



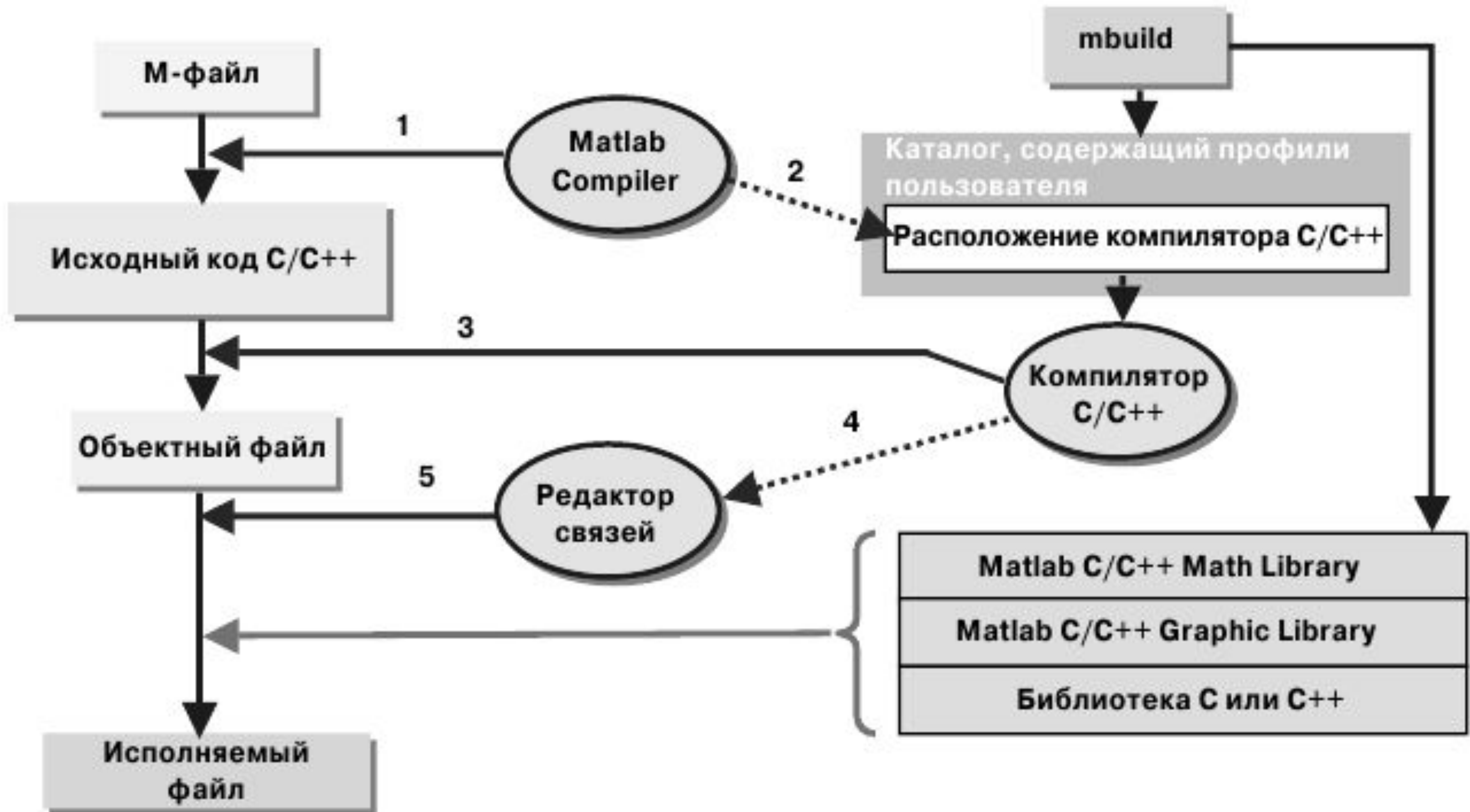
# Лекция № 4

## «MATLAB. C#.NET»

Ведущий преподаватель: канд. техн. наук, доцент кафедры ИУТС Альчаков Василий Викторович

# 2 Этапы создания приложения

## Основные этапы



# 3

## Этапы создания приложения

### Основные команды

```
>> mbuild -setup
```

```
Welcome to mbuild -setup. This utility will help you set up  
a default compiler. For a list of supported compilers, see  
http://www.mathworks.com/support/compilers/R2013a/win64.html
```

```
Please choose your compiler for building shared libraries or COM components:
```

```
Would you like mbuild to locate installed compilers [y]/n?
```

# 4

# Этапы создания приложения

## Основные команды

```
Select a compiler:
[1] Microsoft Visual C++ 2012 in C:\Program Files (x86)\Microsoft Visual Studio 11.0

[0] None

Compiler: 1

Please verify your choices:

Compiler: Microsoft Visual C++ 2012
Location: C:\Program Files (x86)\Microsoft Visual Studio 11.0

Are these correct [y]/n? y
```



# 5 Этапы создания приложения

## Основные команды

```
*****
```

```
Warning: Applications/components generated using Microsoft Visual C++  
2012 require that the Microsoft Visual Studio 2012 run-time  
libraries be available on the computer used for deployment.  
To redistribute your applications/components, be sure that the  
deployment machine has these run-time libraries.
```

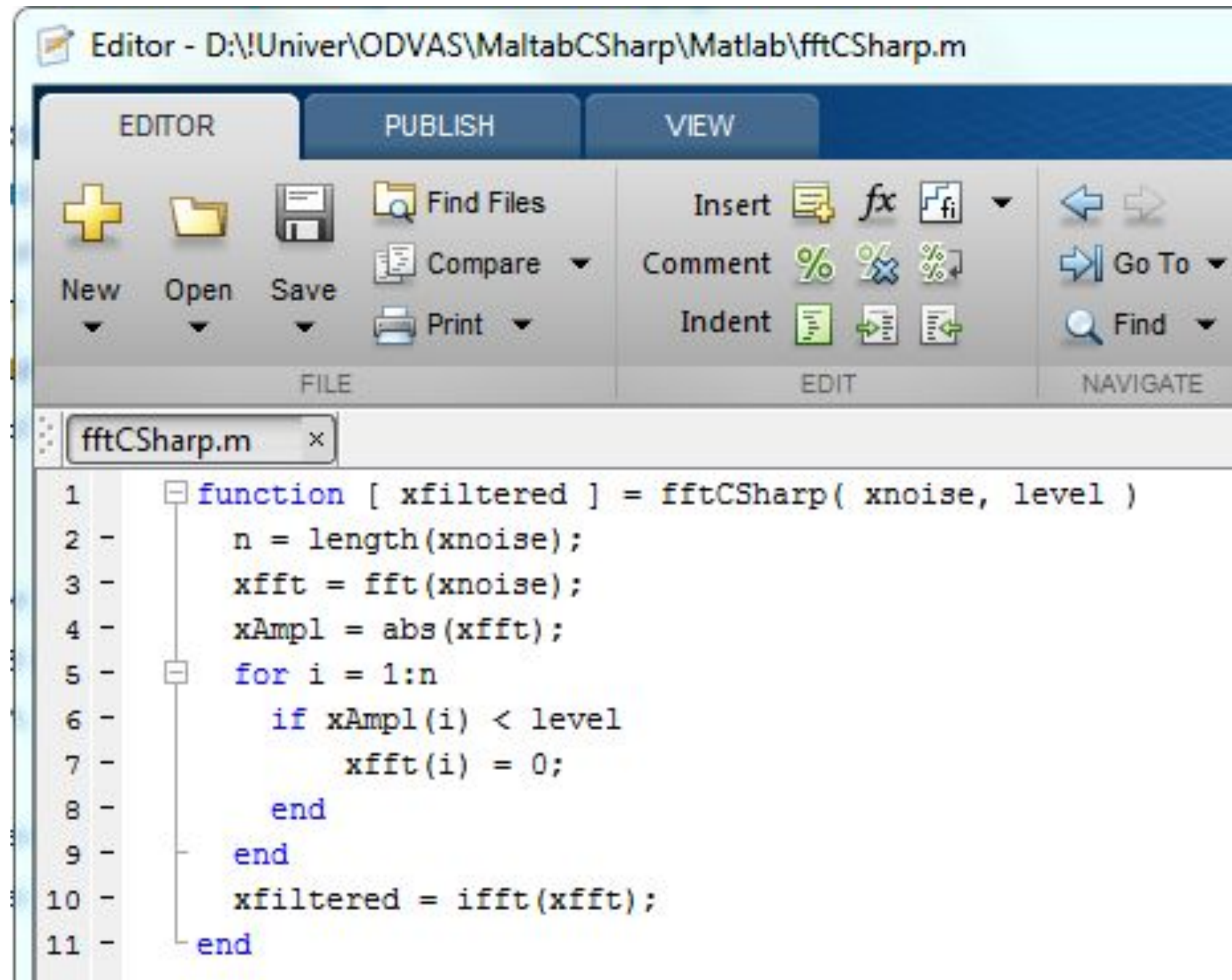
```
*****
```

```
Trying to update options file: C:\Users\VA\AppData\Roaming\MathWorks\MATLAB\R2013a\compopts.bat  
From template: C:\PROGRA~1\MATLAB\R2013a\bin\win64\mbuildopts\msvc110compp.bat
```

```
Done . . .
```

# 6 Этапы создания приложения

## Подготовка m-файла



The screenshot shows the MATLAB Editor interface with the file 'fftCSharp.m' open. The editor has three tabs: EDITOR, PUBLISH, and VIEW. The EDITOR tab is active, showing a ribbon with icons for New, Open, Save, Find Files, Compare, Print, Insert, Comment, Indent, Go To, and Find. The code in the editor is as follows:

```
1 function [ xfiltered ] = fftCSharp( xnoise, level )
2 -     n = length(xnoise);
3 -     xfft = fft(xnoise);
4 -     xAmp1 = abs(xfft);
5 -     for i = 1:n
6 -         if xAmp1(i) < level
7 -             xfft(i) = 0;
8 -         end
9 -     end
10 -     xfiltered = ifft(xfft);
11 - end
```

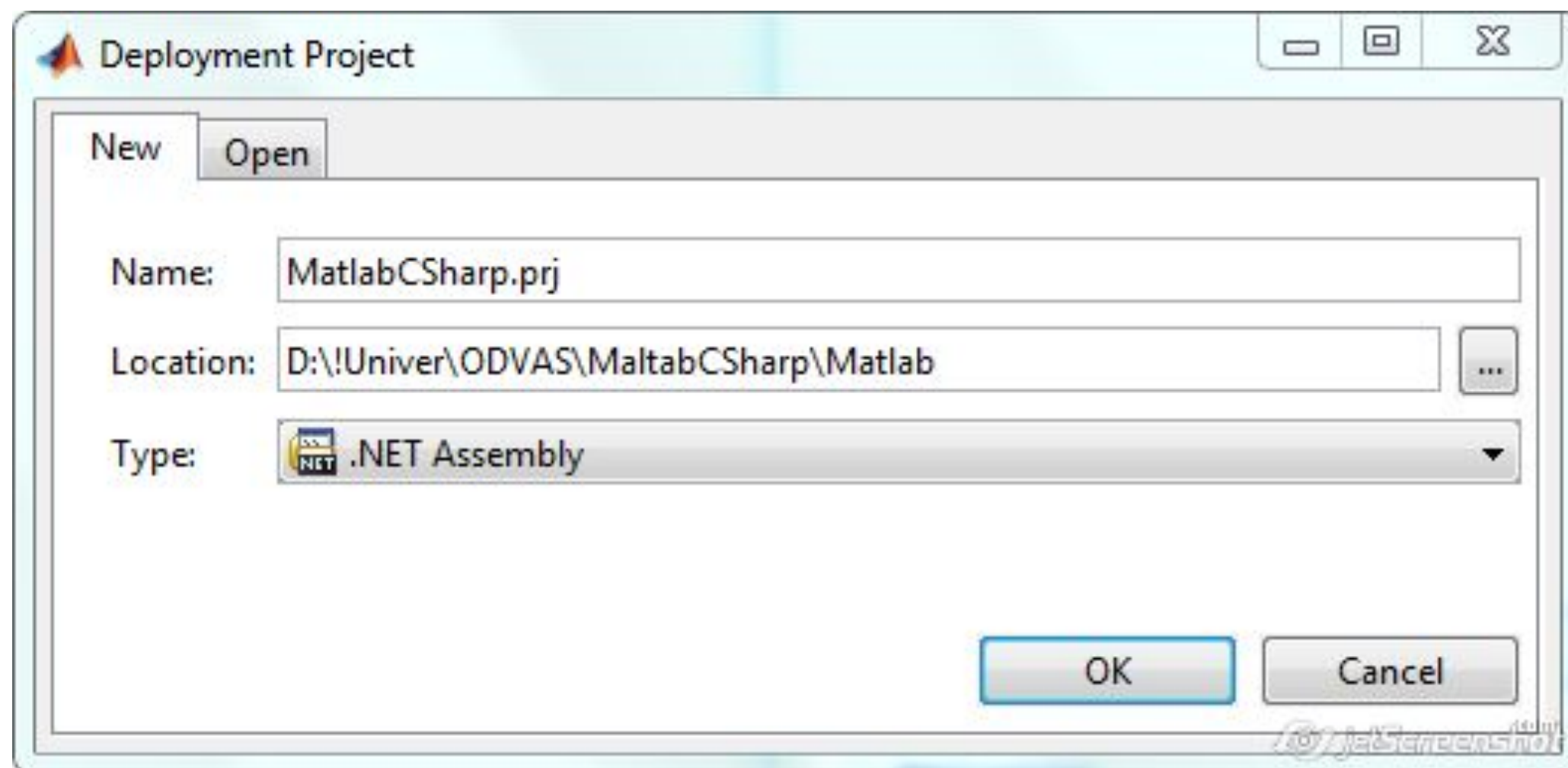
# 7 Этапы создания приложения

## Запуск deploy проекта

Чтобы собрать dll-библиотеку MATLAB'a для интеграции в C#.NET, нужно настроить линкер, т.е. какой средой мы будем собирать проект. Для начала нужно установить среду выполнения MCR (MATLAB Component Runtime). Это набор dll-библиотек для полной поддержки языка MATLAB. Установочный файл можно найти: c:\Program Files\MATLAB\R2013a\toolbox\compiler\deploy\win64\

<http://www.mathworks.com/products/compiler/mcr/index.html>

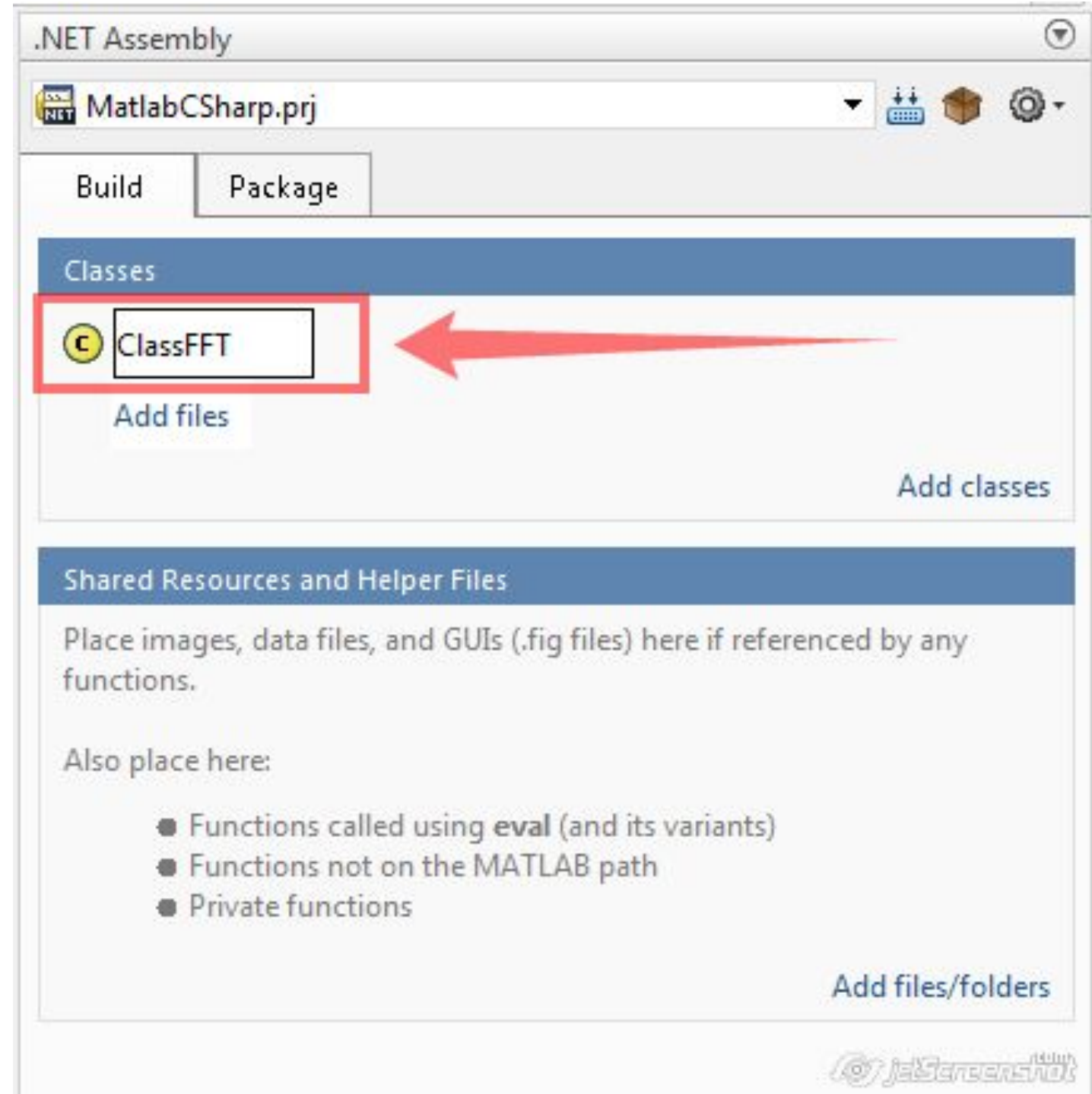
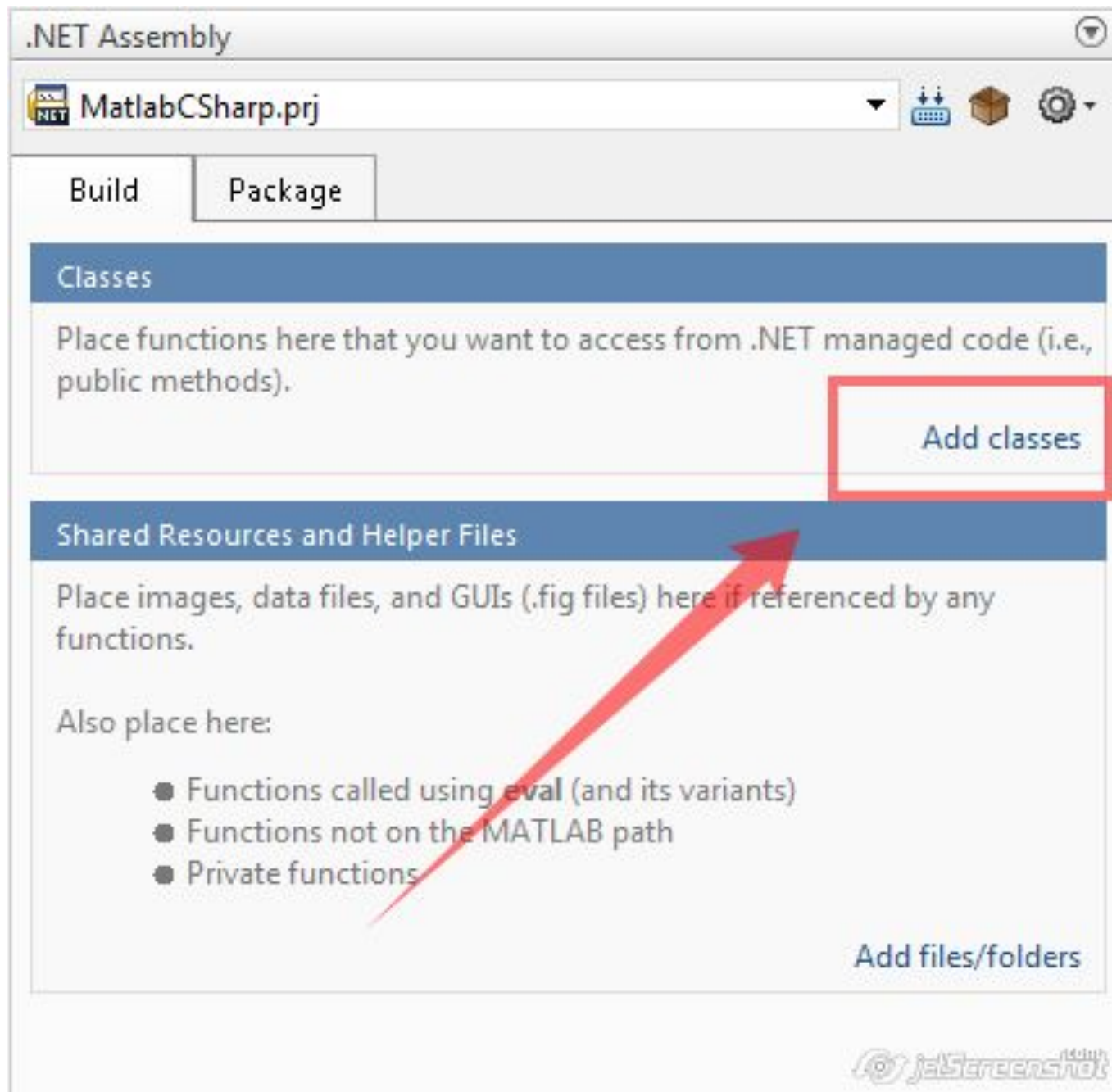
```
>> deploytool
```





# 8 Этапы создания приложения

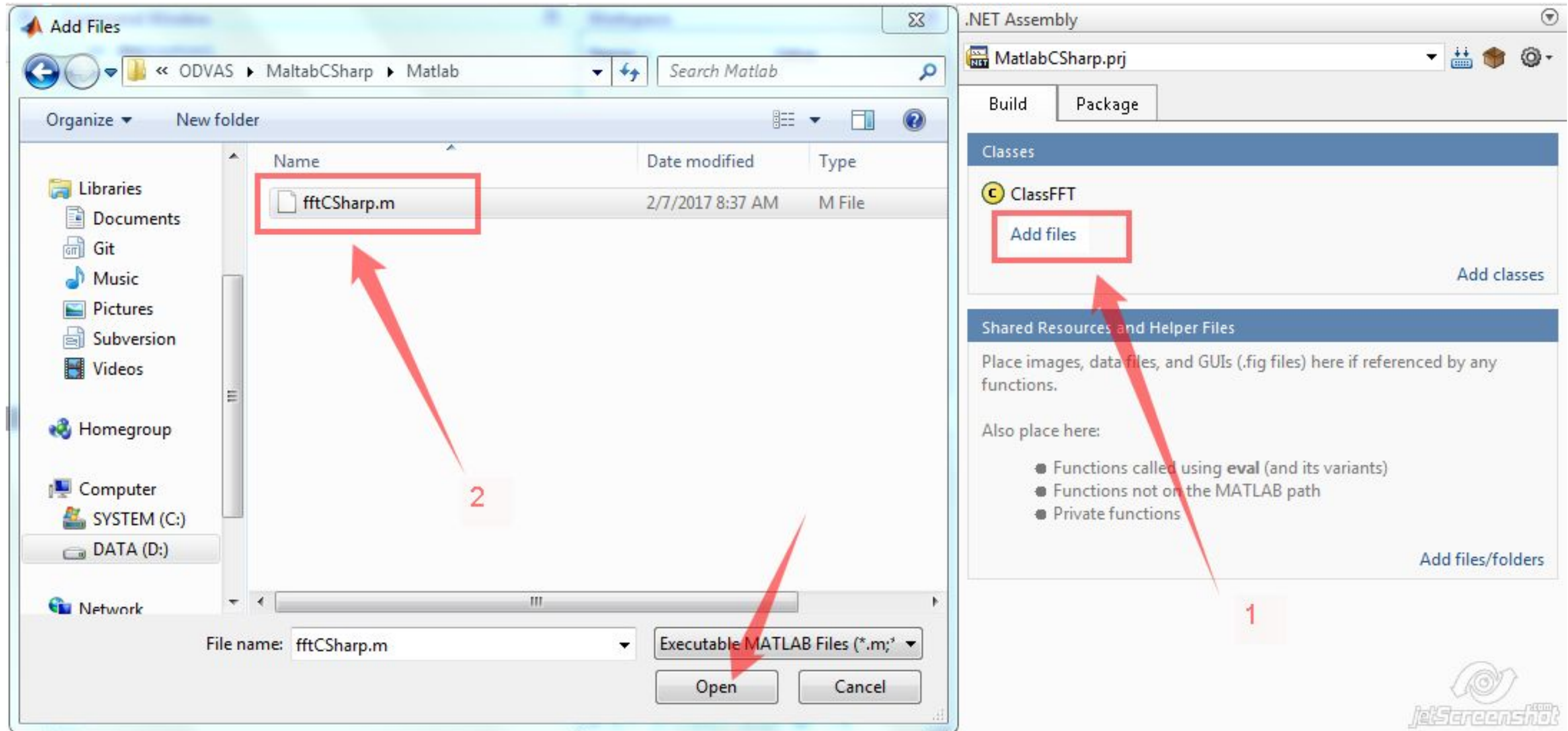
## Deployment Project





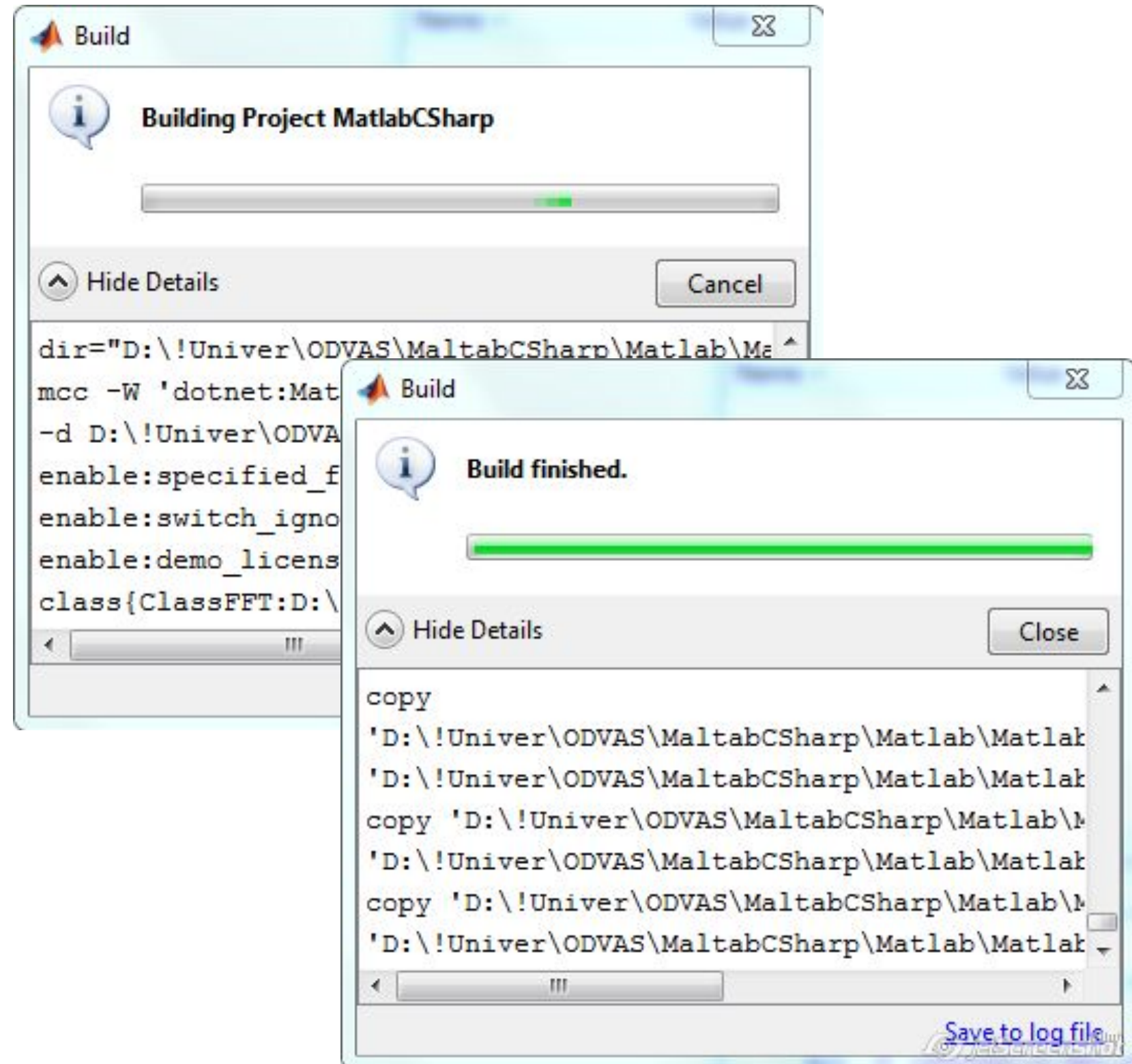
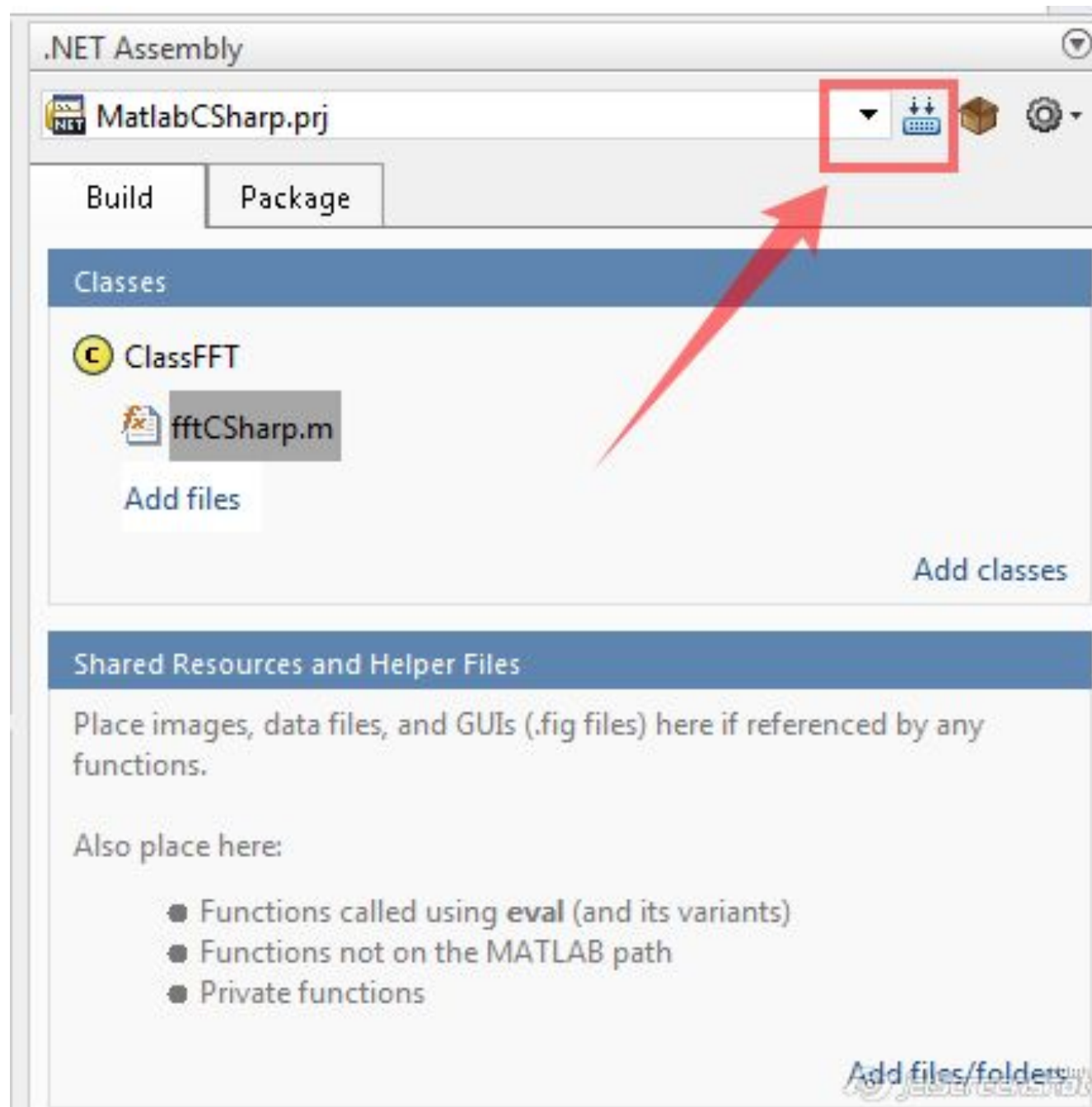
# 9 Этапы создания приложения

## Deployment Project






# 10 Этапы создания приложения

## Deployment Project



# 11 Этапы создания приложения

## Deployment Project





Name	Date modified	Type	Size
 MatlabCSharp	2/7/2017 8:50 AM	File folder	
 fftCSharp.m	2/7/2017 8:37 AM	M File	1 KB
 MatlabCSharp.prj	2/7/2017 8:52 AM	PRJ File	59 KB










Name	Date modified	Type	Size
 distrib	2/7/2017 8:52 AM	File folder	
 src	2/7/2017 8:52 AM	File folder	



# 12 Этапы создания приложения

## Определение версии библиотеки

Name	Date modified	Type	Size
 MatlabCSharp.dll	2/7/2017 8:52 AM	Application extens...	44 KB
 MatlabCSharp_overview	2/7/2017 8:51 AM	Chrome HTML Do...	30 KB
 MatlabCSharpNative.dll	2/7/2017 8:52 AM	Application extens...	44 KB
 readme	2/7/2017 8:51 AM	Text Document	4 KB

Name	Date modified	Type	Size
 ClassFFT.cs	2/7/2017 8:51 AM	Visual C# Source f...	9 KB
 ClassFFTNative.cs	2/7/2017 8:51 AM	Visual C# Source f...	9 KB
 MatlabCSharp.dll	2/7/2017 8:52 AM	Application extens...	44 KB
 MatlabCSharp	2/7/2017 8:52 AM	XML Document	7 KB
 MatlabCSharp_overview	2/7/2017 8:51 AM	Chrome HTML Do...	30 KB
 MatlabCSharpNative.dll	2/7/2017 8:52 AM	Application extens...	44 KB
 MatlabCSharpNative	2/7/2017 8:52 AM	XML Document	7 KB
 mccExcludedFiles	2/7/2017 8:51 AM	Text Document	659 KB
 readme	2/7/2017 8:51 AM	Text Document	4 KB

## Создание проекта

The screenshot shows the Microsoft Visual Studio interface. The 'FILE' menu is open, and the 'New' option is selected. The 'New Project' dialog box is displayed, showing the 'Visual C#' templates. The 'Console Application' template is selected. The project name is 'MatlabCSharpConsole' and the location is 'D:\Univer\ODVAS\MaltabCSharp\Project\'. The 'Create directory for solution' checkbox is checked.

**Visual C# Templates:**

Template Name	Type
Console Application	Visual C#
Class Library	Visual C#
Portable Class Library	Visual C#
WPF Browser Application	Visual C#
Empty Project	Visual C#
Windows Service	Visual C#
WPF Custom Control Library	Visual C#
WPF User Control Library	Visual C#
Windows Forms Control Library	Visual C#

**Project Details:**

- Name: MatlabCSharpConsole
- Location: D:\Univer\ODVAS\MaltabCSharp\Project\
- Solution name: MatlabCSharpConsole

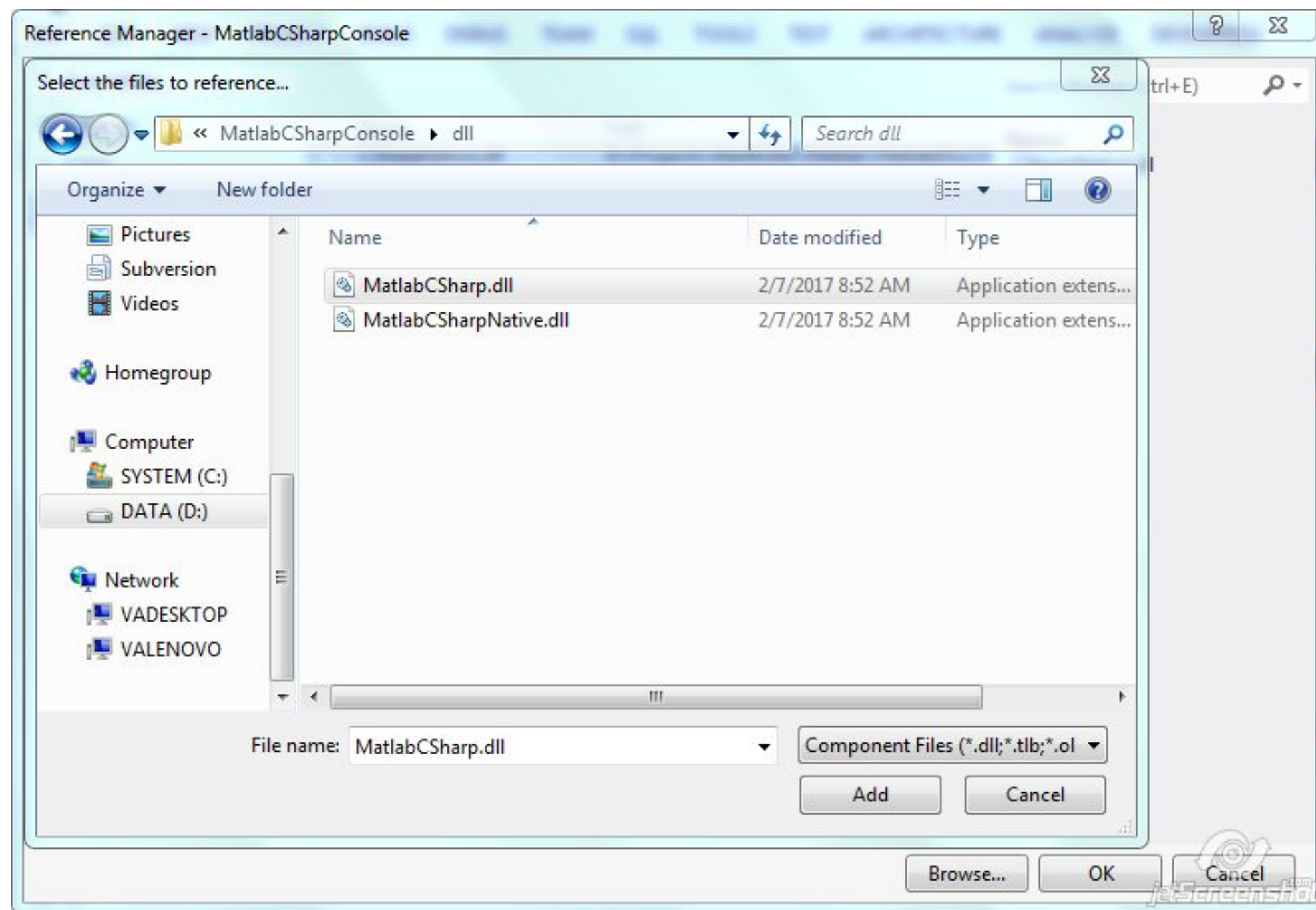
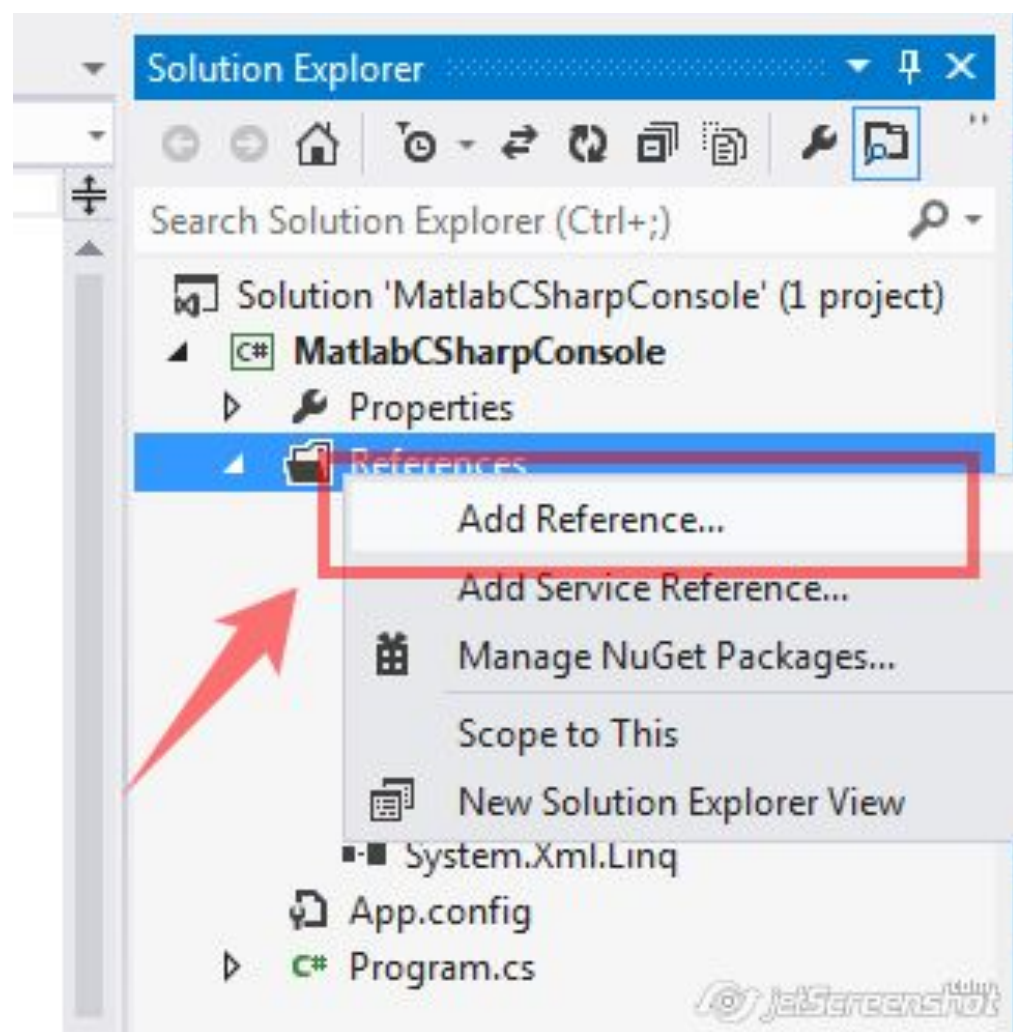
**Options:**

- Create directory for solution
- Add to source control



# 14 Консольное приложение C#

## Создание проекта



c:\Program Files\MATLAB\R2013a\toolbox\dotnetbuilder\bin\win64\v2.0\



# 15 Консольное приложение C#

## Создание проекта

Перед использованием методов проекта, необходимо добавить ссылки на скомпилированную библиотеку MATLABplane.dll и на библиотеку MWArray.dll, найти ее можно по адресу

c:\Program Files\MATLAB\R2013a\toolbox\dotnetbuilder\bin\win64\v2.0\

The screenshot displays the Microsoft Visual Studio IDE. On the left, the Solution Explorer shows a project named 'MatlabCSharpConsole' with a 'References' folder containing 'MatlabCSharp', 'Microsoft.CSharp', and 'MWArray' highlighted in a red box. The main editor window shows the code for 'Program.cs' with the following content:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace MatlabCSharpConsole
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Console.WriteLine("Hello world!");
14            Console.ReadKey();
15        }
16    }
17 }
18
```

On the right, a console window titled 'file:///D:/Univer/ODVAS/MatlabCSharp/Project/MatlabCSharpConsole/MatlabCSharp...' displays the output 'Hello world!'. The status bar at the bottom indicates 'Ready', 'Ln 6', 'Col 1', and 'Ch 1'.

## Создание проекта

Подключаем библиотеки

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 using MathWorks.MATLAB.NET.Utility;
8 using MathWorks.MATLAB.NET.Arrays;
9 using MatlabCSharp;
10
```

```
Random random = new Random();
noise = (random.NextDouble() - 0.5)/10;
```

Код для генерации сигнала

```
int i, k;
int N = 1024; // Counts number
double T = 0.0001; // Sample time
int A = 1; // Amplitude
int F0 = 100; // Base Freq
int dF = 50; // Freq Increment
int Fcount = 5; // Freq count

// Freq vector generator
double[] f = new double[Fcount];
for (i = 0; i < Fcount; i++)
    f[i] = F0 + dF * i;

// Polyharmonic generator
double[] x = new double[N];
double t = 0;
double sinSum = 0d;
for (i = 0; i < N; i++)
{
    sinSum = 0d;
    for (k = 0; k < Fcount; k++ )
        sinSum += Math.Sin(2 * Math.PI * f[k] * t);
    // Bringing to a predetermined amplitude
    sinSum = sinSum * A / Fcount;
    x[i] = sinSum;
    t += T;
}
```



# 17 Консольное приложение C#

## Подключаем класс C#

```
ClassFFT fft = new ClassFFT();  
fft.fftCSharp(  

```

▲ 5 of 7 ▼ MWArray ClassFFT.fftCSharp(MWArray xnoise, MWArray level)

48  
49  
50 MWNumericArray level = new MWNumericArray(10);  
51  
52  
53  
54  
55  
56  
57  
58  
59

**! TypeInitializationException was unhandled**

The type initializer for 'MathWorks.MATLAB.NET.Arrays.MWNumericArray' threw an exception.

**Troubleshooting tips:**

[Get general help for this exception.](#)



# 18 Консольное приложение C#

## Меняем настройки проекта

The screenshot displays the Microsoft Visual Studio interface for a C# console application named 'MatlabCSharpConsole'. The 'Build' tab is active in the Properties window, showing the following settings:

- Configuration: Active (Debug)
- Platform: Active (Any CPU)
- General:
  - Conditional compilation symbols: (empty)
  - Define DEBUG constant
  - Define TRACE constant
  - Platform target: x64
  - Prefer 32-bit
  - Allow unsafe code
  - Optimize code
- Errors and warnings:
  - Warning level: 4
  - Suppress warnings: (empty)
  - Treat warnings as errors: None

A context menu is open over the project in the Solution Explorer, with a red arrow pointing to the 'Properties' option. The menu items include: Build, Rebuild, Clean, Publish..., Run Code Analysis, Scope to This, New Solution Explorer View, Show on Code Map, Calculate Code Metrics, Add DevExpress Item, Add, Add Reference..., Add Service Reference..., Manage NuGet Packages..., View Class Diagram, Set as StartUp Project, Debug, Add Solution to Source Control..., Cut (Ctrl+X), Paste (Ctrl+V), Remove (Del), Rename, Unload Project, Open Folder in File Explorer, and Properties.

# 19 Консольное приложение C#

## Вызов метода из dll

```
MWNumericArray level = new MWNumericArray(10);  
MWNumericArray xnoise = new MWNumericArray(x);  
MWArray xfiltered;  
ClassFFT fft = new ClassFFT();  
xfiltered = fft.fftCSharp(xnoise, level);  
double[,] nativeX = (double[,])xfiltered.ToArray();
```



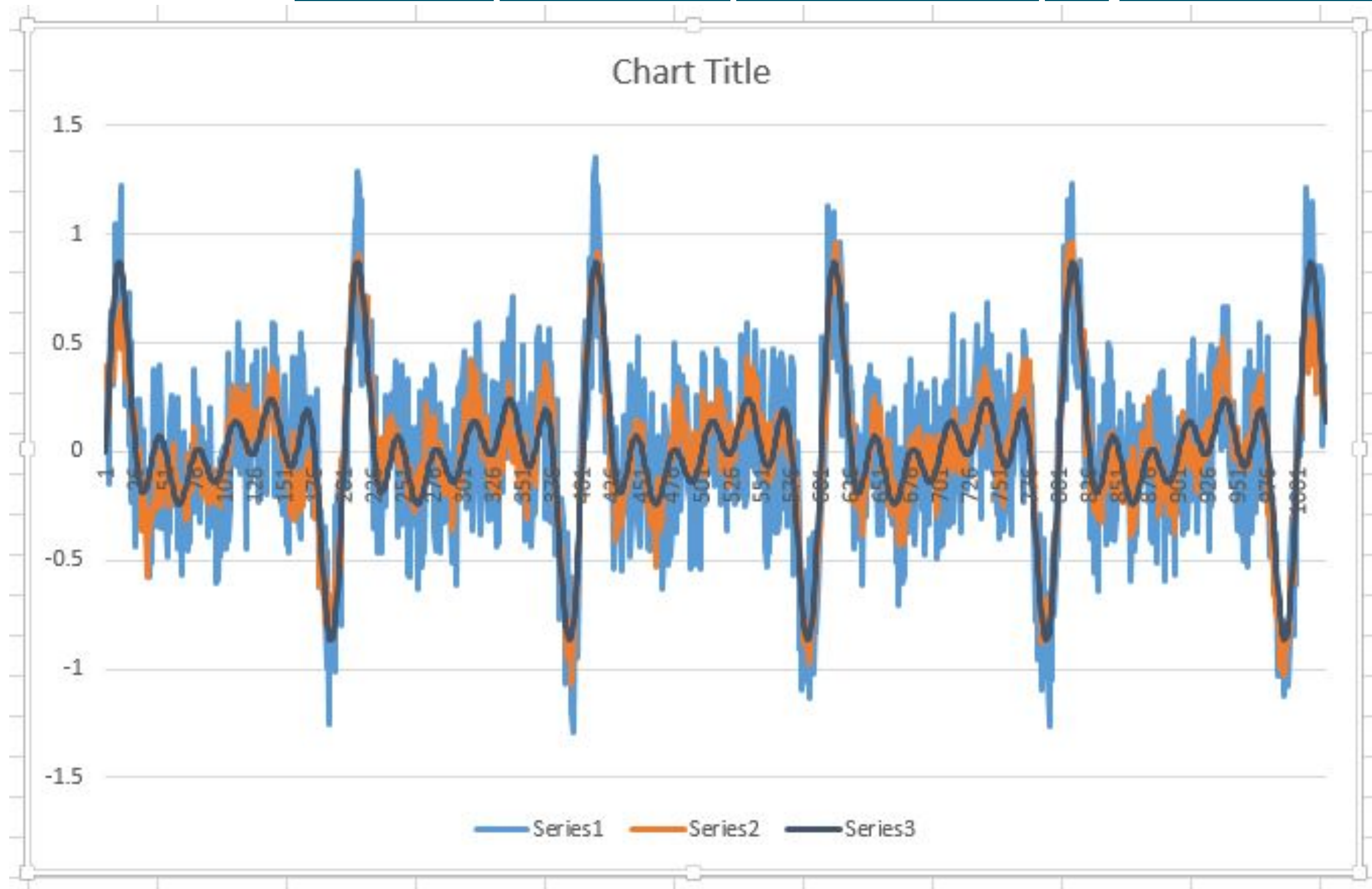
# 20 Консольное приложение C#

## Сохранение результатов в файл

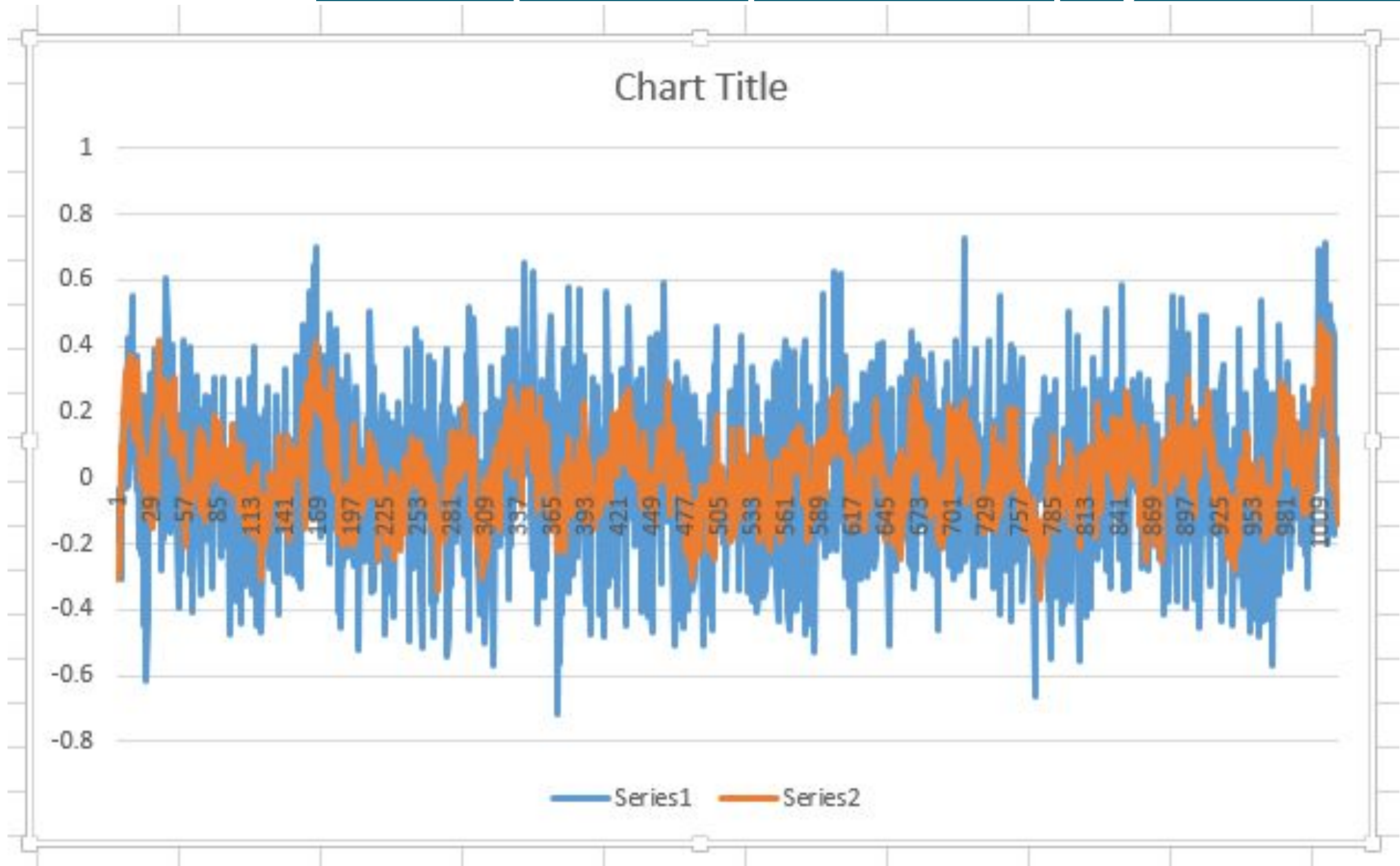
```
// Save to file
System.IO.StreamWriter file = new System.IO.StreamWriter("results.csv");
string line;
double eps1, eps2;
for (i = 0; i < N; i++)
{
    eps1 = xn[i] - nativeX[0, i];
    eps2 = x[i] - nativeX[0, i];
    line = string.Format("{0},{1},{2},{3},{4}", xn[i], nativeX[0, i], x[i], eps1, eps2);
    file.WriteLine(line);
}
file.Close();
```



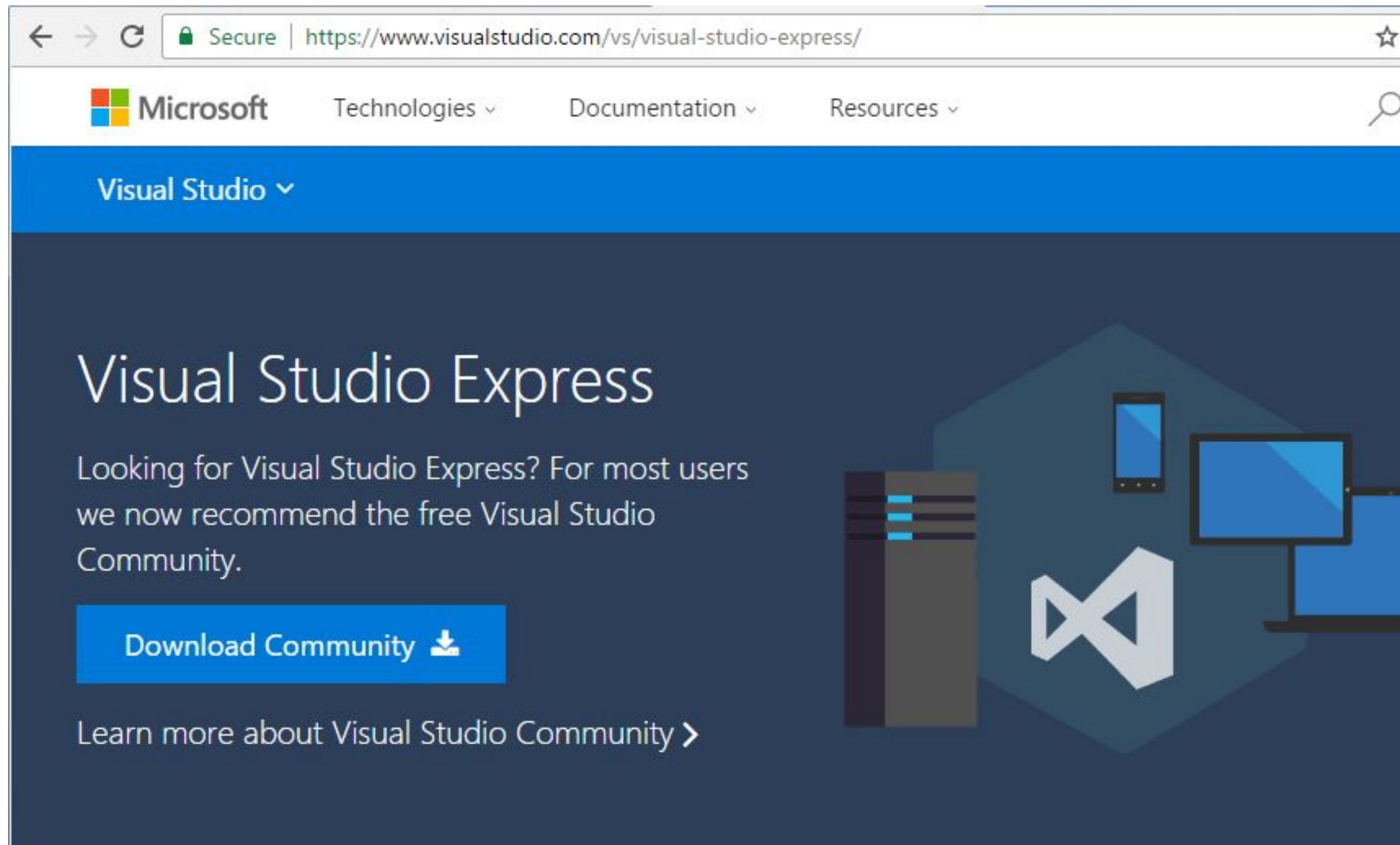
## Сохранение в файл и анализ результата



## Сохранение в файл и анализ результата



## Инструменты



<https://www.visualstudio.com/vs/visual-studio-express/>

<https://blogs.msdn.microsoft.com/microsoftimagine/2015/07/20/visual-studio-2015-is-here-for-students/>

<https://www.visualstudio.com/ru/downloads/>



## Ссылки

<https://habrahabr.ru/post/132487/>

<https://habrahabr.ru/sandbox/30128/>

<https://habrahabr.ru/post/199142/>

### Идея

Идея была проста. Вместо того, чтобы платить зарплату одному умному программисту (который бы умел писать хороший код на C# и разбирался в тонкостях нашего «матана», а такого непросто найти), можно взять двух «полуумных». Первый будет писать прототипы на MATLAB, второй — переносить решение на C#.

MATLAB в таком случае преподносился как инструмент для документирования алгоритма. У такого решения были озвучены следующие преимущества:

- MATLAB — простой язык, который опытный программист может выучить за 1 день
- Все математики пишут на MATLAB

Стоит отметить, что ранее для документирования алгоритма использовался Mathcad, UML-диаграммы и даже просто лист бумаги.

С идеей спорили, ее не принимали. Но в итоге Главный Алгоритмист всех убедил, и была создана команда алгоритмистов, пишущих на матлабе. В эту команду вошел и я. В нашу команду также вошел Один Программист из числа разработчиков, который должен был бы, не думая, переносить наши матлаб-экзерсисы на C#.

Замечу, что кроме предложений отказаться от этой затеи, было и предложение использовать, по крайней мере, python+numpy. Хотя бы той причине, что сколько-нибудь значительного опыта разработки на MATLAB ни у кого не было (я раньше писал на python и R, остальные были математики и инженеры, использующие MATLAB как продвинутый калькулятор). Предложения, как можно понять, отклонили.

*...Идея была проста. Вместо того, чтобы платить зарплату одному умному программисту (который бы умел писать хороший код на C# и разбирался в тонкостях нашего «матана», а такого непросто найти), можно взять двух «полуумных». Первый будет писать прототипы на MATLAB, второй — переносить решение на C#...*

### Прошел год...

Прошел год и сейчас уже можно подвести некоторые итоги:

- Математики не умеют писать код. А в той лапше, что выходит из-под их пера не могут разобраться и они сами. Нет ни специфических знаний, ни опыта, ни культуры программирования
- Раз уж мы отказались от схем и Mathcad, то появилось желание использовать для наших MATLAB программ реальные источники данных. В итоге мы немало времени занимались дублированием части функционала Флагманского продукта на Матлабе
- Пока мы занимались программированием, времени на работу над собственно *алгоритмом* оставалось немного
- MATLAB не подходит для разработки больших программ. По крайней мере, не подходит нам. Динамическая нестрогая типизация, медленные циклы, ооочень-медленный ООП (настолько, что пришлось частично отказаться от использования ООП), неполная поддержка разреженных матриц
- Переносить код с MATLAB на C# непросто. Еще сложнее бэкпортировать рефакторинги и оптимизации

В результате мы не можем заранее оценить, сколько времени потребуется на реализацию той или иной фичи. Неделя или может быть месяц?

### Вместо заключения

В самом начале статьи я немного слухавил, т.к. история продолжается до сих пор. Мы продолжаем писать на MATLAB, наш код продолжают переносить на C#. Хотя теперь уже все согласны, что у идеи Главного Алгоритмиста есть изъяны. Но менять уже что-либо слишком поздно.