

# Проектирование Программного Обеспечения

Преподаватель:  
Бекасов Денис Евгеньевич  
[bekasov@bmstu.ru](mailto:bekasov@bmstu.ru)

# Цель и задачи

## Цель

Ответить для себя на вопрос «Что такое хорошая Архитектура?»

## Задачи

- Критерии хорошей Архитектуры
- Проектирование программных компонентов (SOLID) и их взаимодействия
- Проектирование программных систем
- Жизненный цикл разработки ПО
- Организация процессов разработки, тестирования, развертывания и сопровождения

# Состав курса

- 8 лекций
  - Бекасов Д.Е.
- 8 семинаров
  - Бекасов Д.Е.
  - Бабарыкин Д.С.
  - Жаров С.В.
- 4 лабораторные работы (домашних заданий)
- «Время на обсуждение и сдачу лабораторных работ» - бесценно каждую неделю.

# Семинары

- Повторение материалов
- Обсуждение теоретических моментов с лекций
- Командное решение кейсов
- Обсуждение вопросов лабораторных работ

# Лабораторные работы

«Платформа» для задач – **Курсовой Проект по БД.**

Достоинства:

- Не нужно делать лишнего
- Повышение вероятности выполнения лабораторных работ
- Повышение вероятности выполнения курсового проекта
- Повышение качества выполнения курсового проекта
- Хороший пример псевдо-реальной задачи среднего размера

# Лабораторные работы

«Платформа» для задач – **Курсовой Проект по БД.**

Недостатки:

- Не нужно делать лишнего == «не делаем ничего»
- Повышение вероятности выполнения лабораторных работ == «откладываем лабы на потом»
- Повышение вероятности выполнения курсового проекта == «откладываем курсач на потом»
- Повышение качества выполнения курсового проекта == «нарисуем какие-нибудь схемки»
- Хороший пример псевдо-реальной задачи среднего размера == «очень псевдо»

# Лабораторные работы

Для **старта** необходимо:

- В течение текущей и следующей недель согласовать темы курсовых работ с руководителями
- Внести тему в таблицу

# Архитектура



1949 лет

Сфера применения:

- Амфитеатр
- Арена
- Кладбище
- Церковь
- Крепость
- Каменоломня
- Селитровый завод
- Священное место
- Туристический аттракцион

Модифицировался

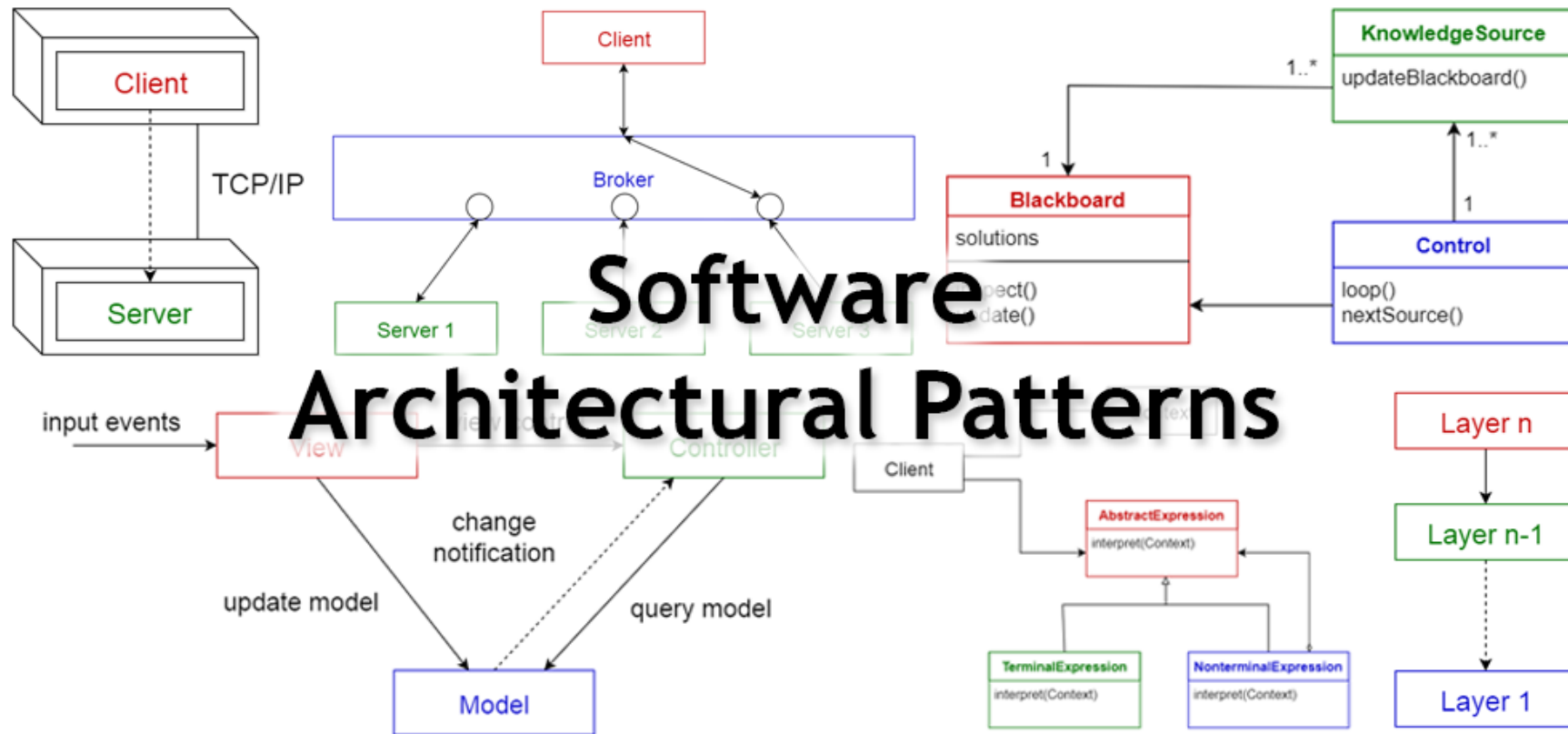
Поддерживается



# Неизбежность



# Архитектура программного обеспечения



# Архитектура программного обеспечения

**Архитектура программного обеспечения** — совокупность важнейших решений об организации программной системы.

- выбор структурных элементов и их интерфейсов, с помощью которых составлена система, а также их поведения в рамках сотрудничества структурных элементов
- соединение выбранных элементов структуры и поведения во всё более крупные системы
- общий архитектурный стиль

# Гради Буч (создатель UML)

Архитектура отражает важные проектные решения по формированию системы, где важность определяется стоимостью изменений

Джозеф Йодер  
(основатель The Refactory, Inc.,)

Если вы думаете, что хорошая архитектура стоит дорого,  
попробуйте плохую архитектуру

# Ральф Джонсон

## (Автор книг по архитектуре ПО)

Архитектура – это набор верных решений, которые хотелось бы принять на ранних этапах работы над проектом, но которые не более вероятны, чем другие

# Том Гилб

## (Инженер-консультант)

Архитектура – это гипотеза, которую требуется доказать реализацией и оценкой

Роберт Мартин  
(Автор книги «Чистая Архитектура»)

Поспешай не торопясь



# Поспешность

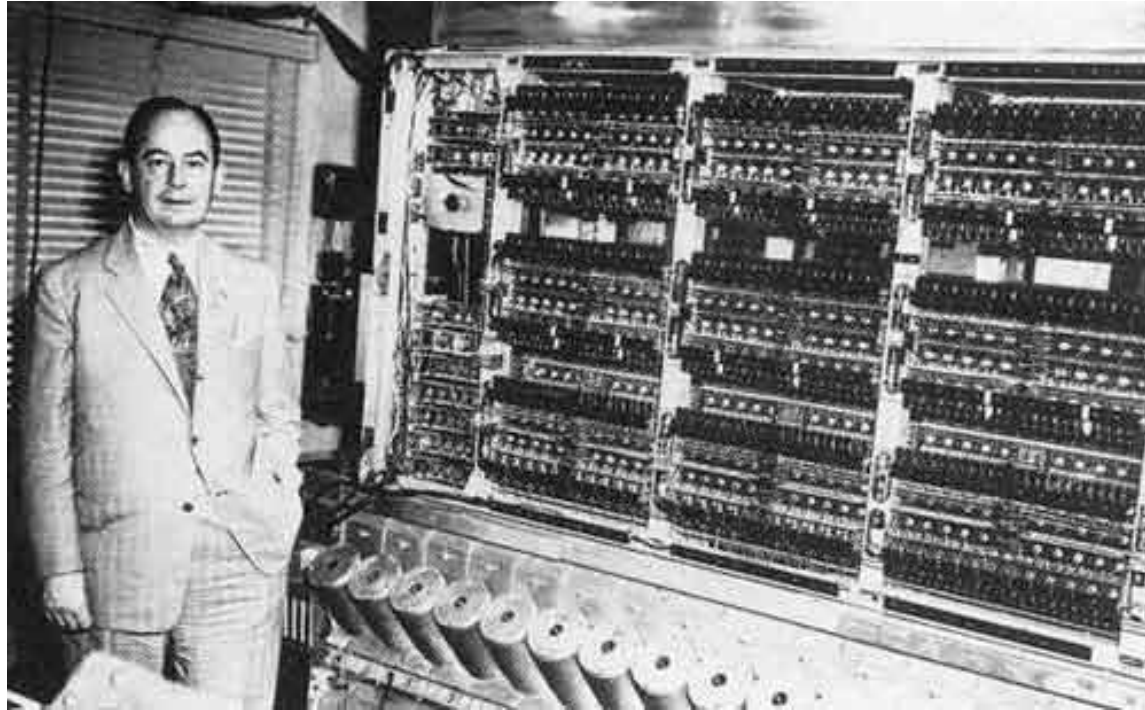
Самонадеянность, управляющая перепроектированием, приведет к тому же беспорядку что и прежде

# Программные архитектуры

- Неизменны на всем протяжении развития IT, начиная с 50-60-х гг XX века
- Низкоуровневые детали и высокоуровневая структура являются частями одного целого.
- Цель – уменьшение трудозатрат на создание и сопровождение системы

# К истокам

Идея ПО – простая возможность менять поведение компьютера



# Что ценно?

Правильная работа системы или простота ее изменения?

# Простота изменений

- Изменение требований
- Исправление ошибок

# Битва за архитектуру

Что такое задачи от любого типа заказчиков?

Это поток фрагментов, которые необходимо встраивать в мозаику со все возрастающей сложностью

- Разработчики – архитектура
- Менеджмент – функциональность

# Парадигмы программирования

- Структурное программирование
- Объектно-ориентированное программирование
- Функциональное программирование

# Структурное программирование

1968, принципы Дейкстры

Структурное программирование накладывает ограничение на прямую передачу управления



# Структурное программирование

**Принцип 1.** Следует отказаться от использования оператора безусловного перехода goto

**Принцип 2.** Любая программа строится из трёх базовых управляющих конструкций: последовательность, ветвление, цикл

**Принцип 3.** В программе базовые управляющие конструкции могут быть вложены друг в друга произвольным образом

**Принцип 4.** Декомпозиция в виде подпрограмм (процедур и функций)

**Принцип 5.** Каждую логически законченную группу инструкций следует оформить как блок

**Принцип 6.** Все перечисленные конструкции должны иметь один вход и один выход

**Принцип 7.** Разработка программы ведётся пошагово, методом «сверху вниз»

# Программирование – это «наука»

«Математический метод»:

Формальное доказательство истинности

«Естественно-научный метод»:

Правило опровергающих доказательств

Хотя, мы прекрасно понимаем, что  
Программирование — это искусство ;)

# Объектно-ориентированное программирование

Объектно-ориентированное программирование накладывает ограничение на косвенную передачу управления

- Инкапсуляция
- Наследование
- Полиморфизм

# Функциональное программирование

Функциональное программирование накладывает ограничение на присваивание