



Как сложность убивает проекты

Вадим Мингажев

О чем

- Почему нужно бороться со сложностью
- Откуда берется
- Кто и как может с ней бороться



Сложность – это усилия, потраченные коллективом на создание определенного количества материала

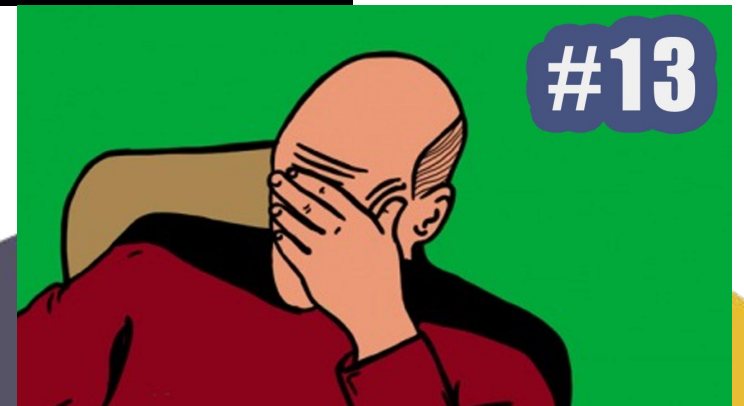
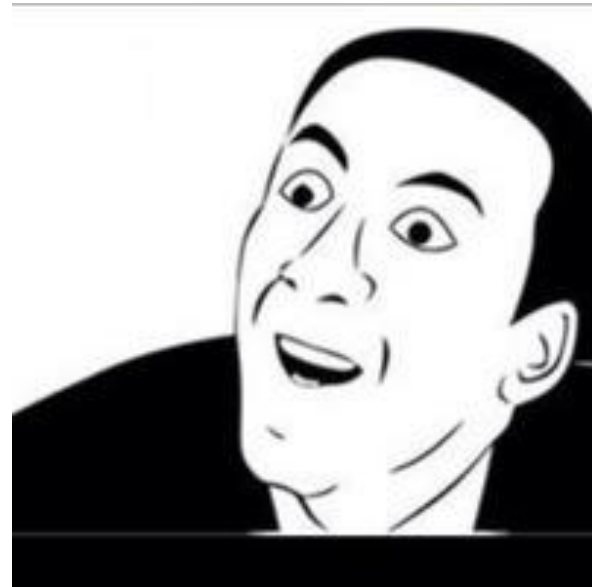
Управление сложностью каждый день

- Команда должна понимать и дописывать код (хотя бы в текущем проекте)
- Сложность, остается в проекте надолго
- Сложность «заразна»
- Чтобы код понимал аналитик, РО, тестировщик



Откуда появляется сложность

- «Протекание» и смешение уровней абстракции
- Неправильное именование
- Слабая связанность, сильная связность
- Отсутствие unit-тестов
- Не применяются SOLID, YAGNI ...
- ...



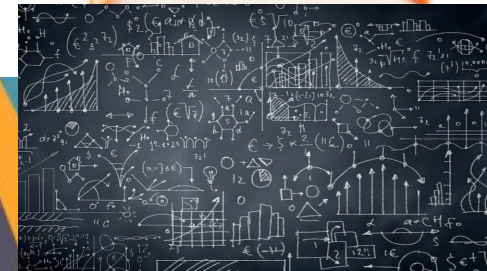
Что мешает развитию системы, в которой в общем-то все неплохо

- Сложность предметной области
- **Сложность, внесенная разработчиком**
- **Сложность, внесенная умелым разработчиком**
- ~~Сложный язык, Framework, сложные задачи~~
- ~~Сложно поднимать окружение~~



Оверинжиниринг

- Желание решить задачу выше своих компетенций
- Недостаточное понимание задачи
- Чаще встречается у молодых разработчиков



Желание написать «классно»


- Желание использовать модный фрэймворк/подход
- Желание развить компетенции
- Чаще свойственно разработчикам среднего уровня



Гений

- Гибкий и качественный код
- Продуманные точки расширения
- Сложные алгоритмы



- 
- CodeReview для всех
 - Отдельный пункт CodeReview – **«проверить понятность»**,
 - Управление творческой энергией разработчика



Практики из SCRUM

- Не давать задачу «целиком»
- Force collective code ownership
- Коммуникации

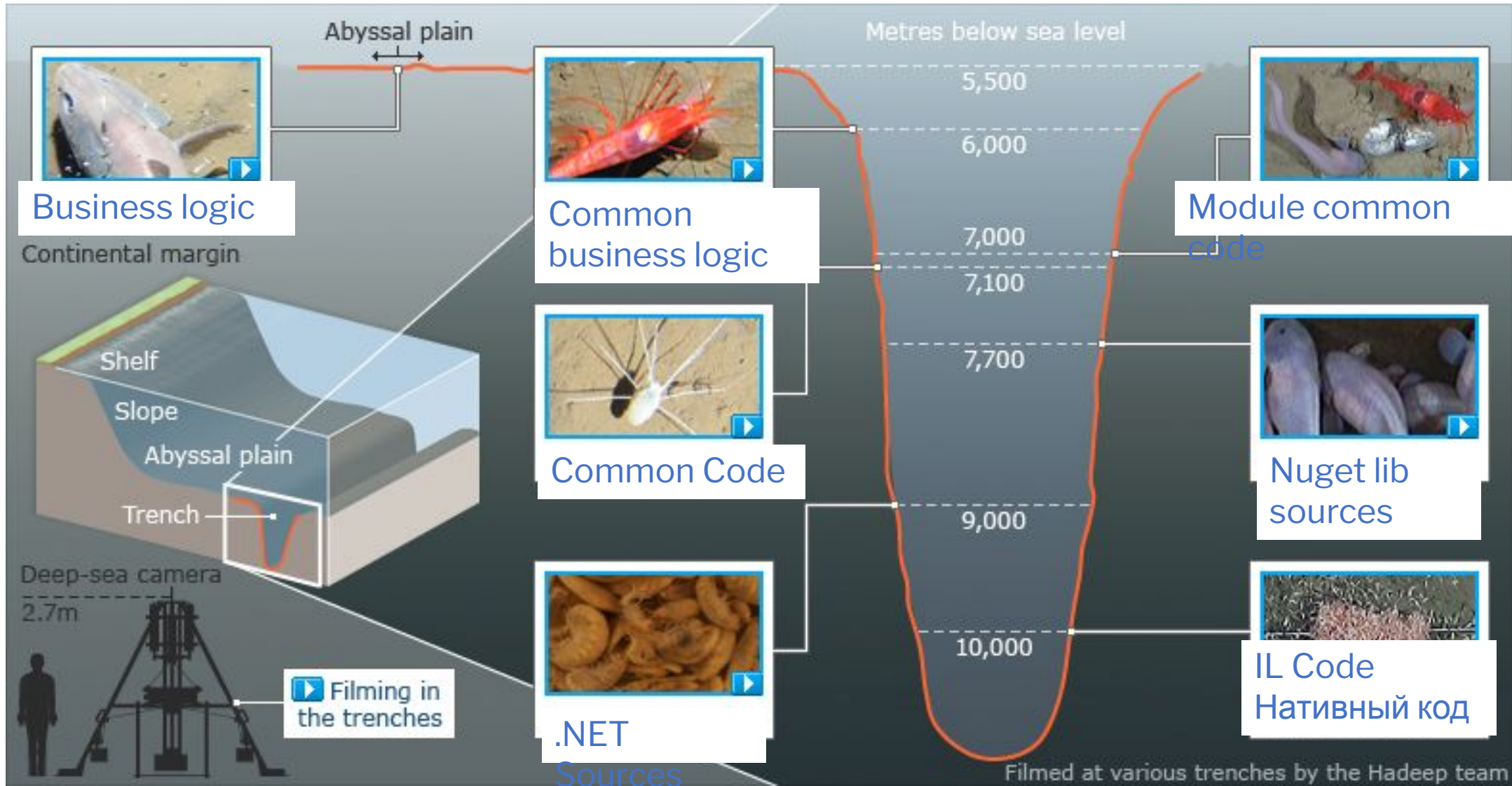


Как повысить качество и *сложность* кода

- DI/IoC
- AOP
- DDD и Enterprise-паттерны
- Асинхронность/многопоточность
- Обобщенные классы и алгоритмы
- Использование готовых КОМПОНЕНТОВ
- Функциональный стиль
- ...



Работа на глубине



Enterprise

**ServiceWrapp
er**

Locator

Layer

ActiveRecord

Tier

Interceptor

Service

Repository

Bus

Broker

Adapter

Microservices

Уровень классов

```
/// < Summary >  
[ServiceBehavior(InstanceContextMode = InstanceContextMode.PerCall, ConcurrencyMode = ConcurrencyMode.Multiple)]  
[ErrorHandlerBehavior]  
[ServerRequestLogBehavior]  
[CheckClientVersionBehavior]  
public class AnyService : IAnyService  
{
```

Поля и свойства

```
[CheckerClientId]  
public byte[] GetAllMetadata()  
{
```

```
public static byte[] GetAllMetadata(Stream outputStream)
```

```
{  
    return AnyManagerBase.GetAllMetadata(outputStream)(outputStream);  
};
```

Services

(Implementation)

Services

Уровень методов

```
return specialFolders.Where(f => f.Kind == SpecialFolderKind.Computable)
    .Where(folder=>!ResourceValidator.Validate(module, folder.NameResourceKey))

return specialFolders.Where(f => f.Kind == SpecialFolderKind.Computable)
    .Where(folder=>!ResourceValidator.Validate(module, folder.NameResourceKey))
    .Select(folder =>
    {
        foreach (var folder in specialFolders.Where(f => f.Kind == SpecialFolderKind.Computable))
        {
            if (!ResourceValidator.Validate(module, folder.NameResourceKey))
                errors.Add(string.Format(emptyFolderTitleTemplate, folder.Name));

            if (folder.DefaultContentType == Guid.Empty || folder.DefaultContentMetadata == null)
                errors.Add(string.Format(emptyDefaultContentTypeTemplate, folder.Name));

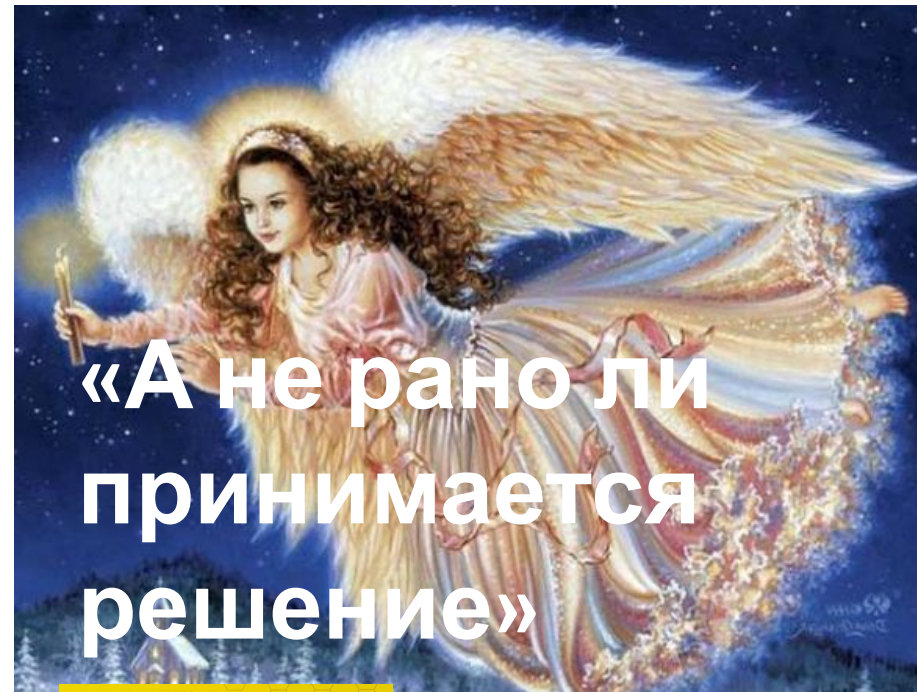
            if (folder.FilterPanel != null)
                errors.AddRange(new FilterPanelValidator(folder.FilterPanel).Validate());

            var folderUseRibbonActions = folder.SeparatedAdditionalRibbonActionNames.Any() ||
                folder.SeparatedAdditionalContextMenuActionNames.Any();

            if (folderUseRibbonActions && folder.AdditionalActionsOwner == null)
                errors.Add(string.Format(invalidRibbonActionsTemplate, folder.Name));
        }
    });

return errors;
```

Что делать?



#1

```
var getFileRequest = new RestRequest(fileId, Method.GET);
var fileBytes = new RestClient(serviceUrl).DownloadData(getFileRequest);
{
    // logic
}
```

#2

```
using (var client = new HttpClient())
{
    var message = await client.GetAsync($"{serviceUrl}/{fileId}");
    using (var stream = await message.Content.ReadAsStreamAsync())
    {
        // logic
    }
}
```

Ограничение набора инструментов

- Знать возможности используемого фреймворка
- Предпочитать встроенные инструменты
- Бюрократия при добавлении новых библиотек – это не всегда плохо



Непреодолимая сложность

- Требования к производительности
- Сложность предметной области
- Компромиссы и ограничения фреймворка, библиотек и бизнес-требований



#1 Просто код

```
public static bool BuffersAreEqual(byte[] buffer1, int buffer1Offset, int buffer1Count, byte[] buffer2,
    bool success = (buffer1Count == buffer2Count);
    for (int i = 0; i < buffer1Count; i++) {
        success &= (buffer1[buffer1Offset + i] == buffer2[buffer2Offset + (i % buffer2Count)]);
    }
    return success;
}
```

#1 КОНТЕКСТ

```
// !! DO NOT CHANGE THIS METHOD WITHOUT SECURITY~
[MethodImpl(MethodImplOptions.NoOptimization)]
public static bool BuffersAreEqual(byte[] buffer1, int buffer1Offset, int buffer1Count, byte[] buffer2,
    bool success = (buffer1Count == buffer2Count);
    for (int i = 0; i < buffer1Count; i++) {
        success &= (buffer1[buffer1Offset + i] == buffer2[buffer2Offset + (i % buffer2Count)]);
    }
    return success;
}
```

#1 Документация и комментарии

```
// Determines if two buffer instances are equal, e.g. whether they contain the same payload. This method
// is written in such a manner that it should take the same amount of time to execute regardless of
// whether the result is success or failure. The modulus operation is intended to make the check take the
// same amount of time, even if the buffers are of different lengths.
//
// !! DO NOT CHANGE THIS METHOD WITHOUT SECURITY~
[MethodImpl(MethodImplOptions.NoOptimization)]
public static bool BuffersAreEqual(byte[] buffer1, int buffer1Offset, int buffer1Count, byte[] buffer2, int buffer2Offset,
    bool success = (buffer1Count == buffer2Count); // can't possibly be successful if the buffers are of different lengths
    for (int i = 0; i < buffer1Count; i++) {
        success &= (buffer1[buffer1Offset + i] == buffer2[buffer2Offset + (i % buffer2Count)]);
    }
    return success;
}
```

#2 Просто код


```
var invoker = new DynamicMethod(string.Empty, typeof(DesignTimeMetadataBase), new[] { typeof(MetadataBase) }, true);
var generator = invoker.GetILGenerator();
generator.Emit(OpCodes.Ldarg_0);
generator.Emit(OpCodes.Newobj, constructor);
generator.Emit(OpCodes.Ret);
return (CreateDesignTime)invoker.CreateDelegate(typeof(CreateDesignTime));
```

#2 Комментарии

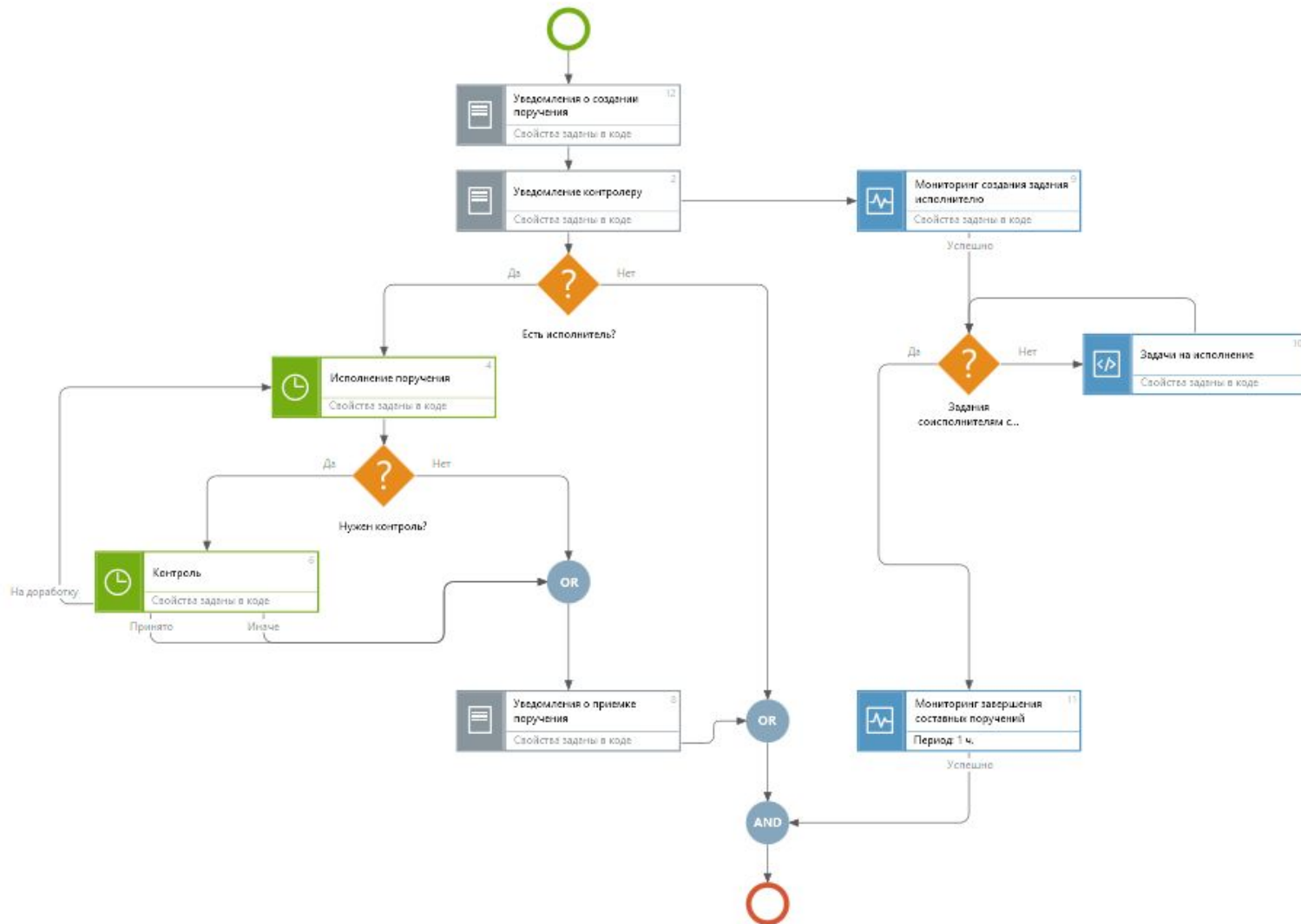
```
// Для создания экземпляра используется ILCodeGenerator, т.к. работает быстрее чем Reflection.  
var invoker = new DynamicMethod(string.Empty, typeof(DesignTimeMetadataBase), new[] { typeof(MetadataBase) }, true);  
var generator = invoker.GetILGenerator();  
generator.Emit(OpCodes.Ldarg_0);  
generator.Emit(OpCodes.Newobj, constructor);  
generator.Emit(OpCodes.Ret);  
return (CreateDesignTime)invoker.CreateDelegate(typeof(CreateDesignTime));
```




Документация, комментарии

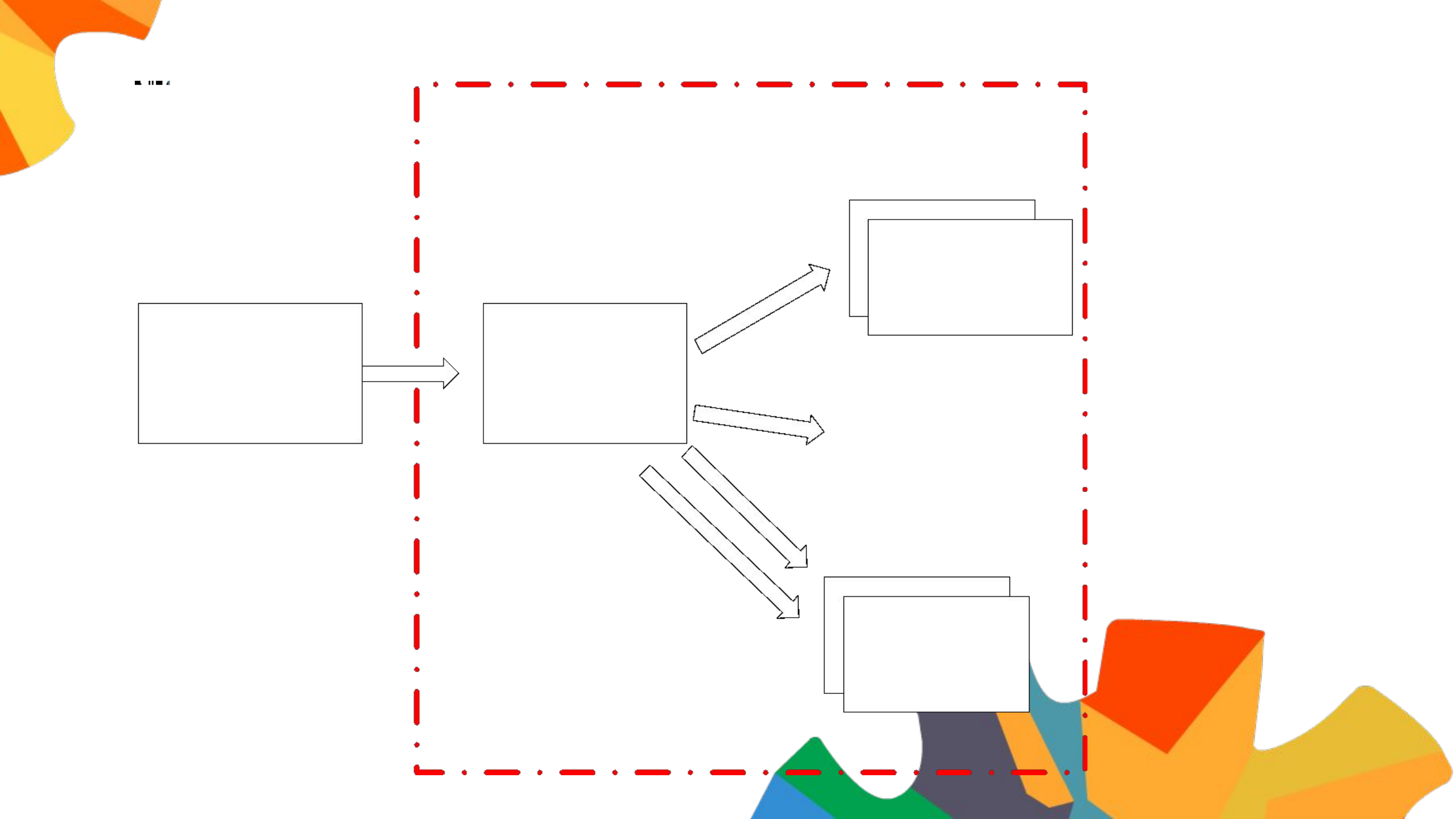
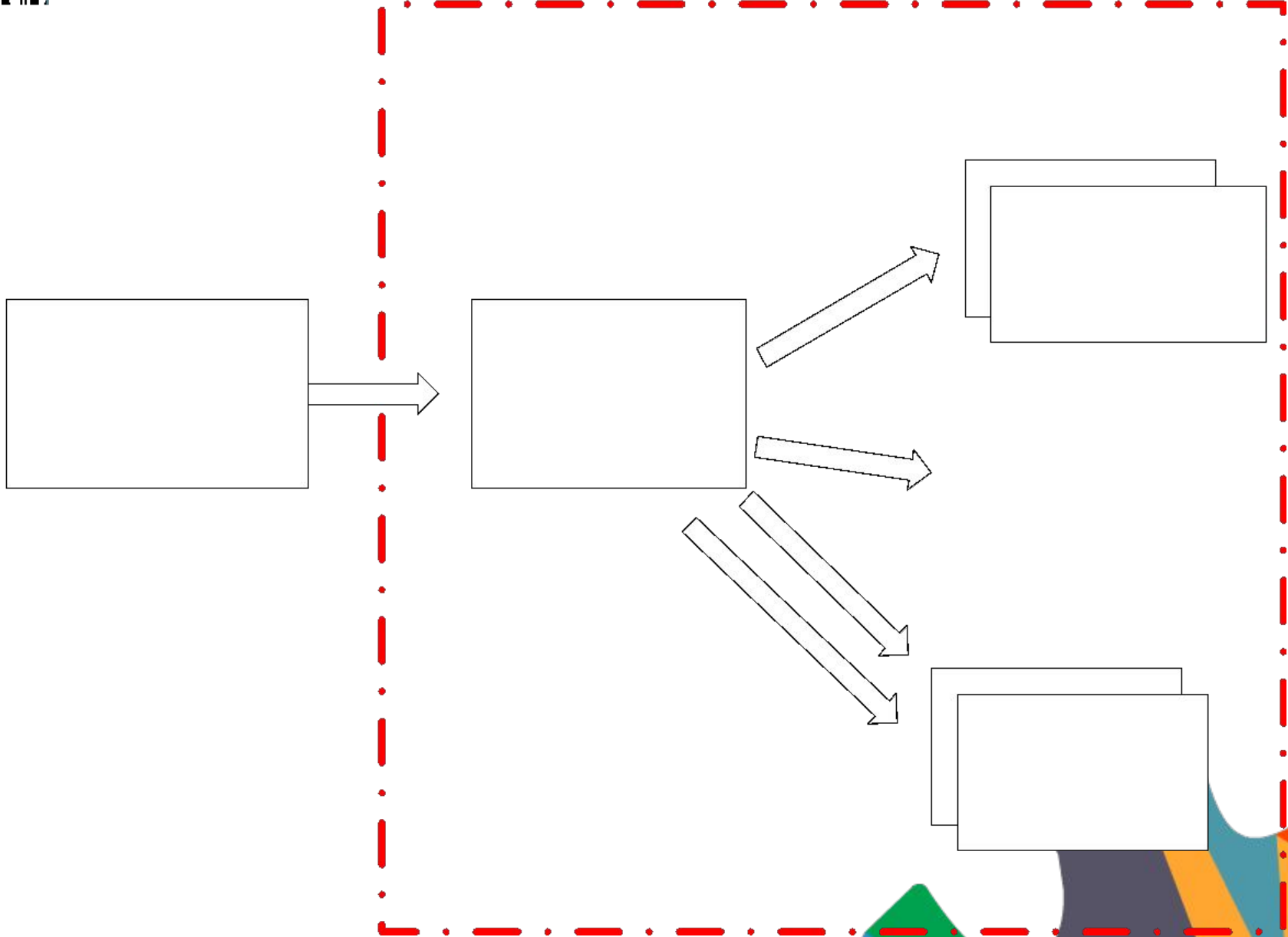
- Описание сложного алгоритма (возможно со ссылками)
 - «Почему так»
 - Контекст
- 

Валидация графа



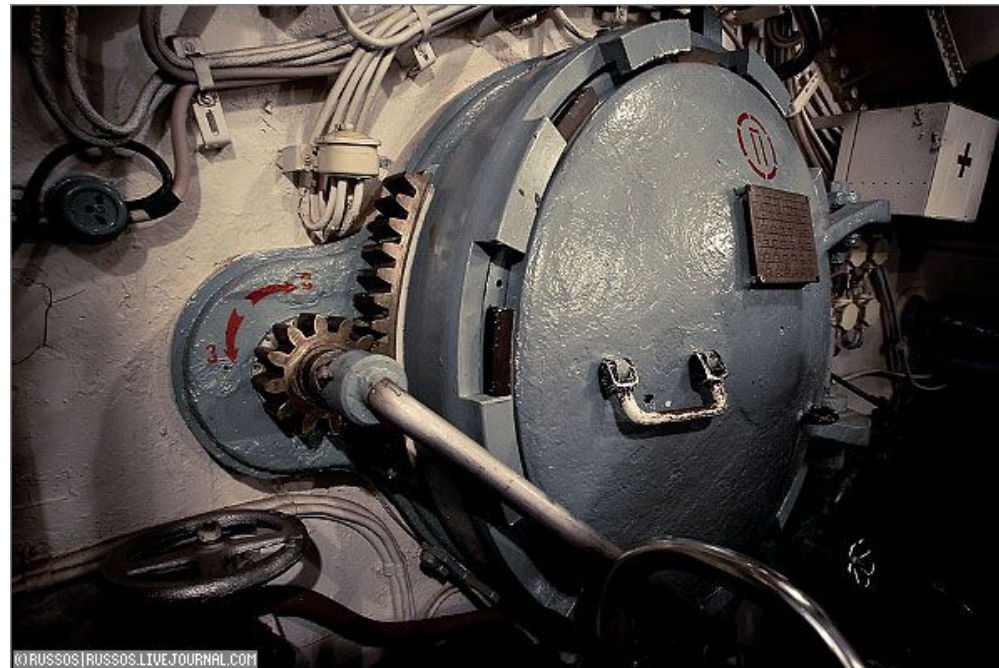
Алгоритм

- Поиск «треугольников»
- Упрощение графа с использованием шагов:
 - Убрать «петли»
 - Сократить пути, относящиеся к одной параллельности
 - Убрать «тупики»
 - Упростить пути, сходящиеся в одной параллельности
- еще сколько-то шагов



Изоляция/Инкапсуляция

- Сложность не отражается на публичных интерфейсах
- От сложности ничего не зависит
- Сложность сконцентрирована в одном месте





Вадим Мингажев

Разработчик



DIRECTUM в г. Уфа

Mingazhev_VR@directum.ru



Благодарности:

Тимур Асадуллин -
рецензирование