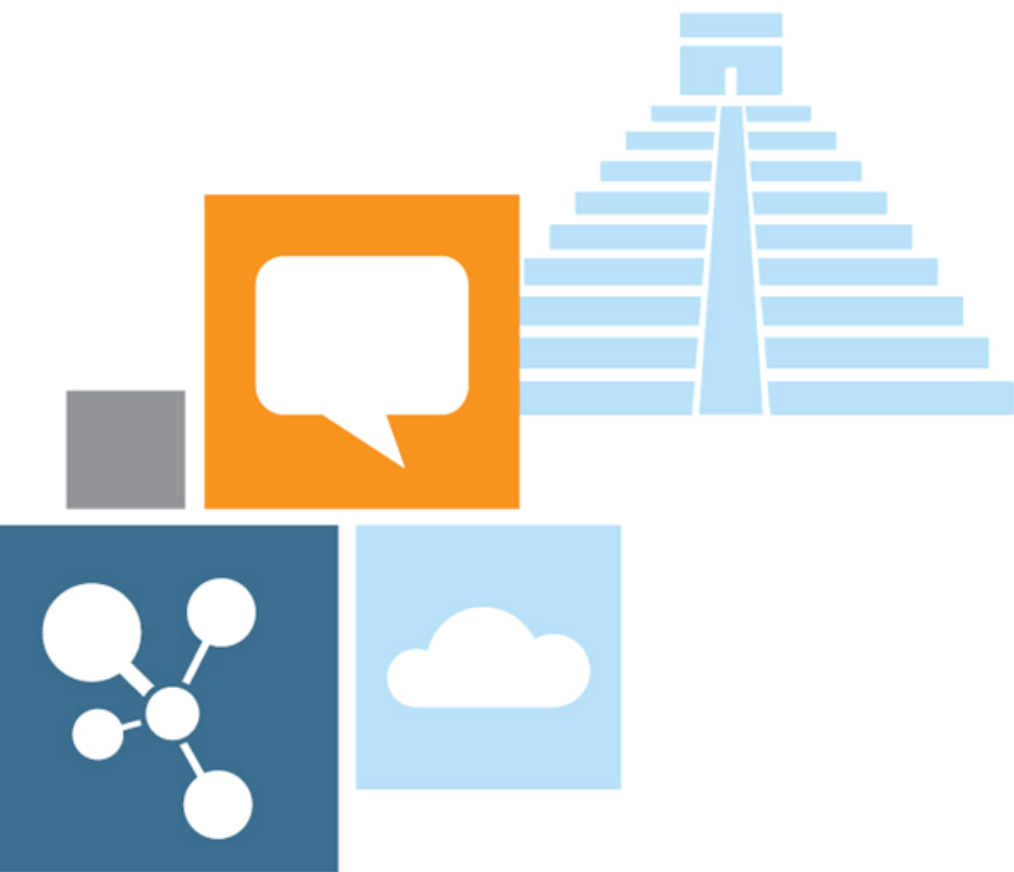


**Microsoft**



Microsoft® Research

# Faculty Summit 2012

Riviera Maya, Mexico | May 23-25 | In partnership with CONACYT

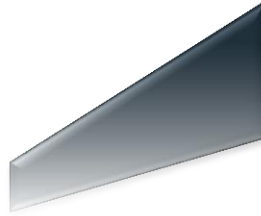


# Bringing theories to life: Computer Science at Microsoft Research

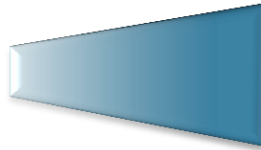
Tony Hey  
Microsoft Research Connections

# Microsoft Research Connections

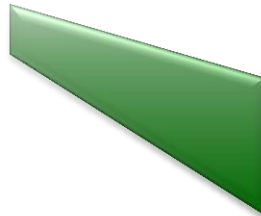
Work with the worldwide academic research community to speed research, improve education, and foster innovation



Collaborations to pursue scientific breakthroughs



Accelerate scientific exploration with computing



Inspire computer and information scientists and engineers



# Engagement and Collaboration Focus

## Core Computer Science



## Natural User Interface



## Earth Energy Environment



## Education and Scholarly Communication



## Health and Wellbeing



### Research Accelerators

- WorldWide Telescope
- Layerscape
- Chronozoom
- .Net Bio
- Chemistry Add-in for Word
- Zentity

### Global Partnerships

- Microsoft Research-INRIA Joint Centre
- Microsoft Research Asia (MSRA) Joint Lab Program
- FAPESP-Microsoft Institute
- Microsoft LACCIR Federation

### External Researchers

- Faculty Fellows
- Graduate Women Scholars
- Jim Gray eScience Award
- Student Internships
- ACM Student Research Competition

# Microsoft Translator Hub



Microsoft  
**Translator Hub**

[Log in using Windows Live ID](#)

Get started by requesting an invitation to build your translation system

[Request an invitation](#)

*Where language meets the world*  
Bridging languages, cultures and technology



Microsoft Translator Hub is helping smaller languages thrive by putting the power to build machine translation systems in the hands of local communities.

- K. David Harrison,  
National Geographic Fellow, author and linguist

## ABOUT MICROSOFT TRANSLATOR HUB

Microsoft Translator Hub empowers businesses and communities to build, improve, and deploy customized automatic language translation systems—bringing better and specialized translation quality to established languages, as well as the many native languages of the world that are not yet supported by major translation providers.

Powered by Windows Azure, Microsoft Translator Hub is an extension of the Microsoft Translator platform and service. You can build a superior translation system easily, within a private website, by combining your translated documents with the power of Microsoft Translator's big data back end. Once you are satisfied with your translation, you may share it publicly on the web.

## MORE INFORMATION

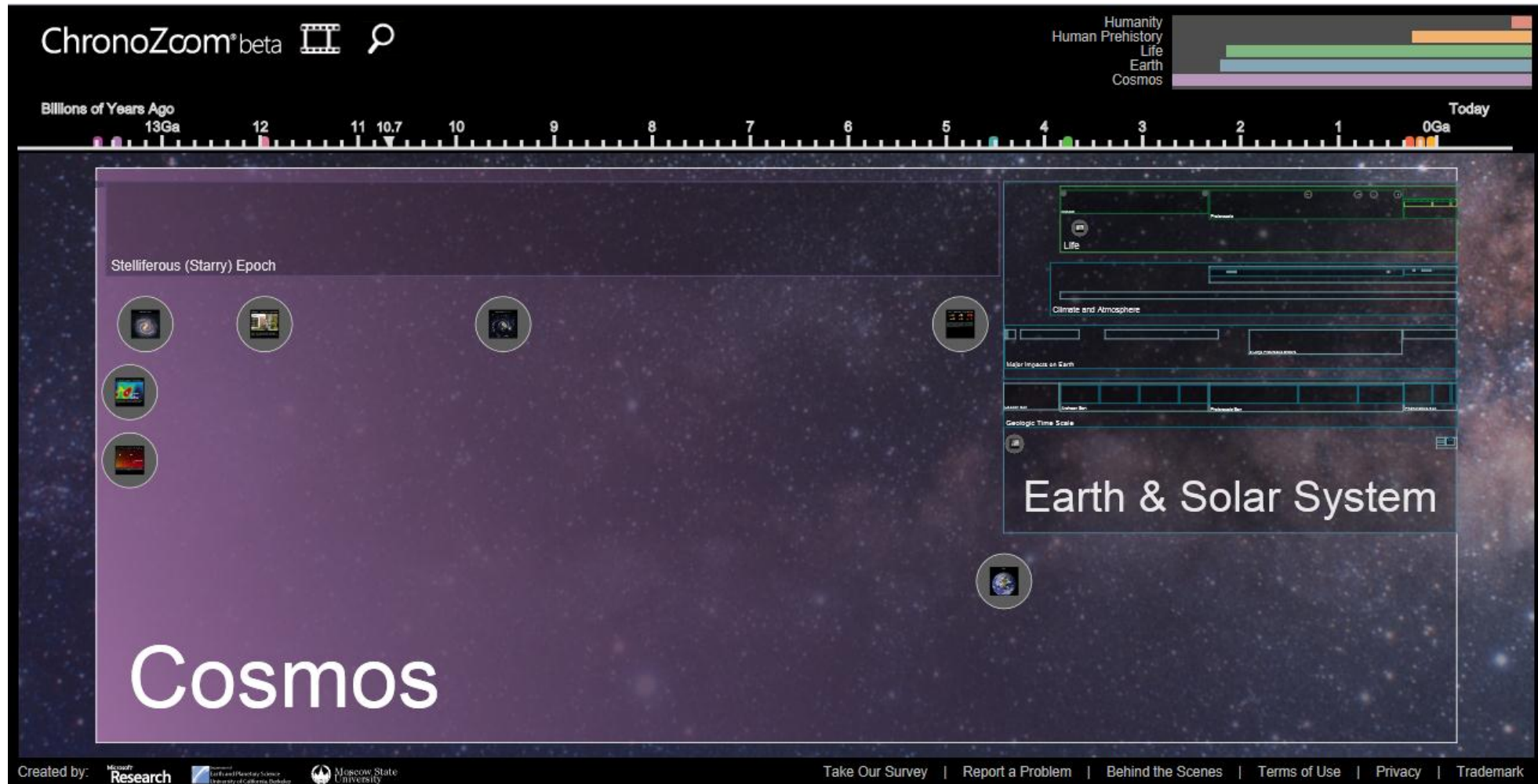
[Microsoft Translator Hub Overview](#)

[Microsoft Translator Hub Forum](#)

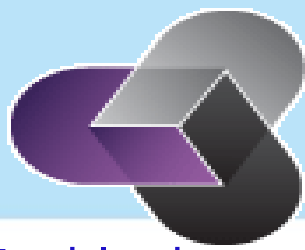


<http://research.microsoft.com/en-us/projects/microsofttranslatorhub/>

# ChronoZoom Demo







**OUTERcurve**  
FOUNDATION

Enable the exchange of code and understanding among software companies and open source communities

## Microsoft Research Connections Research Accelerators Gallery

**Project Trident:** Toolset based on Windows Workflow Foundation that provides scientists' need for a flexible, powerful way to analyze large, diverse datasets.

**Chemistry Add-in for Word:** Chem4Word is an add-in for Microsoft Word that enables semantic authoring of chemical structures.

**ConferenceXP:** Platform for real-time collaboration that seamlessly connects people or groups over a network, providing high-quality, low-latency videoconferencing and a rich set of collaboration capabilities.

**.NET Bio:** This open-source platform features a library of commonly used bioinformatics functions plus applications built upon that framework, and can be extended by using any Microsoft .NET language.

**ChronoZoom:** An open source platform to make time relationships between different studies of history clear and vivid. In the process, it provides a framework for exploring related electronic resources.

# Supporting research through SEIF

## Software Engineering Innovation Foundation

[Research.Microsoft.com/seif](http://Research.Microsoft.com/seif)

\$300,000 in awards each year since 2009

## South American Successes

Diego Garbervetsky, Universidad de Buenos Aires, Argentina, 2010

Sebastian Uchitel, Universidad de Buenos Aires, Argentina, 2010

Karin Breitman, PUC do Rio de Janeiro, Brazil, 2010

Romain Robbes University of Chile, Chile, 2011

## Next call: October 2012







# Microsoft Secrets

- In the 1970's and '80's small teams of highly skilled programmers – 'hackers' – were able to write the first versions of DOS, Word and Excel
  - This 'smart people, small team' approach to writing code became unworkable as PC programs grew into 100's of thousands of lines and then several millions
  - Programming teams also grew into hundreds and then thousands both of developers and testers
- 
- Microsoft developed a 'synch-and-stabilize' approach that allows 'large teams to work like small teams'
  - 'Daily synchronizations through product builds, periodic milestone stabilizations, and continual testing'

**'Microsoft Secrets' by Michael Cusumano and Richard Selby (1998)**



# Size and Scale of Teams and Code

Ship Date	Product	Dev Team Size	Test Team Size	Lines of code (LoC)
July 1993	Windows NT	200	140	4-5 million
December 1999	Windows 2000	1,400	1,700	29+ million
October 2001	Windows XP	1,800	2,200	40 million

Data from 'The Build Master' by Vincent Maraia



# Software Engineering at Microsoft

- A study by NIST in 2002 estimated that software bugs cost the US economy more than \$60B p.a.
  - “Programmers spend far longer fixing bugs in existing code than they do in writing new code”
- 
- Need for software tools to not only find bugs before software is released but also to analyze errors when they are reported
  - “Use software to make software better”



# Software Development Analytics

Then:

40 percent of major decisions are based not on facts, but on the manager's gut.

Accenture Survey of 254 US managers in Industry, 2008

Now:

Empower software development teams to gain and share insight from their data to make better decisions.

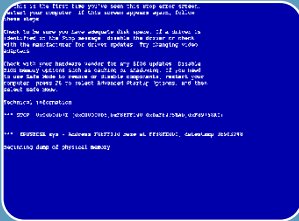
Used in huge code bases.



# Empirical Software Engineering



Socio-technical congruence



Bug reporting and triage



Data-driven software engineering

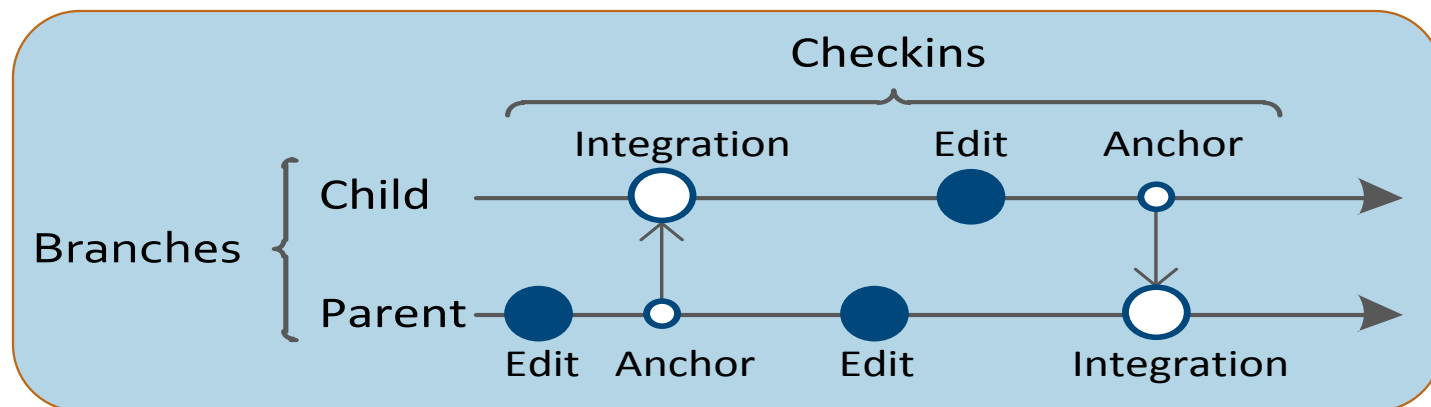
Christian Bird, Nachiappan Nagappan, Brendan Murphy, Thomas Zimmermann. Empirical Software Engineering at Microsoft Research (Showcase Paper). In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2011), Hangzhou, China, March 2011.

# Branching in Source Control Management Systems

Coordinating the work of 100's of developers is difficult

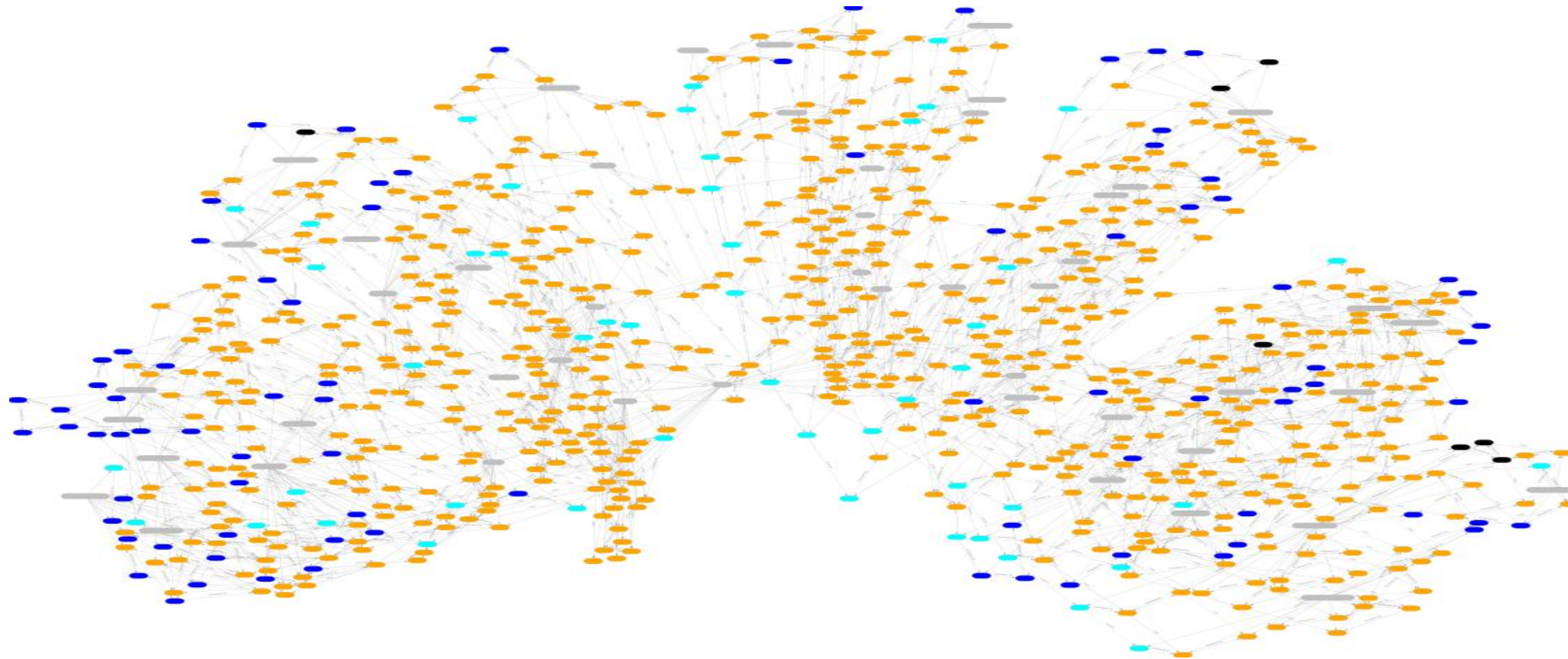
➤ A common solution is to **use branches in SCM** systems

- **Benefits:** Isolating concurrent work during times of instability
- **Cost:** Increase the time that changes percolate through the system (Code Velocity)



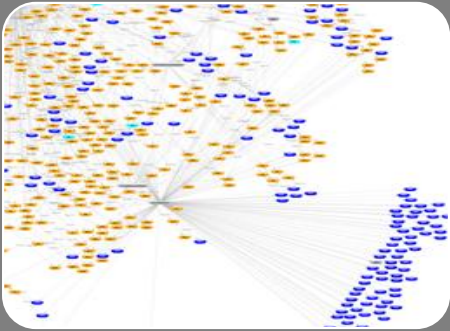


# Code flow for a single file

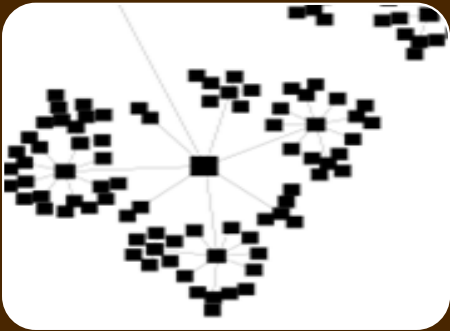


**Code  
Velocity for  
this file is  
particularly  
bad...**

# Branch Analytics (1)

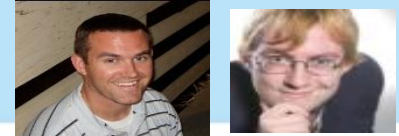


How do we coordinate parallel development?



How do we structure the branch hierarchy? Can we reduce the complexity of branching?

# Branch Analytics (2)



## Techniques:

- **Survey** devs to understand their problems with branching
- **Mine** dev. data for relationship of teams and branches
- **Simulate** benefits and cost of alternative branch structures

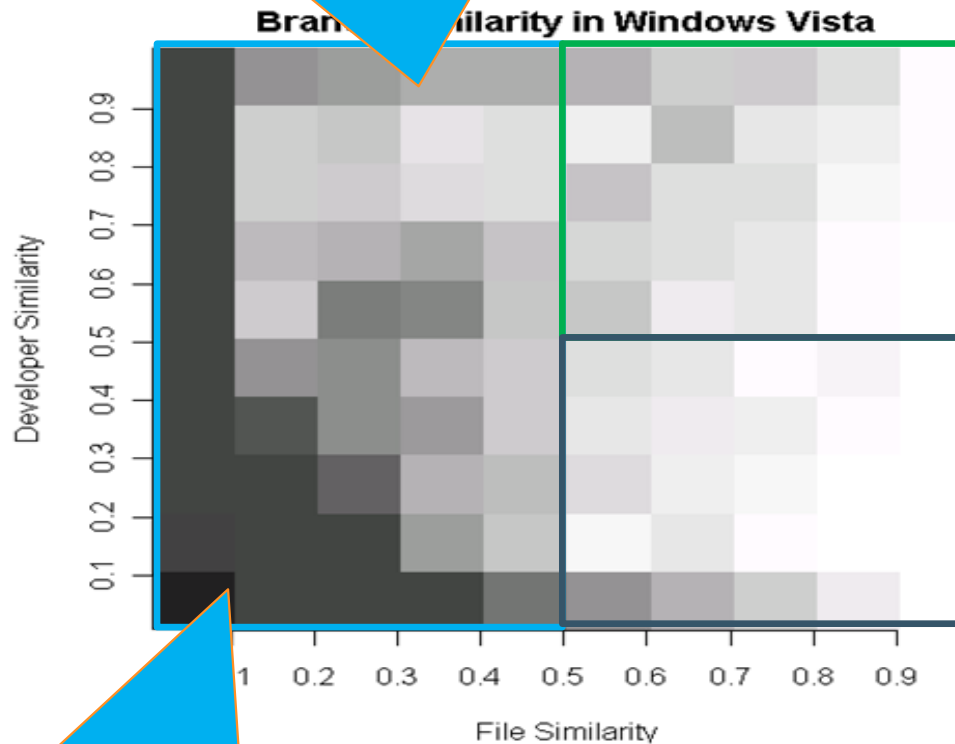
## Actions/Tools:

- **Alert** users about possible conflicts
- **Recommend** branch structure, e.g. del., add, merge etc.
- **Perform** semi-automatic branch refactoring

# Mine "File Similarity" / "Developer Similarity"

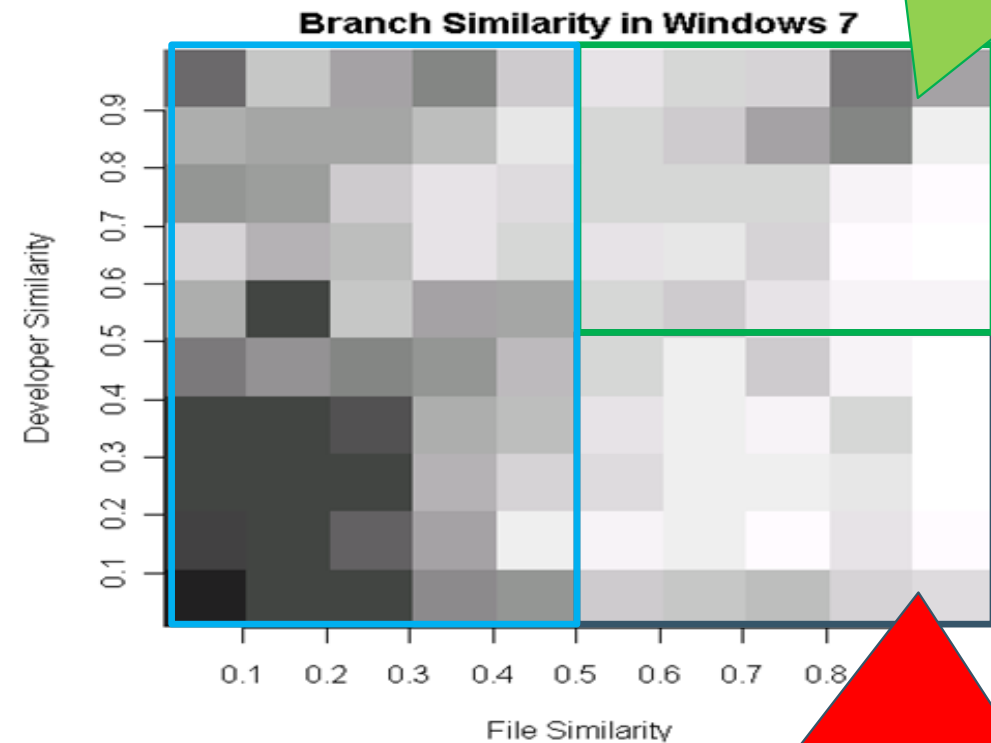
Dark areas mean many branch pairs in that area.

Same devs working on different things is OK



Most pairs of branches are not similar

Same files should mean same people



Same files, but different team means possible problems



# Assessing Branches (1)

Simulate alternate branch structure to assess cost and benefit of individual branches by re-playing Windows history

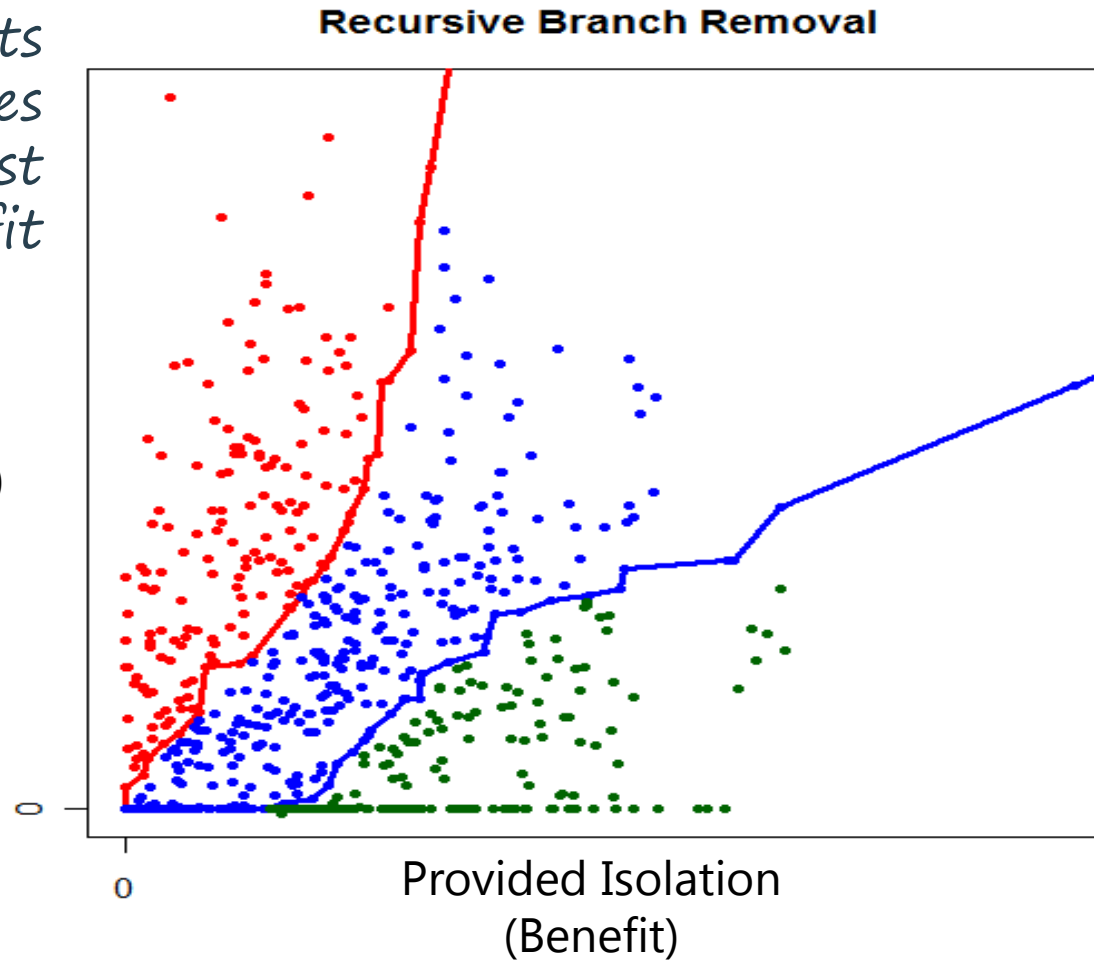
- **Cost: Average Delay Increase per Edit**  
*How much delay does a branch introduce into development?*
- **Cost: Integrations per Edit on a Branch**  
*What is the integration/edit within a branch?*
- **Benefit: Provided Isolation per Edit**  
*How many conflicts does a branch prevent per edit?*

# Assessing Branches (2)

Red dots  
are branches  
with high cost  
but low benefit

Delay  
(Cost)

Each dot  
is a branch



Green dots  
are branches  
with high benefit  
and low cost



# Assessing Branches (3)

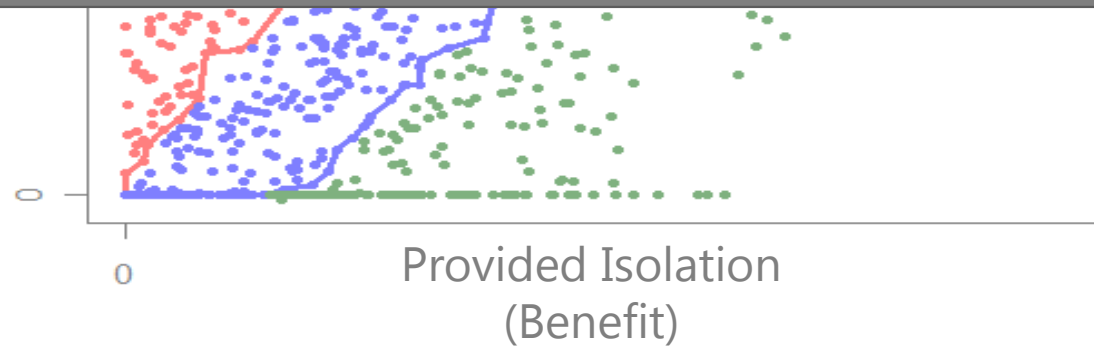
Red dots  
are branches  
with high cost  
and low benefit

Recursive Branch Removal



If high-cost-low-benefit had been removed, changes would each have saved 8.9 days of delay and only introduced 0.04 additional conflicts.

Each dot  
is a branch



Green dots  
are branches  
with high benefit  
and low cost



# Summary: Branch Analytics

Software Analytics makes software development data actionable

- Branch analytics key to improve code velocity
- Better design of development structure
- Efficient scheduling
- Reliable systems with low conflicts

# Improving Bug Triage



## Next stage: the bug gets to a developer

- But it might be the wrong developer, or more skills might be needed
- In Mozilla and Eclipse, between 37%-44% of bug reports are “tossed” to another developer
- Tossing increases time-to-correction

## Microsoft Research teams have been able to reduce tossing by up to 72%

- And prediction accuracy goes up by 23% points compared to traditional approaches
- Uses a graph model based on Markov chains, which capture a bug tossing history and discover team structures

Gaeul Jeong, Sunghun Kim, Thomas Zimmermann. Improving Bug Triage with Bug Tossing Graphs. In (ESEC/FSE 2009), Amsterdam, The Netherlands, August 2009.

# Global Software Servicing

Microsoft's major code is maintained and evolved by programmers who were not the developers

Testers in e.g. China and India lack in-depth knowledge and institutional memory

## Issues:

Strategic (keep projects disjoint architecturally)

Cultural (face-to-face is important)

Inadequate communication (time zones)

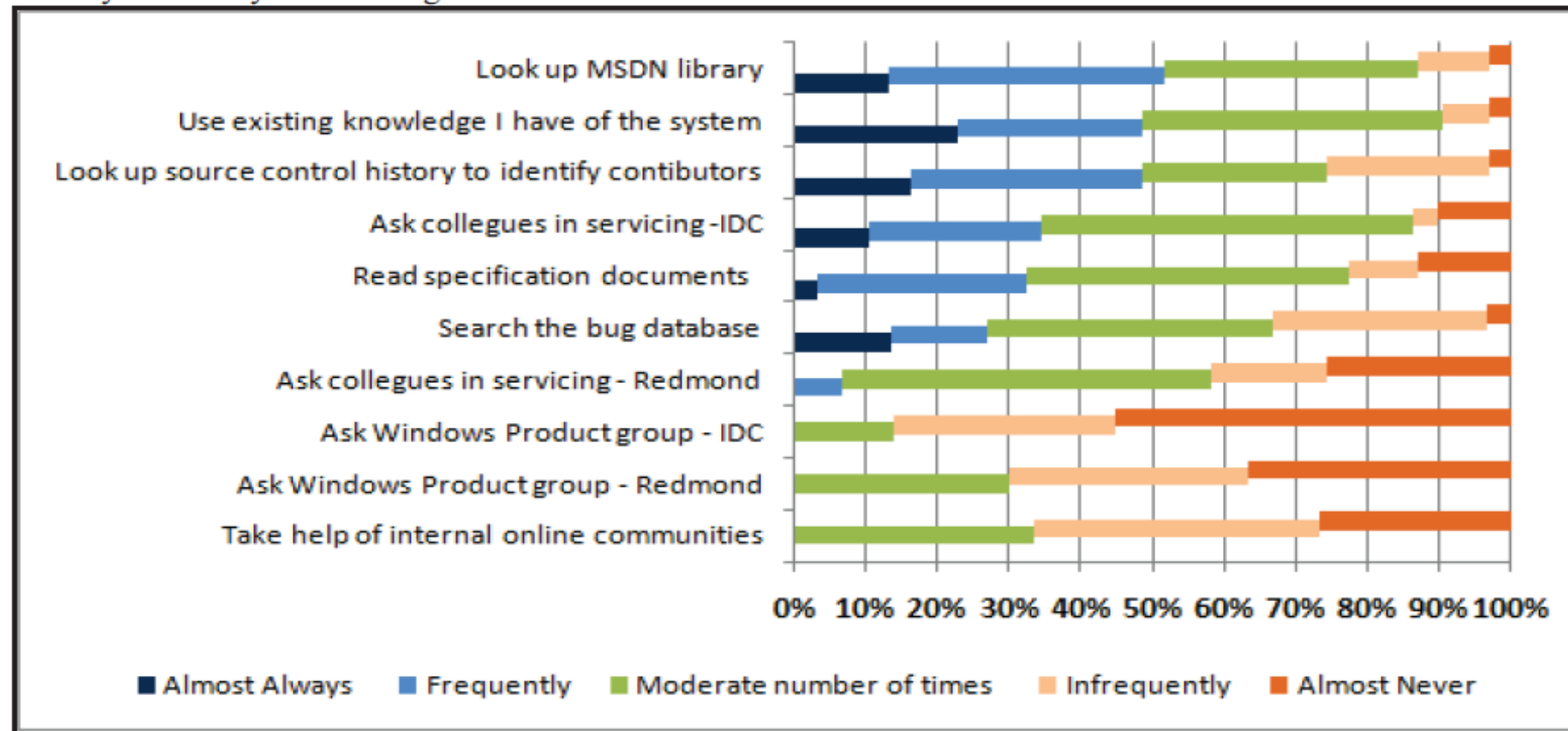
Knowledge management (move people between zones)

Project and process management (synchronized deadlines)

Technical issues (bandwidth)



# Source of satisfying information at MS India



**Global Software Servicing: Observational Experiences at Microsoft**, Shilpa Bugde, Nachi Nagappan, Sriram Rajamani, G.Ramalingam, IEEE International Conference on Global Software Engineering (ICGSE 2008), Bangalore, India.



# From SLAM to SDV: Model Checking in Action

***“Things like even software verification, this has been the Holy Grail of computer science for many decades but now in some very key areas, for example, driver verification we’re building tools that can do actual proof about the software and how it works in order to guarantee the reliability.”***

Bill Gates, April 18, 2002



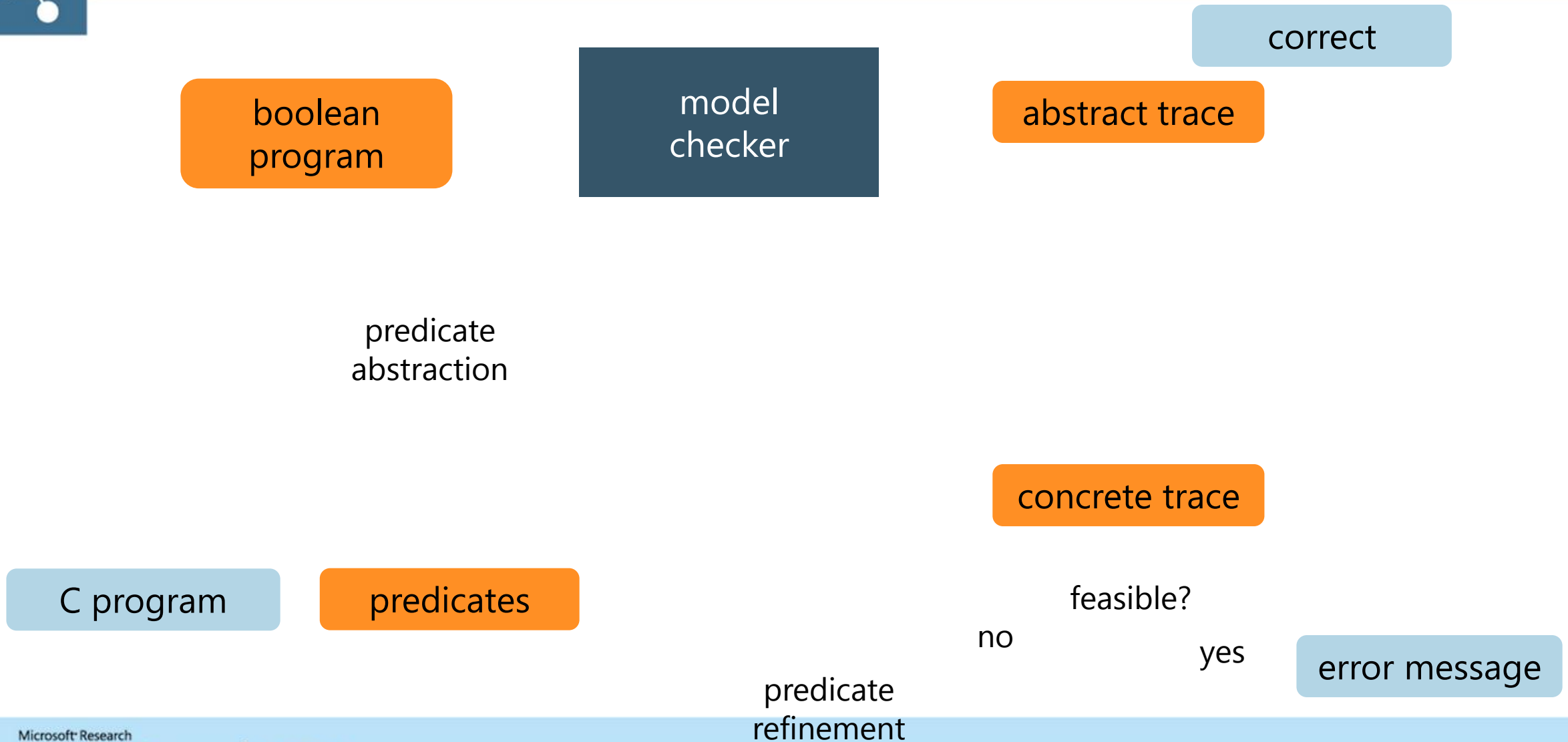


# Important 'niche' application: Device Drivers

- Bugs can have severe effects
- Relatively small code base (usually  $\leq 50$  KLOC)
- Crash-freedom specified by finite-state API protocols
- Correctness depends mostly on flow of control, not data

Acquire()  
Release()  
Release() Acquire()

# Predicate abstraction and refinement



- [illegible]



# Z3

# SMT Theorem Prover

Z3 is a high-performance theorem prover developed by the RiSE group in Microsoft Research. It is freely available for academic research:

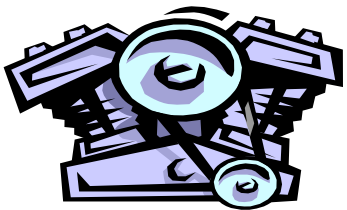
<http://research.microsoft.com/projects/z3>

DPLL

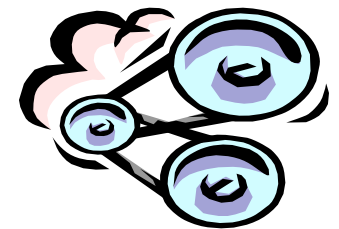
Simplex

Rewriting

Superposition



**Z3 is a collection of  
Symbolic Reasoning Engines**



Congruence  
Closure

Groebner  
Basis

$\exists\forall$   
elimination

Euclidean  
Solver

# Impact of Z3

Z3 is used by many research groups (> 700 citations)

More than 17,000 downloads

Z3 placed 1<sup>st</sup> in 17/21 categories in the 2011 competition

**SAGE**



Securu

Z3 used to check  
**Azure Firewall Policies**

Modeling Foundations.

**SLAM**

**TERMINATOR**

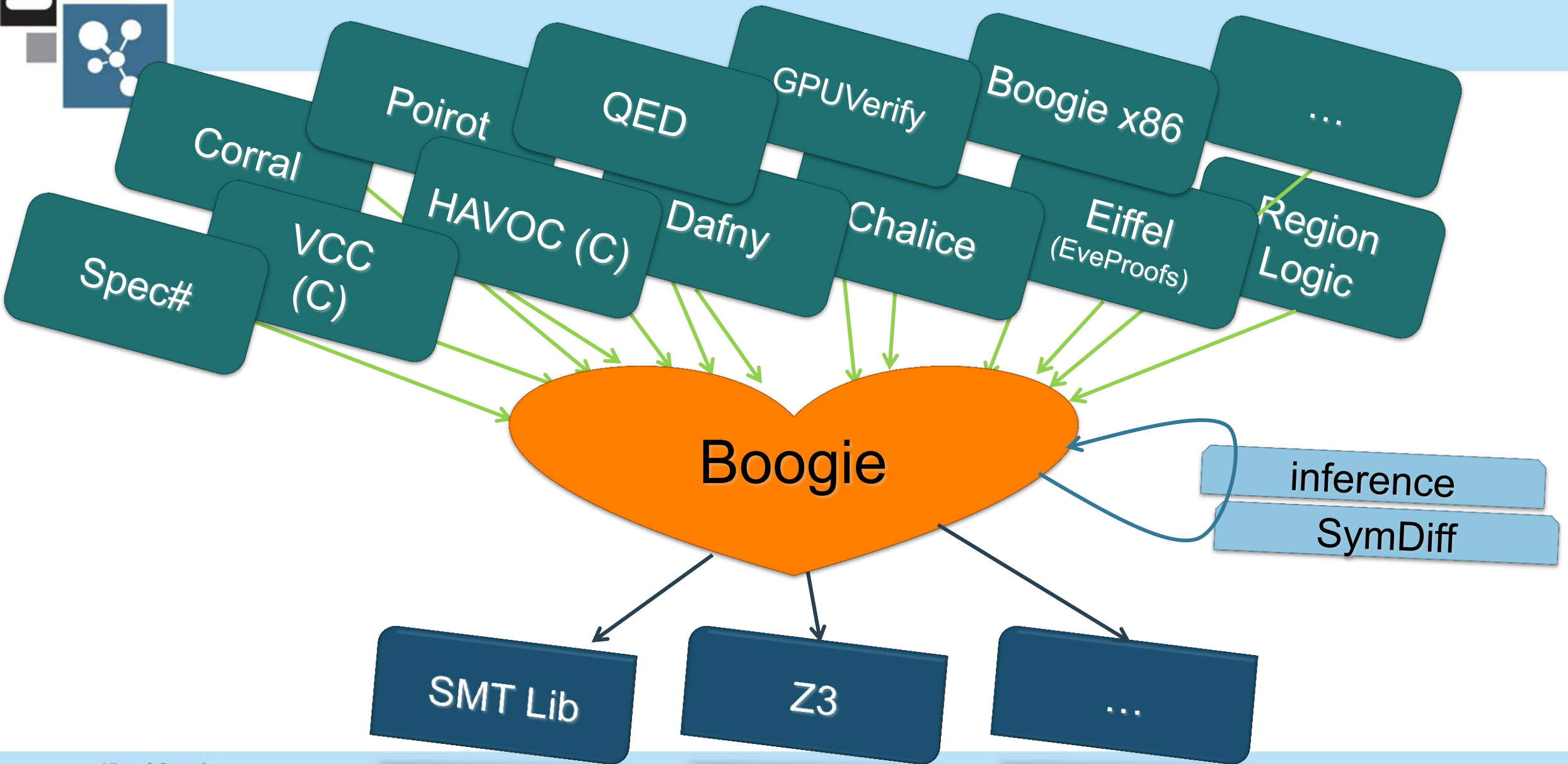
Z3 solved more than **3 billion**  
constraints created by SAGE!  
**Checking Win8 and Office.**

Z3 **ships** with the  
Static Driver Verifier

Microsoft | Virtualization

Microsoft

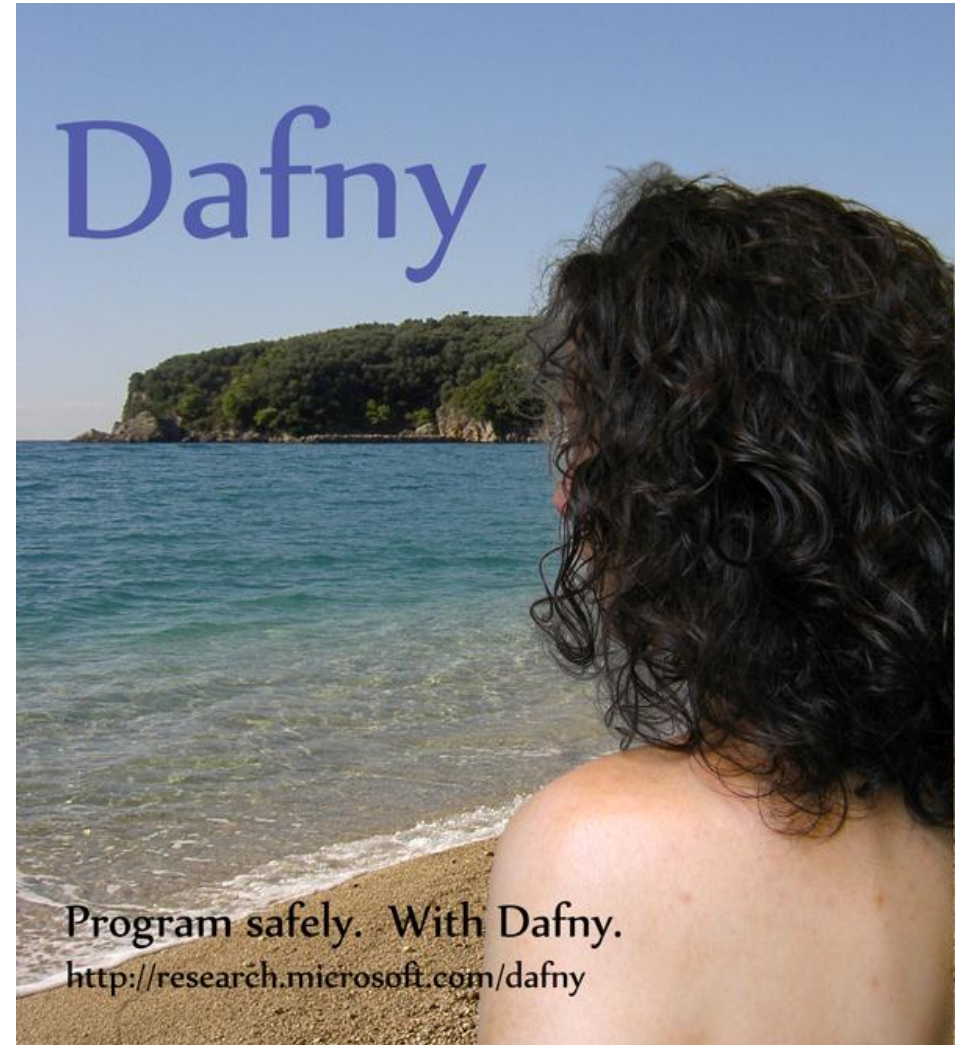
# Verification Architecture



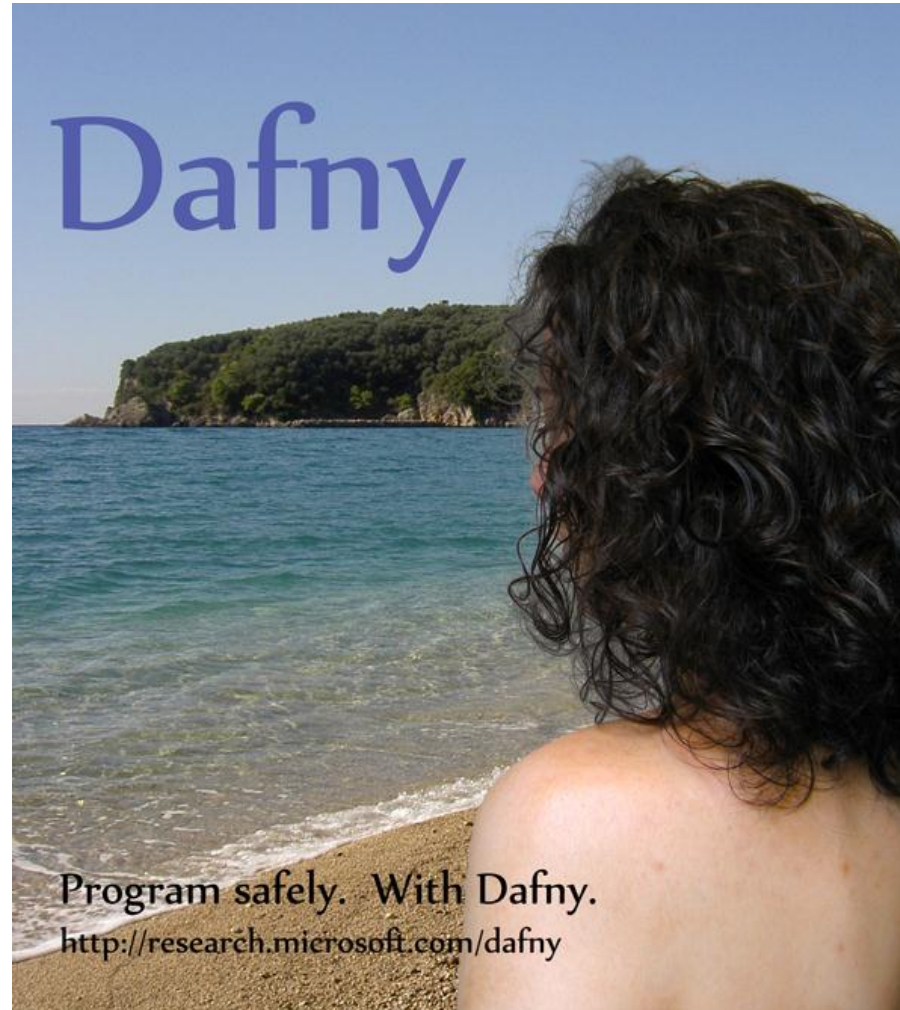


# About Dafny

- Dafny is programming language with a program verifier
- Pushes state-of-the-art
- Used in teaching
- >100K programs submitted for verification on rise4fun.com
- Used by two medalist teams in VSTTE 2012 program verification competition



# Dafny Demo





# Thanks to our featured Microsoft Researchers



Christian  
Bird



Tom  
Zimmermann



Nachi  
Nagappan



Brendan  
Murphy



Sriram  
Rajamani



Rustan  
Leino



Wolfram  
Schulte



Tom  
Ball



# Summary: Future of Software Engineering

Software Analytics enables data-driven decision  
which process, practice, tool to use and  
deploy under which context

Logic based tools help develop better software artifacts  
help model, analyze, optimize, and  
synthesize software artifacts

Future platforms excite and pose new challenges  
web, mobile devices (phone, tablet),  
datacenter, games



# Some Resources

- Microsoft Research
  - [\*\*http://research.microsoft.com\*\*](http://research.microsoft.com)
- Microsoft Research downloads:
  - [\*\*http://research.microsoft.com/research/downloads\*\*](http://research.microsoft.com/research/downloads)
- Microsoft Research Connections
  - [\*\*http://research.microsoft.com/en-us/collaboration/\*\*](http://research.microsoft.com/en-us/collaboration/)
- Research in Software Engineering (RiSE) Group
  - [\*\*http://research.microsoft.com/en-us/groups/rise/default.aspx\*\*](http://research.microsoft.com/en-us/groups/rise/default.aspx)
- Scholarly Communications
  - [\*\*http://www.microsoft.com/scholarlycomm\*\*](http://www.microsoft.com/scholarlycomm)
- Outercurve Foundation
  - [\*\*http://www.outercurve.org/\*\*](http://www.outercurve.org/)
- Tony Hey on eScience
  - <http://tonyhey.net/>