Why Software Engineering is the Best Job in 2012

Judith Bishop
Director of Computer Science
Microsoft Research Connections
From a report in March 2012

Based on

- Work Environment
- Physical Demands
- Job Outlook
- Income Levels
- Stress

Some History

Eclipse software development environment with its extensible plug-in system.

- From IBM Canada, 2001
- Now in a Foundation
- Free and open source under its own license
- Strong community base

Java programming language and its run-time platform

- From Sun Microsystems in 1995
- Now with Oracle
- Free and Open Source under GNU public license
- Strong research, education and developer base
- Part of browser technology
Microsoft collaborates with universities

**Why?**
- Expand a lab’s research base from 100s to 10,000s
- Verify the work in practice
- Launch new applications
- Support tomorrow’s leaders

**How?**
- Awards, grants, prizes
- Interns (1,000 a year)
- Visitors
- And more
Amazing numbers

- More than **100** top PhD research students from leading European universities in Microsoft Research Fellowships
- More than **25,000** scientists, academics and students have attended Microsoft-sponsored conferences, summits and workshops since 2005
- Nearly **1,000** interns selected for internships at our labs in Redmond, China and India annually
- **465** students from more than 35 countries have interned at the Cambridge lab in the past 7 years
- **50** conferences sponsored in the past 5 years for $350,000
Computer Science Activities

Software Engineering
- Innovation Foundation (SEIF)
- Interaction with RiSE
- Promoting Visual Studio through ICSE events

Semantic Computing
- Interaction with NLP, ISRC, Bing
- Developed Language Models Services
- Bing-MSR Speller Challenge
- Developing Knowledge Services

Mobile Computing
- Interaction with Network Research team in MSR-R
- Student projects on MSR-R phones at 17 universities
- Promoting Azure and WP7

Parallelism and Concurrency
- Book on Patterns and Practices group
- Course on Concurrency and Parallelism Promoting C# and F#

Programming Languages
- Developing a Try F# system in a browser
- Workshops on “F# in Education”
- Supporting Pex4Fun
- TouchDevelop outreach and book

Community interaction
- CS EdWeek
- Work with the NSF, ACM, IEEE and IFIP
- MSR Software Summits
- Support for 12 conferences

CS is research-oriented with strong internal and external impact
How collaboration has grown, 1996-2011

Mapping Collaboration Networks in ‘Programming’ Conferences

* Circle size represents number of publications authored by each country. Lines between countries indicate publication co-authorship. Line width indicates the degree of collaboration (wider = more collaborations). Conferences analyzed include PPOPP, POPL, PLDI, and OOPSLA.
PL Collaboration 2011
## Top 12 CS Research Organizations

<table>
<thead>
<tr>
<th>Organization</th>
<th>Publications</th>
<th>H-Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford University</td>
<td>35354</td>
<td>327</td>
</tr>
<tr>
<td>University of California Berkeley</td>
<td>32888</td>
<td>315</td>
</tr>
<tr>
<td>Massachusetts Institute of Technology</td>
<td>36272</td>
<td>312</td>
</tr>
<tr>
<td>Microsoft</td>
<td>36967</td>
<td>303</td>
</tr>
<tr>
<td>Carnegie Mellon University</td>
<td>38826</td>
<td>272</td>
</tr>
<tr>
<td>IBM</td>
<td>47624</td>
<td>261</td>
</tr>
<tr>
<td>Princeton University</td>
<td>11242</td>
<td>211</td>
</tr>
<tr>
<td>Cornell University</td>
<td>13488</td>
<td>208</td>
</tr>
<tr>
<td>University of Illinois Urbana Champaign</td>
<td>26523</td>
<td>204</td>
</tr>
<tr>
<td>AT&amp;T Labs Research</td>
<td>9471</td>
<td>203</td>
</tr>
<tr>
<td>University of Washington</td>
<td>15524</td>
<td>202</td>
</tr>
<tr>
<td>Google Inc.</td>
<td>9459</td>
<td>192</td>
</tr>
<tr>
<td>Organizations</td>
<td>Publications</td>
<td>H-Index</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>Microsoft</td>
<td>741</td>
<td>29</td>
</tr>
<tr>
<td>Carnegie Mellon University</td>
<td>568</td>
<td>23</td>
</tr>
<tr>
<td>IBM</td>
<td>796</td>
<td>20</td>
</tr>
<tr>
<td>The French National Institute for Research</td>
<td>448</td>
<td>19</td>
</tr>
<tr>
<td>University College London</td>
<td>208</td>
<td>19</td>
</tr>
<tr>
<td>University of Washington</td>
<td>192</td>
<td>19</td>
</tr>
<tr>
<td>Georgia Institute of Technology</td>
<td>270</td>
<td>18</td>
</tr>
<tr>
<td>University of Maryland</td>
<td>256</td>
<td>18</td>
</tr>
<tr>
<td>University of California Irvine</td>
<td>233</td>
<td>18</td>
</tr>
</tbody>
</table>
Agenda

1. Software engineering in the large
2. Software in the browser
3. Software social experience
4. Social aspects in the large
5. Software development kits
2. SOFTWARE ENGINEERING IN THE LARGE
Microsoft ships software to 1 billion users around the world. How do we find out when things go wrong?

We want to:

- fix bugs regardless of source
- prioritize bugs that affect most users
- generalize the solution for any programmer
- get the solutions out to users efficiently
- try to prevent bugs in the first place
Debugging in the Large with WER...
Engine for WER bucketing heuristics
Extension to the *Debugging Tools for Windows*
input is a minidump, output is bucket ID
runs on WER servers (and programmers desktops)
500 heuristics
grows ~ 1 heuristic/week
Top 20 Buckets for *MS Word 2010*

Just 20 buckets account for **50% of all errors**

Fixing a small # of bugs will help **many users**
Fixing bugs in software

- First use found $\geq$5-year old heisenbugs in Windows
- Windows Vista team fixed 5,000 bugs in beta
- Anti-Virus vendor fixed top 20 buckets and dropped from 7.6% to 3.6% of all kernel crashes
- Office 2010 team fixed 22% of reports in 3 weeks
- And you can fix yours...
WER helped fix hardware error

Manufacturer **could** have caught this earlier with WER
Renos Malware

Early detection w/o user action (renos, blaster, slammer, etc.)

WER scales to handle global events
3. SOFTWARE IN THE BROWSER
Types of software

1. Only a browser, e.g., Explorer, Firefox, Safari

2. A platform and language(s), e.g., a CLI implementation and C# or the JVM

3. An integrated development environment (IDE), e.g., Visual Studio or Eclipse
Browser based software

**Sandbox Approach**

- Download a Silverlight/ Moonlight control with a complete compiler
- All interaction directed to the control from the browser
- Computation on client
- Con: Effort to create the system
- Pro: No additional hardware needed

**Server Approach**

- Maintain a server (or cloud) presence
- All interaction directed to the server from the browser
- Computation on server
- Con: Scalability issue
- Pro: Can gather data on usability
Building Rich User Interfaces

Don Syme, Dean Guo, Christophe Poulain, Joe Pamer, Laurent le Brun, Nigel Horspool, Judith Bishop, 2010
Try F# Control
Built in Silverlight

Control is scriptable:

```csharp
string Script { get; set; }
string Output { get; }
void ClearOutput()
int LinesToEcho { get; set; }
Canvas Canvas { get; }
void ClearCanvas()
CanvasPosition CanvasPosition { get; set; }
void LoadFromString(string script)
void LoadFromUrl(string url)
void LoadFromFile()
bool IsDirty { get; set; }
void Save()
void Execute(string script)
void Cancel()
void Reset()
```

Compiled into XAP file
- Manifest
- DLL for our control
- FSharp.Hostable.dll
- FSharp.Core.dll
- .NET Silverlight DLLs

Try F# Web Site
Deployed in Windows Azure

ASP.NET Web Site
- Tutorials.aspx
- <other>.aspx
- XAP file
- Tutorial Content
  - Index (XML)
  - Directory
  - HTML pages
  - Resources

Hosted in Windows Azure
- Front-end Web Role #1
- Front-end Web Role #N
- Public Internet
- Load Balancer
- Blob storage (IIS Logs)
- Offline processing of IIS logs
Demo of F#
- **Pex** - Visual Studio 2010 Power Tool developed by Microsoft Research to help unit testing of .NET applications.
  - Can be launched from the command line and run as Type 2 or Type 3 software.

- **Pex4Fun**
  - radically simplified version of the fully featured Pex
  - accessed via a browser
  - all the work happens on one of Microsoft Research servers
  - creates a game out of unit testing by providing existing or user entered code puzzles in C#, Visual Basic, or F#
  - Users determine from the unit tests what code needs to be added or changed.
The code is a puzzle. Do you understand what the code does? Click Ask Pex! to find out.

```csharp
using System;

public class Program {
    // What values of v can cause an exception? Ask Pex to find out!
    public static void Puzzle(int[] v) {
        if (v != null &&
            v.Length > 0 &&
            v[3] == 12345)
            throw new Exception("hidden bug!");
    }
}
```

Done. 5 interesting inputs found. How does Pex work?

<table>
<thead>
<tr>
<th>v</th>
<th>Output/Exception</th>
<th>Error Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{0}</td>
<td>IndexOutOfRangeException</td>
<td>Index was outside the bounds of the array.</td>
</tr>
<tr>
<td>{0, 0, 0, 0}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{0, 0, 12345, 0}</td>
<td></td>
<td>hidden bug!</td>
</tr>
<tr>
<td>(0, 0, 12345, 0)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

void CoverMe(int[] a) {
    if (a == null) return;
    if (a.Length > 0) {
        if (a[0] == 1234567890)
            throw new Exception("bug");
    }
}

---

Dynamic Symbolic Execution in Pex

- Generates test data systematically

<table>
<thead>
<tr>
<th>Constraints to solve</th>
<th>Input</th>
<th>Observed constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>a==null</td>
<td>null</td>
<td>a==null</td>
</tr>
<tr>
<td>a!=null</td>
<td>{}</td>
<td>a!=null &amp;&amp; !a.Length&gt;0</td>
</tr>
<tr>
<td>a!=null &amp;&amp; a.Length&gt;0</td>
<td>{}</td>
<td>a==null &amp;&amp; a.Length&gt;0 &amp;&amp; a[0]!=1234567890</td>
</tr>
<tr>
<td>a!=null &amp;&amp; a.Length&gt;0 &amp;&amp; a[0]==1234567890</td>
<td>{123...}</td>
<td>a==null &amp;&amp; a.Length&gt;0 &amp;&amp; a[0]==1234567890</td>
</tr>
</tbody>
</table>

Done: There is no path left.

http://pex4fun.com/HowDoesPexWork
Dynamic Symbolic Execution Summary

- “Ask Pex” sends code to cloud
- Code is compiled and analyzed in cloud
- Dynamic Symbolic Execution automatically finds relevant interesting test inputs that achieve high code coverage
- Results are shown in browser
**Auto-Completion**

Curious? Learn More!

Pex for fun - from Microsoft Research - Windows Internet Explorer

Random Puzzle Learn New

**The code is a puzzle. Do you understand what the code does? Click Ask Pex! to find out.**

```csharp
using System;

public class Program {
    public static void Puzzle() {
        Console.W
```
Research in Software Engineering (RiSE)

RiSE coordinates Microsoft’s Research in Software Engineering in Redmond, USA. Our mission is to advance the state of the art in Software Engineering and Programming Languages, and to bring those advances to Microsoft’s business.

Using video clips on Channel9

http://rise4fun.com, our tools in your browser.
4. SOCIAL EXPERIENCE
Try F# on Facebook

TryFSharp created an event.
F# in Education Workshop, Nov 5, 2010
Wednesday, December 29, 2010 at 9:00am
Cambridge, Massachusetts

1,430 Impressions · 0.28% Feedback

Like · Comment · Share · December 29, 2010 at 2:08p

3 people like this.
This puzzle is an interactive Coding Duel. Can you write code that matches a secret implementation? Other people have already won this Duel 1184 times!

```csharp
using System;

public class Program {
    // Can you fill the puzzle method to match the secret arithmetic operation?
    public static int Puzzle(int x) {
        return 0;
    }
}
```

Ask Pex!  
Done. 2 interesting inputs found. How does Pex work?

Pex found 1 difference between your puzzle method and the secret implementation. Improve your code, so that it matches the other implementation, and 'Ask Pex!' again. You are not signed in. Sign In to track your achievements.

<table>
<thead>
<tr>
<th>x</th>
<th>your result</th>
<th>secret implementation result</th>
<th>Output/Exception</th>
<th>Error Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>Mismatch</td>
<td>Your puzzle method produced the wrong result.</td>
</tr>
</tbody>
</table>
Coding Duels - Fun and Engaging

- Iterative gameplay
- Adaptive
- Personalized
- No cheating
- Clear winning criterion
Social Experience

Pex for fun - from Microsoft Research - Livefeed - Windows Internet Explorer

http://pex4fun.com/Livefeed.aspx

User79033 asked Pex about a puzzle
11 seconds ago

User79033 tried to win C# - «ChallengeArithmetic1»
14 seconds ago

User79033 asked Pex about a puzzle
16 seconds ago

User79032 asked Pex about a puzzle
20 seconds ago

User79029 tried to win C# - «ChallengeDigits2»
32 seconds ago

User79029 tried to win C# - «ChallengeDigits2»
59 seconds ago

User79031 tried to win C# - «ChallengeArithmetic1»
2 minutes ago

User79018 made 17th attempt to win C# - «ChallengeWordReverse»
Social Experience Summary

- Community
- High score lists, leaderboard
- Live feed

http://pex4fun.com/Community.aspx
http://pex4fun.com/Livefeed.aspx
Demo of Pex4Fun
5. SOCIAL ASPECTS IN THE LARGE
MS Products’ (Bing) History is Huge

- Over one million change-sets.
- One change-set can contain 200K check-ins.
  - e.g., branch creations
- +700 branches, 6M different files, 150M different depot paths, 450M different revisions, 150M check-ins, 130M integrations
- Almost impossible to crawl incrementally
  - At least not in 3 months!
Speculative Analysis

- Predict what the user might do
- Do that action in advance, store the results
- If the user chooses to do it,
- Present the pre-calculated information

- Aim: increase developer awareness with precise information
Development History (with BEACON)
BEACON Example
6. SOFTWARE DEVELOPMENT KITS
# Software Development Kits

- **Project Hawaii**
  - On WP7
  - Services execute in the cloud (Azure)
  - OCR, Speech to text etc
  - **WP7 phones loaned to universities worldwide**
  - C#

- **Kinect SDK**
  - Drivers and rich APIs for raw sensor streams and human motion tracking
  - **Kinect unit is $150**
  - Available to academics
  - C++, C#, VB

- **Web-NGram**
  - Content and model types
  - N-gram availability to 5
  - Training size: *All documents indexed by Bing in the en-us market*
  - Updated Periodically

---

Research.microsoft.com/cs

---

Arjmand Samuel, Stewart Tansley, Evelyne Viegas
Web N-Gram in Public Beta

Single Tag Cloud

Multi Tag Cloud

Ref: Dr. Li Ding, Rensselaer Polytechnic Institute
Linked Data

4,500,000,000 triples
180,000,000 data links

Attribution: Richard Cyganiak
Concept Search
<table>
<thead>
<tr>
<th>Year</th>
<th>Movie</th>
<th>Worldwide gross</th>
<th>Budget</th>
<th>Distributor</th>
<th>Director</th>
</tr>
</thead>
<tbody>
<tr>
<td>1977</td>
<td>Star Wars</td>
<td>$782400000</td>
<td>$11000000</td>
<td>20th Century Fox</td>
<td>George Lucas</td>
</tr>
<tr>
<td>1997</td>
<td>Titanic</td>
<td>$1848813795</td>
<td>$200000000</td>
<td>Paramount Pictures</td>
<td>James Cameron</td>
</tr>
<tr>
<td>1993</td>
<td>Jurassic Park</td>
<td>$914691118</td>
<td>$95000000</td>
<td>Universal Studios</td>
<td>Steven Spielberg</td>
</tr>
<tr>
<td>1995</td>
<td>Toy Story</td>
<td>$365000000</td>
<td>$90000000</td>
<td>Walt Disney Pictures</td>
<td>John Lasseter</td>
</tr>
<tr>
<td>1972</td>
<td>The Godfather</td>
<td>$245066411</td>
<td>$60000000</td>
<td>Paramount Pictures</td>
<td>Francis Ford Coppola</td>
</tr>
<tr>
<td>2009</td>
<td>Avatar</td>
<td>$2606954237</td>
<td>$237000000</td>
<td>20th Century Fox</td>
<td>James Cameron</td>
</tr>
<tr>
<td>1975</td>
<td>Jaws</td>
<td>$470600000</td>
<td>$7000000</td>
<td>Universal Studios</td>
<td>Steven Spielberg</td>
</tr>
<tr>
<td>1996</td>
<td>Independence Day</td>
<td>$816969268</td>
<td>$75000000</td>
<td>20th Century Fox</td>
<td>Roland Emmerich</td>
</tr>
<tr>
<td>1998</td>
<td>Armageddon</td>
<td>$553709788</td>
<td>$140000000</td>
<td>Touchstone Pictures</td>
<td>Michael Bay</td>
</tr>
</tbody>
</table>
SEIF Awards

Software Engineering Innovation Foundation (SEIF) with MSR’s-RiSE group supports academic research in software engineering
- Technologies, tools and practices
- Teaching methods

Awards of US$ 25,000 for
- seed-funding larger initiatives,
- proofs of concept, or
- demonstrations of feasibility

In association with the RiSE and Mobile groups at Microsoft Research

- LATAM Awards
  - 2010 – 3 out of 12
  - 2011 – 1 out of 10
  - In open global competition

- Connected to major conferences a workshop and an event

SEIF DAY, July 18, 2012
Conclusion - best practices

- Software engineering is analytic and deals with huge data

- Technology transfer to academia by moving to
  - Browsers
  - SDKs
  - New devices

- Encouraging community through social media