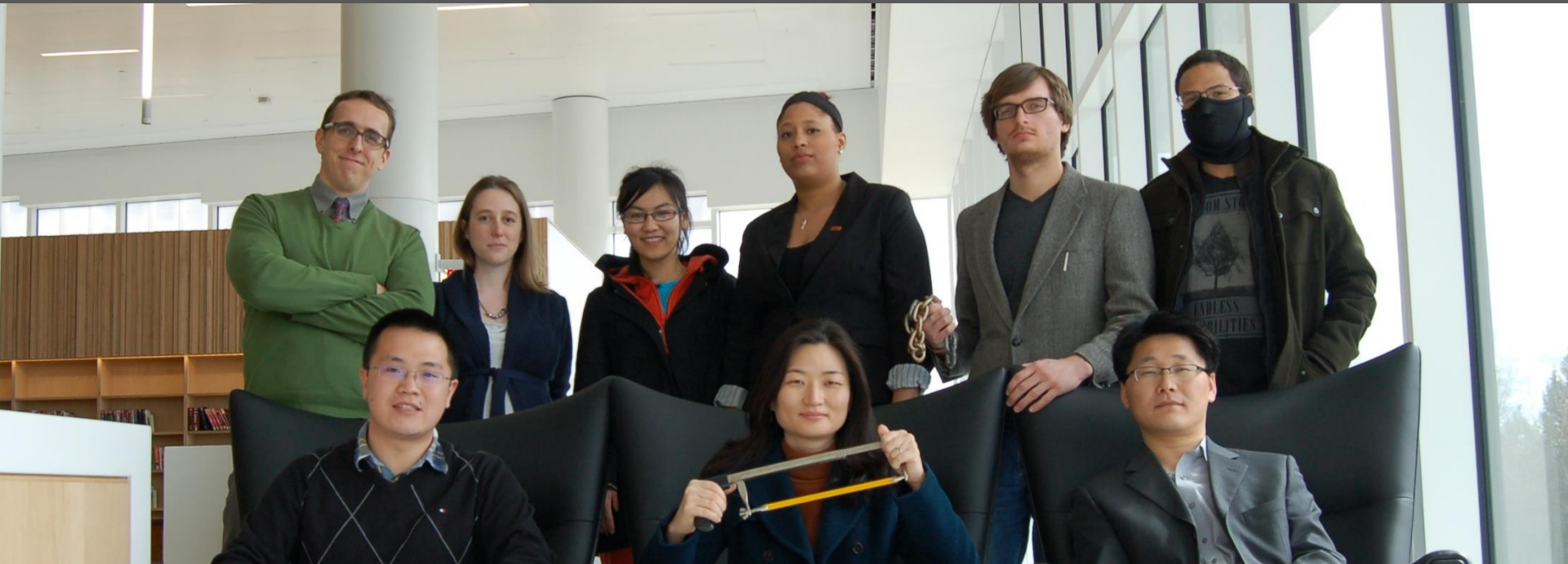


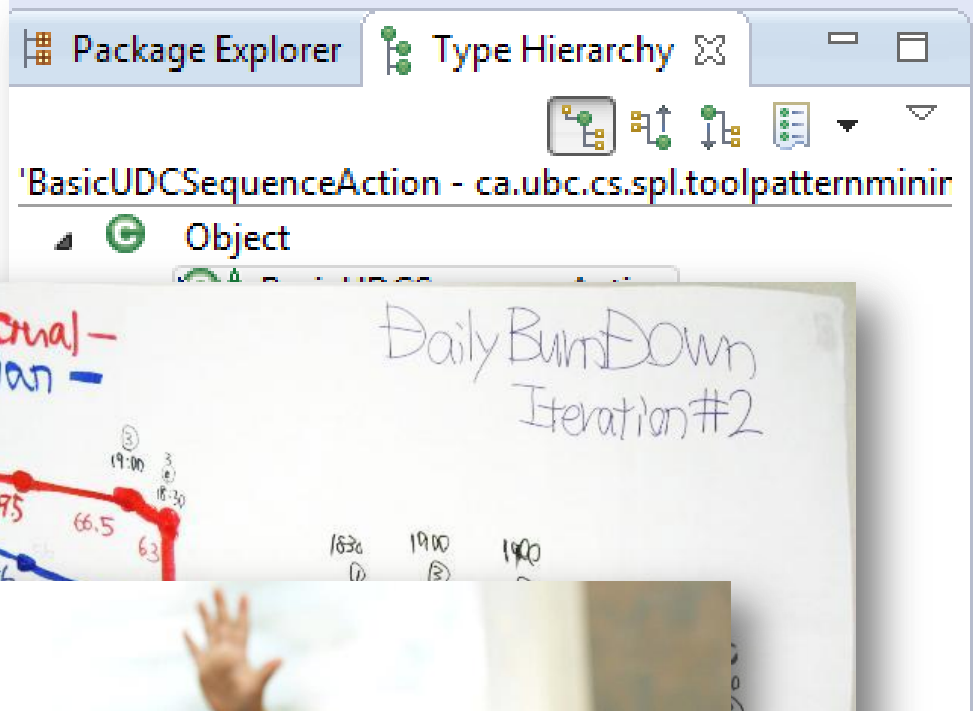
# Helping Developers Find Useful Tools

Developer Liberation Front



# Helping Developers Find Useful Tools

# Helping Developers Find Useful **Tools**



# Tools

# Process

# Education

# Tools

# Helping Developers Find **Useful** Tools



# One Kind of Tool: Refactoring Tools

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

INTRODUCE EXPLAINING VARIABLE



EXTRACT INTERFACE



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

RENAME METHOD

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

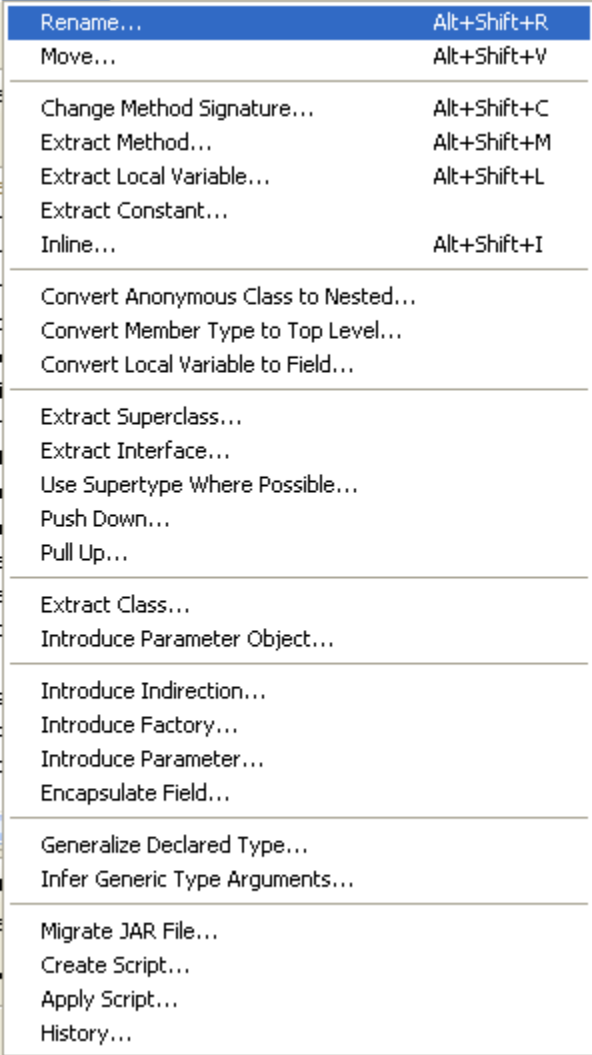


# One Kind of Tool: Refactoring Tools

Refactoring

(1) Faster to write code by hand

(2) Less likely to introduce errors than by hand



The screenshot shows a refactoring menu with various options. The first section includes 'Rename...', 'Move...', 'Change Method Signature...', 'Extract Method...', 'Extract Local Variable...', 'Extract Constant...', and 'Inline...'. The second section includes 'Convert Anonymous Class to Nested...', 'Convert Member Type to Top Level...', and 'Convert Local Variable to Field...'. The third section includes 'Extract Superclass...', 'Extract Interface...', 'Use Supertype Where Possible...', 'Push Down...', and 'Pull Up...'. The fourth section includes 'Extract Class...', 'Introduce Parameter Object...', 'Introduce Indirection...', 'Introduce Factory...', 'Introduce Parameter...', and 'Encapsulate Field...'. The fifth section includes 'Generalize Declared Type...' and 'Infer Generic Type Arguments...'. The sixth section includes 'Migrate JAR File...', 'Create Script...', 'Apply Script...', and 'History...'. A 'Cancel' button is visible in the bottom right of the menu area.

```
class ...
int ...
int ...
return pawCount;
}
}
...
interface Primate{
    abstract int paws();
}

...
interface Primate{
    abstract int feet();
}
```

# 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-java-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 average Eclipse t

9 out of 10

resource

# *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?

```
public static Object invoke(Object obj, String
    mthName, Class argType, Object arg){
    try {
        Method mth;
        mth = obj.getClass().getMethod(mthName,
            new Class[] { argType });
        return mth.invoke(obj, new Object[] { arg });
    } catch (NoSuchMethodException | SecurityException e)
        e.printStackTrace();
    } catch (IllegalArgumentException | InvocationTargetException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return null;
}
```

**void goOnVacation()**

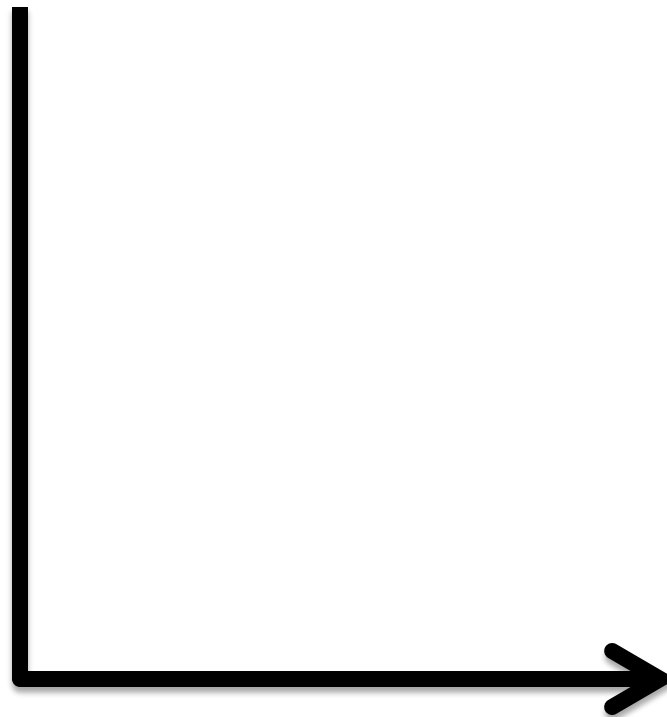
Bike roadBike = getRoadBike();  
Bike mountainBike = getMtnBike();

loadOnCar(roadBike, mountainBike);

**Power Usability Awareness**

# Helping Developers

## **Find** Useful Tools



Power

Usability

Awareness

# Discovering a tool that might be useful:

You know you need it

The compiler

An editor

Basic search

Basic Tools → Search or Explore

You don't know you need it

Refactoring tools

Navigation tools

Static analysis

Innovative Tools → ???

Power

Usability

Awareness

# Helping Developers Find Useful Tools

Innovative Tools → ???

Don't Require Awareness Before Use  
Tool Recommendation  
Connect Developers Together



```
public static void main(String[] args) {
```

```
args.
```

```
System
```

```
int x
```

```
extra
```

```
int w
```

```
System
```

- clone() : String[] - String
- equals(Object obj) : boolean - Object
- getClass() : Class<?> - Object
- hashCode() : int - Object
- notify() : void - Object
- notifyAll() : void - Object
- toString() : String - Object
- wait() : void - Object
- wait(long timeout) : void - Object

Press 'Ctrl+Space' to show Template Proposals

```
/**  
 * The entry point to the program  
 */
```

```
public static void main (St
```

```
args.
```

```
System.out.println(new
```

```
int xxx = foo();
```

The word 'progam' is not correctly spelled

4 quick fixes available:

- ➔ [Change to 'program'](#)
- ☒ [Disable spell checking](#)
- ☒ [Ignore 'progam' during the current session](#)
- ➕ [Add 'progam' to dictionary](#)

Press 'F2' for focus

# What tools could you redesign this way?

- Refactoring: BeneFactor

## Don't Require Awareness Before Use

## Tool Recommendation

## Connect Developers Together

```
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

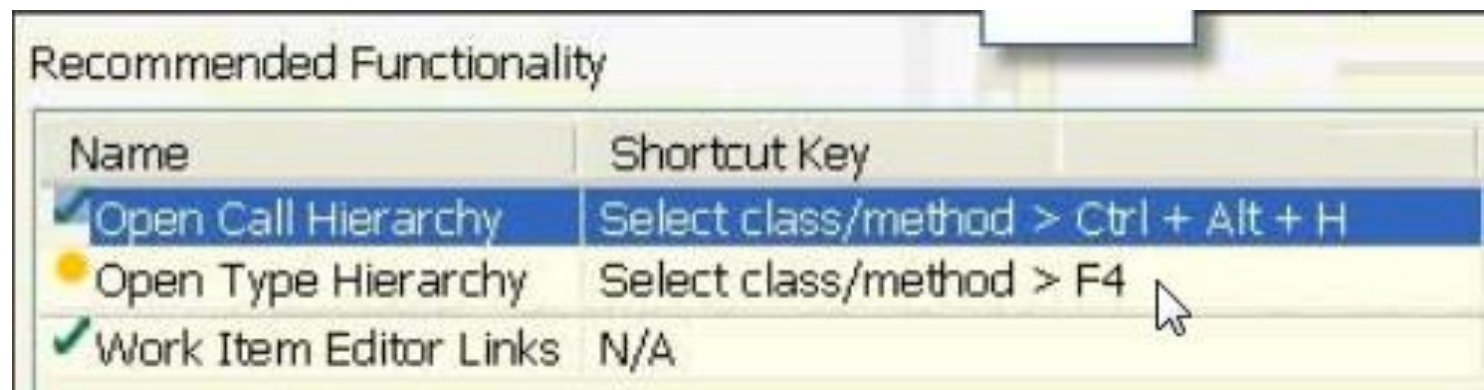
Tool Recommendation

Connect Developers Together

# Recommend tools that developers aren't using

## Examples

- Spyglass
- Collaborative Filtering



A screenshot of a software interface window titled "Recommended Functionality". It contains a table with two columns: "Name" and "Shortcut Key". The table lists three items: "Open Call Hierarchy" (checked with a green checkmark), "Open Type Hierarchy" (marked with a yellow dot), and "Work Item Editor Links" (checked with a green checkmark). The first two items have associated shortcut keys: "Ctrl + Alt + H" and "F4" respectively. A mouse cursor is pointing at the "F4" shortcut.

Name	Shortcut Key
✓ Open Call Hierarchy	Select class/method > Ctrl + Alt + H
● Open Type Hierarchy	Select class/method > F4
✓ Work Item Editor Links	N/A

Don't Require Awareness Before Use  
Tool Recommendation

Connect Developers Together

relevance | real problem | self discovery | context association



Don't Require Awareness Before Use  
Tool Recommendation  
Connect Developers Together

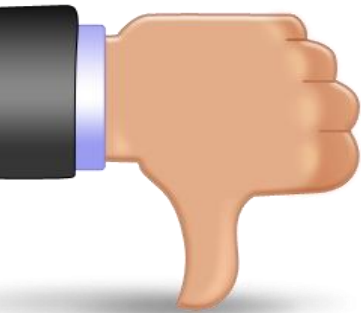
[www.flickr.com/photos/a2gemma/552208117](http://www.flickr.com/photos/a2gemma/552208117)

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



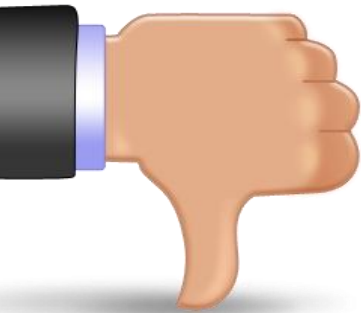
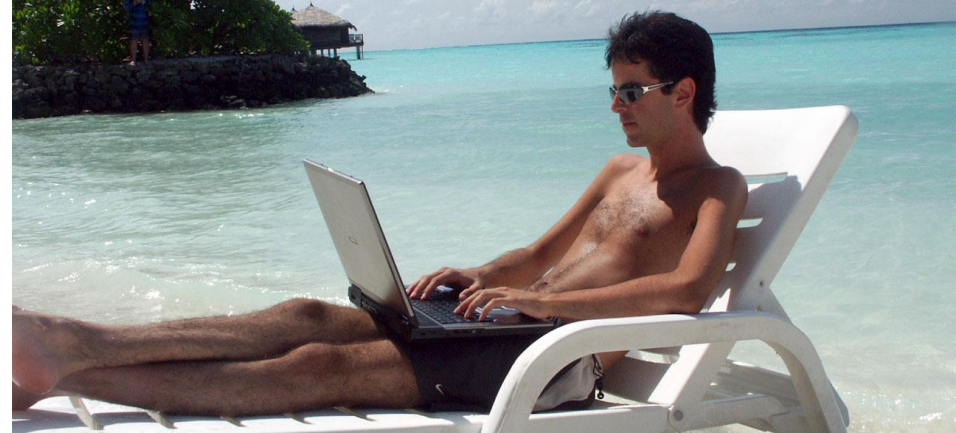
## Connect Developers Together



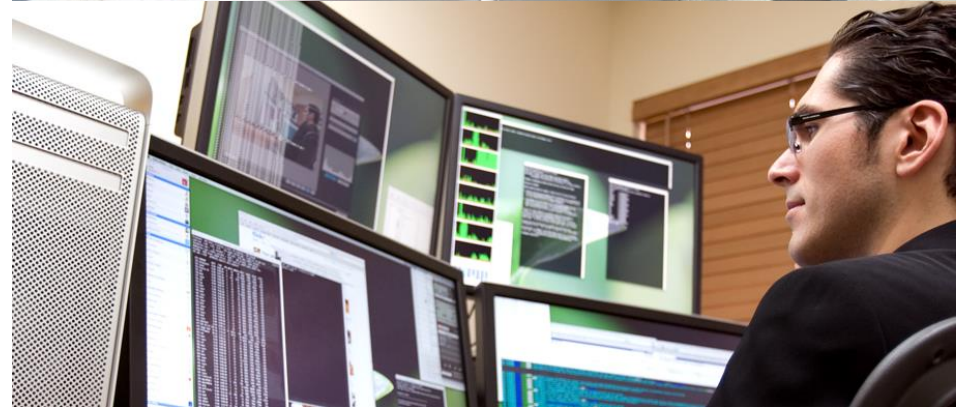
# Remote Pair Programming



Works remotely



Doesn't work  
when people  
working  
asynchronously



## Connect Developers Together

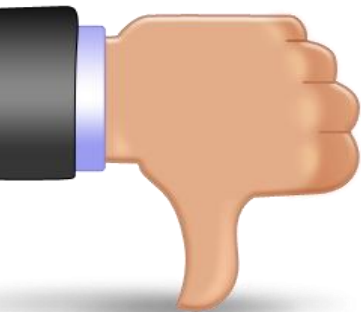
<http://www.flickr.com/photos/novecentino/2339687721/>

<http://www.flickr.com/photos/totalaldo/2400635097>

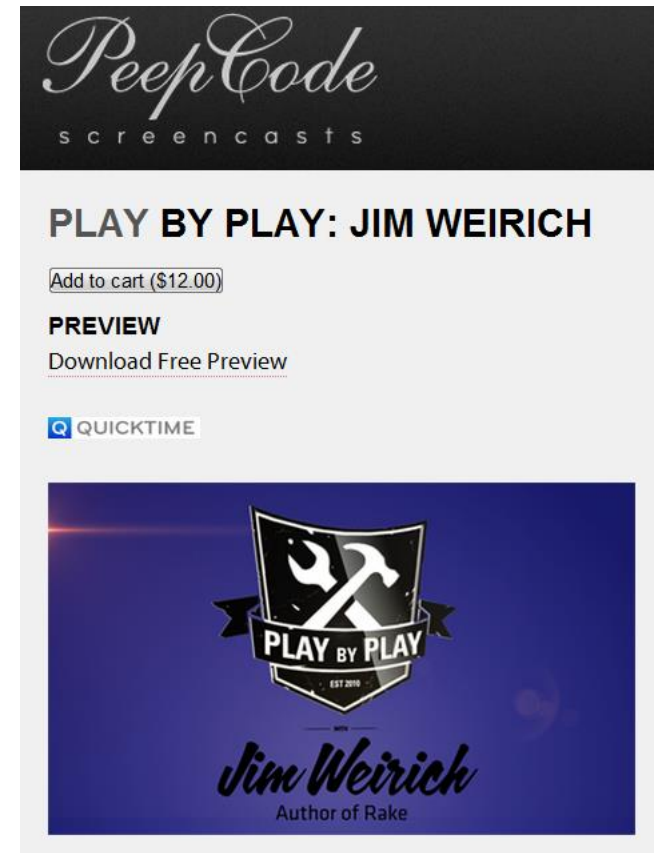
# Screencasts



Allows people to  
learn remotely  
and  
asynchronously



Not personal:  
your peers aren't  
making them



## Connect Developers Together

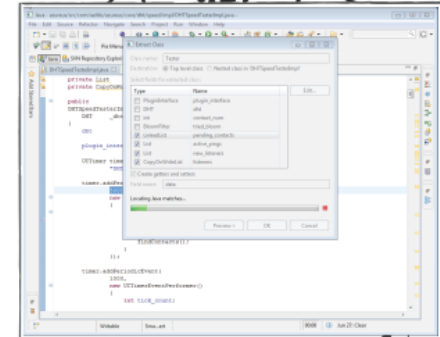
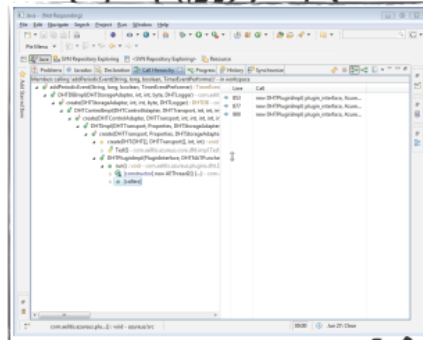
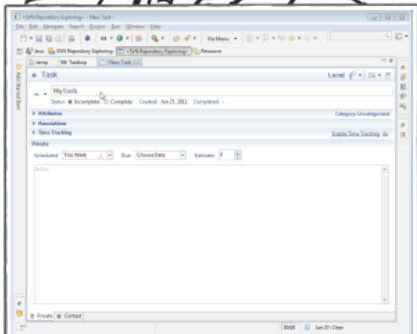
<https://peepcode.com/products/play-by-play-jimweirich-ruby>

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

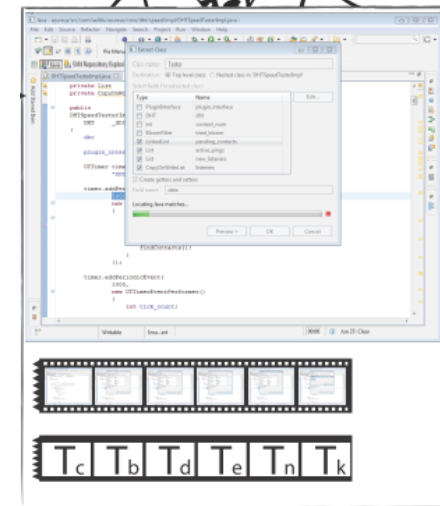
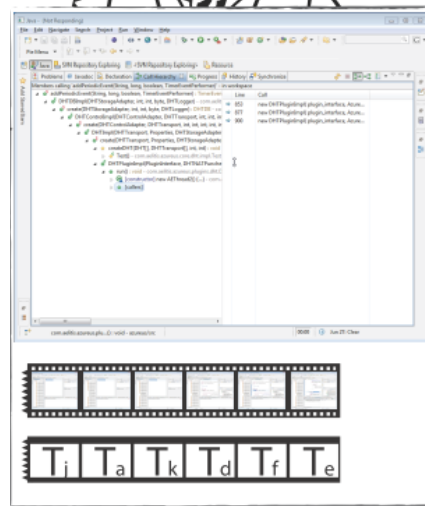
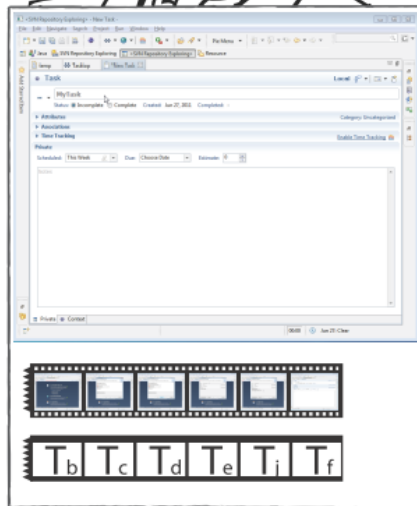
Connect Developers Together



# 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	T <sub>b</sub>	T <sub>c</sub>	T <sub>d</sub>	T <sub>e</sub>	T <sub>j</sub>	T <sub>f</sub>
cm2	T <sub>j</sub>	T <sub>a</sub>	T <sub>k</sub>	T <sub>d</sub>	T <sub>f</sub>	T <sub>e</sub>
cm3	T <sub>c</sub>	T <sub>b</sub>	T <sub>d</sub>	T <sub>e</sub>	T <sub>n</sub>	T <sub>k</sub>

# 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