Восьмая независимая
научно-практическая конференция
«Разработка ПО 2012»

1 - 2 ноября, Москва

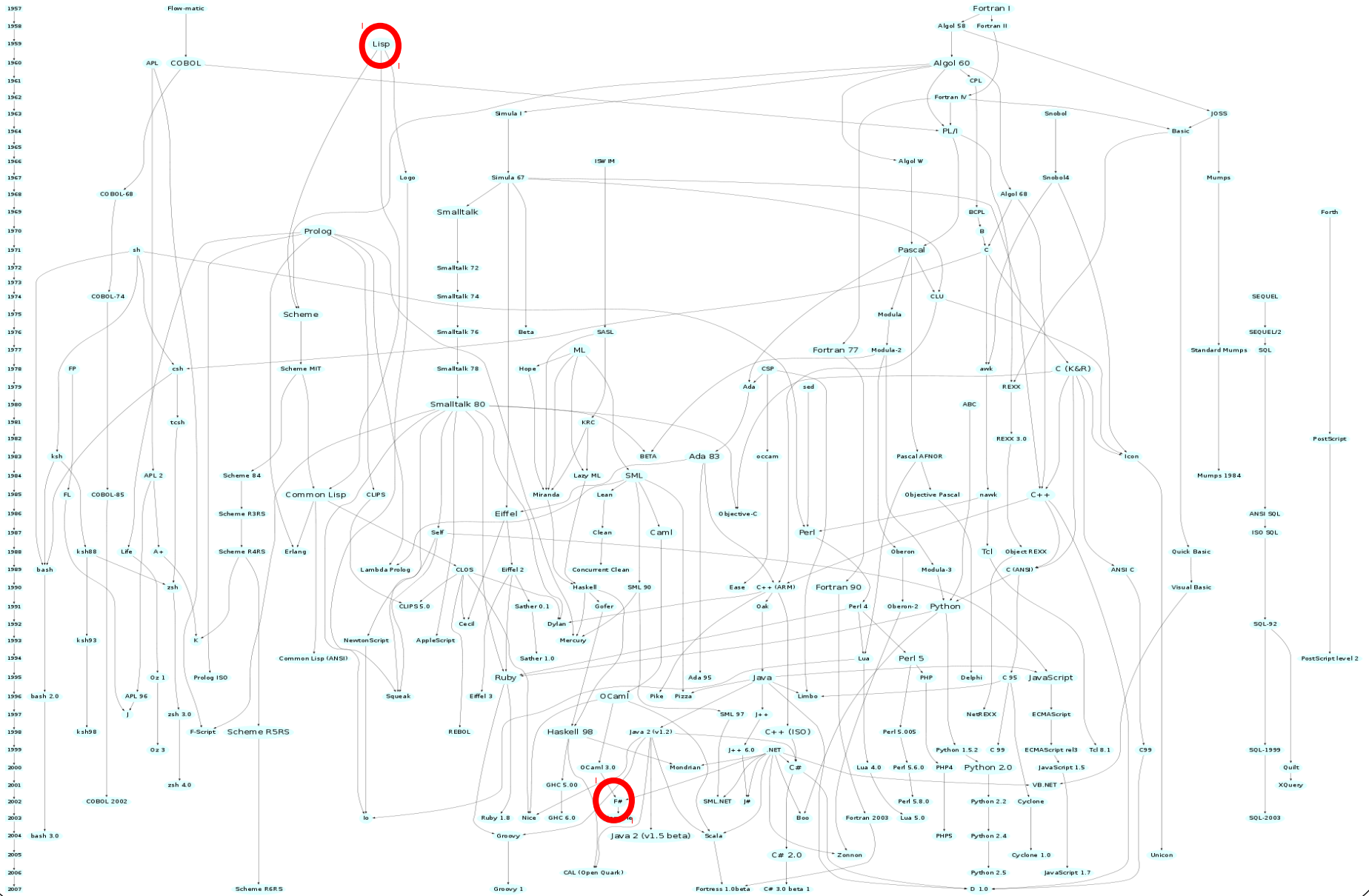
F#:
Использование функционального подхода для обработки данных

Дмитрий Сошников

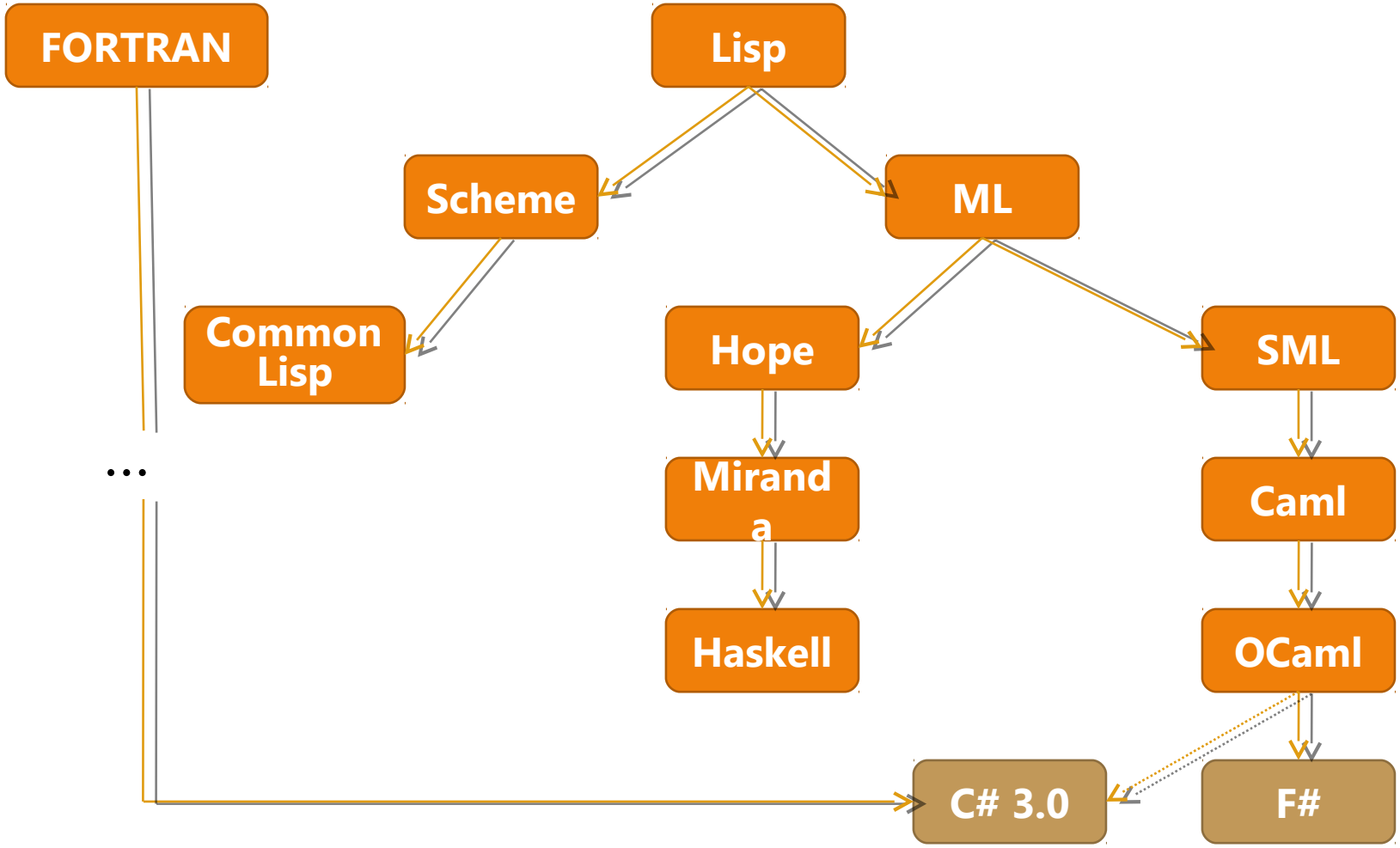
Майкрософт | НИУ ВШЭ | МАИ

F#x3

Немного истории



Немного истории



Зачем?

- John Hughes, Why Functional Programming Matters, 1984

Привкус функциональщины (F#)

Посчитать сумму квадратов от 1 до 10

Глупо

```
1*1+2*2+3*3+4*4+5*5+6*6+7*7+8*8+9*9+10*10
```

Лаконично

```
let rec sumsq a b =
  if a>=b then a*a
  else a*a+sumsq (a+1) b

sumsq 1 10
```

Обобщённо

```
let rec num_map_reduce fm fr a b =
  if a>=b then fm a
  else fr (fm a) (num_map_reduce fm fr (a+1) b)

let sumsq = num_map_reduce (fun x->x*x) (+)
```

Эффективно

```
let num_map_reduce fm fr a b =
  let rec num_map_reduce' acc a b =
    if a>b then acc
    else num_map_reduce' (fr acc (fm a)) (a+1) b
  num_map_reduce' (fm a) (a+1) b
```

Во всех случаях – без побочных эффектов

Обработка данных: списки

map

```
[1..10] |> List.map (fun x->x*x)
```

reduce

fold

```
[1..10] |> List.map (fun x->x*x) |> List.reduce (+)

[1..10] |> List.map (fun x->x*x) |> List.fold (+) 0
```

filter

```
[1..10] |> List.filter (fun x-> x%3=0)
```

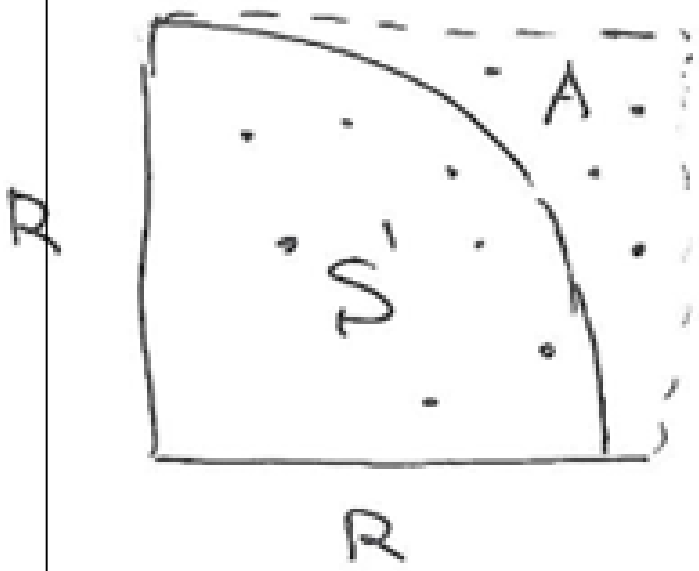
Бесконечные последовательности

```
let rand max n =
   seq {
     let r = new System.Random(n)
     while true do yield r.NextDouble()*max
   }

let monte_carlo hit max iters =
  let hits = (float)(
      Seq.zip (rand max 1) (rand max 3) |>
      Seq.take iters |>
      Seq.filter hit |>
      Seq.length) in
  4.0*max*max*hits/((float)iters)

let area radius =
 monte_carlo
   (fun (x,y) -> x*x+y*y<=radius*radius)
   radius 10000

let Pi = area 1.0
```



Практический пример: частот.словарь

```
let ReadLines fn =
  seq { use inp = File.OpenText fn in
            while not(inp.EndOfStream) do
               yield (inp.ReadLine()) }
```

```
let FreqDict word_seq =
   word_seq |>
   Seq.fold(
     fun (ht:Map<string,int>) v ->
      if v="" then ht else
       if Map.containsKey v ht then Map.add v ((Map.find v ht)+1) ht
       else Map.add v 1 ht)
     (Map.empty)
```

1. F# - исключительно удобный язык для обработки данных

- Лаконичный (вывод типов)
- Без побочных эффектов
- Интерактивный процесс разработки
- Мощный набор абстракций

Параллельная и асинхронная обработка

```
let task1 = async { return 10+10 }
let task2 = async { return 20+20 }
Async.Run (Async.Parallel [ task1; task2 ])
```

```
let map' func items =
    let tasks =
        seq {
            for i in items -> async { return (func i) } }
    Async.RunSynchronously (Async.Parallel tasks)
```

```
List.map (fun x -> fib(x)) [1..30];;
map' (fun x -> fib(x)) [1..30];;
```

Частотный словарь 2.0

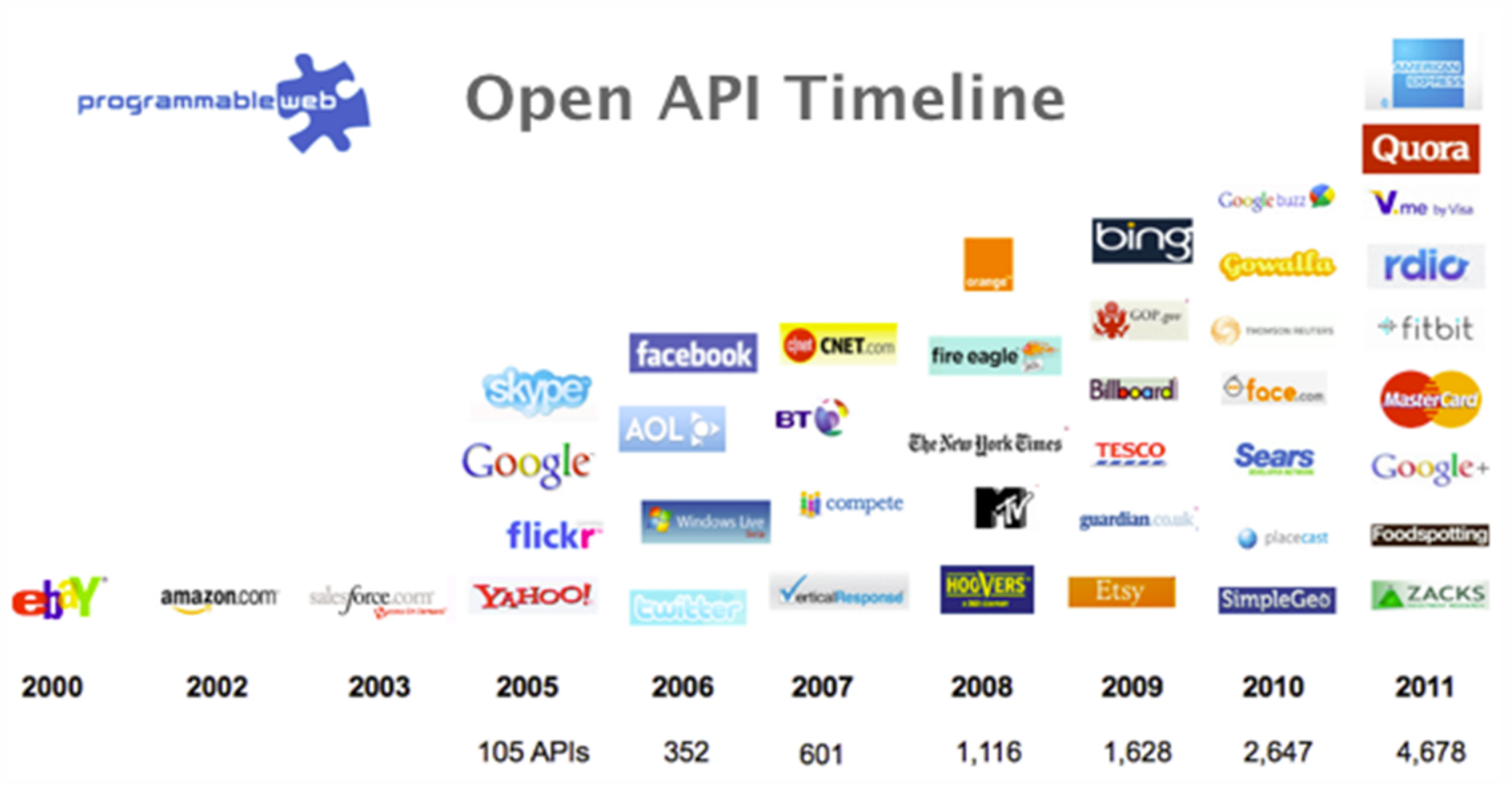
```
let ProcessFileAsync f =
  async {
    let! str = ReadFileAsync f
    let dict = FreqWords str
    do! WriteDictAsync dict (f+".res")
  }
```

1. F# - исключительно удобный язык для обработки данных

2. F# позволяет легко распараллеливать и асинхронить вычисления

- Нет состояния => легко параллелить
- Asynch workflows – мощнее, чем в C#5.0

Проблема…


programmableweb
Open API Timeline
2000
2002
2003
2005
2006
2007
2008
2009
2010
2011
105 APIs
352
601
1,116
1,628
2,647
4,678

F# 3.0: Type Providers

- Type Provider – это:
  - design-time-компонент, предоставляющий вычисляемое пространство типов/классов и методов
  - расширение компилятора / IDE
  - адаптер между источниками данных и языками .NET
- Обеспечивает:
  - Построение типов схемы «на лету» для статической типизации
  - Автоматическое обновление схемы на этапе сборки

TypeProviders

Встроенные

- SqlDataConnection
- SqlEntityConnection
- ODataService
- WsdlService

Сторонние

- FreeBase
- Regex
- FileSystem
- Csv
- Excel
- JSON
- XML
- Registry
- XAML
- AppSettings
- …

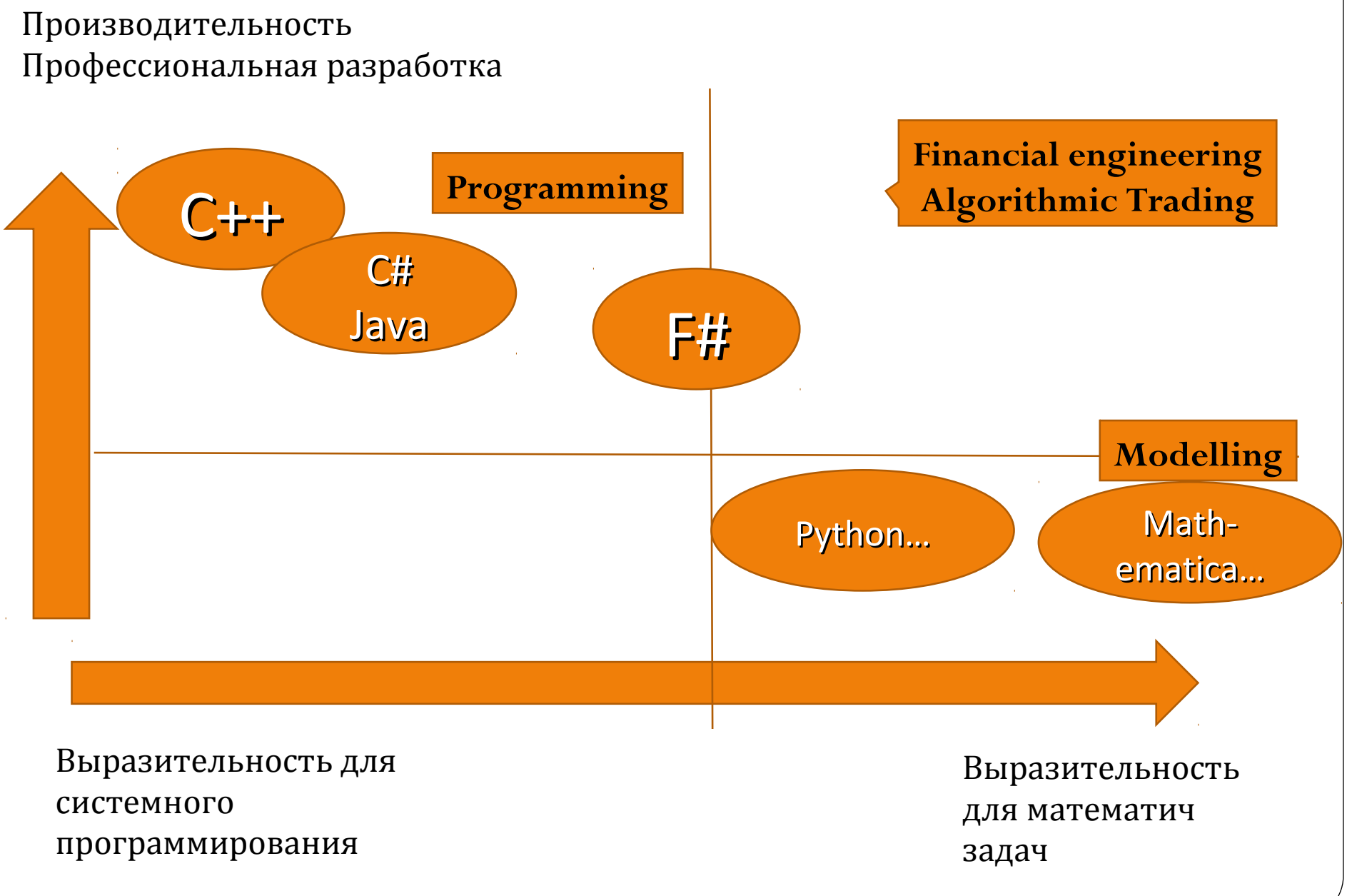
А как же облако?

- Встроенная поддержка Windows Azure Compute Instance для F#
- Поддержка функционального программирования гибких веб-интерфейсов: WebSharper
- Вычисления в облаке – проект Cloud Numerics F# Extensions
- В перспективе – подход, основанный на DSL/Computational Workflows:
  - let result = cloud { …. }

1. F# - исключительно удобный язык для обработки данных
2. F# позволяет легко распараллеливать и асинхронить вычисления
3. Инновации в F# делают его передовым языком для обработки данных в глобальной сети и облаке

Роль F# в экосистеме языков

- F# - новаторский язык, в котором воплощаются смелые идеи
  - F# 2.0 – асинхронное/параллельное программирование (перекочевало в C# 5.0)
  - F# 3.0 – type providers
- F# используется во внутренних проектах
  - Driver Development Kit 0- верификация кода
  - F# написан на F# (догадайтесь, как такое возможно?)
  - Игры XBox Live Arcade
  - AdCenter Challenge
- F# начинает проникать в различные сферы
  - Финансовая аналитика (на смену Excel)
  - Научные вычисления (на смену Matlab/MathCad -> eg. VSLab)
  - Genome Viewer
  - DSL


Производительность
Профессиональная разработка
C++
C#
Java
Programming
F#
Financial engineering
Algorithmic Trading
Modelling
Python…
Math-
ematica…
Выразительность для
системного
программирования
Выразительность
для математич
задач

Example #1 (Power Company)

I have written an application to balance the national power generation schedule ... for an energy company.

...the calculation engine was written in F#.

The use of F# to address the complexity at the heart of this application clearly demonstrates a sweet spot for the language ... algorithmic analysis of large data sets.

Simon Cousins (Eon Powergen)

Examples #2/#3: Finance companies

Insurance Company Improves Time-to-Market with Enhanced Rating Engine

Overview

Country or Region: United States
Industry: Financial services—Insurance

Customer Profile

Headquartered in Columbus, Ohio, Grange Insurance offers automobile, life, home, and business insurance protection to policyholders in 13 U.S. states. It employs 1,500 people.

Business Situation

To maintain its competitive standing and its reputation among agents for being easy to do business with, Grange Insurance needed to keep its rating engine working at top performance.

Solution

Using Microsoft® Visual Studio® Team System and Visual F#, the company

"With this streamlined developmer
rapidly deliver more powerful solu
they can deliver more choices and
policyholders that much faster."

Glenn Watson, Associate Vice President, Personal Lines, I

For nearly 75 years, Grange Insurance ha
products and services to policyholders ir
states. To maintain its well-earned reputa
company decided to enhance its rating e
for rating policies and performing what-i
analyses, and other vital activities. Worki
Group and using the Microsoft® Visual S
development environment and Microsof
ming language, Grange Insurance paralle

Banking Firm Uses Functional Language to Speed Development by 50 Percent

Customer: Financial services firm
Country or Region: Europe
Industry: Financial services—Banking

Customer Profile

A large European financial services firm offers banking and asset-management services to clients in 50 countries. In 2009, the bank earned more than U.S.$6 billion in income.

Software and Services

- Microsoft Visual Studio
  - Microsoft Visual F#
  - Microsoft Visual Studio 2010
- Technologies
  - Microsoft .NET Framework
  - Windows Presentation Foundation

"We could not have developed 200 models in two years without F# and Visual Studio. It would have taken us at least twice as long with our previous tools."

Director at a large European financial services firm

A large financial services firm in Europe sought new development tools that could cut costs, boost productivity, and improve the quality of its mathematical models. To address its needs, the bank deployed Microsoft F#, the Microsoft .NET Framework, and Microsoft Visual Studio. It will soon upgrade to Visual Studio 2010 and the integrated Microsoft Visual F#. With its new tools, the bank can speed development by 50 percent or more, improve quality, and reduce costs.

Business Needs

A large European financial services

desktop and on a remote cluster of servers that includes hundreds of systems

Example #4: Biotech

...F# rocks - building algorithms for DNA processing and it's like a drug. 12-15 at Amyris use F#...

F# has been phenomenally useful. I would be writing a lot of this in Python otherwise and F# is more robust, 20x - 100x faster to run and faster to develop.

Darren Platt, Amyris BioTechnologies

Case Study #5: Microsoft "adPredict"

- 4 week project, 4 machine learning experts
- 100million probabilistic variables
- Processes 6TB of training data
- Real time processing (**N,000** impression updates / sec)

"F# was absolutely integral to our success"

"We delivered a robust, high-performance solution on-time."

"We couldn't have achieved this with any other tool given the constraints of the task"

"F# programming is fun – I feel like I learn more about programming every day"

Используйте F#!

2012 CEE-SEC(R) Разработка ПО

2012 CEE-SEC(R) Software Engineering Conference in Russia

Дмитрий Сошников

dmitri@soshnikov.com
facebook.com/shwars
twitter.com/shwars
vk.com/shwars
blogs.msdn.com/sos