

Имитация Сложности

Простое не требует Сложного

Март 31, 2018

SAR
DOT
NET

О себе



NIKITA DANILOV

Lead Software Engineer

Back-end .NET Developer


7 лет в EPAM

Специализация:

.NET, SOA, WCF, WebAPI, MS SQL, CI/CD

Бизнес-область:

Страхование (Авто, Туризма, всего)



Простое → Сложное



Простое ← Имитация Сложности ← **Сложное**

Малое



Простое



Сложное и Сложность?



Сложное и Сложность?

CBO (Coupling Between Classes), **NOC** (Number of Children),
DIT (Depth of Inheritance), **WMC** (Weighted Methods per Class),
LCOM (Lack of Cohesion of Methods),
Complexity: **Cyclomatic**, **Data**, **Decisional**, **Data Flow**

СЛОЖНОСТЬ

Git

IoC/DI

NuGet

Docker

Chef

Лямбды

HTML

LESS

CSS

Flex

Рефлексия

RabbitMQ

MongoDB

PowerShell

Шаблоны

=> ?. ??

Angular

React

ООП

АЭС

CRM

Биржа

Вклады

Законы

Логика

ГОСТ

Сети

ISO

Люди

Опционы

Финансы

Логистика

Экология

Заводы

Физика

Бизнес

Химия

Счета

ОНА ПОВСЮДУ





Простое ← **Имитация Сложности** ← Сложное

imitari (праиндоевр.) — подражать, воспроизводить.

imitatio (лат.) — подражание.

 Project.Contracts

 Project.Core

 Project.Data

 Project.Web


 Project.Facades

 Project.FooClient


 Project.Proxies

 Project.Contracts

 Facades

 IItemFacade

 IUserFacade

 Entities

 IItem

 IUser

 IValidatable

 Project.Facades

 Item

 ItemFacade

 User

 UserFacade

 Project.Web

 Controllers

 ItemController

 App_Start

 WebApiConfig

 FilterConfig

 Startup



Project.Contracts



Facades

C# IItemFacade

C# IUserFacade



Entities

C# IItem

C# IUser

C# IValidatable



Project.Facades



Item

C# ItemFacade



User

C# UserFacade



Project.Web



Controllers

C# ItemController



App_Start

C# WebApiConfig

C# FilterConfig

C# Startup

Controller

...

Controller

Accessor

Manager

Helper

Provider

Gateway

Facade

Controller

Accessor

Proxy

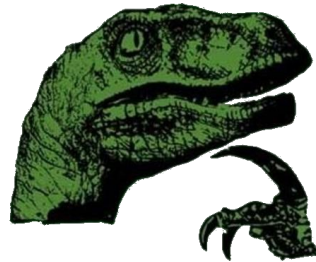
Repository

Service

Processor

Client

Имитация Сложности?



Имитация сложности - подражание сложности.

Подсмотренной в книгах или на конференциях.
Чтобы выглядеть посложнее, солиднее, богаче,
изучить новый подход, ~~вдруг диплом не примут.~~



Простое ← Имитация Сложности ← Сложное

KISS = Keep It Simple Stupid



1. Проще → Быстрее Разработка.

Например:



1. Проще → Быстрее Разработка.

Еще пример:



2. Проще → Надежнее Работа.

Например:

Вероятность ошибки в строке кода?

1% или 0,1% или 0,01%?

$$0,01\% * 10,000 \text{ SLOCs} = 100\%$$

Что то здесь не так...



$$0,0001\% * 10,000 \text{ SLOCs} = 1\%$$

$$1\% * 100 \text{ Зависимостей} = 100\%$$



3. Проще → Понятнее для Поддержки.

Например:

Абстракция для Упрощения = Моделирования?

Абстракция для сокрытия Сложности?

Manager

Helper

Provider

Gateway

Facade

Controller

Accessor

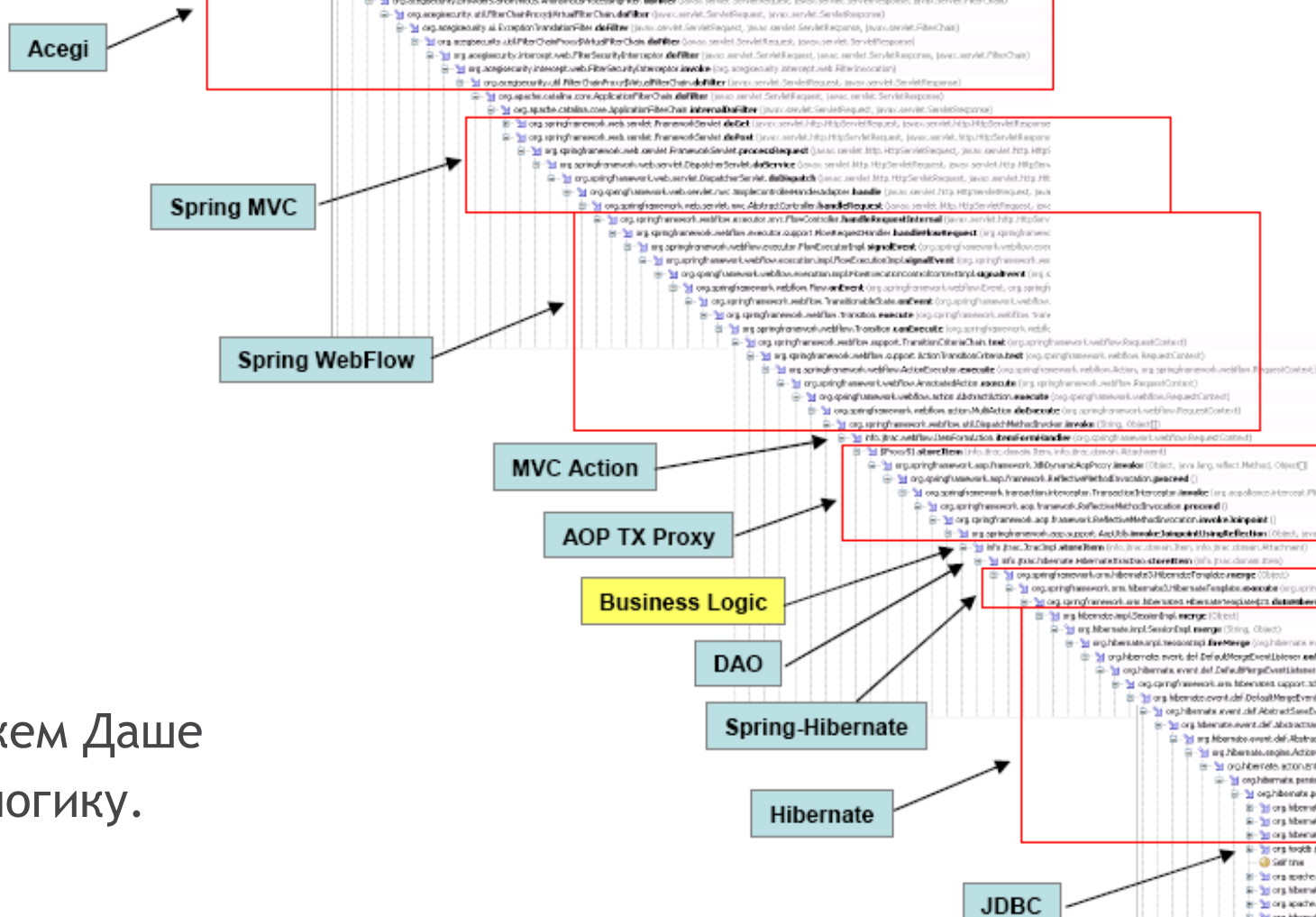
Proxy

Repository

Service

Processor

Client



<epam>

```
class Controller
{
    Controller(IManager, ILogger, ISettings) { ... }

    Load()
    {
        ...
        var data = _manager.Load(...);
        ...
    }
}
```

```
class Manager
{
    Manager(IFacade, ILogger, IChecker, IWrapper) { ... }

    Load()
    {
        ...
        var data = _facade.Load(...);
        ...
    }
}
```



```
class Facade
{
    Facade(IAccessor, ILogger, IPermission, IPhone) { ... }

    Load()
    {
        ...
        var data = _accessor.Load(...);
        ...
    }
}
```

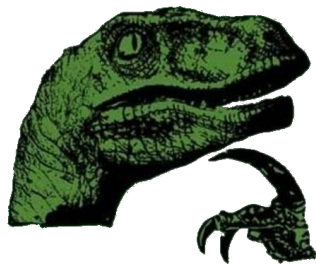
```
class Accessor
{
    Accessor() { }

    Load(...)
    {
        NLog.Instance.Write(...);
        var context = new DbContext("connectionString");
        var data = context.Entities.ToList();
    }
}
```

Скорость, Надежность, Поддерживаемость.

«Обалдеть, дайте две!» ©

Почему не делаем всё Просто?





Скорость

Поддерживаемость

Надежность



~~Кто виноват?~~ Что делать?

Упрощайте постановку проблемы до начала решения.

Simplify the problem state before you start.

«Всё следует упрощать до тех пор,
пока это возможно, но не более того»
Альберт Эйнштейн



Простое ↔ Имитация Сложности ↔ Сложное

Ссылки

- Software: Managing the Complexity,
<https://medium.com/@gaperton/software-managing-the-complexity-caff5c4964cf>
- Keep It Simple Stupid (KISS),
http://principles-wiki.net/principles:keep_it_simple_stupid
- [Основы теории надежности](#) ,
Расчет показателей надежности невозстанавливаемых нерезервированных систем.



Спасибо