

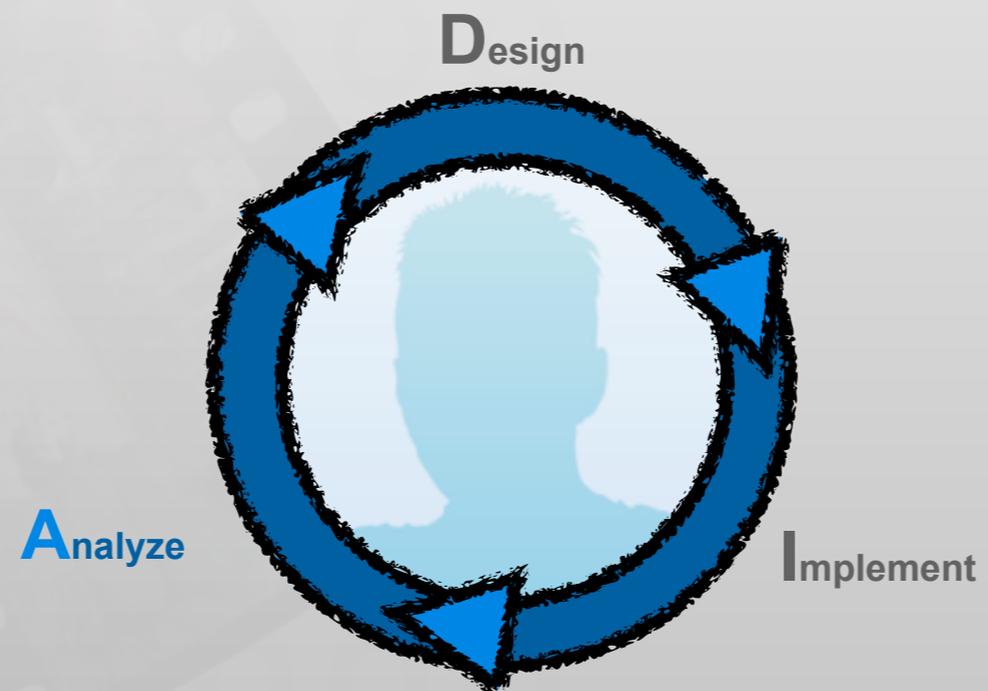
# Research in Coding and IDEs

*Jan-Peter Krämer  
Media Computing Group  
RWTH Aachen University*

<http://hci.rwth-aachen.de/cthci>

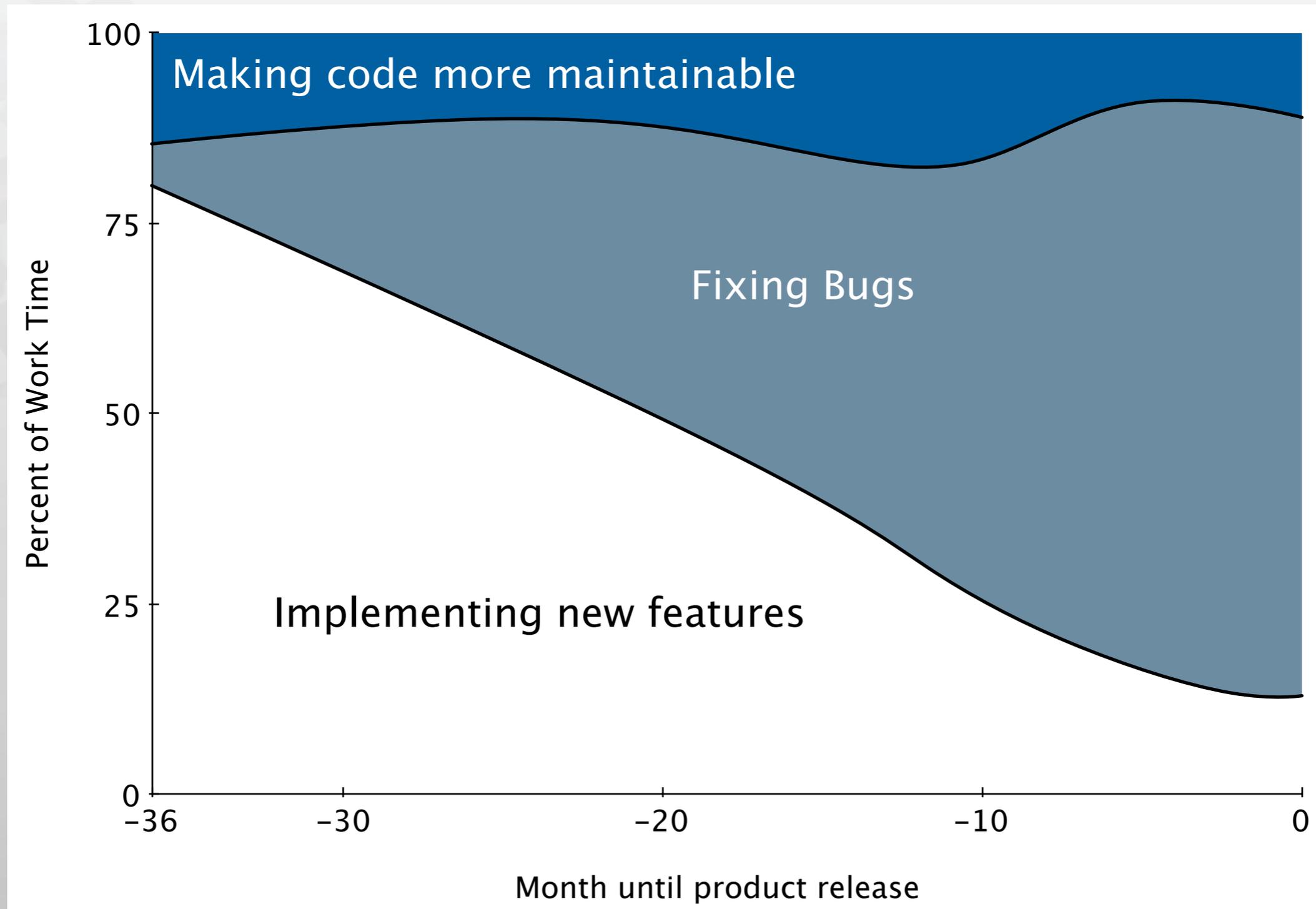


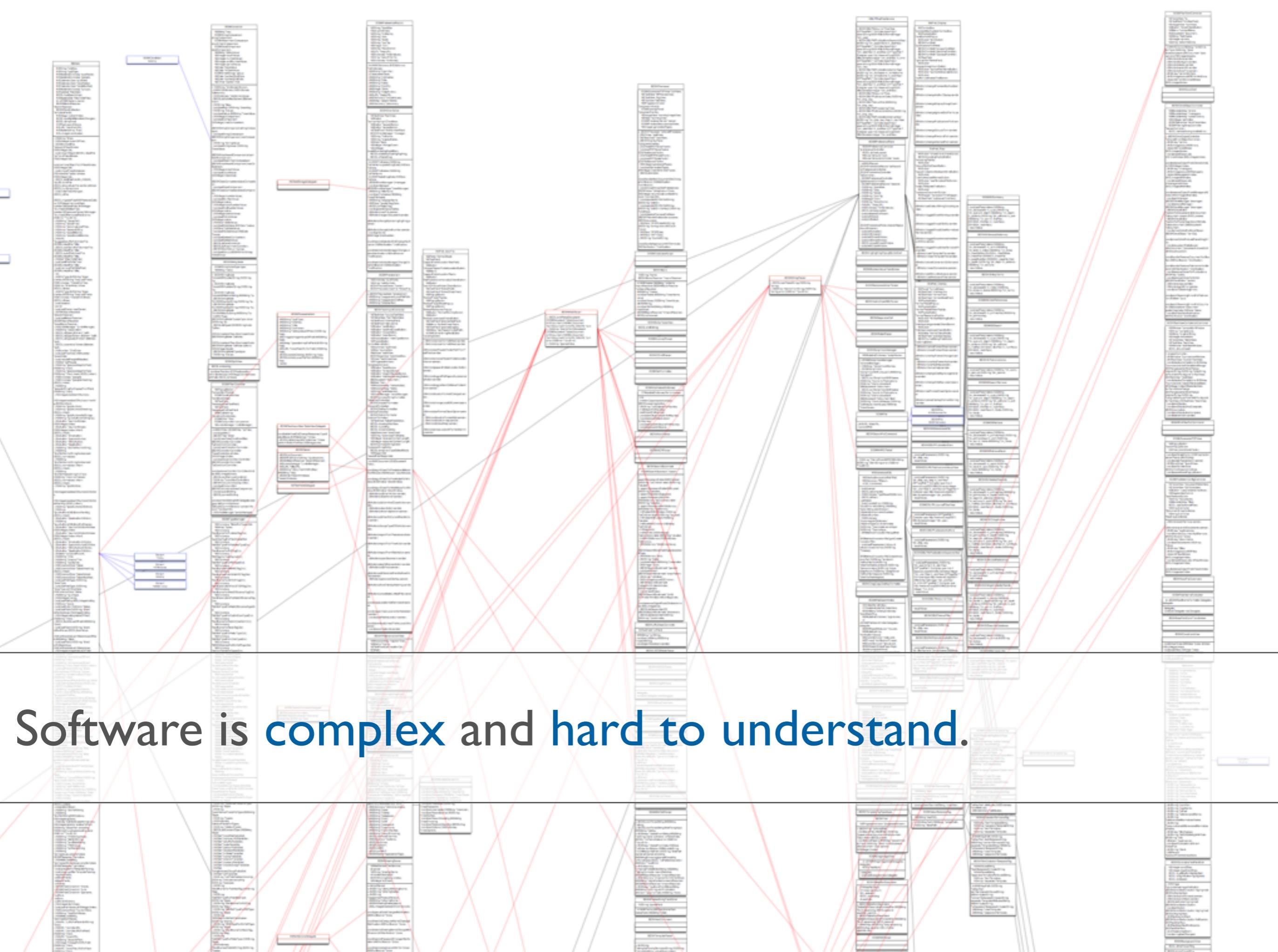
# Status Quo



# Time in Software Development

[LaToza2006, Maintaining mental models: a study of developer work habits]





Software is **complex** and **hard to understand**.

# Task context

```
#import <Cocoa/Cocoa.h>

extern NSString *BDSKFileKey;
extern NSString *BDSKFilePublicationKey;
extern NSString *BDSKFileOldPathKey;
extern NSString *BDSKFileNewPathKey;
extern NSString *BDSKFileStatusKey;
extern NSString *BDSKFileFlagKey;
extern NSString *BDSKFileFlxKey;

@class BibDocument;

enum {
    BDSKLastError = 0,
    BDSKSourceFilesNotExistErrorMask = 1,
    BDSKTargetFileExistsErrorMask = 2,
    BDSKCannotMoveFileErrorMask = 4,
    BDSKCannotRemoveFileErrorMask = 8,
    BDSKCannotResolveAliasErrorMask = 16,
    BDSKCannotCreateParentErrorMask = 32,
    BDSKIncompleteFieldsErrorMask = 64
};

enum {
    BDSKInitialAutoFileOptionMask = 1,
    BDSKCheckCompleteAutoFileOptionMask = 2,
    BDSKForceAutoFileOptionMask = 4
};

@interface BDSKFilter : NSWindowController {
    IBOutlet NSProgressIndicator *progressIndicator;
}

+ (BDSKFilter *)sharedFilter;

- (void)autoFileLinkedFilesFromDocument:(BibDocument *)doc check;

- (void)movePapers:(NSArray *)papers fromDocument:(BibDocument *)doc toPath:(NSString *)path;

- (void)movePath:(NSString *)path toPath:(NSString *)newPath;

@end

@interface NSFileManager (BDSKFilterExtensions)

- (void)movePath:(NSString *)path toPath:(NSString *)newPath;

@end
```

```
... *)paperInfos forField:
... (NSInteger)mask{
... NSFileManager defaultManager
... [paperInfos count];
... nil;
... nil;
... foDicts = [NSMutableDictionary
... infoDicts = [NSMutableDictionary
... info = nil;
```

```
NSMutableArray Class Reference
arrayWithCapacity:
Creates and returns an NSMutableArray object with enough allocated memory to initially hold a given number of objects.
Parameters:
numbers
```

### Hacking BibDesk

Michael O. McCracken

#### About This Document

This document is a place for notes explaining various tricky things about how BibDesk is organized.

#### Class Hierarchy notes

There is separate (but incomplete) HTML documentation generated from inline comments about each class. Refer to that for method and function definitions. This section only contains conceptual notes and other miscellany.

#### BibEditor

#### BibDocument

#### Toolbar setup

The BibDocument toolbar is kind of complicated. It is set up mostly in the BibDocumentView.Toolbar.m file.

To add a toolbar item, you will need to do the following steps:

1. Add a toolbar identifier. At the beginning of BibDocumentView.Toolbar.m, add something like this:  
static NSString \*MyNewToolbarIdentifier = @"Some string that identifies the new item and says toolbar identifier too";
2. Add a call to add(toolbarItem) in the -setupToolbar method for your new toolbar item. Take a look at add(toolbarItem) to figure out what to put in the call. It's just a convenience func.
3. Double check that if you're adding a view and not an image, the view is an actual drawable NSView. If it isn't, go wrap it in an NSBox and add that instead.
4. Add MyNewToolbarIdentifier to the array returned by toolbarAllowedItemIdentifiers. If it should go in the default one, add it to toolbarDefaultItemIdentifiers too.
5. If the item is a view and not an image you will also need to keep around the toolbarItem we get from toolbarWillAddItem. see the quicksearch field for an example. this involves adding an instance variable.

#### BibTypeManager

#### BibItem

#### File I/O

#### File Types

Different file types can contain different entry types, and fields within entries may not be named the same. The BibTypeManager class abstracts the file type.

It provides the available entry types for a given file type, and the optional and required fields for each entry. It is essentially an interface to the property list file TypesInfo.plist, which is located in the application package's Resources folder.

#### The TypesInfo.plist file

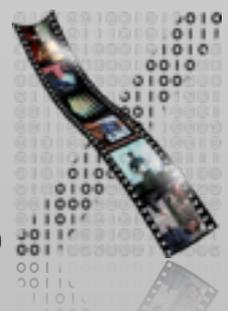
This file defines the data used by the BibTypeManager class.

It has four top-level keys:

- AllRemovableFieldNames is an Array of field names that are used by the BibItem class when changing pub types of a BibItem (in the method -makeType). All names in this array that do not contain data and do not appear in the new type are removed. This array should include all "standard" field names.
- DefaultFieldNames is an Array of field names that are used by the BibItem class when creating a new BibItem. It should include all "standard" field names.
- FileTypes is a dictionary of information about different file types. The dictionary contains the file extension and the default publication type for that file type.
- TypeInfo.plist explains which of the types in FieldInfo.plist are applicable for a given file type. Note that this is only normally used in some places. Making this doesn't mean that I'm overloading BibDesk to support and discriminate between many file formats actually. I

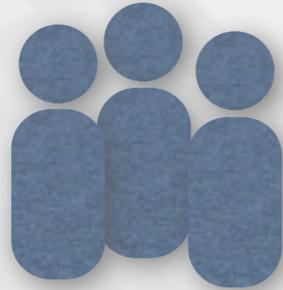
- What is relevant information?
- What strategies are applied to find information?

```
pub = [paperInfo valueForKey:BDSKFilterPublicationKey];
```



# Models for Developer Strategies

[Ko2006, An Exploratory Study of How Developers Seek, Relate, and Collect Relevant Information during Software Maintenance Tasks]



31 Professional Java Developers



5 Maintenance tasks  
(3 Bugs, 2 Enhancements)

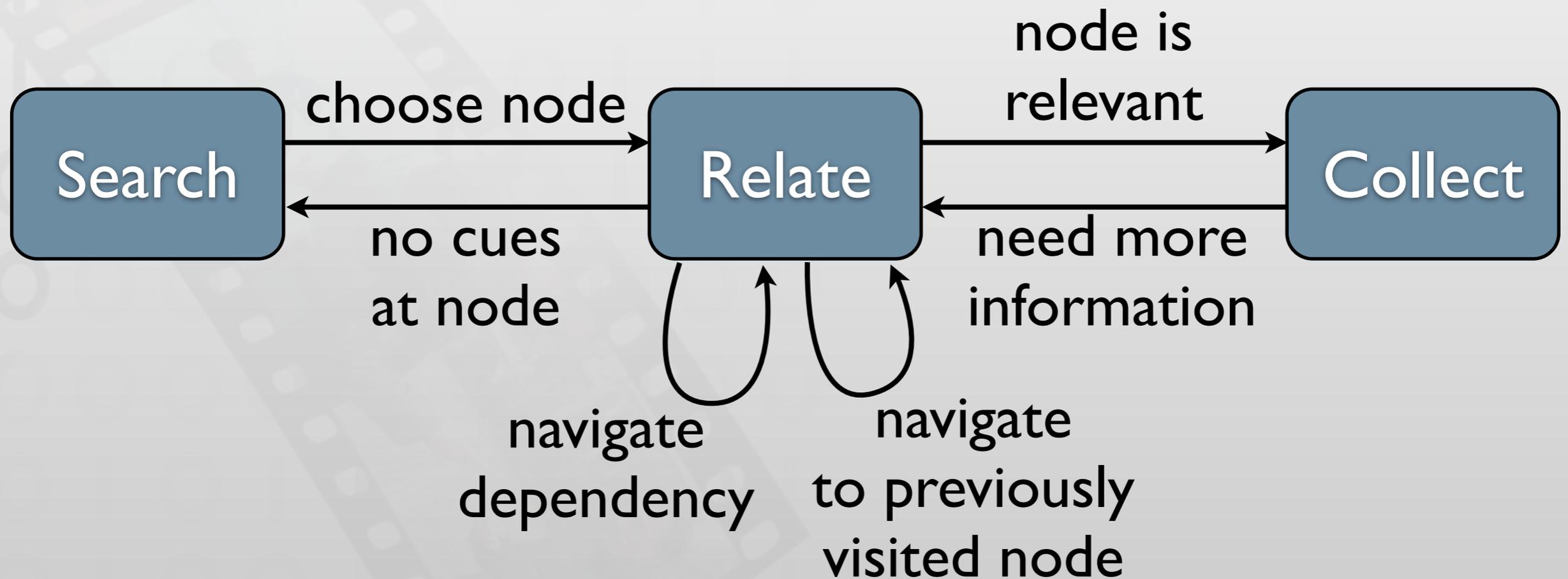


500 SLOC Java Paint  
Application



# Models for Developer Strategies

[Ko2006, An Exploratory Study of How Developers Seek, Relate, and Collect Relevant Information during Software Maintenance Tasks]

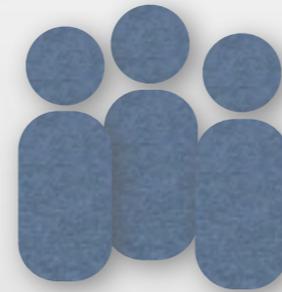


# Models for Developer Strategies

[Sillito2008, Asking and Answering Questions during a Programming Change Task]



9 experienced developers (pair programming)



16 developers from industry



1 of 5 maintenance tasks per session



Real world change task



ArgoUML  
60k SLOC

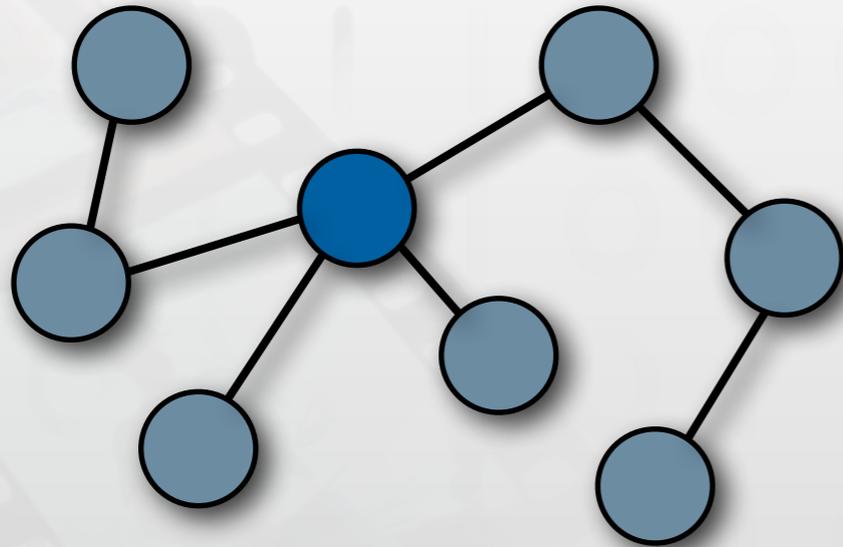


Real world sour code

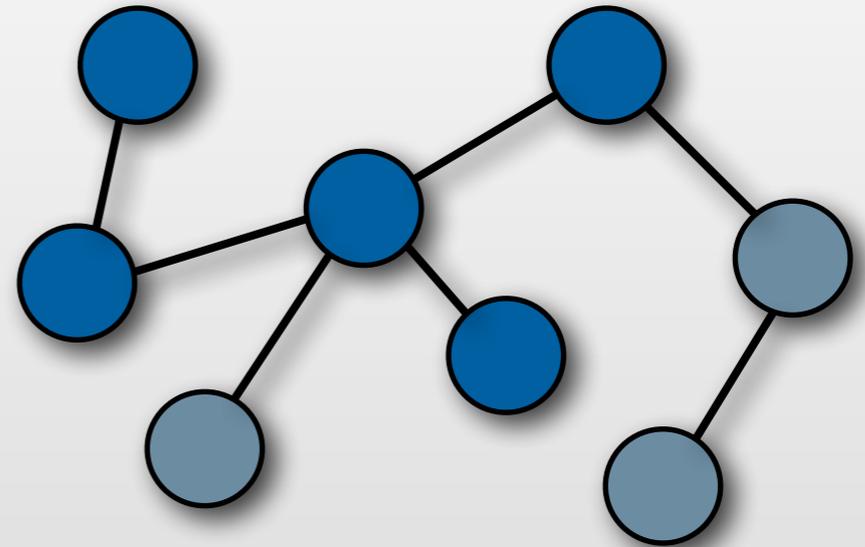


# Models for Developer Strategies

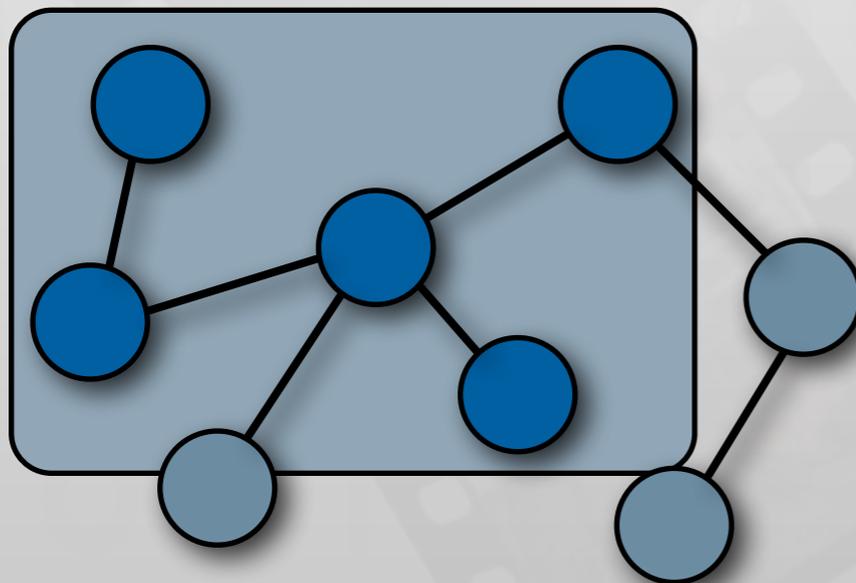
[Sillito2008, Asking and Answering Questions during a Programming Change Task]



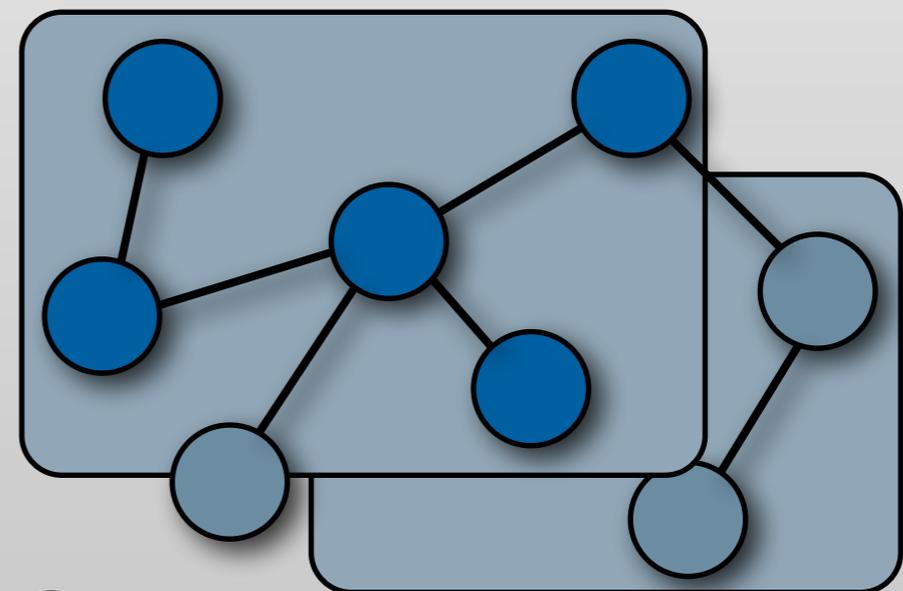
Finding focus points



Expanding focus points



Understanding a subgraph



Questions over groups of subgraphs



Package Explorer showing the project structure:

- org.jhotdraw
  - org.jhotdraw.applet
  - org.jhotdraw.application
  - org.jhotdraw.contrib
    - org.jhotdraw.contrib.dnd
    - org.jhotdraw.contrib.html
    - org.jhotdraw.contrib.zoom
    - org.jhotdraw.figures
    - org.jhotdraw.framework
    - org.jhotdraw.images
    - org.jhotdraw.samples
      - org.jhotdraw.samples.javadraw
        - AnimationDecorator.java
        - Animator.java
        - BouncingDrawing.java
        - FollowURLTool.java
        - JavaDrawApp.java
        - JavaDrawApplet.java
        - JavaDrawViewer.java
        - MySelectionTool.java
        - PatternPainter.java
        - URLTool.java
        - JavaDrawAppletHelp.html
      - org.jhotdraw.samples.javadraw.sam
      - org.jhotdraw.samples.minimap
      - org.jhotdraw.samples.net
      - org.jhotdraw.samples.nothing
      - org.jhotdraw.samples.pert
      - org.jhotdraw.samples.pert.images
      - org.jhotdraw.standard
      - org.jhotdraw.test
        - org.jhotdraw.test.contrib
        - org.jhotdraw.test.figures
        - org.jhotdraw.test.framework
        - org.jhotdraw.test.samples.javadraw
        - org.jhotdraw.test.samples.minimap
        - org.jhotdraw.test.samples.net
        - org.jhotdraw.test.samples.nothing
        - org.jhotdraw.test.samples.pert
        - org.jhotdraw.test.standard
        - org.jhotdraw.test.util
        - org.jhotdraw.test.util.collections.jdk
        - org.jhotdraw.test.util.collections.jdk
        - org.jhotdraw.util
        - org.jhotdraw.util.collections.jdk11
        - org.jhotdraw.util.collections.jdk12

```

@(#)DesktopEvent.java

package org.jhotdraw.contrib;

import org.jhotdraw.framework.DrawingView;

/**
 * @author C.L.Gilbert <dnoyeb@users.sourceforge.net>
 * @version <CURRENT_VERSION$>
 */
public class DesktopEvent extends EventObject {
    private DrawingView myDrawingView;

    /**
     * Some events require the previous DrawingView (e.g. when a new DrawingView
     * is selected).
     */
    private DrawingView myPreviousDrawingView;

    public DesktopEvent(Desktop newSource, DrawingView newDrawingView) {
        this(newSource, newDrawingView, null);
    }

    public DesktopEvent(Desktop newSource, DrawingView newDrawingView, DrawingView newPreviousDV) {
        super(newSource);
        setDrawingView(newDrawingView);
        setPreviousDrawingView(newPreviousDV);
    }

    private void setDrawingView(DrawingView newDrawingView) {
        myDrawingView = newDrawingView;
    }

    public DrawingView getDrawingView() {
        return myDrawingView;
    }

    private void setPreviousDrawingView(DrawingView newPreviousDrawingView) {
        myPreviousDrawingView = newPreviousDrawingView;
    }

    public DrawingView getPreviousDrawingView() {
        return myPreviousDrawingView;
    }
}

```

Outline view showing the class structure:

- org.jhotdraw.contrib
  - import declarations
  - DesktopEvent
    - myDrawingView : DrawingView
    - myPreviousDrawingView : Drawl
    - DesktopEvent(Desktop, Drawing
    - DesktopEvent(Desktop, Drawing
    - setDrawingView(DrawingView) :
    - getDrawingView() : DrawingView
    - setPreviousDrawingView(Drawin
    - getPreviousDrawingView() : Draw

Problems, Javadoc, Declaration, Call Hierarchy tabs. Call Hierarchy view is active:

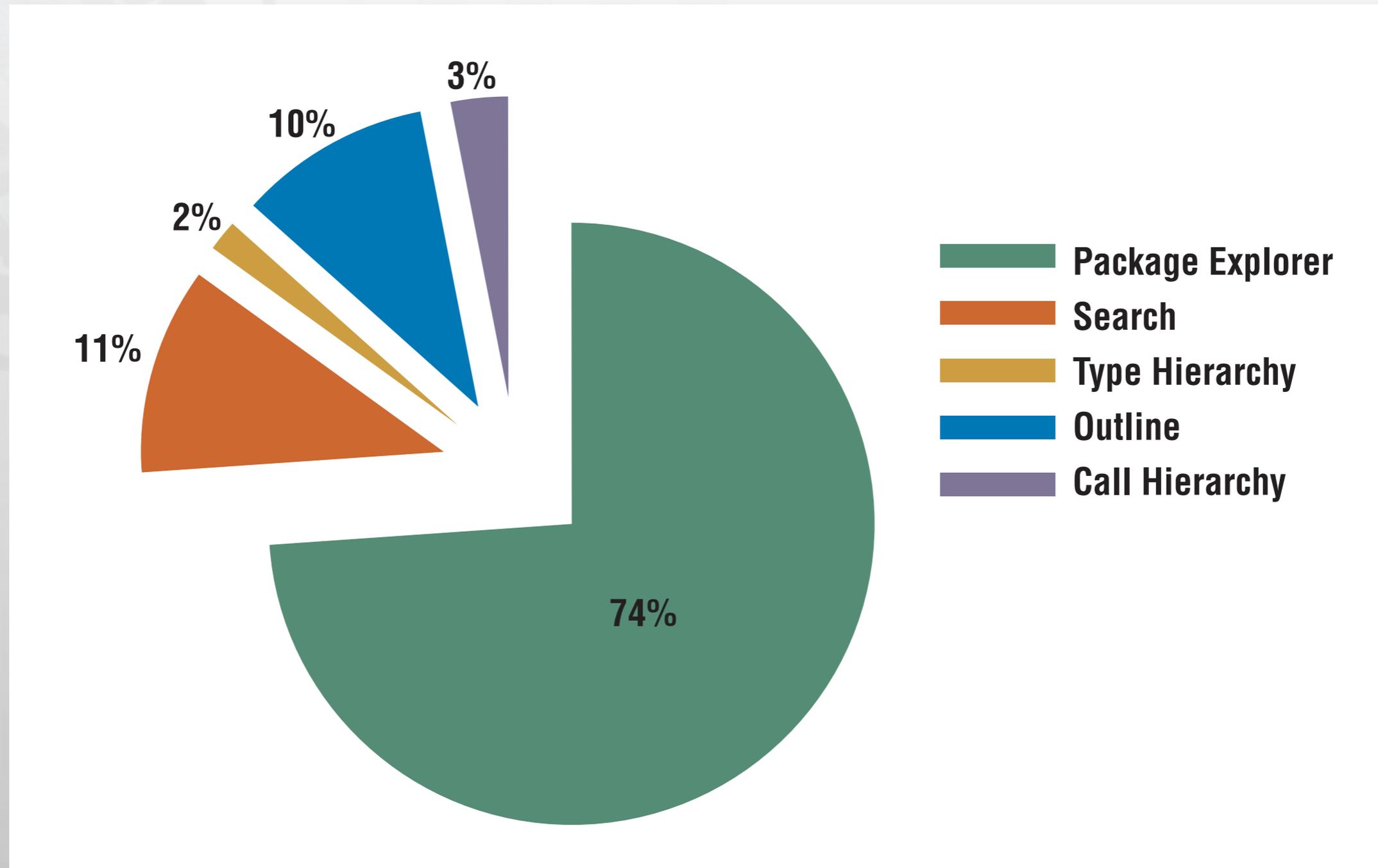
Calls from 'DesktopEvent(Desktop, DrawingView, DrawingView)' - in workspace

- DesktopEvent(Desktop, DrawingView, DrawingView) - org.jhotdraw.contrib.DesktopEvent
  - EventObject(Object) - java.util.EventObject
  - setDrawingView(DrawingView) : void - org.jhotdraw.contrib.DesktopEvent
  - setPreviousDrawingView(DrawingView) : void - org.jhotdraw.contrib.DesktopEvent

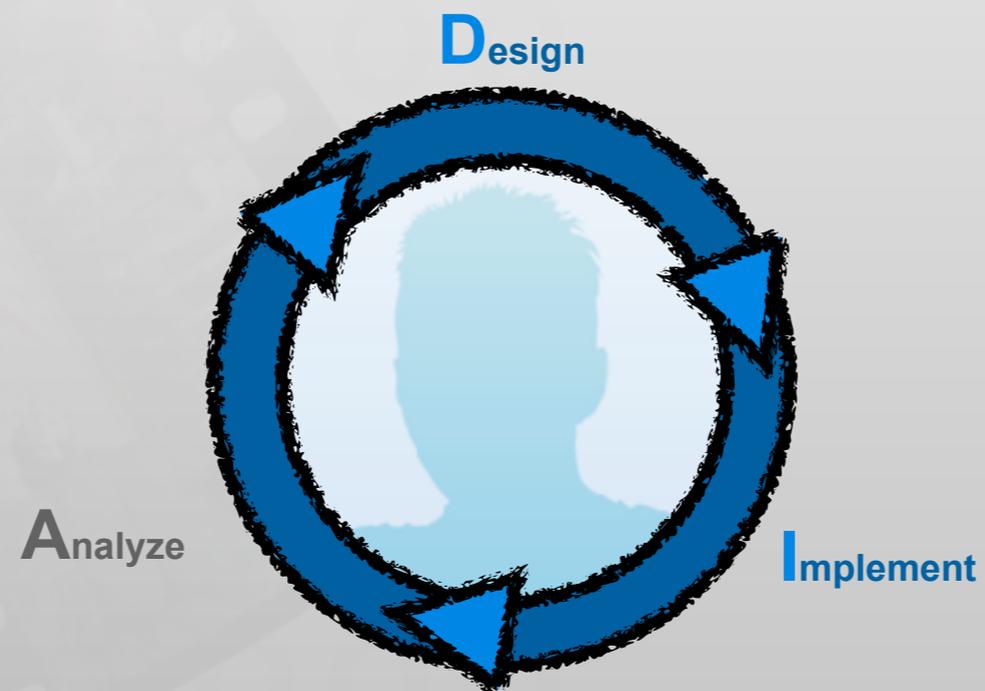
Line	Call

# Tools Used in Eclipse

[Murphy2006, How Are Java Software Developers Using the Eclipse IDE?]



# Easing Access to Task Context



The screenshot displays the Eclipse IDE interface for managing bug reports. The main window is titled "Bug 216640" and contains the following elements:

- Package Explorer (Left):** Shows the project structure for "org.eclipse.myllyn.bugzilla.core" and "org.eclipse.myllyn.bugzilla.ui".
- Bug Report View (Center):**
  - Title:** Attach error log entries into new bug reports
  - Metadata:** P5 Status: NEW Bug: 216640 Opened: Jan 25, 2008 Modified: Jan 29, 2008 2:32 AM
  - Attributes:** (Collapsible)
  - Attachments (0):** (Collapsible)
  - Description:** (Collapsible)
  - Comments (1):**
    - 1: Mik Kersten, Jan 29, 2008 1:54 AM
  - New Comment:** Looking into this now, see also related [bug 124224](#).
  - Actions:**
    - Leave as NEW
    - Accept (change status to ASSIGNED)
    - Resolve as: FIXED
    - Duplicate of: [Empty]
    - Reassign to: mik
    - Reassign to default: mik.kersten@tasktop.com, mylyn-inbox@eclipse.org
    - Submit
  - People:**
    - Assigned to: Mylyn Inbox <mylyn-inbox@eclipse.org>
    - Reporter: Willian Mitsuda <wmitsuda@eclipse.org>
    - QA Contact: [Empty]
    - Add CC: jacek.pospychala@pl.ibm.com, mik.kersten@tasktop.com
- Task List (Right):**
  - Mylyn - Mik [Bugzilla]
    - 162335: [context] Allow "Make Landi"
    - 267524: create Mylyn 3.2 release plan
  - Mylyn - Rob [Bugzilla]
    - 124224: Allow to edit priorities in the task list
  - Project - Steffen [Trac]
    - 83: filter sort results
    - 84: upgrade ws api
    - 5: address web site nits
  - Target - Rob [Atlassian]
    - PLE-290: logged in as wrong user wh
    - PLE-327: Documentation for Eclipse
    - PLE-172: Create Crucible review with
    - PLE-238: add an action to a cha
- Synchronize (Bottom Right):** Change Sets for CVS (Workspace)
  - Allow to edit priorities in the task list
  - streamline task attachments

[Kersten2006, Using Task Context to Improve Programmer Productivity]

# Recommender Tools

[Singer2005, NavTracks: supporting navigation in software maintenance]

[DeLine2005, Easing program comprehension by sharing navigation data]

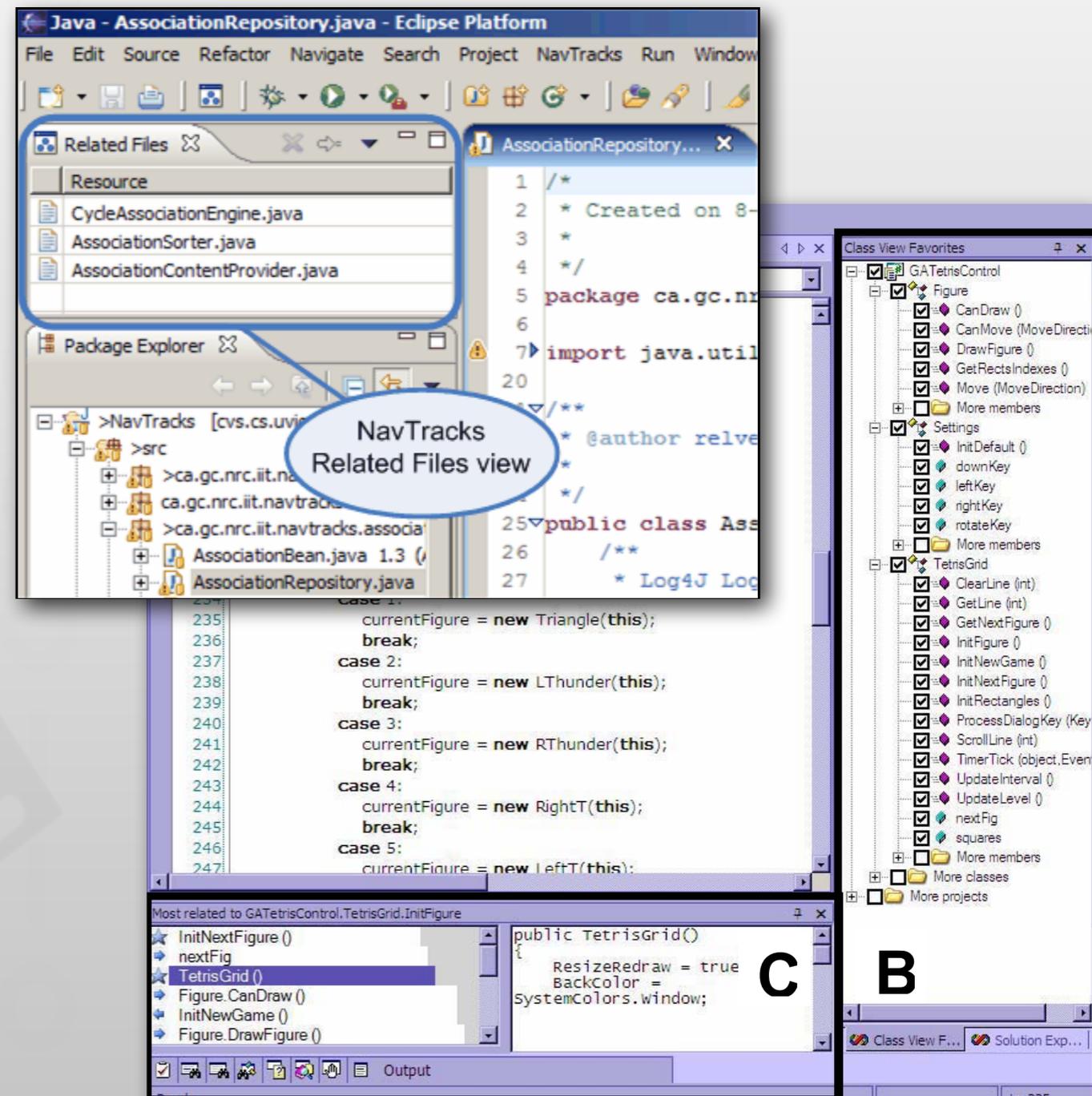
[Čubranic'2005, Hipikat: recommending pertinent software development artifacts]

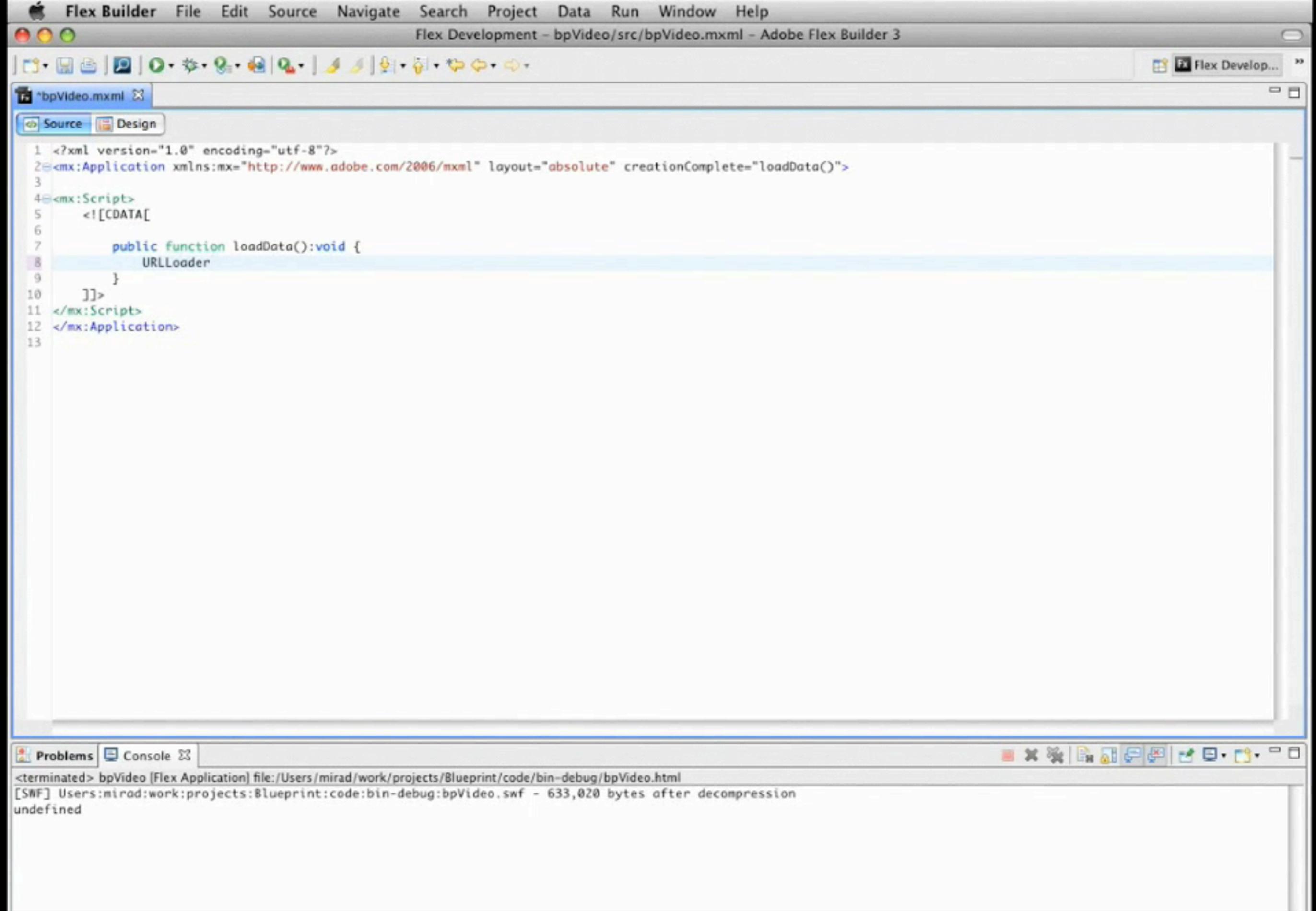
- Calculate a Degree of Interest for source code elements based on:

- reading history
- editing history
- history of other team members
- information from version control systems

- Remaining Problems:

- Still only text-based visualization
- Recommendations for irrelevant code are still irrelevant

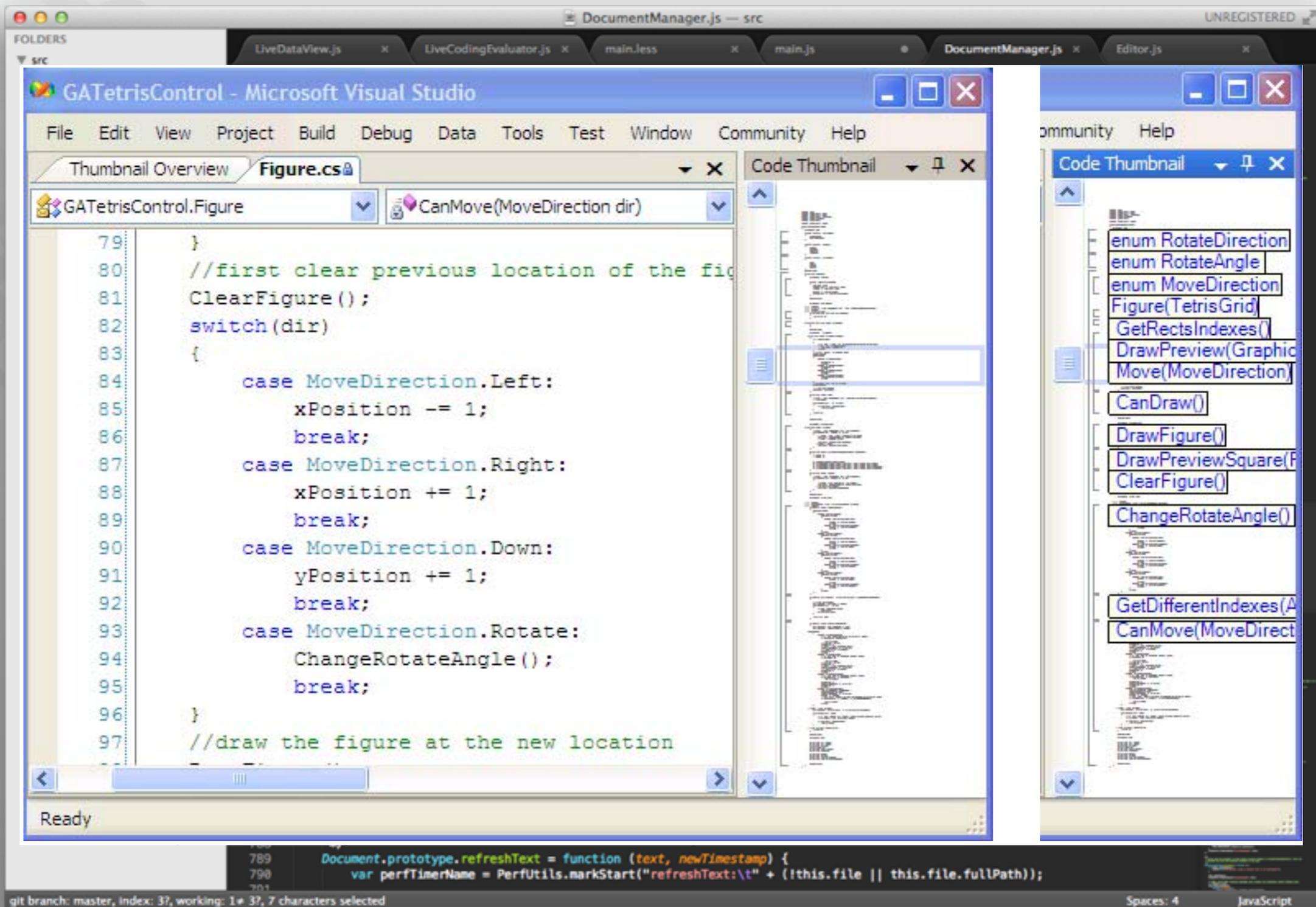


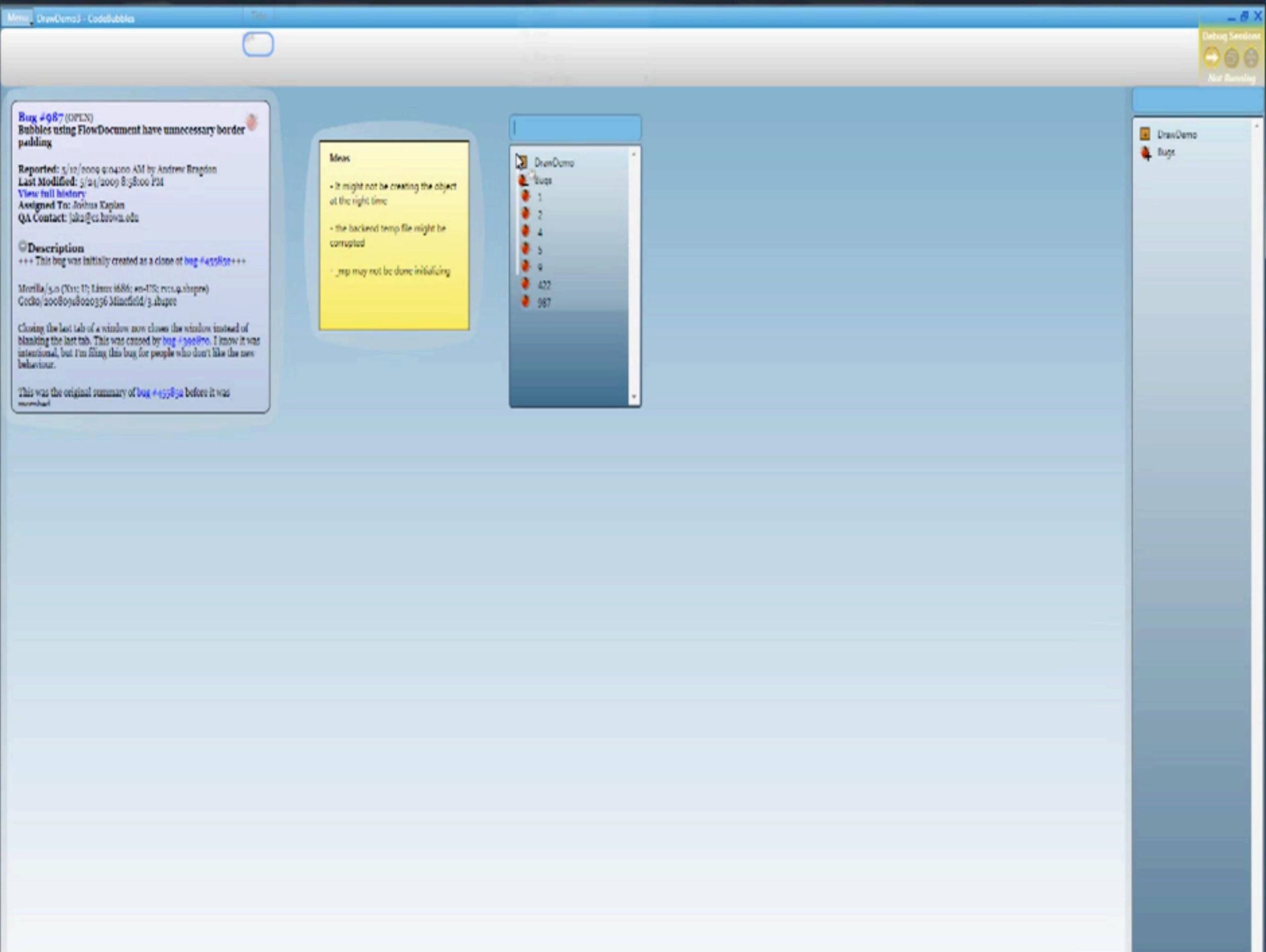


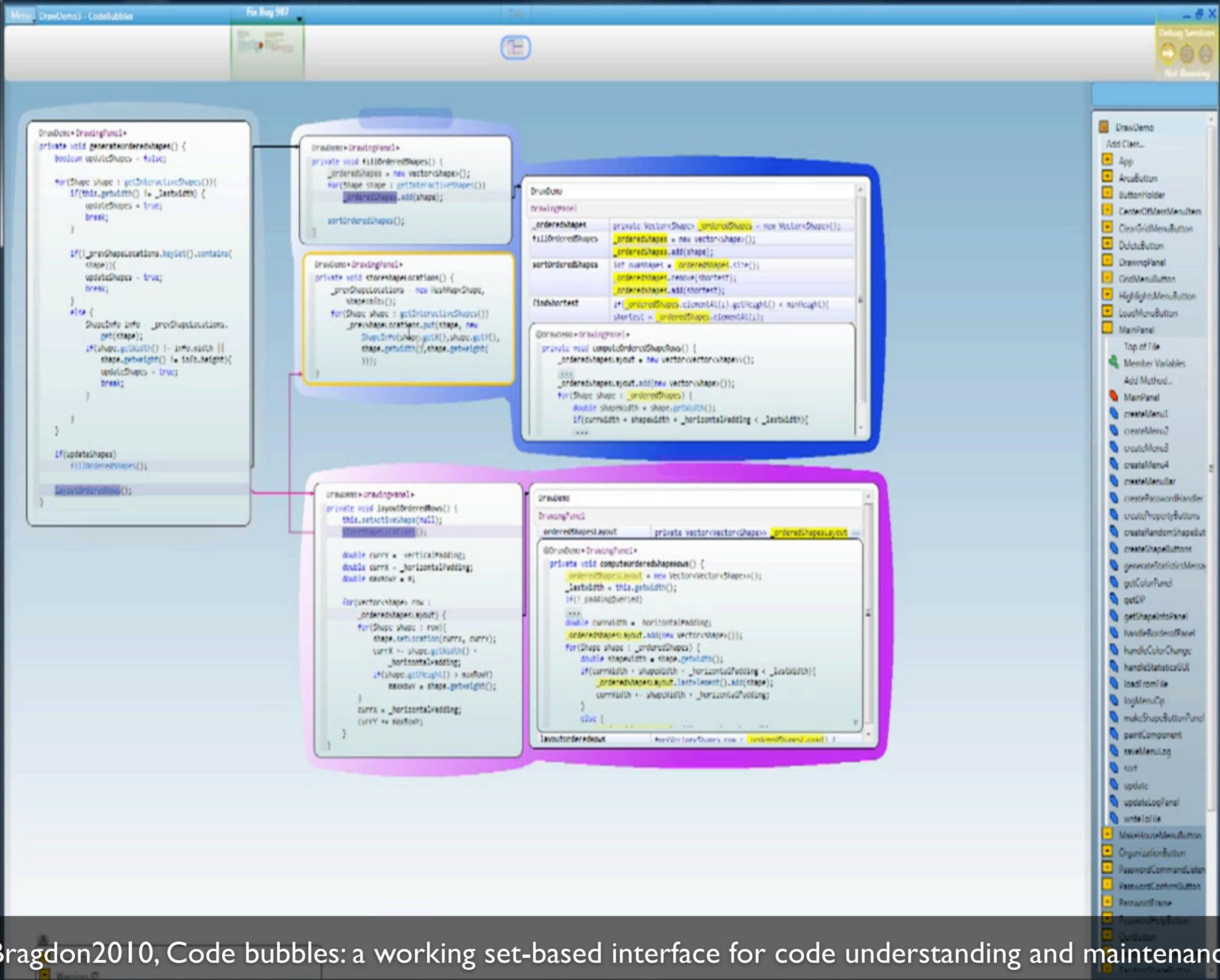
[Brandt2010, Example-centric programming: integrating web search into the development environment]

# Changing the Presentation

[DeLine2006, CodeThumbnail2, Using Special Memory to Navigate Source Code]







[Bragdon2010, Code bubbles: a working set-based interface for code understanding and maintenance]

Changes to UI elements

```

@Demo+MainPanel+
lic string[] generateStatisticsMessages(){
String msg1 =
    "here is currently s shape on the screen"

string[] messages = new string[1];
messages[0] = msg1;

return messages;
}

```

```

@Demo+MainPanel+
lic void handleBorderOfPanel(JPanel panel,
color c){
panel.setBorder(new javax.swing.border.
MatteBorder(5,5,5,5,c));

init();
}

```

```

@Demo+MainPanel+
lic void logKeyPress(String action)
_actionList.add(action);
writeMenuLog();
}

```

```

DrawDemo+MainPanel+
public void handleStatisticsGUI(String[]
messages) {
_statsButton = StatsButton.getInstance(
messages);
}

```

```

DrawDemo+StatsButton+
protected void handleClick(){
string displayText = "";
displayText += _messages[0];
updateLogPanel(displayText);
}

```

```

DrawDemo+MainPanel+
public ShapeButton[] createShapeButtons(){
ShapeButton[] shp = new ShapeButton[4];
shp[0]=ShapeButton.getInstance("Line",
this);
shp[1] = ShapeButton.getInstance(
"Round Rectangle",this);
shp[2]= ShapeButton.getInstance("Ellipse",
this);
shp[3]=ShapeButton.getInstance("Pie Shape",
this);

_holder = new ButtonHolder(_lines);
return shp;
}

```

```

DrawDemo+MainPanel+
public static void test()
{
}

```

Toolbar Setup Code

```

DrawDemo+MainPanel+
private static final long serialVersionUID =
1L;
private JButton _line0;
private JButton _statsButton;
private JButton _deleteShape;
private ButtonHolder _holder;
public LinkedList<String> _actions = new
LinkedList<String>();
private DrawingPanel _dr;
private ShapeButtonPanel _shapeButtonPanel;

```

```

@DRAWINGPANEL
buttonholder
_currentSelection private ShapeButton _currentSelection, _nextSelection;
setSelected public void setSelected(ShapeButton button){
DrawingPanel
isSpecifyDimensions return ShapeButton.isSpecifyDimensions();
MainPanel
_line0 private ShapeButton _line;
MainPanel
MainPanel [ShapeButton] shapeButtons = this.createShapeButtons();
createShapeButtons public ShapeButton[] createShapeButtons(){
ShapeButton[] shp = new ShapeButton[4];
ShapeButton shp = new ShapeButton(4);
shp[0]=ShapeButton.getInstance("Line",this);
shp[1] = ShapeButton.getInstance("Round Rectangle",this);
shp[2]= ShapeButton.getInstance("Elllipse",this);
shp[3]=ShapeButton.getInstance("Pie Shape",this);
--ShapeButtonPanel public JPanel makeShapeButtonPanel(ShapeButton[] button){
ShapeButton

```

- DrawDemo
- Add Class...
- App
- ActionButton
- ButtonHolder
- CenterOfMassMenuItem
- ClearGridMenuButton
- DeleteButton
- DrawingPanel
- GridMenuButton
- HighlightedMenuButton
- LoadMenuButton
- MainPanel
- Top of file
- Member Variables
- Add Method...
- MainPanel
- createMenu1
- createMenu2
- createMenu3
- createMenu4
- createMenuBar
- createPasswordHandler
- createPropertyButtons
- createRandomShapeBut
- createShapeButton
- generateStatisticsMessa
- getColorPanel
- getDP
- getShapeIntoPanel
- handleBorderOfPanel
- handleColorChange
- handleStatisticsGUI
- loadFromFile
- logMenuTip
- makeShapeButtonPanel
- paintComponent
- saveMenuLog
- sort
- update
- updateLogPanel
- writeToFile
- MainMenuMenuButton
- OrganizationButton
- PasswordCommandListen
- PasswordConfirmButton
- PasswordFrame
- PasswordHelpButton
- QuitButton
- ...

```

DrawDemo+DrawingFunc1+
private void generateOrderedShapes() {
    boolean updateShapes = false;

    for(Shape shape : getInteractiveShapes()){
        if(this.getWidth() != _lastWidth) {
            updateShapes = true;
            break;
        }

        if(!_prevShapeLocations.keySet().contains(
            shape)){
            updateShapes = true;
            break;
        }
        else {
            ShapeInfo info = _prevShapeLocations.
                get(shape);
            if(shape.getWidth() != info.width ||
                shape.getHeight() != info.height){
                updateShapes = true;
                break;
            }
        }
    }

    if(updateShapes)
        fillOrderedShapes();

    layoutOrderedRows();
}
    
```

```

DrawDemo+DrawingPanel+
private void fillOrderedShapes() {
    _orderedShapes = new Vector<Shape>();
    for(Shape shape : getInteractiveShapes())
        _orderedShapes.add(shape);

    sortOrderedShapes();
}
    
```

```

DrawDemo+DrawingPanel+
private void storeShapeLocations() {
    _prevShapeLocations = new HashMap<Shape,
        ShapeInfo>();
    for(Shape shape : getInteractiveShapes())
        _prevShapeLocations.put(shape, new
            ShapeInfo(shape.getWidth(), shape.getHeight(),
                shape.getWidth(), shape.getHeight()));
}
    
```

```

DrawDemo
DrawingPanel
_orderedShapes private Vector<Shape> _orderedShapes = new Vector<Shape>();
_fillOrderedShapes _orderedShapes = new Vector<Shape>();
_orderedShapes.add(shape);
_sortOrderedShapes let numShapes = _orderedShapes.size();
_orderedShapes.removeAll(Collections.singleton(Shape));
_orderedShapes.add(shape);
_findShortest if(_orderedShapes.elementAt(i).getWidth() < minWidth){
    shortest = _orderedShapes.elementAt(i);
}

@DrawDemo+DrawingPanel+
private void computeOrderedShapeFlow() {
    _orderedShapesLayout = new Vector<Vector<Shape>>();
    ...
    _orderedShapesLayout.add(new Vector<Shape>());
    for(Shape shape : _orderedShapes) {
        double shapeWidth = shape.getWidth();
        if(currWidth + shapeWidth + _horizontalPadding < _lastWidth){
            ...
        }
    }
}
    
```

```

DrawDemo+DrawingPanel+
private void layoutOrderedRows() {
    this.setActiveShape(null);
    _orderedShapesLayout();

    double currY = _verticalPadding;
    double currX = _horizontalPadding;
    double maxRow = 0;

    for(Vector<Shape> row :
        _orderedShapesLayout) {
        for(Shape shape : row) {
            shape.setLocation(currX, currY);
            currX += shape.getWidth() +
                _horizontalPadding;
            if(shape.getHeight() > maxRow)
                maxRow = shape.getHeight();
        }
        currX = _horizontalPadding;
        currY += maxRow;
    }
}
    
```

```

DrawDemo
DrawingFunc1
orderedShapesLayout private Vector<Vector<Shape>> _orderedShapesLayout

@DrawDemo+DrawingFunc1+
private void computeOrderedShapesLayout() {
    _orderedShapesLayout = new Vector<Vector<Shape>>();
    _lastWidth = this.getWidth();
    if(!_paddingQueried)
        ...
    double currWidth = _horizontalPadding;
    _orderedShapesLayout.add(new Vector<Shape>());
    for(Shape shape : _orderedShapes) {
        double shapeWidth = shape.getWidth();
        if(currWidth + shapeWidth + _horizontalPadding < _lastWidth){
            _orderedShapesLayout.lastElement().add(shape);
            currWidth += shapeWidth + _horizontalPadding;
        }
    }
}
    
```

**Things to Remember**

- OrderedShapesLayout is what you want, not orderedShapes
- Be sure to call layoutOrderedRows first

DrawDemo

- Add Class...
- App
- ActionButton
- ButtonHolder
- CanvasOfMainMenuItems
- ClearGridMenuButton
- DeleteButton
- DrawingPanel
- GridMenuButton
- HighlightsMenuButton
- LoadMenuButton
- MainPanel

Top of file

- Member Variables
- Add Method...
- MainPanel
- createMenu1
- createMenu2
- createMenu3
- createMenu4
- createMenuJar
- createPasswordHandler
- createPropertyButtons
- createRandomShapeBut
- createShapeButton
- generateStatisticsMenu
- getColorPanel
- getDP
- getShapeIntoPanel
- handleBorderOfPanel
- handleColorChange
- handleStatisticsGUI
- loadFromFile
- logMenuCip
- makeShapeButtonPanel
- paintComponent
- resetMenuLog
- sort
- update
- updateLogPanel
- writeToFile
- MakeHistogramMenuButton
- OrganizationButton
- PasswordCommandListener
- PasswordConfirmButton
- PasswordFrame
- PasswordInputDialog
- QuitButton

```

DrawDemo+ShapeButton+
public void handleClick() {
    _pf.update(this);
    DrawingFunc1 df = _pf.getDF();
    Shape newShape = df.createShapeFromText(
        _text);
    newShape.setLocation(df.getRandom(), df.
        getRandom());
    df.add(newShape);
    df.makeShapeInteractive(newShape);
    _pf.setActiveShape(newShape);
}
    
```

```

DrawDemo+DrawingPanel+
public Shape createShapeFromText(String text) {
    Shape s = null;

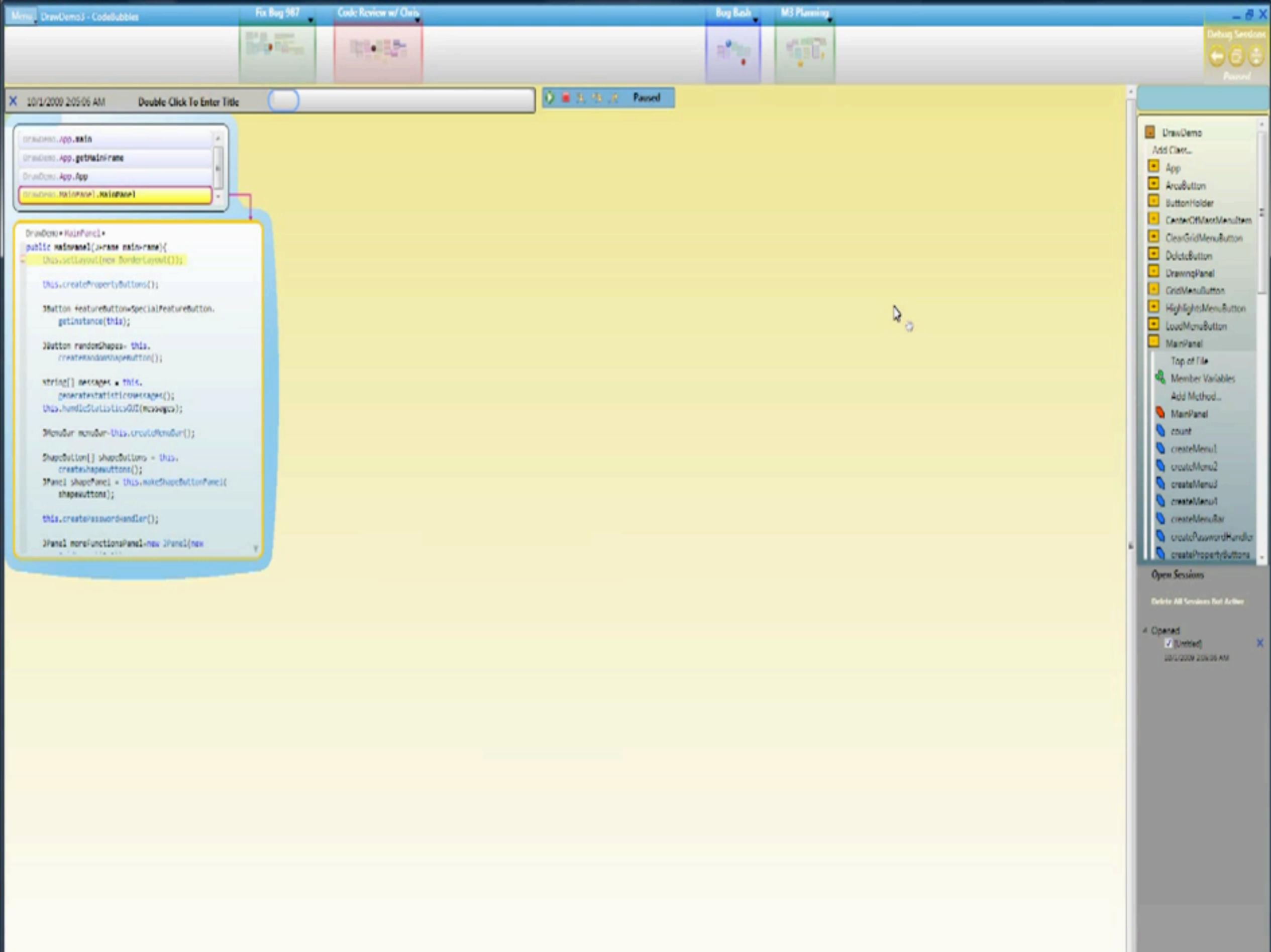
    if (this.isSpecifyDimensions()) {
        this.initialiseDimensions();
        this.setSpecifyDimensions();
    }
}
    
```

```

DrawDemo+DrawingPanel+
public void setSpecifyDimensions() {
    if (_currentButtonShape != null) {
        _currentButtonShape.setSpecify(
            _specificWidth, _specificHeight);
        updateColorKey();
    }
}
    
```

```

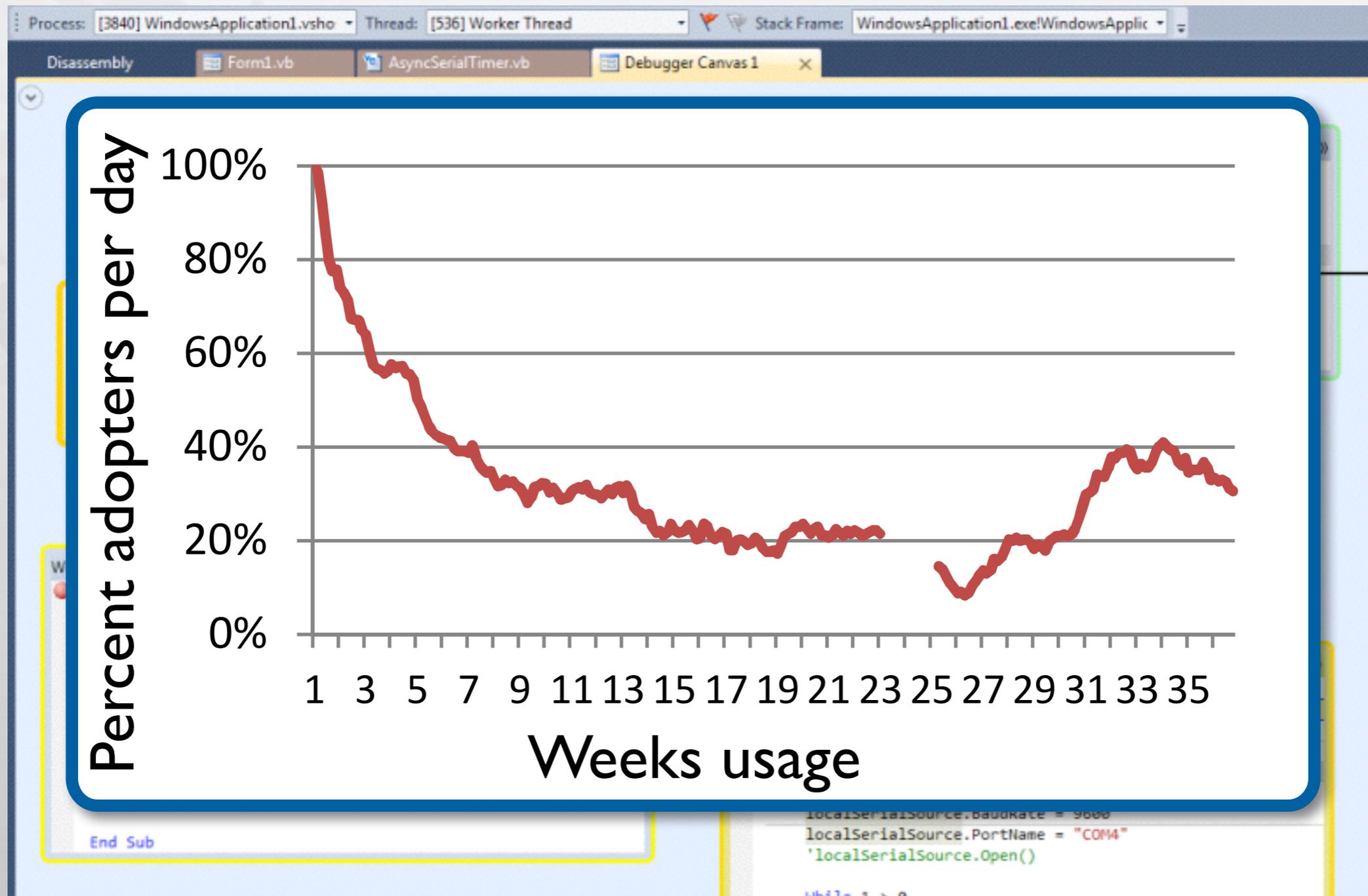
DrawDemo+DrawingPanel+
public void updateDisplay() {
    for (int i = 0; i < _items.length; i++)
        updateItem(i);
}
    
```



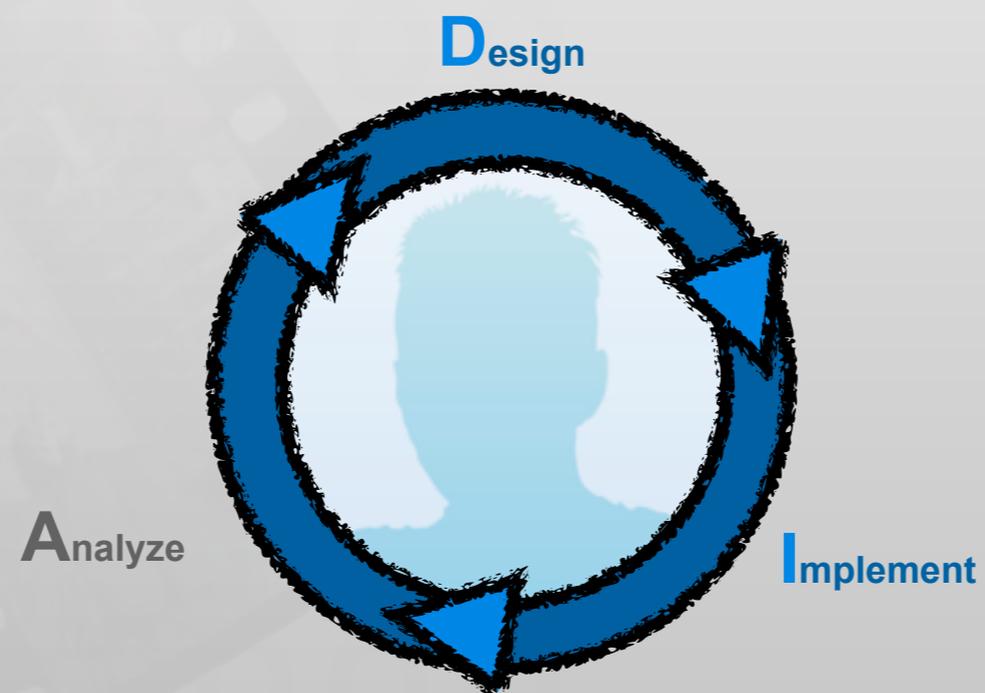
[Bragdon2010, Code bubbles: a working set-based interface for code understanding and maintenance]

# Canvas Interfaces in the Wild

[DeLine2012, Debugger Canvas: Industrial experience with the code bubbles paradigm]



# Utilizing the Call Graph



# Reachability Questions

[LaToza2010, Developers ask reachability questions]

Where is method <i>m</i> generating an error?	Find output of error text downstream from <i>m</i>
What resources are being acquired to cause this deadlock?	Find calls to methods acquiring the resources
What is this test doing which is different from what my app is doing?	Find differences between two call stacks
Is there another reason why this status is non-zero?	Find upstream statements that change the status



```

- (void)movePapers:(NSArray *)paperInfos forField:(NSString *)field fromDocument:
(BibDocument *)doc options:(NSInteger)mask{
    NSFileManager *fm = [NSFileManager defaultManager];
    NSInteger numberOfPapers = [paperInfos count];
    BibItem *pub = nil;
    BDSKLinkedFile *file = nil;
    NSString *oldPath = nil;
    NSString *newPath = nil;
    NSMutableArray *files = [NSMutableArray arrayWithCapacity:numberOfPapers];
    NSMutableArray *pubs = [NSMutableArray arrayWithCapacity:5];
    NSMutableDictionary *infoDict = [NSMutableDictionary];
    NSError *error = nil;

    BOOL initial = (mask & BSKFileOptionMask);
    BOOL force = (mask & BSKFileOptionForce);
    BOOL check = (initial == NO) && (mask & BSKCheckCompleteAutoFile);

    if (numberOfPapers == 0)
        return;

    if (initial && [field isEqualToString:BDSKLocalFileString] == NO)
        [NSException raise:BDSKUnimplementedException format:@"%s is only implemented
local files for initial moves.",NSStringFromSelector(selector)];

    if (numberOfPapers > 1) {
        [self window];
        [progressIndicator setMaxValue:numberOfPapers];
        [progressIndicator setDoubleValue:0.0];
        [[self window] orderFront:nil];
    }

    for (id paperInfo in paperInfos) {
        file = [paperInfo valueForKey:BDSKFileFileKey];
        pub = [paperInfo valueForKey:BDSKFilePublicationKey];
        oldPath = [[file URL] path];
        if (initial) // action: array of BDSKLinkedFiles
            newPath = [fm URLForLinkedFile:file];
        else // an error message from one of info dictionaries
            newPath = [fm URLForLinkedFile:file];

        if (numberOfPapers > 1) {
            [progressIndicator setDoubleValue:1.0];
            [progressIndicator setMaxValue:numberOfPapers];
        }

        if ([NSString isEqualToString:oldPath] || [NSString isEmptyString:newPath] ||
[oldPath isEqualToString:newPath]) {
            [pub removeFileToBeFiled:file];
            continue;
        }

        info = [NSMutableDictionary dictionaryWithCapacity:6];
        [info setValue:file forKey:BDSKFileFileKey];
        [info setValue:oldPath forKey:BDSKFileOldPathKey];
        [info setValue:pub forKey:BDSKFilePublicationKey];
        error = nil;

        if (check && NO == [pub canSetURLForLinkedFile:file]) {
            [info setValue:NSString(@"Incomplete information to generate file
name.",@"") forKey:BDSKFileStatusKey];
            [info setValue:[NSNumber numberWithInt:BDSKIncompleteFieldsErrorMask]
forKey:BDSKFileFlagKey];
            [info setValue:NSString(@"Move anyway.",@"") forKey:BDSKFileFixKey];
            [info setValue:newPath forKey:BDSKFileNewPathKey];
        }
    }
}

```

```

- (NSURL *)suggestedURLForLinkedFile:(BDSKLinkedFile *)file
{
    NSString *papersFolderPath = [BDSKFormatParser folderPathForFilingPapersFromDocumentAtP
    NSString *relativeFile = [BDSKFormatParser parseFormatForLinkedFile:file ofItem:self];
    if ([[NSUserDefaults standardUserDefaults] objectForKey:BDSKLocalFileFormat])
        relativeFile = [relativeFile lowercaseString];
    return [NSURL URLWithString:[papersFolderPath stringByAppendingPathComponent:relativeFile]];
}

+ (NSString *)parseFormatForLinkedFile:(BDSKLinkedFile *)file ofItem:(id <BDSKParseableItem
{
    NSString *localFileFormat = [[NSUserDefaults standardUserDefaults] objectForKey:BDSKLoc
    [self parseFormat:localFileFormat forLinkedFile:file ofItem:pub];
}

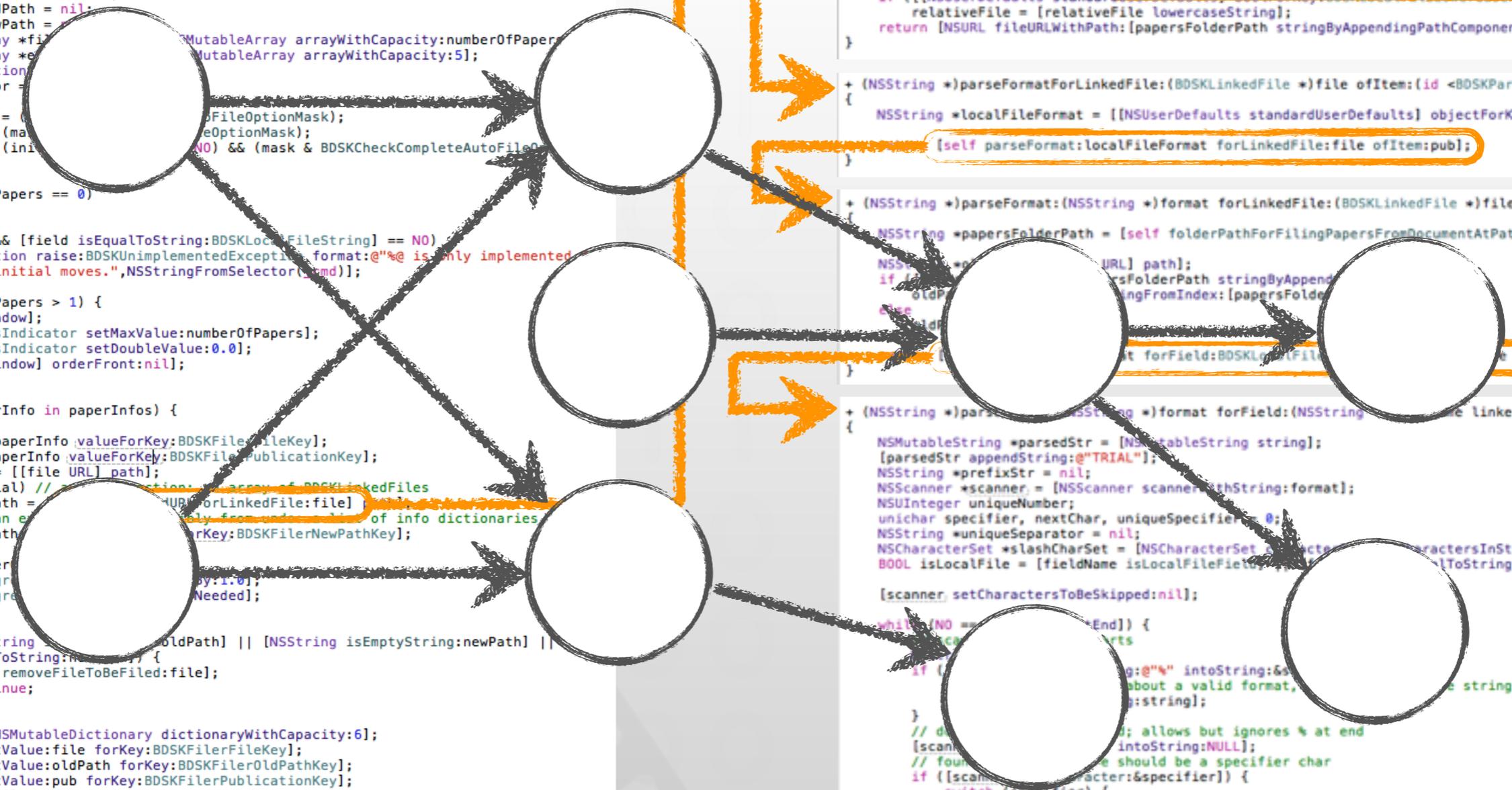
+ (NSString *)parseFormat:(NSString *)format forLinkedFile:(BDSKLinkedFile *)file ofItem:(id
{
    NSString *papersFolderPath = [self folderPathForFilingPapersFromDocumentAtPath:[pub o
    NSString *relativeFile = [BDSKFormatParser parseFormatForLinkedFile:file ofItem:self];
    if ([papersFolderPath stringByAppendingPathComponent:relativeFile] != nil)
        relativeFile = [relativeFile lowercaseString];
    else
        relativeFile = [relativeFile lowercaseString];
    return [NSURL URLWithString:[papersFolderPath stringByAppendingPathComponent:relativeFile]];
}

+ (NSString *)parseFormat:(NSString *)format forField:(NSString *)field linkedFile:(BDSK
{
    NSMutableString *parsedStr = [NSMutableString string];
    [parsedStr appendString:@"TRIAL"];
    NSString *prefixStr = nil;
    NSScanner *scanner = [NSScanner scannerWithString:format];
    NSInteger uniqueNumber;
    unichar specifier, nextChar, uniqueSpecifier = 0;
    NSString *uniqueSeparator = nil;
    NSScanner *slashCharSet = [NSScanner scannerWithCharacters:@"/*"];
    BOOL isLocalFile = [field isLocalFile];

    [scanner setCharactersToBeSkipped:nil];

    while (NO == [scanner isAtEnd]) {
        if ([scanner scanString:@"%" intoString:&specifier]) {
            if ([specifier isEqualToString:@"a"]) {
                // about a valid format,
                // the string
            }
            // do not allow % at end; allows but ignores % at end
            [scanner scanString:@"%" intoString:NULL];
            // found % should be a specifier char
            if ([specifier isEqualToString:@"a"]) {
                switch (specifier) {
                    case 'a':
                    case 'p':
                    {
                        // author names, optional [separator], [etal], #names and #chars
                        NSInteger numChars = 0;
                        NSInteger i, numAuth = 0;
                        NSString *authSep = @" ";
                        NSString *etal = @" ";
                        BOOL isLast = NO;
                        if (NO == [scanner isAtEnd]) {

```



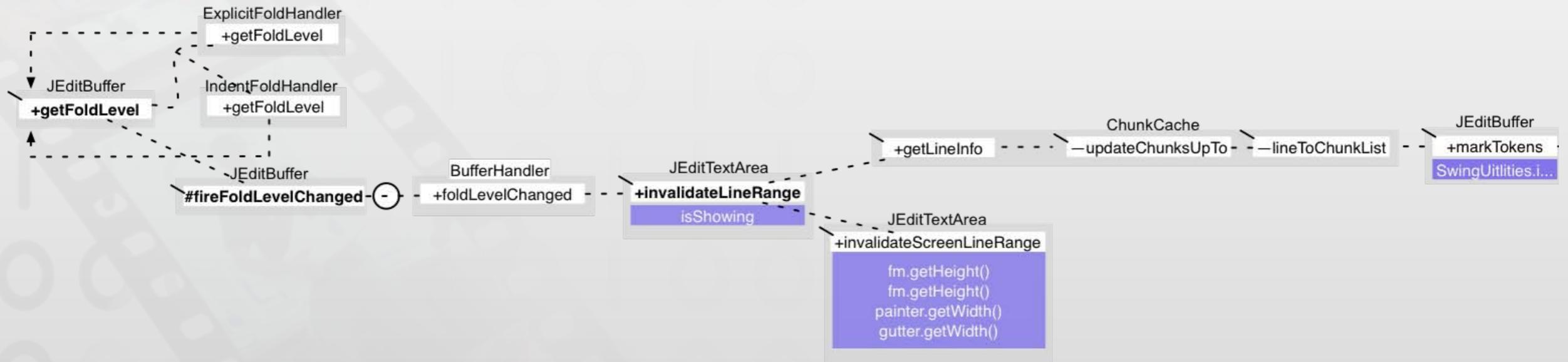
# In practice: Feasible paths most interesting

[LaToza2010, Developers ask reachability questions]



# Utilizing Call Graph Information

[LaToza2010, Searching Across Paths]



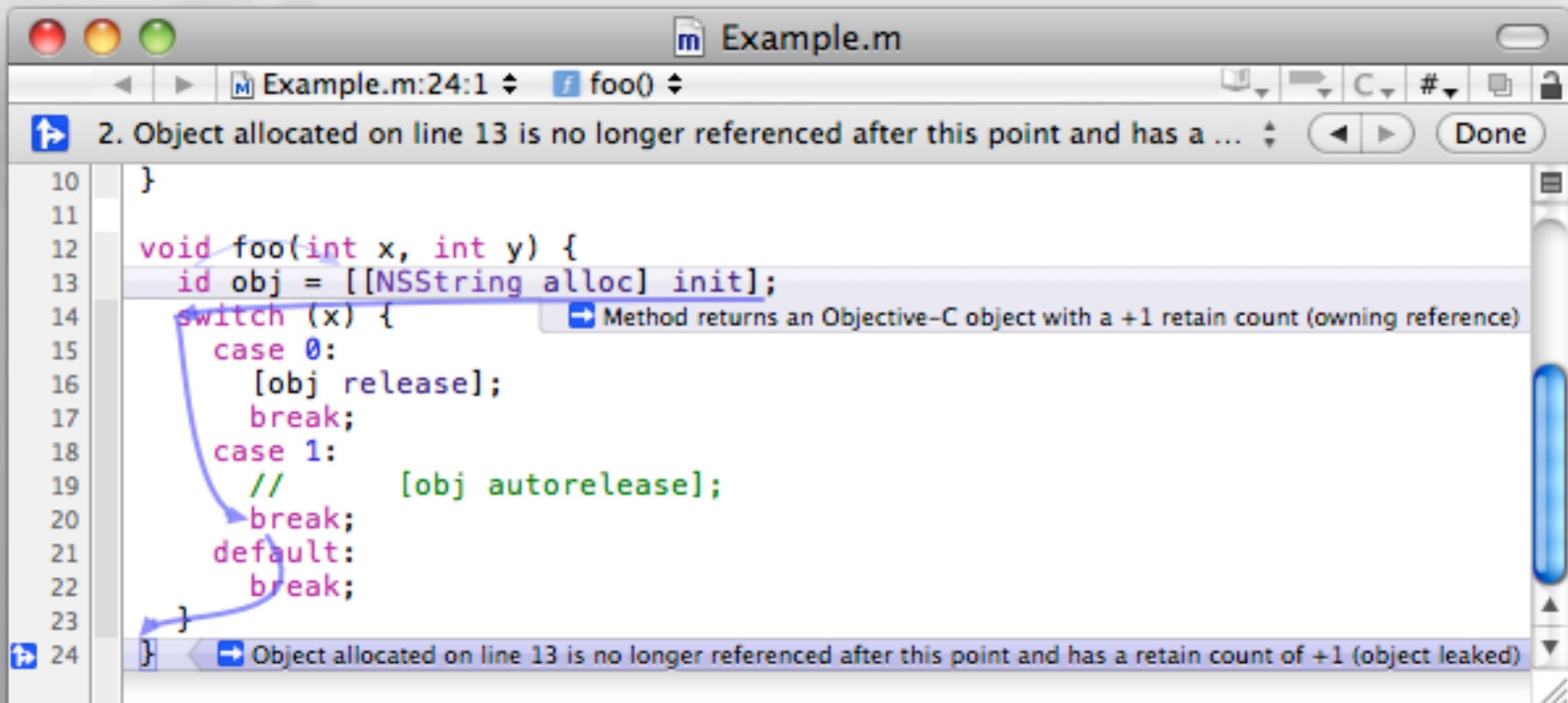
## Legend

<code>+methodName</code>	public / protected / private method	————	method call that is always executed		recursive method call
<code>#methodName</code>	method visited by developer	- - - -	method call that might execute		paths of calls with hidden methods
<code>-methodName</code>	method with callers that are not shown		mutually exclusive method calls		data flow
<code>+methodName</code>	method with callers that are not shown		method call in a loop		expression that matches search
TypeName	type with type name				



# Static Analysis in the Wild

[Clang Static Analyzer, <http://clang-analyzer.lvm.org/>]



The screenshot shows a code editor window titled "Example.m" with a Clang Static Analyzer warning. The warning message is: "2. Object allocated on line 13 is no longer referenced after this point and has a ...". The code snippet is as follows:

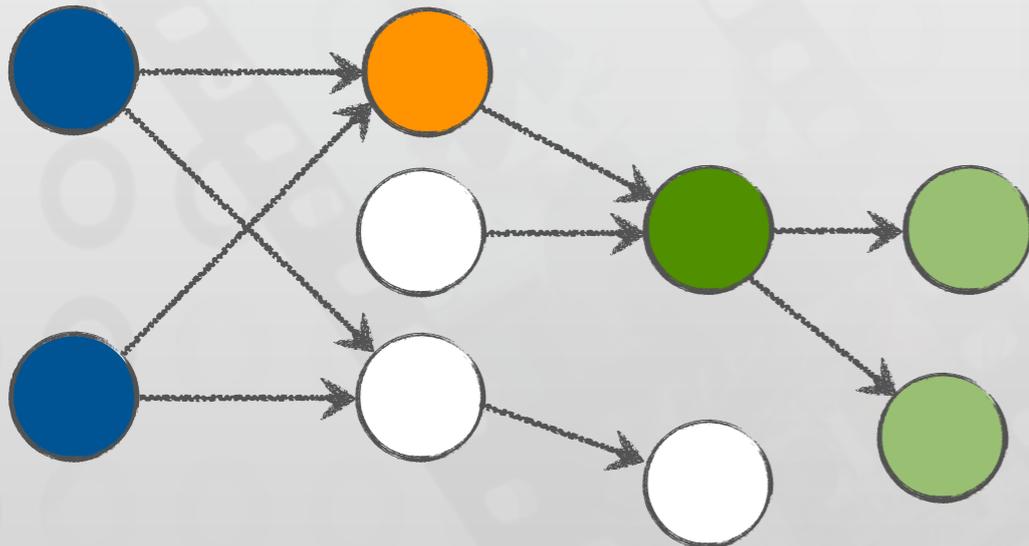
```
10 }
11
12 void foo(int x, int y) {
13     id obj = [[NSString alloc] init];
14     switch (x) {
15         case 0:
16             [obj release];
17             break;
18         case 1:
19             // [obj autorelease];
20             break;
21         default:
22             break;
23     }
24 }
```

Annotations in the image include:

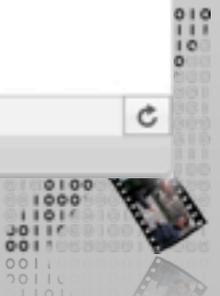
- A blue arrow pointing from the warning message to line 13.
- A blue arrow pointing from the warning message to line 24.
- A blue arrow pointing from the warning message to the closing brace of the switch statement on line 23.
- A tooltip on line 14: "Method returns an Objective-C object with a +1 retain count (owning reference)".
- A tooltip on line 24: "Object allocated on line 13 is no longer referenced after this point and has a retain count of +1 (object leaked)".



# Call Hierarchy

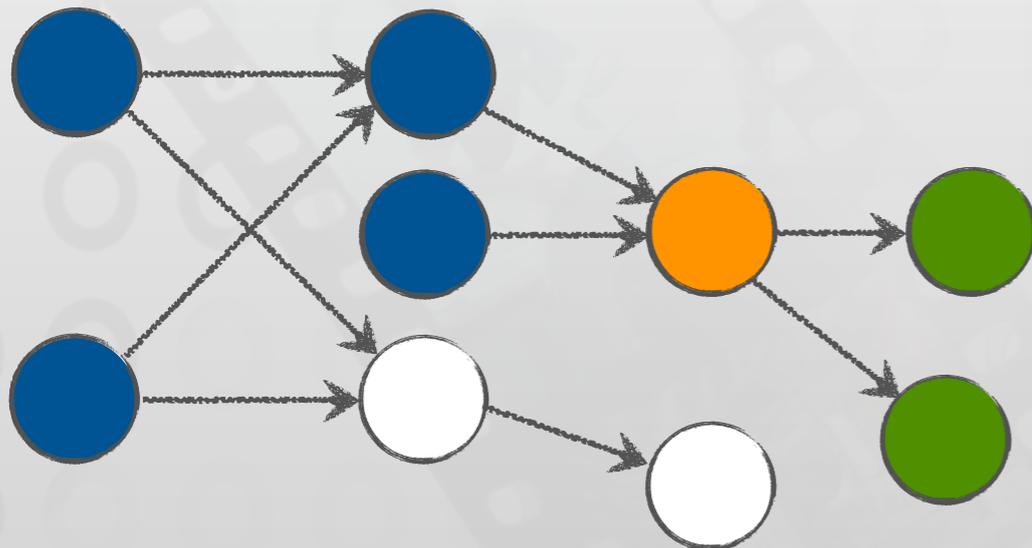


```
Converter.xcodeproj
MainController.m:39:1 -performC... C #
29 -(void)convertClicked:(id)sender;
30 {
31     //do something
32     if (self.theConverter == nil) {
33         self.theConverter = [[Converter alloc] init];
34     }
35
36     [self performConvert];
37 }
38
39 -(void)performConvert;
40 {
41     if ([[self.input floatValue] != 0] || [[self.input st
42     {
43         [self convert];
44     }
45 }
46
47 -(void)menuCallback:(id)sender;
48 {
49     [self convert];
50 }
51
52 //convert from Celsius
53 //to Fahrenheit
54 -(void)convert;
55 {
56     //get celsius value
57     float c = [self.input floatValue];
58
59     //convert to fahrenheit
60     float f = [self.theConverter c2f:c];
61
62     //update view
63     [self update:f];
64 }
65
66 -(void)update:(float)f;
67 {
68     //do something
69 }
70
71 @end
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
```



# Stacksplorer

[Karrer2011, Stacksplorer: Call Graph Navigation Helps Increasing Code Maintenance Efficiency]



```
class MainController {
  M init
  M convertClicked:
}

-(void)performConvert;
{
  if ([[self.input floatValue] != 0] || ([[self.input stringValue]
  {
    [self convert];
  }
}

-(void)menuCallback:(id)sender;
{
  [self convert];
}

//convert from Celsius
//to Fahrenheit
-(void)convert;
{
  //get celsius value
  float c = [self.input floatValue];

  //convert to fahrenheit
  float f = [self.theConverter c2f:c];

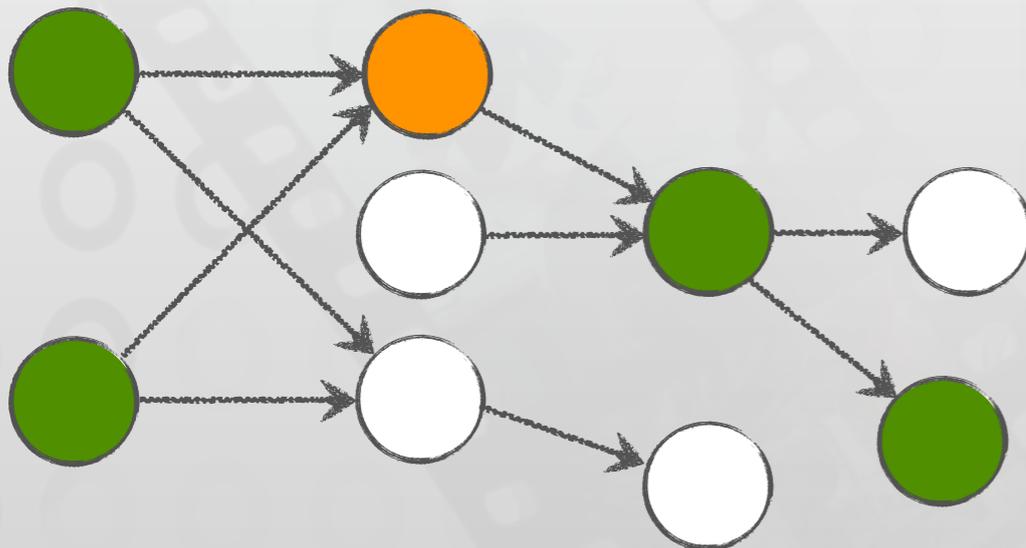
  //update view
  [self update:f];
}

-(void)update:(float)f;
{
  //do something
}
```



# Blaze

[Krämer2012, Blaze: Supporting Two-phased Call Graph Navigation in Source Code]

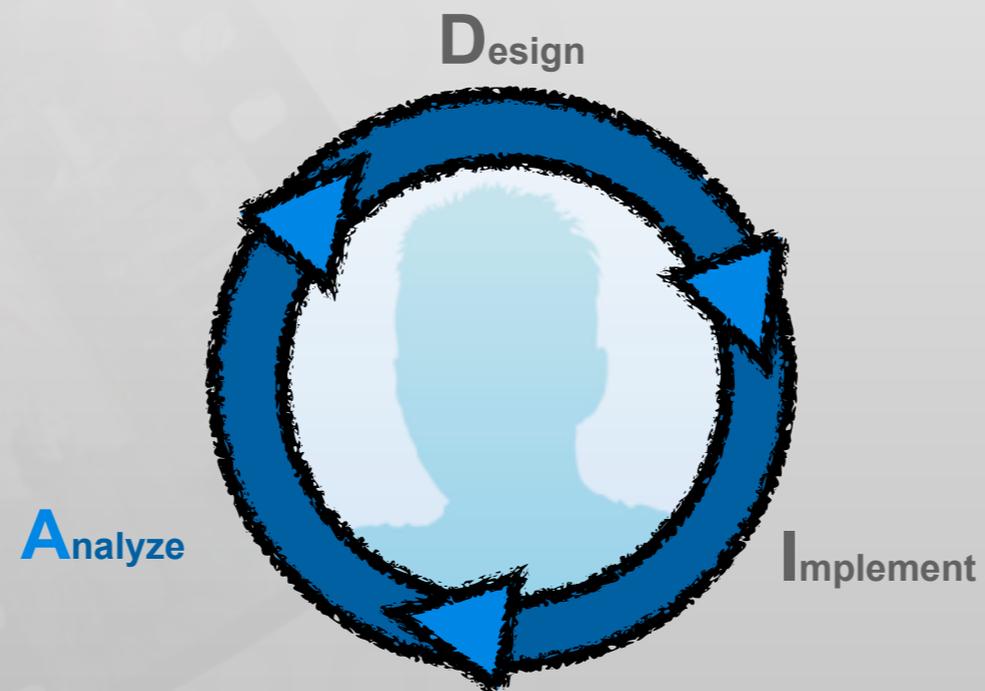


The screenshot shows the source code for `MainController.m` and `Converter.m`. The code is annotated with call graph navigation elements. The `convertClicked:` method in `MainController.m` is highlighted in orange, and the `convert:` method in `Converter.m` is highlighted in green. The call graph navigation elements are shown on the right side of the IDE, with green boxes highlighting the `convertClicked:` and `convert:` methods.

```
24 }
25 return self;
26 }
27
28
29 -(void)convertClicked:(id)sender;
30 {
31     //do something
32     if (self.theConverter == nil) {
33         self.theConverter = [[Converter alloc] init];
34     }
35
36     [self performConvert];
37 }
38
39 -(void)qw;
40 {
41     if ([[self.input floatValue] != 0] || ([[self.input stringValue] isEqual:
42         {
43             [self convert];
44         }
45     }
46
47 -(void)menuCallback:(id)sender;
48 {
49     [self convert];
50 }
51
52 //convert from Celsius
53 //to Fahrenheit
54 -(void)convert;
55 {
56     //get celsius value
57     float c = [self.input floatValue];
58
59     //convert to fahrenheit
60     float f = [self.theConverter c2f:c];
61
62     //update view
63     [self update:f];
64 }
65
66
```



# Analyzing Navigation Behavior



# Information Foraging Theory



Predator



Scent



Prey



# Information Foraging Theory

[Lawrance2010, Reactive information foraging for evolving goals]



Predator



Scent

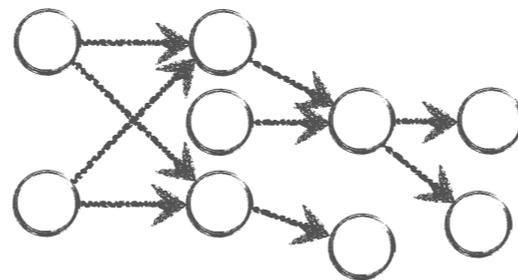


Prey

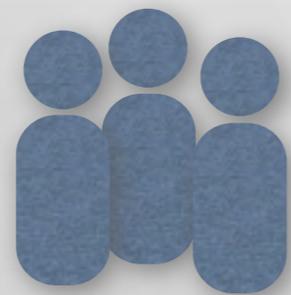


```
- (id)initWithBundle:(NSBundle *)plugin
{
    if (self = [super init]) {
        // reference to plugin's bundle, for resource access
        self.bundle = plugin;

        //register to notifications
        [[NSNotificationCenter defaultCenter] addObserver:self
        selector:@selector(onDidFinishSetup:) name:
        IDESourceCodeEditorDidFinishSetupNotification object:nil];
        [[NSNotificationCenter defaultCenter] addObserver:self
        selector:@selector(onTest:) name:
        DVTSourceExpressionSelectedExpressionDidChangeNotification
        object:nil];
    }
}
```



	Xcode	Call Hierarchy	Stacksplorer	Blaze
Find Change Location	Task Success Task Completion Time			
Side Effects of Change				



33 Developers



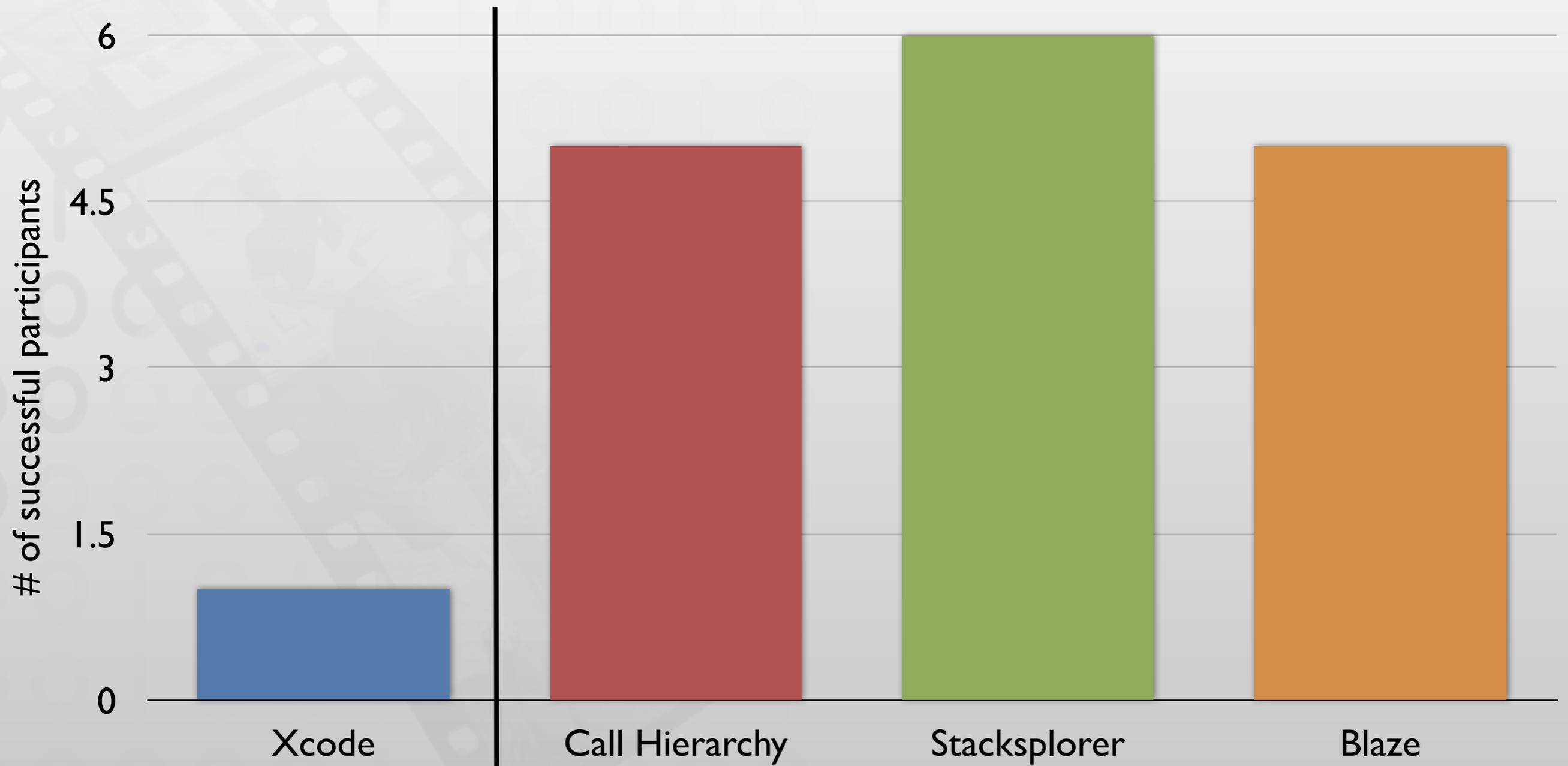
80.000 Lines of Code

[Krämer2013, How Tools in IDEs Shape Developers' Navigation Behavior]



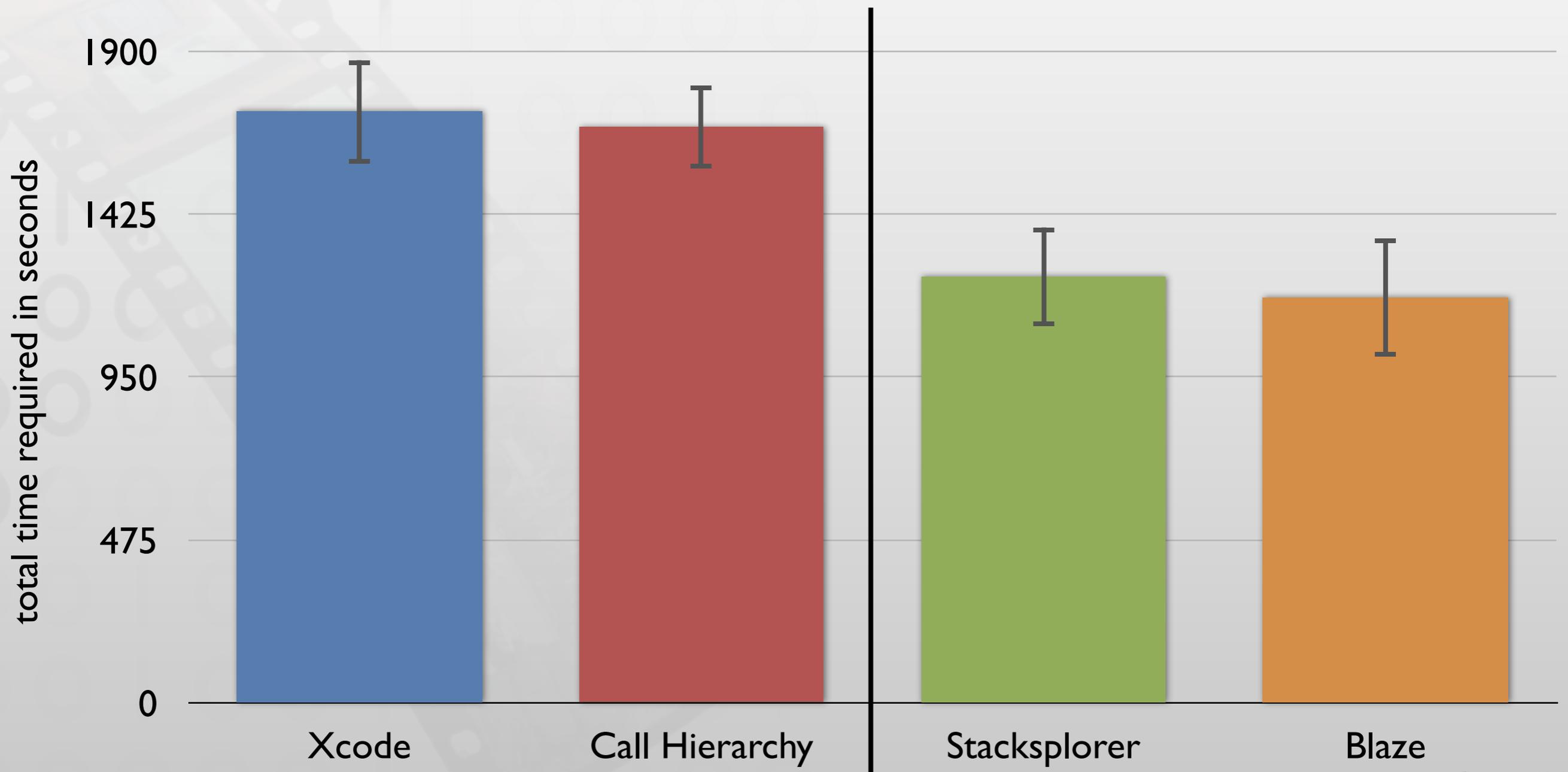
# Task Success

$p = 0.015$



# Task Completion Time

$p=0.022$



Effectiveness

Xcode

Call  
Hierarchy

Stacksplorer

Blaze

Efficiency

Xcode

Call  
Hierarchy

Stacksplorer

Blaze

Why?

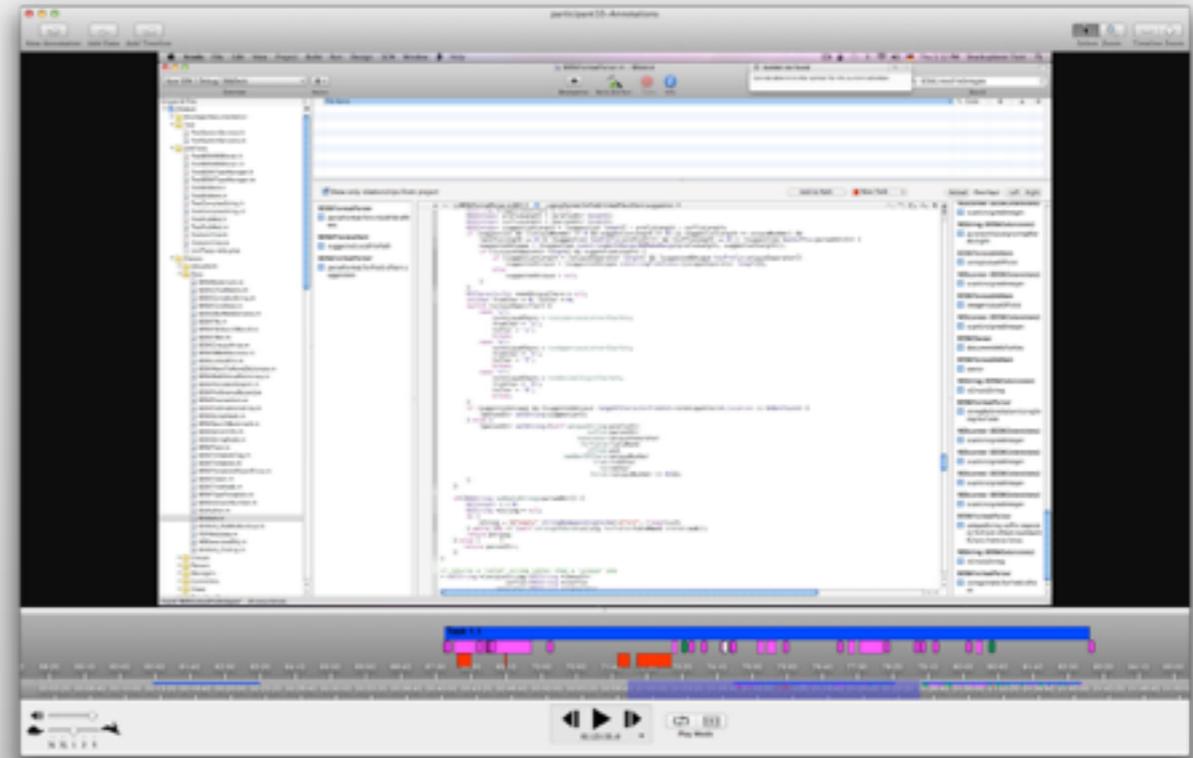
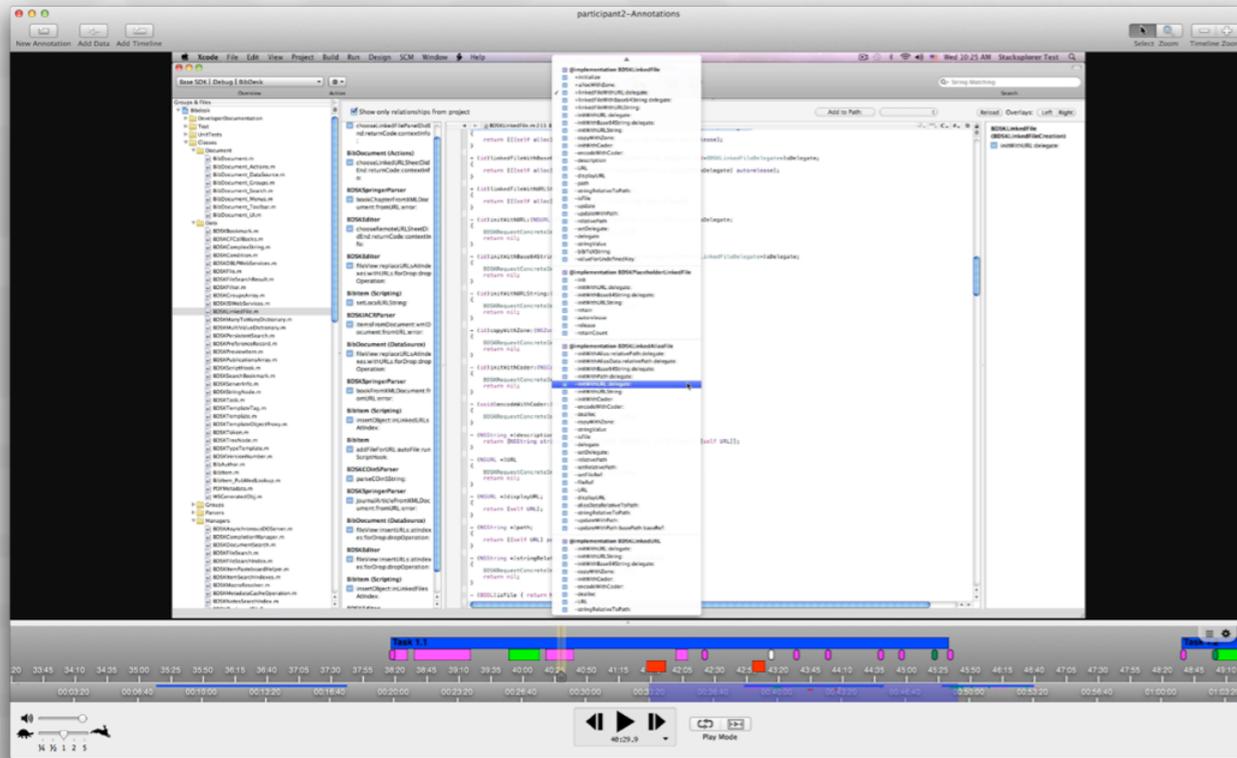
UI Differences

Navigation Behavior

The screenshot displays a video player interface. The main content area shows a software development environment (Xcode) with the title 'participant2--Annotations'. The Xcode interface includes a project navigator on the left, a central editor showing code for 'BDSKLinkedFile', and a right-hand pane with a search bar and overlays. The timeline at the bottom shows a duration of 01:03:20 with various colored segments and playback controls.

[Fouse2011, ChronoViz: A system for supporting navigation of time-coded data]

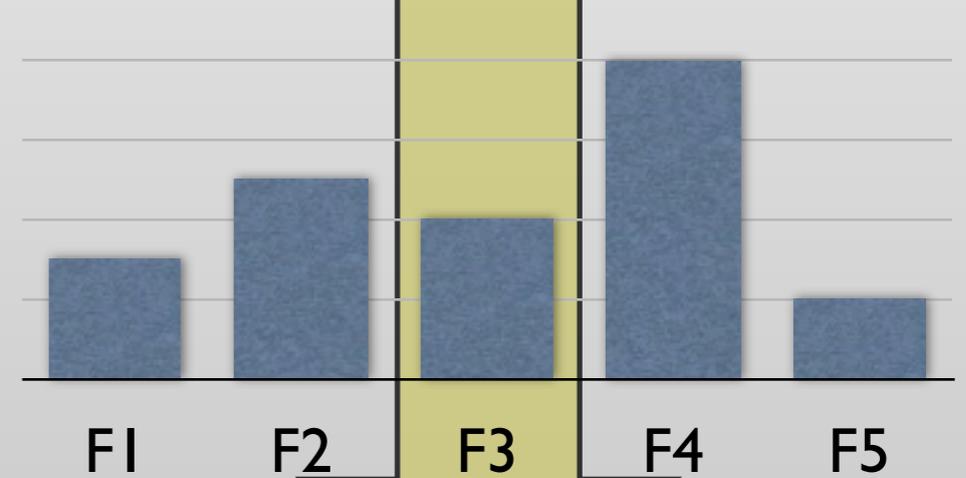
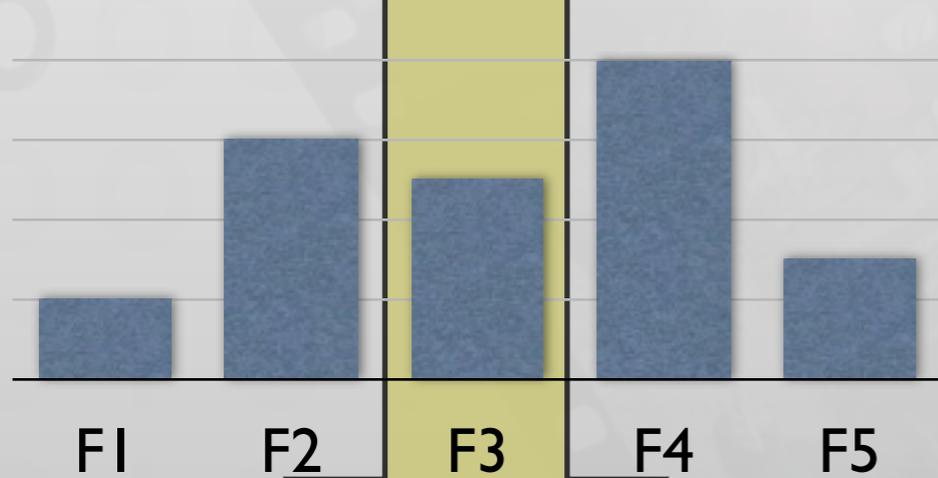
# Comparing Navigation Behavior





$$I_1 = (p_{1,1}, \dots, p_{640,480})$$

$$I_2 = (p_{1,1}, \dots, p_{1024,768})$$

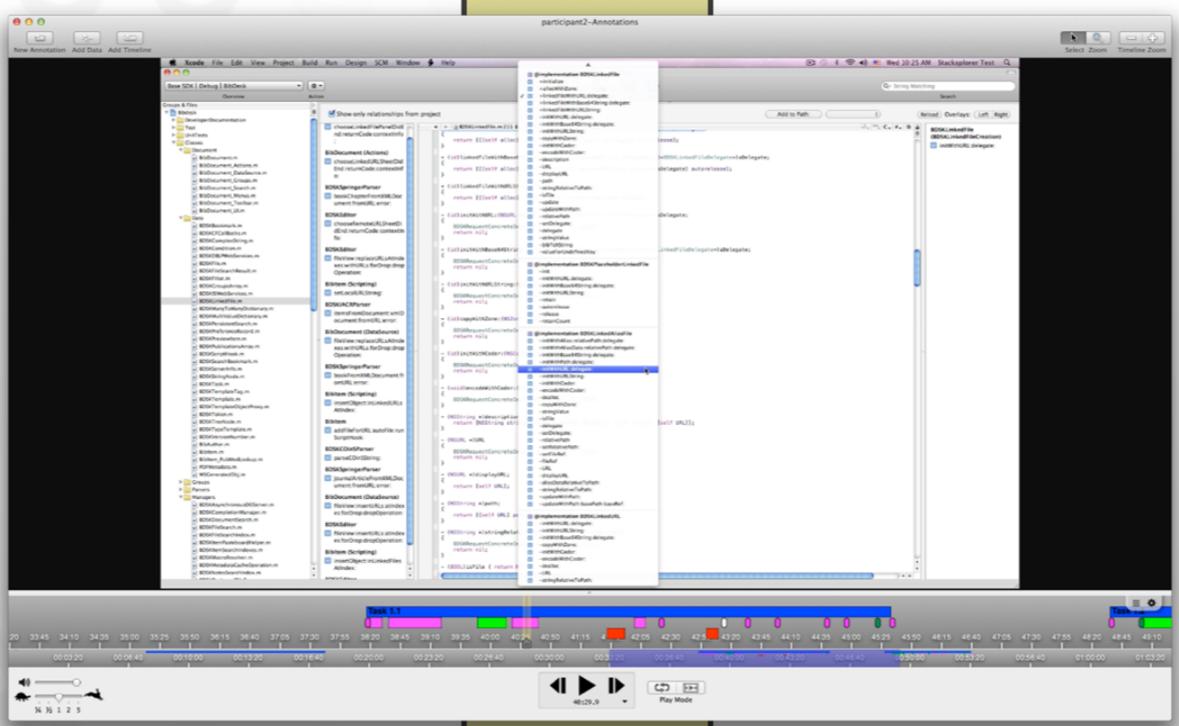


=

1. Features
2. Transformations

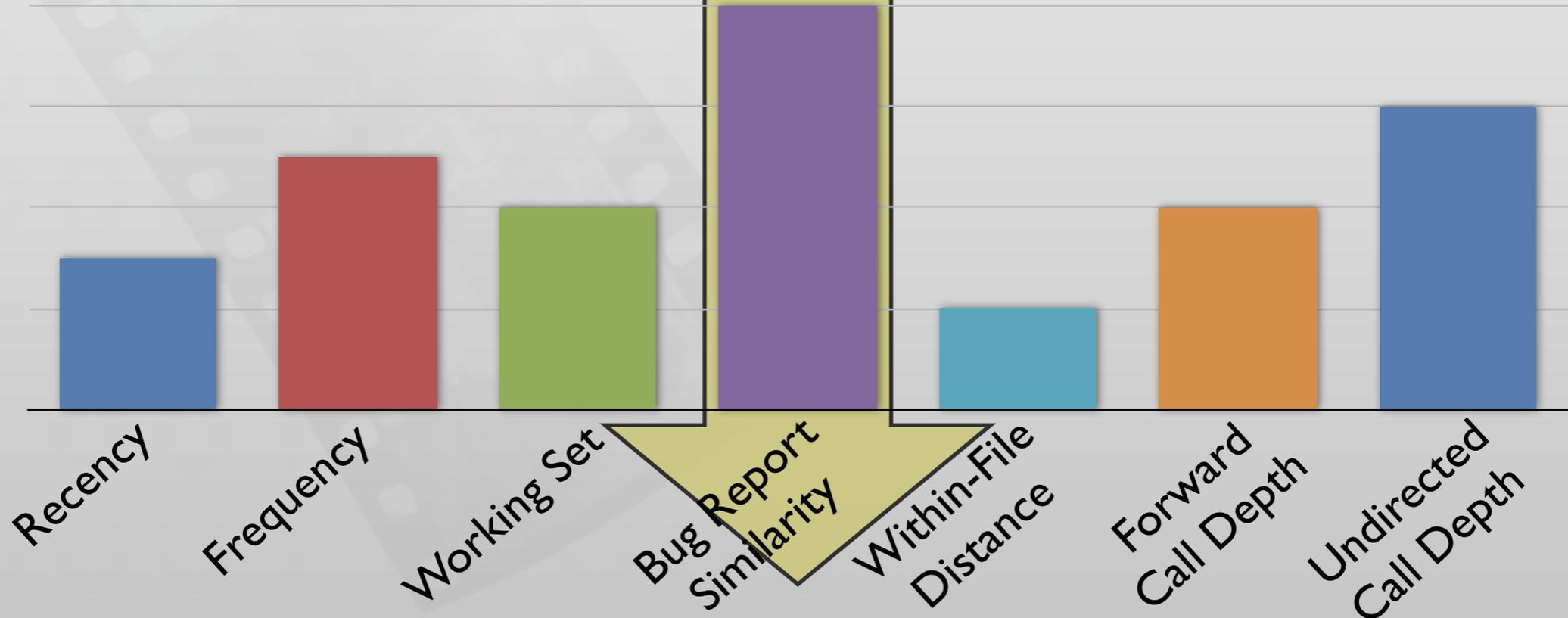


[Piorkowski2011,  
Modeling programmer  
navigation:A head-to-  
head empirical evaluation  
of predictive models]



$$H=(m_1, \dots, m_i)$$

Prediction Accuracy



# A Predictor

[Piorkowski2011, Modeling programmer navigation: A head-to-head empirical evaluation of predictive models]

$$H = (m_1, \dots, m_i)$$

Navigation History

$$H = (a, b, a, d)$$

$$M_i$$

All methods known  
to developer at time  
 $i$

$$M_4 = \{a, b, d\}$$

$$A_i: M_i - \{m_i\} \rightarrow \mathbb{R}$$

Activation value for  
each method in  $M_i$

$$A_4(a) = 3$$

$$A_4(b) = 2$$

$$R_i: M_i - \{m_i\} \rightarrow \mathbb{N}$$

Rank-transformed  
version of  $A_i$

$$R_4(a) = 1$$

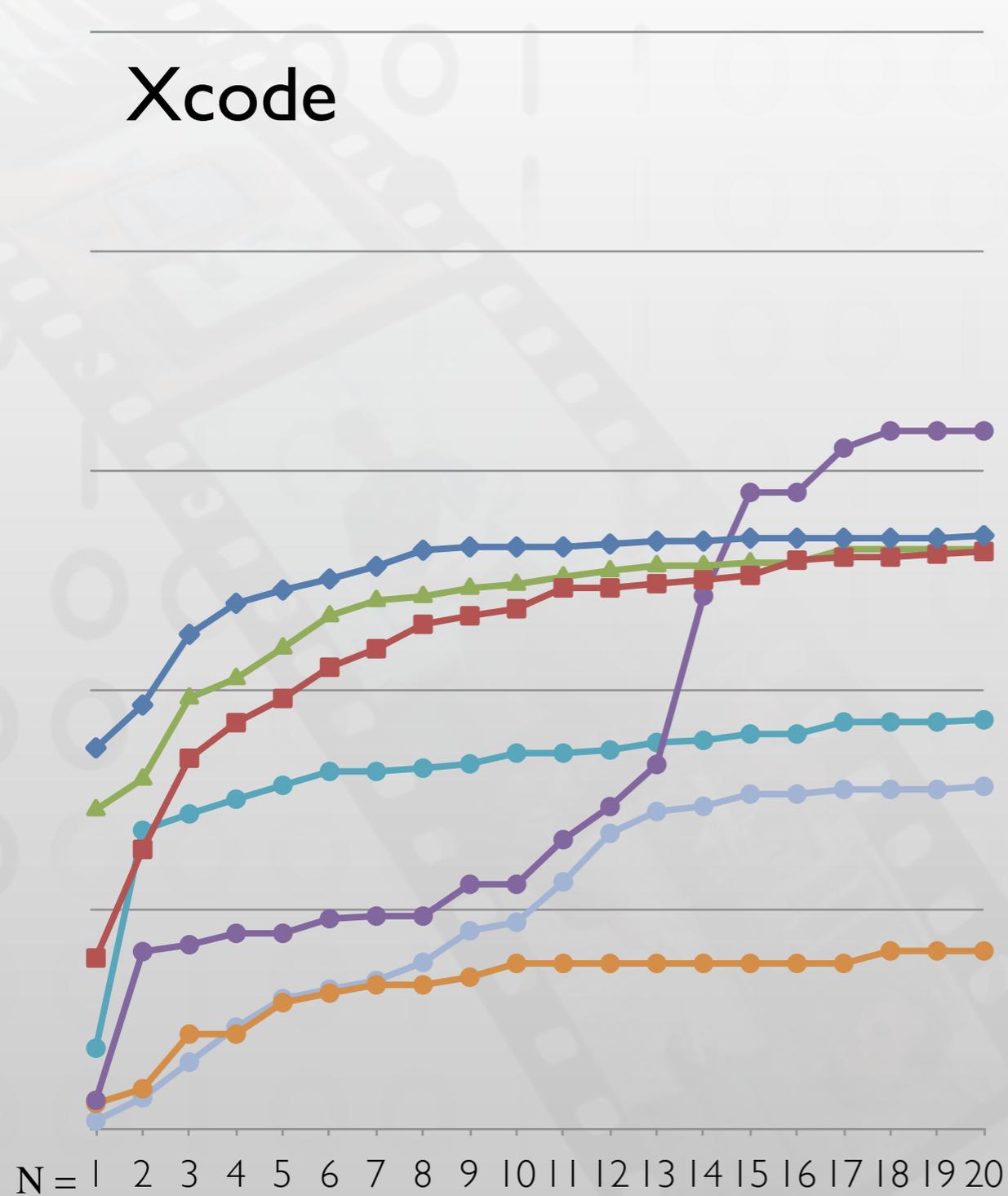
$$R_4(b) = 2$$

Result:  $N$  top-ranked methods

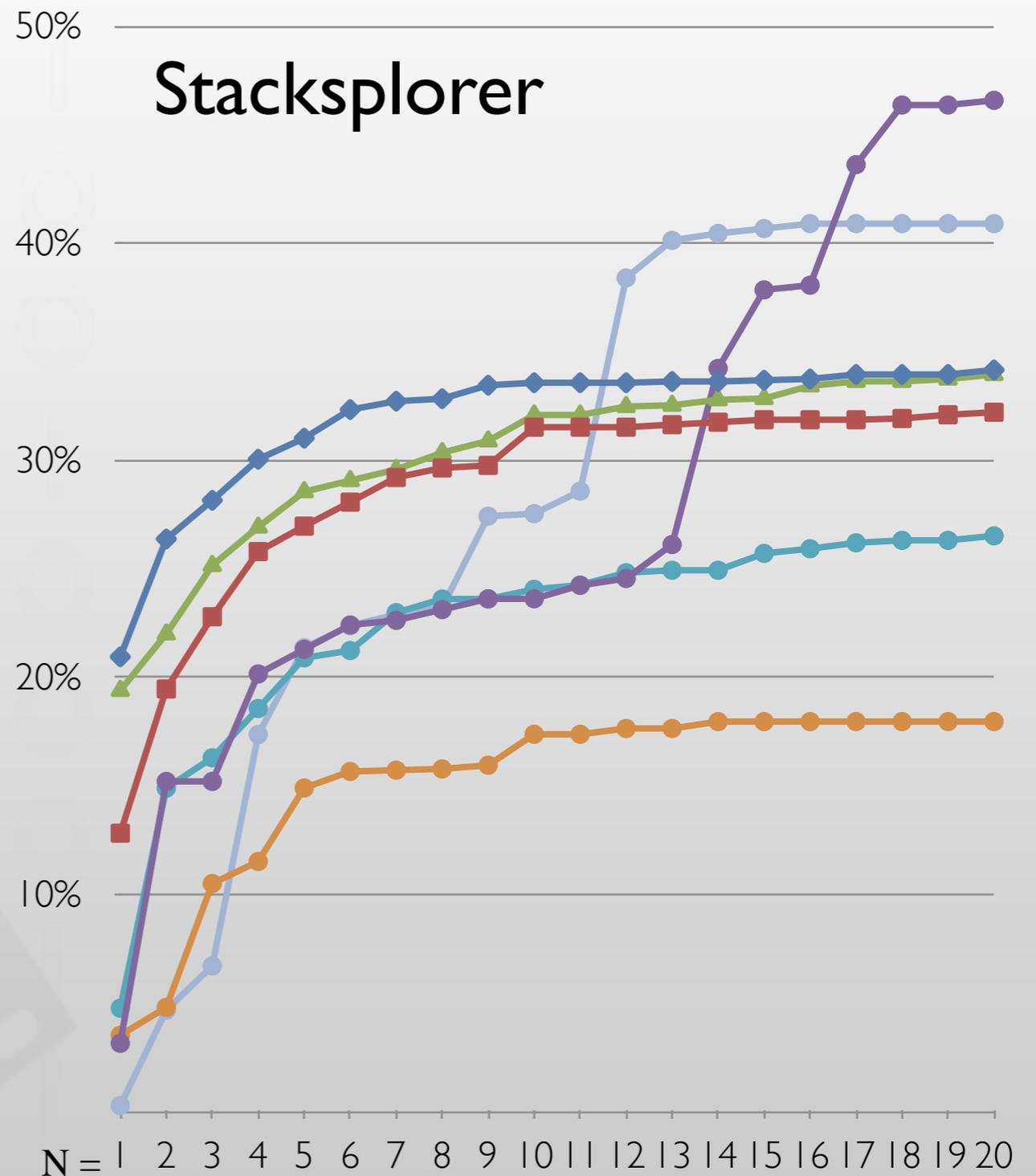


# Prediction Accuracy

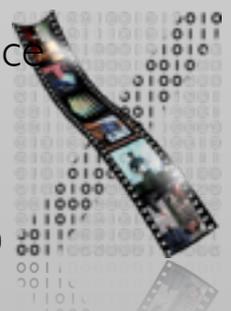
## Xcode



## Stacksplorer

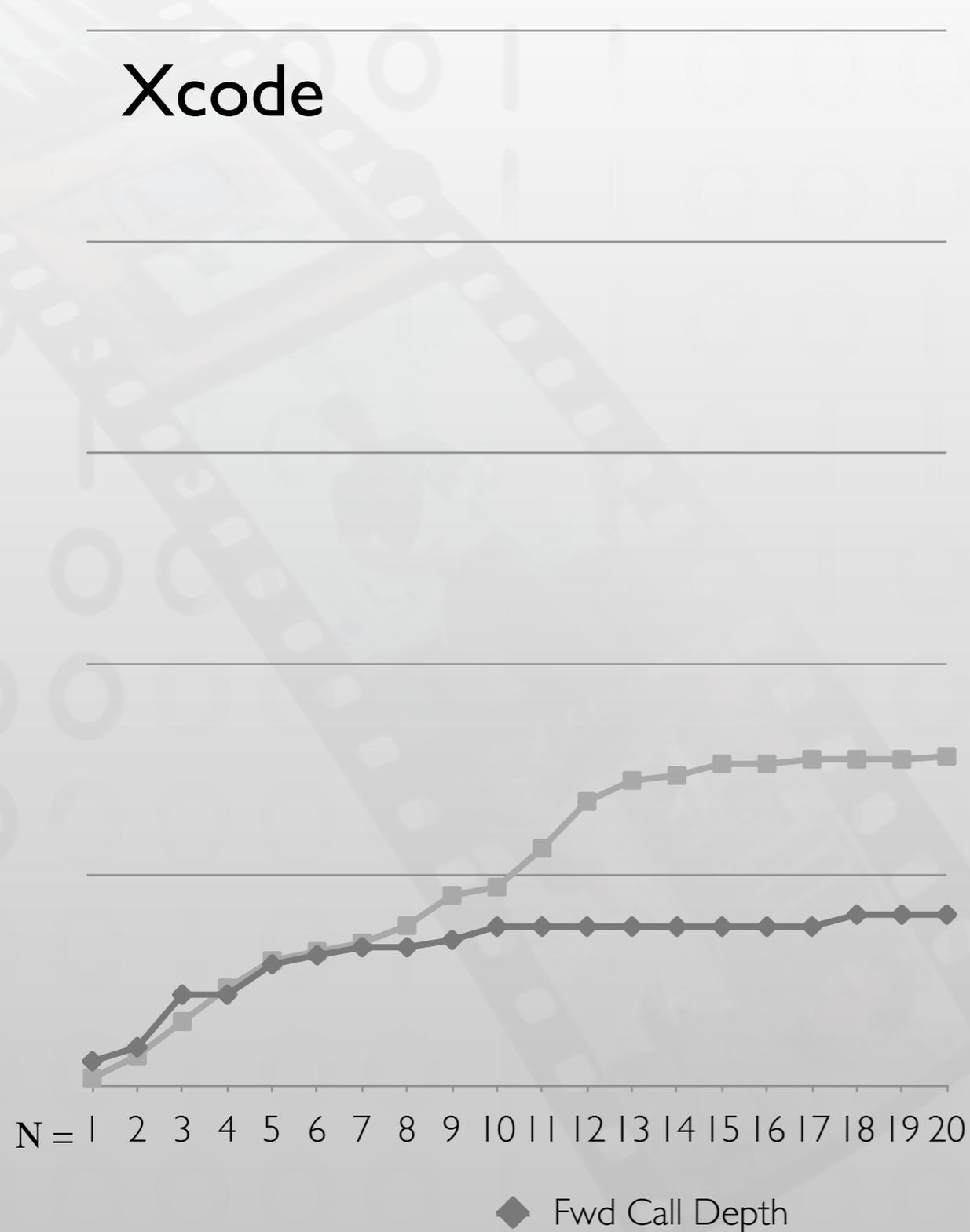


- ◆ Recency
- Frequency
- ▲ Working Set
- Bug Report Similarity
- Within File Distance
- Fwd Call Depth
- Undirected Call Depth

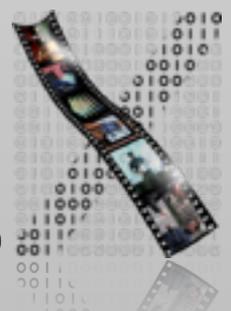
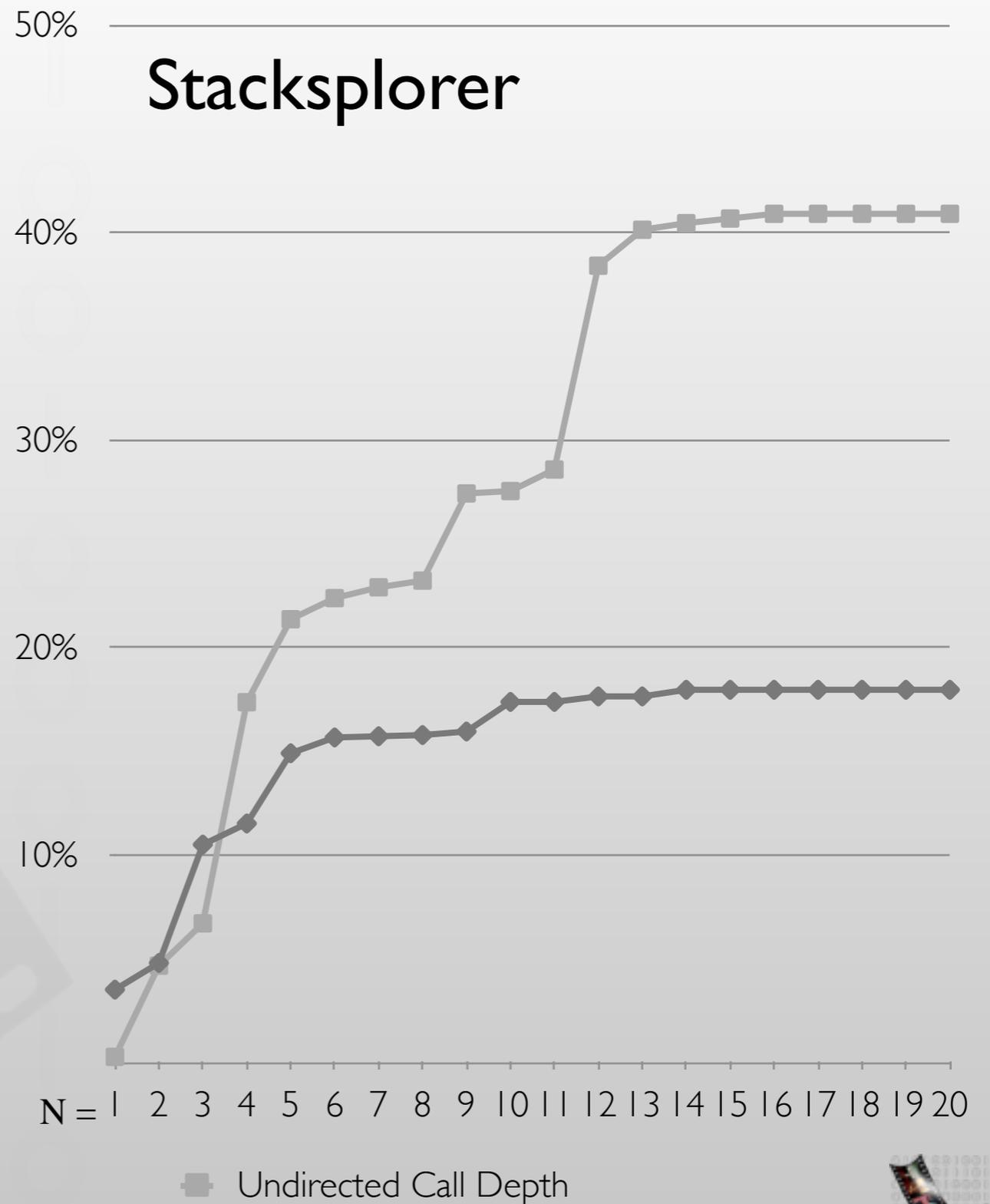


# Prediction Accuracy

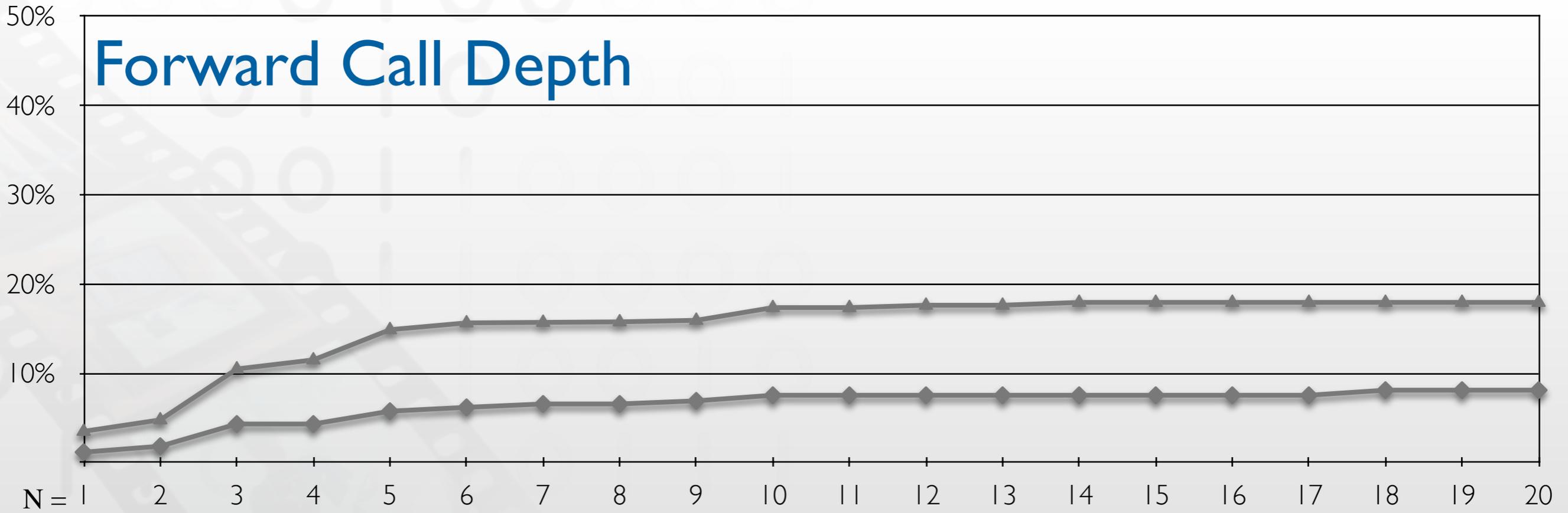
## Xcode



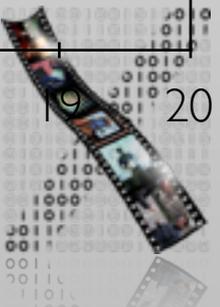
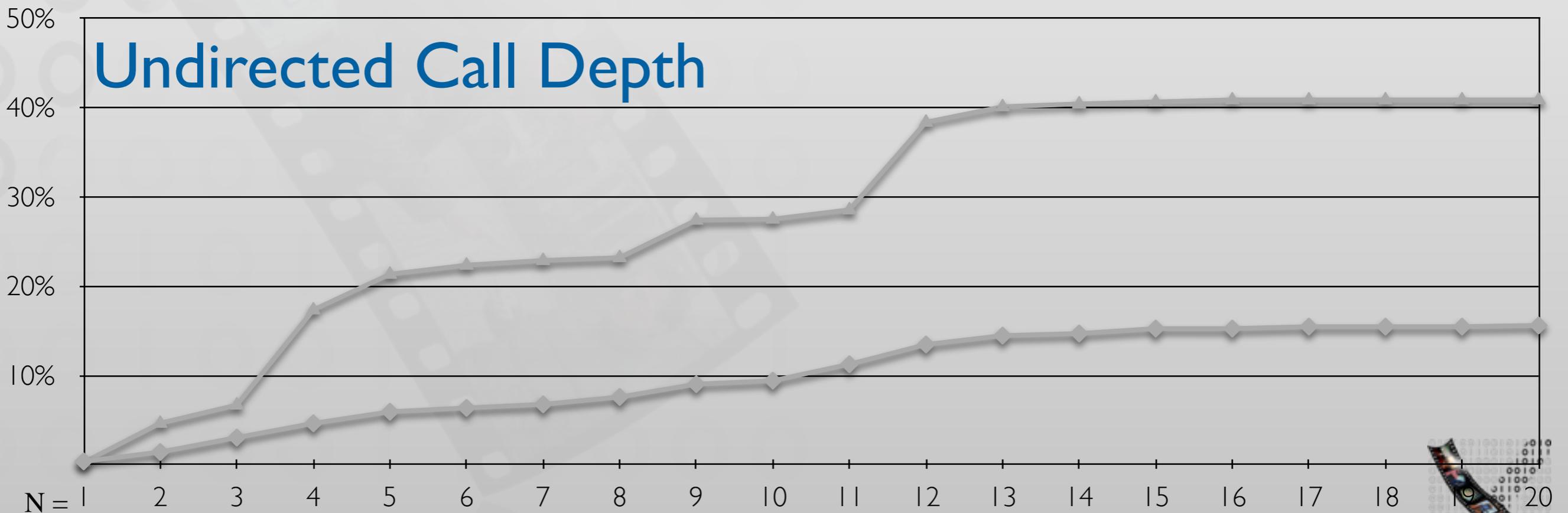
## Stacksplorer



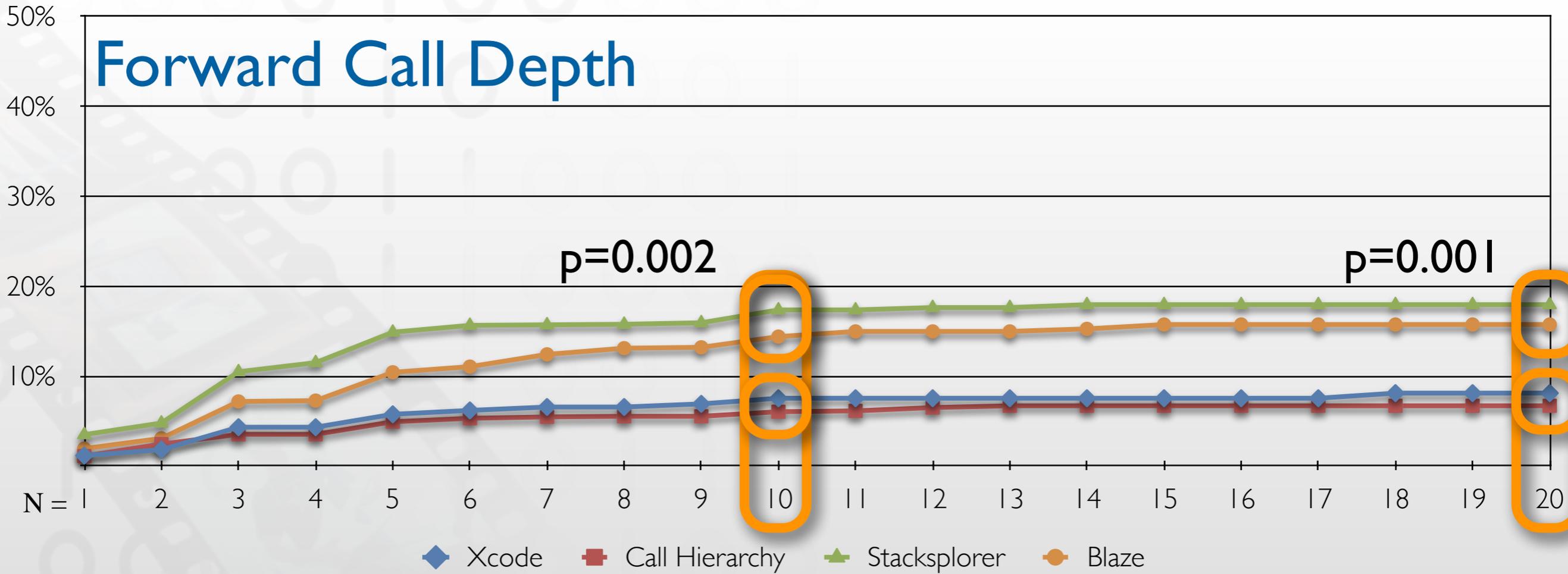
# Forward Call Depth



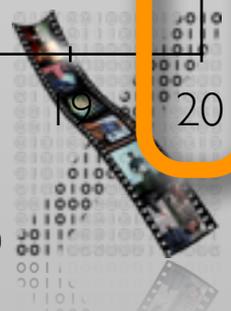
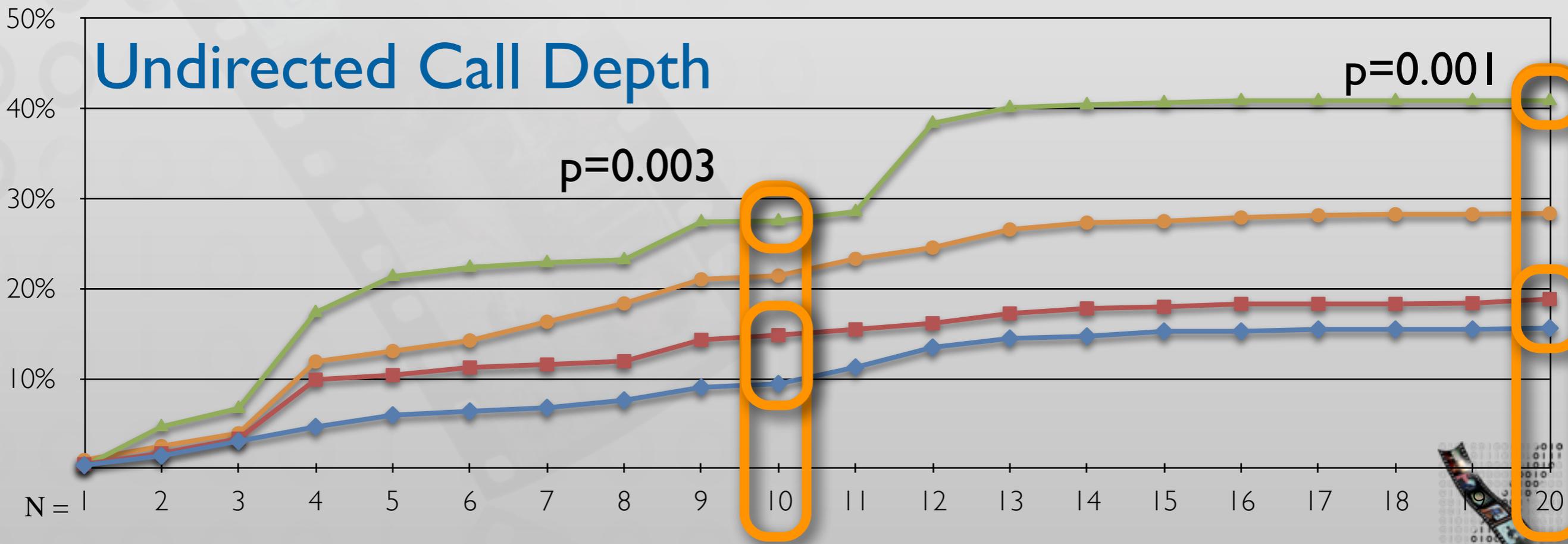
# Undirected Call Depth



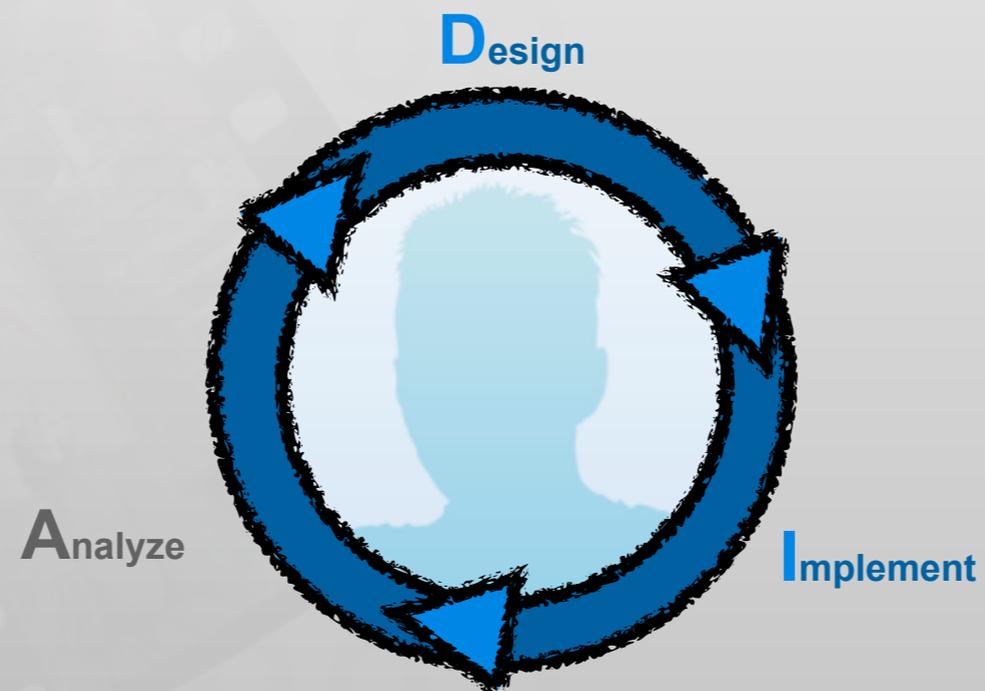
# Forward Call Depth



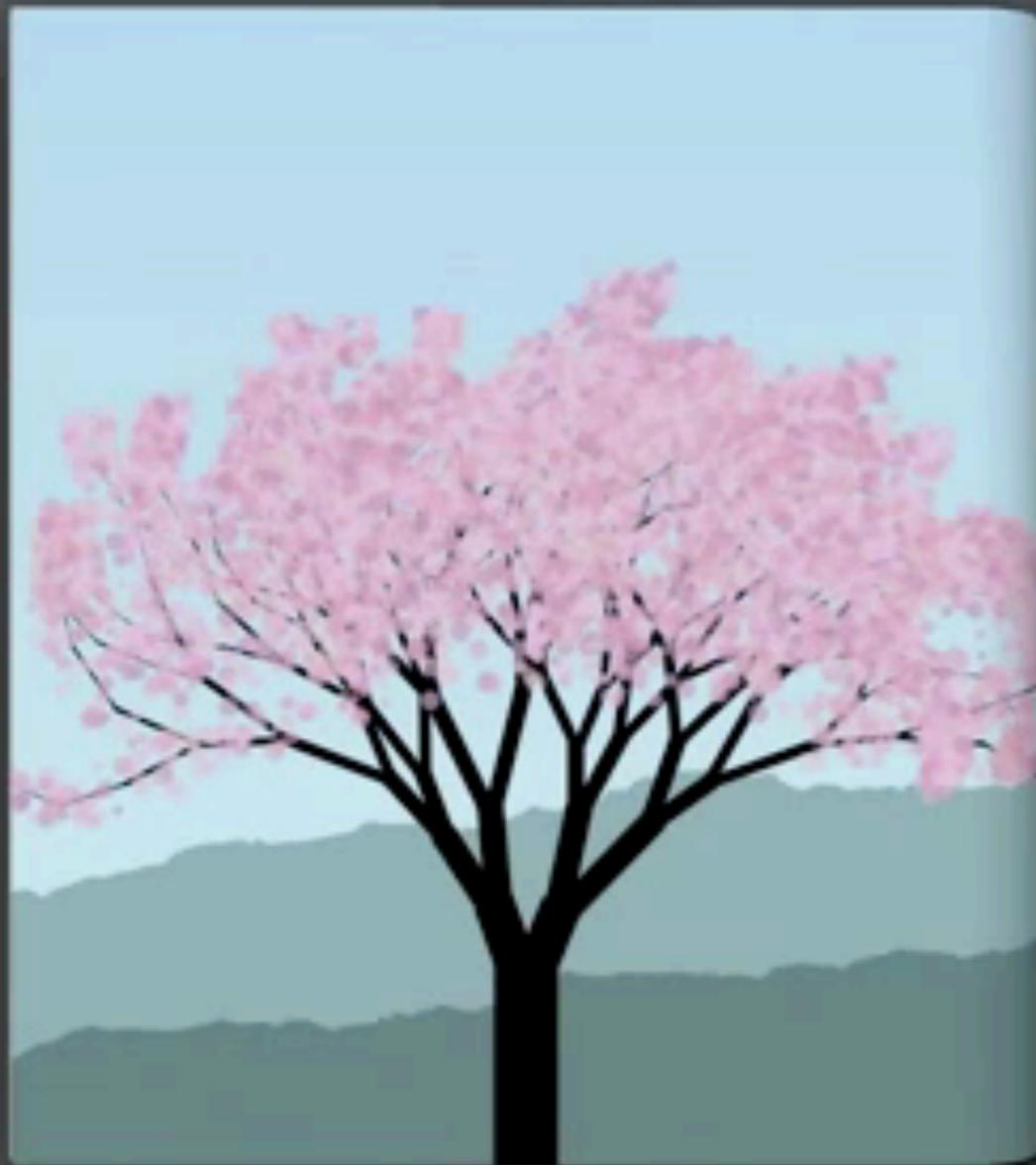
# Undirected Call Depth



# Away from static analysis only



*// Introducing Codelets...*



```
// tree
//
function drawTree () {
  var blossomPoints = [];

  resetRandom();
  drawBranches(0, -Math.PI/2, canvasWidth/2, canvasHeight, 30,
  resetRandom();
  drawBlossoms(blossomPoints);
}

function drawBranches (i,angle,x,y,width,blossomPoints) {
  ctx.save();

  var length = tween(1, 1, 60, 12, 3) + random(0.7, 1.3);
  if (i == 0) { length = 97; }

  ctx.translate(x,y);
  ctx.rotate(angle);
  ctx.fillStyle = "#f000";
  ctx.fillRect(0, -width/2, length, width);

  ctx.restore();

  var tipX = x + (length - width/2) * Math.cos(angle);
  var tipY = y + (length - width/2) * Math.sin(angle);

  if (i > 4) {
    blossomPoints.push([x,y,tipX,tipY]);
  }

  if (i < 6) {
    drawBranches(i + 1, angle + random(-0.15, -0.05) * Math.PI);
    drawBranches(i + 1, angle + random( 0.15,  0.05) * Math.PI);
  }
  else if (i < 12) {
    drawBranches(i + 1, angle + random(-0.25, -0.05) * Math.PI);
    drawBranches(i + 1, angle + random( 0.25,  0.05) * Math.PI);
  }
}
```

```

1  /*jshint plusplus: true, node: true, white: tru
2
3  var arr = [6, 2, 8, 4, 3];
4
5  var i;
6  for (i = 1; i < arr.length; i++) {
7      var j = i;
8
9      while ((j > 0) && (arr[j-1] > arr[j])) {
10         var tmp = arr[j-1];
11         arr[j-1] = arr[j];
12         arr[j] = tmp;
13
14         j = j-1;
15     }
16 }
17 console.log(arr);

```

[6, 2, 8, 4, 3]

*undefined*

< 1/4 > 1 *truthy(true)*

1

< 1/1 > *truthy(true)*

6

2

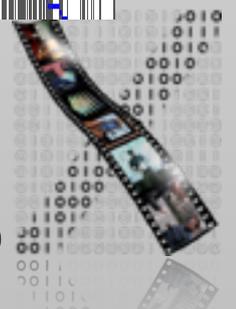
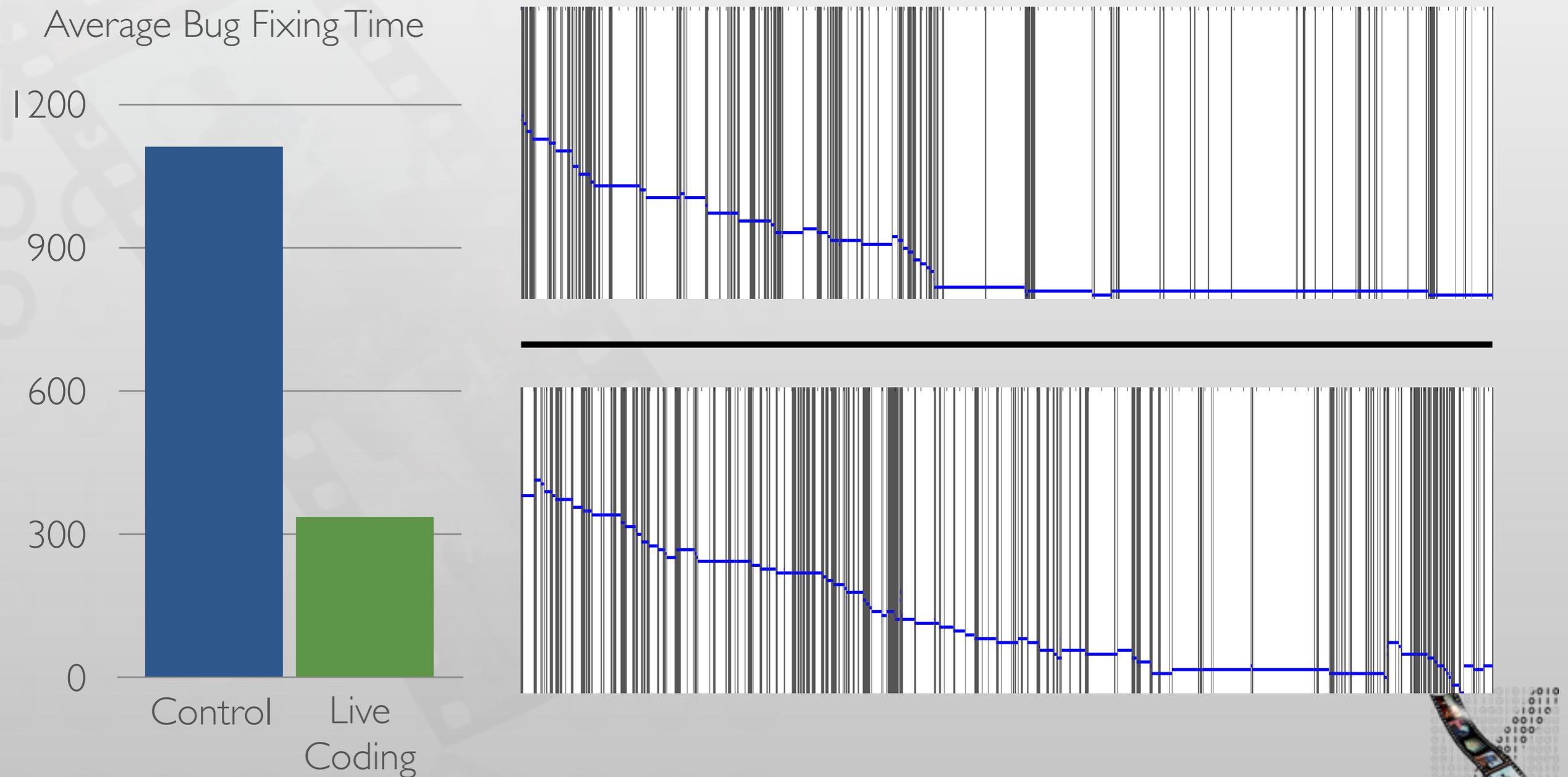
6

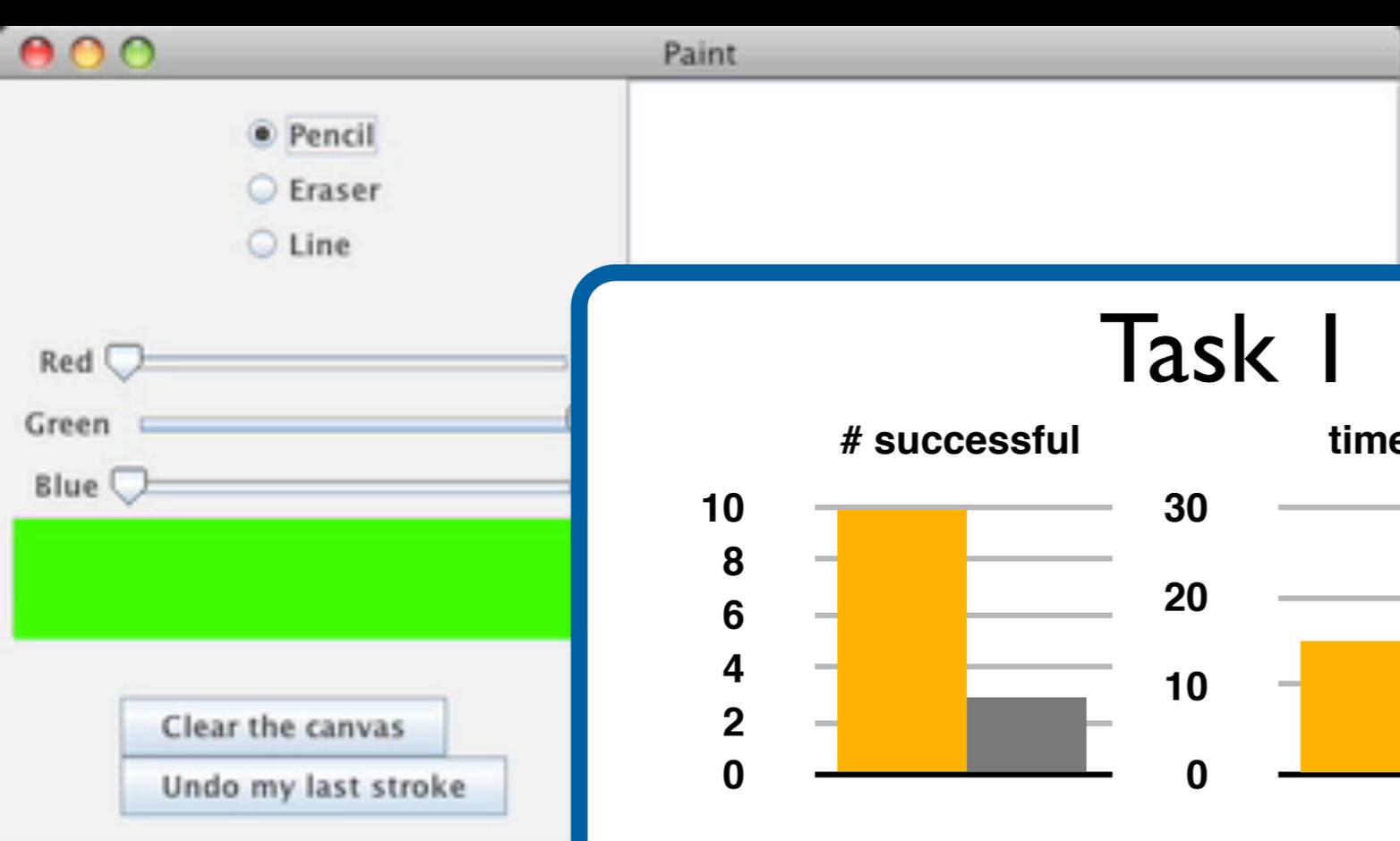
0

[2, 3, 4, 6, 8]

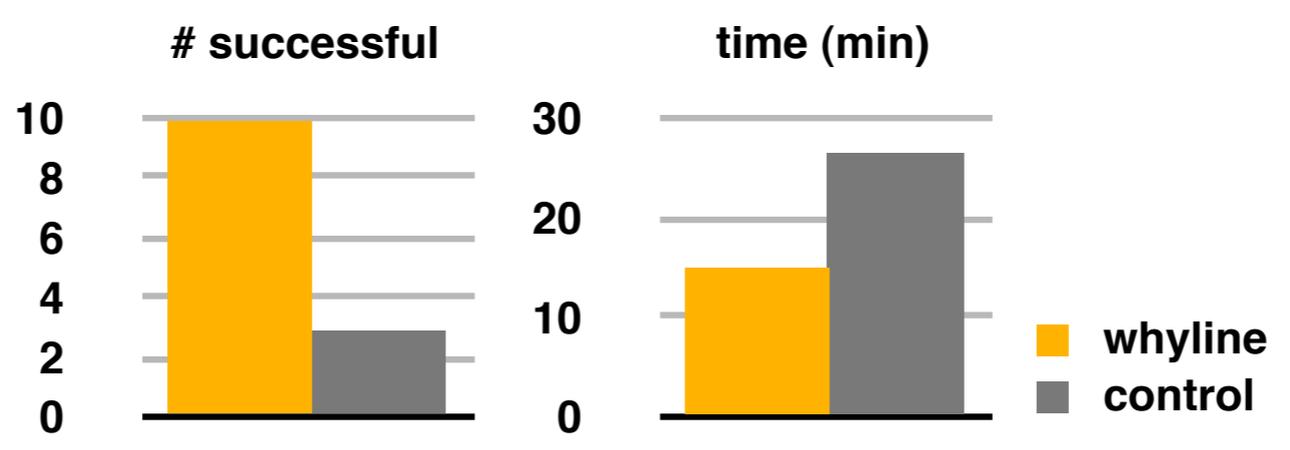
# Live Coding Affects Coding Behavior

[Krämer2014, How Live Coding Affects Developers' Coding Behavior]

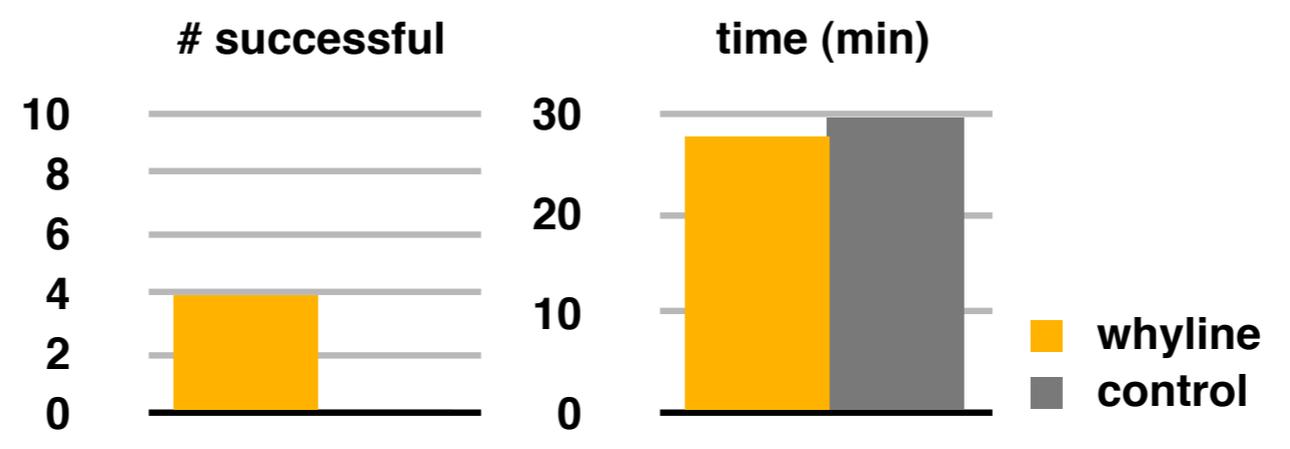




# Task 1



# Task 2



The Lorax on Blu-Ray

Television

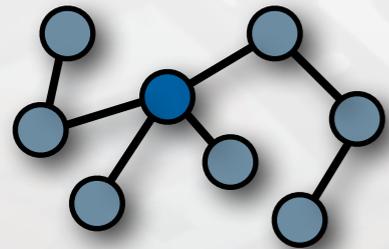
Toaster

Coconut Water

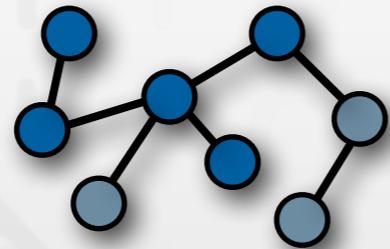
Hulk Action Figure

```
41 <script>
42 var items = ["The DaVinc
43 var basketItems = [];
44
9 calls function displayItem(nam
46     var item = $("<div c
47     item.draggable({
48         appendTo: "body"
49         helper: "clone",
0 calls     drag: function (
51         console.log(
52     },
0 calls     stop: function (
54         console.log(
55     },
56     ));
57 }
58
```

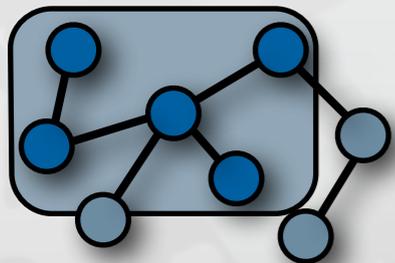
# Summary



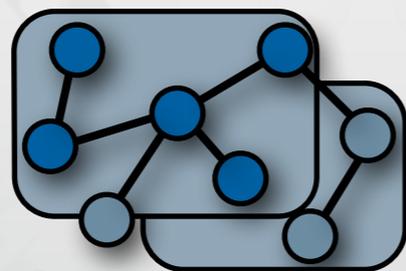
Finding focus points



Expanding focus points



Understanding a subgraph



Questions over groups of subgraphs

