**Лекция 4: Проектирование информационных систем**

**1. Введение в проектирование информационных систем**

Проектирование информационных систем — это процесс, в рамках которого создаётся архитектура и структура будущей системы. Этот этап следует за сбором и анализом требований и обеспечивает детальную проработку того, как система будет реализована на техническом уровне. Основной целью проектирования является создание модели системы, которая удовлетворяет все требования, определённые на предыдущем этапе, и при этом обладает хорошей производительностью, масштабируемостью, безопасностью и удобством использования.

Проектирование информационных систем можно разделить на два основных уровня:

* **Концептуальное проектирование**: фокусируется на общей структуре системы и её компонентах, без детальной проработки технических аспектов.
* **Детальное проектирование**: предполагает проработку технических деталей, таких как базы данных, интерфейсы, алгоритмы и программные модули.

Процесс проектирования важен для того, чтобы избежать ошибок на этапе разработки и обеспечения согласованности между всеми частями системы.

**2. Архитектурное проектирование информационных систем**

Архитектура информационной системы — это структура, которая описывает компоненты системы, их взаимосвязи, взаимодействие и функционирование. Проектирование архитектуры играет ключевую роль в успешной реализации и эксплуатации информационных систем, так как она определяет, как различные модули и компоненты будут интегрироваться и взаимодействовать друг с другом.

**Основные подходы к архитектурному проектированию:**

* **Многоуровневая архитектура (Layered Architecture)**: система делится на несколько уровней, каждый из которых выполняет свою функцию. Например:
  + **Представление (Presentation Layer)**: отвечает за взаимодействие с пользователем (интерфейсы).
  + **Логика (Business Logic Layer)**: реализует основные бизнес-процессы.
  + **Данные (Data Layer)**: отвечает за хранение и управление данными.

Многоуровневая архитектура обеспечивает модульность, улучшает масштабируемость и облегчает поддержку системы.

* **Клиент-серверная архитектура**: в этой архитектуре система делится на две основные части — клиентскую и серверную. Клиент отправляет запросы серверу, который их обрабатывает и возвращает результаты. Такая архитектура широко используется для создания веб-приложений и корпоративных информационных систем.

Преимущества клиент-серверной архитектуры:

* + Централизованное управление данными на сервере.
  + Упрощённая поддержка и обновление серверной части без необходимости обновления клиентов.
* **Микросервисная архитектура**: система делится на набор независимых микросервисов, каждый из которых выполняет свою задачу и может работать независимо от других. Микросервисы взаимодействуют через API, что делает систему более гибкой и масштабируемой.

Преимущества микросервисной архитектуры:

* + Возможность независимого масштабирования отдельных компонентов системы.
  + Лёгкость в поддержке и развитии системы, так как каждый микросервис может разрабатываться и поддерживаться отдельной командой.

**3. Проектирование баз данных**

Проектирование базы данных — важная часть проектирования информационной системы. От того, как будет структурирована база данных, зависит производительность, масштабируемость и поддержка системы.

Основные этапы проектирования базы данных:

* **Определение данных**: на основе требований определяются ключевые объекты данных, такие как пользователи, заказы, продукты и т.д.
* **Нормализация данных**: процесс разбиения данных на таблицы для устранения избыточности и уменьшения потенциальных аномалий. Основные формы нормализации:
  + Первая нормальная форма (1NF): каждая ячейка таблицы должна содержать одно значение.
  + Вторая нормальная форма (2NF): таблица должна удовлетворять 1NF и все неключевые атрибуты должны зависеть от всего ключа.
  + Третья нормальная форма (3NF): таблица должна удовлетворять 2NF и все неключевые атрибуты должны зависеть только от ключа, а не от других неключевых атрибутов.
* **Проектирование ключей**: ключи (первичные и внешние) обеспечивают целостность данных. Первичный ключ — это уникальный идентификатор для каждой записи в таблице, а внешний ключ связывает таблицы друг с другом.
* **Оптимизация запросов**: правильная организация индексов и оптимизация структуры таблиц может существенно повысить производительность запросов к базе данных.

**4. Проектирование пользовательского интерфейса**

Пользовательский интерфейс (UI) — это та часть системы, с которой взаимодействуют пользователи. Проектирование интерфейсов важно для обеспечения удобства использования системы и повышения её эффективности.

**Основные принципы проектирования интерфейсов:**

* **Простота**: интерфейс должен быть интуитивно понятным и простым в использовании. Чем меньше действий нужно выполнить пользователю для достижения цели, тем лучше.
* **Консистентность**: все элементы интерфейса должны быть согласованными по стилю, цветам и расположению, чтобы пользователю было легче ориентироваться.
* **Обратная связь**: система должна предоставлять пользователю обратную связь о выполненных действиях, чтобы он мог понимать, что происходит.
* **Адаптивность**: интерфейс должен корректно отображаться на различных устройствах и экранах.
* **Доступность**: интерфейс должен быть доступен для людей с ограниченными возможностями, включая поддержку экранных читалок и возможность управления с клавиатуры.

Важные инструменты для проектирования интерфейсов:

* **Прототипирование**: создание низкокачественных макетов интерфейса для тестирования его логики и структуры на ранних стадиях проектирования.
* **Юзабилити-тестирование**: проверка интерфейса на реальных пользователях для выявления проблем и их устранения.

**5. Моделирование системы**

Моделирование системы — это важный шаг в проектировании информационных систем. Оно позволяет визуализировать структуру и поведение системы до её реальной разработки.

**Популярные типы моделей:**

* **Диаграммы классов (UML)**: отображают структуру системы, её классы и взаимосвязи между ними. Полезны для проектирования объектно-ориентированных систем.
* **Диаграммы прецедентов**: показывают, как пользователи (актеры) взаимодействуют с системой через различные сценарии использования (прецеденты). Это помогает визуализировать требования и определить, какие функции должна выполнять система.
* **Диаграммы последовательностей**: иллюстрируют, как объекты системы взаимодействуют друг с другом в рамках выполнения определённого сценария.
* **Диаграммы компонентов**: показывают, как компоненты системы взаимодействуют между собой, и помогают понять её модульность.

**6. Документирование проектных решений**

Документация — важный аспект проектирования, так как она служит основой для разработки и последующей поддержки системы. Документы должны содержать полное описание архитектуры, ключевых решений и обоснование выбора тех или иных подходов.

**Основные типы документации:**

* **Архитектурная спецификация**: описание структуры системы, её компонентов и взаимосвязей.
* **Технические спецификации**: детализированные описания алгоритмов, структуры баз данных, API и других технических аспектов системы.
* **Протоколы решений**: документирование всех ключевых решений, принятых в ходе проектирования, с обоснованиями.

**Заключение**

Проектирование информационных систем — это сложный и важный этап, от которого зависит успех всего проекта. На этом этапе определяется архитектура системы, её компоненты, проектируются базы данных, интерфейсы и взаимодействие между частями системы. Правильное проектирование позволяет создать систему, которая будет эффективно функционировать, легко поддерживаться и масштабироваться.

**Дополнительные материалы:**

1. **Книги**:
   * Bass L., Clements P., Kazman R. "Software Architecture in Practice."
   * Fowler M. "UML Distilled: A Brief Guide to the Standard Object Modeling Language."
2. **Статьи**:
   * "Designing Scalable and Maintainable Systems" (ACM Software Engineering Notes).
   * "The Role of Architecture in System Performance" (IEEE Transactions on Software Engineering).

**Задания:**

1. Создать архитектурную схему системы на основе описанного проекта, включая диаграмму компонентов и классов.
2. Подготовить прототип пользовательского интерфейса для системы и провести юзабилити-тестирование на группе пользователей.