

# Особенности разработки сложных программных систем



**TRAINING**  
CENTER

— <epam> —



## КОНСТАНТИН КОСТРИЦКИЙ

### *.Net Developer*

- Прошел .Net курс тренинг центра ЕПАМ
- Работаю в компании Епам 4.5 года.
- Был разработчиком на проектах Касперского, Билайн, РосГосСтрах

# Главные отличия небольших проектов

---

- Отсутствие четкой структуры процесса разработки
- Отсутствие документации
- Отсутствие строгих сроков этапов разработки
- Отсутствие peer review
- Нечеткие роли людей в команде
- Отсутствие требований к CodeStyle

# Инструменты разработки

- Системы управления версиями (Git, SVN, Mercurial)



- Системы отслеживания ошибок (Jira, TFS, Trello)

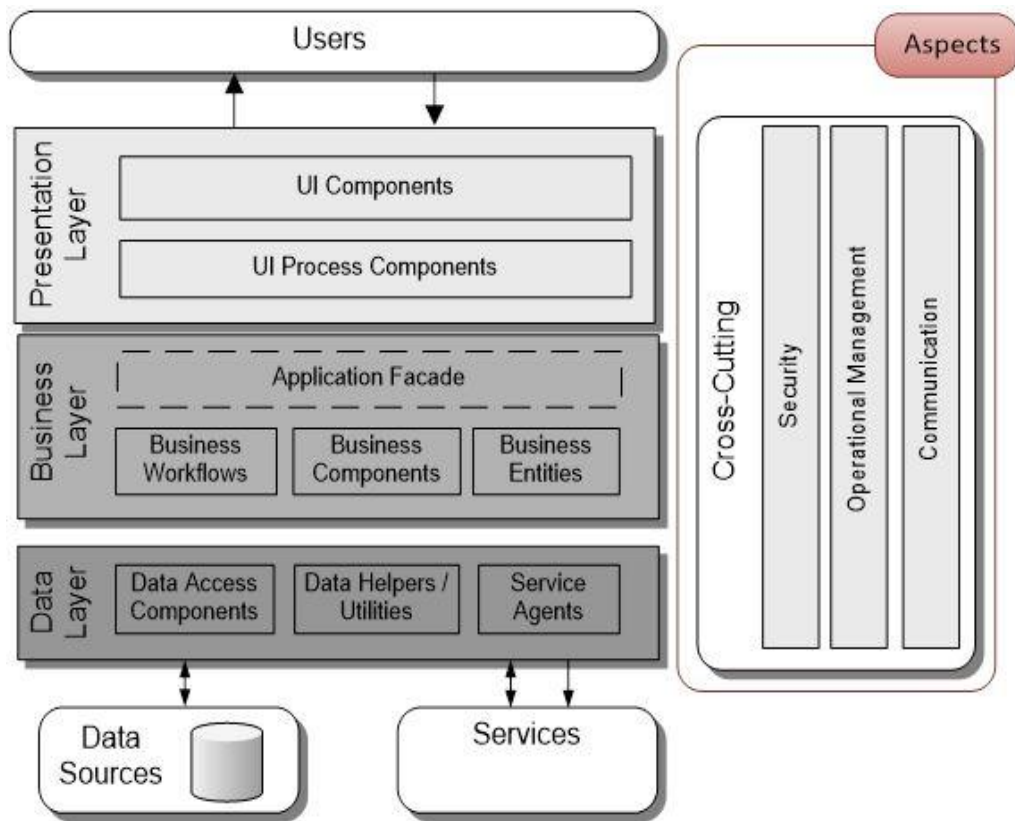


- Системы непрерывной интеграции (TeamCity, Bamboo, Jenkins)



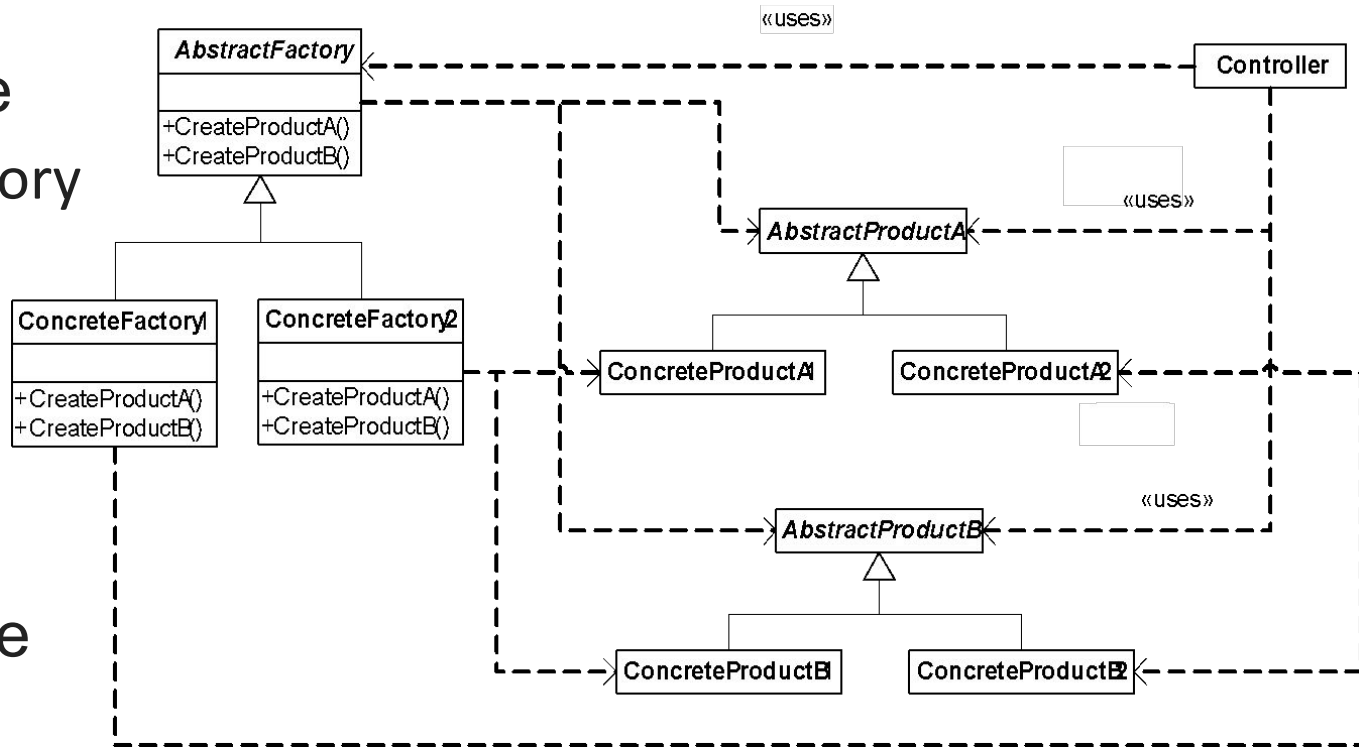
Jenkins

# Архитектура приложения



# Паттерны проектирования

- Порождающие
  - Abstract factory
- Структурные
  - Adapter
  - Façade
- Поведенческие
  - Strategy



# Принципы разработки

- KISS (Keep it short and simple)
- YAGNI (You aren't gonna need it)
- DRY (Don't repeat yourself)
- SOLID



# Code Style

```
function pow(x, n) {  
    var result = 1;  
    for (var i = 0; i < n; i++) {  
        result *= x;  
    }  
    return result;  
}  
  
var x = prompt("x?", "");  
var n = prompt("n?", "");  
if (n < 0) {  
    alert('Степень ' + n +  
        'не поддерживается, введите целую степень, большую 0');  
} else {  
    alert(pow(x, n));  
}
```

Пробел между параметрами

Между именем функции и скобкой ( между скобкой и первым параметром нет пробела)

Фигурная скобка { на той же строке, через пробел

Отступ 2 пробела

Пробелы вокруг операторов

Пробел после for

точка с запятой; обязательна

Пробел между параметрами

Пустая строка между логическими блоками

Длина строки не более 80 символов

} else { без перевода строки

Пробелы вокруг вложенного вызова