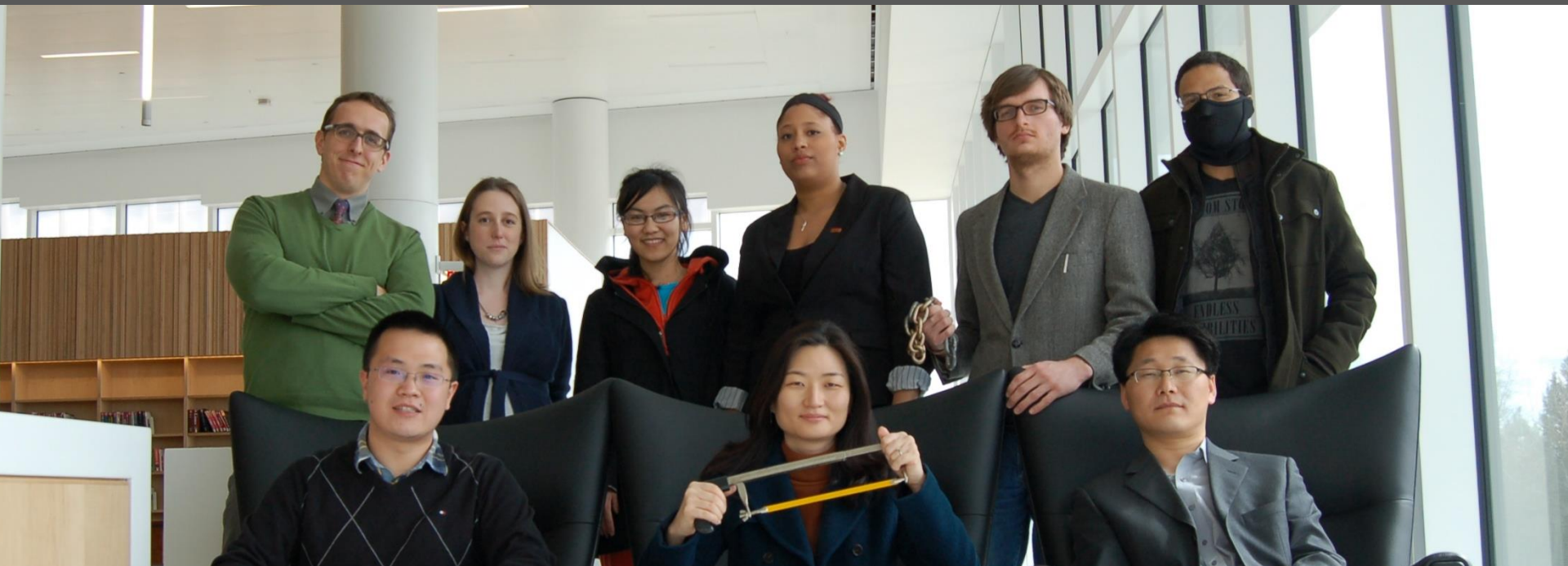
Emerson Murphy-Hill | North Carolina State University

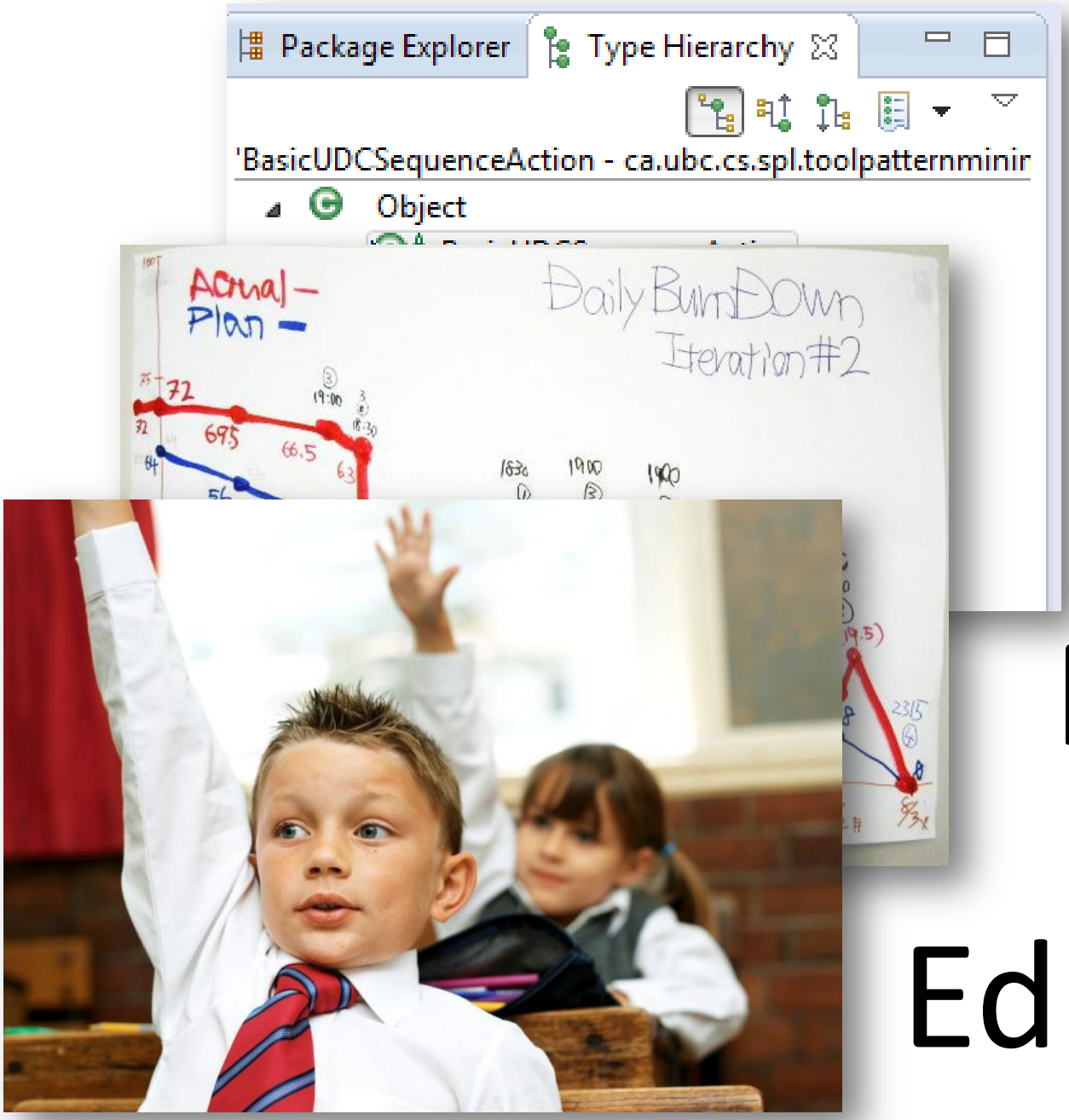
# Helping Developers Find Useful Tools

Developer Liberation Front



# Helping Developers Find Useful Tools

# Helping Developers Find Useful Tools




# Tools

# Process

# Education

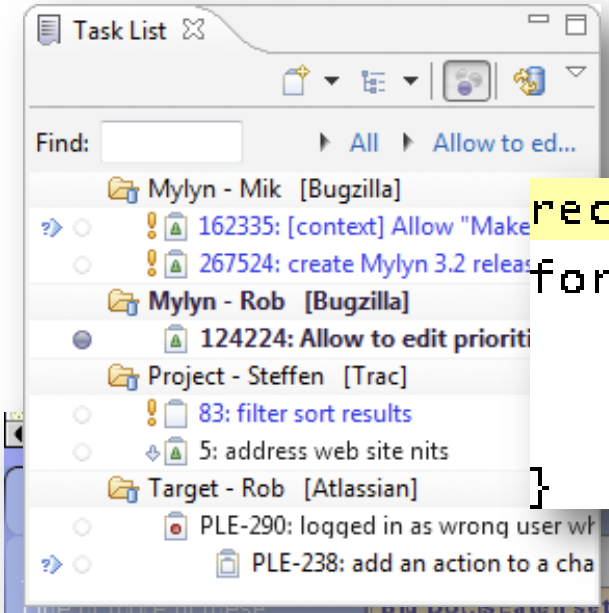
http://www.flickr.com/photos/kakutani/2761992149

# Tools

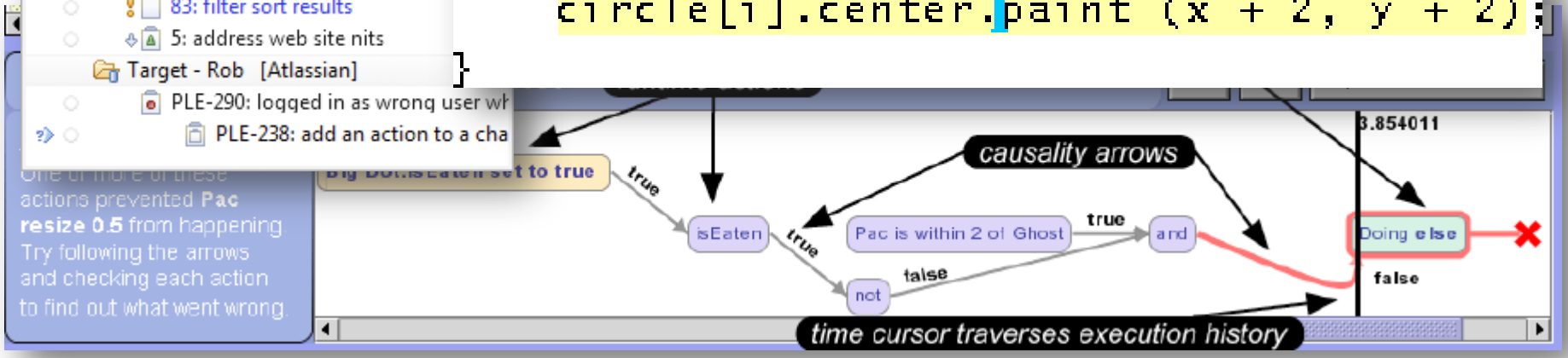
http://www.flickr.com/photos/kakutani/2761992149

# Helping Developers Find Useful Tools

# Useful Tools


```
rectangle.paint (0, 0);
for (int i = 0; i < circle.length; ++i) {
    circle[i].paint (x, y);
    circle[i].center.paint (x + 2, y + 2);
}
```





Checking system rules using system-specific, programmer-written compiler extensions. D Engler, B Chelf, A Chou, S Hallem. OSDI 2001.

Interactive simultaneous editing of multiple text regions. RC Miller, BA Myers. USENIX 2001.

Designing the Whyline: a debugging interface for asking questions about program behavior. AJ Ko, BA Myers. CHI 2004.

Mylar: a degree-of-interest model for IDEs. M Kersten, GC Murphy. ICSE 2004.

# One Kind of Tool: Refactoring Tools

```
class Gorilla{
  int paws(){
    return 4;
  }
}
```

INTRODUCE EXPLAINING VARIABLE →

```
class Gorilla{
  int paws(){
    int pawCount = 4;
    return pawCount;
  }
}
```

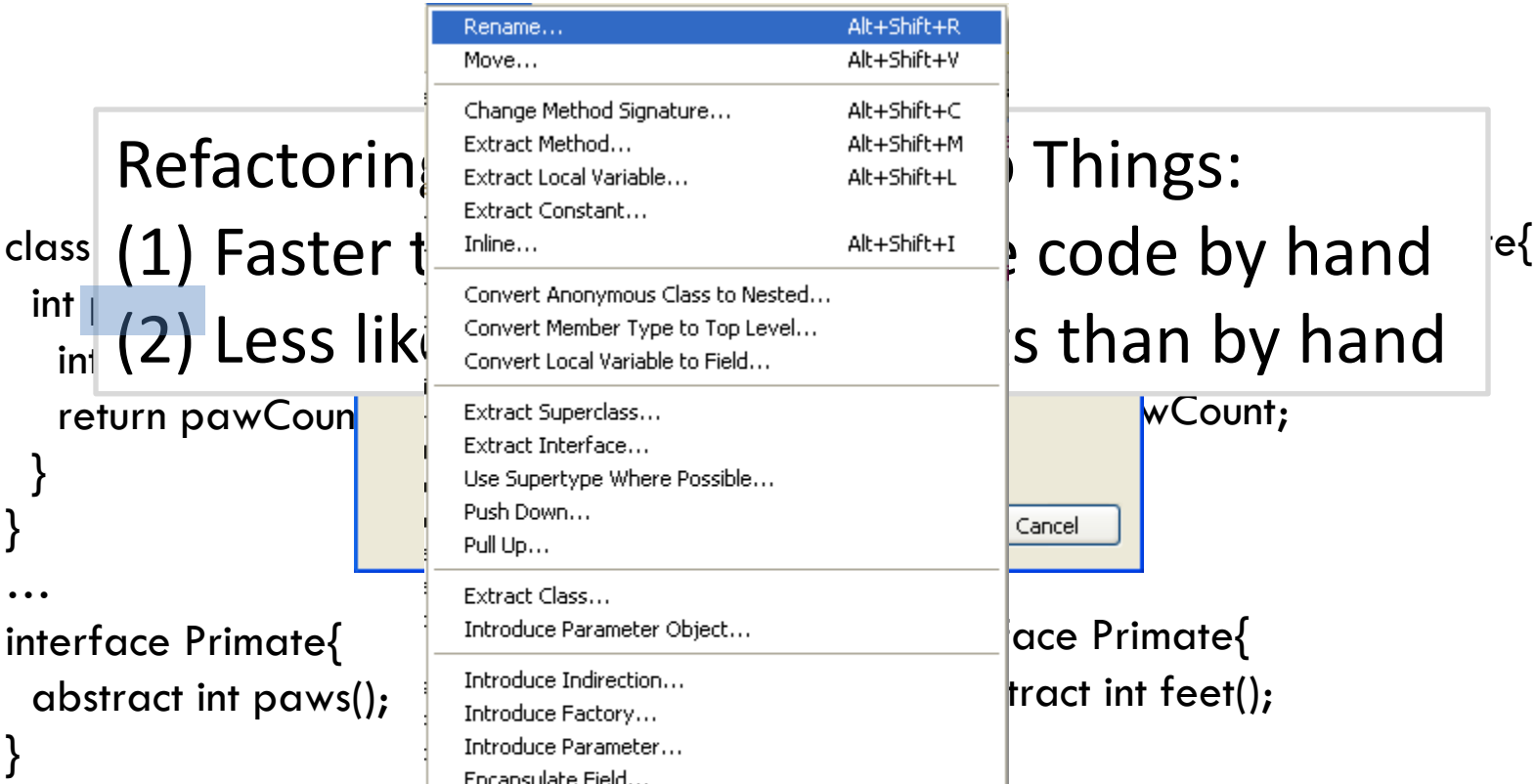
EXTRACT INTERFACE ↓

```
class Gorilla implements Primate{
  int paws(){
    int pawCount = 4;
    return pawCount;
  }
}
…
interface Primate{
  abstract int paws();
}
```

RENAME METHOD →

```
class Gorilla implements Primate{
  int feet(){
    int pawCount = 4;
    return pawCount;
  }
}
…
interface Primate{
  abstract int feet();
}
```

# One Kind of Tool: Refactoring Tools



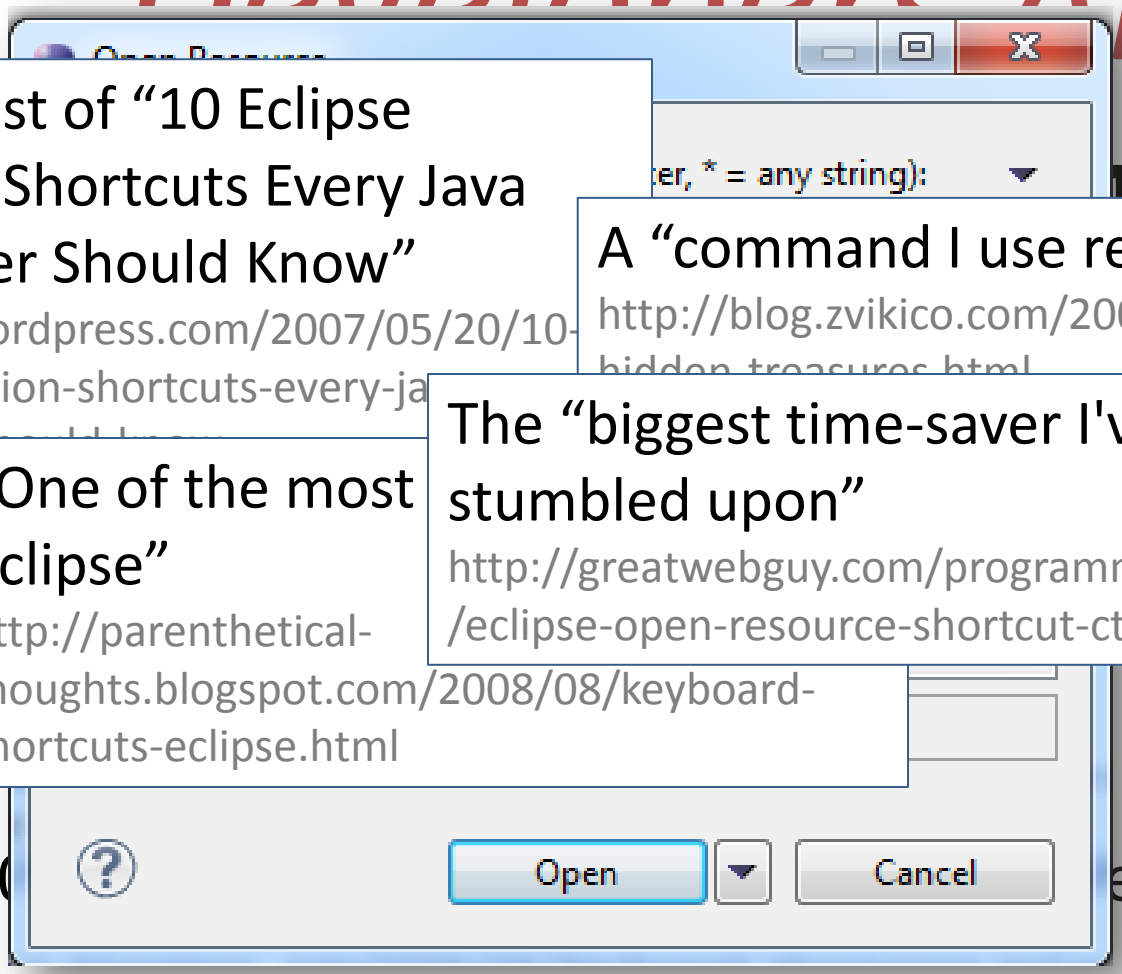
# Developers Aren't

#1 on the list of "10 Eclipse Navigation Shortcuts Every Java Programmer Should Know"
http://rayfd.wordpress.com/2007/05/20/10-eclipse-navigation-shortcuts-every-ja
programmer-should-know

A "command I use religiously"
http://blog.zvikico.com/2009/07/eclipse-35-hidden-treasures.html

The "biggest time-saver I've stumbled upon"
http://greatwebguy.com/programming/eclipse/eclipse-open-resource-shortcut-ctrlshiftr

"One of the most Eclipse"
http://parenthetical-thoughts.blogspot.com/2008/08/keyboard-shortcuts-eclipse.html




The ave
Eclipse t

9 out of 1 esource

How We Refactor, and How We Know It. Emerson Murphy-Hill, Chris Parnin, and Andrew P. Black. TSE, 2011.
Peer Interaction Effectively, yet Infrequently, Enables Programmers to Discover New Tools. Emerson Murphy-Hill and Gail Murphy. CSCW, 2011.

# *Developers Aren't Using* Useful Tools

*Why not?*

Power

Usability

Awareness

More bugs

More scalable verification

Reduce false positives

Better bug prediction model

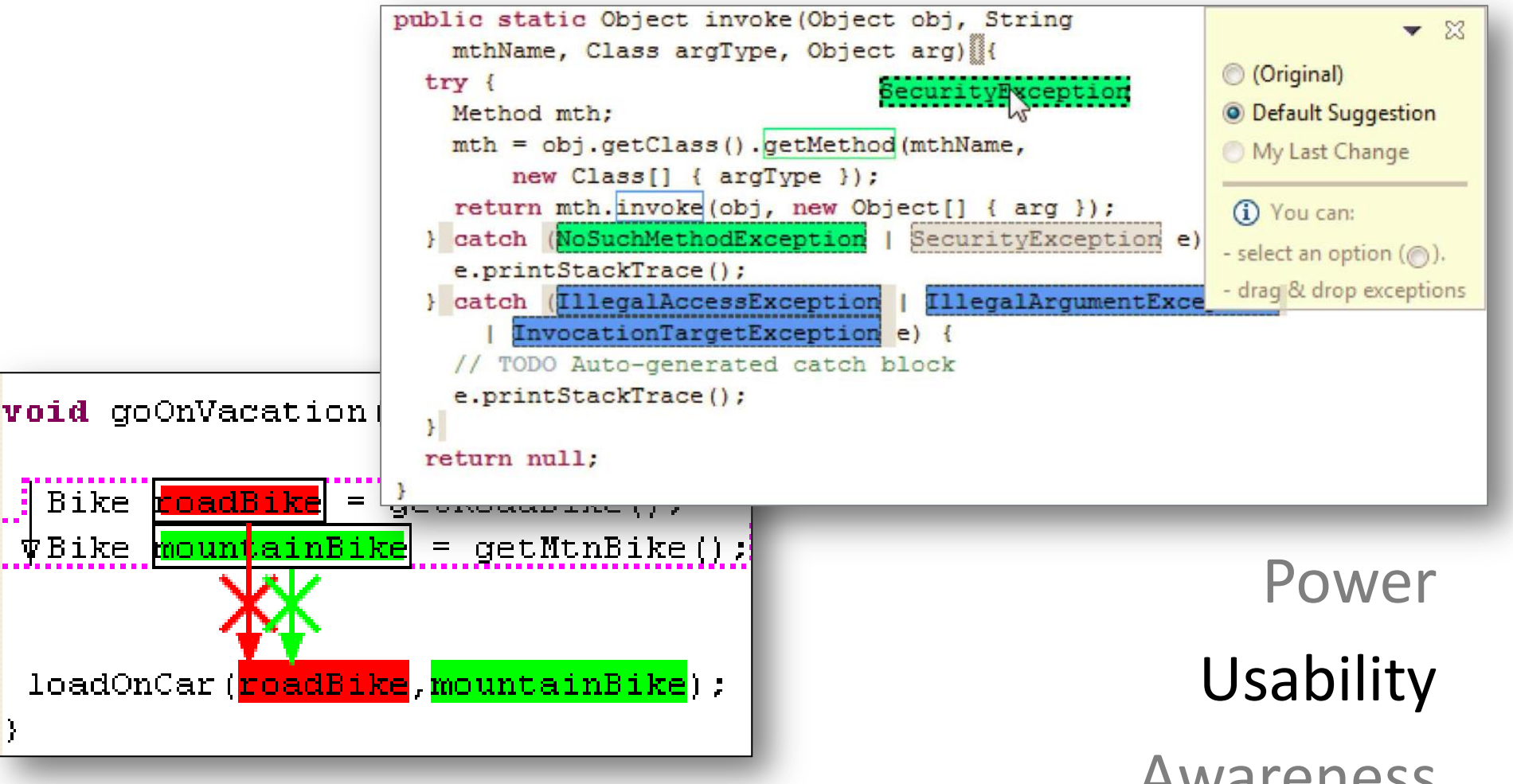
More expressive specifications


Power

Usability

Awareness

# Can developers use the tool in a manner that's quick, error-free, enjoyable, and easy to learn?




Power

Usability

Awareness

Programmer-Friendly Refactoring Errors. Emerson Murphy-Hill and Andrew P. Black. Transactions on Software Engineering, 2011.
Rethinking Quick Fixes: A Follow-through Principle for Tool Design. Yoonki Song, Titus Barik, Brittany Johnson and Emerson Murphy-Hill. In submission to FSE14.

# Helping Developers Find Useful Tools

# Discovering a tool that might be useful:

- You know you need it
  - The compiler
  - An editor
  - Basic search
- You don't know you need it
  - Refactoring tools
  - Navigation tools
  - Static analysis

Basic Tools → Search or Explore

Innovative Tools → ???

Power

Usability

Awareness

# Helping Developers

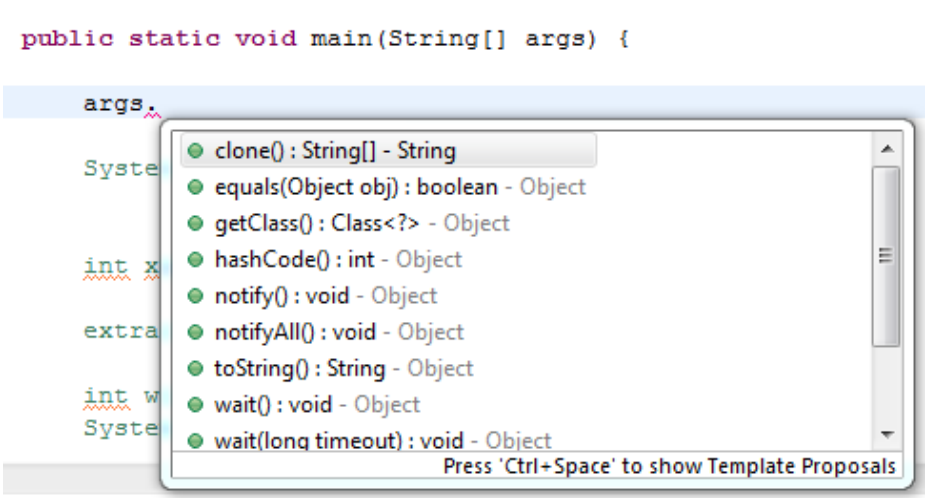
## Find Useful Tools

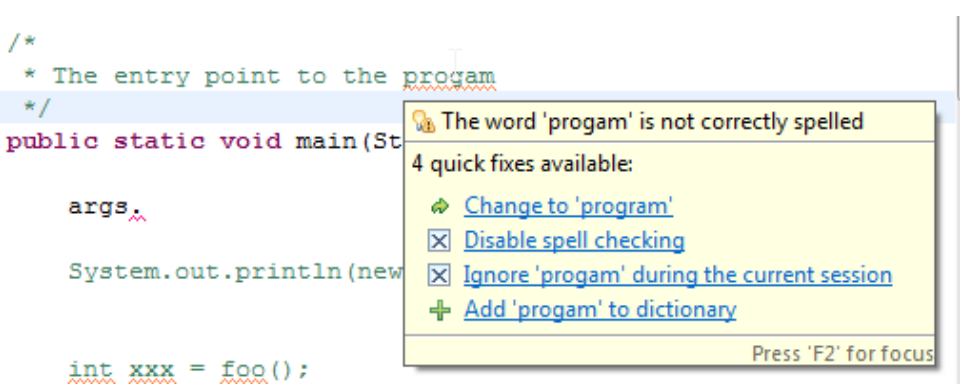
Innovative Tools → ???

Don’t Require Awareness Before Use

Tool Recommendation

Connect Developers Together




# What tools could you redesign this way?

– Refactoring: BeneFactor

# Don't Require Awareness Before Use

## Tool Recommendation

## Connect Developers Together

Improving program navigation with an active help system. Petcharat Viriyakattiyaporn and Gail C. Murphy. CASCON, 2010.

Improving Software Engineers' Fluency by Recommending Dev. Environment Commands. E Murphy-Hill, R Jiresal, and G Murphy, FSE, 2012.

```
CachePeer[] peers = new CLCacheDiscovery().lookup( torrent );
System.out.println( "peers=" + peers.length );
System.out.println();
printPeers(peers);
```

```
private static void printPeers(CachePeer[] peers) {
 for (int i=0;i<peers.length;i++){
  System.out.println("cache:" +
  peers[i].getAddress() + ":" + peers[i].getPort());}}
```

What tools could you redesign this way?

- Refactoring: BeneFactor
- Lesson: Identify first part of tool-less activity, offer tool support

Don’t Require Awareness Before Use

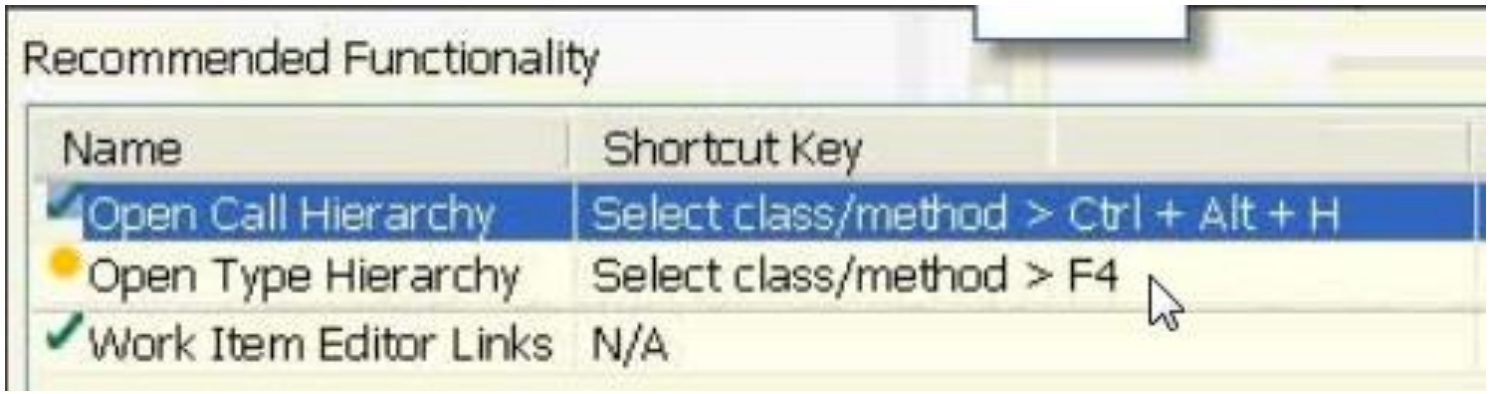
Tool Recommendation

Connect Developers Together

# Recommend tools that developers aren't using

## Examples

- Spyglass
- Collaborative Filtering




Don't Require Awareness Before Use

Tool Recommendation

Connect Developers Together

Improving program navigation with an active help system. Petcharat Viriyakattiyaporn and Gail C. Murphy. CASCON, 2010.
Improving Software Engineers' Fluency by Recommending Dev. Environment Commands. E Murphy-Hill, R Jiresal, and G Murphy, FSE, 2012.

relevance | real problem | self discovery | context association


Don’t Require Awareness Before Use

Tool Recommendation

# Connect Developers Together

www.flickr.com/photos/a2gemma/552208117

Peer Interaction Effectively, yet Infrequently, Enables Programmers to Discover New Tools. Emerson Murphy-Hill and Gail Murphy. CSCW, 2011.

# Pair Programming

Personal, contextual, efficient

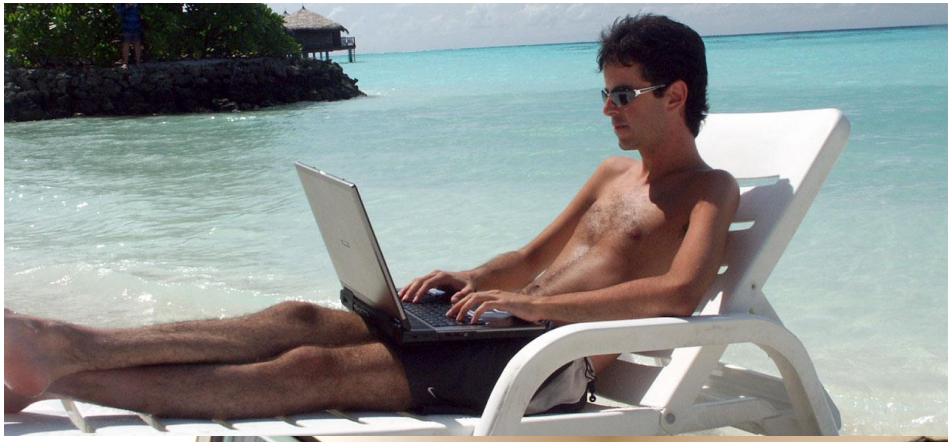
Doesn't work in remote work situations

Connect Developers Together

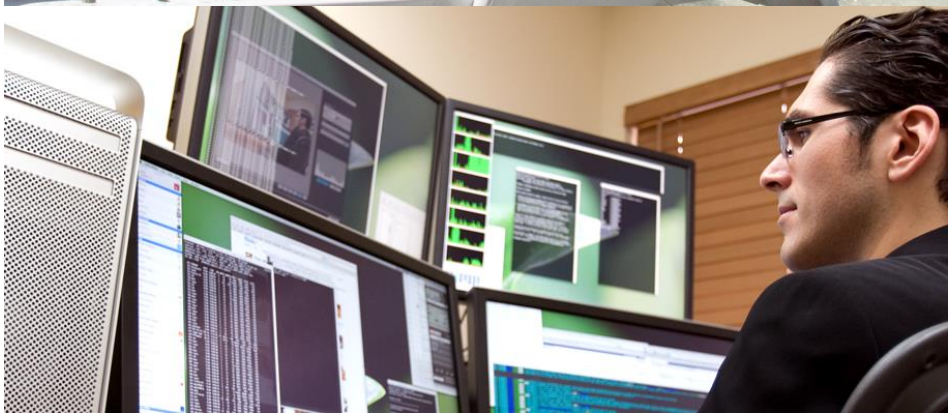
http://www.flickr.com/photos/menlopics/3928252631

# Remote Pair Programming

Works remotely



Doesn't work when people working asynchronously



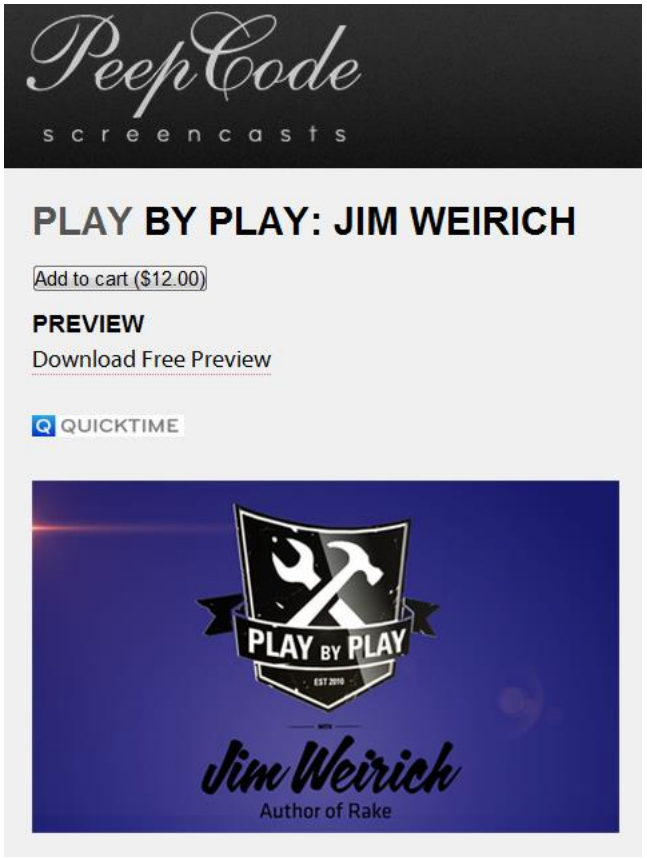
## Connect Developers Together

http://www.flickr.com/photos/novecentino/2339687721/
http://www.flickr.com/photos/totalaldo/2400635097

# Screencasts

Allows people to learn remotely and asynchronously

Not personal: your peers aren't making them



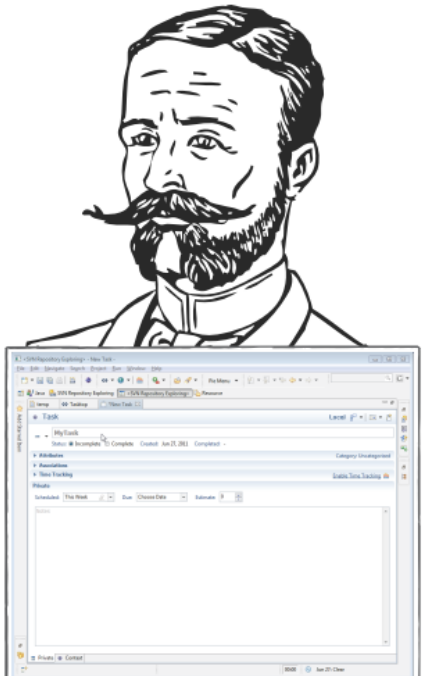

# Connect Developers Together

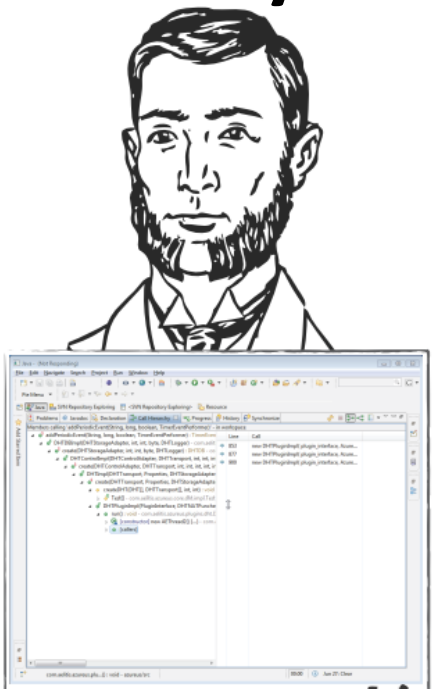
https://peepcode.com/products/play-by-play-jimweirich-ruby

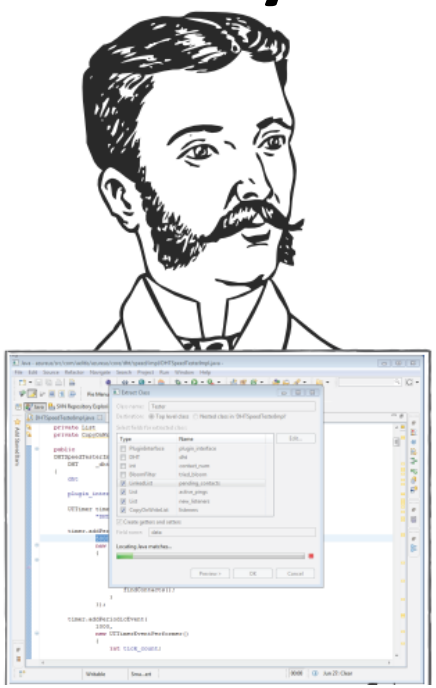
How can developers discover tools from **peers** when they work **remotely** and **asynchronously**?

Connect Developers Together

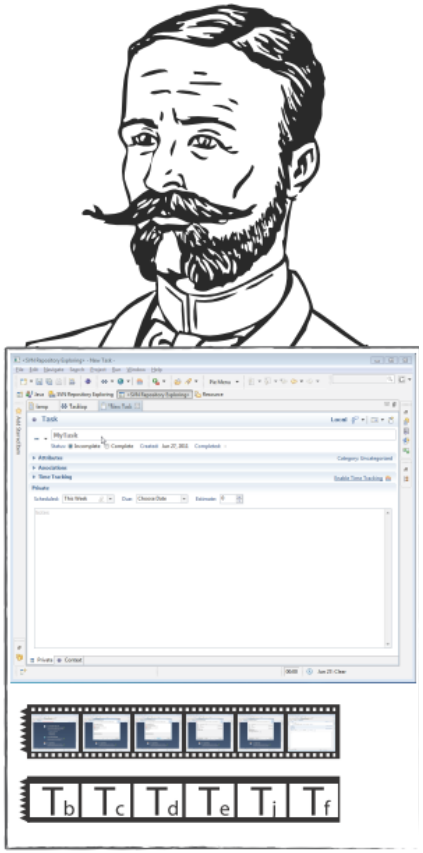
# How can developers discover tools from **peers** when they work **remotely** and **asynchronously**?

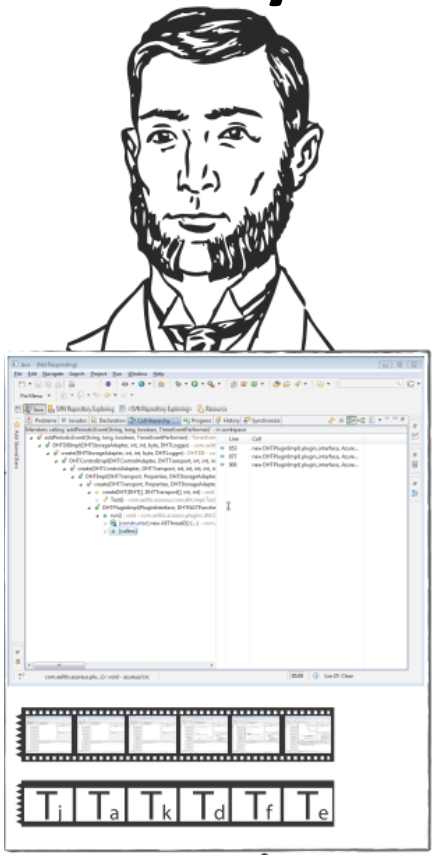


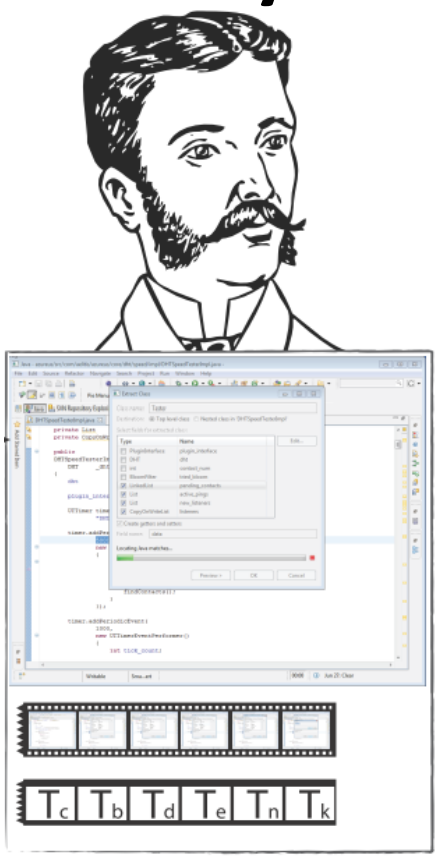




# How can developers discover tools from peers when they work **remotely** and **asynchronously**?




Server

| | | | | | | |
|---|---|---|---|---|---|---|
| *cm1* | Tb | Tc | Td | Te | Tj | Tf |
| *cm2* | Tj | Ta | Tk | Td | Tf | Te |
| *cm3* | Tc | Tb | Td | Te | Tn | Tk |

Continuous Social Screencasting to Facilitate Software Tool Discovery. Emerson Murphy-Hill. ICSE NIER, 2012.

Conclusion

# Helping Developers Find Useful Tools

All our toolsmithing is for naught if no one uses our tools.

We can encourage more tool use by:

- Designing our tools so that awareness is not a prerequisite
- Building environments that recommend useful tools
- Enable developers to learn from one another