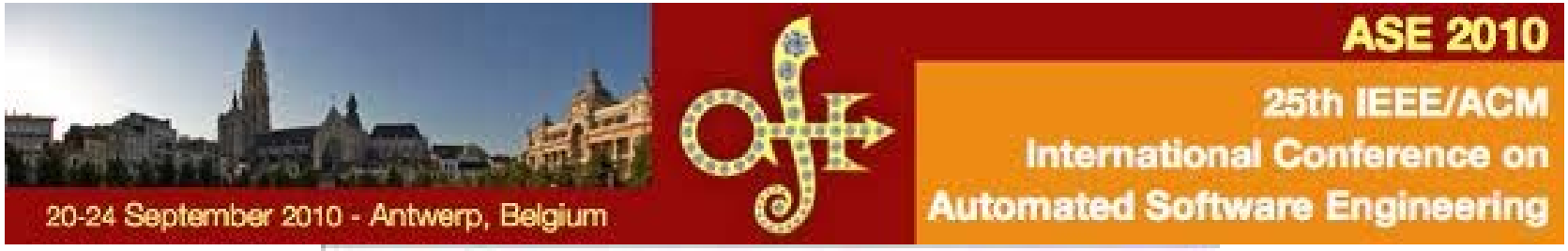
Automated Software Engineering Research Group @NCSU

Cooperative Testing and Analysis: Human-Tool, Tool-Tool, and Human-Human Cooperations to Get Work Done

Tao Xie
Peking University, China (2011-2012)
North Carolina State University, USA

Supported in part by NSF CAREER 0845272 and Microsoft Research SEIF 2011

Automation in Software Engineering



Remember that "Automation" is key to ASE -- authors should have clearly indicated how their work contributes, either directly or indirectly, to the automation of software engineering.

Automated Software Engineering Research Group @NCSU

ASE

Automation in Software Testing

ACM SIGSOFT International Symposium on Software Testing and Analysis

Dagstuhl Seminar 10111

Practical Software Testing: **Tool Automation** and **Human Factors**

Example: Automating Test Generation

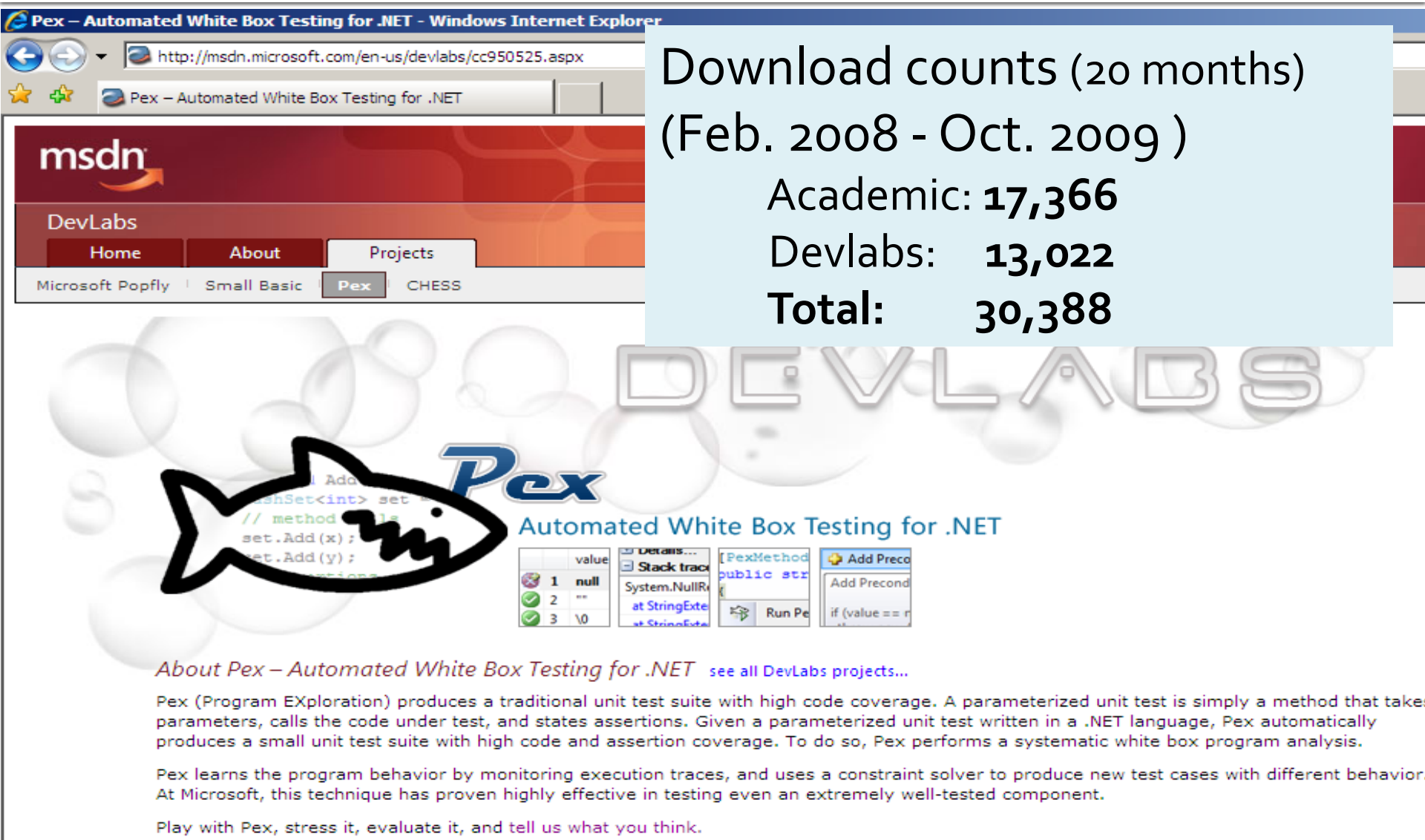
@NCSU ASE

- Method sequences
  - MSeqGen/Seeker [Thummalapenta et al. OOSPLA 11, ESEC/FSE 09], Covana [Xiao et al. ICSE 2011], OCAT [Jaygarl et al. ISSTA 10], Evacon [Inkumsah et al. ASE 08], Symclat [d'Amorim et al. ASE 06]
- Environments e.g., db, file systems, network, …
  - DBApp Testing [Taneja et al. ESEC/FSE 11], [Pan et al. ASE 11]
  - CloudApp Testing [Zhang et al. IEEE Soft 12]
- Loops
  - **Fitnex [Xie et al. DSN 09]**
- Code evolution
  - eXpress [Taneja et al. ISSTA 11]

Major work done in collaboration with MSR Pex team

Pex on MSDN DevLabs

Incubation Project for Visual Studio



Download counts (20 months)
(Feb. 2008 - Oct. 2009)

Academic: **17,366**

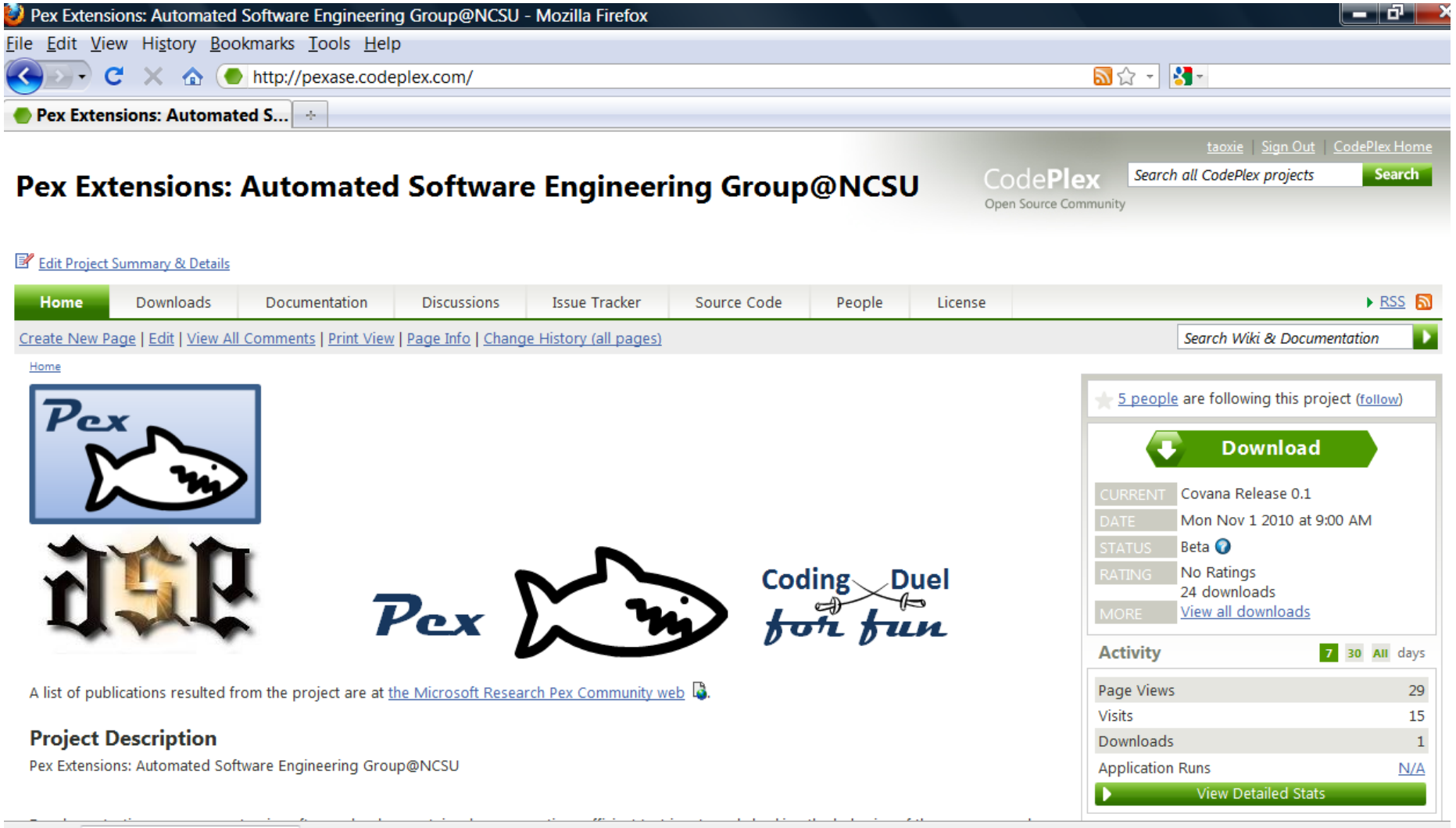
Devlabs: **13,022**

Total: 30,388

Open Source Pex extensions

http://pexase.codeplex.com/

Publications: http://research.microsoft.com/en-us/projects/pex/community.aspx#publications



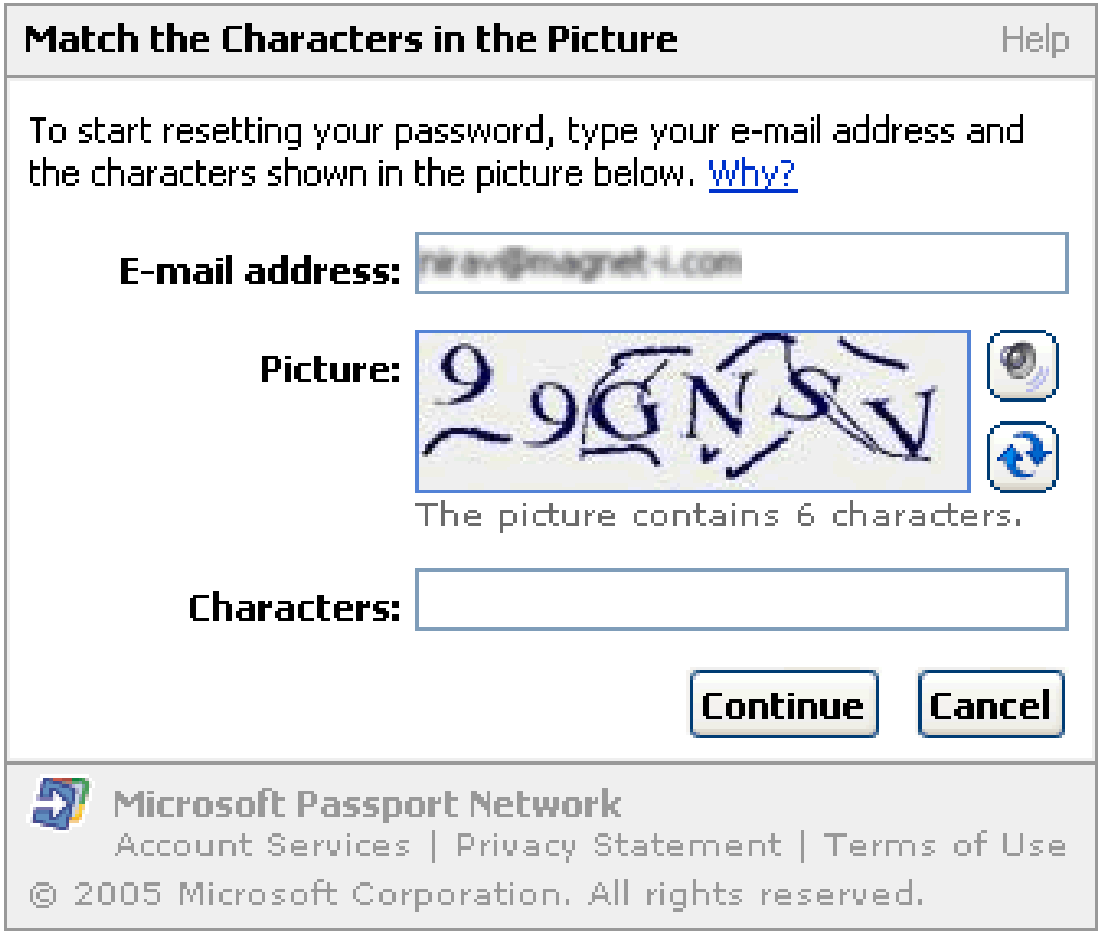
Reality Check

- **Machine is better at task set A**
  - Mechanical, tedious, repetitive tasks, …
  - Ex. solving constraints along a long path
- **Human is better at task set B**
  - Intelligence, human intention, abstraction, domain knowledge, …
  - Ex. local reasoning after a loop

= A U B?

CAPTCHA

"**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part"



Human Uses Machine

- 50 years of automated debugging research
  - N papers → only 5 evaluated with actual programmers

> "Programmers have been waiting a long time for usable automated debugging tools, and we have already gone a long way from the early days of debugging. We believe that, to further advance the state of the art in this area, we must steer research towards more promising directions that take into account the way programmers actually debug in real scenarios."

[ISSTA11 Parnin&Orso]

Automation in Software Testing

ISSTA 2011

ACM SIGSOFT International Symposium on Software Testing and Analysis

Dagstuhl Seminar 10111

Practical Software Testing: **Tool Automation** and **Human Factors**

Human Factors

Human Factors

Dagstuhl Seminar 10111

Practical Software Testing: **Tool Automation** and **Human Factors**

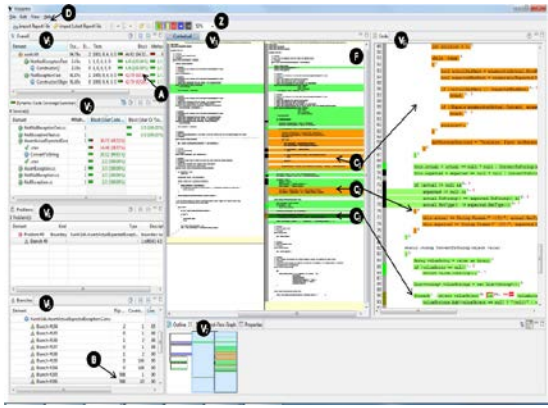
Human-Centric Computing in Software Engineering

The IEEE Symposium on *Visual Languages and Human-Centric Computing* (VL/HCC)

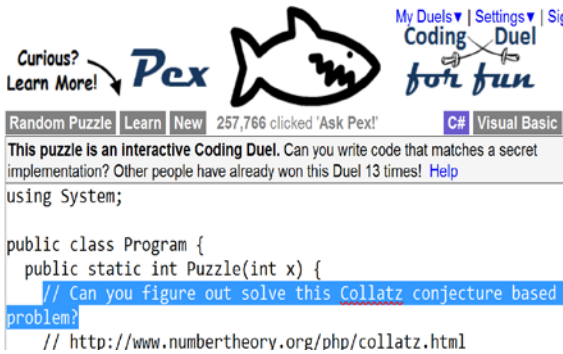
ACM Symposium on SOFTWARE VISUALIZATION
http://www.softvis.org

Cooperation Between Human and Machine

- **Computing**-Centric **Human**
  - Driver: tool←→ Helper: human
  - Ex. Covana [Xiao et al. ICSE 2011]



- **Human**-Centric **Computing**
  - Driver: human ←→ Helper: tool
  - Ex. Coding duels @Pex for Fun

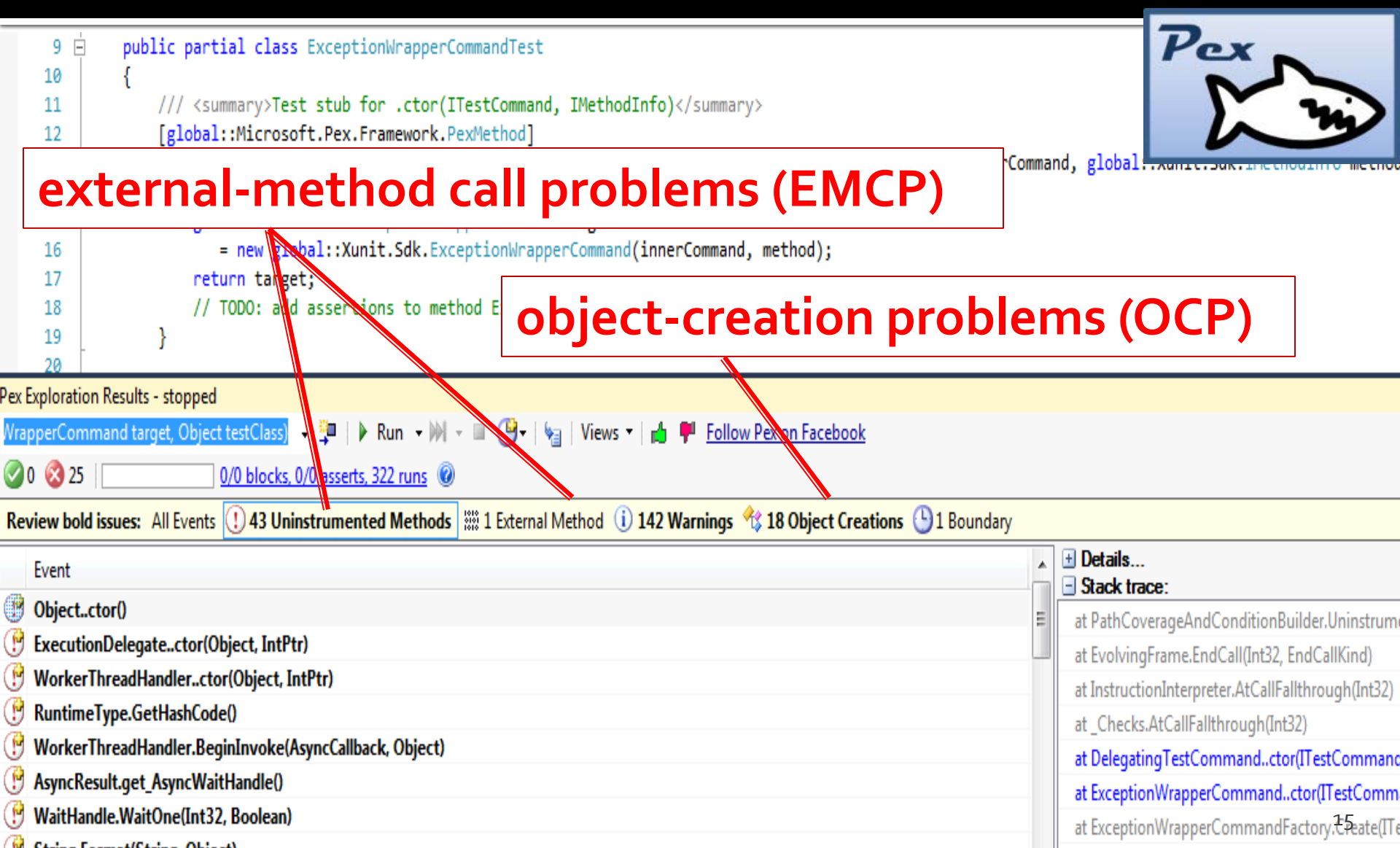


***Interfaces** are important. **Contents** are important too!*

Computing-Centric Human

- Motivation
  - Tools are often not powerful enough (at least for now)
  - Human is good at some aspects that tools are not
- **Task for Tool**: What needs to automate?
- Tool → Human
  - What difficulties does the tool face?
  - How to communicate info to the user to get her help?
- Tool ←Human
  - How does the user help the tool based on the info?
- Iterations to form feedback loop?

Problems Faced by Automated-Test-Generation Tool



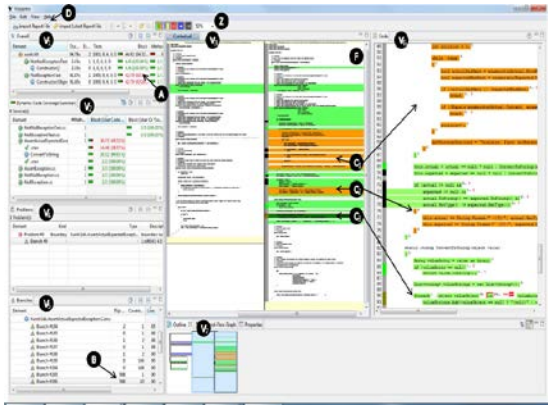
Cooperation Between Human and Machine – Covana

[Xiao et al. ICSE 2011]

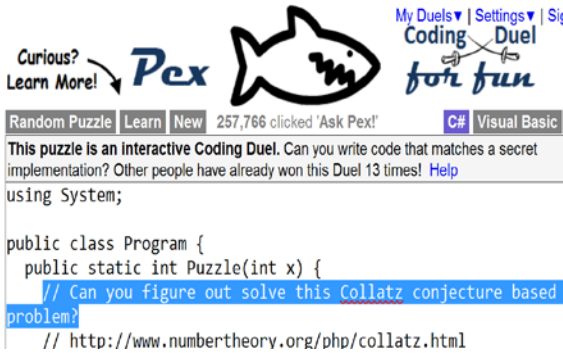
- **Task:** What need to automate?
  - Test-input generation
- What difficulties does the tool face?
  - Doesn't know which methods to instrument and explore
  - Doesn't know how to generate effective method sequences
- How to communicate info to the user to get her help?
  - Report encountered problems
- How does the user help the tool based on the info?
  - Instruct which external methods to instrument/write mock objects
  - Write factory methods for generating objects
- Iterations to form feedback loop?
  - Yes, till the user is happy with coverage or impatient

Cooperation Between Human and Machine

- **Computing**-Centric **Human**
  - Driver: computer ←→ Helper: human
  - Ex. Covana [Xiao et al. ICSE 2011]



- **Human**-Centric **Computing**
  - Driver: human ←→ Helper: computer
  - Ex. Coding duels @Pex for Fun



***Interfaces** are important. **Contents** are important too!*

Behind the Scene of Pex for Fun

[ASE 08sp]

Secret Impl $\overset{\text{behavior}}{==}$ Player Impl?

Secret Implementation

```
class Secret {
    public static int Puzzle(int x) {
        return x * 3 + 10;
    }
}
```

Player Implementation

```
class Player {
    public static int Puzzle(int x) {
        return x;
    }
}
```


```
class Test {
    public static void Driver(int x) {
        if (Secret.Puzzle(x) != Player.Puzzle(x))
            throw new Exception("Found a Difference");
    }
}
```

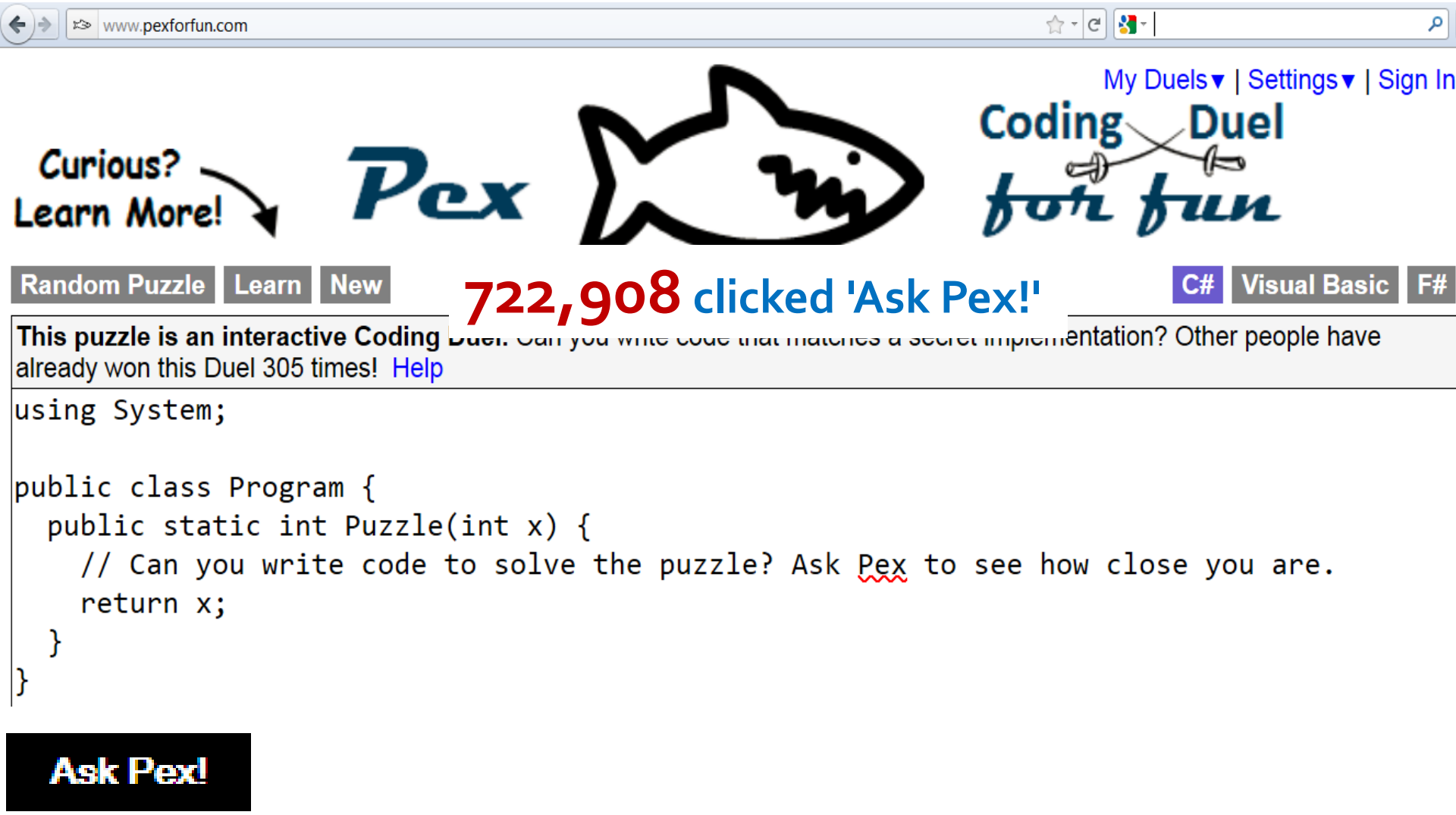
Pex

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.

	x	y	your result	secret implementation result	Output/Exception	Error Message
✗	0	0	2	22	Mismatch	Your puzzle method produced the wrong result.
✓	-1458398958	515739696	1378169382	1378169382		

Migrating Pex to the Web/Cloud

Try it at http://www.pexforfun.com/



HCC: Pex for Fun

- Coding duels at http://www.pexforfun.com/
- **Task** for Human: write behavior-equiv code

```
using System;
public class Program {
  public static int Puzzle(int x, int y) {
    /* Could you re-order the statements t
of the secret implementation? */
    y = x * 10;
    y = x;
    x = y + 2;
    return (x + y);
  }
}
```

Ask Pex!

- Human → Tool
  - Does my new code behave differently? How exactly?

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.

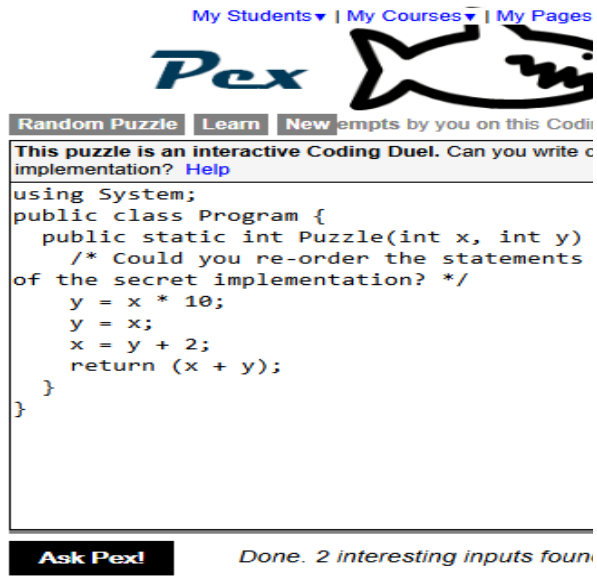
	x	y	your result	secret implementation result	Output/Exception	Error Message
✗	0	0	2	22	Mismatch	Your puzzle method produced the wrong result.
✓	-1458398958	515739696	1378169382	1378169382		

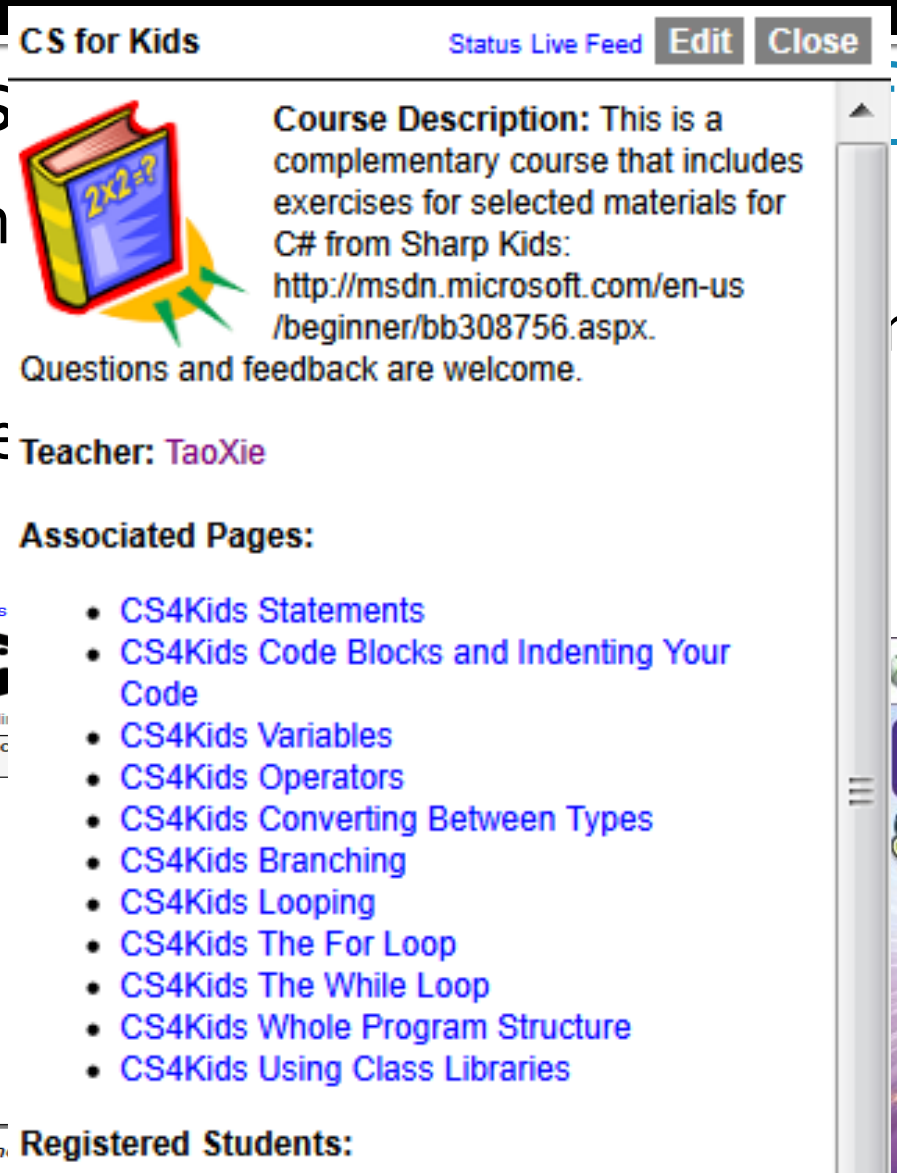
- Human ← Tool
  - Could you fix your code to handle **failed/passed tests**?

- Iterations to form feedback loop?
  - Yes, till tool generates no failed tests/player is impatient

Human-Centric Computing

- Coding duels [un.com/](un.com/)
  - Brain exercisin
  - Fun: iterative, n criterion
  - Abstraction/ge blem solving





Brain exercising

Coding Duels Go Wild @ICSE 11

«ICSELevel2Challenge10» - Can you fill the puzzle method to match the secret computation? - *you already made 13 attempts*

Coding Duels	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22	#23	#24	#25	#26	#27	#28	#29
TaoXie	3	3	2	3	3	3	5	12	6	2	20	4		13	3	2	24													
Felienne	3	7	4	4	8	10	12	17	25	10	20	7																		
RPortoAbreu	4	3	4	8	12	3	33	40	31	17	112	98	70	161	2	4	7	8	13	3	3	2	11	8	21	10	6	17	23	63
jomaras	4	2	3	4	5	9	8	10	40	8	14	6	3	17	4	2	4	9	10	3	6	2		34		8	3			
chenfucn	3	2	5	5	7	6	2	1	3		18			1																
cdragert	4	6	4	26	8	3	21	24	15	22	4	18	8	180	10	2	50	3	3	2	5	2	5	4	8	16	7	9	10	21
jamshaidm	4																													
JMacFan	2	2	2	18	12	6	18																							
Meilies	11	2	4	5	13	14	18	17	9	1																				
shauvik	2	5	2	3	5	11	6																							
almsantos	4	2	4	3	5	10	4	1		3	2	2	1	5	7	2	22	5	3	5	5	3	15	15	21	8	12		22	1
SCBSUFPE	4	5	3	4	7	7	7	21	18	3	5			14	4	2	5	6	5	3	4	3	9	15	3	11	11	15	19	7
theofour	2	2	2	4	6	12	14	4	7	14	14	37	5	58	2	2	8	6	9	8	2	2	5	4	6	7	4	7	4	12
anchi	6	4	3	4	12	15	6	15	21	7	15	28	2	7	4	2	14	5	7	4	2	3	7	3	10	6	13	17	12	62
malteres	2	5	2	3	6	5	23	18	36	10	17	36	26	224	2	4	12	5	9	2	4	8	9	2	18	15	3	17	29	79
rla4	4	2	2	4	6	7	10	22	11		9	4	12	12	7	2	11	6	8	6	2	1								
krw7c	7	6	8	3	7	7																								
MIKAND	2	2	1	5	8	5	9	7	17	5	9	7	2	9	4	2	4	4	5				5	2	5	5	5	4	13	10
ariboira	6	7	6	14	13	22	26	51	56	11	147	96	61	226	23	3	16	13	21	7	3	11	21	3	7	2	21	26	27	22
Benny	4	5	5	4	13	6	10	13	22	32	53	31	30	9	2	2	7	5	9	2	2	4	7	8	4	14	22	21	11	29
TheRama	3	2	2	3	3	18	4	8	29	7	17	10	3	37	3	2	12	7	6	2	6	2	10	5	3	11	5	22	8	12
madking	2	2	2	5	11	2	8	6	13	15	8	10	4	386	4	2	10	6	6	2	2	2	5	7	4	7	5	3	2	15
nipun	19	9	5	10	2	2	12	12	2	12				8	3	3	10	6					8	4	15				24	
(no nickname)	3	7	2	3																										
eijiadachi	6	3	2	4	7	13	17	11	9	2	3	3	24	1	3	2	3	4	12	8	3	4	6	25	8	16	6	18	134	17
schroeter	3	2	3	6	3	3	6	6	18	4	12																			
Ezzo	3	1	1																											
(no nickname)	3	4	4	11																										

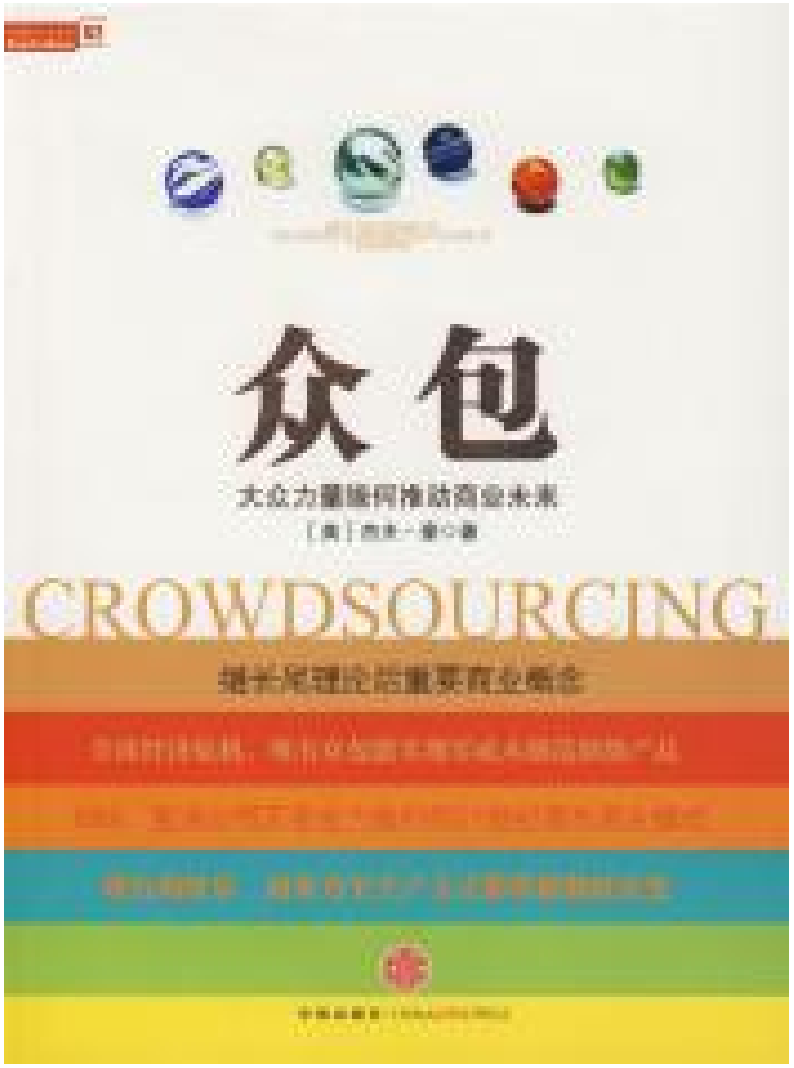
#10
20
20
112
14
18
4
2
5
14
15
17
9
9
147
53
17
8

This is your last attempt

```
using System;

public class Program {
  public static int Puzzle(int x) {
        if (x <= 0) return 0;
    if (x == 1) return 0;//1
    if (x == 2) return 0;//2
    if (x == 3) return 0;//3
    if (x == 4) return 0;//5
    if (x == 5) return 4;//7
    if (x == 6) return 4;//13
    if (x == 7) return 4;//21
    if (x == 8) return 0;//34
    if (x == 9) return 0;//55
    if (x == 10) return 0;//89
    if (x == 11) return 0;//144
    if (x == 12) return 0;//233
    if (x == 72) return (x-8);

    if (x == 963) return (x-3);
    if (x == 964) return (x-4);
    if (x == 965) return (x-1);
    if (x == 966) return (x-2);

    if (x == 995) return (x-3);
    if (x == 996) return (x-4);
    if (x == 997) return (x-1);
    if (x == 998) return (x-2);
    if (x == 999) return (x-3);
    return 0;
  }
```

Data-Driven Software Engineering in the Large **AND Small**

Human-Human Cooperation



百度众测 Baidu CrowdTesting

http://test.baidu.com/

百度众测 Baidu CrowdTesting

http://test.baidu.com/

百度众测 Baidu CrowdTesting

http://test.baidu.com/

活动介绍：

参加狂测行动微风级活动，完成活动中的项目，会根据狂测达人所提交的bug等级给予平台积分，达到一定等级积分，给予众测平台等级头衔，同时得到丰富的百度奖品。

奖励方式：

参与众测平台微风级项目活动的达人们，可以根据提交bug等级得到平台积分，积分达到相应等级，百度礼品相送。

奖品设定：



百度浏览器随意测
Baidu Browser Causal Testing

http://test.baidu.com/

项目名称	百度浏览器随意测v1
项目类型	测试类型
项目内容	使用百度浏览器的各个功能，进行各种随机测试，找出bug。主要测试点有： 1、百度浏览器窗口操作（拖动、多窗口、弹出窗口等） 2、百度浏览器快捷键操作 3、百度浏览器页面布局
项目验收标准	1、下载附件中的百度浏览器版本 2、参考测试指南，按testcase的测试方向进行测试 3、进行各种随机操作和测试 4、对于发现的bug，填写详细的bug复现步骤，若有截图最好
项目指南	1、请在testcase完成操作标记 2、发现bug，请在bug区，按照要求提交 3、其它测试指南，请参看附件“百度浏览器测试指南.docx”
测试环境	浏览器： 百度浏览器
项目开始日期	2011-09-28
项目结束日期	2011-10-04
参与人数	10000
附件地址	百度浏览器测试指南(2).docx bdbrowser_setup-33.exe

The Concept: Crowd Sourced Formal Verification

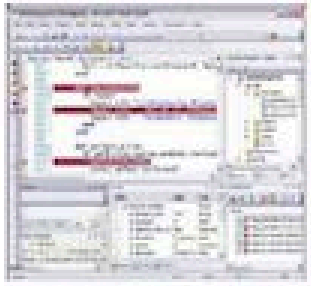
"Game-ify" Geeky Formal Verification

Applies game solutions to the original formal verification problem

Exploits a large user base requiring no formal verification expertise

Code

Model



Verified Code

Verified Model

Source: University of Washington

CSFV New Capabilities

Tool-Tool Cooperation

- Static analysis + dynamic analysis
  - Code Contract Static Checker + Pex [ICSE 10 demo]
- Dynamic analysis + static analysis
  - Fix generation + fix validation [ongoing work w/ PKU]
- Static analysis + static analysis
  - …
- Dynamic analysis + dynamic analysis
  - Evacon [ASE 08]: branch ranking metric

Branch Ranking

Evacon [ASE 08]

- What does branch coverage of two tools: 85% > 75% tell?
  - Tool with 75% may be better at covering those difficult-to-cover branches when used in tool combination
- Need take into account difficulties of branches being covered (esp. using tools in combination)
- Proposed metric: #branches categorized into:

  Branch-1: covered by only 1 tool under comparison

  …

  Branch-n: covered by only n tools under comparison

Covering more branches in Branch-1 means uniquely covering more branches not being covered by the other tools under comparison

Branch Ranking Example

Evacon [ASE 08]

Branch rank	Evacon-A	Evacon-B	eToc	jCUTE	JUnit Fact	Randoop
1	5/13	0/13	2/13	2/13	2/13	2/13
2	5/17	1/17	2/17	7/17	13/17	6/17
3	13/16	7/16	3/16	4/16	13/16	8/16
4	49/49	39/49	27/49	24/49	33/49	24/49
5	129/129	127/129	120/129	86/129	78/129	105/129

#Covered Branches/#Branches in category Branch-n

Evacon-A is best in terms of uniquely covering branches in Branch-1

Using Evacon-A + JUnit Factory is the best choice if only two tools are to be used (not necessarily Evacon-A + Randoop!)

Conclusion: Cooperative Testing and Analysis

- **Computing**-Centric **Human**: Test/Analysis Tools
  - Tool → Human: expose more/less details?
  - Tool ←Human: not reliable guidance?
- **Human**-Centric **Computing**: Educational Tools
  - Human → Tool: more input modalities?
  - Human ← Tool: tutoring hints?
- **Human-Human** (crowdsourcing)
- **Computing-Computing** (synergetic analysis)

Thank you!

Questions ?

https://sites.google.com/site/asergrp/

Automated Software Engineering Research Group @NCSU

Cooperation Between Human and Machine – Other Work

- Agitator testing w/ invariant inference
  [ISSTA 06 Boshernitsan, Doong, and Savoia]
- Parameterized unit testing with Pex
  [ESEC/FSE 05 Tillmann, Schulte]
- End-user program analysis
  [Dissertation 08, Chang]
- Explaining failures of program analyses
  [PLDI 08 von Dincklage, Diwan]
- Measuring effectiveness of error messages designed for novice programmers
  [SIGCSE 11 Marceau, Fisler, Krishnamurthi]

Cooperative Developer Testing

- Developers provide guidance to help tools achieve higher structural coverage

- Apply tools to generate tests
- Tools report achieved coverage & problems
- Developers provide guidance
  - EMCP: Instrumentation or Mock Objects
  - OCP: Factory Methods

DSE Challenges - Preliminary Study

Project	LOC	Cov %	OCP	EMCP	Boundary	Limitation
SvnBridge	17.1K	56.26	11 (42.31%)	15 (57.69%)	0 (0%)	0 (0%)
xUnit	11.4K	15.54	8 (72.73%)	3 (27.27%)	0 (0%)	0 (0%)
Math.Net	3.5K	62.84	17 (70.83%)	1 (4.17%)	4 (16.67%)	2 (8.33%)
QuickGraph	8.3K	53.21	10 (100%)	0 (0%)	0 (0%)	0 (0%)
Total	40.3K	49.87	46 (64.79%)	19 (26.76%)	4 (5.63%)	2 (2.82%)

The total block coverage achieved is 49.87%, with the lowest coverage being 15.54%.

- object-creation problems (OCP) - 64.79%
- external-method call problems (EMCP) - 26.76%
- boundary problems – 5.63%
- limitations of the used constraint solver – 2.82%

External-Method Call Problems (EMCP) Example

- Example 1:
  - **File.Exists** has data dependencies on program input
  - Subsequent branch at Line 1 using the return value of **File.Exists**.
- Example 2:
  - **Path.GetFullPath** has data dependencies on program input
  - **Path.GetFullPath** throws exceptions.
- Example 3: **Stirng.Format** do not cause any problem

```
static string GetDefaultConfigFile(string assembly-
File)  {
00:  string configFilename = assemblyFile + ".config";
01:  if (File.Exists(configFilename))
02:    return configFilename;
03:  return null;
04:  }
...
public ExecutorWrapper(string assemblyFilename, ...) {
05:  ...
06:  assemblyFilename = Path.GetFullPath(assemblyFilename);
07:  ...
}
public AssertActualExpectedException
                (object expected, object actual, ...) {
08:  ...
09:  this.actual += String.Format("(0)",
                        actual.GetType().FullName);
10:  this.expected += String.Format("(0)",
                        expected.GetType().FullName);
11:  ...
}
```

1 2 3

Figure 1: Three simplified methods from xUnit

Object-Creation Problems (OCP) Example

- To cover true branch at Line 5, tools need to generate sequences of method calls:

  Stack s1 = new Stack();

  s1.Push(new object());

  .....

  s1.Push(new object());

  FixedSizeStack s2 = new FixedSizeStack (s1);

- Most tools cannot generate such sequence
- true branch at Line 5 has data dependencies on stack.items (*List<object>*)

```
public class FixedSizeStack {
00:  private Stack stack;
01:  public FixedSizeStack(Stack stack) {
02:    this.stack = stack;
03:  }
04:  public void Push(object item) {
05:    if(stack.Count() == 10) {
06:      throw new Exception("full");
07:    }
08:    stack.Push(item);
09:  }
10:  ...
}
11:  public void TestPush(FixedSizeStack stack,
                          object item){
12:    stack.Push(item);
13:  }
```

stack.Count() returns the size of stack.items