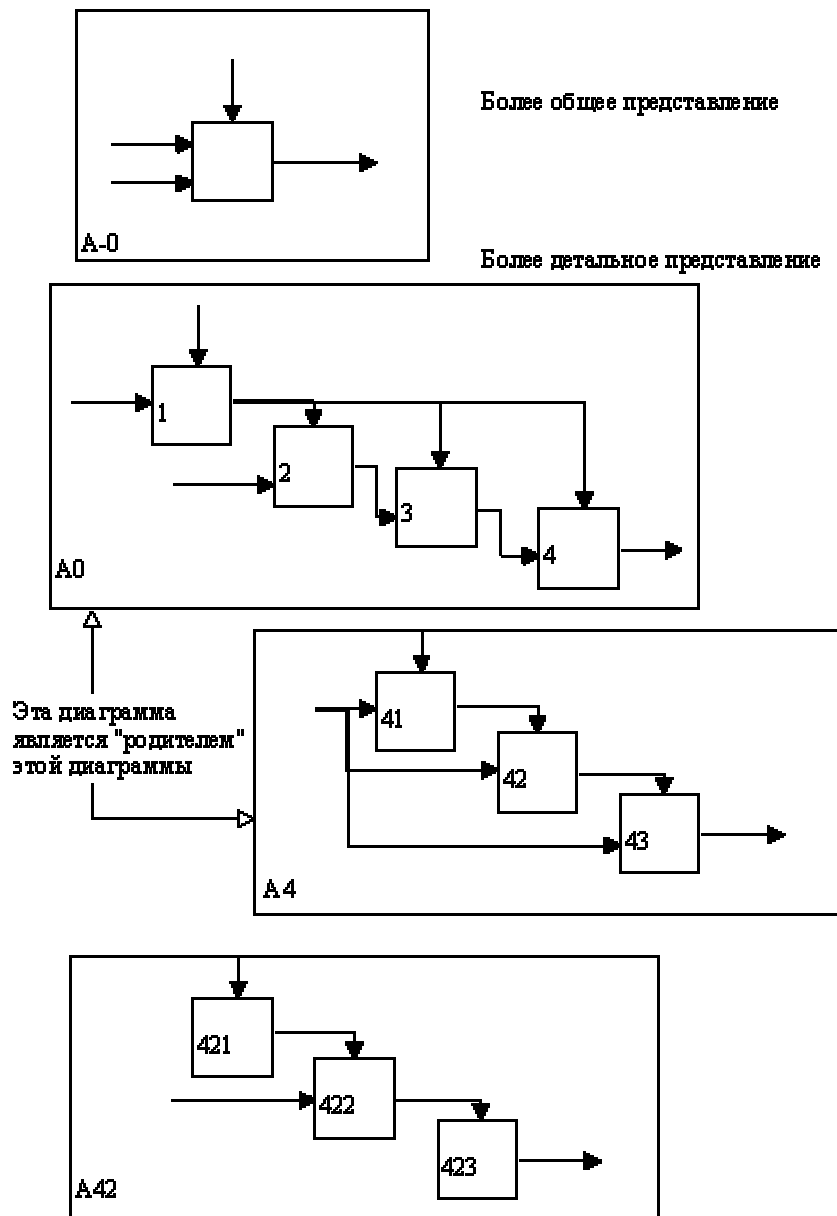


Рис. 2. Структура SADT-модели. Декомпозиция диаграмм

Построение *SADT*-модели начинается с представления всей системы в виде простейшей компоненты - одного блока и дуг, изображающих интерфейсы с функциями вне системы. Поскольку единственный блок представляет всю систему как единое целое, имя, указанное в блоке, является общим. Это верно и для интерфейсных дуг - они также представляют полный набор внешних интерфейсов системы в целом.

Затем блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами. Эти блоки представляют основные подфункции исходной функции. Данная декомпозиция выявляет полный набор подфункций, каждая из которых представлена как блок, границы которого определены интерфейсными дугами. Каждая из этих подфункций может быть декомпозирована подобным образом для более детального представления.

Во всех случаях каждая подфункция может содержать только те элементы, которые входят в исходную функцию. Кроме того, модель не может опустить какие-либо элементы,

т.е., как уже отмечалось, родительский блок и его интерфейсы обеспечивают контекст. К нему нельзя ничего добавить, и из него не может быть ничего удалено.

Модель **SADT** представляет собой серию диаграмм с сопроводительной документацией, разбивающих сложный объект на составные части, которые представлены в виде блоков. Детали каждого из основных блоков показаны в виде блоков на других диаграммах. Каждая детальная диаграмма является декомпозицией блока из более общей диаграммы. На каждом шаге декомпозиции общая диаграмма называется родительской для более детальной диаграммы.

Дуги, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются точно теми же самыми, что и дуги, входящие в диаграмму нижнего уровня и выходящие из нее, потому что блок и диаграмма представляют одну и ту же часть системы.

Некоторые дуги присоединяются к блокам диаграммы обоими концами, у других же один конец остается не присоединенным. Не присоединенные дуги соответствуют входам, управлениям и выходам родительского блока. Источник или получатель этих пограничных дуг может быть обнаружен только на родительской диаграмме. Не присоединенные концы дуг должны соответствовать дугам на исходной диаграмме. Все граничные дуги должны продолжаться на родительской диаграмме, чтобы она была полной и непротиворечивой.

На **SADT**-диаграммах явно не указываются ни последовательность, ни время. Обратные связи, итерации, продолжающиеся процессы и перекрывающиеся (по времени) функции могут быть изображены с помощью дуг. Обратные связи могут выступать в виде комментариев, замечаний, исправлений и т.д. (рисунок 3).

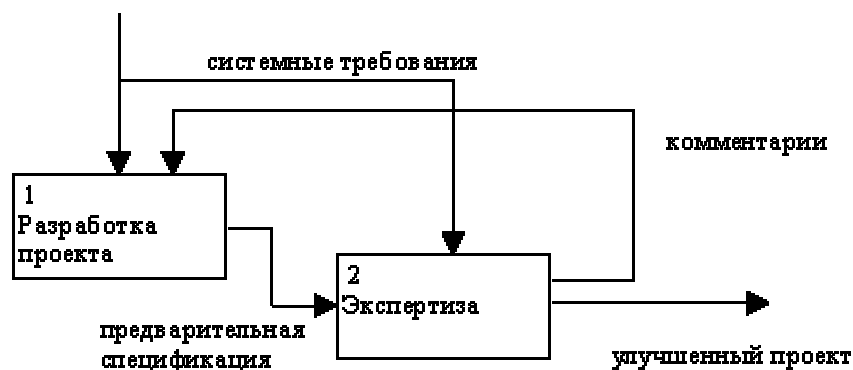


Рис. 3. Пример обратной связи

Как было отмечено, механизмы (дуги с нижней стороны) показывают средства, с помощью которых осуществляется выполнение функций. Механизм может быть человеком, компьютером или любым другим устройством, которое помогает выполнять данную функцию.

Каждый блок на диаграмме имеет свой номер. Блок любой диаграммы может быть далее описан диаграммой нижнего уровня, которая, в свою очередь, может быть далее детализирована с помощью необходимого числа диаграмм. Таким образом, формируется иерархия диаграмм. Для того, чтобы указать положение любой диаграммы или блока в иерархии, используются номера диаграмм

Одним из важных моментов при проектировании ИС с помощью методологии **SADT** является точная согласованность типов связей между функциями. Различают, по крайней мере, семь типов связывания:

0. **Тип случайной связности:** наименее желательный. Случайная связность возникает, когда конкретная связь между функциями мала или полностью отсутствует. Это относится к ситуации, когда имена данных на *SADT*-дугах в одной диаграмме имеют малую связь друг с другом. Крайний вариант этого случая показан на рисунке 4.

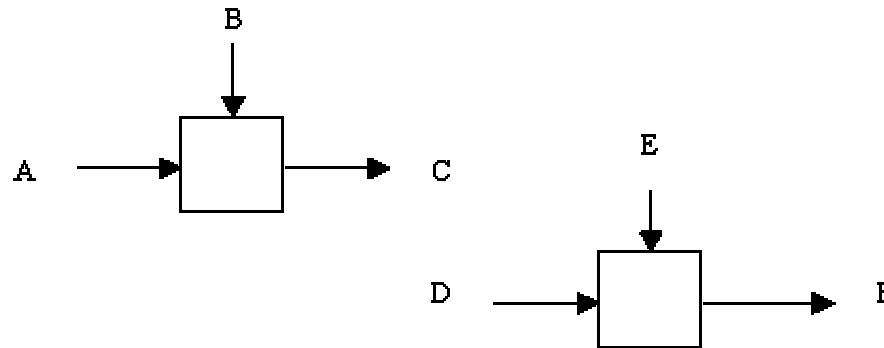


Рис. 4. Случайная связность

1. **Тип логической связности.** Логическое связывание происходит тогда, когда данные и функции собираются вместе вследствие того, что они попадают в общий класс или набор элементов, но необходимых функциональных отношений между ними не обнаруживается.
2. **Тип временной связности.** Связанные по времени элементы возникают вследствие того, что они представляют функции, связанные во времени, когда данные используются одновременно или функции включаются параллельно, а не последовательно.
3. **Тип процедурной связности.** Процедурно-связанные элементы появляются сгруппированными вместе вследствие того, что они выполняются в течение одной и той же части цикла или процесса. Пример процедурно-связанной диаграммы приведен на рисунке 5.

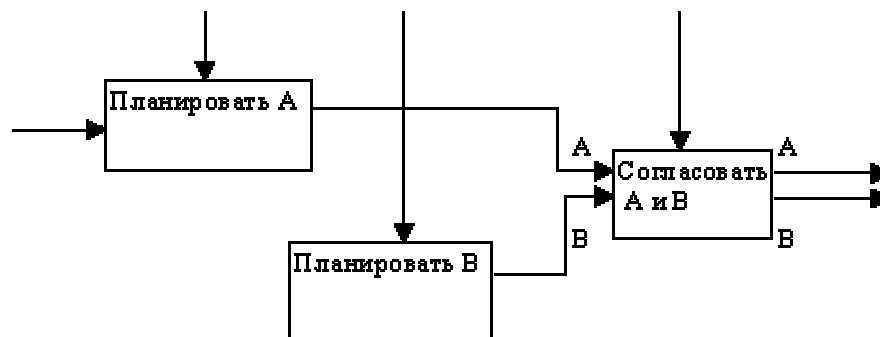


Рис. 5. Процедурная связность

4. **Тип коммуникационной связности.** Диаграммы демонстрируют коммуникационные связи, когда блоки группируются вследствие того, что они используют одни и те же входные данные и/или производят одни и те же выходные данные (рисунок 6).

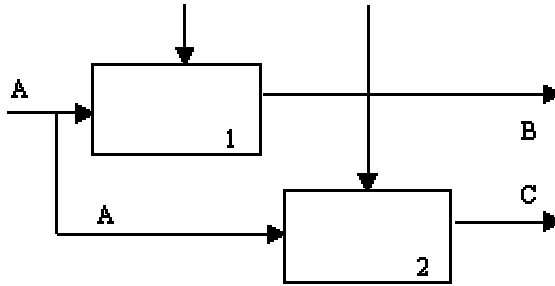


Рис. 6. Коммуникационная связность

5. **Тип последовательной связности.** На диаграммах, имеющих последовательные связи, выход одной функции служит входными данными для следующей функции. Связь между элементами на диаграмме является более тесной, чем на рассмотренных выше уровнях связок, поскольку моделируются причинно-следственные зависимости (рисунок 7).

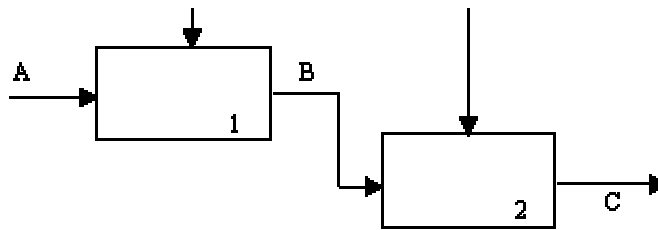


Рис. 7. Последовательная связность

6. **Тип функциональной связности.** Диаграмма отражает полную функциональную связность, при наличии полной зависимости одной функции от другой. Диаграмма, которая является чисто функциональной, не содержит чужеродных элементов, относящихся к последовательному или более слабому типу связности. Одним из способов определения функционально-связанных диаграмм является рассмотрение двух блоков, связанных через управляющие дуги, как показано на рисунке 8. В математических терминах необходимое условие для простейшего типа функциональной связности, показанной на рисунке 8, имеет следующий вид: $C = g(B) = g(f(A))$

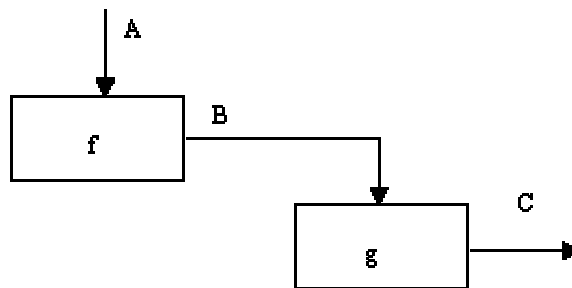


Рис. 8. Функциональная связность

Ниже в таблице представлены все рассмотренные типы связей. Важно отметить, что уровни 4-6 устанавливают типы связностей, которые разработчики считают важнейшими для получения диаграмм хорошего качества.

| Значимость | Тип связности | Для функций | Для данных |
|------------|------------------|--|--|
| 0 | Случайная | Случайная | Случайная |
| 1 | Логическая | Функции одного и того же множества или типа (например, "редактировать все входы") | Данные одного и того же множества или типа |
| 2 | Временная | Функции одного и того же периода времени (например, "операции инициализации") | Данные, используемые в каком-либо временном интервале |
| 3 | Процедурная | Функции, работающие в одной и той же фазе или итерации (например, "первый проход компилятора") | Данные, используемые во время одной и той же фазы или итерации |
| 4 | Коммуникационная | Функции, использующие одни и те же данные | Данные, на которые воздействует одна и та же деятельность |
| 5 | Последовательная | Функции, выполняющие последовательные преобразования одних и тех же данных | Данные, преобразуемые последовательными функциями |
| 6 | Функциональная | Функции, объединяемые для выполнения одной функции | Данные, связанные с одной функцией |

4.2 Моделирование потоков данных (процессов)

В основе данной методологии (методологии *Gane/Sarson*) лежит построение модели анализируемой ИС - проектируемой или реально существующей. В соответствии с методологией модель системы определяется как иерархия диаграмм потоков данных (ДПД или **DFD**), описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю. Диаграммы верхних уровней иерархии (контекстные диаграммы) определяют основные процессы или подсистемы ИС с внешними входами и выходами. Они детализируются при помощи диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут такой уровень декомпозиции, на котором процесс становится элементарными и детализировать их далее невозможно.

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. Таким образом, основными компонентами диаграмм потоков данных являются:

- **Внешняя сущность** представляет собой материальный предмет или физическое лицо, представляющее собой источник или приемник информации, например, заказчики, персонал, поставщики, клиенты, склад. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой ИС. В процессе анализа некоторые внешние сущности могут быть перенесены внутрь диаграммы анализируемой ИС, если это необходимо, или, наоборот, часть процессов ИС может быть вынесена за пределы диаграммы и представлена как внешняя сущность. Внешняя сущность обозначается квадратом (рисунок 9), расположенным как бы "над" диаграммой и бросающим на нее тень, для того, чтобы можно было выделить этот символ среди других обозначений:



Рис. 9. Внешняя сущность

- **Системы и подсистемы.** При построении модели сложной ИС она может быть представлена в самом общем виде на так называемой контекстной диаграмме в виде одной системы как единого целого, либо может быть декомпозирована на ряд подсистем. Подсистема (или система) на контекстной диаграмме изображается следующим образом (рисунок 10).

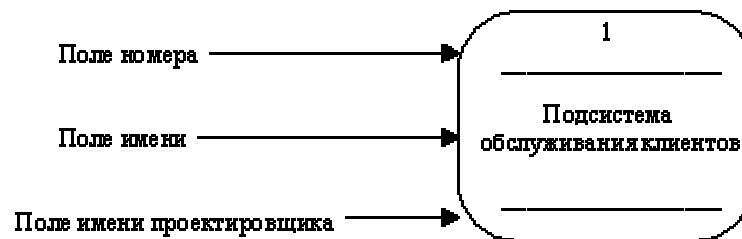


Рис. 10. Подсистема

Номер подсистемы служит для ее идентификации. В поле имени вводится наименование подсистемы в виде предложения с подлежащим и соответствующими определениями и дополнениями.

- **Процесс** представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. Физически процесс может быть реализован различными способами: это может быть подразделение организации (отдел), выполняющее обработку входных документов и выпуск отчетов, программа, аппаратно реализованное логическое устройство и т.д. Процесс на диаграмме потоков данных изображается, как показано на рисунке 11.

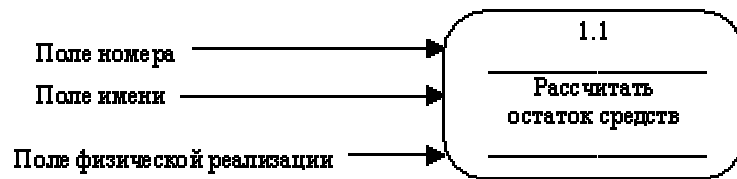


Рис. 11. Процесс

Номер процесса служит для его идентификации. В поле имени вводится наименование процесса в виде предложения с активным недвусмысленным глаголом в неопределенной форме (вычислить, рассчитать, проверить, определить, создать, получить), за которым следуют существительные в винительном падеже (например, "Ввести сведения о клиентах"; "Выдать информацию о текущих расходах"; "Проверить кредитоспособность клиента" и т.д.). Использование таких глаголов, как "обработать", "модернизировать" или "отредактировать" означает, как правило, недостаточно глубокое понимание данного процесса и требует дальнейшего анализа. Информация в поле физической реализации показывает, какое подразделение организации, программа или аппаратное устройство выполняет данный процесс.

- **Накопитель 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 микрофиши, ящика в картотеке, таблицы в оперативной памяти, файла на магнитном носителе и т.д. Накопитель данных на диаграмме потоков данных изображается, как показано на рисунке 12.

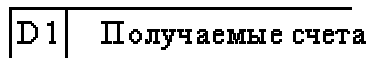


Рис. 12. Накопитель данных

Накопитель данных идентифицируется буквой "D" и произвольным числом. Имя накопителя выбирается из соображения наибольшей информативности для проектировщика.

Накопитель данных в общем случае является прообразом будущей базы данных и описание хранящихся в нем данных должно быть увязано с информационной моделью.

- **Поток данных** определяет информацию, 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. Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока (рисунок 13). Каждый поток данных имеет имя, отражающее его содержание.

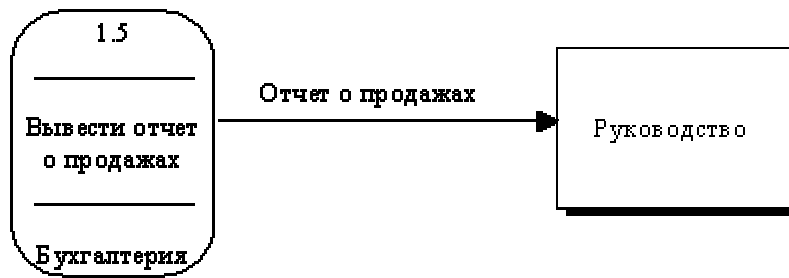


Рис. 13. Поток данных

Первым шагом при построении иерархии *ДПД* является построение контекстных диаграмм. Обычно при проектировании относительно простых ИС строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.

Если же для сложной системы ограничиться единственной контекстной диаграммой, то она будет содержать слишком большое количество источников и приемников информации, которые трудно расположить на листе бумаги нормального формата, и кроме того, единственный главный процесс не раскрывает структуры распределенной системы. Признаками сложности (в смысле контекста) могут быть:

- наличие большого количества внешних сущностей (десять и более);
- распределенная природа системы;
- многофункциональность системы с уже сложившейся или выявленной группировкой функций в отдельные подсистемы.

Для сложных ИС строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

Иерархия контекстных диаграмм определяет взаимодействие основных функциональных подсистем проектируемой ИС как между собой, так и с внешними входными и выходными потоками данных и внешними объектами (источниками и приемниками информации), с которыми взаимодействует ИС.

Разработка контекстных диаграмм решает проблему строгого определения функциональной структуры ИС на самой ранней стадии ее проектирования, что особенно важно для сложных многофункциональных систем, в разработке которых участвуют разные организации и коллективы разработчиков.

После построения контекстных диаграмм полученную модель следует проверить на полноту исходных данных об объектах системы и изолированность объектов (отсутствие информационных связей с другими объектами).

Для каждой подсистемы, присутствующей на контекстных диаграммах, выполняется ее детализация при помощи *ДПД*. Каждый процесс на *ДПД*, в свою очередь, может быть детализован при помощи *ДПД* или мини-спецификации. При детализации должны выполняться следующие правила:

- **правило балансировки** - означает, что при детализации подсистемы или процесса детализирующая диаграмма в качестве внешних источников/приемников данных может иметь только те компоненты (подсистемы, процессы, внешние сущности, накопители данных), с которыми имеет информационную связь детализируемая подсистема или процесс на родительской диаграмме;

- **правило нумерации** - означает, что при детализации процессов должна поддерживаться их иерархическая нумерация. Например, процессы, детализирующие процесс с номером 12, получают номера 12.1, 12.2, 12.3 и т.д.

Мини-спецификация (описание логики процесса) должна формулировать его основные функции таким образом, чтобы в дальнейшем специалист, выполняющий реализацию проекта, смог выполнить их или разработать соответствующую программу.

Мини-спецификация является конечной вершиной иерархии *ДПД*. Решение о завершении детализации процесса и использовании мини-спецификации принимается аналитиком исходя из следующих критериев:

- наличия у процесса относительно небольшого количества входных и выходных потоков данных (2-3 потока);
- возможности описания преобразования данных процессом в виде последовательного алгоритма;
- выполнения процессом единственной логической функции преобразования входной информации в выходную;
- возможности описания логики процесса при помощи миниспецификации небольшого объема (не более 20-30 строк).

При построении иерархии *ДПД* переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных, которые описываются при помощи структур данных. Структуры данных конструируются из элементов данных и могут содержать альтернативы, условные вхождения и итерации. Условное вхождение означает, что данный компонент может отсутствовать в структуре. Альтернатива означает, что в структуру может входить один из перечисленных элементов. Итерация означает вхождение любого числа элементов в указанном диапазоне. Для каждого элемента данных может указываться его тип (непрерывные или дискретные данные). Для непрерывных данных может указываться единица измерения (кг, см и т.п.), диапазон значений, точность представления и форма физического кодирования. Для дискретных данных может указываться таблица допустимых значений.

После построения законченной модели системы ее необходимо верифицировать (проверить на полноту и согласованность). В полной модели все ее объекты (подсистемы, процессы, потоки данных) должны быть подробно описаны и детализированы. Выявленные не детализированные объекты следует детализировать, вернувшись на предыдущие шаги разработки. В согласованной модели для всех потоков данных и накопителей данных должно выполняться правило сохранения информации: все поступающие куда-либо данные должны быть считаны, а все считываемые данные должны быть записаны.

4.3 Case-метод Баркера

Цель моделирования данных состоит в обеспечении разработчика ИС концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

Наиболее распространенным средством моделирования данных являются *диаграммы "сущность-связь" (ERD)*. С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). *ERD* непосредственно используются для проектирования реляционных баз данных.

Нотация *ERD* была впервые введена *П. Ченом (Chen)* и получила дальнейшее развитие в работах *Баркера*. Метод *Баркера* будет излагаться на примере моделирования деятельности компании по торговле автомобилями. Ниже приведены выдержки из интервью, проведенного с персоналом компании.

Главный менеджер: одна из основных обязанностей - содержание автомобильного имущества. Он должен знать, сколько заплачено за машины и каковы накладные расходы. Обладая этой информацией, он может установить нижнюю цену, за которую мог бы продать данный экземпляр. Кроме того, он несет ответственность за продавцов и ему нужно знать, кто что продает и сколько машин продал каждый из них.

Продавец: ему нужно знать, какую цену запрашивать и какова нижняя цена, за которую можно совершить сделку. Кроме того, ему нужна основная информация о машинах: год выпуска, марка, модель и т.п.

Администратор: его задача сводится к составлению контрактов, для чего нужна информация о покупателе, автомашине и продавце, поскольку именно контракты приносят продавцам вознаграждения за продажи.

Первый шаг моделирования - извлечение информации из интервью и выделение сущностей.

Сущность (Entity) - реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области, информация о котором подлежит хранению (рисунок 14).

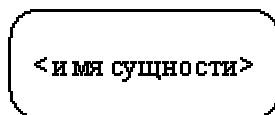


Рис. 14. Графическое изображение сущности

Каждая сущность должна обладать уникальным идентификатором. Каждый экземпляр сущности должен однозначно идентифицироваться и отличаться от всех других экземпляров данного типа сущности. Каждая сущность должна обладать некоторыми свойствами:

- каждая сущность должна иметь уникальное имя, и к одному и тому же имени должна всегда применяться одна и та же интерпретация. Одна и та же интерпретация не может применяться к различным именам, если только они не являются псевдонимами;
- сущность обладает одним или несколькими атрибутами, которые либо принадлежат сущности, либо наследуются через связь;
- сущность обладает одним или несколькими атрибутами, которые однозначно идентифицируют каждый экземпляр сущности;
- каждая сущность может обладать любым количеством связей с другими сущностями модели.

Обращаясь к приведенным выше выдержкам из интервью, видно, что сущности, которые могут быть идентифицированы с главным менеджером - это автомашины и продавцы. Продавцу важны автомашины и связанные с их продажей данные. Для администратора важны покупатели, автомашины, продавцы и контракты. Исходя из этого, выделяются 4 сущности (автомашина, продавец, покупатель, контракт), которые изображаются на диаграмме следующим образом (рисунок 15).



Рис. 15. Сущности рассматриваемого примера

Следующим шагом моделирования является идентификация связей.

Связь (Relationship) - поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Связь - это ассоциация между сущностями. При этом каждый экземпляр одной сущности, называемой родительской сущностью, ассоциирован с произвольным (в том числе нулевым) количеством экземпляров второй сущности, называемой сущностью-потомком. Каждый экземпляр сущности-потомка ассоциирован в точности с одним экземпляром сущности-родителя. Таким образом, экземпляр сущности-потомка может существовать только при существовании сущности родителя.

Связи может даваться имя, выражаемое грамматическим оборотом глагола и помещаемое возле линии связи. Имя каждой связи между двумя данными сущностями должно быть уникальным, но имена связей в модели не обязаны быть уникальными. Имя связи всегда формируется с точки зрения родителя, так что предложение может быть образовано соединением имени сущности-родителя, имени связи, выражения степени и имени сущности-потомка.

Например, связь продавца с контрактом может быть выражена следующим образом:

- продавец может получить вознаграждение за 1 или более контрактов;
- контракт должен быть инициирован ровно одним продавцом.

Степень связи и обязательность графически изображаются следующим образом (рисунок 16).



Рис. 16. Графическое изображение связей

Таким образом, 2 предложения, описывающие связь продавца с контрактом, графически будут выражены следующим образом (рисунок 17).

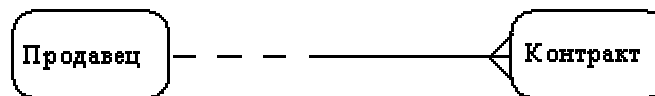


Рис. 17. Пример связи продавца с контрактом

Описав также связи остальных сущностей, получим следующую схему (рисунок 18).

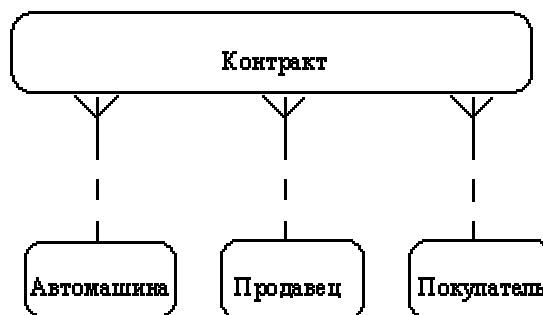


Рис. 18. Схема связей рассматриваемого примера

Последним шагом моделирования является идентификация атрибутов.

Атрибут - любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Атрибут представляет

тип характеристик или свойств, ассоциированных со множеством реальных или абстрактных объектов (людей, мест, событий, состояний, идей, пар предметов и т.д.). Экземпляр атрибута - это определенная характеристика отдельного элемента множества. Экземпляр атрибута определяется типом характеристики и ее значением, называемым значением атрибута. В **ER**-модели атрибуты ассоциируются с конкретными сущностями. Таким образом, экземпляр сущности должен обладать единственным определенным значением для ассоциированного атрибута.

Атрибут может быть либо обязательным, либо необязательным (рисунок 19).

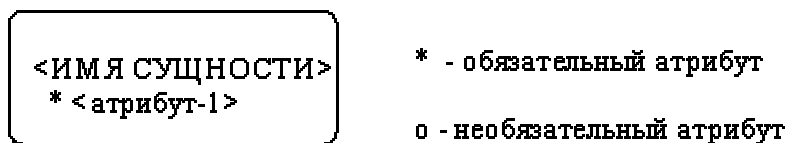


Рис. 19. Графическое изображение атрибута

Обязательность означает, что атрибут не может принимать неопределенных значений (*null values*). Атрибут может быть либо описательным (т.е. обычным дескриптором сущности), либо входить в состав уникального идентификатора (первичного ключа).

Уникальный идентификатор - это атрибут или совокупность атрибутов и/или связей, предназначенная для уникальной идентификации каждого экземпляра данного типа сущности. В случае полной идентификации каждый экземпляр данного типа сущности полностью идентифицируется своими собственными ключевыми атрибутами, в противном случае в его идентификации участвуют также атрибуты другой сущности-родителя.

Каждый атрибут идентифицируется уникальным именем, выражаемым грамматическим оборотом существительного, описывающим представляемую атрибутом характеристику. Атрибуты изображаются в виде списка имен внутри блока ассоциированной сущности, причем каждый атрибут занимает отдельную строку. Атрибуты, определяющие первичный ключ, размещаются наверху списка и выделяются знаком "#".

Каждая сущность должна обладать хотя бы одним возможным ключом. Возможный ключ сущности - это один или несколько атрибутов, чьи значения однозначно определяют каждый экземпляр сущности. При существовании нескольких возможных ключей один из них обозначается в качестве первичного ключа, а остальные - как альтернативные ключи.

С учетом имеющейся информации дополним построенную ранее диаграмму (рисунок 20).

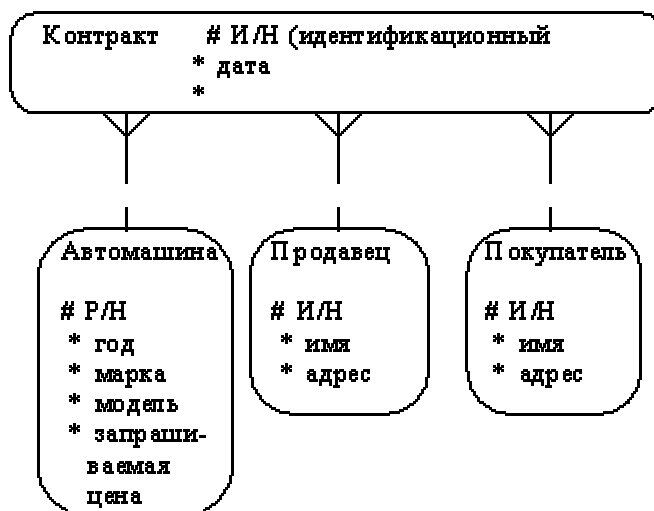


Рис. 20. Диаграмма рассматриваемого примера

Помимо перечисленных основных конструкций модель данных может содержать ряд дополнительных.

Подтипы и супертипы: одна сущность является обобщающим понятием для группы подобных сущностей (рисунок 21).

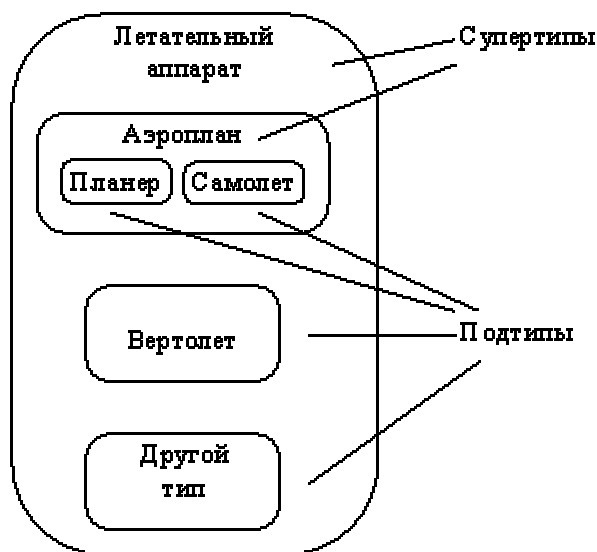


Рис. 21. Подтипы и супертипы

Взаимно исключающие связи: каждый экземпляр сущности участвует только в одной связи из группы взаимно исключающих связей (рисунок 22).

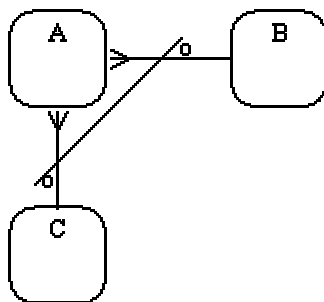


Рис. 22. Взаимно исключающие связи

Рекурсивная связь: сущность может быть связана сама с собой (рисунок 23).

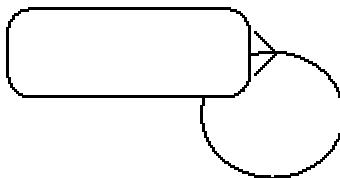


Рис. 23. Рекурсивная связь

Неперемещаемые (*non-transferrable*) связи: экземпляр сущности не может быть перенесен из одного экземпляра связи в другой (рисунок 24).

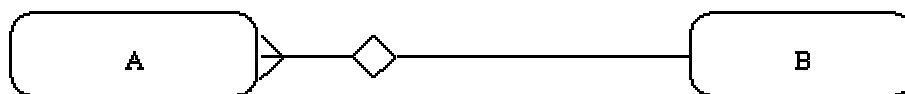


Рис. 24. Неперемещаемая связь

5 Технология внедрения CASE-средств

Приведенная в данном разделе технология базируется в основном на стандартах IEEE (IEEE - Institute of Electrical and Electronics Engineers - Институт инженеров по электротехнике и электронике). Процесс внедрения CASE-средств состоит из следующих этапов:

- определение потребностей в CASE-средствах;
- оценка и выбор CASE-средств;
- выполнение пилотного проекта;
- практическое внедрение CASE-средств.

Процесс успешного внедрения CASE-средств не ограничивается только их использованием. На самом деле он охватывает планирование и реализацию множества технических, организационных, структурных процессов, изменений в общей культуре организации, и основан на четком понимании возможностей CASE-средств.

5.1 Определение потребностей в CASE-средствах

Данный этап (рисунок 4.1) включает достижение понимания потребностей организации и технологии последующего процесса внедрения CASE-средств. Он должен привести к выделению тех областей деятельности организации, в которых применение CASE-средств может принести реальную пользу. Результатом данного этапа является документ, определяющий стратегию внедрения CASE-средств.



Рис. 4.1. Определение потребностей в CASE-средствах

5.1.1 Анализ возможностей организации

На данном этапе необходимо провести анализ направлений деятельности организации, её технологической базы, персонала и финансовых возможностей

Рассмотрим вопросы, которые могут помочь в составлении соответствующей оценки и, по существу, являются руководством по сбору информации, необходимой для определения степени готовности организации к внедрению CASE-технологии.

Общие вопросы

- используемая модель ЖЦ (каскадная или спиральная);
- используемые методы (структурные, объектно-ориентированные). Степень адаптации метода к потребностям организации; квалификация сотрудников;
- наличие документированных стандартов (формальных или неформальных) по анализу требований, спецификациям и проектированию, кодированию и тестированию;
- количественные метрики, используемые в процессе разработки ПО, их использование;
- виды документации, выпускаемой в процессе ЖЦ ПО;
- наличие группы поддержки средств проектирования.

Проекты, ведущиеся в организации

- средняя продолжительность проекта в человеко-месяцах;
- среднее количество специалистов, участвующих в проектах различных категорий (небольших, средних и крупных);
- средний размер проектов различных категорий в терминах кодовых метрик (например, в строках исходных кодов), способ измерения.

Технологическая база

- доступные вычислительные ресурсы, платформа разработки;
- уровень доступности ресурсов, узкие места, среднее время ожидания ресурсов;
- ПО, используемое в организации, и его характер (готовые программные продукты, собственные разработки);
- степень интеграции используемых программных продуктов, механизмы интеграции (существующие и планируемые);
- тип и уровень сетевых возможностей, доступных группе разработчиков;
- используемые языки программирования;
- средний процент вновь разрабатываемых, повторно используемых и реально эксплуатируемых приложений.

Персонал, готовность к возможным изменениям

- реакция сотрудников организации на внедрение новой технологии. Наличие опыта успешных или безуспешных внедрений;
- наличие лидеров, способных серьезно повлиять на отношение к новым средствам;
- наличие стремления "снизу" к совершенствованию средств и технологии;
- объем обучения, необходимого для ориентации пользователей в новой технологии;
- стабильность и уровень текучести кадров.

Готовность организации

- поддержка проекта со стороны высшего руководства;
- готовность организации к долгосрочному финансированию проекта;
- готовность организации к выделению необходимых специалистов для участия в процессе внедрения и к их обучению;
- готовность персонала к существенному изменению технологии своей работы;
- степень понимания персоналом масштаба изменений;
- готовность технических специалистов и менеджеров пойти на возможное кратковременное снижение продуктивности своей работы;
- готовность руководства к долговременному ожиданию отдачи от вложенных средств.

Оценка готовности организации к внедрению CASE-технологии должна быть откровенной и тщательной, поскольку в случае отсутствия такой готовности все усилия по внедрению потерпят крах.

5.1.2 Определение организационных потребностей

Организационные потребности следуют непосредственно из проблем организации и целей, которые она стремится достичь. Вопросы, касающиеся определения целей, потребностей и ожидаемых результатов, приведены ниже. Определение потребностей должно выполняться в сочетании с обзором рынка CASE-средств, поскольку информация о технологиях, доступных на рынке в данный момент, может оказать влияние на потребности.

Цели организации

Цели организации играют главную роль в определении ее конкретных потребностей и ожидаемых результатов. Для их понимания необходимо ответить на следующие вопросы:

- намерение организации использовать CASE-технологию для помощи в достижении определенных целей или ожиданий;
- восприятие CASE-технологии как фактора, способствующего достижению стратегических целей организации;
- наличие у организации собственной программы совершенствования процесса разработки ПО;
- восприятие инициативы внедрения CASE-технологии как части более широкомасштабного проекта по созданию среды разработки ПО.

Потребности организации

Определение потребностей организации, связанных с использованием CASE-технологии, включает анализ целей и существующих возможностей. Определению потребностей организации могут помочь ответы на следующие вопросы:

- каким образом продуктивность и качество деятельности организации сравниваются с аналогичными показателями подобных организаций (к сожалению, многие организации не располагают данными для такого сравнения);
- какие процессы ЖЦ ПО дают наилучшую (и, соответственно, наихудшую) отдачу; существуют ли конкретные процессы, которые могут быть усовершенствованы путем использования новых методов и средств.

Каждая потребность должна иметь определенный приоритет, зависящий от того, насколько критической она является для достижения успеха в организации. В конечном

счете, должно четко прослеживаться воздействие каждой функции или возможности приобретаемых средств на удовлетворение конкретных потребностей.

Результатом данного действия является формулировка потребностей с их приоритетами, которая используется на этапе оценки и выбора в качестве "пользовательских потребностей".

5.1.3 Анализ рынка CASE-средств

Потребности организации в CASE-средствах должны соразмеряться с реальной ситуацией на рынке или собственными возможностями разработки. Исследование рынка проводится путем изучения литературы по CASE-средствам, посещения конференций и семинаров, проводимых поставщиками (их перечень приведен в конце данного обзора) и пользователями CASE-средств. При проведении данного анализа необходимо выяснить возможность интеграции конкретного CASE-средства с другими средствами, используемыми (или планируемыми к использованию) организацией. Кроме того, важно получить достоверную информацию о средствах, основанную на реальном пользовательском опыте и сведениях от пользовательских групп.

5.1.4 Определение критериев успешного внедрения

Определяемые критерии должны позволять количественно оценивать степень удовлетворения каждой из потребностей, связанных с внедрением. Кроме того, по каждому критерию должно быть определено его конкретное оптимальное значение. На определенных этапах внедрения эти критерии должны анализироваться для того, чтобы определить текущую степень удовлетворения потребностей.

Обычно, большинство организаций осуществляет внедрение CASE-средств для того, чтобы повысить продуктивность процессов разработки и сопровождения ПО, а также качество результатов разработки.

Помимо продуктивности и качества, полезную информацию о состоянии внедрения CASE-средств также могут дать и другие характеристики организационных процессов и персонала:

- рейтинговые оценки уровня квалификации специалистов, связанные с использованием CASE-средств
- процент проектов, использующих CASE-средства
- согласованность проектных результатов;
- точность стоимостных и плановых оценок;
- изменчивость внешних требований;
- соблюдение стандартов организации;
- степень повторного использования существующих компонентов ПО;
- объем и виды необходимого обучения;
- типы и моменты обнаружения проектных ошибок;
- вычислительные ресурсы, используемые CASE-средствами.

5.1.5 Разработка стратегии внедрения CASE-средств

Стратегия внедрения должна обеспечивать удовлетворение потребностей и критериев, определенных ранее. Стратегия включает следующие составляющие:

- организационные потребности;
- базовые метрики, необходимые для последующего сравнения результатов;

- критерии успешного внедрения, связанные с удовлетворением организационных потребностей, включая ожидаемые результаты последовательных этапов процесса внедрения;
- подразделения организации, в которых должно выполняться внедрение CASE-средств;
- влияние, оказываемое на другие подразделения организации;
- стратегии и планы оценки и выбора, пилотного проектирования и перехода к полномасштабному внедрению;
- основные факторы риска;
- ориентировочный уровень расходов и источники финансирования процесса внедрения CASE-средств;
- ключевой персонал и другие ресурсы.

Существенное внимание должно уделяться ролям различных групп, вовлеченных в процесс таких изменений. Наиболее существенные роли включают следующие:

- спонсор (обычно из числа менеджеров высшего уровня). Данная роль является критической для поддержки проекта и обеспечения необходимого финансирования. Спонсор должен обладать четким пониманием необходимости серьезных усилий, связанных с внедрением CASE-средств, и длительности периода ожидания осязаемых результатов;
- исполнитель - обычно лицо (или группа лиц), осознающее потенциальные возможности новой технологии, пользующееся авторитетом среди технического персонала и способное возглавить процесс внедрения новой технологии;
- целевая группа - обычно включает менеджеров и технический персонал, которые будут привлечены к непосредственному использованию CASE-средств, а также специалистов, которые будут привлечены косвенно, таких, как специалисты по документированию, персонал поддержки сети и заказчики.

Существует несколько подходов к разработке стратегии внедрения CASE-средств. Относительные преимущества того или иного подхода перед другими должны рассматриваться в контексте специфики конкретной организации. Особое значение при этом придается персоналу организации и процессу разработки ПО.

Нисходящий подход основывается на том, что исследование всех типов CASE-средств и документирование процессов разработки и сопровождения ПО в данной организации должно выполняться до определения требования к CASE-средствам. При этом выполняется общий анализ процесса создания и сопровождения ПО в организации. Данный подход зачастую влечет за собой общую реорганизацию процессов создания и сопровождения ПО в той степени, в какой это связано с CASE-средствами. Результатом такой реорганизации становится крупномасштабная стратегия автоматизации процессов создания и сопровождения ПО. Данный подход рекомендуется для относительно зрелых организаций с устоявшимся процессом создания и сопровождения ПО.

Восходящий подход начинается с определения некоторого средства или типа средств, которые потенциально могут помочь организации в улучшении выполнения текущей работы. Организация может затем оценить возможное воздействие средств на процесс разработки и сопровождения ПО. Восходящий подход рекомендуется для организаций с узко специфическими потребностями в автоматизации, не нуждающихся в общем совершенствовании процессов.

Наиболее рациональная стратегия может сочетать характеристики обоих подходов. Например, нисходящие методы могут использоваться для определения стандартов качества

организации, потребностей в средствах и ожидаемых результатов, тогда как восходящие методы могут использоваться для оценки и выбора конкретных CASE-средств, разработки планов внедрения и контроля его результатов.

5.2 Оценка и выбор CASE-средств

5.2.1 Общие сведения

Модель процесса оценки и выбора, рассматриваемая ниже (рисунок 4.2), описывает наиболее общую ситуацию оценки и выбора, а также показывает зависимость между ними. Как можно видеть, оценка и выбор могут выполняться независимо друг от друга или вместе, каждый из этих процессов требует применения определенных критериев.

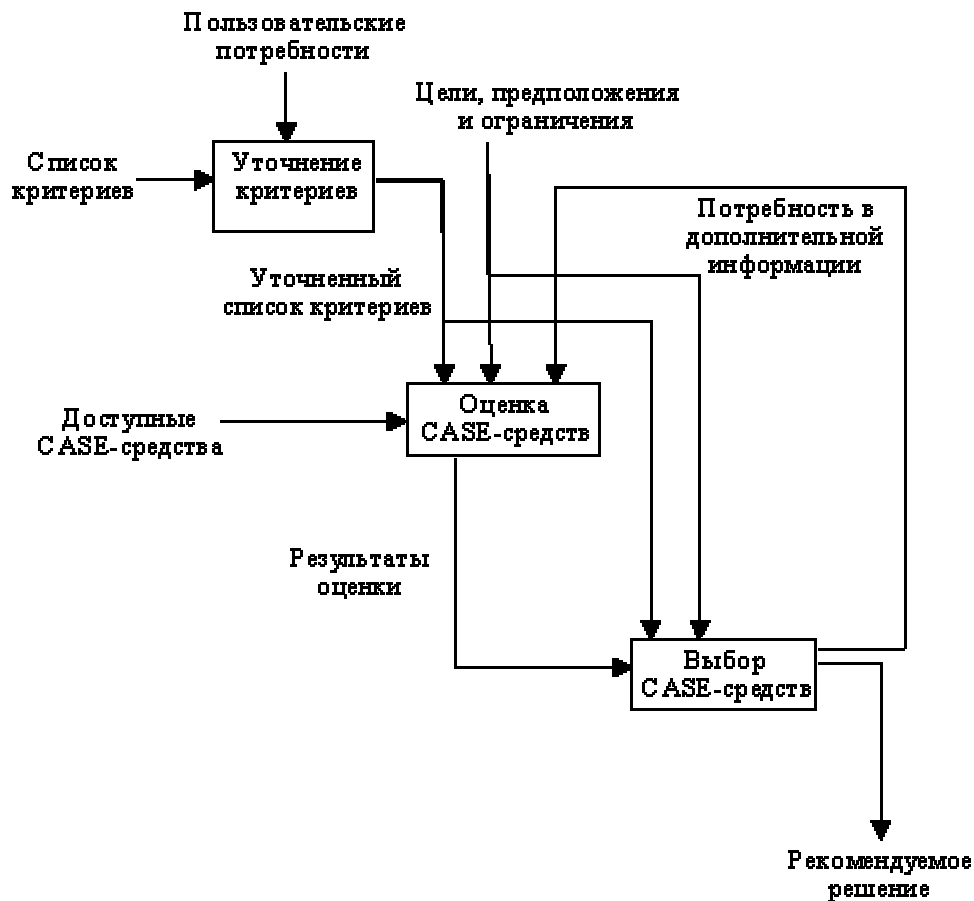


Рис. 4.2. Модель процесса оценки и выбора

Как видно из рисунка, входной информацией для процесса оценки является результаты оценки. Результаты оценки могут включать результаты предыдущих оценок. При этом не следует забывать, что набор критериев, использовавшихся при предыдущей оценке, должен быть совместимым с текущим набором. Конкретный вариант реализации процесса (оценка и выбор, оценка для будущего выбора или выбор, основанный на предыдущих оценках) определяется перечисленными выше целями.

Процесс оценки и/или выбора может быть начат только тогда, когда лицо, группа или организация полностью определила для себя конкретные потребности и формализовала их в виде количественных и качественных требований в заданной предметной области. Термин "пользовательские требования" далее означает именно такие формализованные требования.

Пользователь должен определить конкретный порядок действий и принятия решений с любыми необходимыми итерациями. Например, процесс может быть представлен в виде

дерева решений с его последовательным обходом и выбором подмножеств кандидатов для более детальной оценки. Описание последовательности действий должно определять поток данных между ними.

5.2.2 Процесс оценки

Целью процесса оценки является определение функциональности и качества CASE-средств для последующего выбора. Оценка выполняется в соответствии с конкретными критериями, ее результаты включают как объективные, так и субъективные данные по каждому средству.

Процесс оценки включает следующие действия:

- формулировка задачи оценки, включая информацию о цели и масштабах оценки;
- определение критериев оценки, вытекающее из определения задачи;
- определение средств-кандидатов;
- оценка средств-кандидатов в контексте выбранных критериев
- подготовка отчета по результатам оценки.

Одним из важнейших критериев в процессе оценки может быть потенциальная возможность интеграции каждого из средств-кандидатов с другими средствами, уже находящимися в эксплуатации или планируемыми к использованию в данной организации.

Оценка и накопление соответствующих данных может выполняться следующими способами:

- анализ CASE-средств и документации поставщика;
- опрос реальных пользователей;
- анализ результатов проектов, использовавших данные CASE-средства;
- просмотр демонстраций и опрос демонстраторов;
- выполнение тестовых примеров;
- применение CASE-средств в пилотных проектах;
- анализ любых доступных результатов предыдущих оценок.

Результаты оценки должны быть стандартным образом документированы (для облегчения последующего использования) и, при необходимости, утверждены.

5.2.3 Процесс выбора

Процессы оценки и выбора тесно взаимосвязаны друг с другом. По результатам оценки цели выбора и/или критерии выбора и их веса могут потребовать модификации. В таких случаях может потребоваться повторная оценка. Когда анализируются окончательные результаты оценки и к ним применяются критерии выбора, может быть рекомендовано приобретение CASE-средства или набора CASE-средств. Альтернативой может быть отсутствие адекватных CASE-средств, в этом случае рекомендуется разработать новое CASE-средство, модифицировать существующее или отказаться от внедрения.

Процесс выбора тесно взаимосвязан с процессом оценки и включает следующие действия:

- формулировка задач выбора, включая цели, предположения и ограничения;
- выполнение всех необходимых действий по выбору, включая определение и ранжирование критериев, определение средств-кандидатов, сбор необходимых данных и применение ранжированных критериев к результатам оценки для определения

средств с наилучшими показателями. Для многих пользователей важным критерием выбора является интегрируемость CASE-средства с существующей средой;

- выполнение необходимого количества итераций с тем, чтобы выбрать (или отвергнуть) средства, имеющие сходные показатели;
- подготовка отчета по результатам выбора.

Алгоритмы, обычно используемые для выбора, могут быть основаны на масштабе или ранге. Алгоритмы, основанные на масштабе, вычисляют единственное значение для каждого CASE-средства путем умножения веса каждого критерия на его значение (с учетом масштаба) и сложения всех произведений. CASE-средство с наивысшим результатом получает первый ранг. Алгоритмы, основанные на ранге, используют ранжирование CASE-средств - кандидатов по отдельным критериям или группам критериев в соответствии со значениями критериев в заданном масштабе. Затем, аналогично предыдущему, ранги сводятся вместе и вычисляются общие значения рангов.

При анализе результатов выбора предполагается, что процесс выбора завершен, CASE-средство выбрано и рекомендовано к использованию. Тем не менее, может потребоваться более точный анализ для определения степени зависимости значений ключевых критериев от различий в значениях характеристик CASE-средств - кандидатов. Такой анализ позволит определить, насколько результат ранжирования CASE-средств зависит от оптимальности выбора весовых коэффициентов критериев. Он также может использоваться для определения существенных различий между CASE-средствами с очень близкими значениями критериев или рангами.

Рекомендации по выбору должны быть строго обоснованы. В случае отсутствия адекватных CASE-средств, как было отмечено выше, рекомендуется разработать новое CASE-средство, модифицировать существующее или отказаться от внедрения.

5.2.4 Критерии оценки и выбора

Структура набора критериев приведена на рисунке 4.3. Каждый критерий должен быть выбран и адаптирован экспертом с учетом особенностей конкретного процесса. В большинстве случаев только некоторые из множества описанных ниже критериев оказываются приемлемыми для использования, при этом также добавляются дополнительные критерии. Выбор и уточнение набора используемых критериев является критическим шагом в процессе оценки и/или выбора.

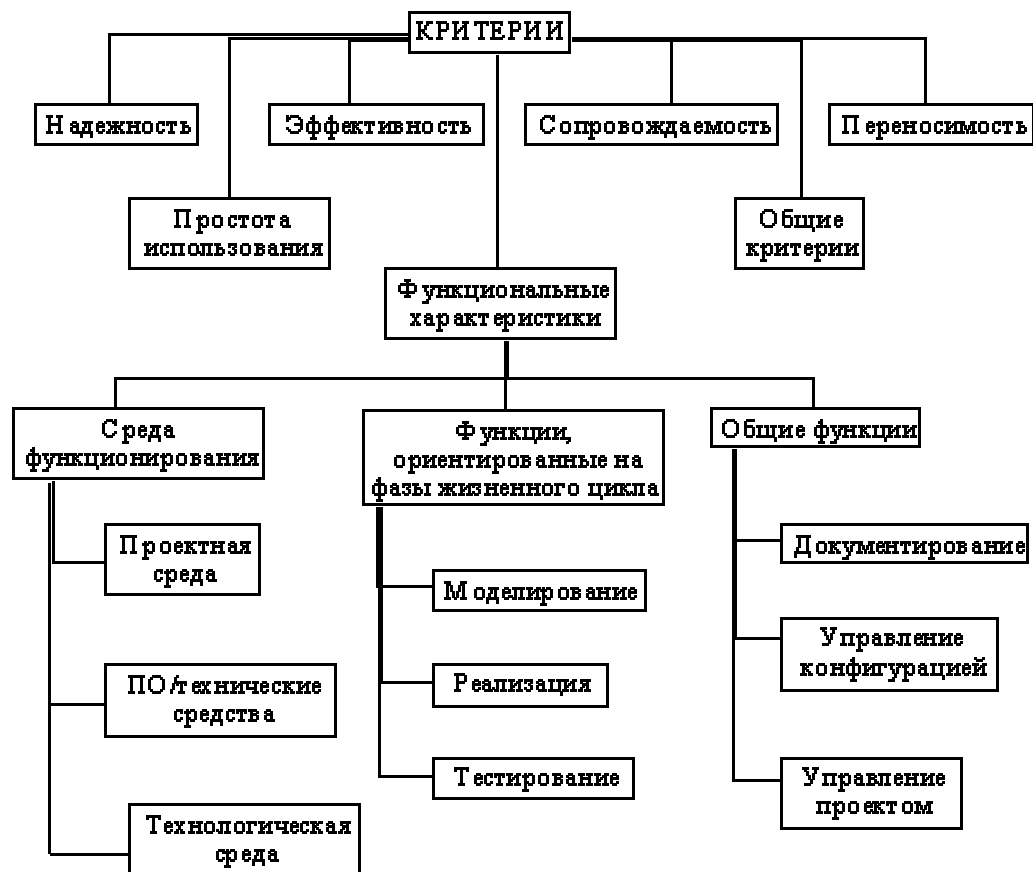


Рис. 4.3. Структура набора критериев

Для оценки по каждому из критериев надо составить список основных элементов данного критерия, определить их значимость и провести анализ в соответствии с этим списком.

5.3 Выполнение пилотного проекта

Перед полномасштабным внедрением выбранного CASE-средства в организации выполняется пилотный проект, целью которого является экспериментальная проверка правильности решений, принятых на предыдущих этапах, и подготовка к внедрению.

Пилотный проект представляет собой первоначальное, реальное использование CASE-средства в предназначенной для этого среде и обычно подразумевает более широкий масштаб использования CASE-средства по отношению к тому, который был достигнут во время оценки. Пилотный проект должен обладать многими из характеристик реальных проектов, для которых предназначено данное средство. Он преследует следующие цели:

- подтвердить достоверность результатов оценки и выбора;
- определить, действительно ли CASE-средство годится для использования в данной организации, и если да, то определить наиболее подходящую область его применения;
- собрать информацию, необходимую для разработки плана практического внедрения;
- приобрести собственный опыт использования CASE-средства.

Пилотный проект позволяет получить важную информацию, необходимую для оценки качества функционирования CASE-средства и его поддержки со стороны поставщика после того, как средство установлено.

Важной функцией пилотного проекта является принятие решения относительно приобретения или отказа от использования CASE-средства. Провал пилотного проекта

позволяет избежать более значительных и дорогостоящих неудач в дальнейшем, поскольку пилотный проект обычно связан с приобретением относительно небольшого количества лицензий и обучением узкого круга специалистов.

Первоначальное использование новой CASE-технологии в пилотном проекте должно тщательно планироваться и контролироваться. Пилотный проект включает следующие шаги (рисунок 4.4).



Рис. 4.4. Шаги пилотного проекта

5.3.1 Определение характеристик пилотного проекта

Пилотный проект должен обладать следующими характеристиками:

- *Область применения.* Чтобы облегчить окончательное определение области применения CASE-средства, предметная область пилотного проекта должна быть типичной для обычной деятельности организации. Пилотный проект должен помочь определить любую дополнительную технологию, обучение или поддержку, которые необходимы для перехода от пилотного проекта к широкомасштабному использованию средства. В рамках этих ограничений пилотный проект должен иметь небольшой, но значимый размер.
- *Масштабируемость.* Результаты, полученные в пилотном проекте, должны показать масштабируемость средства. Цель - получить четкое представление о масштабах проектов, для которых данное средство применимо.
- *Представительность.* Пилотный проект не должен быть необычным или уникальным для организации. CASE-средство должно использоваться для решения задач, относящихся к предметной области, хорошо понимаемой всей организацией.

- *Критичность.* Пилотный проект должен иметь существенную значимость, чтобы оказаться в центре внимания, но не должен быть критичным для успешной деятельности организации в целом. Необходимо осознавать, что первоначальное внедрение новой технологии подразумевает определенный риск. При выборе пилотного проекта приходится решать следующую дилемму: успех незначительного проекта может остаться незамеченным, с другой стороны, провал значимого проекта может вызвать чрезмерную критику.
- *Авторитетность.* Группа специалистов, участвующих в проекте, должна обладать высоким авторитетом, при этом результаты проекта будут всерьез восприняты остальными сотрудниками организации.
- *Характеристики проектной группы.* Проектная группа должна обладать готовностью к нововведениям, технической зрелостью и приемлемым уровнем опыта и знаний в данной технологии и предметной области. С другой стороны, группа должна отражать в миниатюру характеристики всей организации в целом.

5.3.2 Планирование пилотного проекта

План должен содержать следующую информацию:

- цели, задачи и критерии оценки;
- персонал;
- процедуры и соглашения;
- обучение;
- график и ресурсы.

Цели, задачи и критерии оценки

Ожидаемые результаты пилотного проекта должны быть четко определены. Степень соответствия этим результатам представляет собой основу для последующей оценки проекта.

Персонал

Специалисты, выбранные для участия в пилотном проекте, должны иметь соответствующий авторитет и влияние и быть сторонниками новой технологии. Группа должна включать как технических специалистов, так и менеджеров, заинтересованных в новой технологии и разбирающихся в ее использовании. Группа должна обладать высокими способностями к коммуникации, знанием особенностей организационных процессов и процедур, а также предметной области. Группа не должна, тем не менее, состоять полностью из специалистов высшего звена, она должна представлять средний уровень организации.

Процедуры и соглашения

Необходимо четко определить процедуры и соглашения, регулирующие использование CASE-средств в пилотном проекте.

Обучение

Должны быть определены виды и объем обучения, необходимого для выполнения пилотного проекта.

Поставщики CASE-средств обычно предлагают обучение использованию поставляемых ими средств.

График и ресурсы

Должен быть разработан график, включающий ресурсы и сроки (этапы) проведения работ. Ресурсы включают персонал, технические средства, ПО и финансирование.

5.3.3 Выполнение пилотного проекта

Пилотный проект должен выполняться в соответствии с планом. Организационная деятельность, связанная с выполнением пилотного проекта и подготовкой отчетов, должна выполняться в установленном порядке. Пилотная природа проекта требует специального внимания к вопросам приобретения, поддержки, экспертизы и обновления версий.

Приобретение, установка и интеграция

Процесс приобретения может включать подготовку контракта, переговоры, лицензирование.

После того, как процесс приобретения завершен, средство должно быть установлено, оттестировано и принято в эксплуатацию. Особое внимание необходимо уделить соблюдению всех требований поставщика к параметрам среды функционирования CASE-средства.

После завершения приемки может потребоваться настройка и интеграция. Настройка может включать модификацию интерфейсов, связанную с требованиями специалистов проектной группы, а также установкой прав доступа и привилегий. Если новое средство должно использоваться в совокупности с некоторыми другими средствами, необходимо определить взаимодействие средств и требуемую интеграцию.

Поддержка

Доступная поддержка должна включать (по соглашению) "горячую линию" поставщика и поддержку местного поставщика, поддержку в самой организации, контакты с опытными пользователями в других организациях и участие в работе групп пользователей.

Внутренняя поддержка должна осуществляться специалистами, знакомыми с установкой средств и работой с ними.

Периодические экспертизы

Обычные процедуры экспертизы проектов, существующие в организации, должны выполняться и для пилотного проекта, при этом особое внимание должно уделяться именно пилотным аспектам проекта. Помимо этого, результаты экспертиз должны служить мерой успешного использования CASE-средств.

Обновление версий

Пользователи CASE-средства могут ожидать периодического обновления версий со стороны поставщика в течение выполнения пилотного проекта.

Оценка пилотного проекта

В процессе оценки пилотного проекта организация должна определить свою позицию по следующим трем вопросам:

- Целесообразно ли внедрять CASE-средство ?
- Какие конкретные особенности пилотного проекта привели к его успеху (или неудаче) ?
- Какие проекты или подразделения в организации могли бы получить выгоду от использования средств ?

5.3.4 Принятие решения о внедрении

Возможным решением должно быть одно из следующих:

- *Внедрить средство.* В этом случае рекомендуемый масштаб внедрения должен быть определен в терминах структурных подразделений и предметной области.
- *Выполнить дополнительный пилотный проект.* Такой вариант должен рассматриваться только в том случае, если остались конкретные неразрешенные

вопросы относительно внедрения CASE-средства в организации. Новый пилотный проект должен быть таким, чтобы ответить на эти вопросы.

- *Отказаться от средства.* В этом случае причины отказа от конкретного средства должны быть определены в терминах потребностей организации или критериев, которые остались неудовлетворенными. Перед тем, как продолжить деятельность по внедрению CASE-средств, потребности организации должны быть пересмотрены на предмет своей обоснованности.
- *Отказаться от использования CASE-средств вообще.* Пилотный проект может показать, что организация либо не готова к внедрению CASE-средств, либо автоматизация данного аспекта процесса создания и сопровождения ПО не дает никакого эффекта для организации. В этом случае причины отказа от CASE-средств должны быть также определены в терминах потребностей организации или критериев, которые остались неудовлетворенными. При этом необходимо понимать отличие этого варианта от предыдущего, связанного с недостатками конкретного средства.

Результатом данного этапа является документ, в котором обсуждаются результаты пилотного проекта и детализируются решения по внедрению.

5.4 Переход к практическому использованию CASE-средств

Процесс перехода к практическому использованию CASE-средств начинается с разработки и последующей реализации плана перехода. План перехода должен включать:

- информацию относительно целей, критериев оценки, графика и возможных рисков, связанных с реализацией плана;
- информацию относительно приобретения, установки и настройки средства (совокупность программных компонентов, документации и обучения, которые необходимо приобретать для каждой отдельной платформы; механизм получения новых версий; настройка средства, необходимая для выполнения существующих в организации процедур и соглашений; лицо или подразделение, ответственное за установку, интеграцию, настройку и эксплуатацию средства; план конвертирования данных и снятия старых средств с эксплуатации). Задачи приобретения, установки и настройки должны быть как можно быстрее переданы из группы пилотного проекта в существующую службу системной поддержки ПО организации;
- информацию относительно интеграции средства с существующими средствами, включая как интеграцию CASE-средств друг с другом, так и их интеграцию в процессы разработки и эксплуатации ПО, существующие в организации;
- ожидаемые потребности в обучении и ресурсы, используемые в течение и после завершения процесса перехода;
- определение стандартных процедур использования средств, включающих следующее:
 - стандарты использования средств;
 - руководства по моделированию и проектированию;
 - соглашения по присвоению имен;
 - процедуры контроля качества и процессов приемки, включая расписание экспертиз и используемые методологии;
 - процедуры резервного копирования, защиты мастер-копий и конфигурирования базы данных;
 - процедуры интеграции с существующими средствами и базами данных;
 - процедуры совместного использования данных и контроля целостности БД;
 - стандарты и процедуры обеспечения секретности;

- стандарты документирования.

Стандарты использования CASE-средств, выработанные во время пилотного проекта, должны использоваться в качестве отправной точки для разработки более полного набора стандартов использования средств в данной организации. При этом должен учитываться опыт участников пилотного проекта.

Реализация плана перехода требует постоянного мониторинга использования CASE-средств, обеспечения текущей поддержки, сопровождения и обновления средств по мере необходимости.

Достигнутые результаты должны периодически подвергаться экспертизе, план перехода должен корректироваться при необходимости. Постоянное внимание должно уделяться степени удовлетворения потребностей организации и критериев успешного внедрения CASE-средств. Периодические экспертизы должны продолжаться и после завершения процесса внедрения. Такие экспертизы могут анализировать метрики и другую информацию, получаемую в процессе работы с CASE-средствами, чтобы определять, насколько хорошо они продолжают выполнять требуемые функции. Такие экспертизы могут также указать на необходимость дополнительной модификации процессов.

Необходимо определить источники для текущей поддержки CASE-средств. Такая поддержка необходима для следующего:

- ответов на вопросы, связанные с использованием средств;
- передачи информации о достигнутых успехах и полученных уроках другим специалистам организации;
- модификации и совершенствования стандартов, соглашений и процедур, связанных с использованием средства;
- интеграции новых средств с существующими и сопровождения интегрированных средств по мере появления новых версий;
- помощи новым сотрудникам в освоении средств и связанных с ними процедур;
- планирования и контроля обновления версий;
- планирования внедрения новых возможностей средств в организационные процессы.

Результатом данного шага является внедрение CASE-средств в повседневную практику организации, при этом больше не требуется какого-либо специального планирования. Кроме того, поддержка CASE-средств включается в план текущей поддержки ПО в данной организации.

Литература:

- А.М. Вендров “Современные методы и средства проектирования информационных систем “
- Ю.В. Новоженев, М.З. Звонкин, Н.Н. Тимонин “ Объектно-ориентированные CASE-средства”
- Г. Калянов “Номенклатура CASE-средств и виды проектной деятельности”
- А.М. Вендров “ Практические рекомендации по освоению и внедрению CASE-средств”
- С.В. Горин, А.Ю. Тандоев “ Применение CASE-средства ERwin 2.0 для информационного моделирования в системах обработки данных ”