АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ ЭИС (CASE-ТЕХНОЛОГИЯ)

# Основные понятия

Термин CASE (Computer Aided System/Software Engineering) используется в довольно широком смысле. Первоначальное значение термина CASE, ограниченное вопросами автоматизации разработки только лишь программного обеспечения, в настоящее время приобрело новый смысл, охватывающий процесс разработки сложных ЭИС в целом.

CASE-технология (Computer Aided Software Engineering) представляет собой совокупность методологий анализа, проектирования, разработки и сопровождения сложных систем программного обеспечения, поддержанную комплексом взаимоувязанных средств автоматизации.

С самого начала CASE-технологии развивались с целью преодоления ограничений при использовании структурной методологии проектирования (сложности понимания, высокой трудоемкости и стоимости использования, трудности внесения изменений в проектные спецификации и т.д.) за счет ее автоматизации и интеграции поддерживающих средств. Таким образом, CASE-технологии не могут считаться самостоятельными, они только обеспечивают, как минимум, высокую эффективность их применения, а в некоторых случаях и принципиальную возможность применения соответствующей методологии. Большинство существующих CASE-систем ориентировано на автоматизацию проектирования программного обеспечения и основано на методологиях структурного (в основном) или объектно-ориентированного проектирования и программирования, использующих спецификации в виде диаграмм или текстов для описания системных требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств. В последнее время стали появляться CASE-системы, уделяющие основное внимание проблемам спецификации и моделирования технических средств.

Наибольшая потребность в использовании CASE-систем испытывается на начальных этапах разработки, а именно на этапах анализа и спецификации требований к ЭИС. Это объясняется тем, что цена ошибок, допущенных на начальных этапах, на несколько порядков превышает цену ошибок, выявленных на более поздних этапах разработки.

Появлению CASE-технологии предшествовали исследования в области методологии программирования. Программирование обрело черты системного подхода с разработкой и внедрением языков высокого уровня, методов структурного и модульного программирования, языков проектирования и средств их поддержки, формальных и неформальных языков описания системных требований и спецификаций и т.д. Кроме того, этому способствовали перечисленные ниже факторы:

* подготовка аналитиков и программистов, восприимчивых к концепциям модульного и структурного программирования;
* широкое внедрение и постоянный рост производительности персональных ЭВМ, позволяющих использовать эффективные графические средства и автоматизировать большинство эта­пов проектирования;
* внедрение сетевой технологии, предоставившей возможность объединения усилий отдельных исполнителей в единый процесс проектирования путем использования разделяемой базы данных, содержащей необходимую информацию о проекте.

Преимущества CASE-технологии по сравнению с традиционной технологией оригинального проектирования сводятся к следующему:

* улучшение качества разрабатываемого программного приложения за счет средств автоматического контроля и генерации;
* возможность повторного использования компонентов разработки;
* поддержание адаптивности и сопровождения ЭИС;
* снижение времени создания системы, что позволяет на ранних стадиях проектирования получить прототип будущей си­стемы и оценить его;
* освобождение разработчиков от рутинной работы по документированию проекта, так как при этом используется встро­енный документатор;
* возможность коллективной разработки ЭИС в режиме реального времени.

## Содержание CASE-технологии:

* Методология определяет шаги и этапность реализации проекта, а также правила использования методов, с помощью кото­рых разрабатывается проект.
* Метод - это процедура или техника генерации описаний компонентов ЭИС (например, проектирование потоков и структур данных).
* Модель – совокупность символов (вербальных, математических, графических и т.п.), которая адекватно описывает некоторые свойства моделируемого объекта и отношения между ними.
* Нотация - отображение структуры системы, элементов данных, этапов обработки с помощью специальных графических символов диаграмм, а также описание проекта системы на формальных и естественных языках.
* Инструментальные средства CASE - специальные программы, которые поддерживают одну или несколько методологий анализа и проектирования ИС.

Общая архитектура системы CASE-средств:



Ядром системы является база данных проекта - репозиторий (словарь данных). Он представляет собой специализированную базу данных, предназначенную для отображения состояния про­ектируемой ЭИС в каждый момент времени. Объекты всех диаг­рамм синхронизированы на основе общей информации словаря данных.

Репозиторий содержит информацию об объектах проектиру­емой ЭИС и взаимосвязях между ними, все подсистемы обмени­ваются данными с ним. В репозиторий хранятся описания следующих объектов:

* проектировщиков и их прав доступа к различным компонен­там системы;
* организационных структур;
* диаграмм;
* компонентов диаграмм;
* связей между диаграммами;
* структур данных;
* программных модулей;
* процедур;
* библиотеки модулей и т.д.

Графические средства моделирования предметной области позволяют разработчикам автоматизированных ИС в наглядном виде изучать существующую информационную систему, перестра­ивать ее в соответствии с поставленными целями и имеющимися ограничениями. Все модификации диаграмм, выполняемых раз­работчиками в интерактивном (диалоговом) режиме, вводятся в словарь данных, контролируются с общесистемной точки зрения и могут использоваться для дальнейшей генерации действующих функциональных приложений. В любой момент времени диаграм­мы могут быть распечатаны для включения в техническую доку­ментацию проекта.

Графический редактор диаграмм предназначен для отобра­жения в графическом виде в заданной нотации проектируемой ЭИС. Он позволяет выполнять следующие операции:

* создавать элементы диаграмм и взаимосвязи между ними;
* задавать описания элементов диаграмм;
* задавать описания связей между элементами диаграмм;
* редактировать элементы диаграмм, их взаимосвязи и описа­ния.

Верификатор диаграмм служит для контроля правильности построения диаграмм в заданной методологии проектирования ЭИС. Он выполняет следующие функции:

* мониторинг правильности построения диаграмм;
* диагностику и выдачу сообщений об ошибках;
* выделение на диаграмме ошибочных элементов. Документатор проекта позволяет получать информацию о
* состоянии проекта в виде различных отчетов. Отчеты могут стро­иться по нескольким признакам, например по времени, автору, элементам диаграмм, диаграмме или проекту в целом.

Администратор проекта представляет собой инструменты, необходимые для выполнения следующих административных функций:

* инициализации проекта;
* задания начальных параметров проекта;
* назначения и изменения прав доступа к элементам проекта;
* мониторинга выполнения проекта.

Сервис представляет собой набор системных утилит по обслуживанию репозитория. Данные утилиты выполняют функции архивации данных, восстановления данных и создания нового репозитория.

# Классификация

Современные CASE-системы классифицируются по следую­щим признакам:

* по поддерживаемым методологиям проектирования: функционально (структурно)-ориентированные, объектно-ориентированные и комплексно-ориентированные (набор методологий проектирования);
* по поддерживаемым графическим нотациям построения диаграмм: с фиксированной нотацией, с отдельными нотациями и наиболее распространенными нотациями;
* по степени интегрированности: tools (отдельные локальные средства), toolkit (набор неинтегрированных средств, охватывающих большинство этапов разработки ЭИС) и workbench (полностью интегрированные средства, связанные общей базой проектных данных - репозиторием);
* по типу и архитектуре вычислительной техники: ориентированные на ПЭВМ, ориентированные на локальную вычислительную сеть (ЛВС), ориентированные на глобальную вычислительную сеть (ГВС) и смешанного типа;
* по режиму коллективной разработки проекта: не поддерживающие коллективную разработку, ориентированные на режим реального времени разработки проекта, ориентированные на режим объединения подпроектов;
* по области действия в пределах ЖЦ ИС: Upper CASE – средства, используемые на стадии анализа предметной области; Middle CASE – средства, используемые на стадии анализа и проектирования структуры ИС; Lower CASE – средства, используемые на стадиях разработки и внедрения (тестирования); I-CASE – интегрированная система CASE-средств, объединяет возможности Upper- и Lower- CASE.
* По функциональному назначению делятся на средства:
	+ анализа и проектирования ИС (автоматизация наиболее популярных методологий проектирования);
	+ проектирования баз данных (моделирование данных и генерация схем БД);
	+ разработки приложений (в том числе, средства генерации и рефакторинга программного кода, средства быстрой разработки приложений);
	+ обратного инжиниринга (построение моделей действующей ИС для ее переноса в другую среду);
	+ документирования проекта;
	+ управления тестированием ПО;
	+ планирования и управления проектом.

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

Помимо поддержки начальных этапов разработки важное значение приобретают CASE-системы, ориентированные на проектирование и генерацию баз данных и пользовательских интерфейсов.

Генерация интерфейсов с базами данных и возможность преобразования (конвертирования) между различными концептуальными схемами и моделями данных увеличивает мобильность прикладных систем при переходе в другие операционные среды. Генерация кода и (или) таблиц, описывающих интерфейс прикладной системы с базой данных, не только позволяет сократить время разработки, но и дает возможность отделить разработку приложений от ведения архива проектной документации.

Наиболее трудоемкими этапами разработки ЭИС являются этапы анализа и проектирования, поэтому CASE-системы, как правило, предназначены для автоматизации отслеживания качества принимаемых проектных решений и подготовки 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.

Стратегия выбора CASE-систем для конкретного применения зависит как от целей и потребностей самого проекта, так и от квалификации вовлеченных в процесс проектирования специалистов. В большинстве случаев одно средство не может обеспе­чить все потребности проекта. Разработчики, как правило, применяют набор средств. Например, одно средство наилучшим образом подходит для анализа, а другое - для проектирования систем. В общем случае при выборе CASE-системы необходимо учитывать следующие аспекты.

* *Наличие базы проектных данных, архива или словаря.* СУБД и словари данных обеспечивают высокую степень интеграции данных и предоставляют широкие возможности для центра­лизованного сбора, хранения и распределения проектной ин­формации между различными этапами проекта и выполняемыми операциями.
* *Интерфейсы с другими CASE-системами.* В процессе проек­тирования ЭИС могут использоваться различные методо­логии, поэтому важно, чтобы используемые CASE-системы предоставляли возможности для эффективного использо­вания нескольких методов. При этом должна быть обеспе­чена терминологическая совместимость различных методо­логий.
* *Возможности экспорта/импорта.* Спецификации, полученные на этапах анализа, проектирования и кодирования для одной ЭИС, могут быть использованы для проектирования другой системы. Повторное проектирование и кодирование могут быть обеспечены при помощи средств экспорта/импорта спе­цификаций в различные CASE-системы. Многопользовательский режим. Развитые CASE-системы дол­жны обладать возможностями разделения полномочий пер­сонала разработчиков и объединения отдельных работ в об­щий проект.
* *Открытая архитектура.* Открытая к доступу проектировщи­ков информация об используемых форматах файлов и интер­фейсах должна позволять безболезненно переходить от одной CASE-системы к другой.
* *Расширение новыми методологиями.* Как и любое програм­мное средство, CASE-система должна обладать возможнос­тью совершенствоваться с учетом появления новых требова­ний или новых предметных областей.
* *Наличие графических средств поддержки методологий проек­тирования.* Большинство CASE-систем базируется на графи­ческом отображении методологий. Графические элементы структурных диаграмм и объекты словаря должны позволять декомпозировать различные компоненты проекта и детали­зировать изображения с той степенью, с какой это необходи­мо для понимания проектных решений.
* *Обеспечение качества проектной документации.* Это требова­ние относится к возможностям CASE-системы анализировать и проверять описания и документацию на полноту и непро­тиворечивость, а также на соответствие принятым в данной методологии стандартам и правилам, В результате анализа должна формироваться информация, указывающая на имею­щиеся противоречия или неполноту проектной документации, находящейся в архиве или словаре.
* *Автоматическая генерация отчетов о проектных решениях.* Решения (спецификации), созданные в процессе проектиро­вания, служат источником документирования системы. Час­то возникает потребность получения твердой копии специфи­каций в текстовой или графической форме.
* *Генерация кодов программ.* CASE-системы с жесткой ориента­цией на конкретные СУБД должны обеспечивать возможность генерации программ в среде этих СУБД.
* *Планирование и управление проектом.* Использование CASE-систем не исключает потребности в эффективном управлении проектом. Многие развитые CASE-системы имеют в своем составе средства планирования и управления проектом. Спе­цификации, которые используются этими средствами, пред­ставляют собой опорные точки управления, позволяющие определять сроки разработки.