goto; conference aarhus

XQUERY – THE GETTING STUFF DONE LANGUAGE

Jim Fuller, Principle Consultant
MarkLogic

INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

gotocon.com

XQuery - The Getting Stuff Done language

Jim Fuller
email: jim.fuller@marklogic.com twitter: @xquery
Principal Consultant, Europe
13/10/11

I will try XQuery

Agenda

- The 5 W's - 5m
- Xquery Overview – 15m
- Survey Results and Analysis – 10m
- Cool Stuff – 5m
- Summary – 5m

5W's Scientific Method

What's the problem ?

MarkLogic™

The Mythical Man Month - Fred Brooks

'All programmers are optimists.'

'Adding manpower to a late software project makes it later.'

'The fundamental problem with program maintenance is that fixing a defect has a substantial (20-50 percent) chance of introducing another.'

'building software will always be hard. There is no silver bullet.'

Problem #1 - Programming is hard

MarkLogic™

Data problems

impedance mismatch

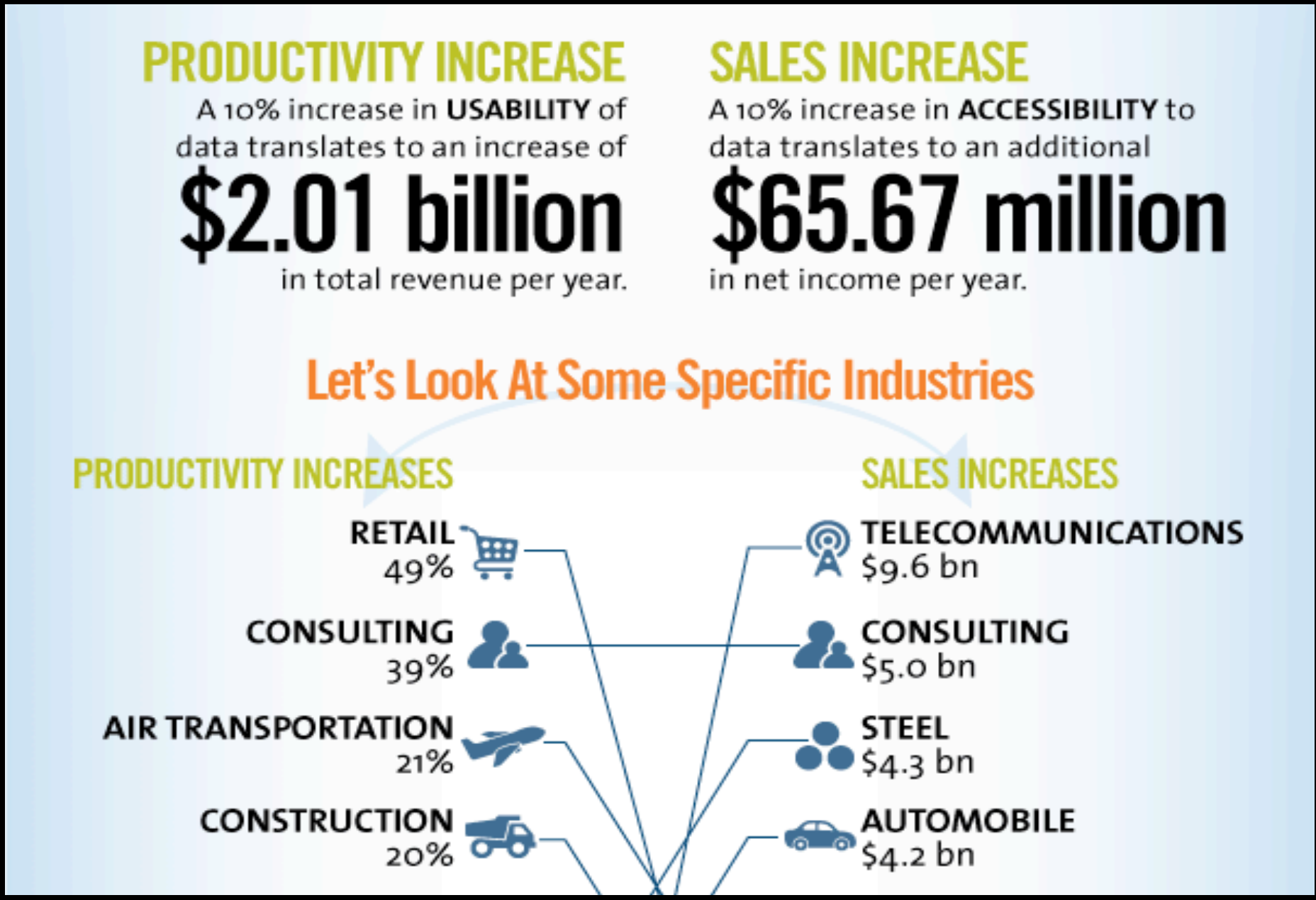
multifarious data models

relaxing constraints in vogue

* http://gigaom.com/cloud/big-data-equals-big-opportunities-for-businesses-infographic/

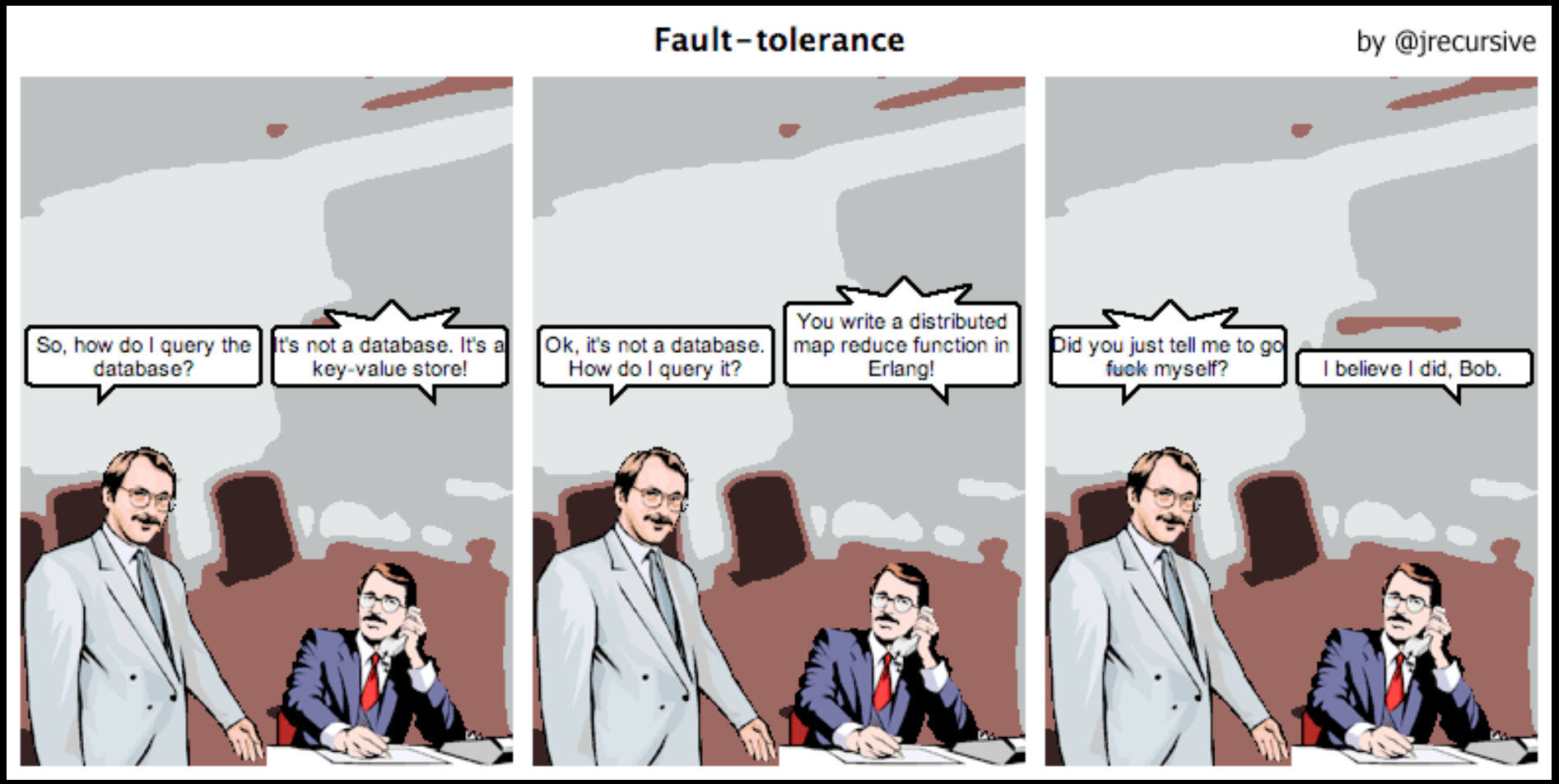
MarkLogic™

* http://gigaom.com/cloud/big-data-equals-big-opportunities-for-businesses-infographic/



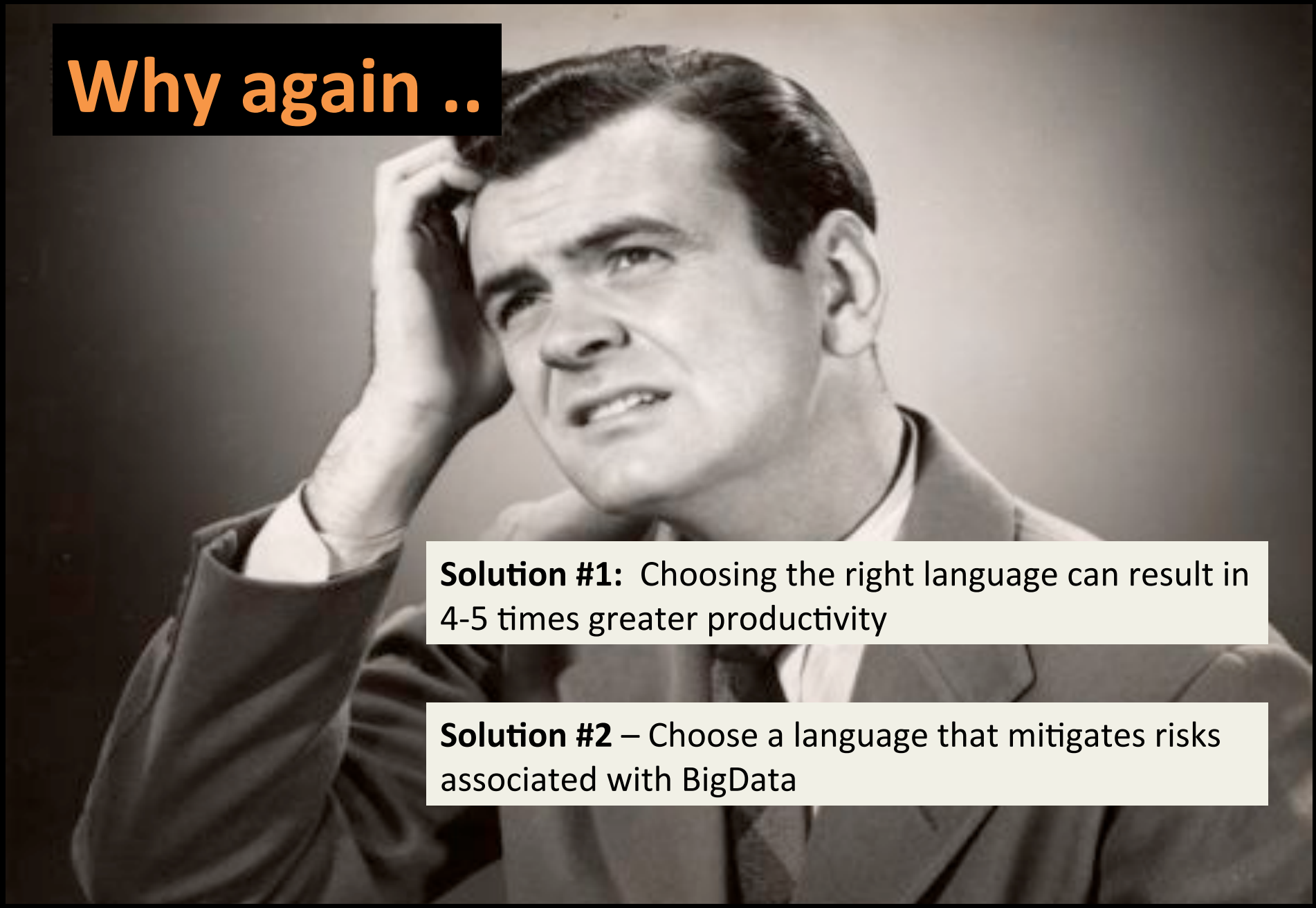
MarkLogic™

managing data variability, volume & velocity is hard



Problem #2 - Your boss knows about the Big Data opportunity

Why again ..



Solution #1: Choosing the right language can result in 4-5 times greater productivity

Solution #2 – Choose a language that mitigates risks associated with BigData

MarkLogic™

XQuery Overview

MarkLogic™

I will try XQuery

misconceptions

XQuery

The dynamic **functional language**

XPATH^2

Domain specific language

Strongly typed

SQL like –Inner Join

```
SELECT * FROM employee, department
    WHERE employee.DepartmentID =
    department.DepartmentID;
```

```
for $emp in //employee
return
    $emp[@id eq //dept/@id]
```

SQL like – left Outer Join

```
for $u in fn:collection('customers')
return
<customer id={$u/custno} name="{$u/name}"> {
    for $p in fn:collection("purchaseorders")//po
        where $u/custno = $p//custno
        return <po>{$p/@id}</po>
} </customer>
```

Functions

```
declare function local:hello($name) {
  concat("Hello ", $name)
};

local:hello(”Aarhus!")
```

Code in the language of the domain

```
declare function local:test(
    $a,$b,$c,$d,$e,$f,$g,$h,$i,$k){
    .....};
```

```
declare function local:test(
    $a as element(record)
){ ......};
```

Literals and Constructors

```
xquery version "1.0";

let $names :=
("Jim","Gabi","Vojtech","Norm","Nuno","Eric")
return
<names>{
  for $name in $names
  return
    <name>{$name}</name>
}</names>

(:

element {$computed-element-name}{
 attribute {$computed-attr-name}{"some attr value"}
}

:)
```

Inline Caching

```
xquery version "1.0-ml";

(: xquery memoization example for use with MarkLogic :)

declare variable $cache := map:map();

declare function local:factorial($n as xs:integer) as xs:integer {
if ($n < 0) then
  (0)
else if ($n = 0) then
  (1)
else
  $n * local:factorial($n - 1)
};

declare function local:memoize($func,$var){
  let $key := xdmp:md5(concat($func,string($var)))
  return
  if(map:get($cache,$key)) then
   (map:get($cache,$key), xdmp:log('cache hit'))
  else
    let $result := xdmp:apply($func,$var)
    return
      (map:put($cache,$key,$result), $result)
};

let $memoize := xdmp:function(xs:QName("local:memoize"))
let $factorial := xdmp:function(xs:QName("local:factorial"))

let $a:= xdmp:apply($memoize, $factorial, 20)
let $b:= xdmp:apply($memoize, $factorial, 20)
let $c:= xdmp:apply($memoize, $factorial, 20)
let $d:= xdmp:apply($memoize, $factorial, 20)
let $e:= xdmp:apply($memoize, $factorial, 20)
return
$a
```

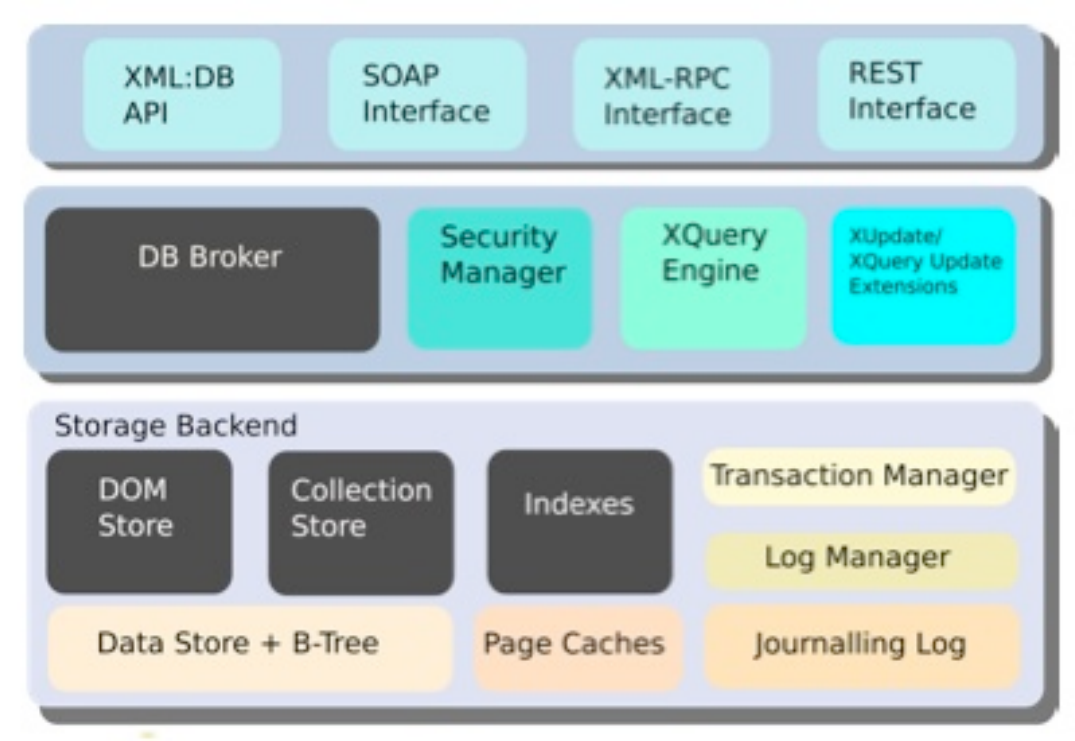
Broad applicability

http://github.com/xquery/xquerydoc – XQuery parsing XQuery

IBM Developerworks – Classify content with XQuery article

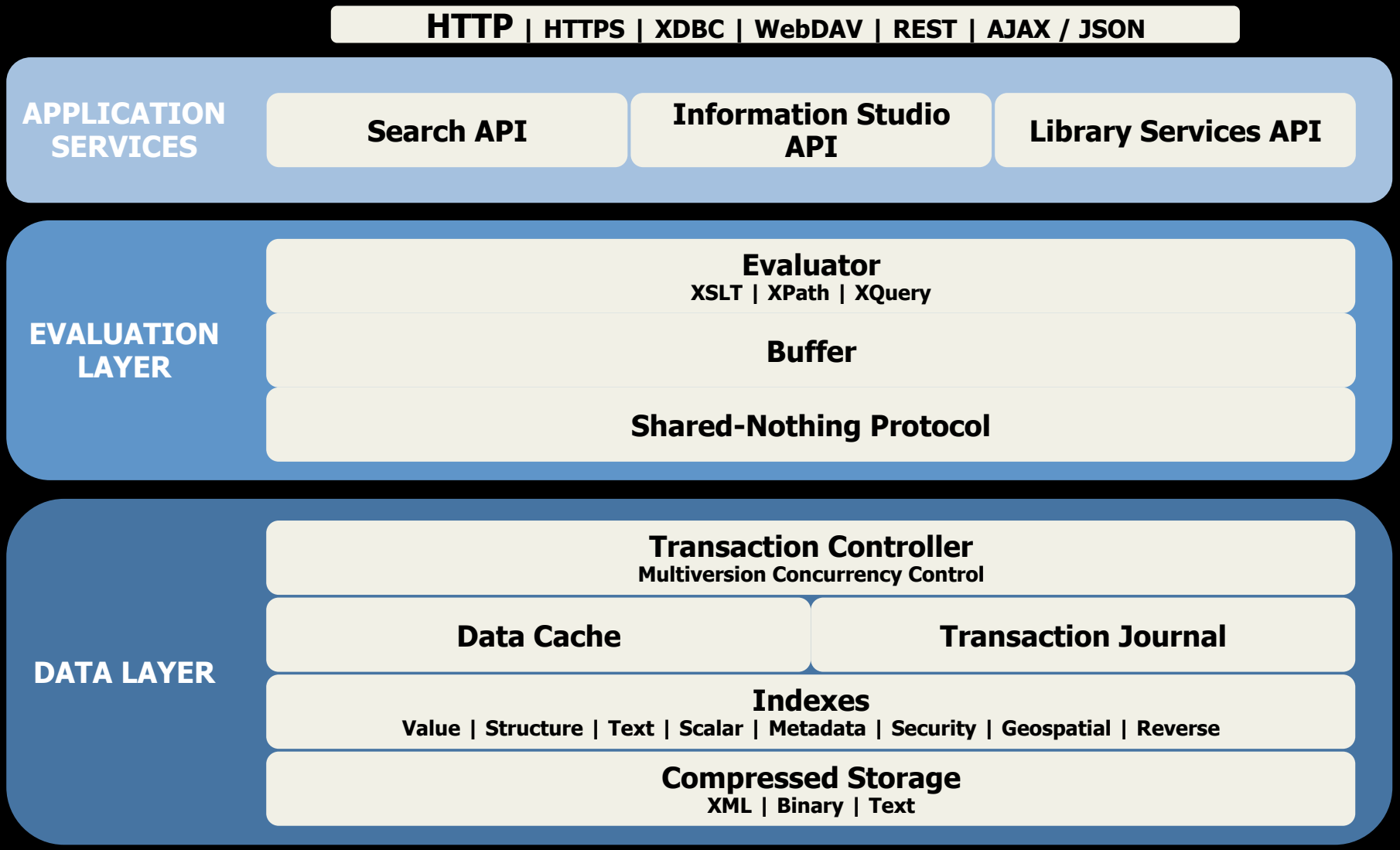
http://try.zorba-xquery.com/

XQuery + database



http://demo.exist-db.org/exist/eXide/index.html

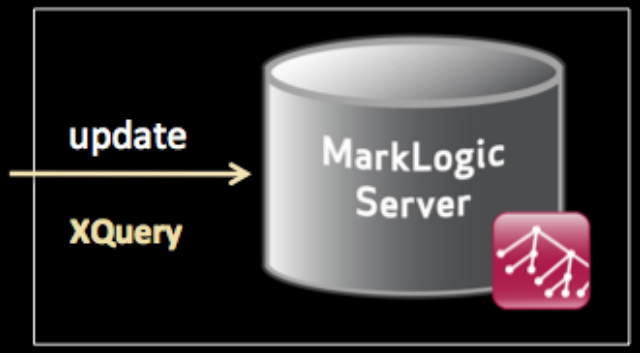
MarkLogic

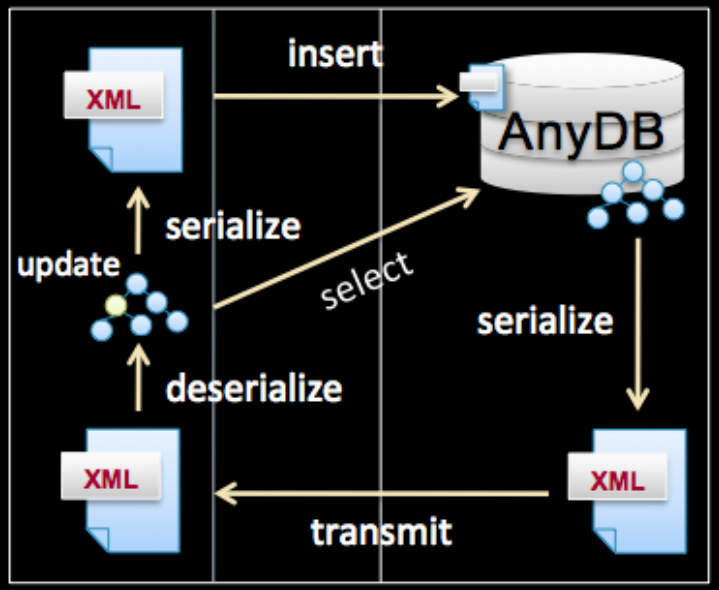


MarkLogic™

XQuery + database

- W3C Open Standard
- Standalone programming language
- Data Focused
- Single Tier Development
- Rich API
  - Regular expressions, strings, sequences, etc...
- Extensible with your own libraries
- Side-effect free
  - Great for concurrency
  - XDM





MarkLogic

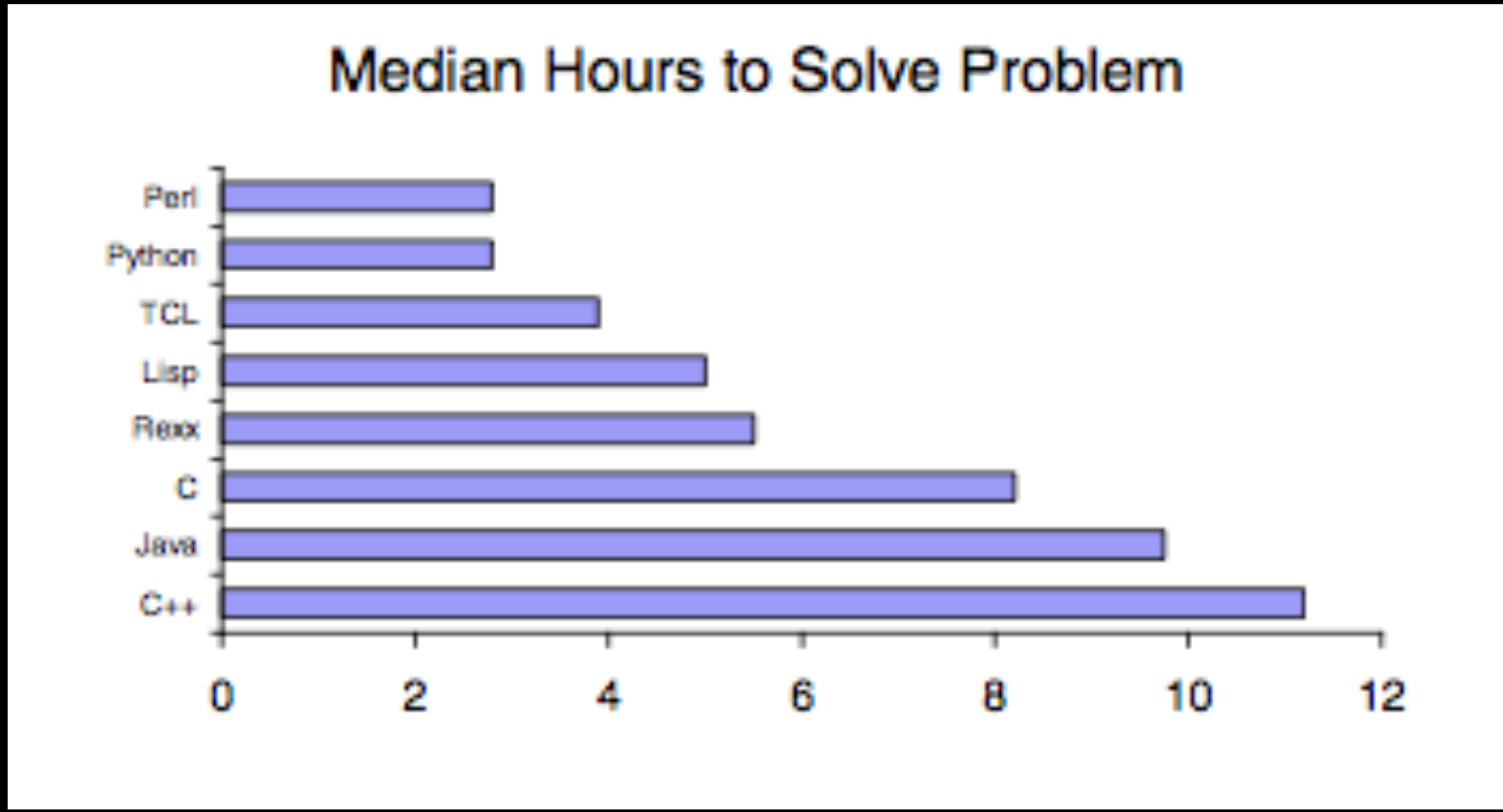
Analysis

MarkLogic™

I will try XQuery

Programming Language Productivity

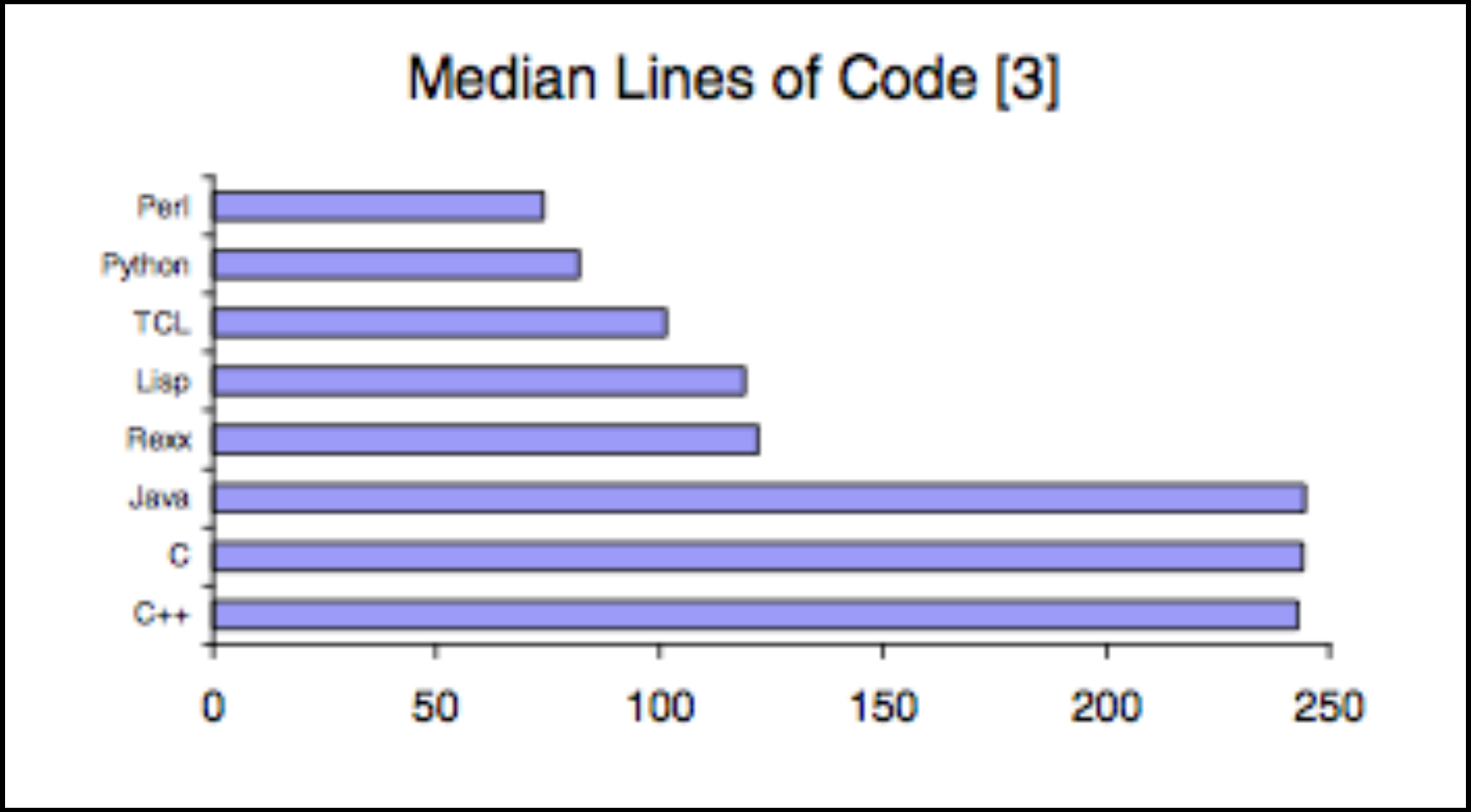
*Data compiled from studies by Prechelt and Garret of a particular **string processing** problem - public domain 2006.*

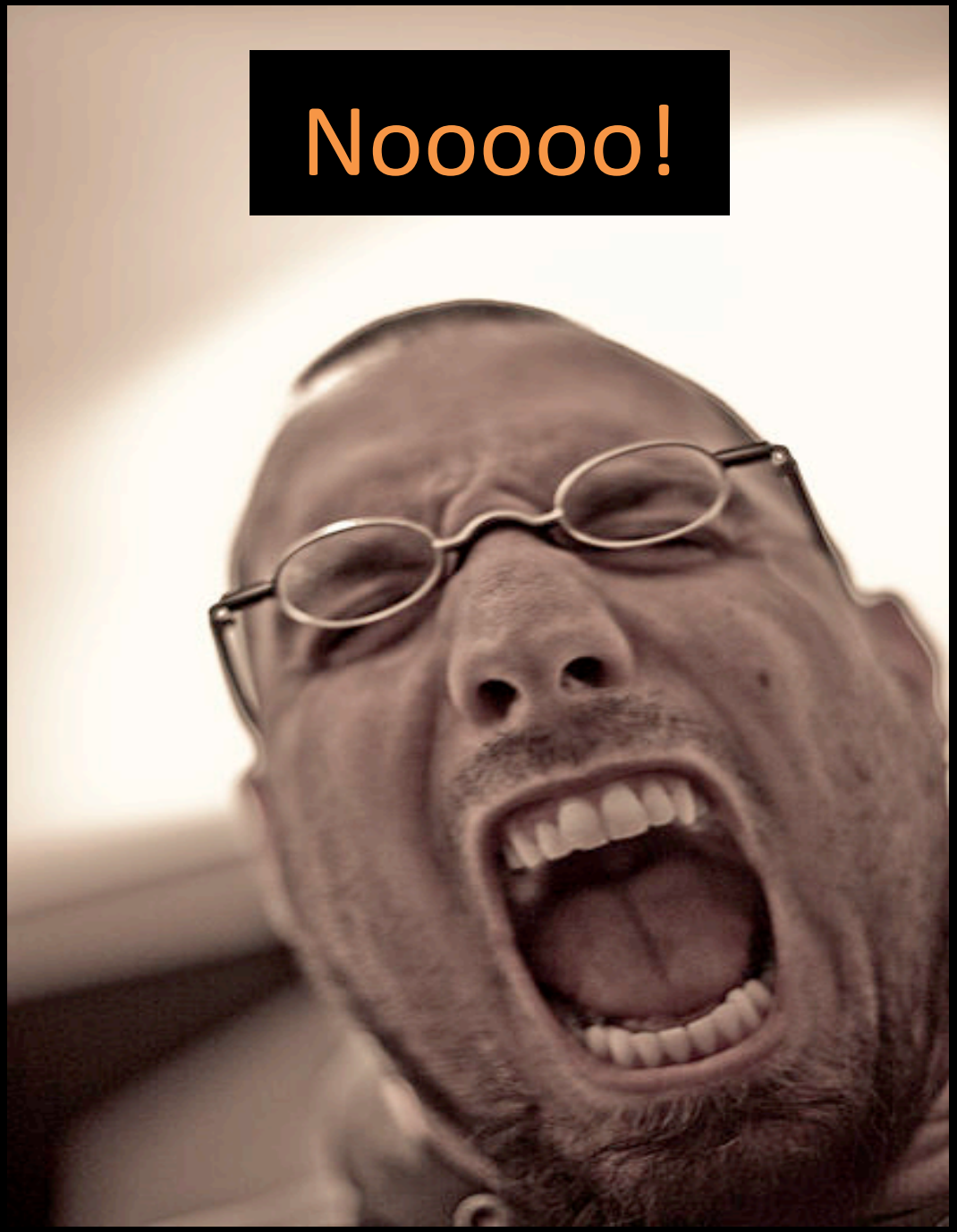


MarkLogic™

Programming Language Productivity

Data compiled from studies by Prechelt and Garret of a particular string processing problem - public domain 2006.





Methodology

#loc per FP

=

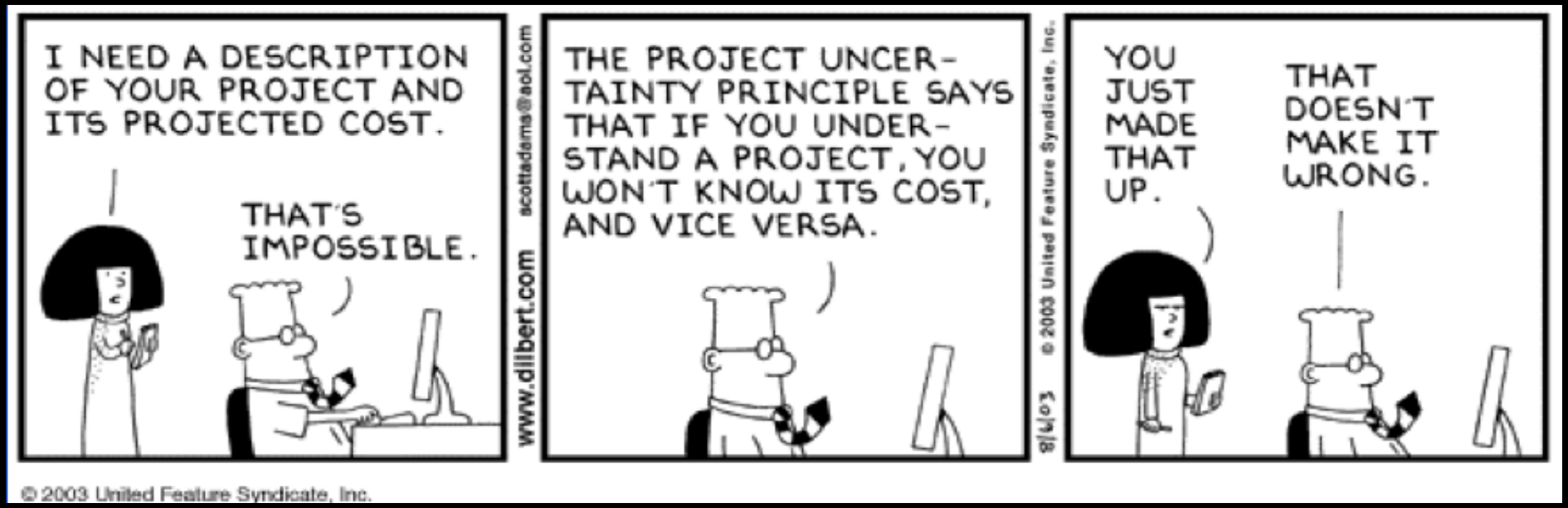
Lines of code

Per

Function Point

MarkLogic™

Project Uncertainty Principle



An empirical comparison of C, C++, Java, Perl, Python, Rexx, and Tcl for a search/string-processing program

Lutz Prechelt (prechelt@ira.uka.de) Fakulta¨t fu¨r Informatik
Universita¨t Karlsruhe

Language	#loc per Function Point
C	91
C++	53
Java	54
Perl	21

* Designing and writing programs using dynamic languages tended to take half as long as well as resulting in half the code.

Historical #loc per FP

Language	#loc per Function Point
Python	42-47
Java	50-80
Javascript	50-55
C++	59-80
C	140
…	…

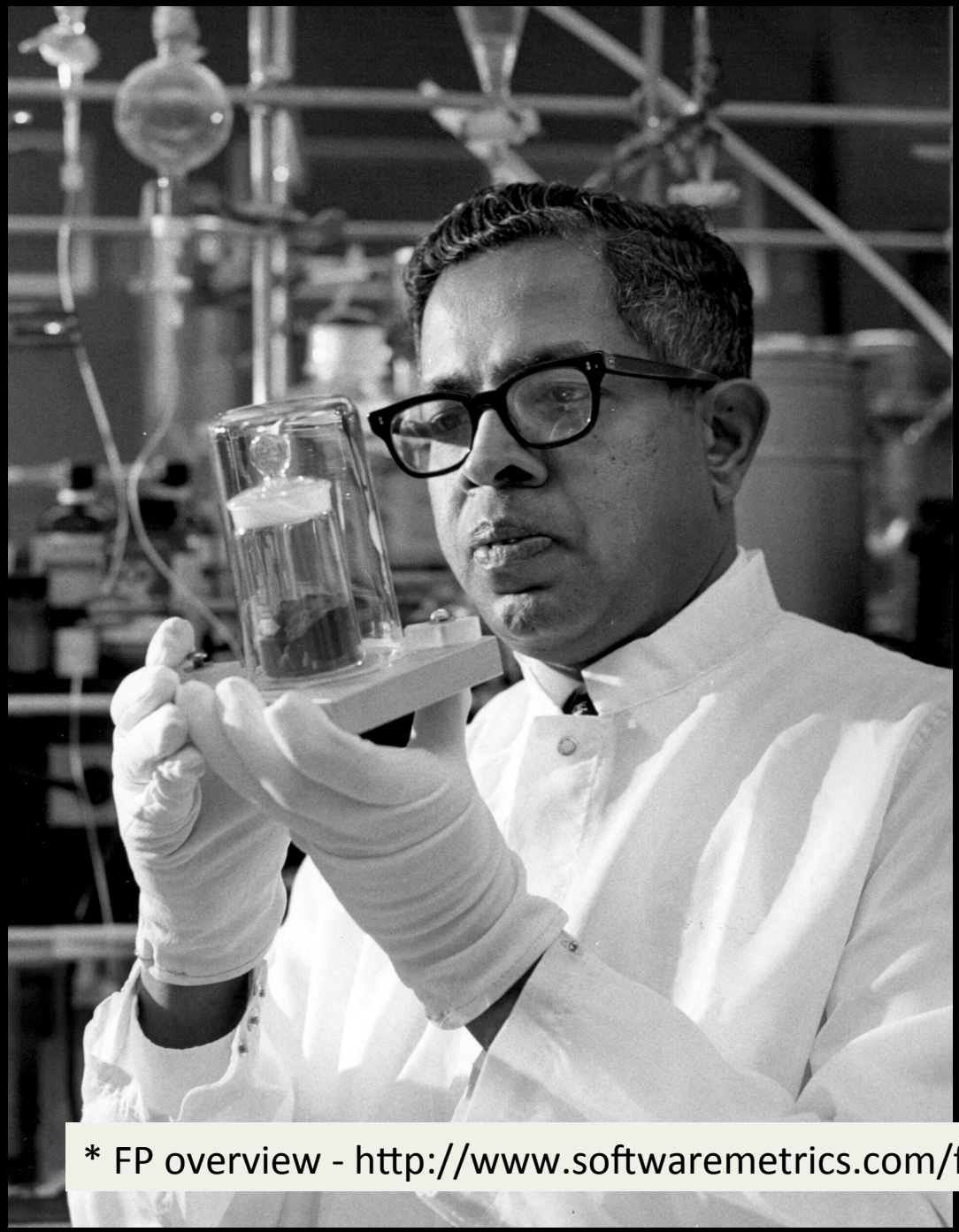
Developing an Enterprise Web Application in Xquery - 2009 Martin Kaufmann, Donald Kossmann

	Java/J2EE	**XQuery**
Model	3100	240
View	4100	1500
Controller	900	1180
	8100 (?)	**2920** (3490)

* 28msec – 2011
http://www.28msec.com/html/home

	Java	XQuery
SimpleDB	2905	572
S3	8589	1469
SNS	2309	455
	13803	**2496**

MarkLogic™



Review 11 projects

FP Analysis

Calc FP inputs/outputs

Calc VAF (0.65 + [(Ci) / 100])

AVP = VAF * sum(FP)

#loc

using cloc

= #loc per FP

* FP overview - http://www.softwaremetrics.com/fpafund.htm

MarkLogic™

Language	#loc per Function Point
Eiffel	21
SQL	13-30
XQuery	27-33
Haskell	38
Erlang	40
Python	42-47
Java	50-80
Javascript	50-55
Scheme	53
C++	59-80
C	128-140

* Cloudera– 2011 http://www.cloudera.com/videos/introduction-to-apache-pig

	locc	hrs
Java	200	4
PigLatin	10	.25

PigLatin is a DSL for data for apache hadoop

Do Software Language Engineers Evaluate their Languages?

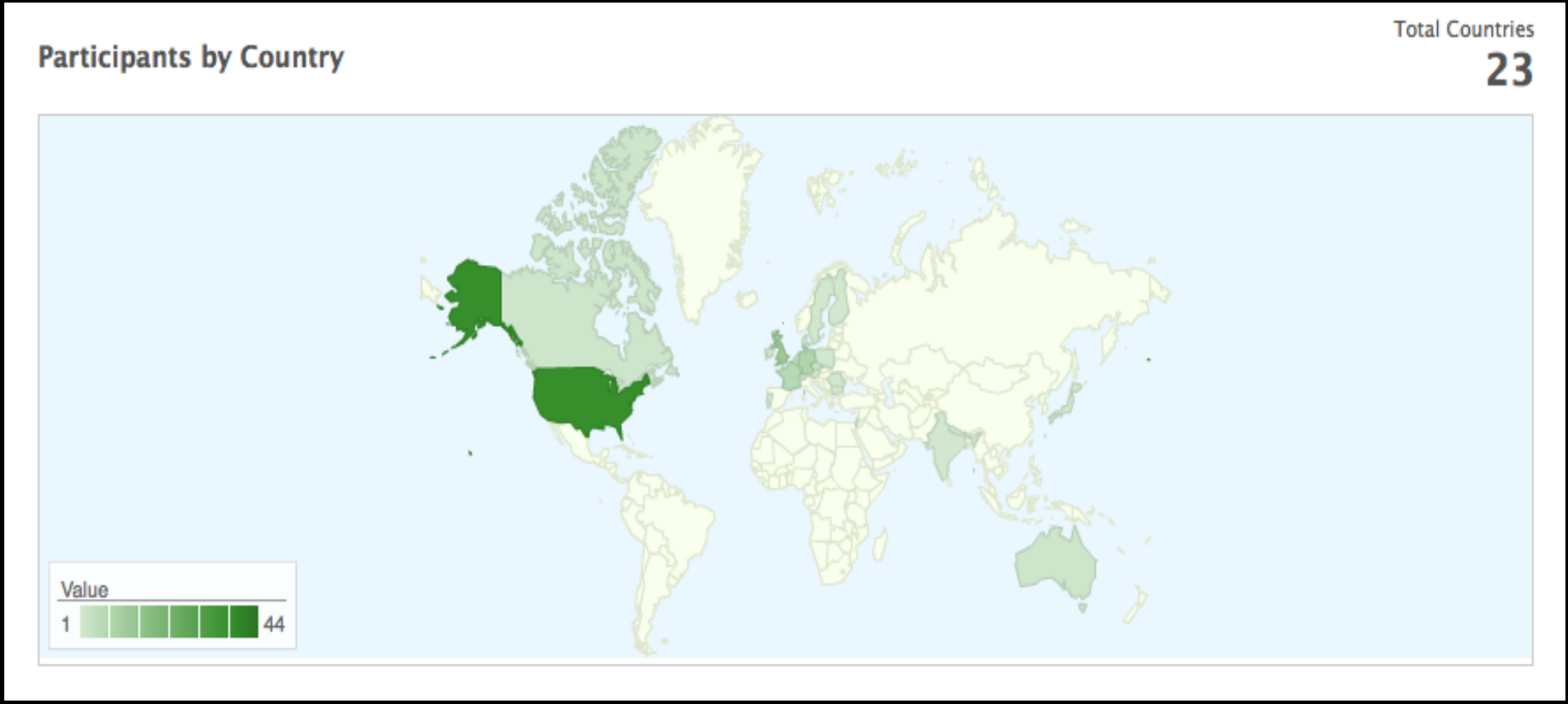
2011 - CITI, Departamento de Informa´tica, Faculdade de Ci^encias e Tecnologia, FCT, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal pedrohgabriel@gmail.com, {miguel.goulao,vasco.amaral}@di.fct.unl.pt http://citi.di.fct.unl.pt/

SLE consistently relax language evaluation

no validation from users

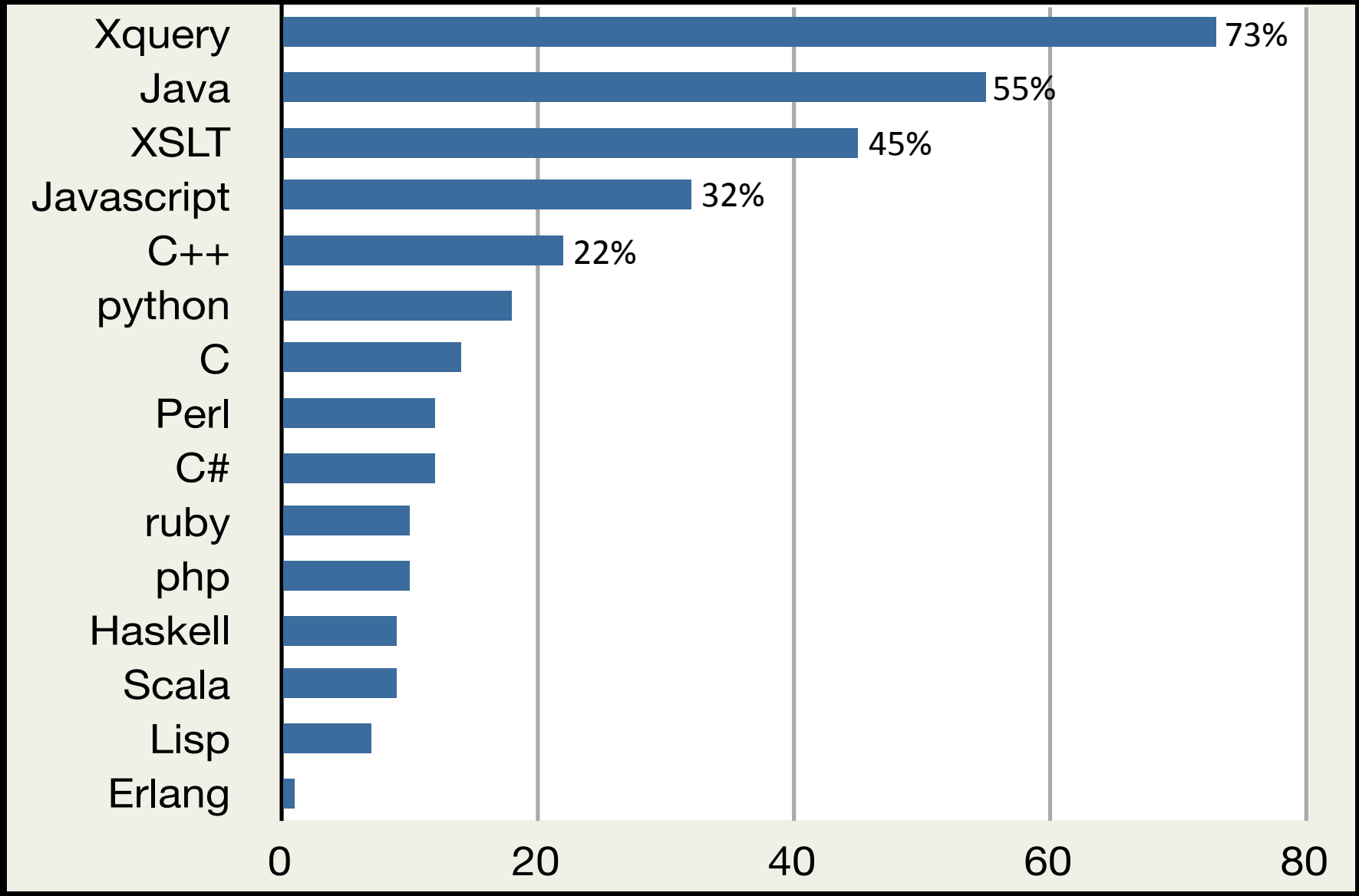
recommend systematic approach to DSL evaluation

Xquery 2011 Survey



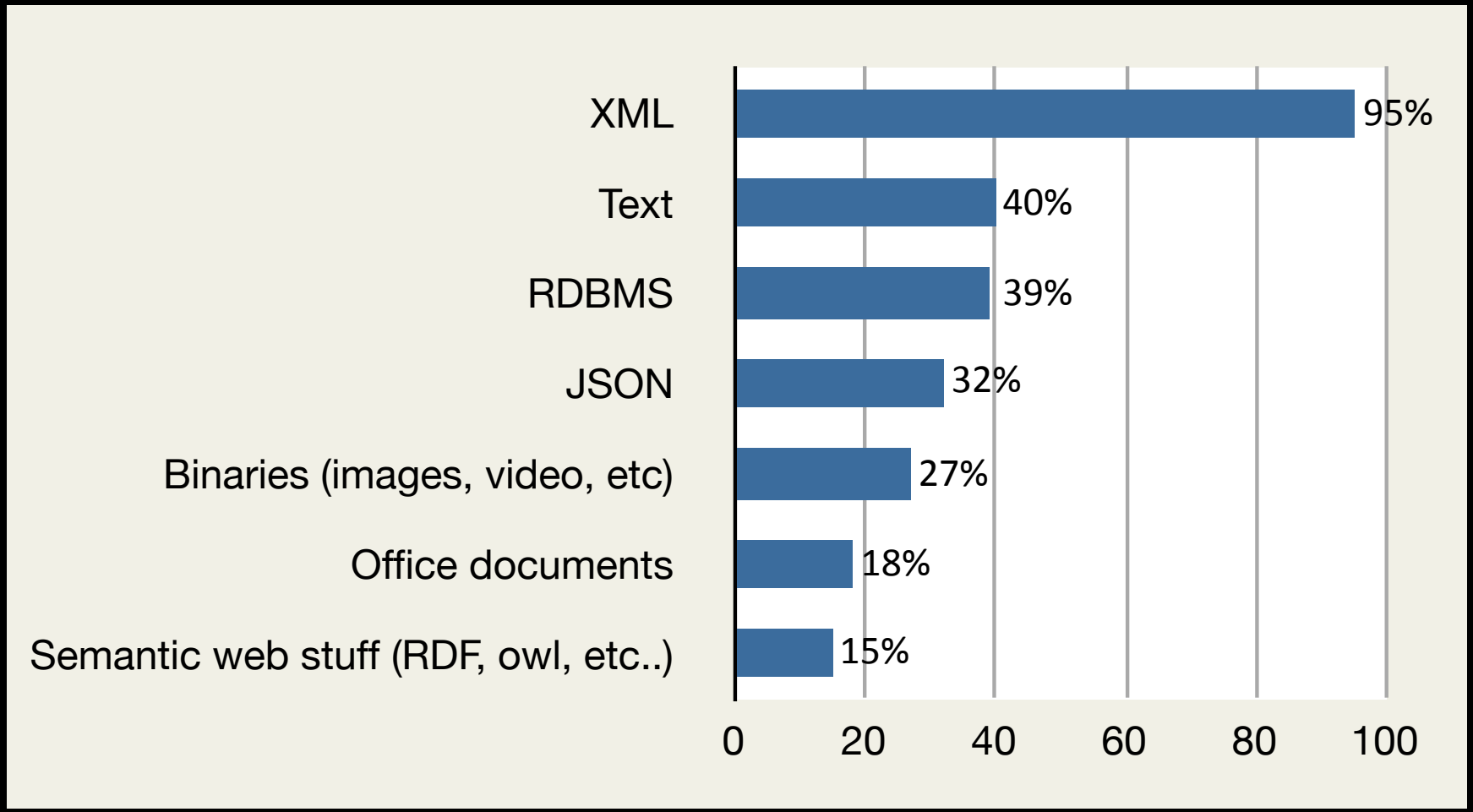
MarkLogic™

Preferred Programming Language



MarkLogic™

Which data formats do you use the most ?



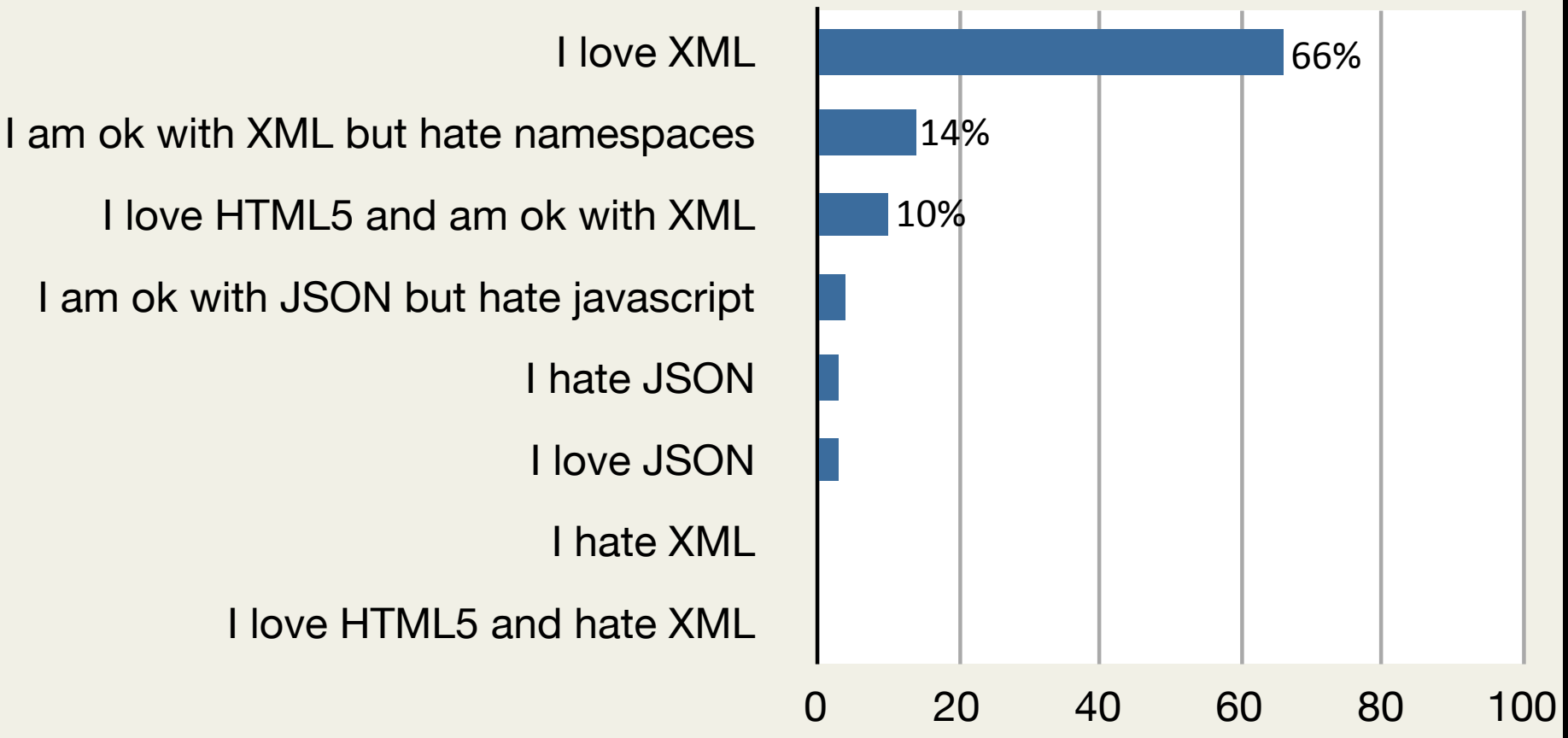
MarkLogic™

Written answers

What was different about learning or using XQuery versus other programming languages ?

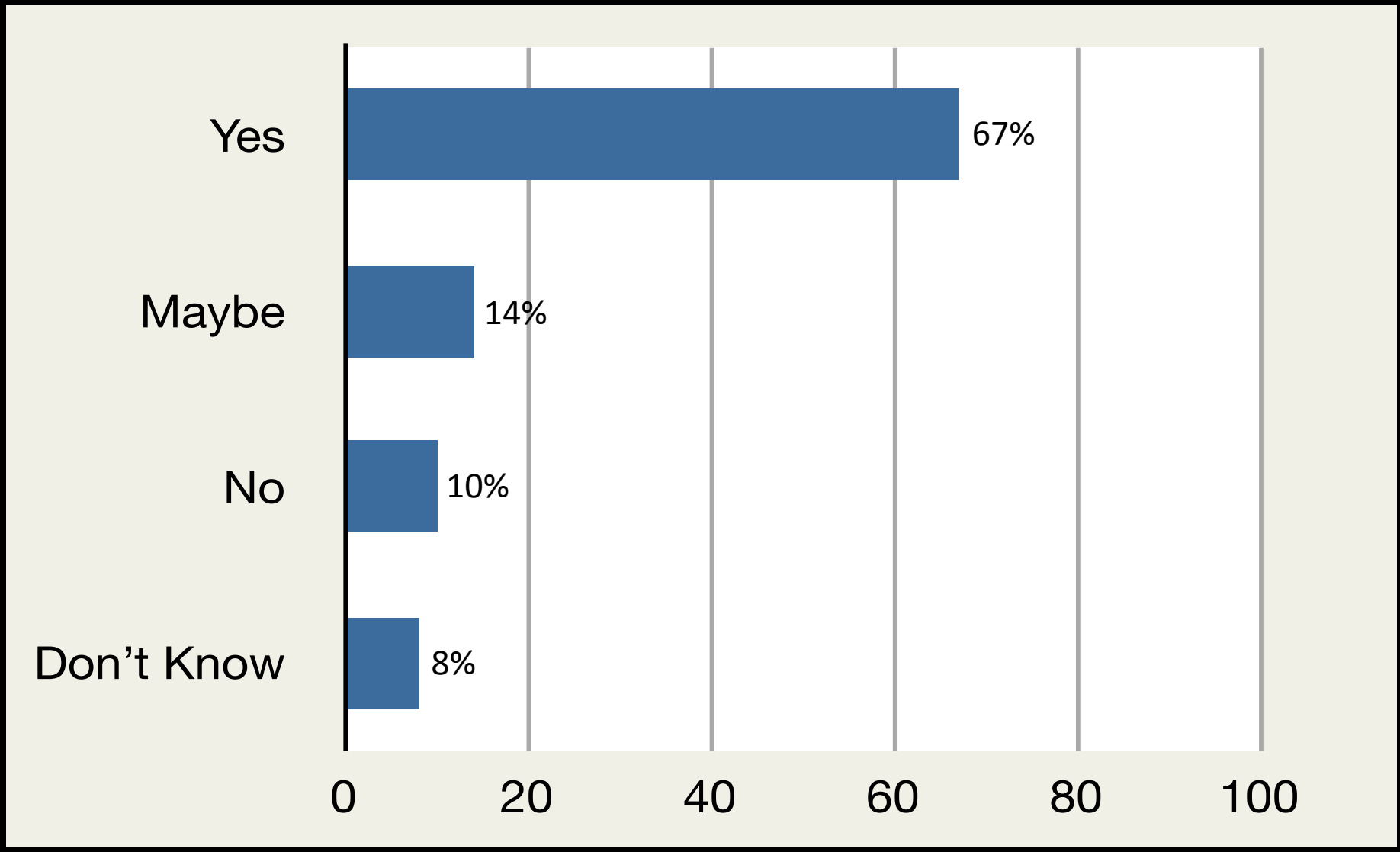
Do you have any opinions if XQuery is more or less productive language versus other programming languages ?

Choose an option that you most strongly believe in …



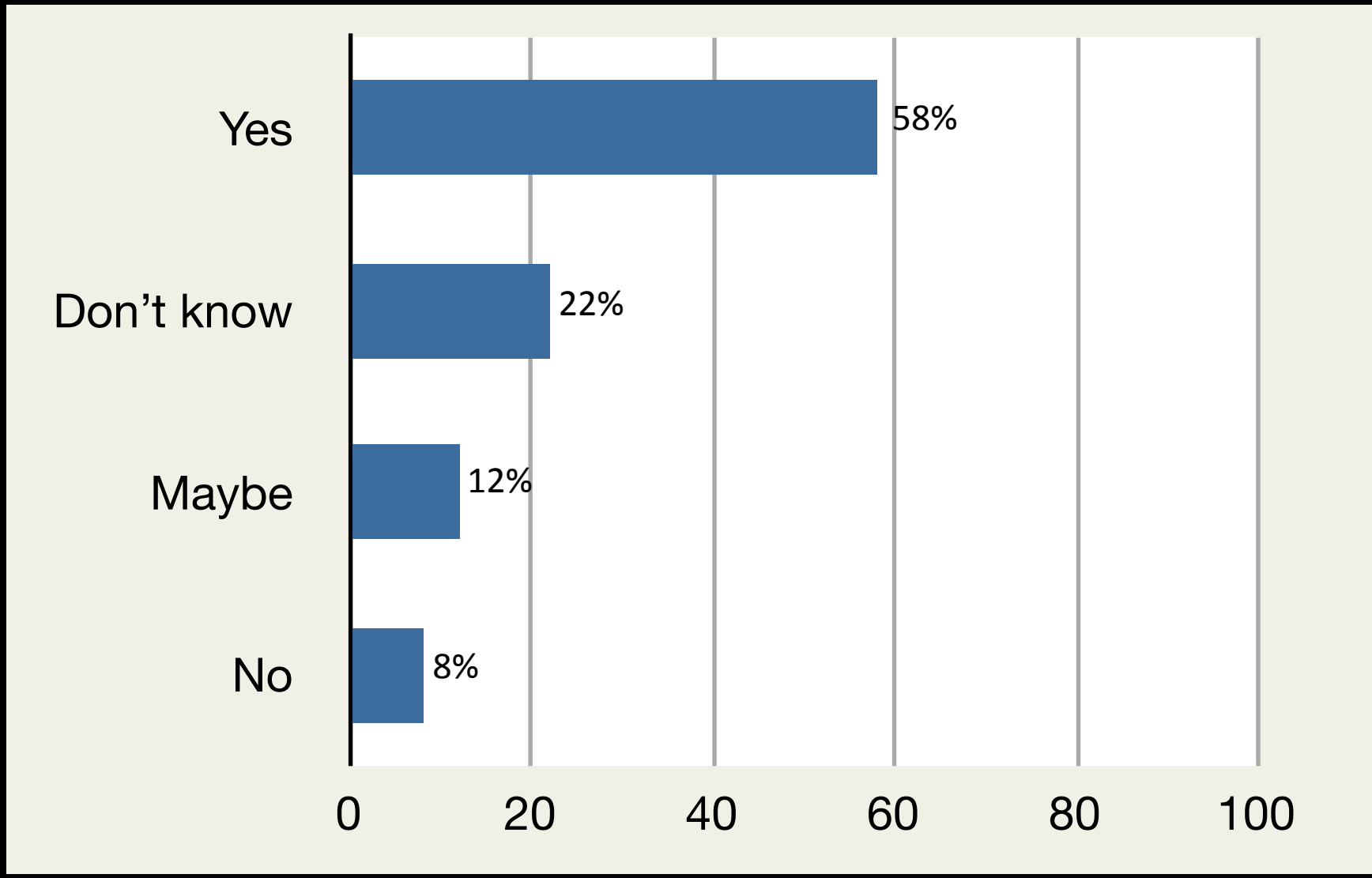
MarkLogic™

Do you think XQuery makes you a more productive programmer ?



MarkLogic™

Is XQuery more productive then Java in developing web based data applications ?



MarkLogic™

Cool Stuff

MarkLogic™

I will try XQuery

Corona

Drop in replacement for NoSQL with no compromise

Open Source development supported by ML

Written in XQuery

MarkLogic™

BigData

- Show ML working with
  - Xml
  - Text
  - Binaries
- Hadoop connector

Summary

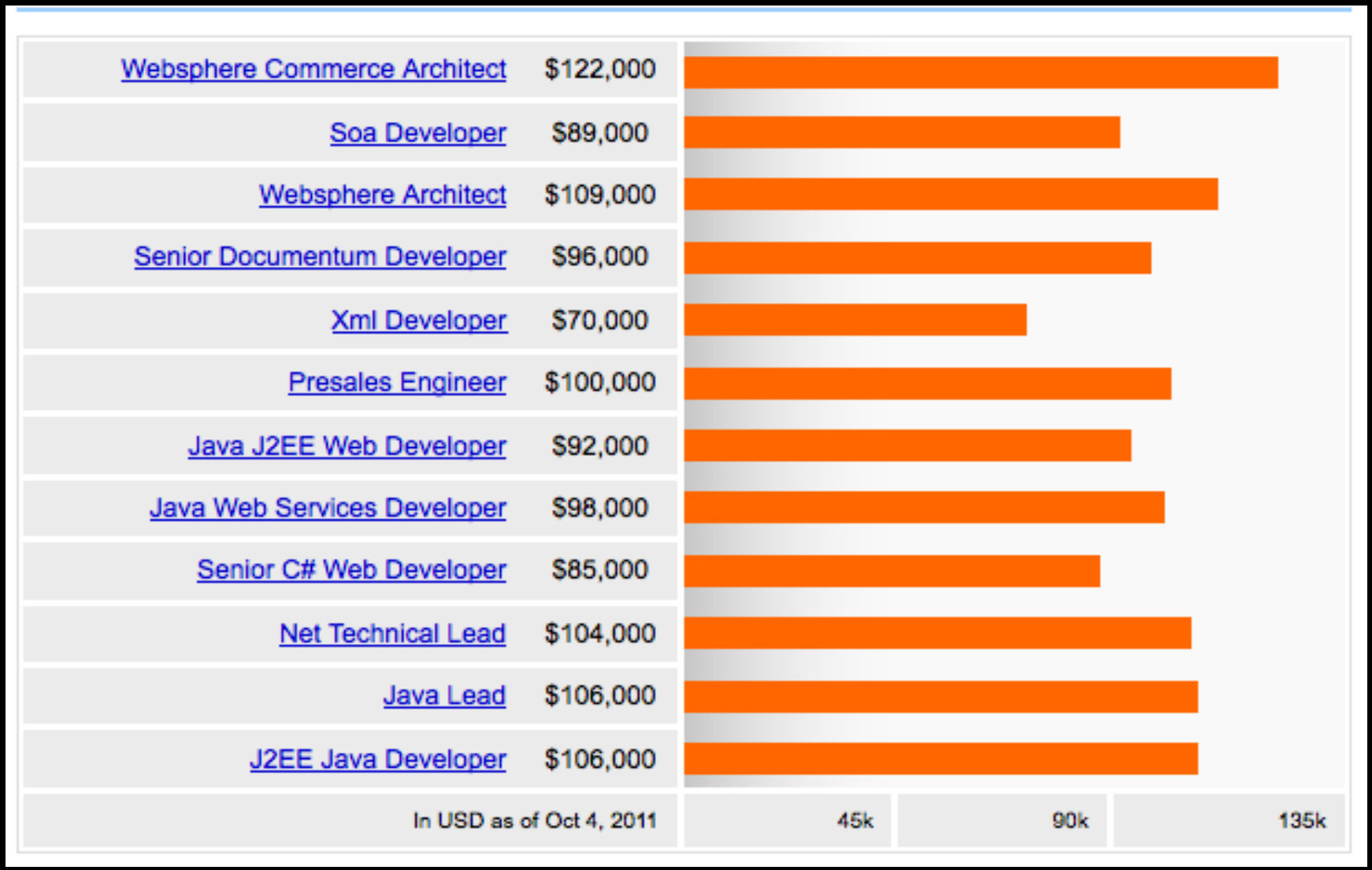
MarkLogic™

I will try XQuery

Sanity Check

MarkLogic™

Xquery job search indeed.com



MarkLogic™

MarkLogic Customers Know

Government Customers

Media Customers

Financial Services and Other Customers

MarkLogic™

MarkLogic™

I will try XQuery

Disclaimer: I have used the above subliminal suggestion throughout the presentation

Thank you ...@xquery Questions ?

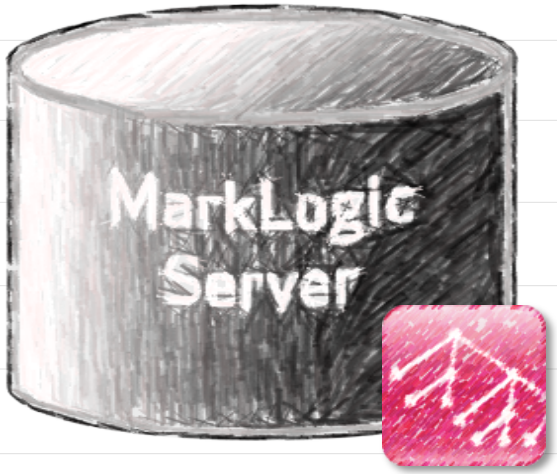
MarkLogic™

MarkLogic®

unstructured
schema-less*
easy evolution.
xml or json.

c++ core
~ pb scale

also stores:
text and binaries



features
acid, backups
replication, query
language (XQuery).

native search
a database built
on a search engine?

!! stop shredding your data
start storing data as is

no tables, rows, columns
thinkin' documents
uris? looks like a filesystem

* they have this universal index thing.
an inverted index that is structure aware

Resources and References