



# Управление зависимостями

Сергей Шкредов,  
dev lead at JetBrains

# Инструменты разработчика

1. Закрытость

2. Нецелевое использование:

- Программирование на JavaScript
- Использование ООП для всего
- Проекты для контроля зависимостей

# Проекты и сборки

## 1. Что это?

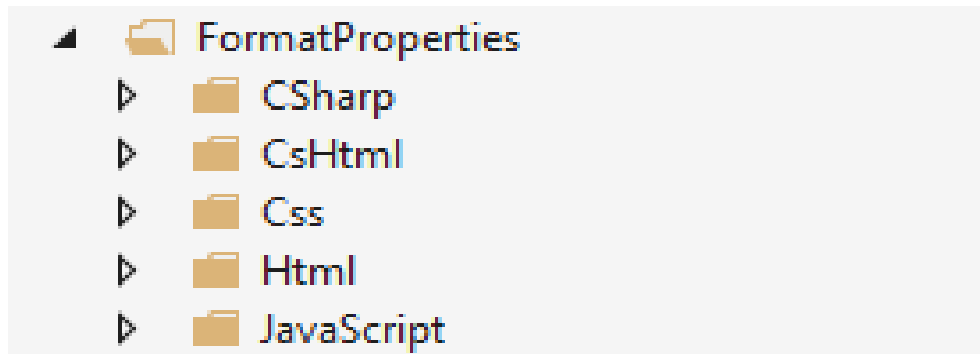
- Единица компиляции
- Единица разработки
- Единица исполнения
- dll-ка, jar-ник

# А что мы делаем?

1. Поставляем пакеты
  - Но разрабатываем сборки
2. Контролируем архитектуру
  - Используем сборки
  - Но хотим большей гранулярности

# С чем мы сталкивались?

- `internal` – продлить область видимости:
  - `InternalsVisibleTo`
- `internal` – сузить область видимости:

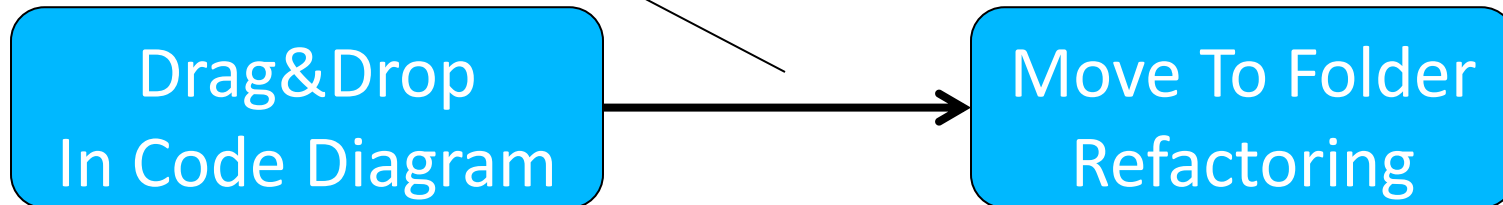


# Про зависимости

- DIP – dependency inversion principle
- IOC – inversion of control
- SOLID
- DI – dependency injection
- **Как помогают инструменты?**

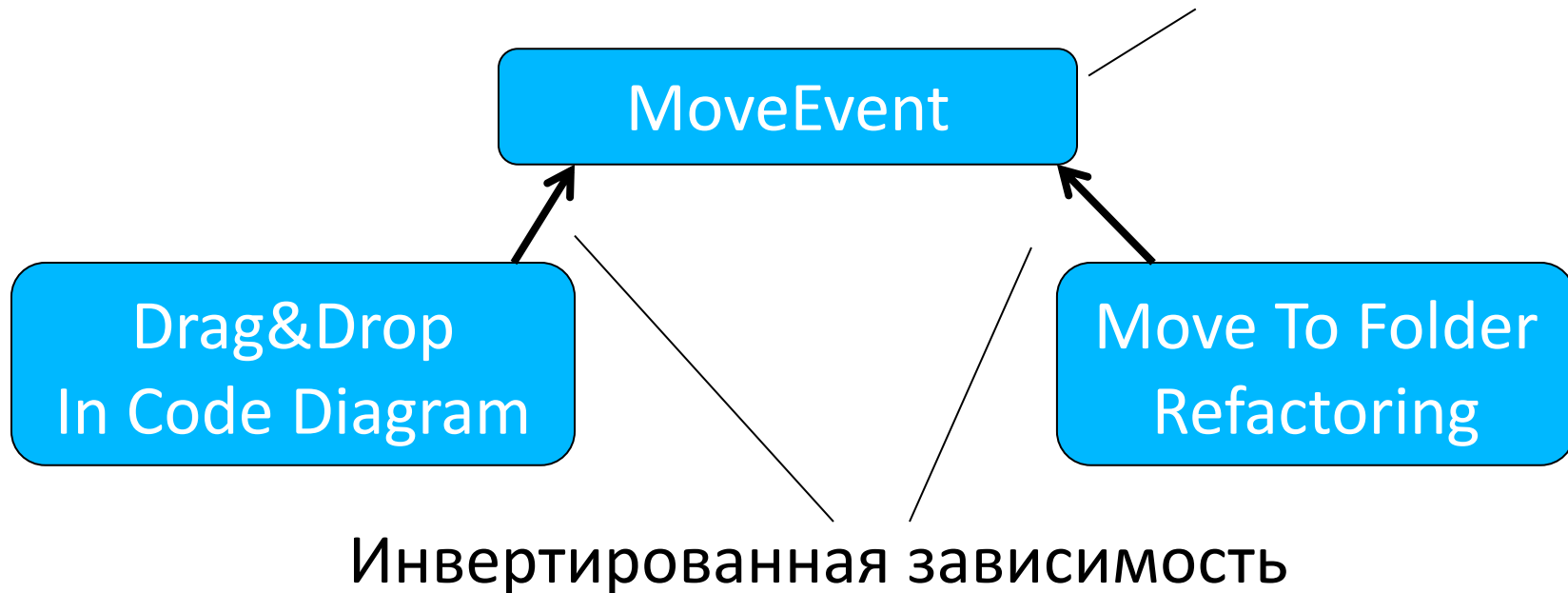
# Инверсия зависимостей

Прямая зависимость



# Инверсия зависимостей

Синтез абстракций

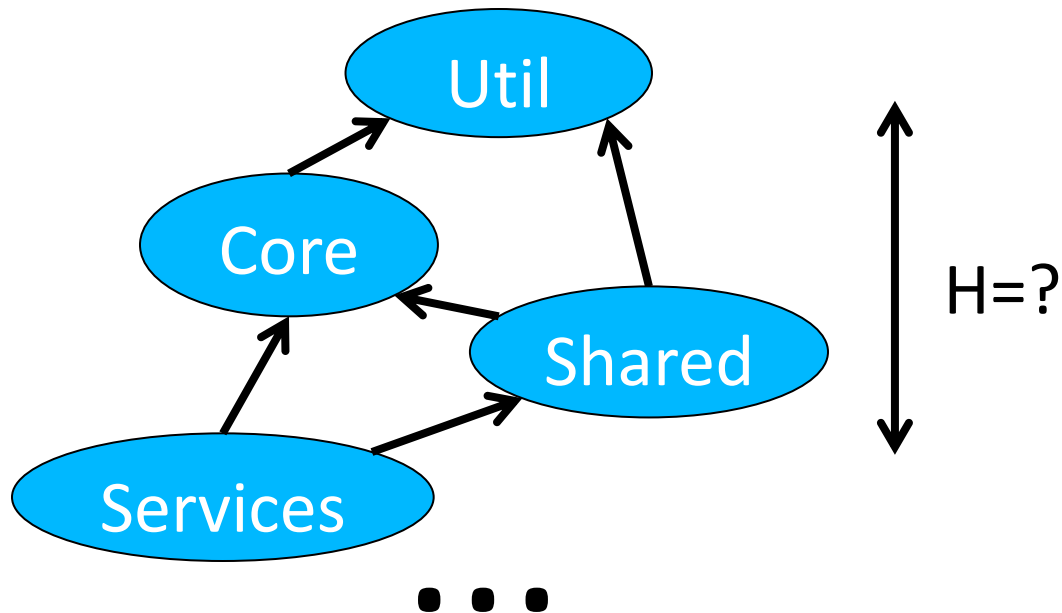




# SRP?

```
public IntellisenseManager(  
    IShellLocks locks,  
    ISolution solution,  
    ISymbolCache cacheManager,  
    ISettingsStore settingsStore,  
    DocumentManager documentManager,  
    IActivityTracking activityTracking,  
    ILookupWindowManager lookupWindowManager,  
    LookupItemsOwnerFactory lookupItemsOwnerFactory,  
    IEnumerable<IAutomaticCodeCompletionStrategy> strategies,  
    IDeclaredElementDescriptionPresenter declaredElementDescriptionPresenter,  
    CompletionStatisticsManager completionStatisticsManager,  
    IEnumerable<ICodeCompletionDisabler> disablers,  
    IntellisensePartsContainer intellisenseParts,  
    ITextControlManager textControlManager,  
    ILanguageManager languageManager,  
    PsiIconManager psiIconManager,  
    FontsManager fontsManager,  
    IPsiServices psiServices,  
    Lifetime lifetime  
)
```

# Используем сборки



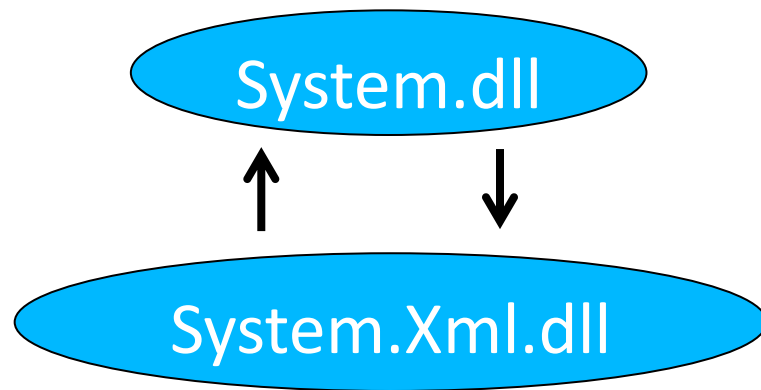
# Используем сборки

System.dll

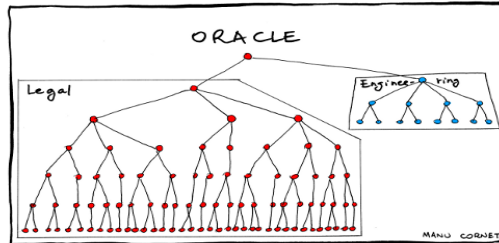
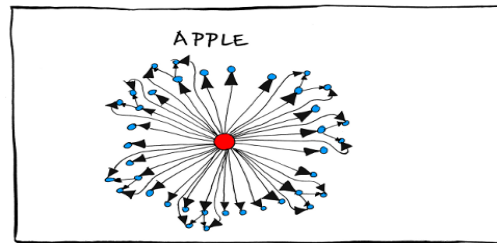
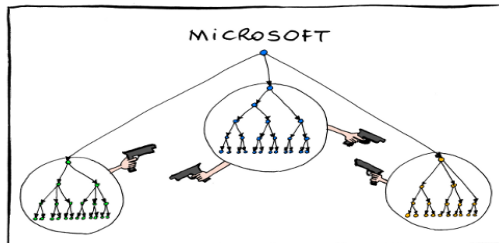
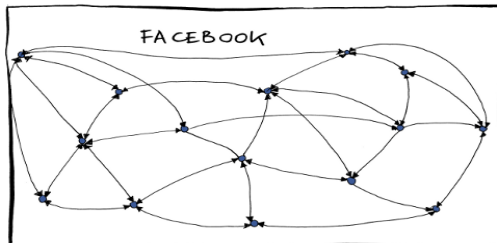
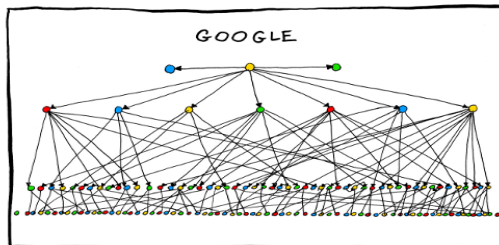
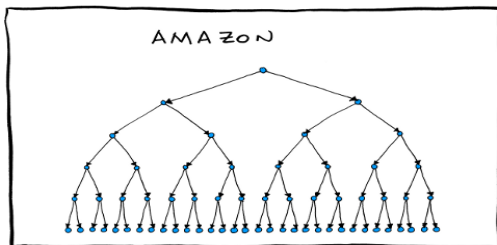
?

System.Xml.dll

# Используем сборки



# Как мы работаем

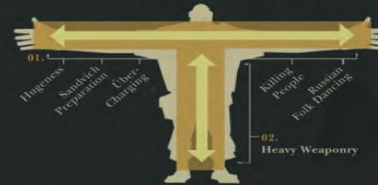


# Как мы работаем

Valve, a company that makes computer games, tries to hire only “T-shaped people.” They explain it this way:

“We value ‘T-shaped’ people. That is, people who are both generalists (highly skilled at a broad set of valuable things—the top of the T) and also experts (among the best in their field within a narrow discipline—the vertical leg of the T). This recipe is important for success at Valve. We often have to pass on people who are very strong generalists without expertise, or vice versa. An expert who is too narrow has difficulty collaborating. A generalist who doesn’t go deep enough in a single area ends up on the margins, not really contributing as an individual.”

## T-SHAPED MODEL: EMPLOYEE



01. Broad-range generalist 02. Deep expertise in one area

What do you think of Valve’s idea about hiring only T-shaped employees? Who is the most T-shaped person that you know? Why? How are you doing at being T-shaped? What would it take for you to become T-shaped?

\* image and quote are from the Valve employee manual, which showed up online this week in numerous places.

# УЖЦ (ака ALM)

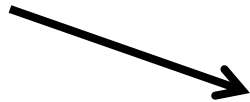
```
public static class StringUtil
{
    public const char DOT = '.';
    public const char GENERIC_DELIMITER = '';
    public const char PLUS = '+';
    public const string SDOT = ".";

    /// <summary>
    /// In order to get varying results on adjacent calls, there must i
    /// </summary>
    /// </summary>
    [NotNull]
    private static readonly Random _random = new Random();

    private static readonly string[] ourPrepositions =
    {
        "of", "the", "for", "in", "as", "at", "by", "from", "to", "via"
    };

    /// <summary>
    /// Caches the macro substitution regex.
    /// </summary>
    private static Regex ourRegexMacro;

    /// <summary>
    /// Returns true if one qualified name starts with another.
    /// </summary>
    public static bool StartsWith(string what, string withwhat, String
    {
        if (what == null || withwhat == null)
            return false;
        if (withwhat == String.Empty)
            return true;
        if (!what.StartsWith(withwhat, comparison))
            return false;
        if (what.Length == withwhat.Length)
            return true;
        return what[withwhat.Length] == DOT;
    }
}
```



# Что еще случится?

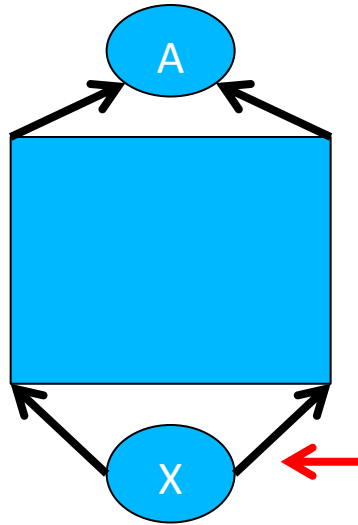
1. Код перейдет другим разработчикам
2. Изменятся внешние условия
3. Изменятся Ваши представления о мире



# Резюме

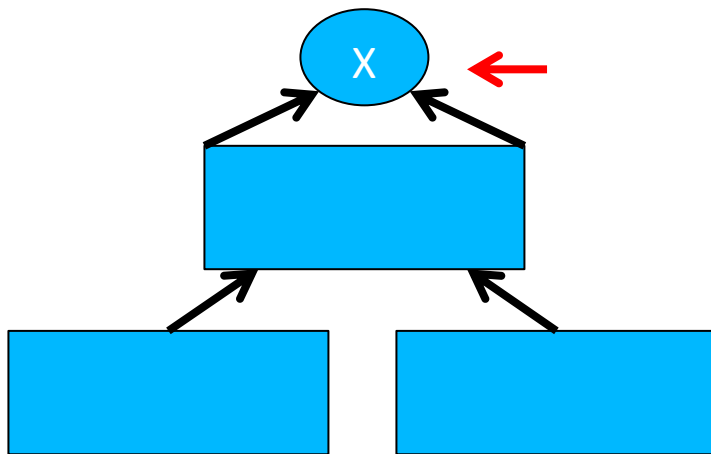
1. Управление пакетами
2. Управление зависимостями
3. Распространение знаний
4. Взаимодействие команд
5. Распределенная ответственность

# А как у Вас?



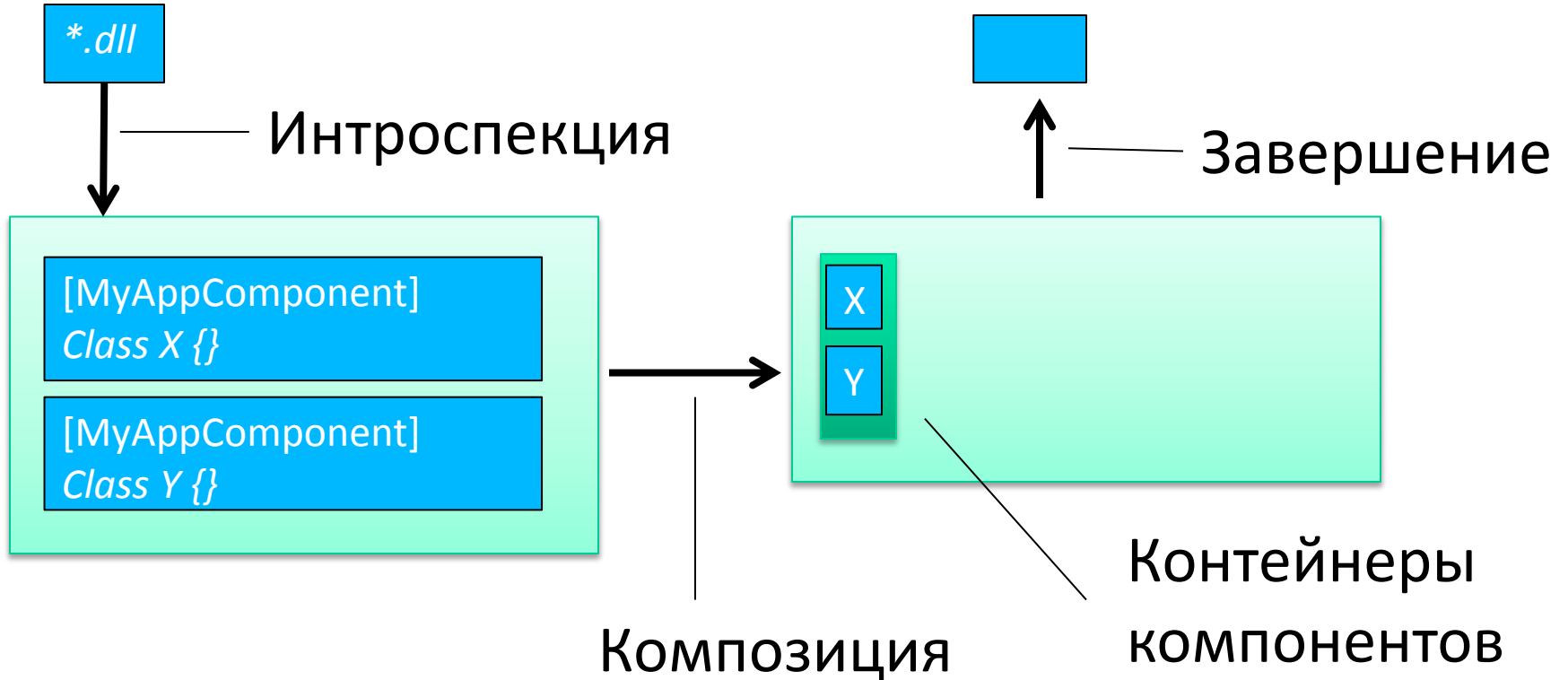
В точно входа мы  
«знаем» про все компоненты

# А как у Вас?

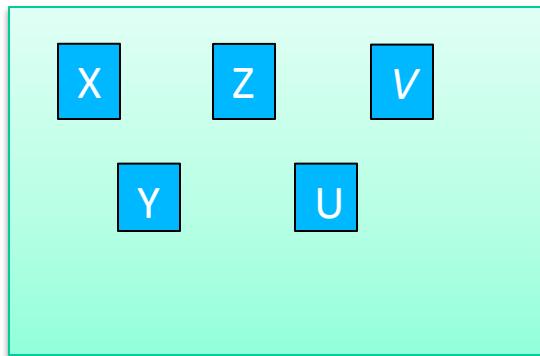


В точке входа мы  
«знаем» только  
что и как загрузить

# Расширяемая программа

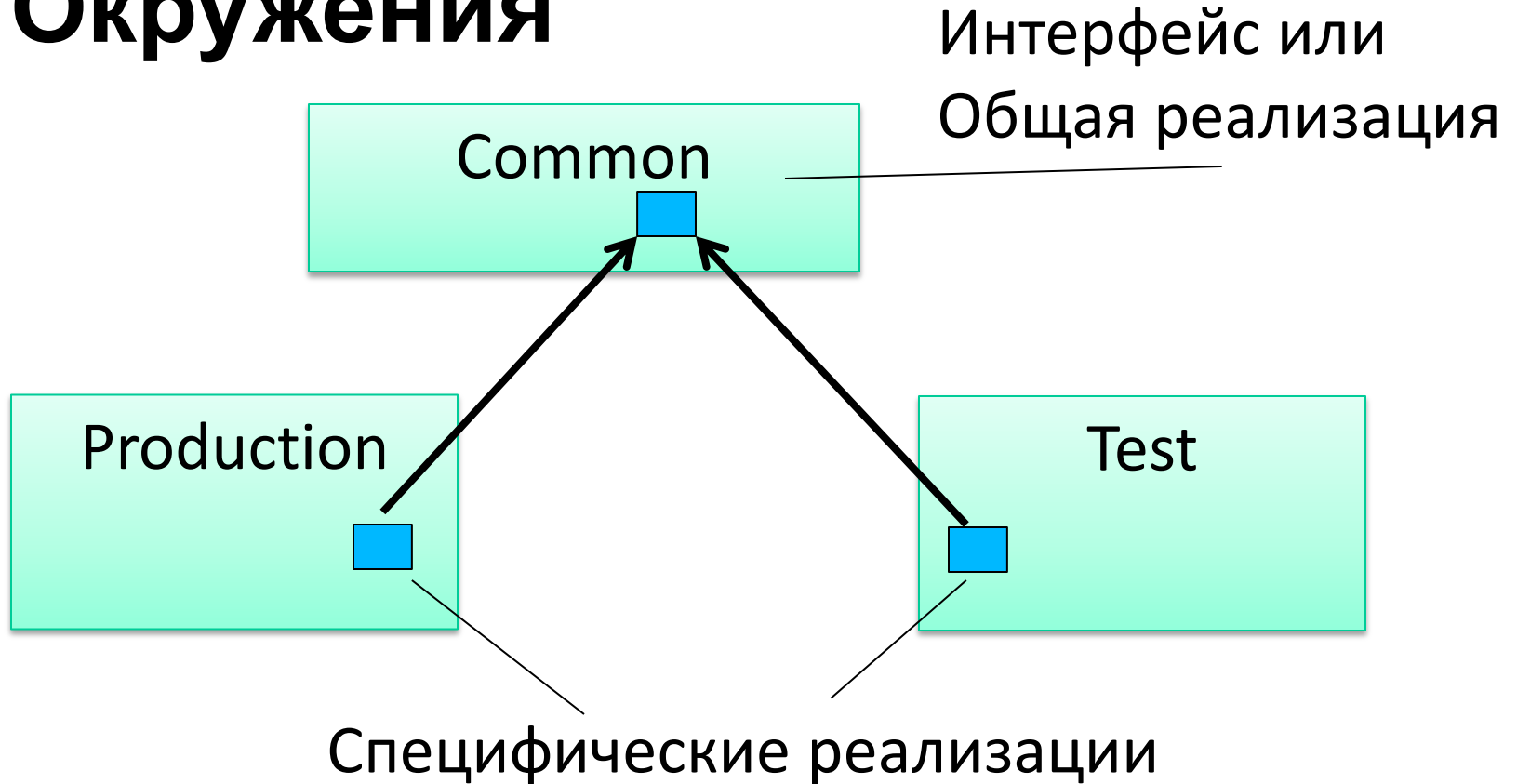


# Контейнеры

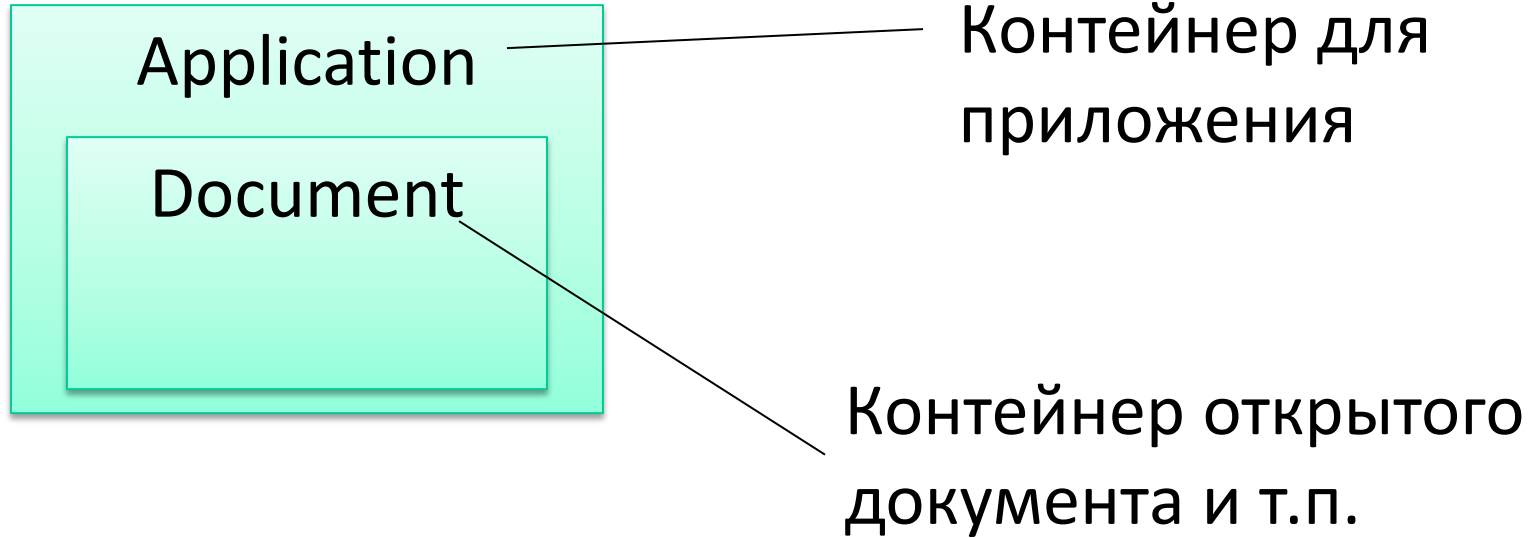


1. Создание
2. Переиспользование
3. Инъекция зависимостей
4. Метапрограммирование

# Окружения



# Время жизни



# Расширяемость

```
CodeCompletionItems (  
    IEnumerable<IItemsProvider> ps  
)
```

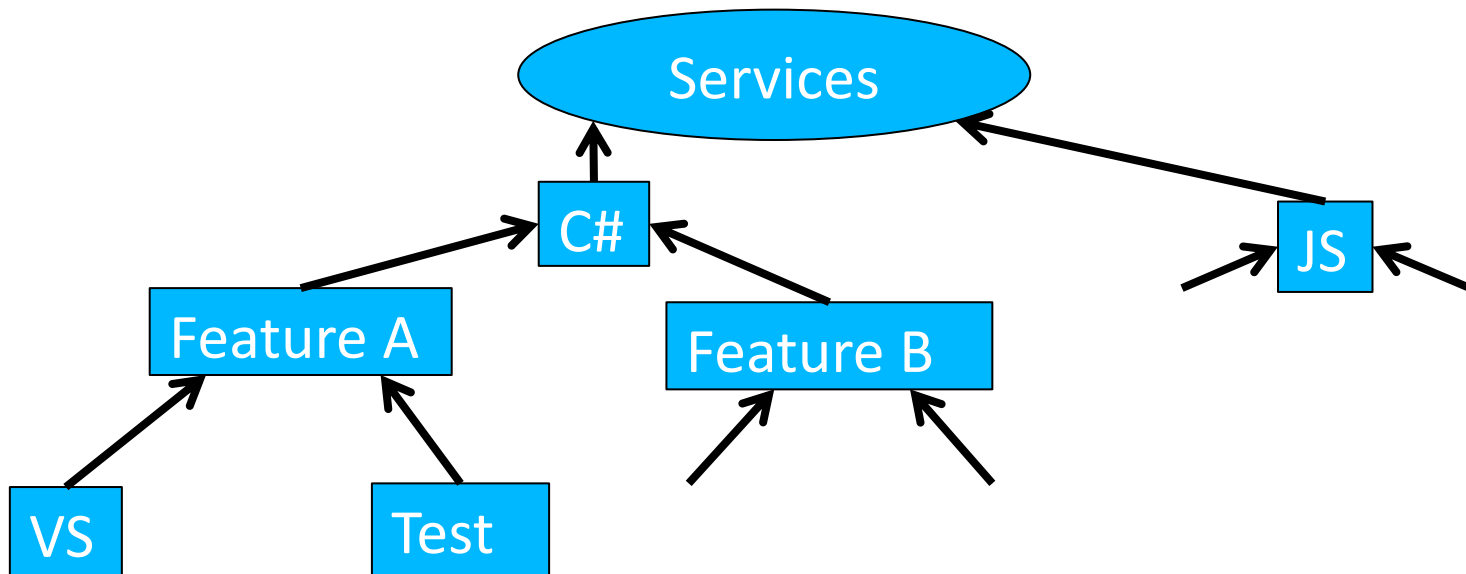
- Агрегация результатов
- Модификация поведения
- Приоритизация



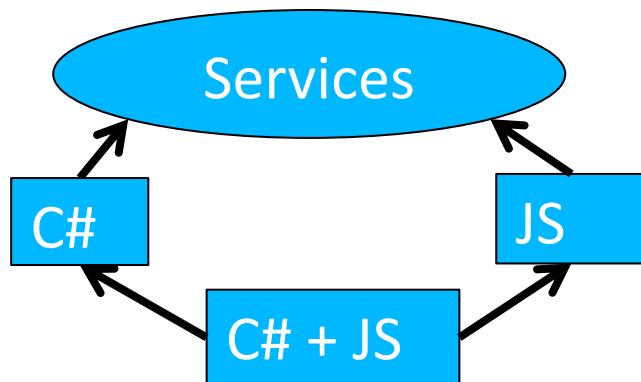
# ReSharper

- Общий код
- 15+ Языков
- Реализация 50+ различных функций
- 5+ Окружений
- 30+ Разработчиков
- 8+ Продуктов

# Сборки для управления зависимостями



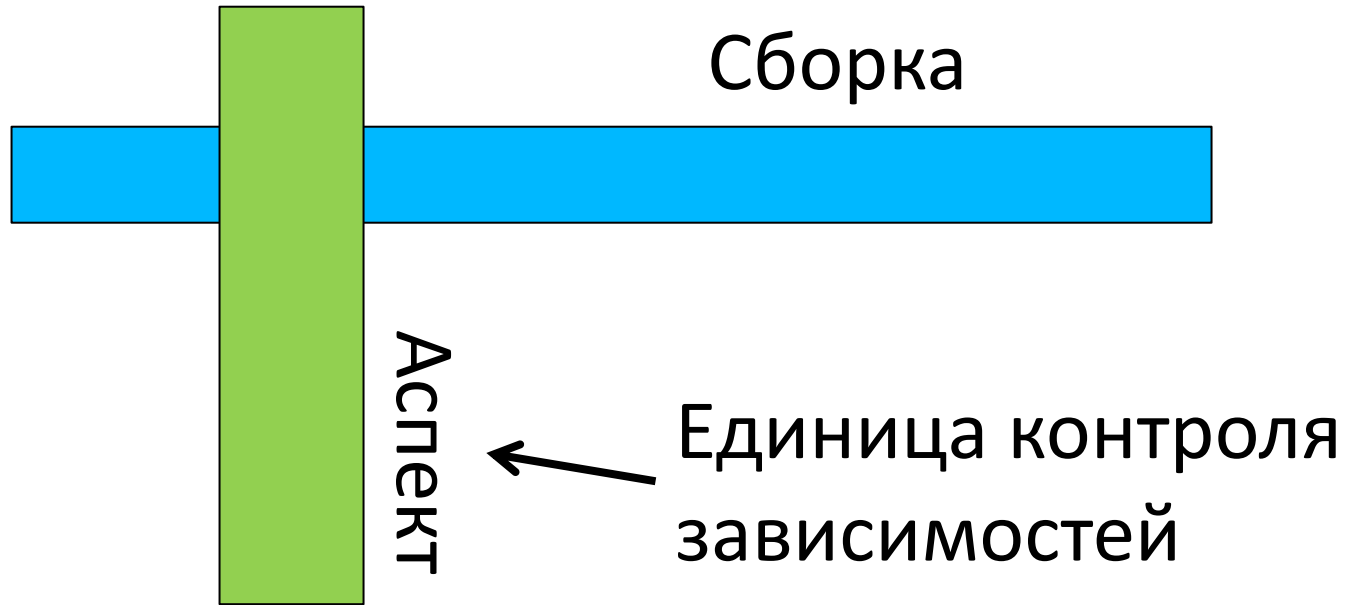
# Ромбики из сборок



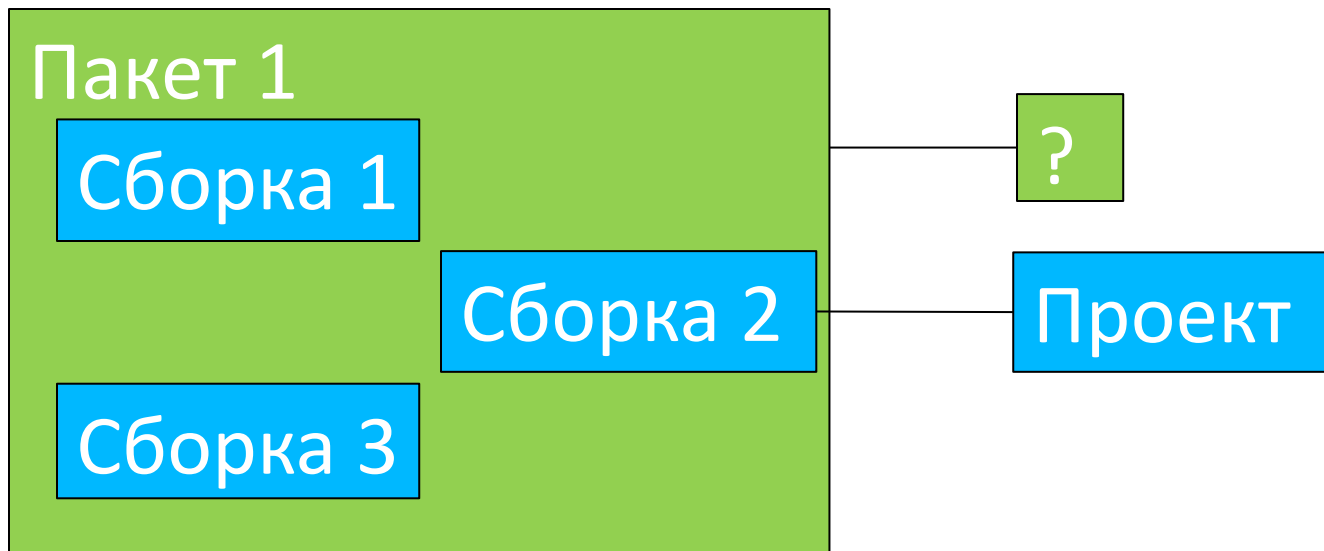
# Продукты и эволюция

- ReSharper
- dotTrace (платформа)
- dotPeek (C#, навигация, ...)
- CLI – инспекции

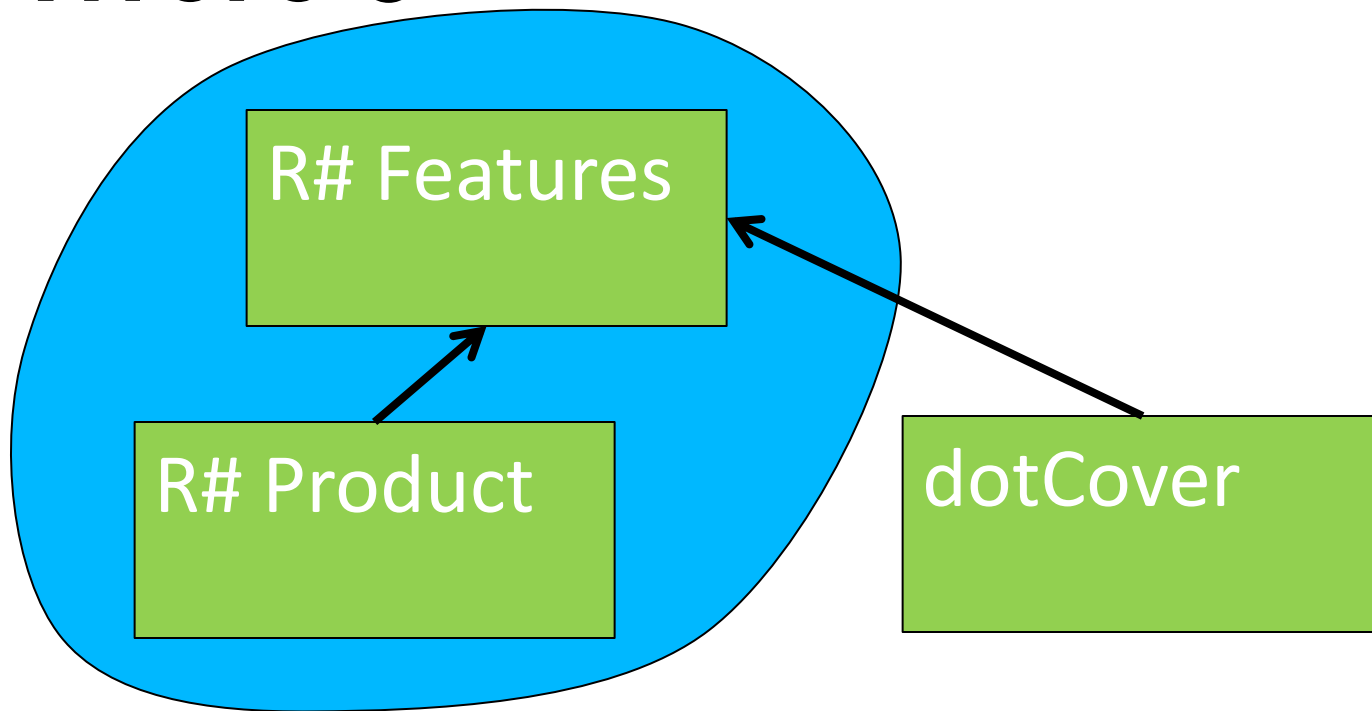
# Итого 1



# Итого 2



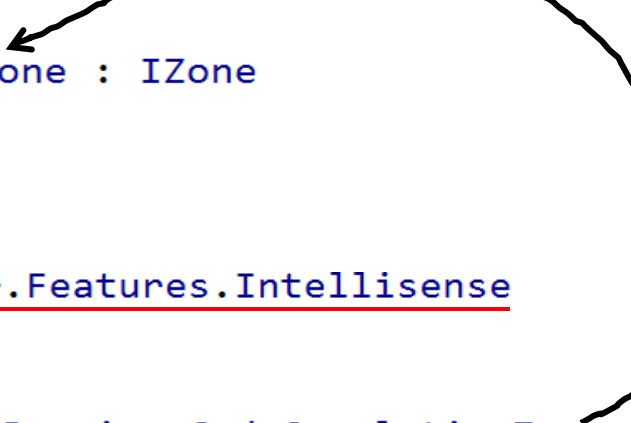
# Итого 3



# Зоны

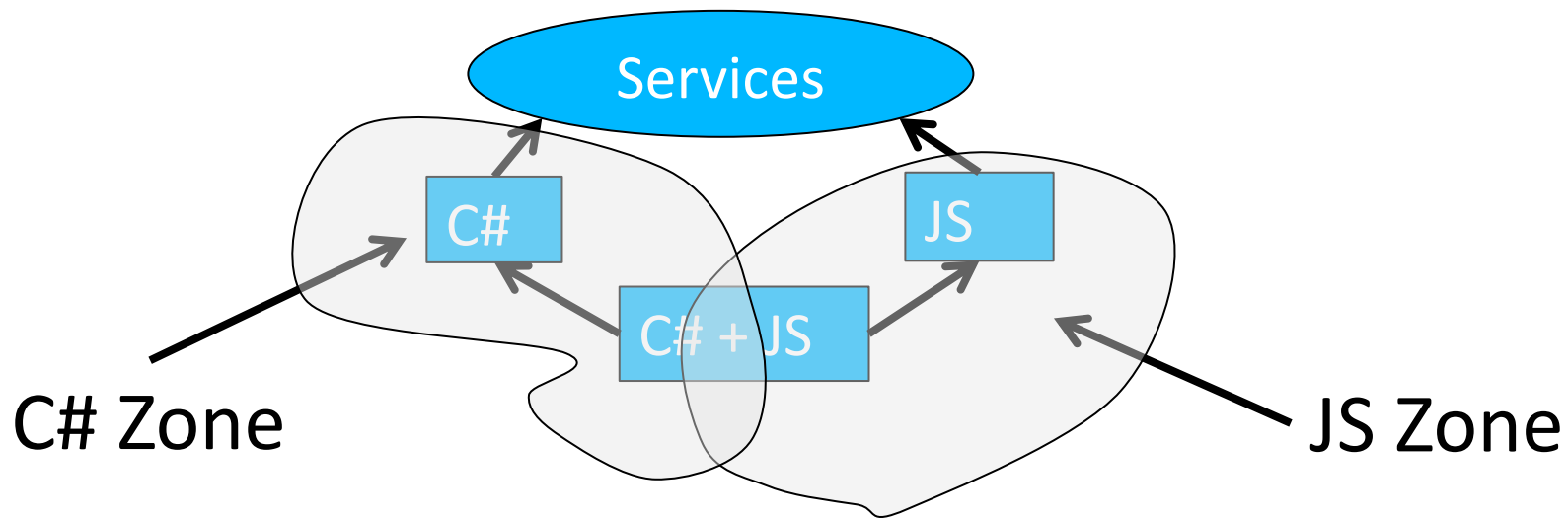
```
[ZoneDefinition]
public class CodeCompletionZone : IZone
{
}

namespace JetBrains.ReSharper.Features.Intellisense
{
    [Zone]
    public class ZoneMarker : IRequire<CodeCompletionZone>
    {
    }
}
```





# Зоны



# Зоны: запуск приложения

- Статический анализ
- Валидация после компиляции
- Продукт определяет актуальные зоны
- DI framework знает про зоны

# Почти все...

- Независимость
- Прозрачность
- Ценности
- **+ Инструмент!**



**Все, спасибо!**

**Сергей Шкредов, JetBrains**

 **serjic**

 **serjic@jetbrains.com**

**+7 (911) 9109032**