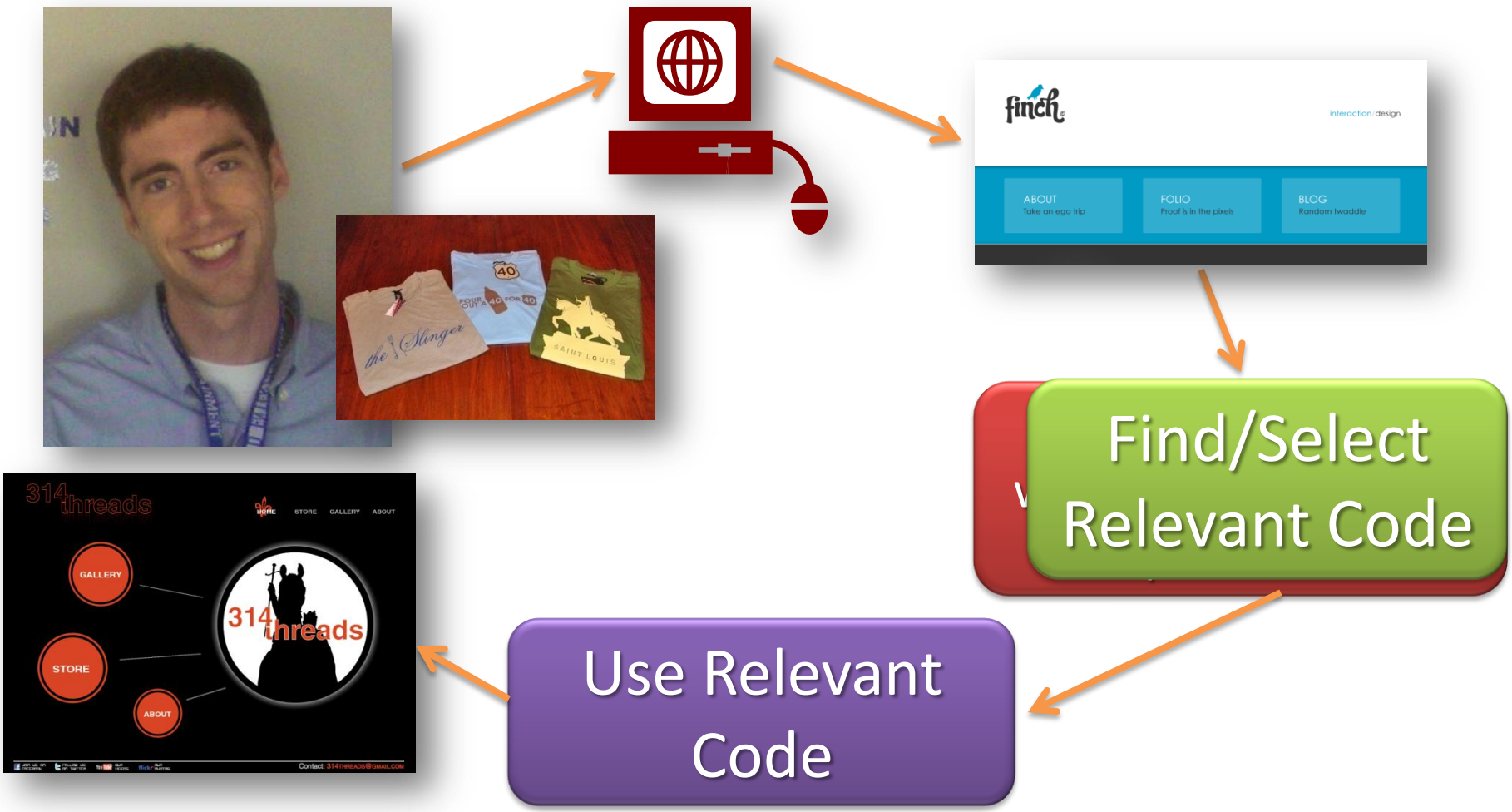
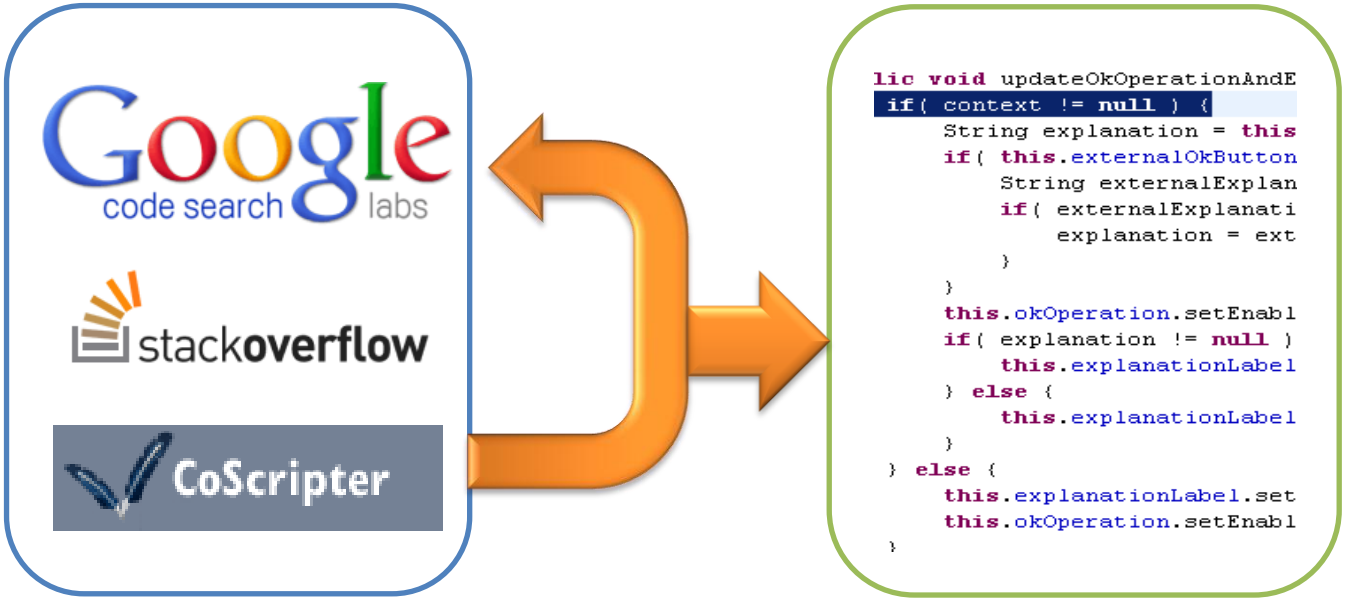
Toward Transforming Freely Available Source Code into Usable Learning Materials for End-Users

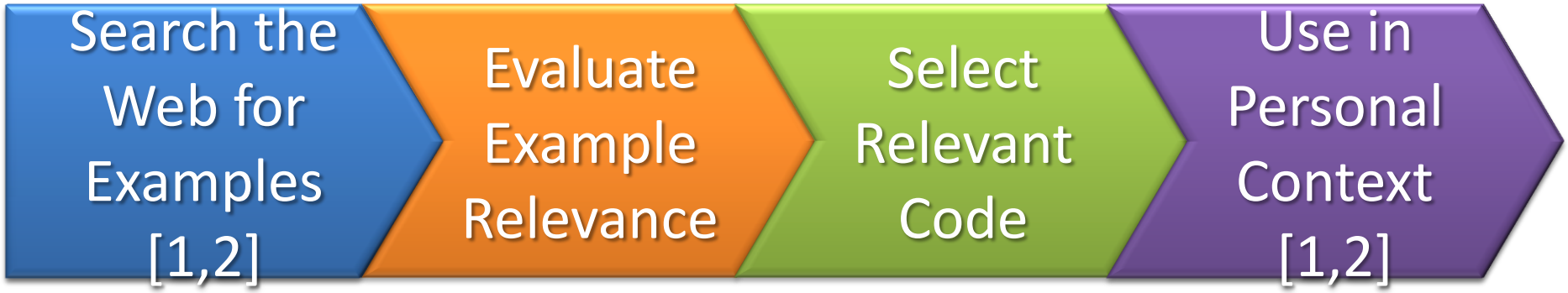
Paul Gross and Caitlin Kelleher
Washington University in St. Louis

Illustrating Available, Unusable Code



An Independent Learning Process for End-user Programmers





[1] Brandt et al., *Two studies of opportunistic programming: interleaving web…*, CHI 2009.
[2] Rosson et al., *Who, what, and how: A survey of informal and professional…*, VL/HCC 2005.

Our Context: Looking Glass

Looking Glass C:\Documents and Settings\grosspa\My Documents\LookingGlass\MyProjects\CafeteriaKiss.a3p

File Edit Run Window Help

this (a.k.a. scene) Run...

this.camera
this.grassyGround
this.sunLight
this.cafeteria
this.kristen
this.philip
this.lunchLady

edit scene

Procedures Functions Properties

this.philip PlugEars

GalleryPerson

Person

this.philip keepTouching target: ???
this.philip touch target: ???
this.philip walk distance: ???
this.philip walkOffscreen
this.philip walkTo target: ???
this.philip fallDown

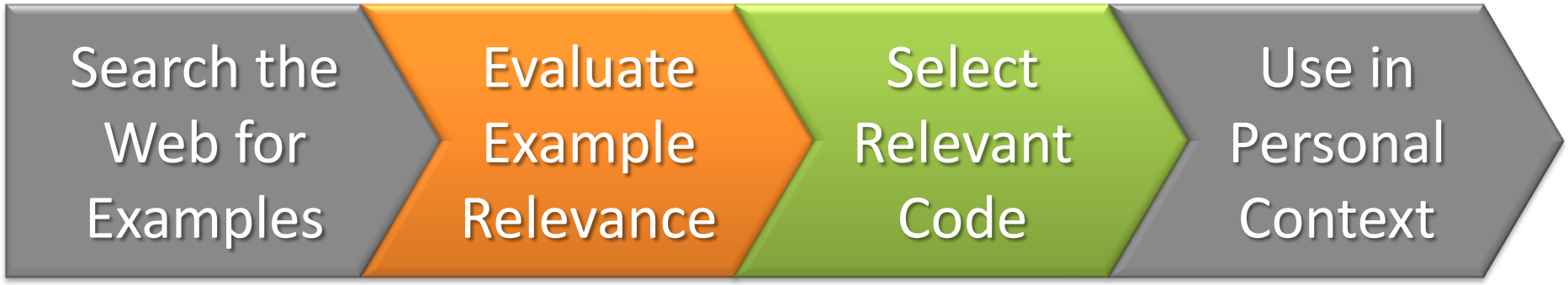
do in order | count | while | for each in | if | do together | each in together | local... | //comment

run | AttemptToBrainwash_lg | kiss

declare procedure run current instance: this

```
do in order
    this.camera showTitle "Cafeteria Romance", duration: 2.0 more
    this.lunchLady say "Crazy kids everywhere!" more
    this.lunchLady leaveOverwhelmed
    this.philip say "Kristen, we are finally alone..." more
    this.philip kiss whoToKiss: this.kristen
    this.lunchLady say "What are you DOING?" more
    this.philip turnToFace this.lunchLady more
    do together
        this.lunchLady AttemptToBrainwash_lg whosHead: (this.philip getPart Head)
        do in order
            this.lunchLady delay 11.0
            this.lunchLady say "You do not like girls." duration: 2.0 more
```

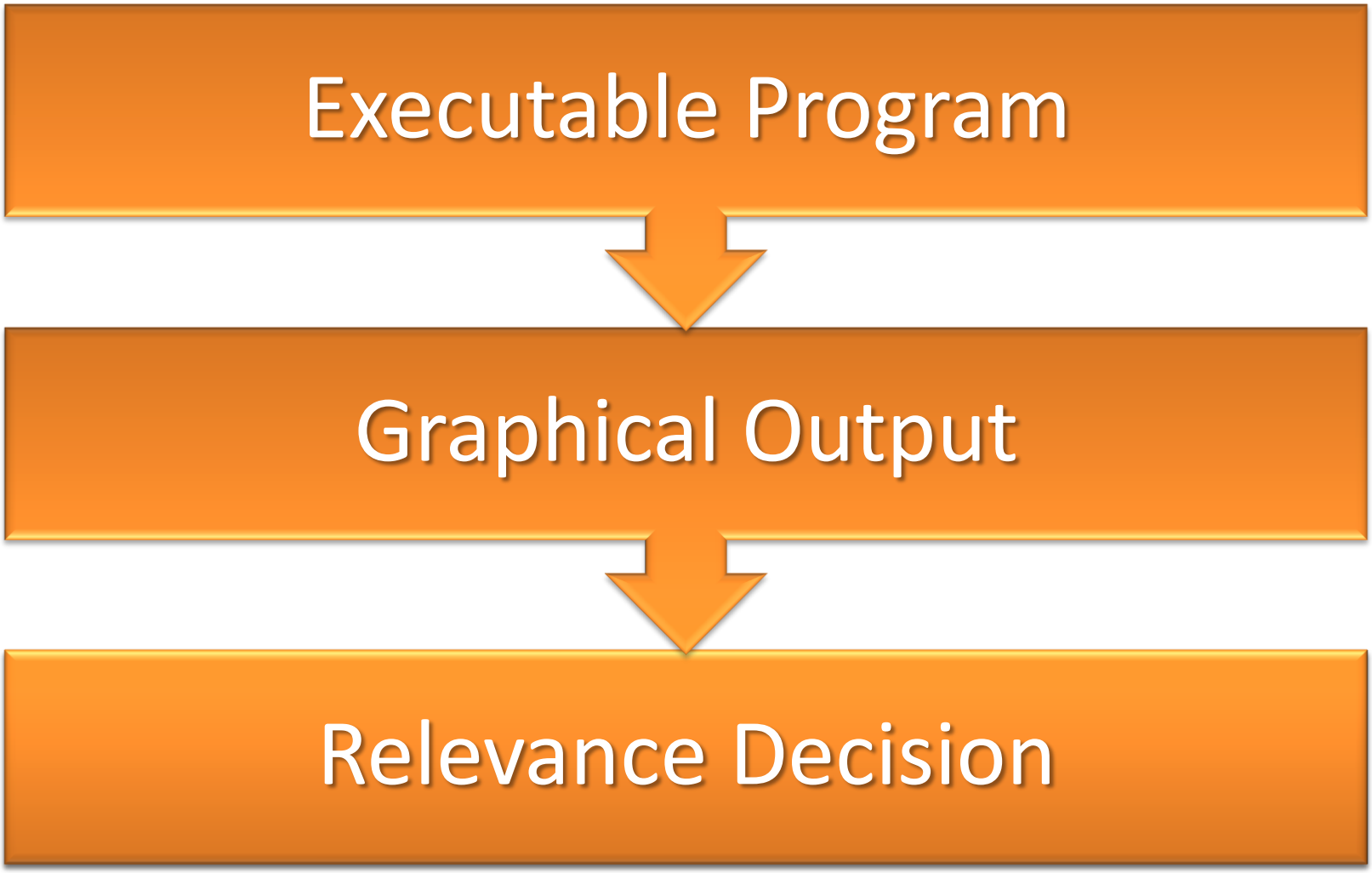
An Independent Learning Process for End-user Programmers



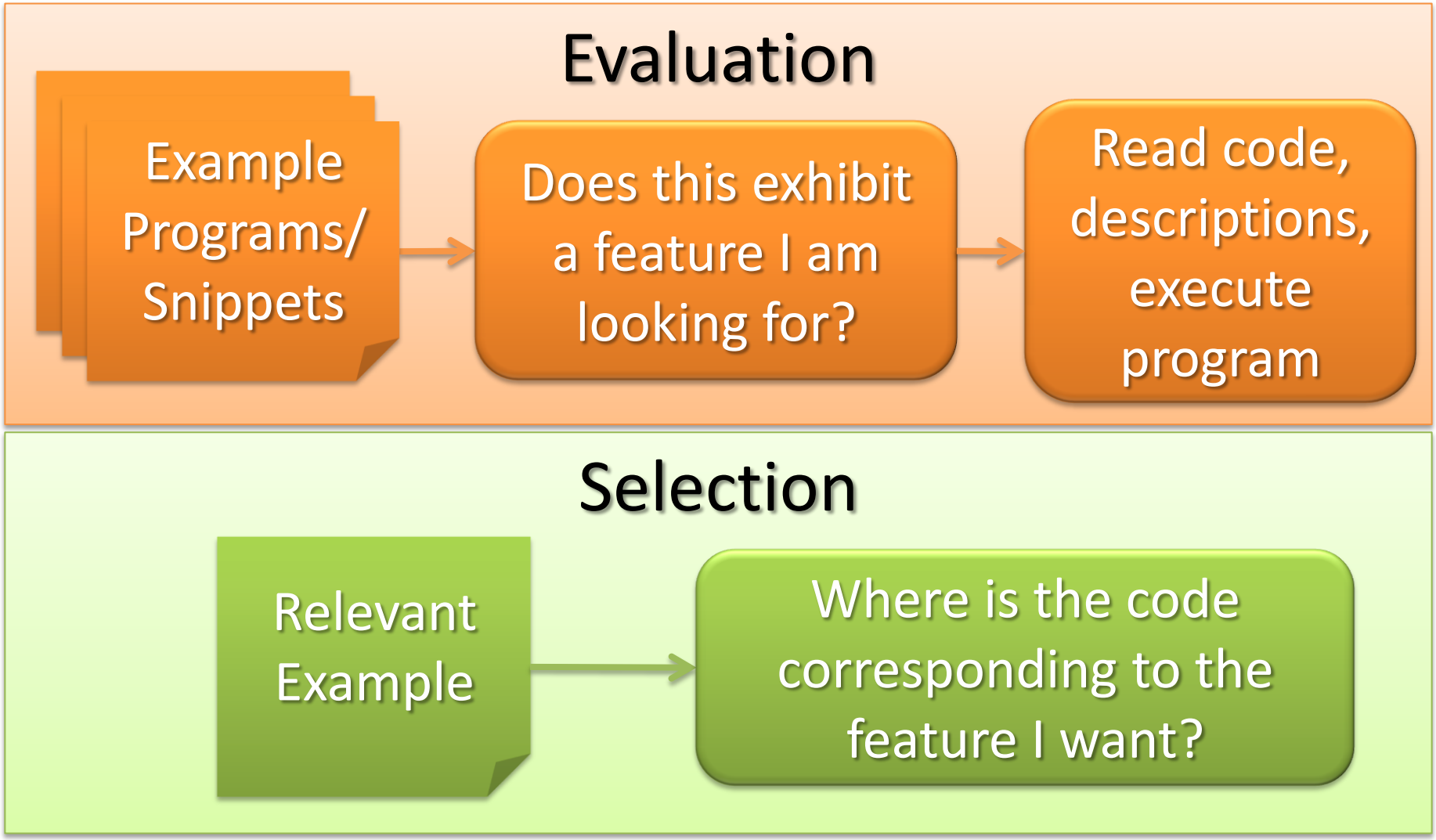
Evaluation by Inexperienced End-Users

- Inexperienced end-users envision output

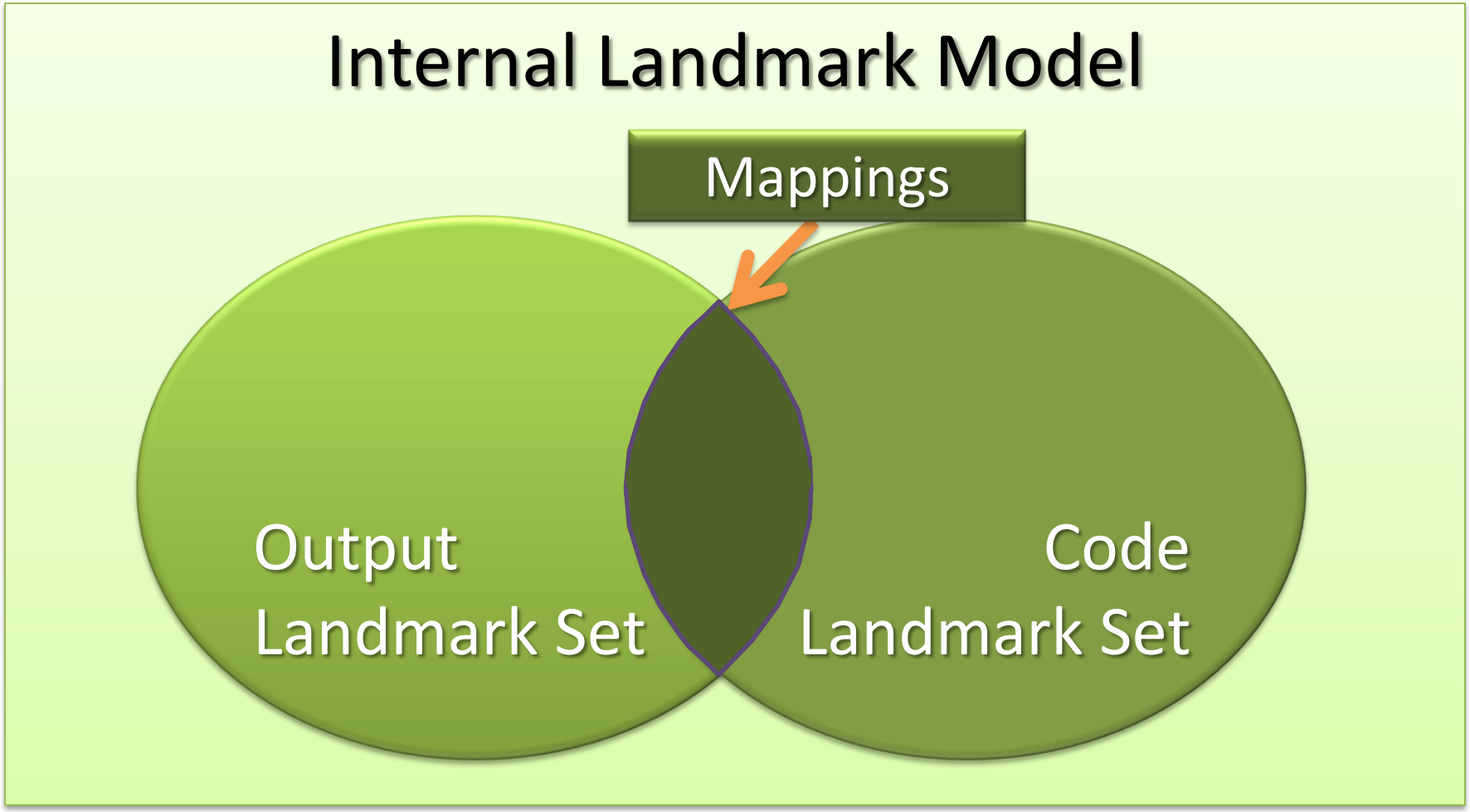
Evaluation by Inexperienced End-Users



Example Code Usability for Inexperienced End-Users



Non-Programmers' Selection Search Model



[3] Gross and Kelleher. *Non-programmers identifying functionality…*, JVLC (21), 5, Aug 2010.

Mapping Code to Output: Debuggers

Debug - ide/src/edu/wustl/cse/lookingglass/reuse/stencil/ui/SelectStatementPanel.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Debug | Hierarchy

org.alice.stageide.EntryPoint at localhost:3420
Thread [AWT-Shutdown] (Running)
Daemon Thread [AWT-Windows] (Running)
Thread [AWT-EventQueue-0] (Suspended)
SelectStatementPanel$StatementLabel.createAwtComponent() line: 103
SelectStatementPanel$StatementLabel(Label).createAwtComponent() line: 1
SelectStatementPanel$StatementLabel(Component<J>).getAwtComponent() line: 210
SelectStatementPanel$StatementLabel(Label).setText(String) line: 79

Breakpoints

Operation [line: 99] - handleFire(ModelContext<?>, EventObject, ViewController<?, ?>)
PartChooserSceneViewer$PartChooserMouseAdapter [line: 610] - mouseClicked(MouseEvent)
ProgramCodeHole$HighlightCodeJComponent [line: 58] - drawHighlightRectangle(Graphics2D, Str
SelectStatementPanel$StatementLabel [line: 100] - createAwtComponent()

SelectStatementPanel.java

```
@Override
protected JLabel createAwtComponent() {
    javax.swing.JLabel rv = new javax.swing.JLabel();
    int w = 175;
    int h = 20;
    rv.setOpaque(true);
    rv.setBackground(java.awt.Color.white);

    SwingUtilities.forceComponentDimensions(rv, w, h);

    return rv;
}
}

protected void refresh() {
```

Variables

Name	Value
this	SelectStatementPanel$Stat...
rv	JLabel (id=111)
w	175
h	20

javax.swing.JLabel[,0,0,0x0,invalid,

Console | Tasks | Call Hierarchy | Search

IDE (1.6) (a) [Java Application] C:\Program Files\Java\jdk1.6.0_20\bin\javaw.exe (Oct 11, 2010 9:32:38 AM)

```
todo: RecentProjectsMenuModel handleMenuSelected
warning: createBufferedImageForUseAsColorBuffer size mismatch
project saved to:  C:\Documents and Settings\grosspa\My Documents\LookingGlass\MyProjects\apprenticeAndFan.lgp
C:\Documents and Settings\grosspa\My Documents\Research\Alice\git_workspace\ide\generated\gallery\assets\edu.wustl.cse.lookingglass.api
```

Hastings: an Output History Explorer Tool

What was happening at 10.41 seconds?

0.00 seconds — 28.00 seconds

The scene at 10.41 s.

Actions at 10.41 s.

- hoodeeny's Actions
  - in hoodeeny.sayTheMagicWords(lamp)
    - hoodeeny.lookAt(lamp)
- sam's Actions

Actions in: MyScene.run()

```
6  do together
     this.hoodeeny takeADeepBreath
     this.sam lookAt this.hoodeeny more
7  do together
     this.hoodeeny sayTheMagicWords sayWordsTo: this.lamp
     this.sam lookAt this.lamp more
8  this.lamp turnGreen
9  this.sam delay 0.5
10 this.camera getCloseUpOf this.sam , spatialRelation: FRONT_LEFT_OF , amount: 1.5 , duration: 0.0
11 this.camera move BACKWARD , 1.0 , duration: 0.0 more
12 this.camera move UP , 0.5 , duration: 0.0 more
13 this.camera move LEFT , 1.0 , duration: 0.0 more
14 this.sam delay 0.5
15 this.sam getPart Mouth turn FORWARD , 0.125 , duration: 0.25 more
16 this.sam delay 1.0
17 this.hoodeeny straightenUp more
18 this.sam getPart Mouth turn BACKWARD , 0.125 more
```

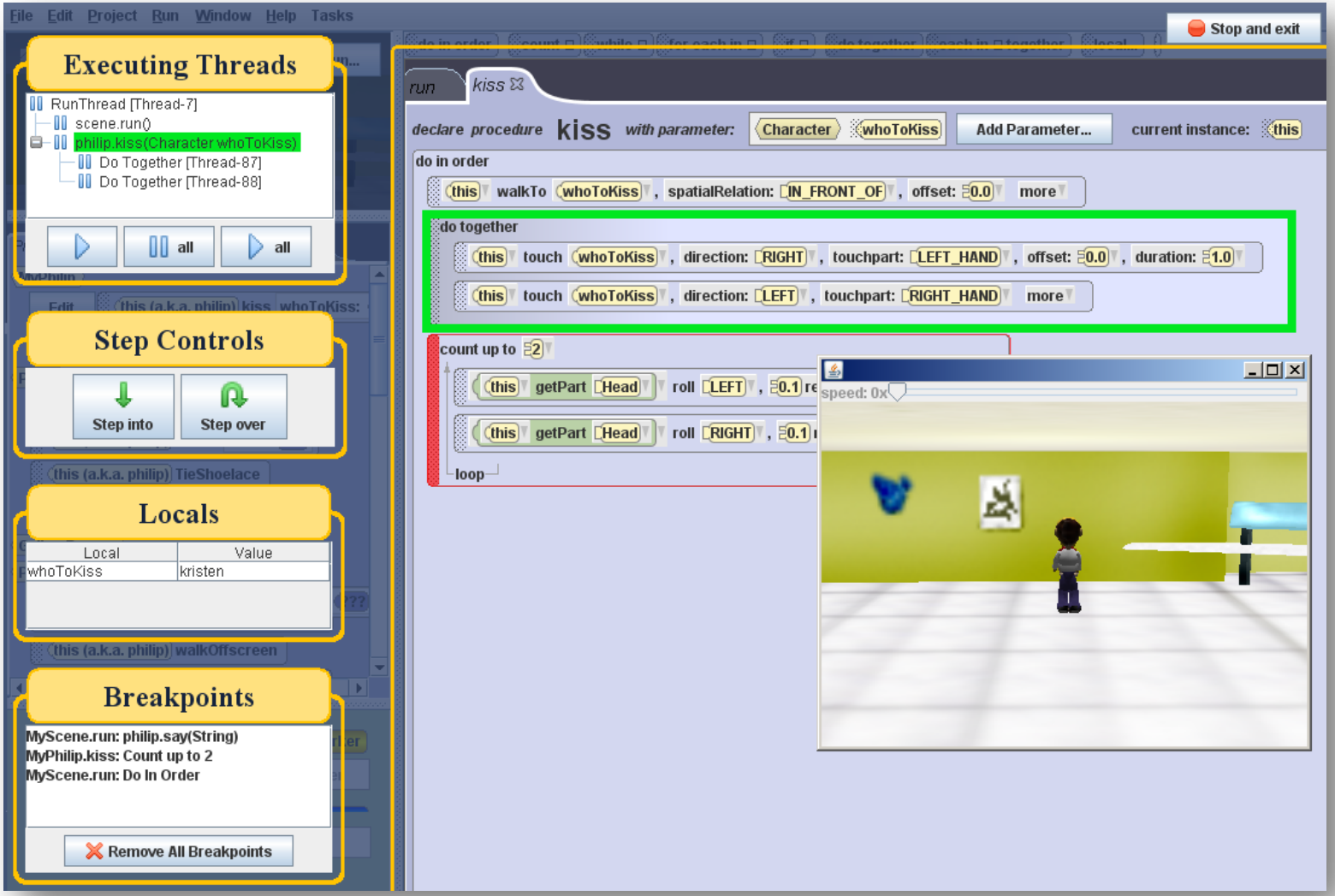
zoom into DoTogether

zoom into DoTogether

explore turnGreen

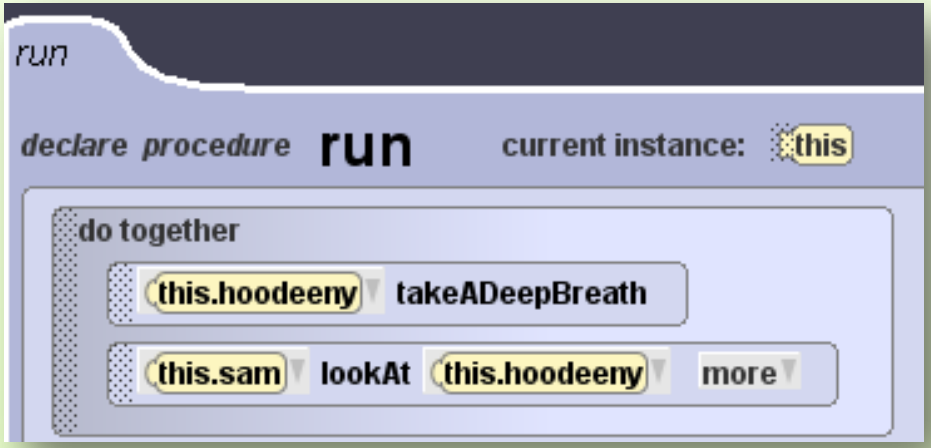
Do these Tools Adequately Assist Non-Programmers with Selection?

Evaluative Study: Hastings vs. a Debugger

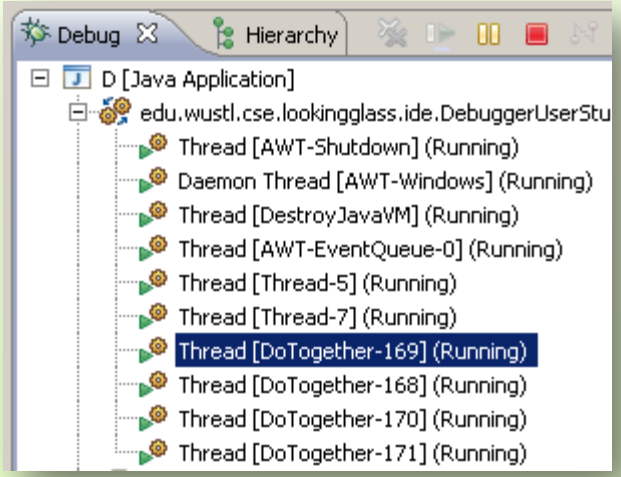


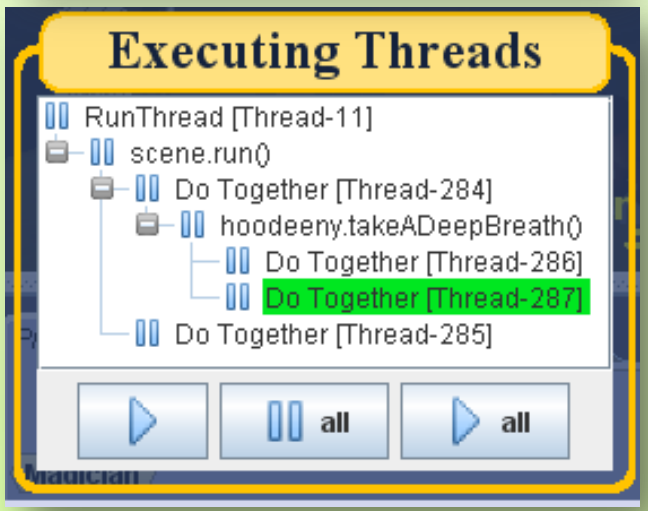
Looking Glass Debugger Threads

Looking Glass Program Code

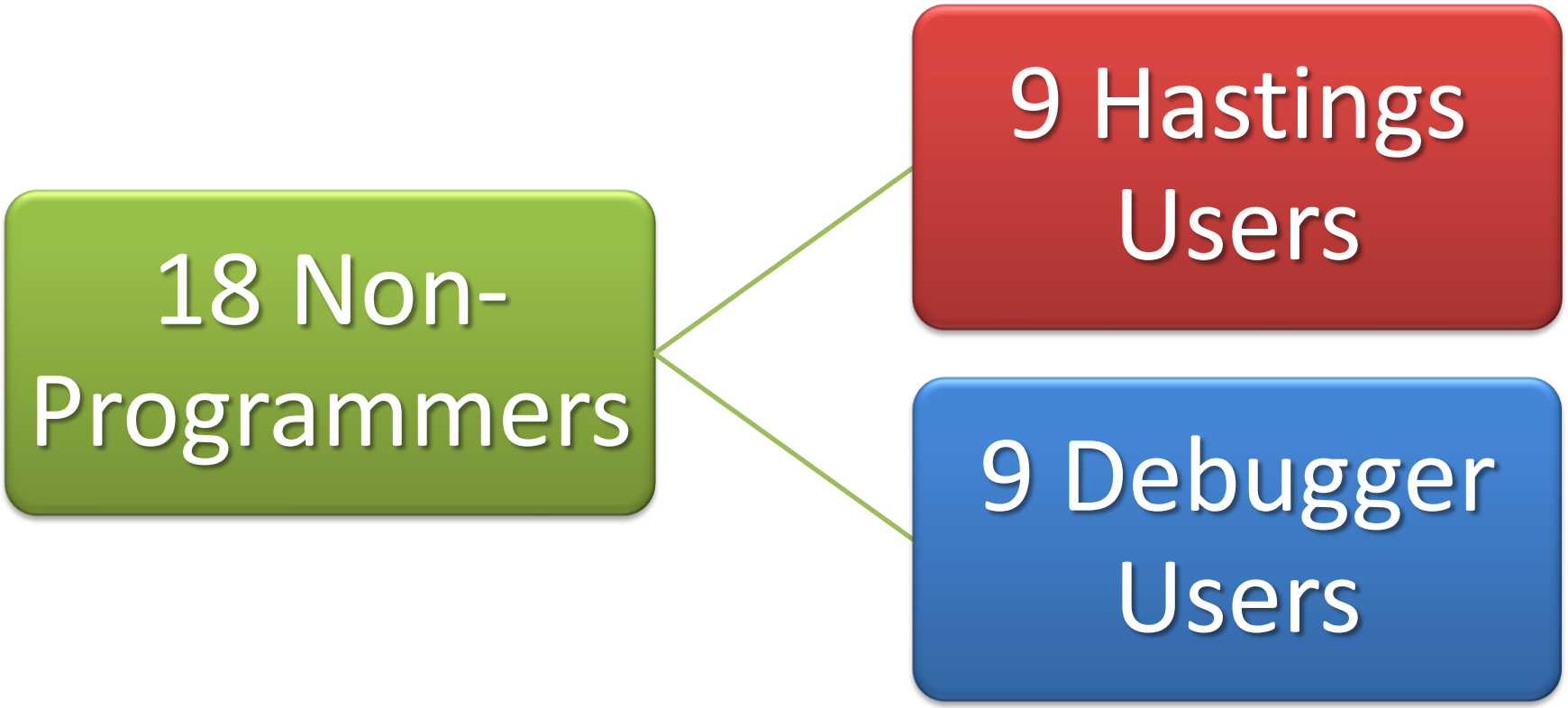


Thread Presentation



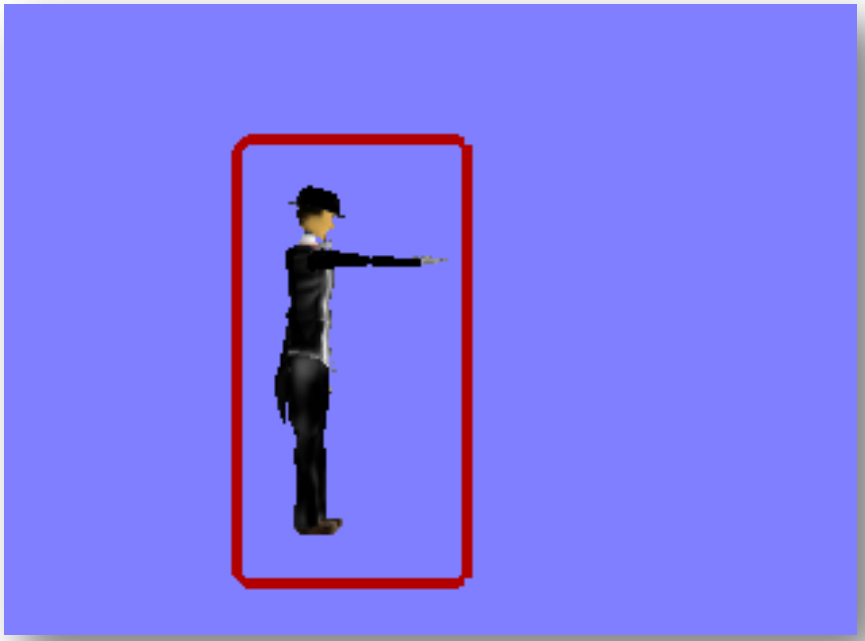


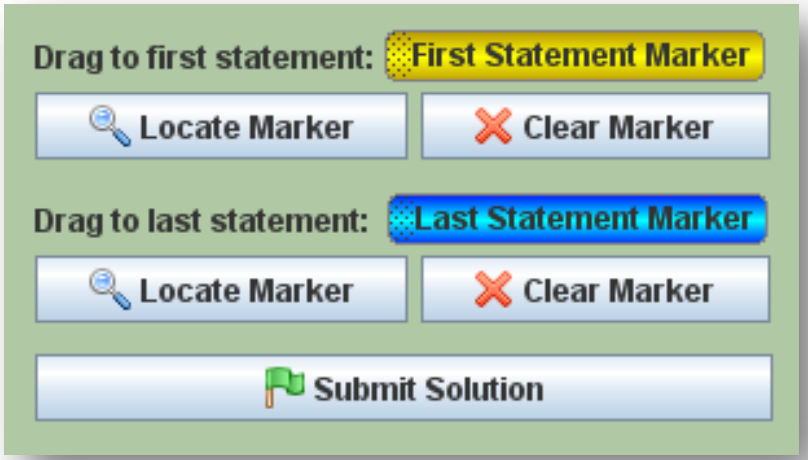
Evaluative Study: Hastings vs. a Debugger



Evaluative Study: Tasks

- Find and mark code corresponding to graphical output





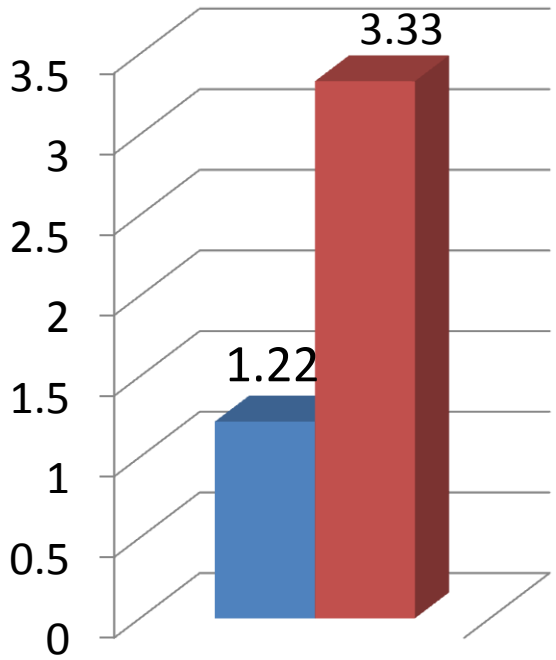
Do these Tools Adequately Assist Non-Programmers with Selection?

Quantitative Performance

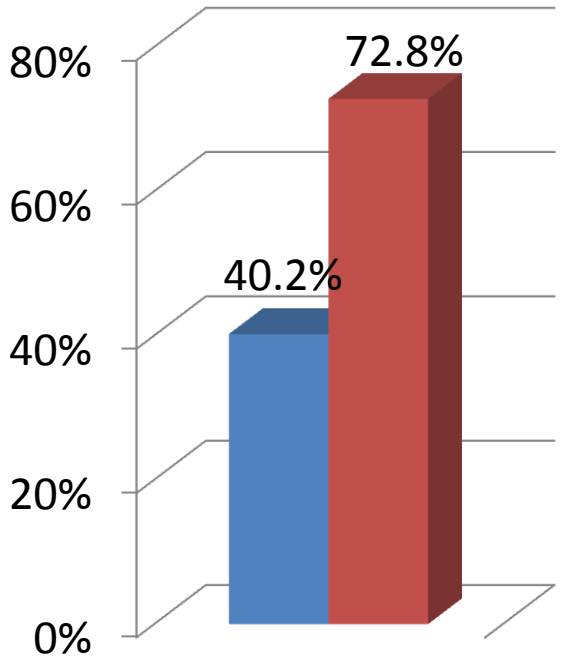
Qualitative Barriers

Quantitative Performance

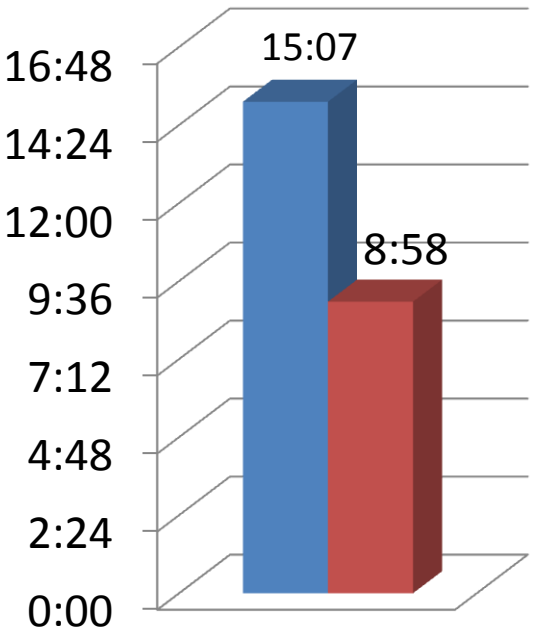
Average Tasks Completed
p < .01



Percentage of Correct Tasks
p < .05



Average Time on Task
p = .06



Debugger Hastings

Qualitative Barriers

Debugger Barriers

- Thread Misunderstanding
- Step Over Misinterpretation

Hastings Barriers

- Concurrency Ambiguity
- Playback Fidelity

A Qualitative Barrier

Concurrency Ambiguity [Hastings]

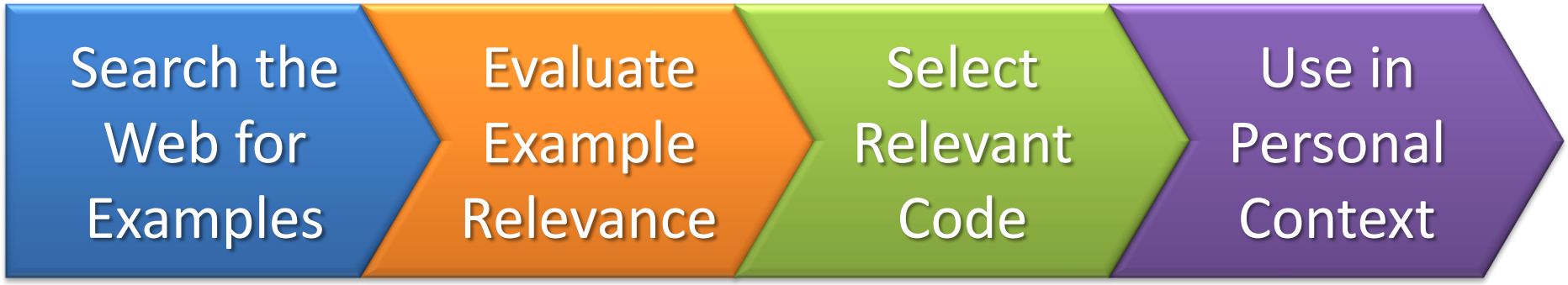
Do these Tools Adequately Assist Non-Programmers with Selection?

Room for improvement

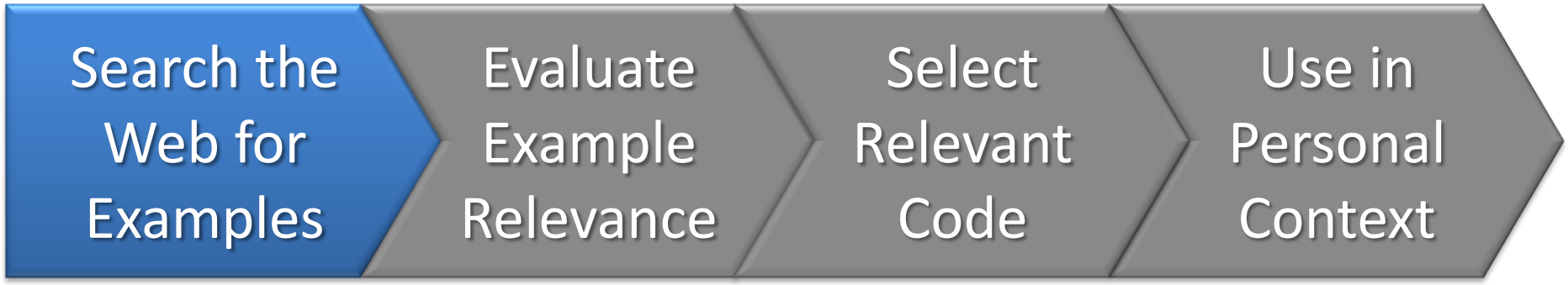
Future Design Guidelines

- Do not rely on execution abstractions
- Provide affordances for replaying code elements
- Use direct code interactions

An Independent Learning Process for End-user Programmers

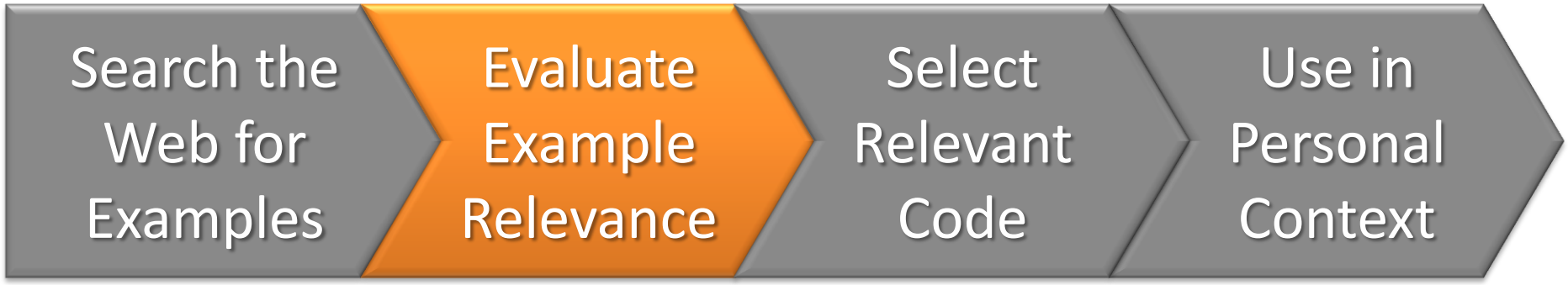


Further Challenges



- Searching with inexperienced end-users' evaluative vocabulary

Further Challenges

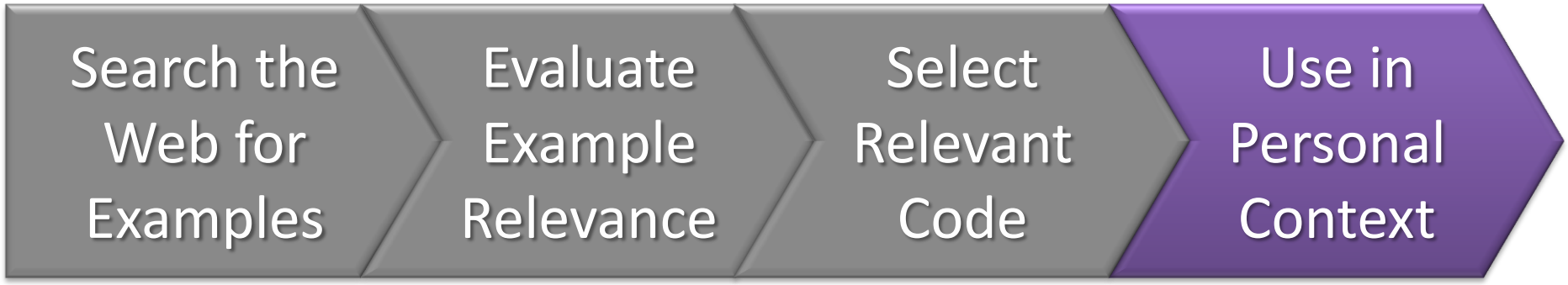


- Supporting efficient evaluation of many search results
- Enabling snippet evaluation

Further Challenges

- Supporting complex selection of code areas

Further Challenges



- Promoting learning from personally motivating examples
- Assisting with reuse and adaptation

Questions?

- {grosspa, ckelleher}@cse.wustl.edu
- http://lookingglass.wustl.edu

References

- [1] J. Brandt, P. J. Guo, J. Lewenstein, M. Dontcheva, and S. R. Klemmer. *Two studies of opportunistic programming: interleaving web foraging, learning, and writing code*. In Proc. of CHI, pages 1589–1598, 2009.
- [2] M. B. Rosson, J. Ballin, and J. Rode. *Who, what, and how: A survey of informal and professional web developers*. In Proc. of VL/HCC, pages 199–206, 2005.
- [3] P. Gross and C. Kelleher. *Non-programmers identifying functionality in unfamiliar code: strategies and barriers*. JVLC, 21 (5):263–276, Aug. 2010.