

Проектирование ПО

Тема 2. Особенности процесса проектирования

Предварительное проектирование

Предварительное проектирование включает:

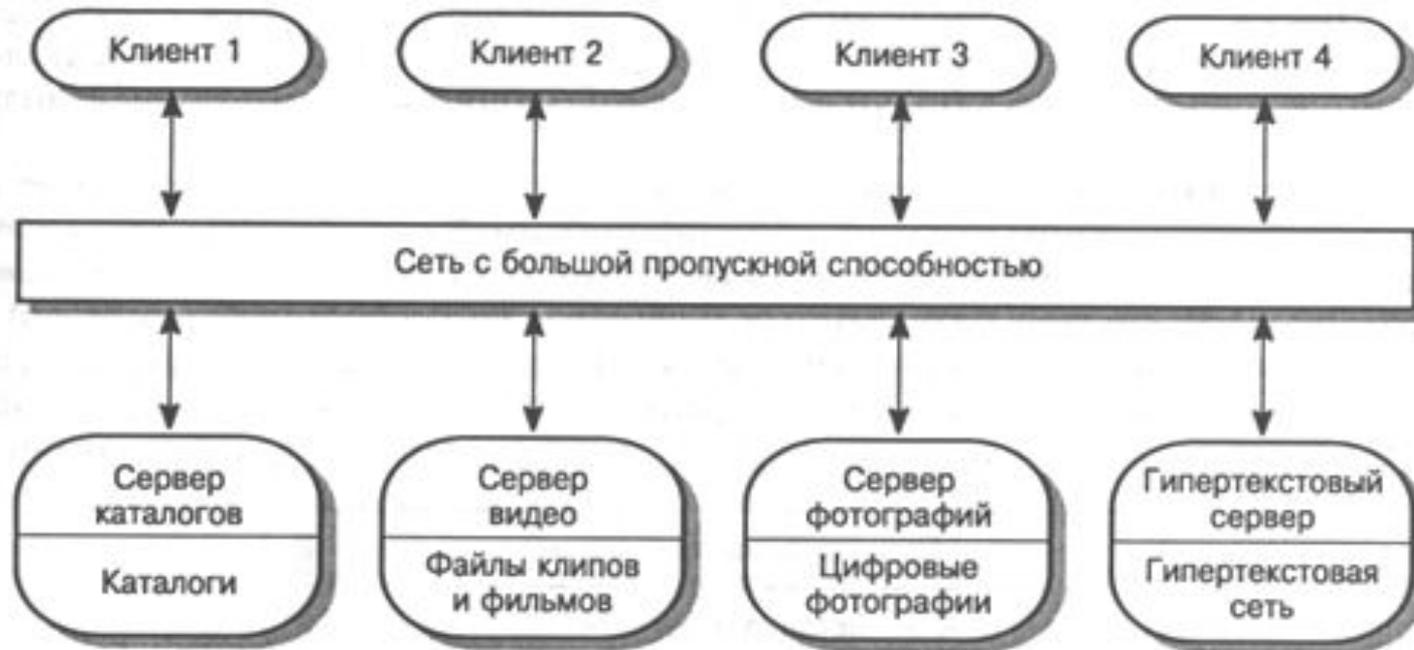
- 1. Структурирование системы.** Система структурируется на несколько подсистем, где под подсистемой понимается независимый программный компонент.
- 2. Моделирование управления.** Определяется модель связей управления между частями системы.
- 3. Декомпозиция подсистем на модули.** Каждая подсистема разбивается на модули. Определяются типы модулей и межмодульные соединения.

1. Структурирование системы

Модели системного структурирования:

- 1) модель клиент-сервер;
- 2) модель хранилища данных;
- 3) трехуровневая модель;
- 4) модель абстрактной машины.

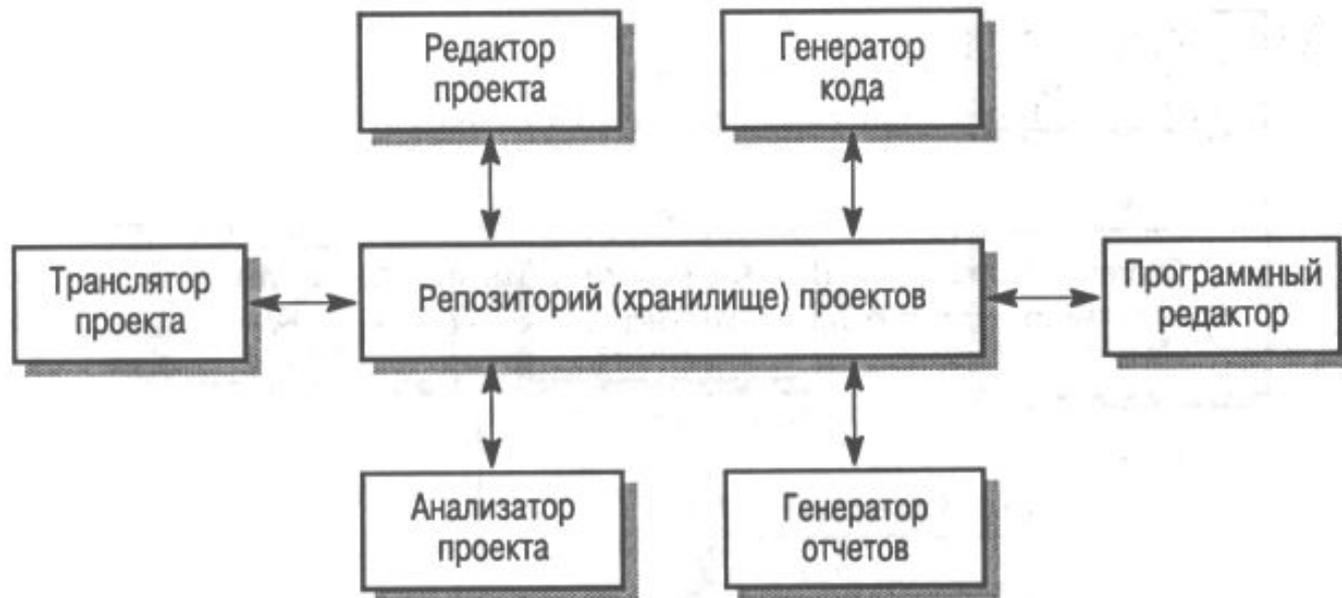
1.1. Модель клиент-сервер



Архитектура библиотечной системы фильмов и фотографий

Используется для распределенных систем, где данные распределены по серверам

1.2. Модель хранилища данных



Архитектура интегрированного набора CASE-средств с общим репозиторием

Подсистемы разделяют данные, находящиеся в общей памяти. Как правило, данные образуют БД.

1.3. Трехуровневая модель



Каждый слой расположен на отдельном компьютере

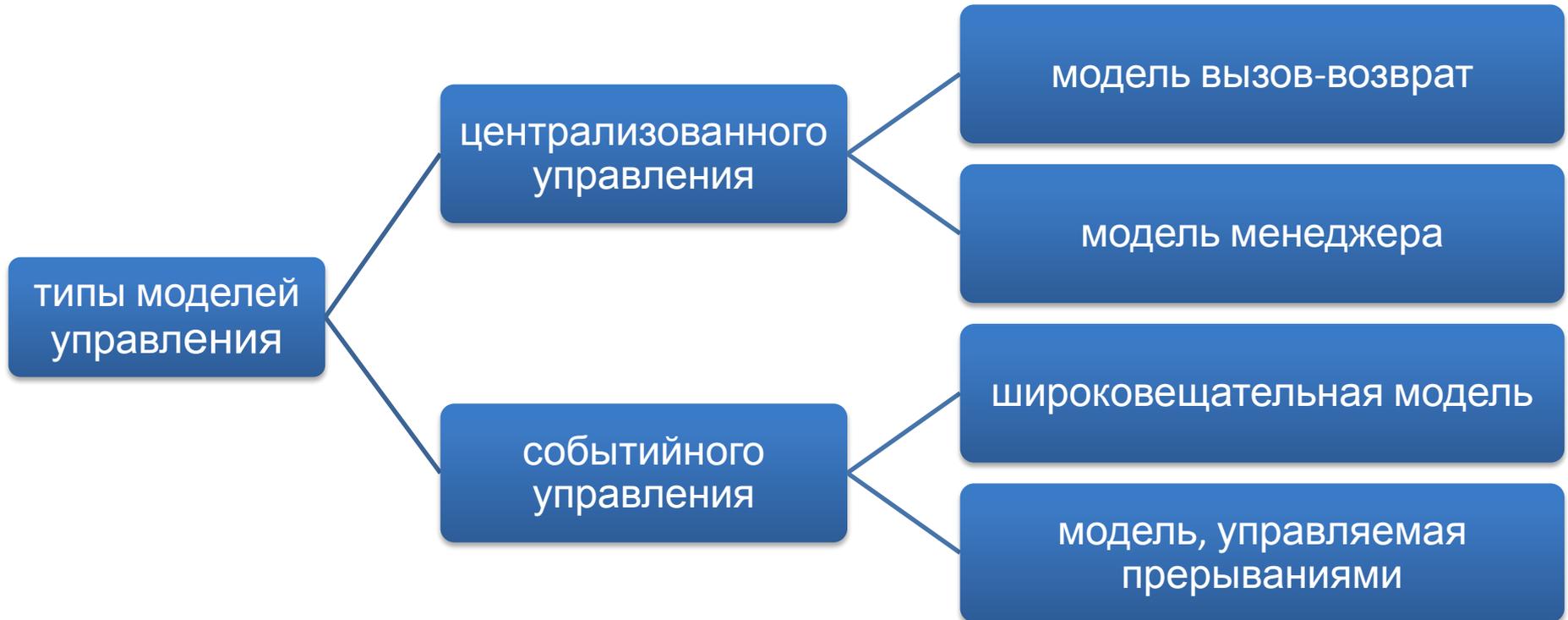
1.4. Модель абстрактной машины



Модель абстрактной машины для системы администрирования версий

Каждый текущий слой реализуется с использованием средств, обеспечиваемых слоем-фундаментом.

2. Моделирование управления



2.1.1. Модель вызов-возврат



2.1.2. Модель менеджера



Модель диспетчера для системы реального времени

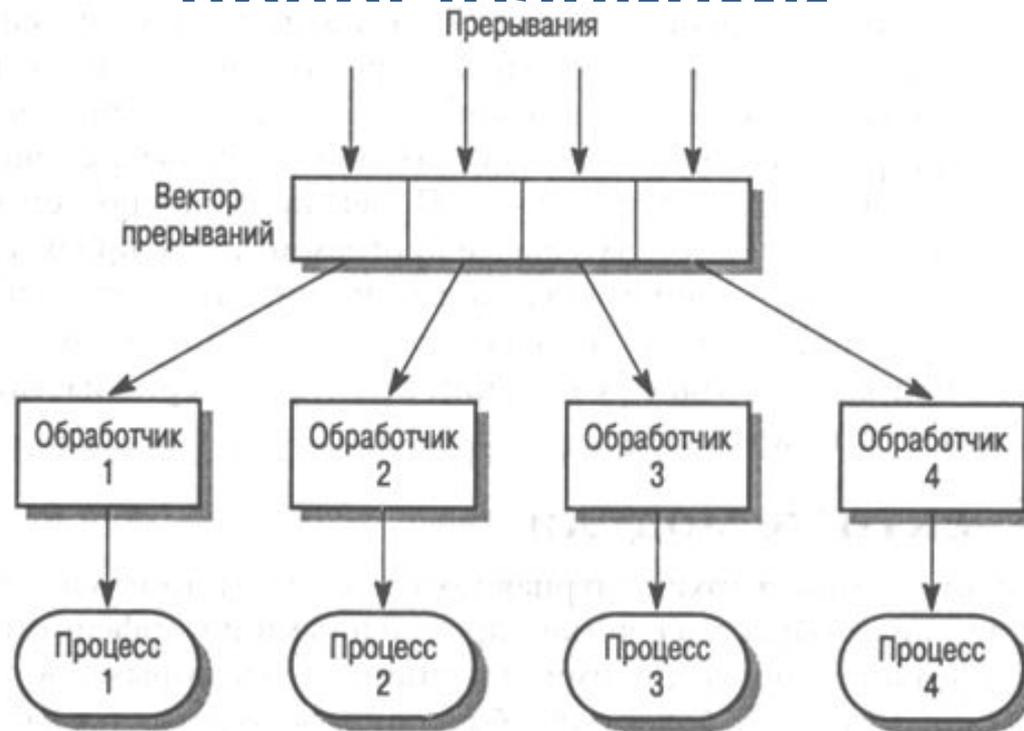
2.2.1. Широковещательная модель



Каждая подсистема уведомляет обработчик о своём интересе к определённым событиям.

Когда событие происходит, обработчик пересылает события всем подсистемам, которые были на него подписаны.

2.2.2. Модель, управляемая прерываниями



Все прерывания разбиты на группы — типы, которые образуют вектор прерываний.

Для каждого типа прерывания есть свой обработчик.

Каждый обработчик реагирует на свой тип прерывания и запускает свой процесс.

Прерывание — это принудительная передача управления от выполняемой программы к системе (а через нее — к соответствующей программе обработки

3. Декомпозиция подсистем на модули

Типы моделей модульной декомпозиции:

- модель потока данных (структурный подход);
- модель объектов (объектно-ориентированный подход).

Модульность

Модульность — свойство системы, которая может подвергаться декомпозиции на ряд внутренне связанных и слабо зависящих друг от друга модулей.

Модульность — свойство ПО, обеспечивающее интеллектуальную возможность создания сколь угодно сложной программы

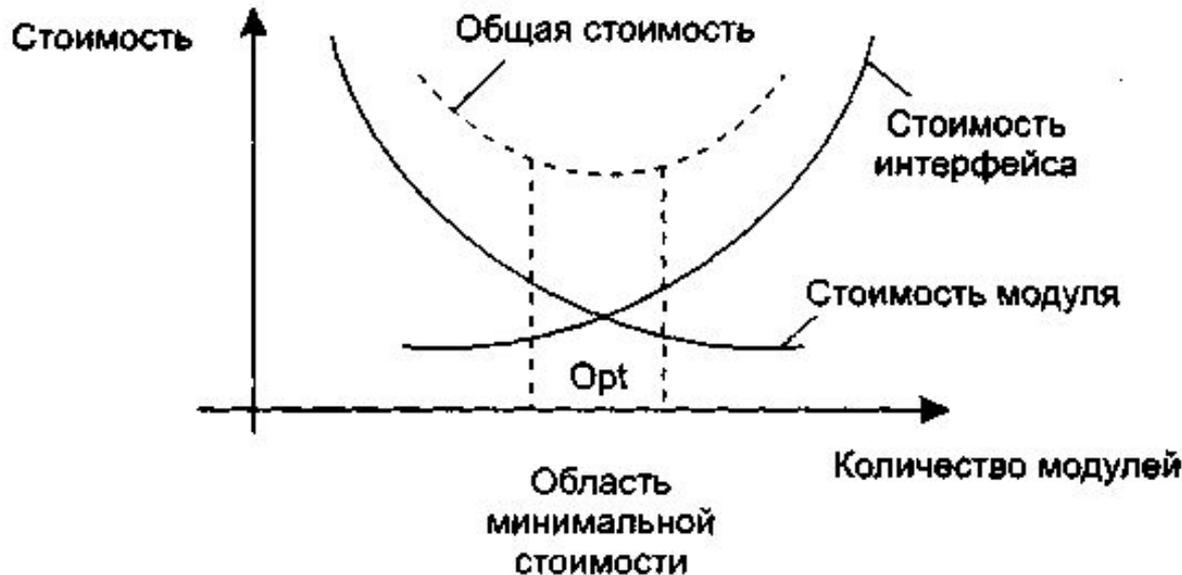


**Гленфорд
Майерс
(Glenford Myers)**

Модуль

Модуль — фрагмент программного текста, являющийся строительным блоком для физической структуры системы.

Как правило, модуль состоит из интерфейсной части и части-реализации

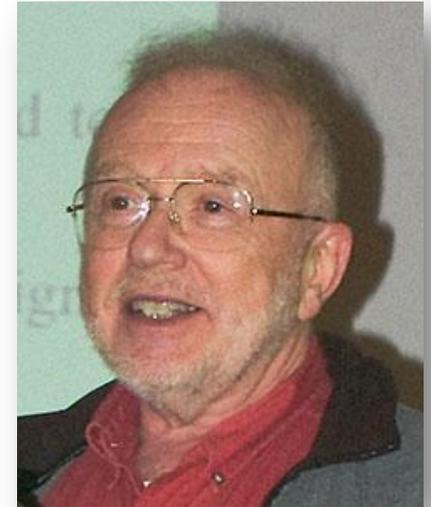


Оптимальный модуль должен удовлетворять двум критериям:

- 1) снаружи он проще, чем внутри;
- 2) его проще использовать, чем построить.

Информационная закрытость модуля

«Содержание модулей должно быть скрыто друг от друга. Модуль должен определяться и проектироваться так, чтобы его содержимое (процедуры и данные) было недоступно тем модулям, которые не нуждаются в такой информации».



**Дэвид
Парнас
(David Parnas)**

Достоинства:

- обеспечивается возможность разработки модулей различными, независимыми коллективами;
- обеспечивается легкая модификация системы (вероятность распространения ошибок очень мала, так как большинство данных и процедур скрыто от других частей системы).

Сложность модуля

Объем программного кода (LOC) - определяет количество строк кода.

Метрики Холстеда. Морис Холстед (Maurice Halstead) предложил (1977) метрики оценки лексической сложности:

1). *Оценочная длина программы* - $N = n_1 \log_2(n_1) + n_2 \log_2(n_2)$, $N = N_1 + N_2$

где n_1, n_2 - число различных операторов и различных операндов;

2). *Объем программы* - $V = N \log_2(n_1 + n_2)$,

3). *Трудность создания программы* - $L = (2 n_2) / (n_1 N_2)$,

где N_1, N_2 - число всех операторов и операндов.

4). *Трудность понимания программы* - $E_c = V / L$.

Цикломатическая сложность. Томас Мак-Кейб (Thomas McCabe) (1976). Это мера сложности логической структуры программы.

$$V(G) = E - N + 2,$$

где E – количество дуг, N – количество вершин.

Для **объектно-ориентированных** систем в качестве метрик, оценивающих сложность и качество ПО, используют: *общее количество классов; суммарное количество методов, определённых во всех классах; число методов на класс; высоту дерева наследования; количество подклассов* и т.

Связность модуля (Cohesion)

Связность — внутренняя характеристика модуля. Это мера зависимости его частей (СС – сила связности). Чем выше связность модуля, тем лучше результат проектирования, то есть тем «черней» его ящик, тем проще им управлять.

1. **Связность по совпадению** (СС=0). В модуле отсутствуют явно выраженные внутренние связи.
2. **Логическая связность** (СС=1). Части модуля объединены по принципу функционального подобия. Например, модуль состоит из разных подпрограмм обработки ошибок.
3. **Временная связность** (СС=3). Части модуля не связаны, но необходимы в один и тот же период работы системы.
4. **Процедурная связность** (СС=5). Части модуля связаны порядком выполняемых ими действий, реализующих некоторый сценарий поведения.
5. **Коммуникативная связность** (СС=7). Части модуля связаны по данным (работают с одной и той же структурой данных).
6. **Информационная (последовательная) связность** (СС=9). Выходные данные одной части используются как входные данные в другой части модуля.
7. **Функциональная связность** (СС=10). Части модуля вместе реализуют одну функцию.
8. **Объектная связность**. Максимальная связность.

Сцепление модулей (Coupling)

Сцепление — внешняя характеристика модуля, которую желательно уменьшать. Это мера взаимозависимости модулей по данным (СЦ - степень сцепления).

Выделяют 6 типов сцепления.

Сцепление по данным (СЦ=1). Модуль А вызывает модуль В. Все входные и выходные параметры вызываемого модуля — простые элементы данных.

Сцепление по образцу (СЦ=3). В качестве параметров используются структуры данных.

Сцепление по управлению (СЦ=4). Модуль А явно управляет функционированием модуля В, посылая ему управляющие данные.

4. **Сцепление по внешним ссылкам (СЦ=5).** Модули А и В ссылаются на один и тот же глобальный элемент данных.

5. **Сцепление по общей области (СЦ=7).** Модули разделяют одну и ту же глобальную структуру данных.

6. **Сцепление по содержанию (СЦ=9).** Один модуль прямо ссылается на содержание другого модуля (не через его точку входа).

Принципиальное преимущество объектно-ориентированного проектирования в том, что природа объектов приводит к созданию слабо сцепленных систем.