Лекция 5. Автоматизированное проектирование ИС

Учебные вопросы:

1. **Понятие CASE-технологии.**
2. **Принципы CASE-технологий.**
3. **Факторы эффективности CASE-технологии.**
4. **Аспекты выбора CASE-технологии.**
5. **Классификация CASE-средств.**
6. **Технология внедрения CASE-средств.**

Понятие CASE-технологии

CASE ***(Computer Aided Software/System Engineering)*** – проектирование программного обеспечения или системы на основе компьютерной поддержки.

CASE-технология – это совокупность методов анализа, проектирования, разработки и сопровождения ИС на основе компьютерной поддержки.

Основная цель CASE-технологии состоит в том, чтобы отделить процесс проектирования ИС от ее кодирования и последующих этапов разработки, а также максимально автоматизировать процесс разработки и функционирования систем.

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 ИС;
- 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 ИС в режиме реального времени.

Инструментальные CASE-средства

Инструментальные средства CASE – это специальные программы, которые поддерживают одну или несколько методологий анализа и проектирования ИС.

Рисунок 1 – Взаимосвязь основных структурных компонентов CASE-средства

Компоненты CASE-средства

Репозиторий – специальная база данных, содержащая информацию о проекте ИС. Репозиторий содержит информацию, характеризующую диаграммы, связи между диаграммами, структуры данных, программные модули, права доступа проектировщиков ИС и т. д.
Репозиторий обеспечивает хранение версий проекта, групповую работу над проектом, контроль полноты и непротиворечивости данных. В репозиторий предусматриваются архивация и резервное копирование проектных данных.

Графический редактор диаграмм предназначен для отображения в заданных нотациях всех диаграмм проектирования ИС. Редактор диаграмм может создавать элементы диаграмм и связи между ними.

Средства контроля и сбора статистики выполняют следующие функции:

- проверка правильности построения диаграмм и выдача сообщений об ошибках;
- выделение на диаграмме ошибочных элементов;
- сбор статистики ошибок в процессе проектирования.

Генератор документов формирует выходные документы, содержащие диаграммы проекта в соответствии с запросом проектировщика.

Администратор проекта занимается административными функциями проектирования, в числе которых:

- назначение и изменение прав доступа к репозиторию;
- мониторинг процесса проектирования.

Браузер позволяет осуществлять просмотр проекта, в том числе переключение от одной диаграммы к другой и т.д.

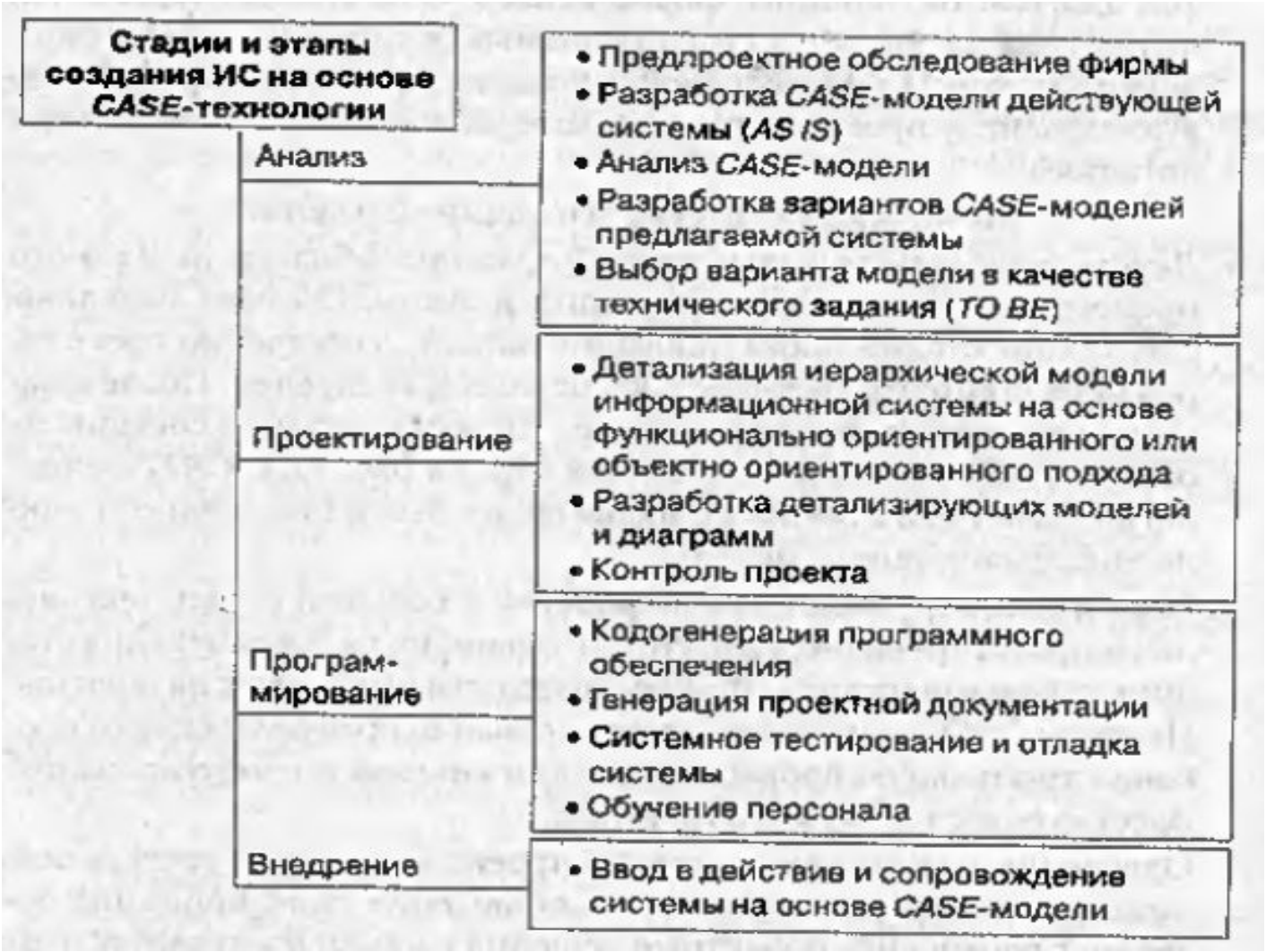
Генератор кодов программ на основе моделей проекта, хранящихся в репозиторий, создает код программы

Принципы CASE-технологий

Существует несколько принципов CASE-технологий:

1. Принцип всесторонней компьютерной поддержки проектирования.
2. Принцип модельного подхода.
3. Иерархическое представление модели предметной области.
4. Наглядность представления модели.
5. Декомпозиция процесса проектирования на стадии и этапы.
6. Перенесение трудоемкости разработки в большей степени на анализ и проектирование.
7. Отделение, независимость стадий проектирования от средств реализации, от программирования.
8. Возможность как прямого, так и обратного проектирования.
9. Использование репозитория.

Последовательность стадий и этапов создания ИС на основе CASE-технологии



Положения в построении CASE-средств

Помимо перечисленных принципов в основе построения CASE-средств лежат следующие положения:

1. **Человеческий** фактор, определяющий разработку ПО как легкий, удобный и экономичный **процесс**.
2. Широкое использование базовых программных средств, получивших массовое распространение в других приложениях (БД и СУБД, компиляторы с различных языков программирования, отладчики, документаторы, издательские системы, оболочки экспертных систем и базы знаний и другое).
3. Автоматизированная или автоматическая **кодогенерация**, выполняющая несколько видов генерации кодов: **преобразования** для получения документации, **формирования** БД, ввода/**модификации** данных, автоматической сборки модулей из словарей и моделей данных и повторно используемых программ.
4. Ограничение **сложности**, позволяющее получать **компоненты**, поддающиеся **управлению**, обозримые и **доступные** для понимания, а также обладающие простой и ясной **структурой**.
5. Доступность для разных категорий пользователей.
6. Рентабельность.
7. **Сопровождаемость**, обеспечивающая способность **адаптации** при **изменении** требований и целей **проекта**.

Факторы эффективности CASE-технологии

Факторы эффективности CASE-технологии

1. CASE-технология создает возможность для реинжиниринга бизнеса и предусматривает перенос центра тяжести трудоемкости создания системы на предпроектную и проектную стадии.
2. Доступная для понимания пользователей-непрограммистов графическая форма представления модели позволяет следовать принципу пользовательского проектирования, предусматривающему участие пользователей в создании системы.
3. Наличие формализованной модели системы создает возможность для многовариантного анализа с прототипированием и ориентировочной оценкой эффективности вариантов.
4. CASE-технология позволяет использовать концепцию сборочного проектирования, основанную на повторном использовании типовых проектных решений (компонентов) системы.
5. Закрепление в формализованном виде требований к системе избавляет проектировщиков от необходимости многочисленных корректировок в соответствии с новыми требованиями пользователей.
6. Отделение проектирования системы от программирования создает устойчивость проектных решений для реализации на разных программно-технических платформах.
7. Наличие формализованной модели реализации системы и соответствующих средств автоматизации позволяет осуществить автоматическую кодогенерацию программного обеспечения системы и создать рациональную структуру базы данных.
8. На стадии эксплуатации системы появляется возможность внесения изменений на уровне модели, не обращаясь к текстам программ, силами специалистов отдела автоматизации фирмы, т. е. осуществить модификацию проекта.
9. Модель системы может использоваться не только как основа, но и в целях автоматизированного обучения персонала с использованием диаграмм.
10. На основе модели действующей системы может выполняться бизнес-анализ для поддержки управленческих решений и бизнес-реинжиниринг при изменении направления деятельности

Аспекты выбора CASE-технологии

При выборе CASE-системы необходимо учитывать следующие **аспекты**:

1. Наличие базы проектных данных, архива или словаря.
2. Интерфейсы с другими CASE-системами.
3. Возможности экспорта/импорта.
4. Многопользовательский режим.
5. Открытая архитектура.
6. Расширение новыми методологиями.
7. Наличие графических средств поддержки методологий проектирования.
8. Обеспечение качества проектной документации.
9. Автоматическая генерация отчетов о проектных решениях.
10. Генерация кодов программ.
11. Планирование и управление проектом.

Классификация CASE-средств

По аналогии с классификацией ИС, для создания которых предназначены CASE-средства выделяют следующие:

- локальные (*Design/IDEF, CASE, Аналитик);*
- малые интегрированные (*AllFusion Modeling Suite, Silverrun);*
- средние интегрированные CASE-средства (*Rational Rose, Designer/2000);*
- крупные интегрированные *CASE-средства (ARIS)*

Помимо приведенной выше классификации возможны и другие классификации, например 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 ИС) 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 типу ОС:** работающие под управлением WINDOWS, UNIX, под управлением различных ОС.