

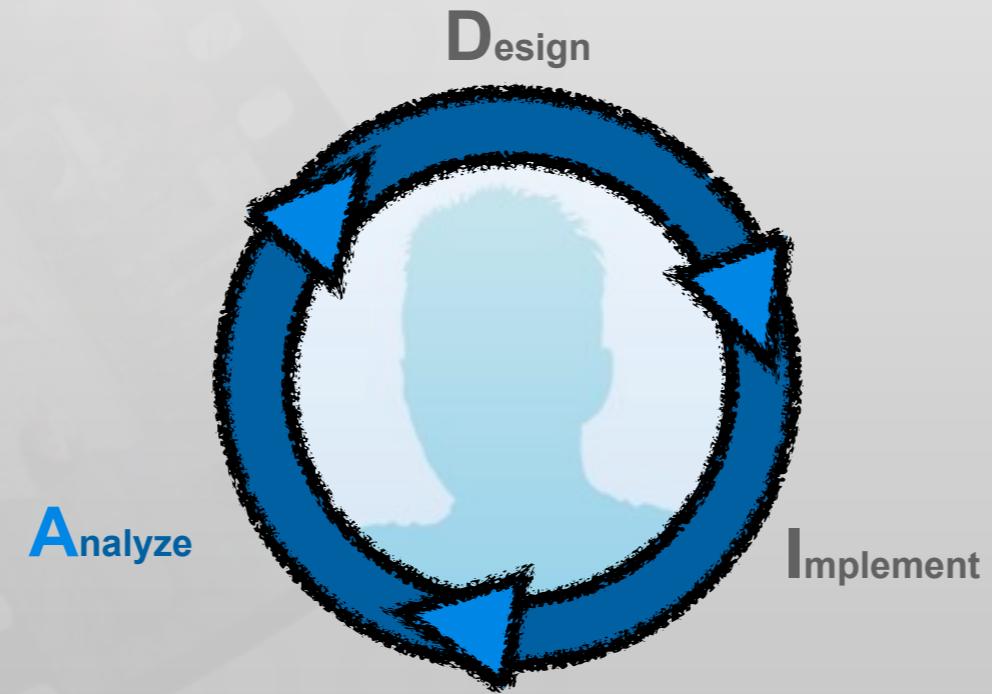
# Research in Coding and IDEs

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

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

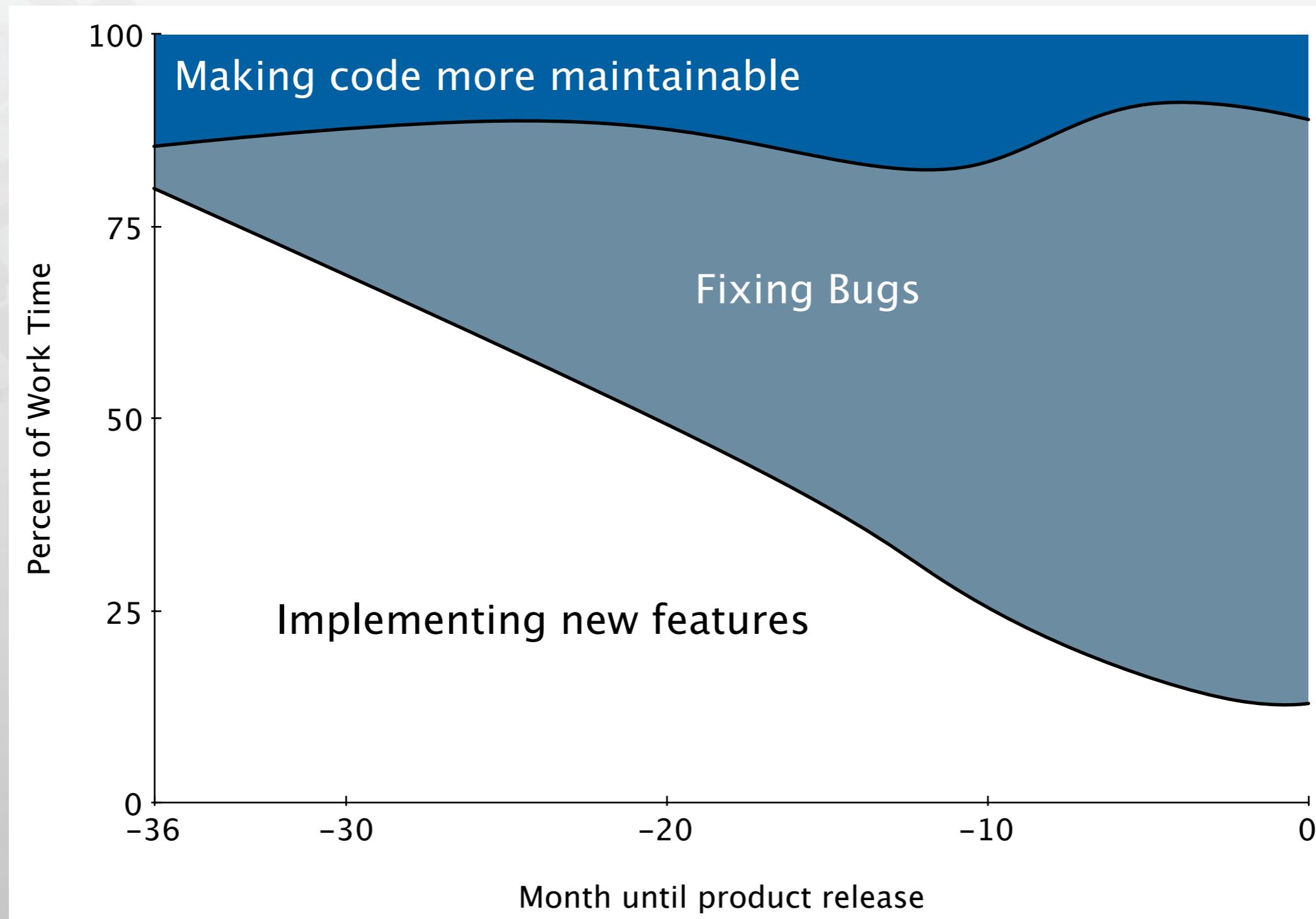


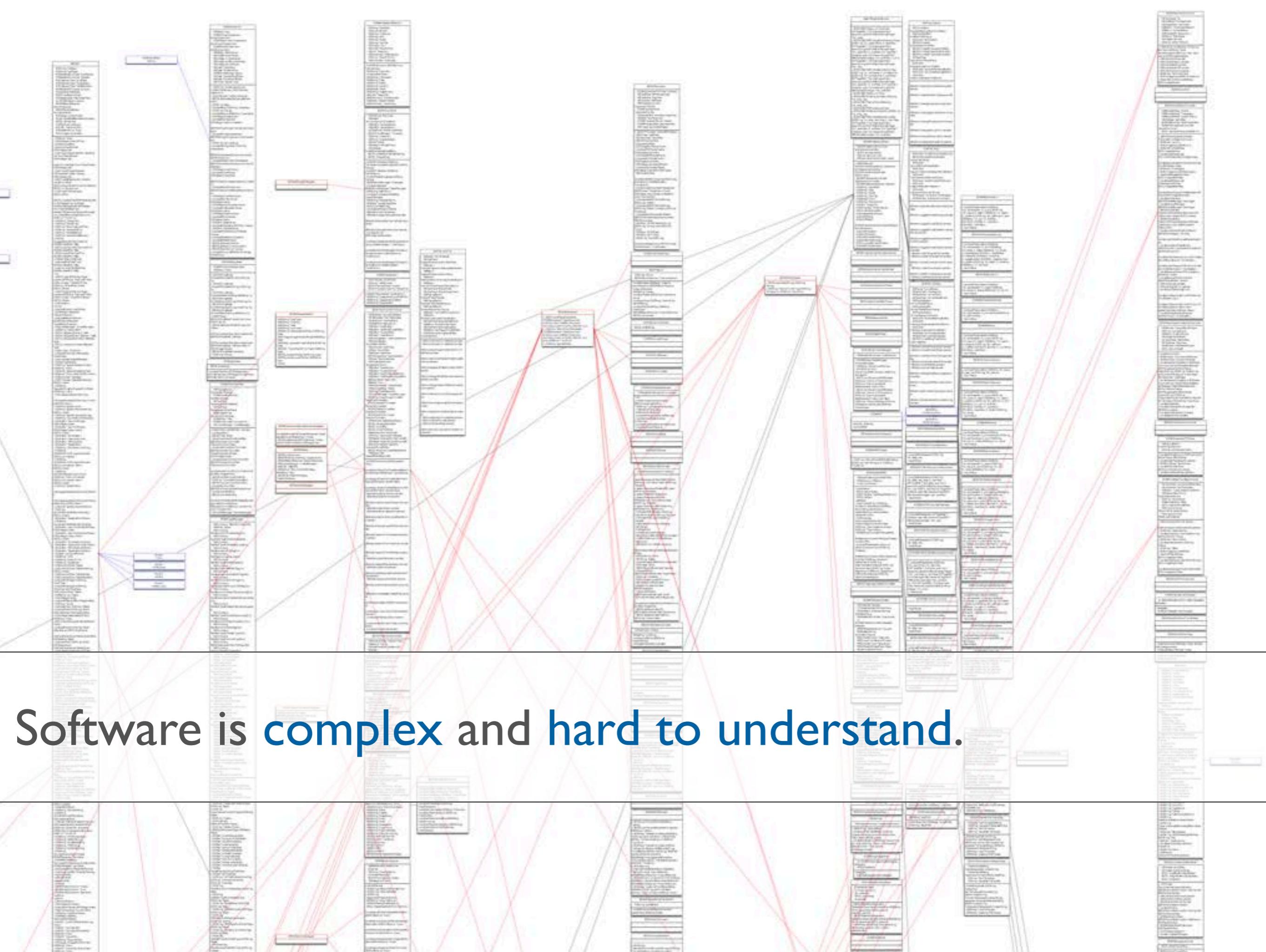
# Status Quo



# Time in Software Development

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





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

# Task context

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
#import <Cocoa/Cocoa.h>
extern NSString *BDSKFileFilterKeys;
extern NSString *BDSKFilePublicationKey;
extern NSString *BDSKFileOldPathKey;
extern NSString *BDSKFileNewPathKey;
extern NSString *BDSKFileStatusKey;
extern NSString *BDSKFileLaptopKey;
extern NSString *BDSKFileFolioKey;
extern NSString *BDSKFileFolioKey;
@class BibDocument;
enum {
    BDSKNoError = 0,
    BDSKSourceFileDoesNotExistErrorMask = 1,
    BDSKTargetFileExistsErrorMask = 2,
    BDSKCannotMoveFileErrorMask = 4,
    BDSKCannotRemoveFileErrorMask = 8,
    BDSKCannotResolveAliasErrorMask = 16,
    BDSKCannotCreateParentErrorMask = 32,
    BDSKIncompleteFieldErrorMask = 64
};

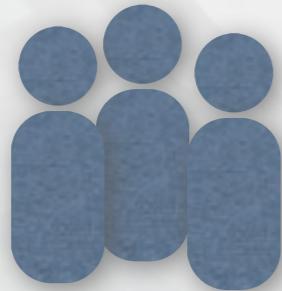
@interface BDSKFilter : NSWindowController {
    IBOutlet NSProgressIndicator *progressIndicator;
}
+ (BDSKFilter *)sharedFilter;
// ...
method autoFileLinkedFiles:(BDSKDocument *)doc check:
{
    abstract Main auto-file routine to file papers in the Papers folder according to a generated location.
    papers An array of linked files to be auto-filed.
    doc The parent document of the papers.
    check Boolean determines whether to move only entries with all necessary fields set.
    discussion This is the main method that should be used to autofile papers.
    It calls the necessary methods to do the move and generates the new locations for them.
    - (NSURL *)suggestedURLForLinkedFile:(BDSKLinkedFile *)file
    {
        NSString *papersFolderPath = [BDSKFormatParser folderPathForBibItem:file];
        NSString *relativeFile = [BDSKFormatParser parseFormatFile:file];
        if ([[NSUserDefaults standardUserDefaults] boolForKey:BDSKInitialAutoFileOptionMask])
            relativeFile = [relativeFile lowercaseString];
        return [NSURL fileURLWithPath:[papersFolderPath stringByAppendingPathComponent:relativeFile]];
    }
    method movePapers:(NSArray *)paperInfos fromDocument:(BDSKDocument *)doc check:
    {
        abstract Tries to move list of papers from a document.
        paperInfos A list of info dictionaries containing a BibItem, a BDSKLinkedFile field for which to move the linked files.
        doc The parent document of the papers.
        check mask Integer, see the AutoFileOptions.
        discussion This is the core method to move files.
        It is unduplicable, but only moves that were successful.
        It can handle aliases and symlinks, plus when they are moved, aliases and symlinks are moved unexpanded. Relative BDSKInitialAutoFileOptionMask should be used for it.
        BDSKCheckIncompleteAutoFileOptionMask indicates that BDSKShareAutoFileOptionMask forces AutoFiling, even if it fails.
        - (void)movePapers:(NSArray *)paperInfos forFields:(NSDictionary *)fields
    }
    interface NSFileManager (BDSKFilterExtensions)
    // ...
    method movePath:(NSString *)path newPath:(NSString *)newPath
    {
        abstract Extension to movePathToPathHandler.
        path The path to the file to move.
        newPath The path where the file should be moved.
        force Boolean. If YES, overwrites a file if it exists.
        discussion -
        - (BOOL)movePath:(NSString *)path newPath:(NSString *)newPath
    }
}
```

The screenshot shows several windows from the Xcode IDE. In the top-left window, there is a snippet of Objective-C code related to file management and progress indicators. In the bottom-left window, there is another snippet of code involving NSFileManager extensions. The central part of the image shows a detailed documentation view for the `NSProgressIndicator` class. The documentation includes sections for `maxValue`, `minValue`, and `pub`. The `maxValue` section describes it as returning the maximum value for a determinate progress bar, with a note that by default, a determinate progress bar's maximum value is 100.0. The `minValue` section describes it as returning the minimum value for a determinate progress bar. The `pub` section shows a line of code: `pub = [paperInfo valueForKey:BDSKFilterPublicationKey];`. To the right of the documentation, there is a sidebar with links to "Mutable Array", "NSMutableArray Class Reference", "Class Methods", and "arrayWithCapacity". Below the sidebar, there is a section titled "Hacking BibDesk" by Michael O. McCracken, with a note about the document being a place for notes explaining various tricky things about how BibDesk is organized. There is also a "Class Hierarchy notes" section with a note about separate but incomplete HTML documentation. At the bottom right, there is a decorative graphic of a film strip.

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

# 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)

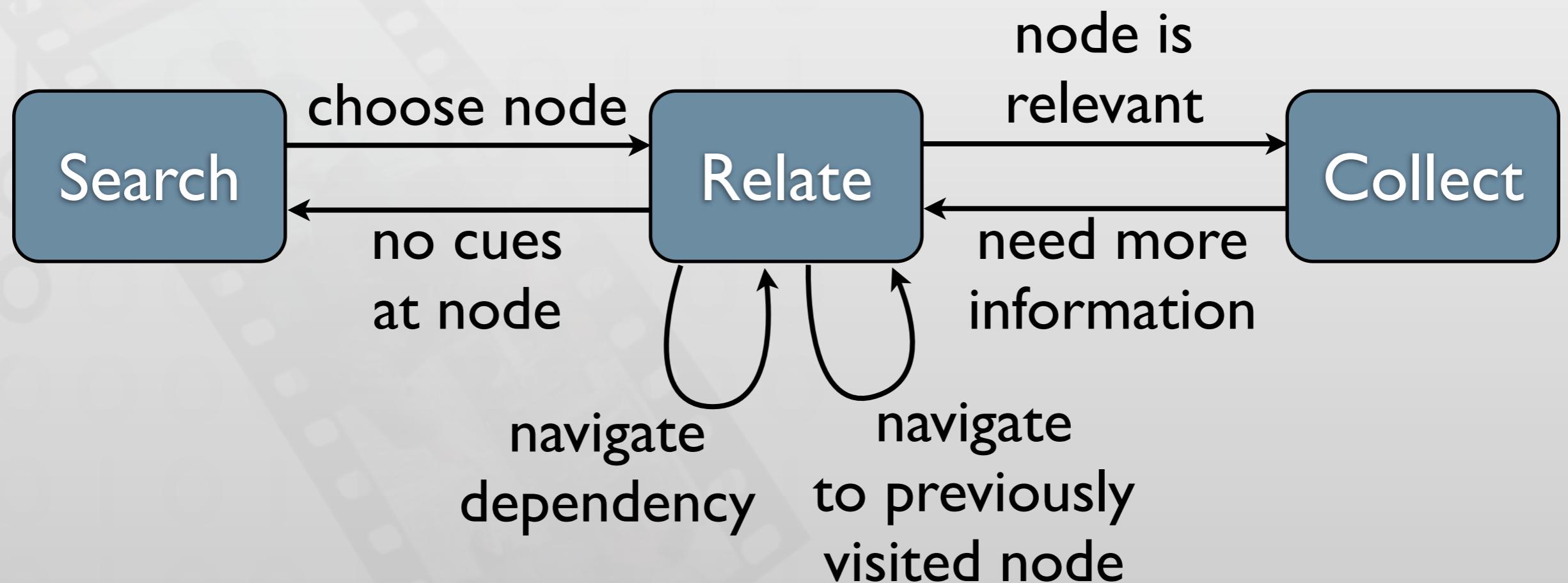


500 SLOC Java Paint  
Application



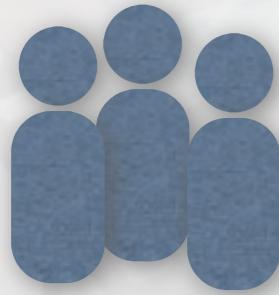
# Models for Developer Strategies

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

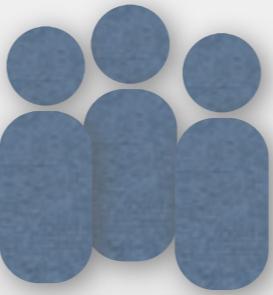


# Models for Developer Strategies

[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



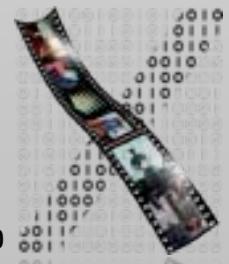
Real world change  
task



ArgoUML  
60k SLOC

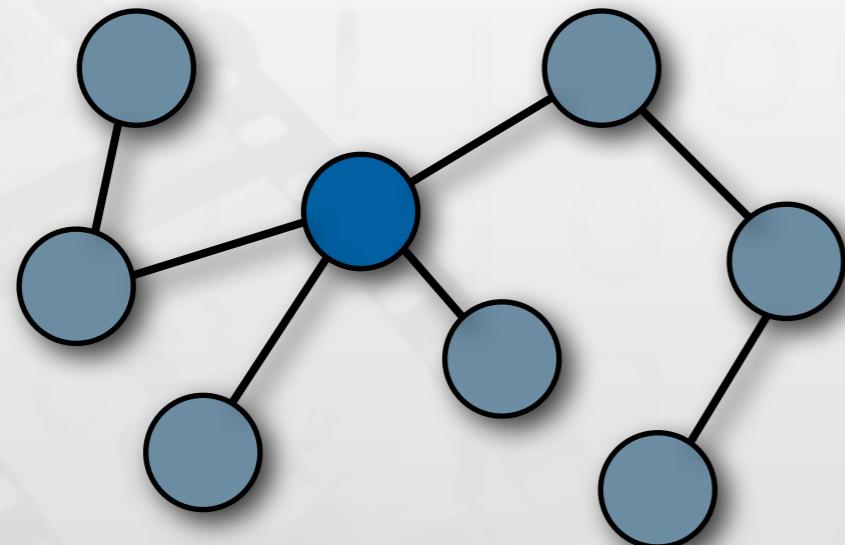


Real world sour code

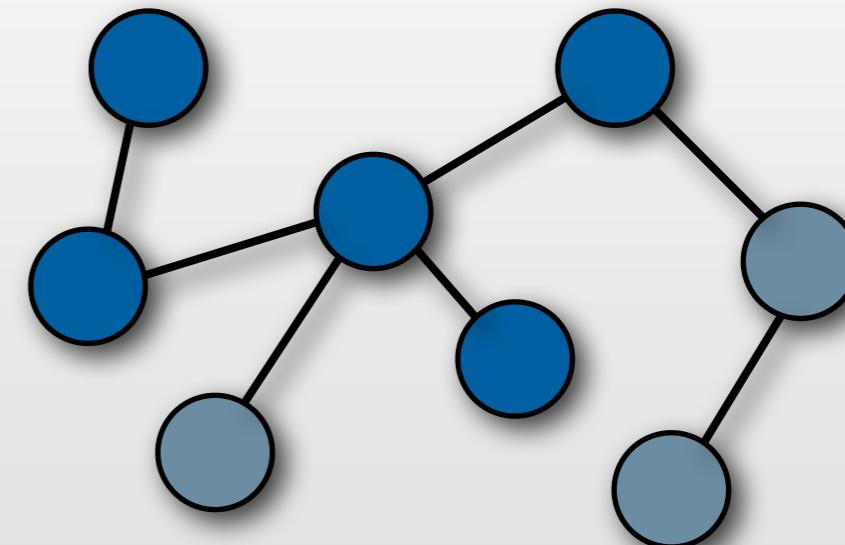


# Models for Developer Strategies

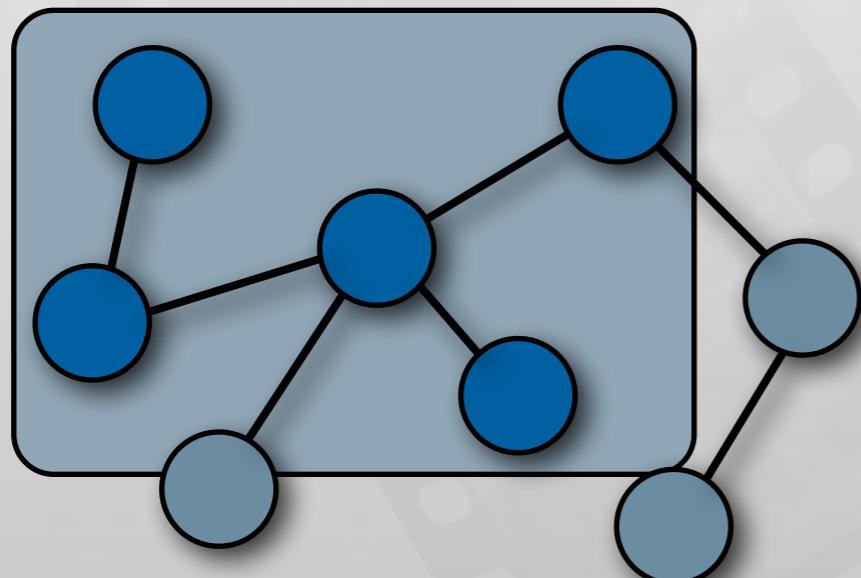
[Sillito2008, Asking and Answering Questions during a Programming Change Task]



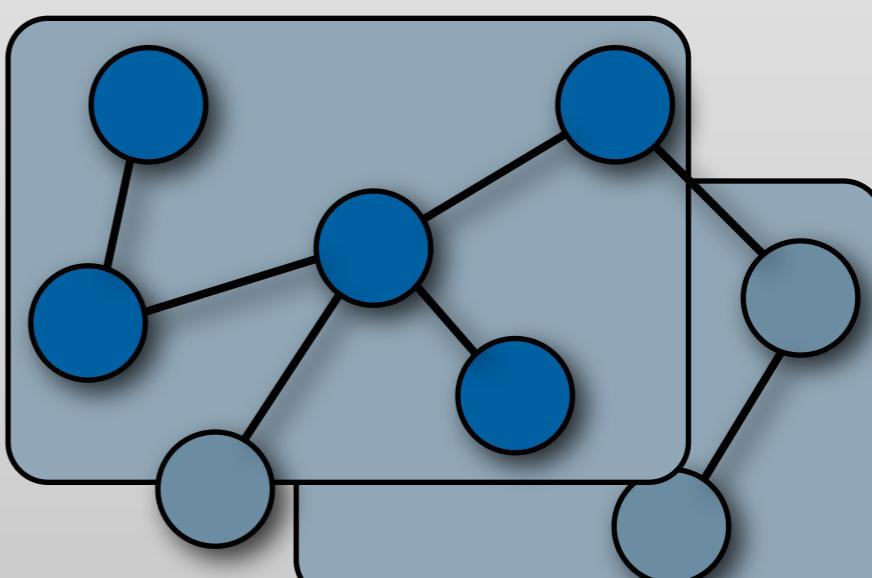
Finding focus points



Expanding focus points



Understanding a subgraph



Questions over groups  
of subgraphs



Package Explorer Type Hierarchy

Desktop.java DesktopEvent.java DesktopEventService.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

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

Problems Javadoc Declaration Call Hierarchy

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

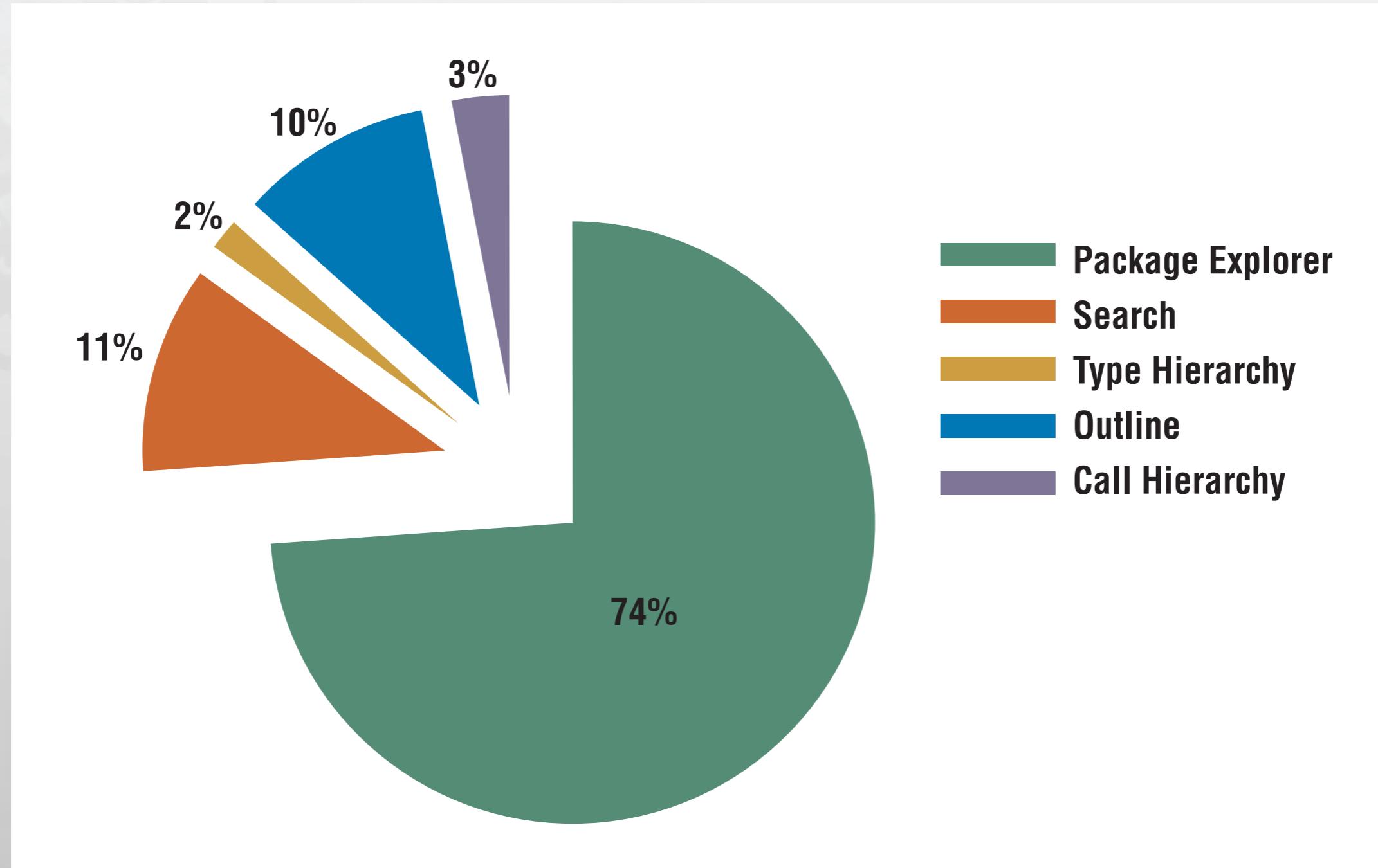
Line Call

JRE System Library [JavaSE-1.6]

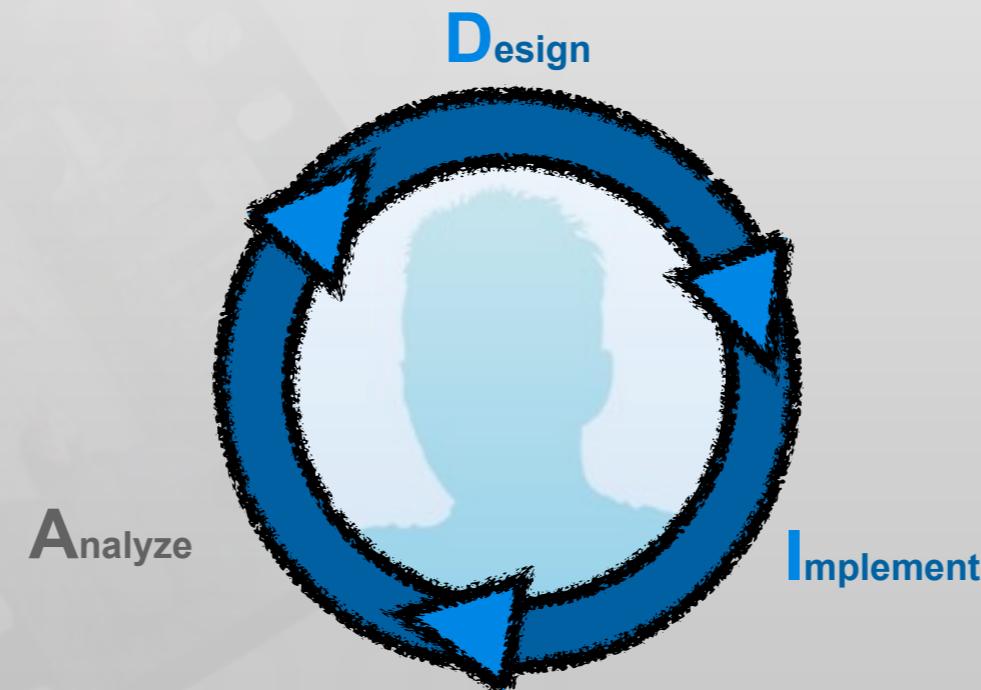
org.jhotdraw.samples.javadraw.Animator.java - JHotDraw/src

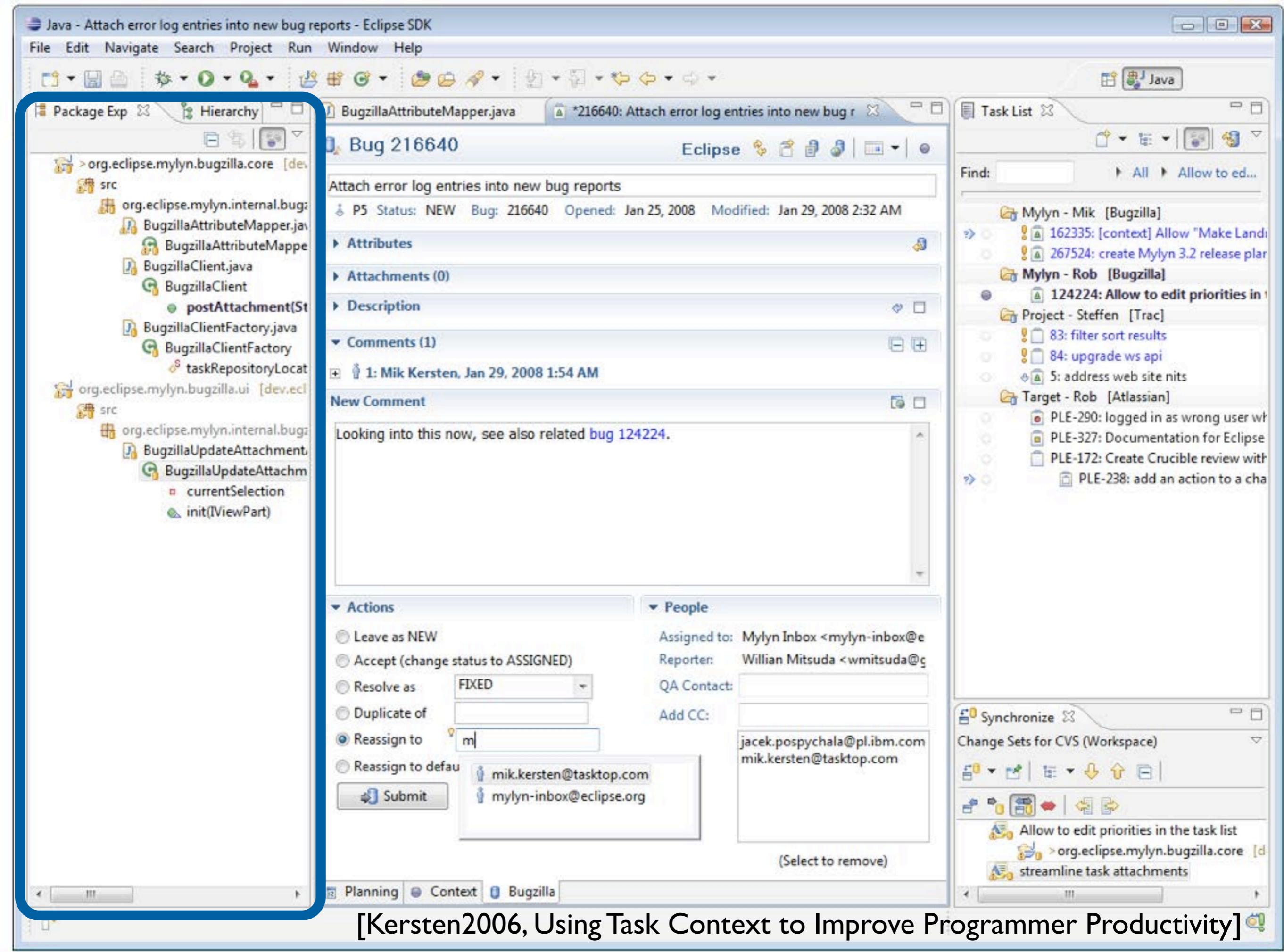
# Tools Used in Eclipse

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



# Easing Access to Task Context





[Kersten2006, Using Task Context to Improve Programmer Productivity]

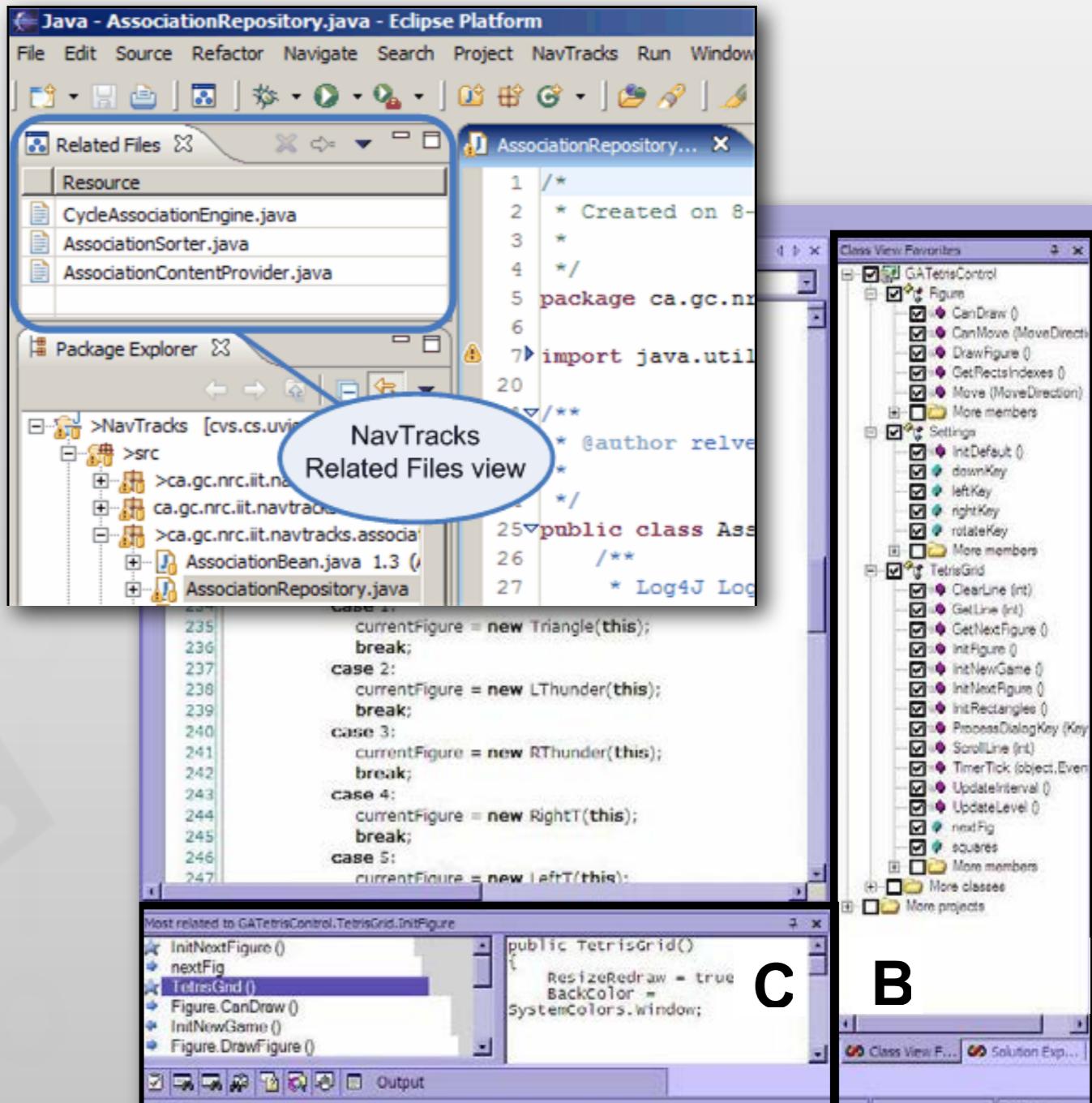
# Recommender Tools

[Singer2005, NavTracks: supporting navigation in software maintenance]

[DeLine2005, Easing program comprehension by sharing navigation data]

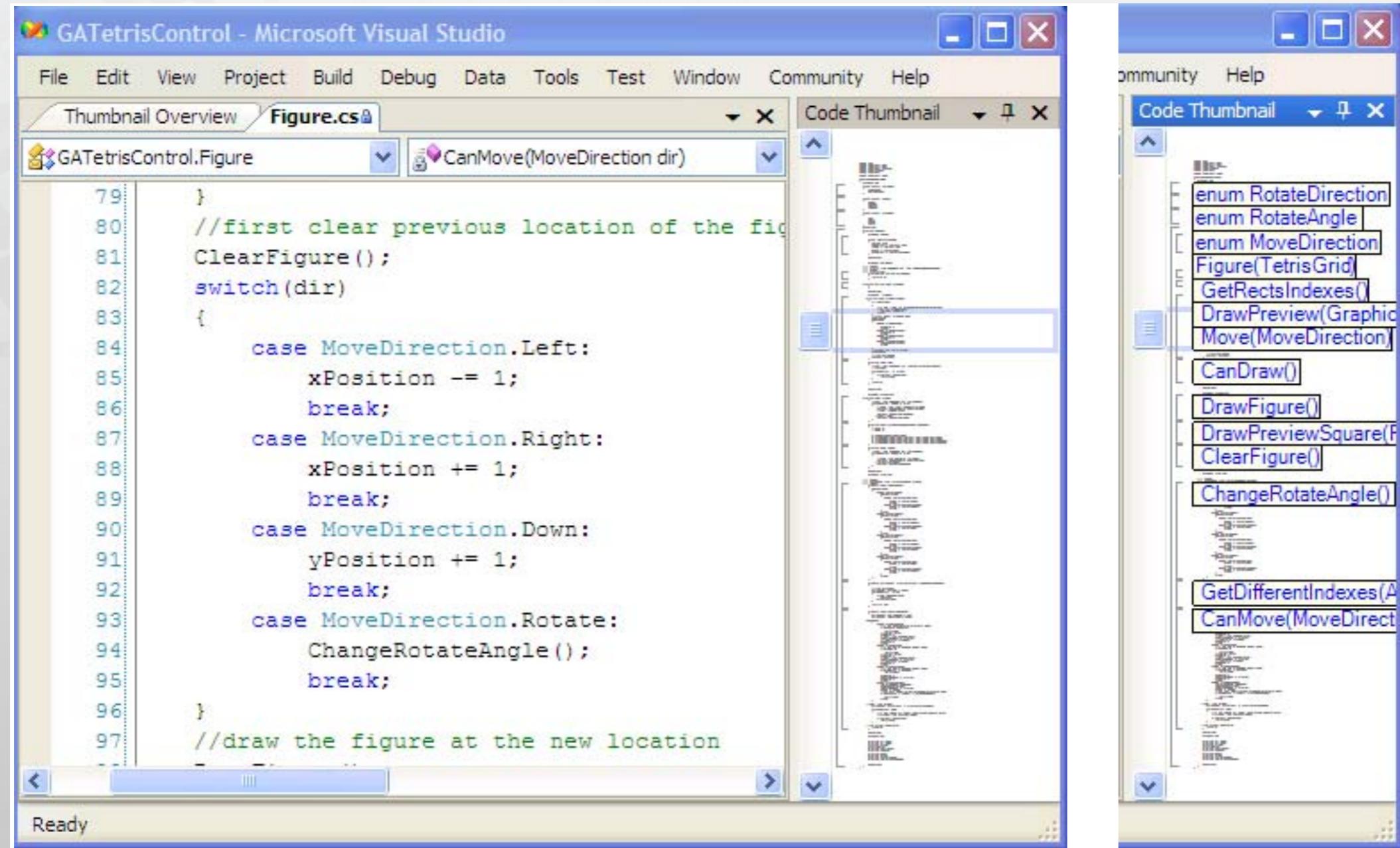
[Čubranić 2005, Hipikat: recommending pertinent software development artifacts]

- 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



# Changing the Presentation

[DeLine2006, Code Thumbnails: Using Spatial Memory to Navigate Source Code]



# Changing the Presentation

[Sublime Text 2, <http://www.sublimetext.com/2>]

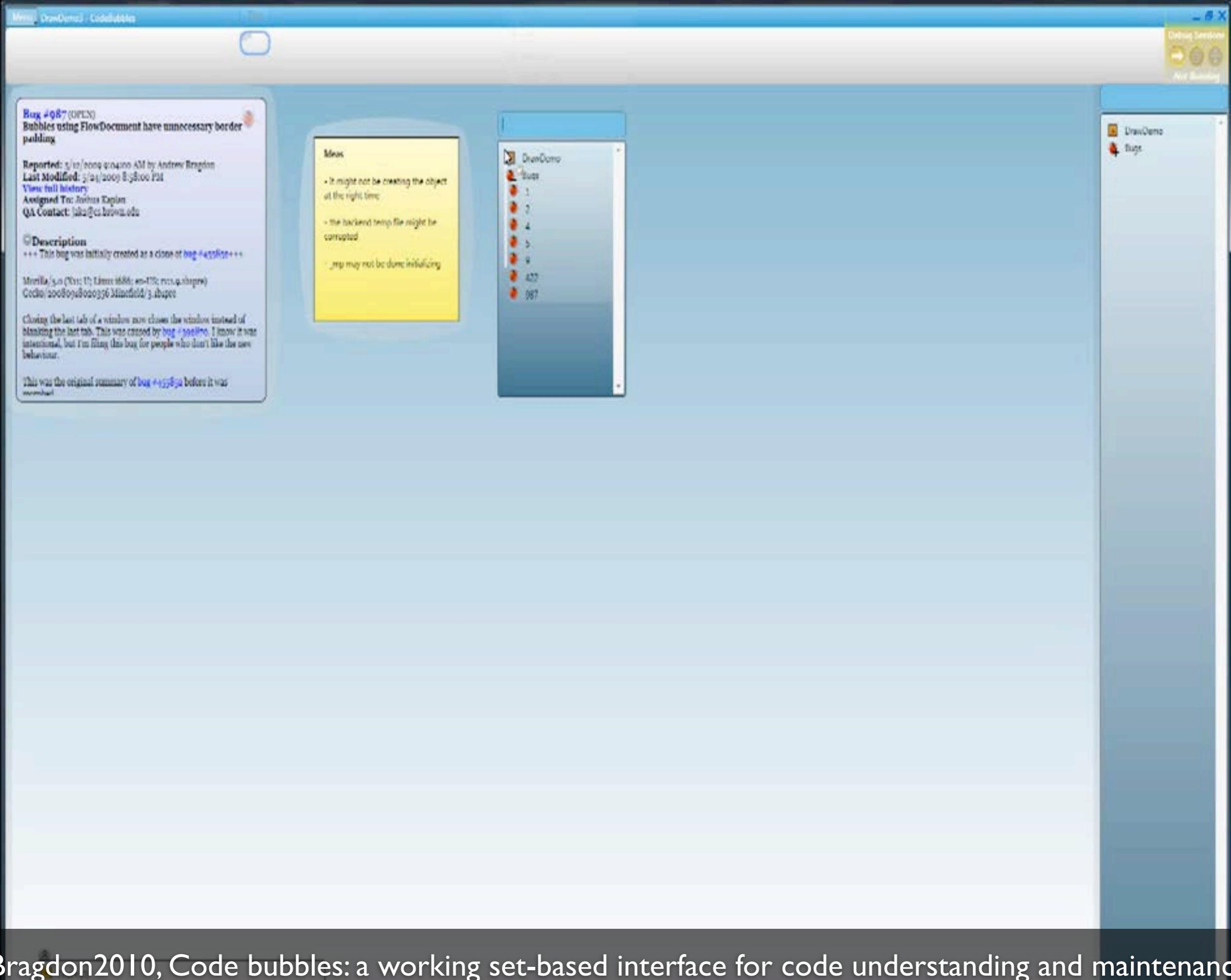


A screenshot of Sublime Text 2 showing the file `DocumentManager.js` in the `src` folder. The code is written in JavaScript and handles document editing and reloading. The interface includes a sidebar with project files like `base-config`, `command`, and `language`. A status bar at the bottom shows git branch information and file statistics.

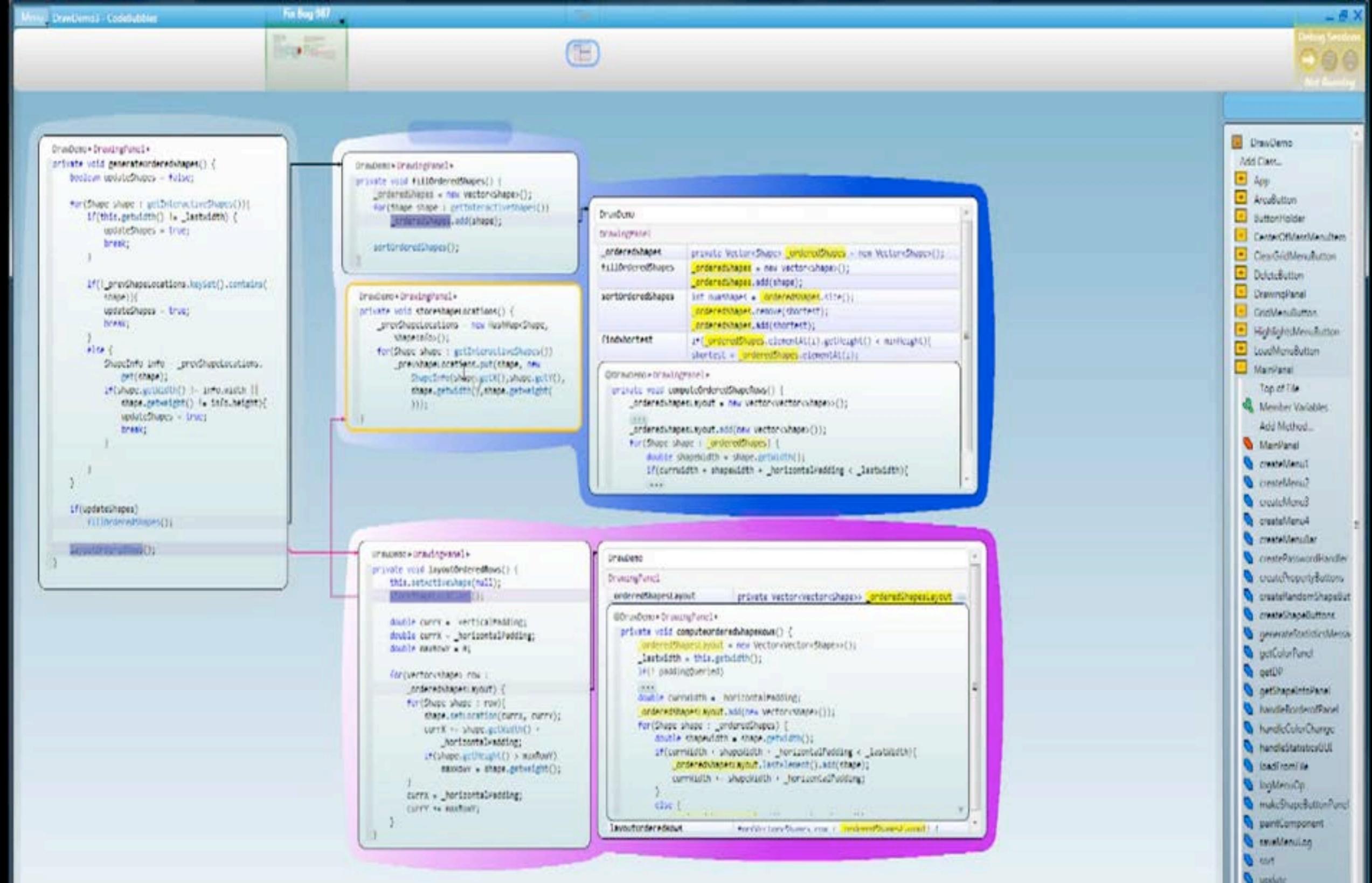
```
Document.prototype.getText = function (useOriginalLineEndings) {
    if (this._masterEditor) {
        // CodeMirror.getValue() always returns text with LF line endings; fix up to match line
        // endings preferred by the document, if necessary
        var codeMirrorText = this._masterEditor._codeMirror.getValue();
        if (useOriginalLineEndings) {
            if (this._lineEndings === FileUtils.LINE_ENDINGS_CRLF) {
                return codeMirrorText.replace(/\n/g, "\r\n");
            }
        }
        return codeMirrorText;
    } else {
        // Optimized path that doesn't require creating master editor
        if (useOriginalLineEndings) {
            return this._text;
        } else {
            return this._text.replace(/\r\n/g, "\n");
        }
    }
};

/**
 * Sets the contents of the document. Treated as an edit. Line endings will be rewritten to
 * match the document's current line-ending style.
 * @param {!string} text The text to replace the contents of the document with.
 */
Document.prototype.setText = function (text) {
    this._ensureMasterEditor();
    this._masterEditor._codeMirror.setValue(text);
    // _handleEditorChange() triggers "change" event
};

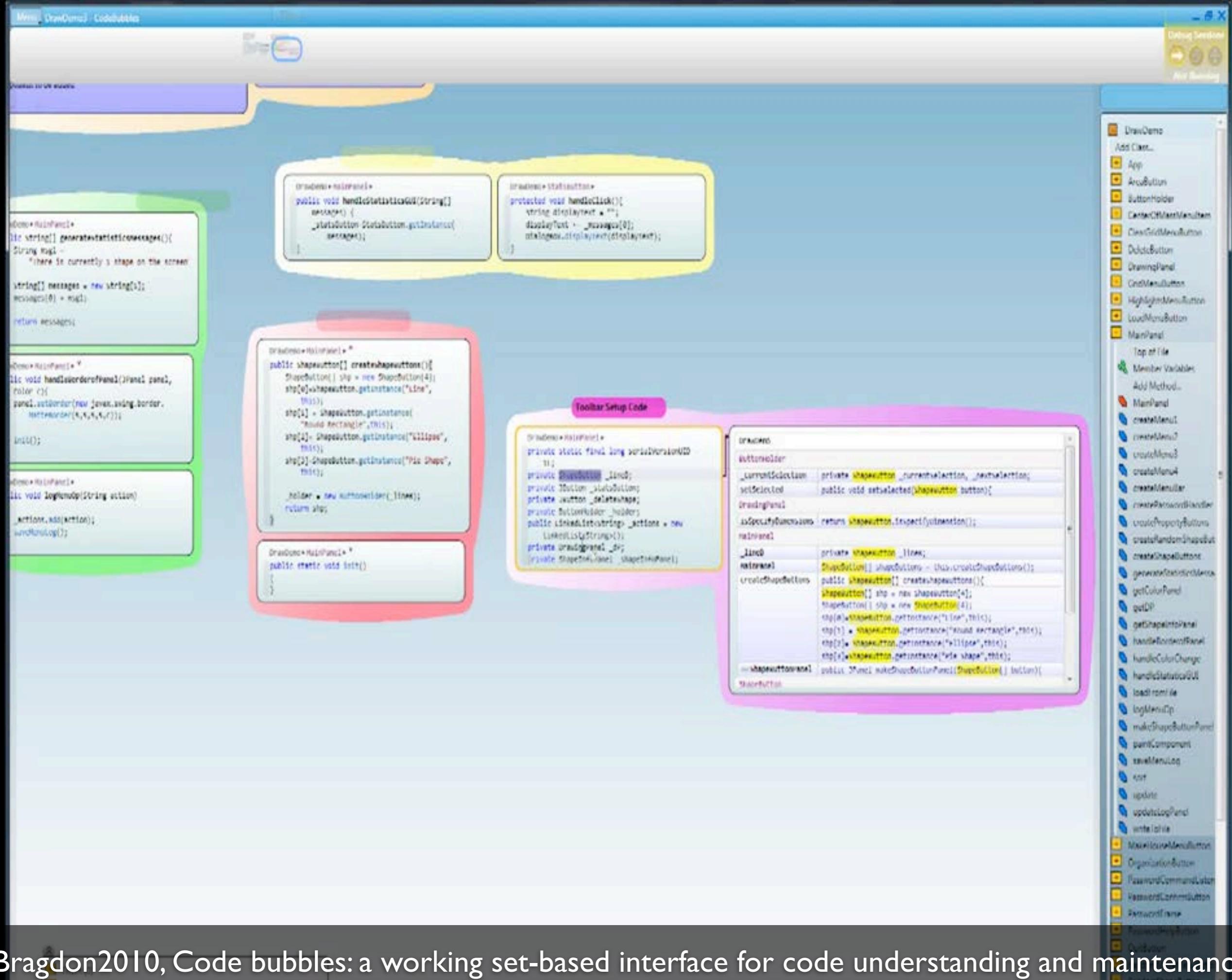
/**
 * Sets the contents of the document. Treated as reloading the document from disk: the document
 * will be marked clean with a new timestamp, the undo/redo history is cleared, and we re-check
 * the text's line-ending style. CAN be called even if there is no backing editor.
 * @param {!string} text The text to replace the contents of the document with.
 * @param {!Date} newTimestamp Timestamp of file at the time we read its new contents from disk.
 */
Document.prototype.refreshText = function (text, newTimestamp) {
    var perfTimerName = PerfUtils.markStart("refreshText:\t" + (!this.file || this.file.fullPath));
}
```



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



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



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

File Bug #87

```

DrawDemo.java
private void generateOrderShapes() {
    boolean updateShapes = false;

    for(Shape shape : getInteractiveShapes()) {
        if(this.getJarWidth() > _lastJarWidth) {
            updateShapes = true;
            break;
        }

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

    if(updateShapes)
        fillOrderShapes();
}

layoutOrderedRows();

```

```

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

    sortOrderShapes();
}

DrawDemo.java
private void sortOrderShapes() {
    _prevShapesLocations = new HashMap<Shape,
        ShapeInfo>();
    for(Shape shape : getInteractiveShapes())
        _prevShapesLocations.put(shape, new
            ShapeInfo(shape.getWidth(), shape.getHeight(),
            shape.getLocation().x, shape.getLocation().y));
}

DrawDemo.java
private void computeOrderShapes() {
    _orderedShapesLayout = new Vector<Vector<Shape>>();
    int curWidth = _jarWidth;
    _orderedShapesLayout.add(new Vector<Shape>());
    for(Shape shape : _orderedShapes) {
        double shapeWidth = shape.getWidth();
        if(curWidth + shapeWidth + _horizontalPadding < _jarWidth)
            curWidth = _jarWidth;
        else
            _orderedShapesLayout.get(_orderedShapesLayout.size() - 1).add(shape);
        curWidth += shapeWidth + _horizontalPadding;
    }
}

```

```

DrawDemo.java
public void handleOrderShapes() {
    DrawingPanel dp = this.getDP();
    dp.update();
}

DrawDemo.java
public Shape createShapeFromString(String text) {
    Shape s = null;
    if(this._currentButtonShape != null) {
        this._currentButtonShape.setLabel(text);
        this._currentButtonShape.setShape(s);
        this._currentButtonShape.setInteractive(true);
        dp.setInteractiveShape(s);
    }
}

DrawDemo.java
public void setSpecificDimensions() {
    if(_currentButtonShape != null) {
        _currentButtonShape.setWidth(
            _specificWidth, _specificHeight);
        _currentButtonShape.setInteractive(true);
    }
}

DrawDemo.java
public void updateDisplay() {
    for(int i = 0; i < _list.length; i++) {
        update(i);
    }
    _updated = true;
}

```

Things to Remember

- OrderShapesLayout is what you want, not orderedShapes
- Be sure to call layoutOrderRows first

DrawDemo

- Add Class...
- App
- ArcButton
- ButtonHolder
- CenterOfMenuMenuItem
- CloseGridMenuItem
- DeleteButton
- DrawingPanel
- GridMenuItem
- HighlightMenuItem
- LoadMenuItem
- ManPanel
- Top of file
- Member Variables...
- Add Method...
- ManPanel
- createMenu1
- createMenu2
- createMenu3
- createMenu4
- createMenuBar
- createPasswordField
- createPropertyButtons
- createRandomShapePanel
- createShapeButton
- generateStaticLabels
- getColorPanel
- getDP
- getShapePanel
- handleOrderPanel
- handleColorChange
- handleStatisticalUI
- loadFromXML
- logMenuItem
- makeShapeButtonPanel
- paintComponent
- readFile
- cut
- update
- updateLogPanel
- writeToFile
- MaximizeBoxMenuItem
- OrganizationButton
- PasswordCommandButton
- RetractCommandButton
- RetractFrame
- PasswordField
- Orientation
- TextPanel

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

The screenshot shows a software interface with a toolbar at the top featuring tabs like 'Fix Bug #87', 'Code Review w/ Devs', 'Log Book', 'M3 Planning', and 'Debug Sessions'. A central window displays a code editor with Java-like code and a code bubble highlighting the 'mainPanel' variable. To the right is a sidebar with a tree view of 'DrawDemo' components and a 'Member Variables' section. At the bottom right is a 'Opened' session panel.

```
DrawDemo.java
public class DrawDemo {
    public static void main(String[] args) {
        DrawDemoPanel mainPanel = new DrawDemoPanel();
        mainPanel.setVisible(true);
    }
}
```

Code Bubble:

```
DrawDemoPanel.java
public class DrawDemoPanel extends JPanel {
    public DrawDemoPanel() {
        this.setLayout(new BorderLayout());
        this.createPropertyButtons();
        JButton featureButton = FeatureButton.getInstance(this);
        JButton randomShapes = this.createRandomShapesButton();
        JMenuItem messageMenuItem = this.createMessageMenuItem();
        this.addMouseListener(messageMenuItem);
        MouseListener mouseListener = this.createMouseListener();
        ShapeButton[] shapeButtons = this.createShapeButtons();
        JPanel shapePanel = this.createShapeButtonsPanel(shapeButtons);
        this.createShapeHandler();
        JPanel moreFunctionsPanel = new JPanel();
    }
}
```

Code Bubble Tree View:

- DrawDemo
- Add Class...
- App
- ArrowButton
- ButtonHolder
- CenterOfMenubar
- CloseGridMenuItem
- DeleteButton
- DrawingPanel
- GridMenuItem
- HighlightMenuItem
- IconMenuItem
- MenuBar
- Top of file
- Member Variables
- Add Method...
- Method
- const
- createMenu1
- createMenu2
- createMenuItem
- createMenuBar
- createPopupMenu
- createPropertyButtons

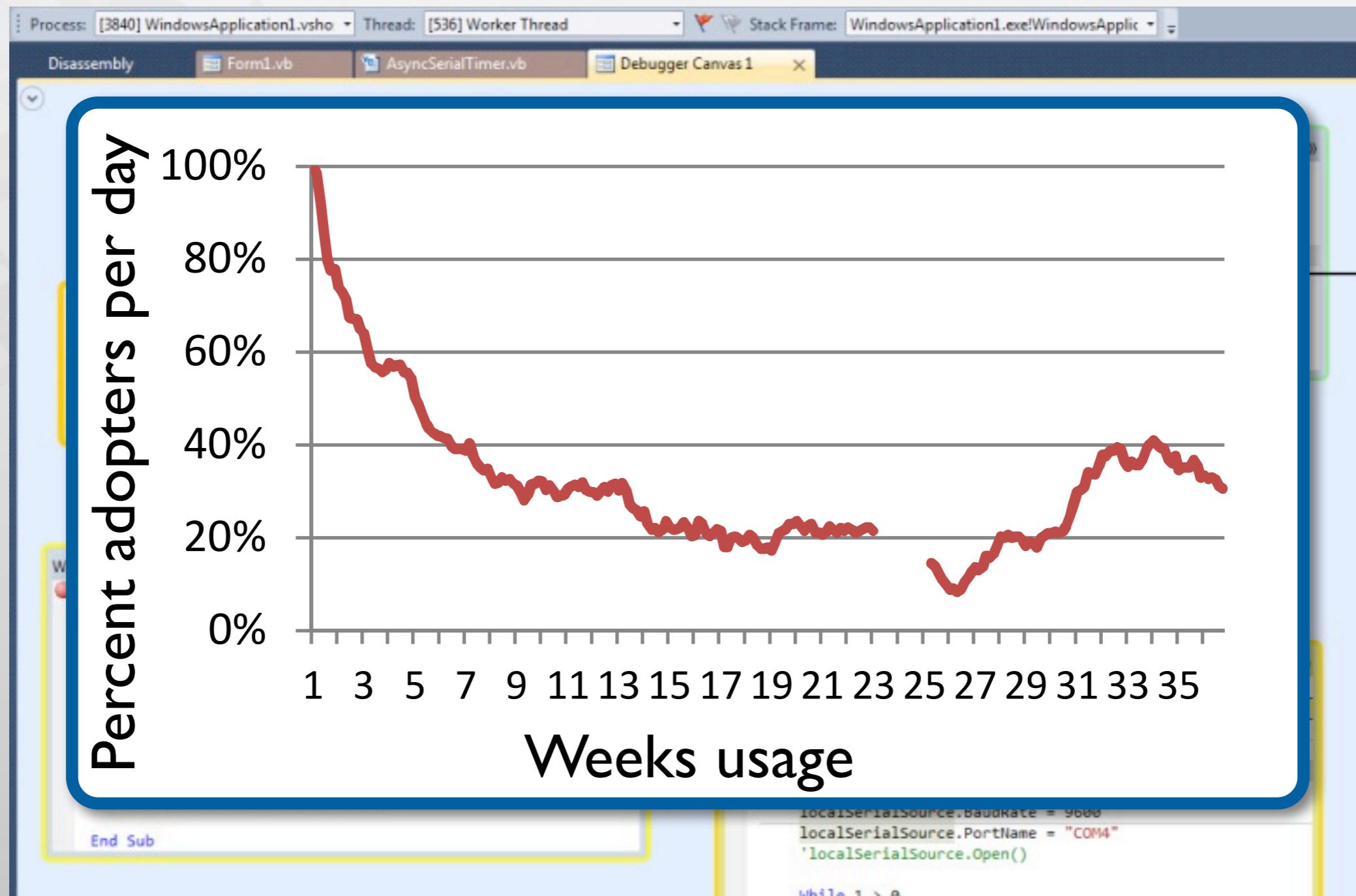
Opened Session:

- Opened
- Untitled
- 2010-10-01 22:35 AM

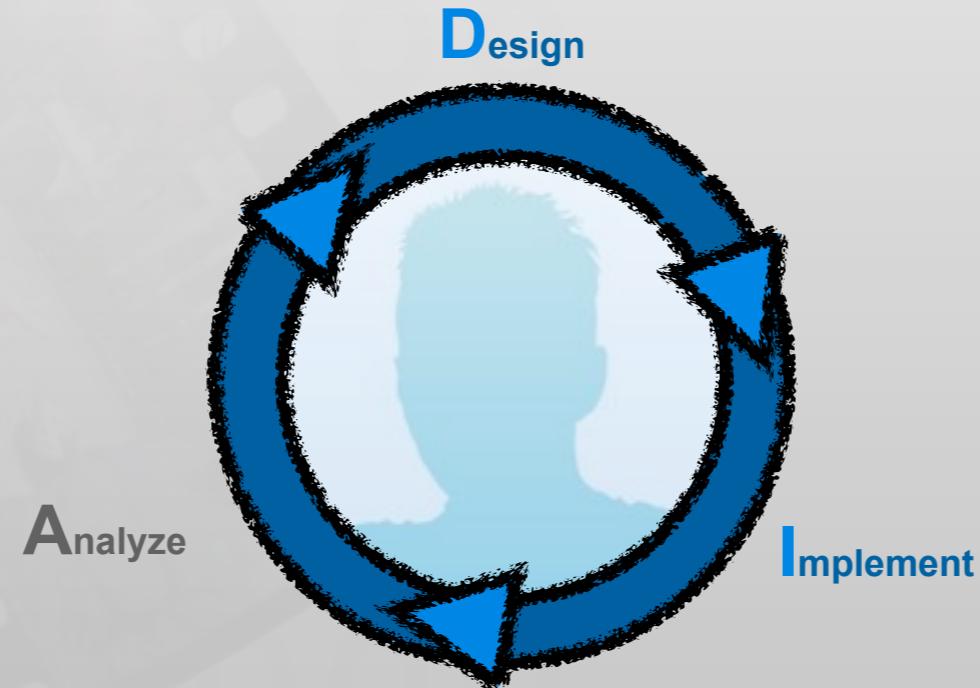
[Bragdon2010, Code bubbles: a working set-based interface for code understanding and maintenance]

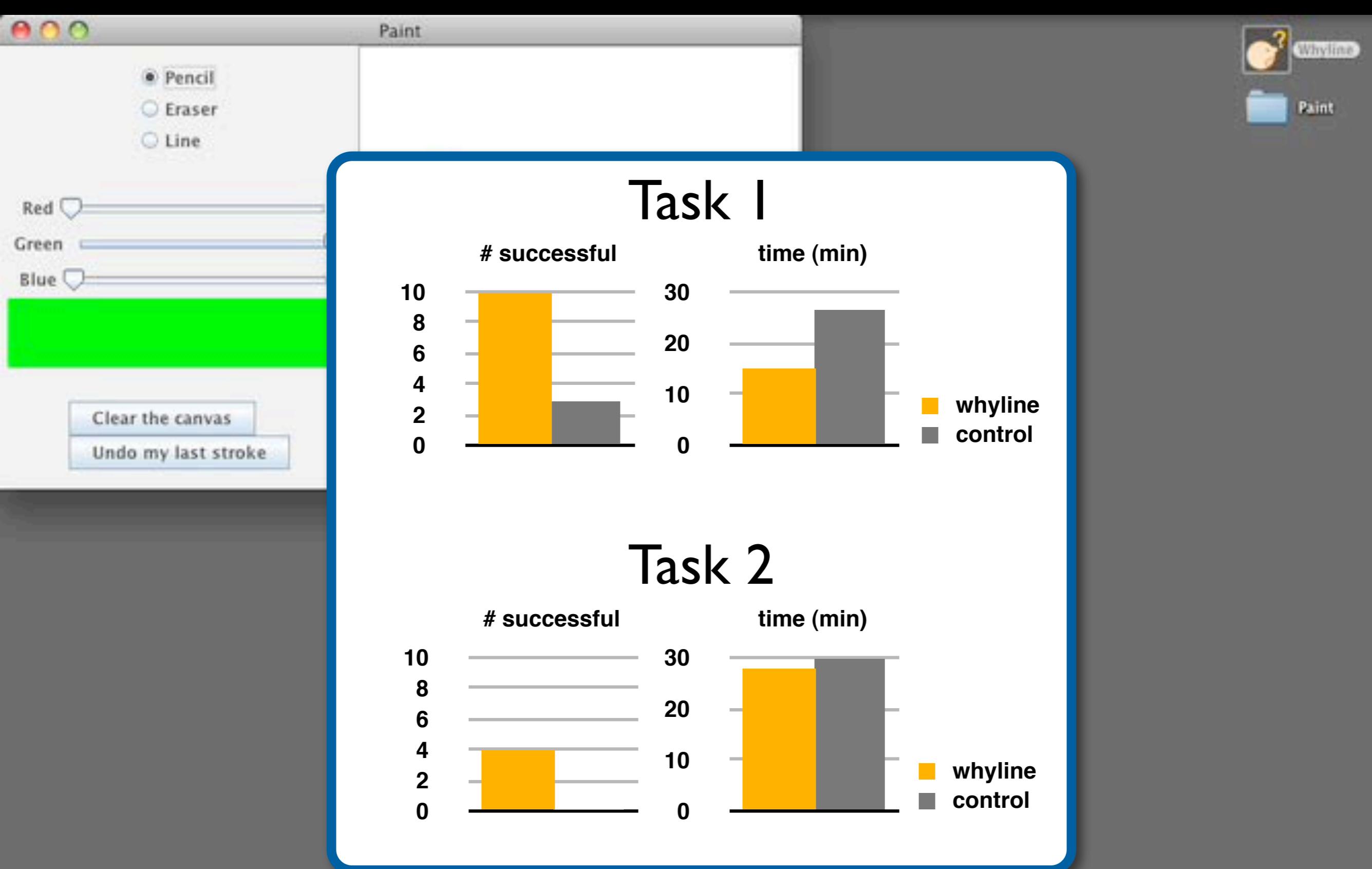
# Canvas Interfaces in the Wild

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



# Utilizing the Call Graph





```
- (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 =
    NSMutableArray *files =
    NSMutableArray *info =
    NSMutableDictionary *
    NSError *error =
    BOOL initial =
    BOOL force = (mask & kBDSKFileMoveForce);
    BOOL check = (initial && (mask & kBDSKCheckCompleteAutoFile));
};

if (numberofPapers == 0)
    return;

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

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

for (id paperInfo in paperInfos) {

    file = [paperInfo valueForKey:BDSKFilerLinkedFileKey];
    pub = [paperInfo valueForKey:BDSKFilerPublicationKey];
    oldPath = [[file URL] path];
    if (initial) // an array of BDSKFileLinkedFiles
        newPath = [BDSKFilerURLForLinkedFile:file];
    else // an array of info dictionaries
        newPath = [BDSKFilerNewPathForInfoDictionary:file forKey:BDSKFilerNewPathKey];

    if (numberofPapers > 1)
        [progressIndicator incrementValue:1.0];
    [progressIndicator setDoubleValue:[progressIndicator doubleValue]/numberofPapers];
}

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

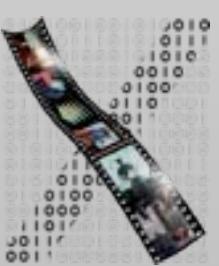
info = [NSMutableDictionary dictionaryWithCapacity:6];
[info setValue:file forKey:BDSKFilerFileKey];
[info setValue:oldPath forKey:BDSKFilerOldPathKey];
[info setValue:pub forKey:BDSKFilerPublicationKey];
error = nil;

if (check && NO == [pub canSetURLForLinkedFile:file]) {

    [info setValue:NSLocalizedString(@"Incomplete information to generate file
name.", @"") forKey:BDSKFilerStatusKey];
    [info setValue:[NSNumber numberWithInt:BDSKIncompleteFieldsErrorMask]
forKey:BDSKFilerFlagKey];
    [info setValue:NSLocalizedString(@"Move anyway.", @"") forKey:BDSKFilerFixKey];
    [info setValue:newPath forKey:BDSKFilerNewPathKey];
}
```

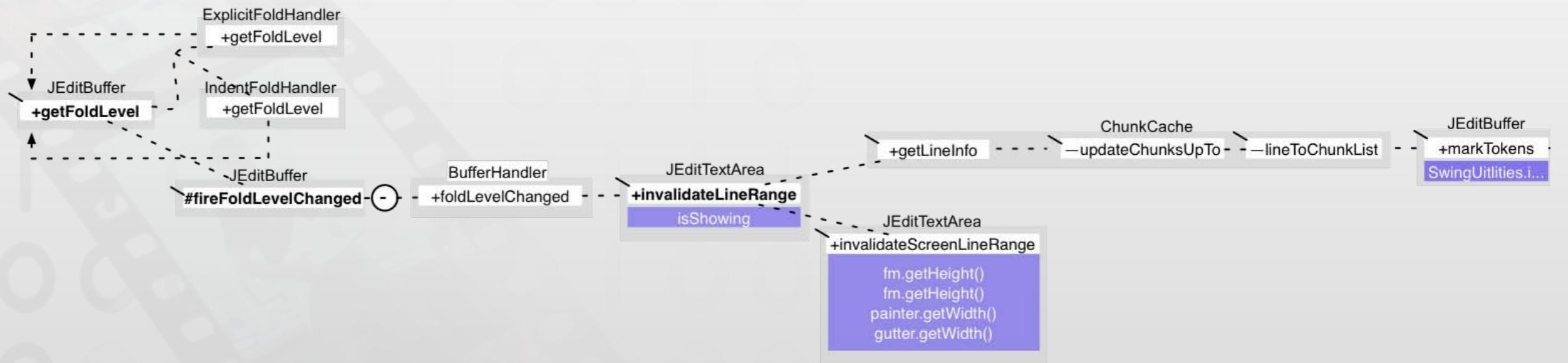
# In practice: Feasible paths most interesting

[LaToza2010, Developers ask reachability questions]



# Utilizing Call Graph Information

[LaToza2010, Searching Across Paths]



## Legend

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



# Static Analysis in the Wild

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

The screenshot shows a Mac OS X window titled "Example.m". The menu bar includes "File", "Edit", "Project", "Build", "Run", "Tools", "Help", and "About". The toolbar has icons for file operations and build status. The main area displays Objective-C code with annotations:

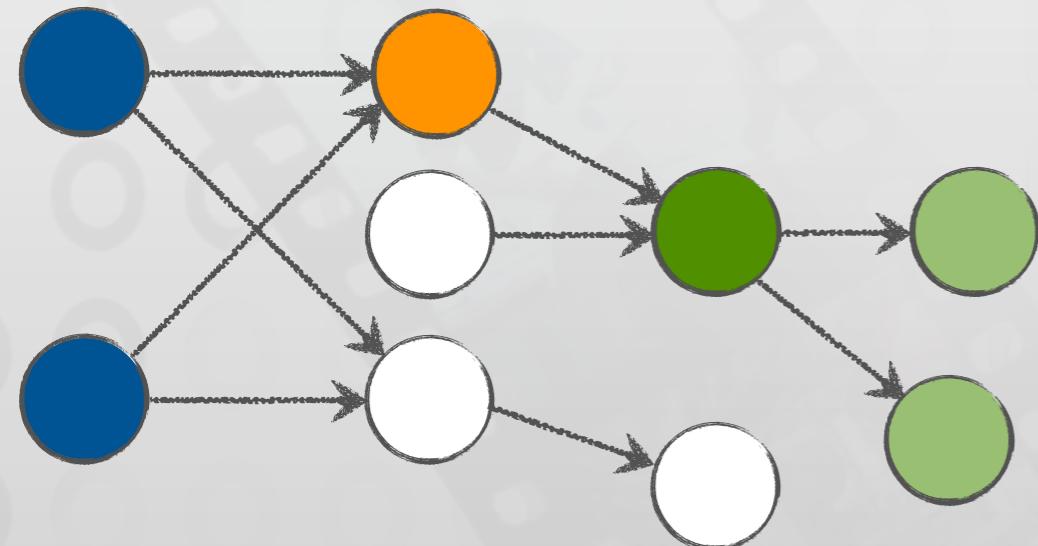
```
10 }
11
12 void foo(int x, int y) {
13     id obj = [[NSString alloc] init];
14     switch (x) {           └ Method returns an Objective-C object with a +1 retain count (owning reference)
15         case 0:
16             [obj release];
17             break;
18         case 1:
19             // [obj autorelease];
20             break;
21         default:
22             break;
23     }
24 }
```

Annotations include:

- A tooltip for the allocation line 13 states: "Method returns an Objective-C object with a +1 retain count (owning reference)".
- A tooltip for the deallocation line 24 states: "Object allocated on line 13 is no longer referenced after this point and has a retain count of +1 (object leaked)".



# Call Hierarchy



The screenshot shows the Xcode IDE with the file `MainController.m` open. The code implements a converter application with methods for conversion logic and UI interaction. A specific section of the code, the implementation of `-performConvert`, is highlighted with an orange box. To the right of the code editor, three overlapping rounded rectangles in blue, light green, and medium green colors represent the call hierarchy for the highlighted method. The blue rectangle is at the top, followed by light green, and then medium green at the bottom.

```
-(void)convertClicked:(id)sender;
{
    //do something
    if (self.theConverter == nil) {
        self.theConverter = [[Converter alloc] init];
    }

    [self performConvert];
}

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

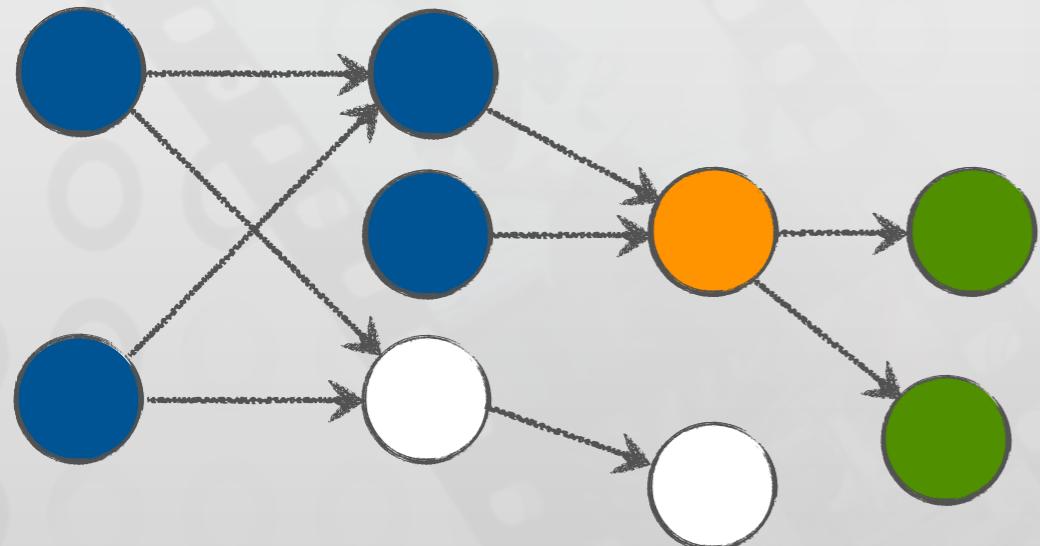
    //update view
    [self update:f];
}

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

@end
```

# Stacksplorer

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



The screenshot shows the Stacksplorer interface integrated with a code editor. The title bar reads "MainController.m - Converter". The code editor displays the following C-like pseudocode:

```
>MainController
  init
  MainController
  convertClicked:

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

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

//convert from Celsius
-(void)convert;
{
    //get celsius value
    float c = [self.input floatValue];
    //convert to fahrenheit
    float f = [self.theConverter c2f:c];
    //update view
    [self update:f];
}

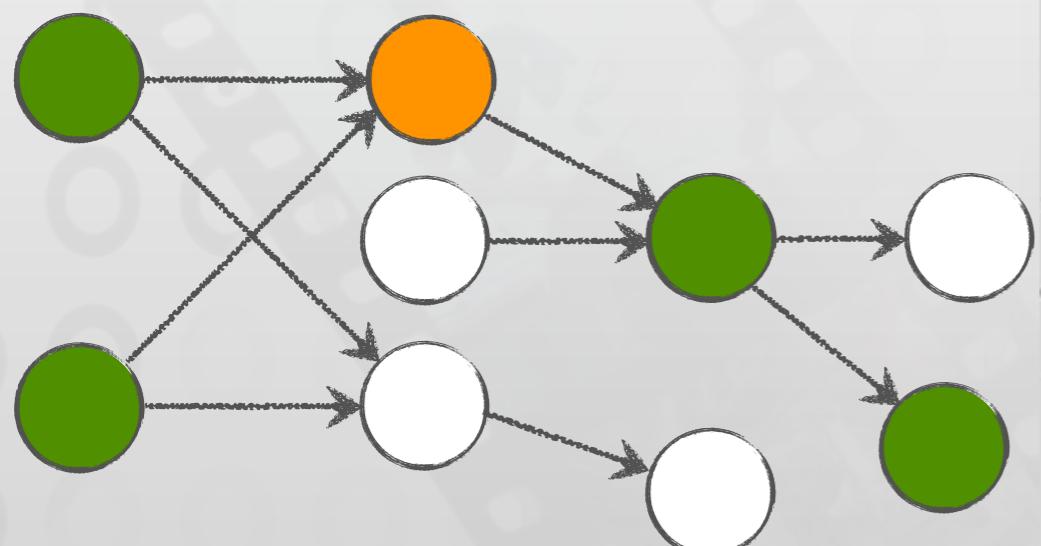
-(void)update:(float)f;
{
    //do something
    if (self.theConverter == nil)
        self.theConverter = [[Converter alloc] init];
    [self performConvert];
}
```

Annotations highlight specific parts of the code:

- A blue box highlights the `init` and `convertClicked:` methods in the sidebar.
- An orange box highlights the `-(void)performConvert;` method body.
- An orange box highlights the `-(void)convert;` method body.
- A green box highlights the `-(void)update:(float)f;` method in the sidebar.

# Blaze

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

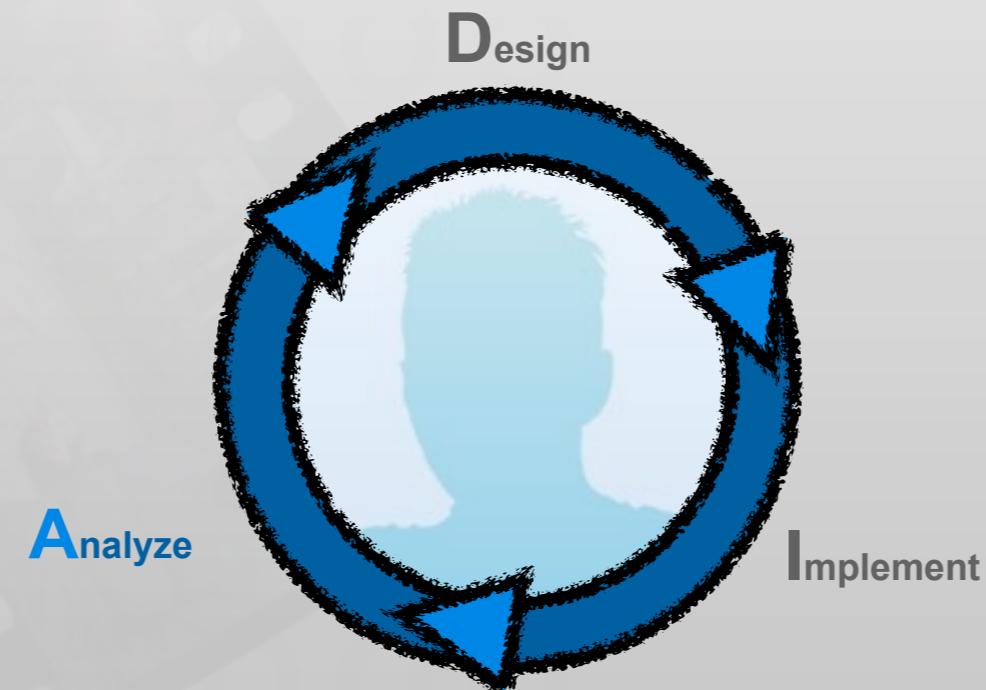


Screenshot of the MainController.m - Converter code editor. The code is as follows:

```
39 } //do something
40 if (self.theConverter == nil) {
41     self.theConverter = [[Converter alloc] init];
42 }
43 [self performConvert];
44
45 -(void)qw{
46     if ([[self.input floatValue] != 0] || ([[self.input stringValue] isEqualToString:@""] || [[self.input stringValue] isEqualToString:@"0"]))
47     {
48         [self convert];
49     }
50 }
51 -(void)menuCallback:(id)sender{
52     [self convert];
53 }
54 //convert from Celsius
55 //to Fahrenheit
56 -(void)convert{
57     //get celsius value
58     float c = [self.input floatValue];
59
60     //convert to fahrenheit
61     float f = [self.theConverter c2f:c];
62
63     //update view
64     [self update:f];
65 }
```

The line `- (void)qw {` is highlighted with an orange rectangle. To the right, a sidebar shows a call graph with four nodes highlighted in green, corresponding to the methods `convertClicked:`, `performConvert`, `convert`, and `c2f:`. The sidebar also includes icons for navigating between these nodes.

# Analyzing Navigation Behavior



Xcode

Call  
Hierarchy

Stacksplorer

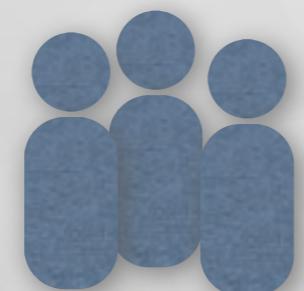
Blaze

Find Change  
Location

Side Effects  
of Change

Task Success

Task Completion Time



33 Developers



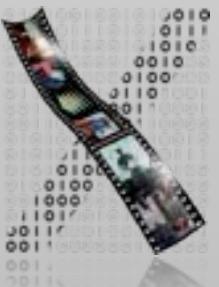
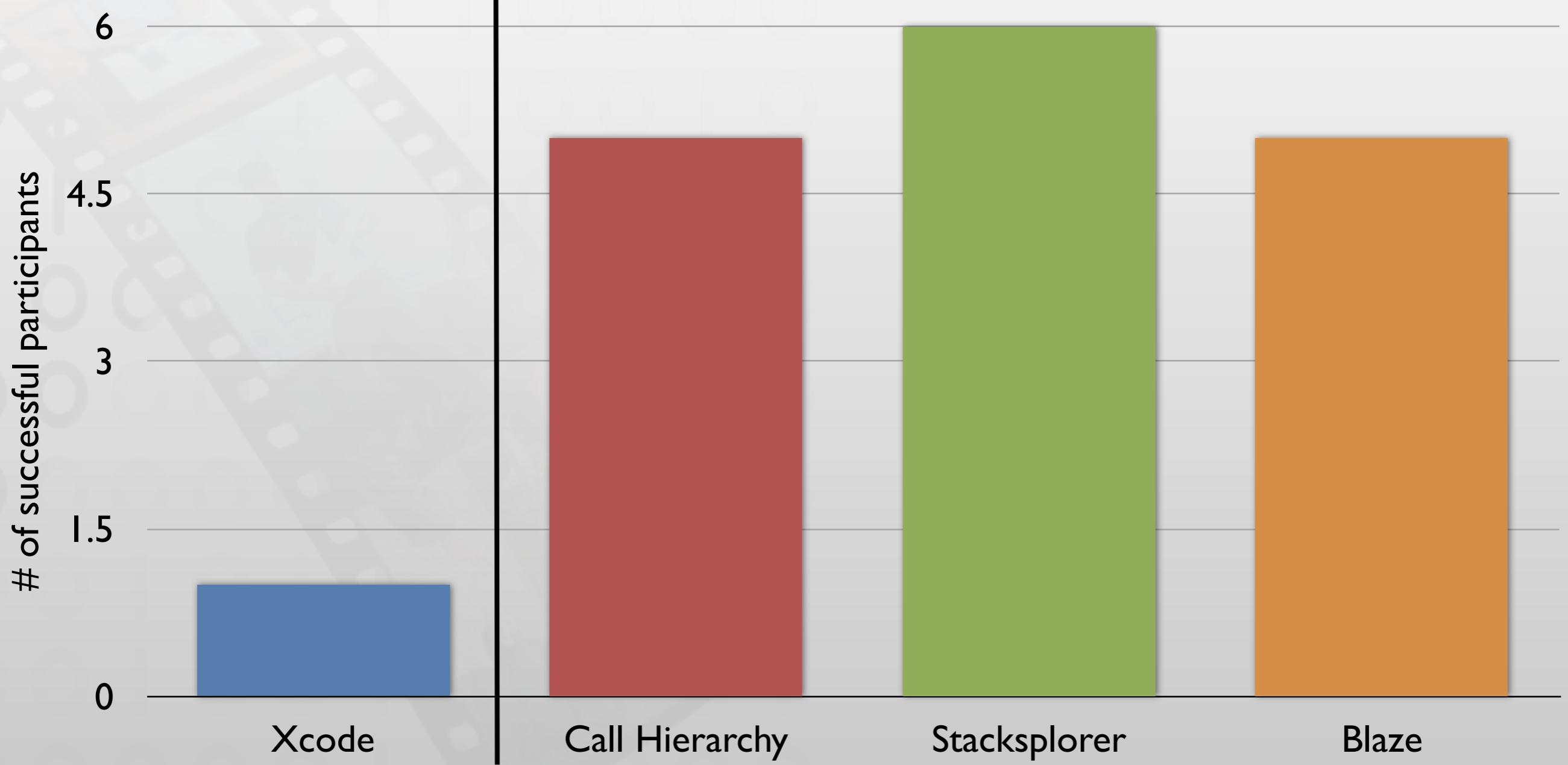
80.000 Lines of Code

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



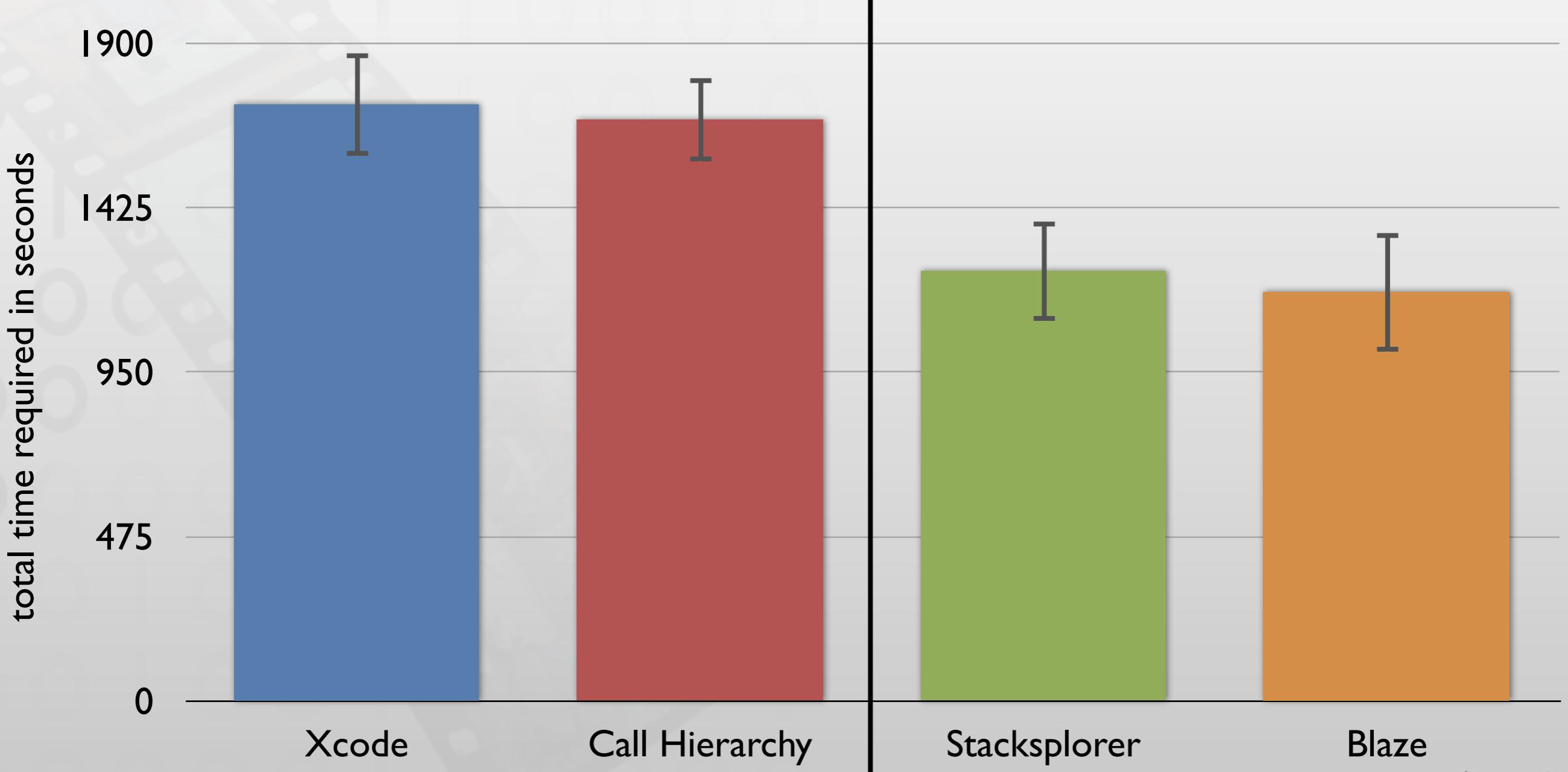
# Task Success

$p = 0.015$



# Task Completion Time

p=0.022



**Effectiveness**

Xcode

Call  
Hierarchy

Stacksplorer

Blaze

**Efficiency**

Xcode

Call  
Hierarchy

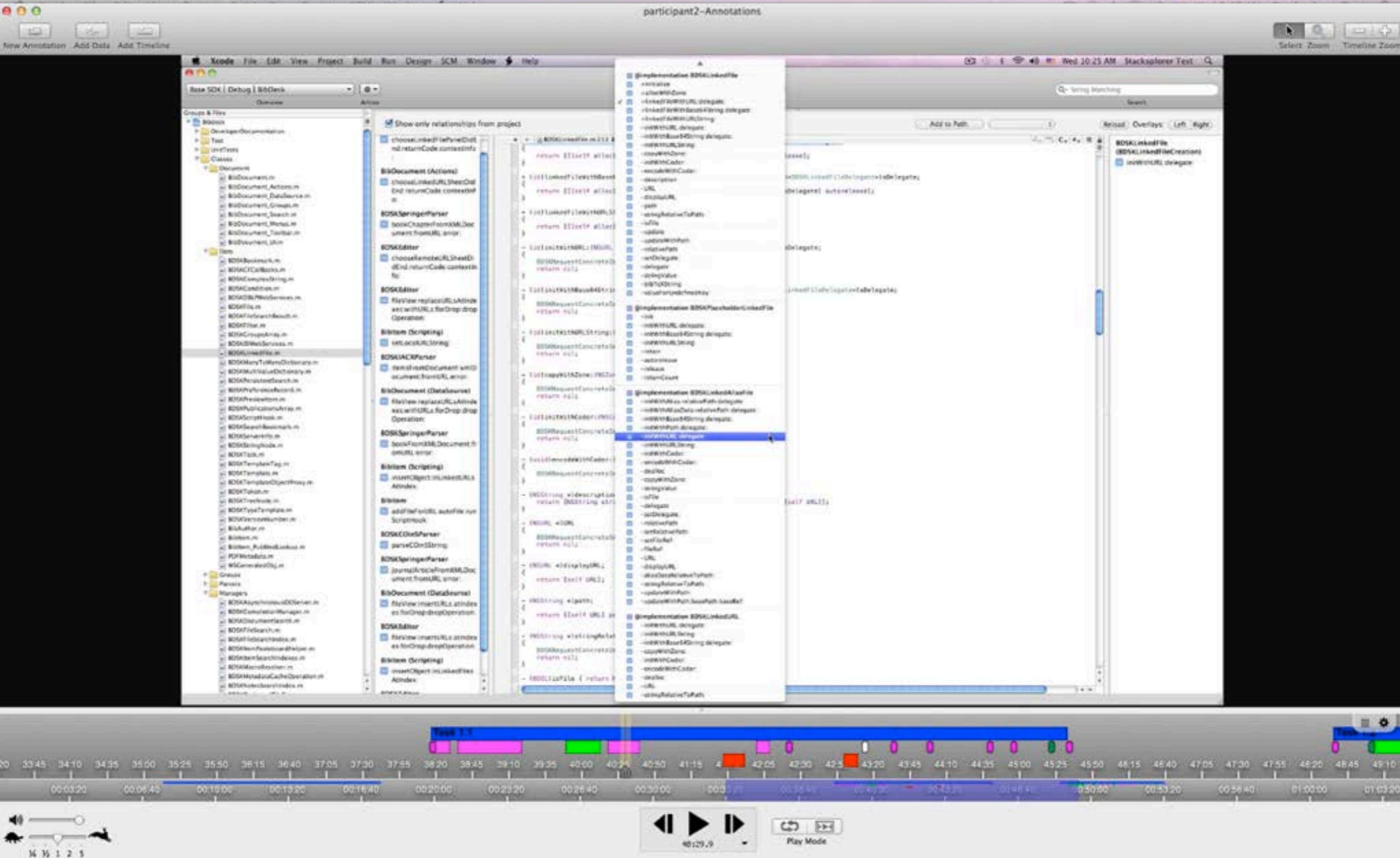
Stacksplorer

Blaze

**Why?**

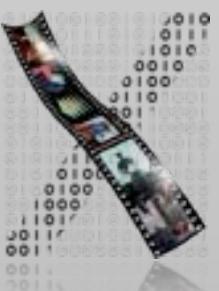
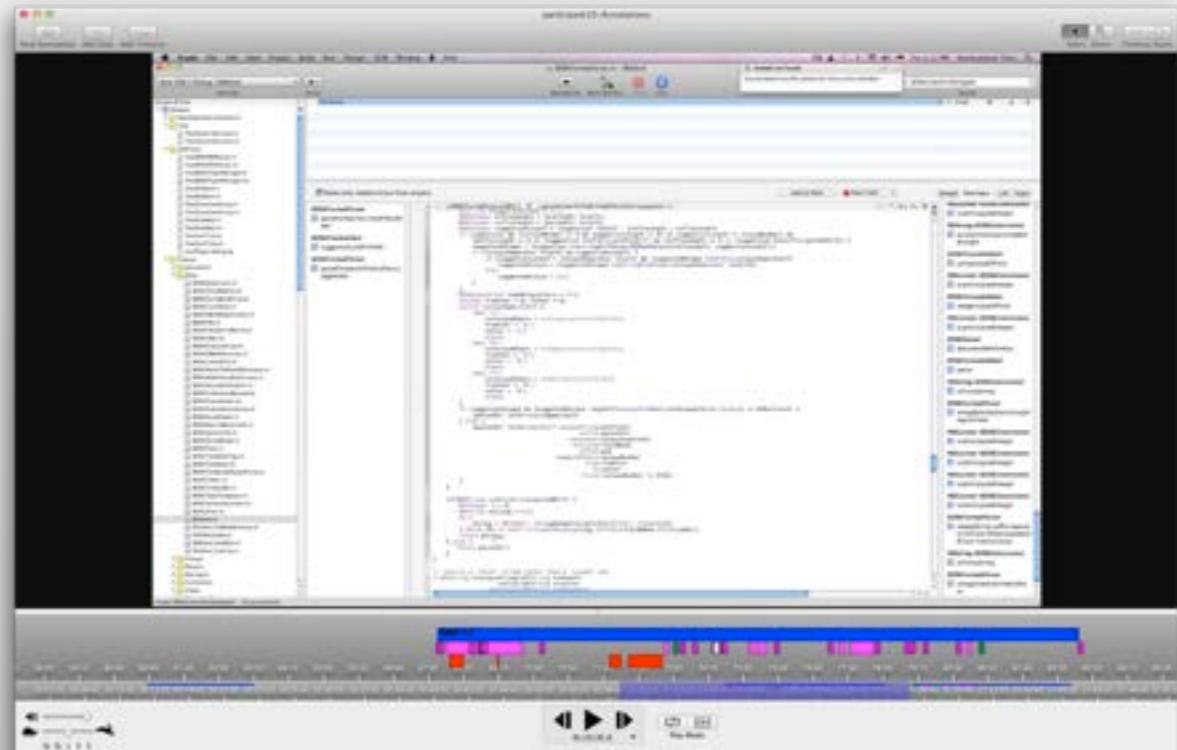
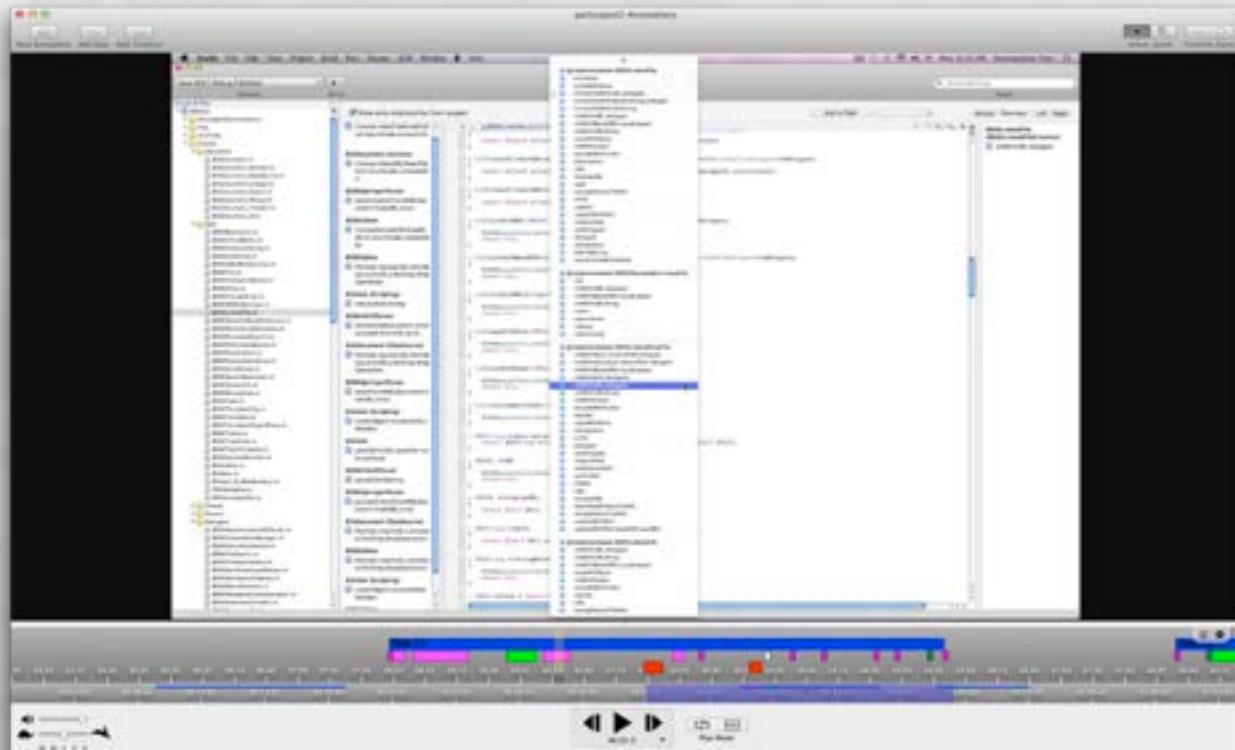
UI Differences

Navigation Behavior



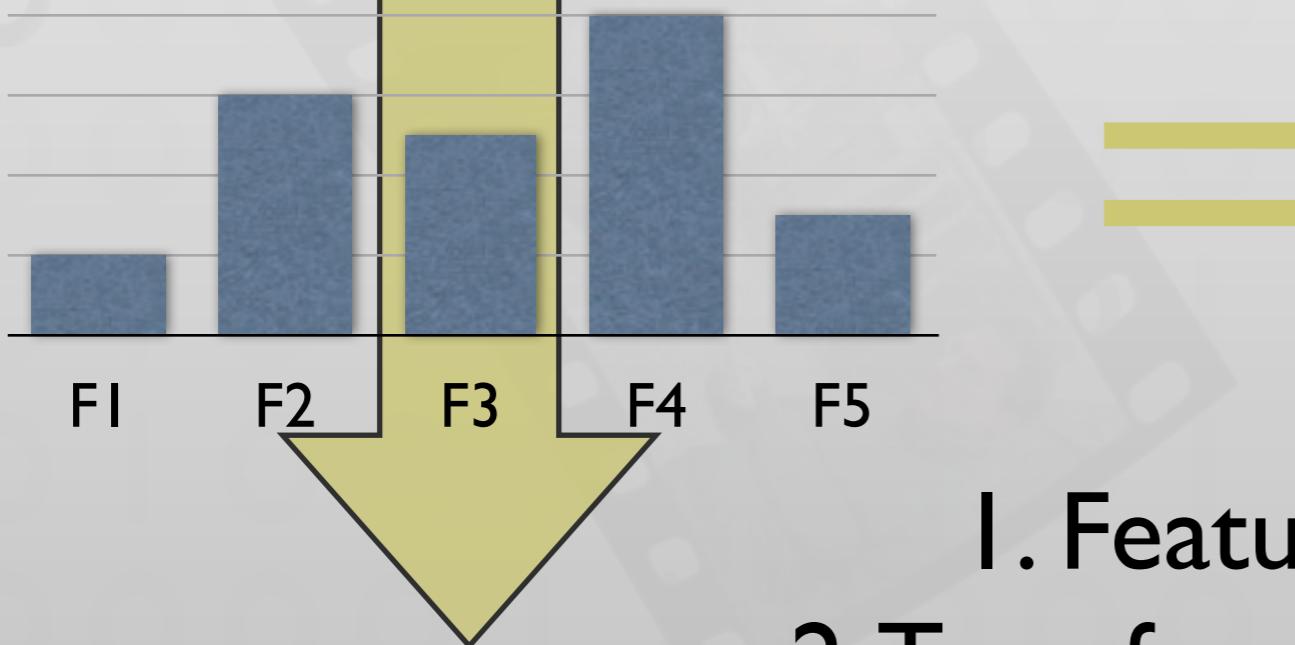
[Fouse2011, ChronoViz: A system for supporting navigation of time-coded data]

# Comparing Navigation Behavior

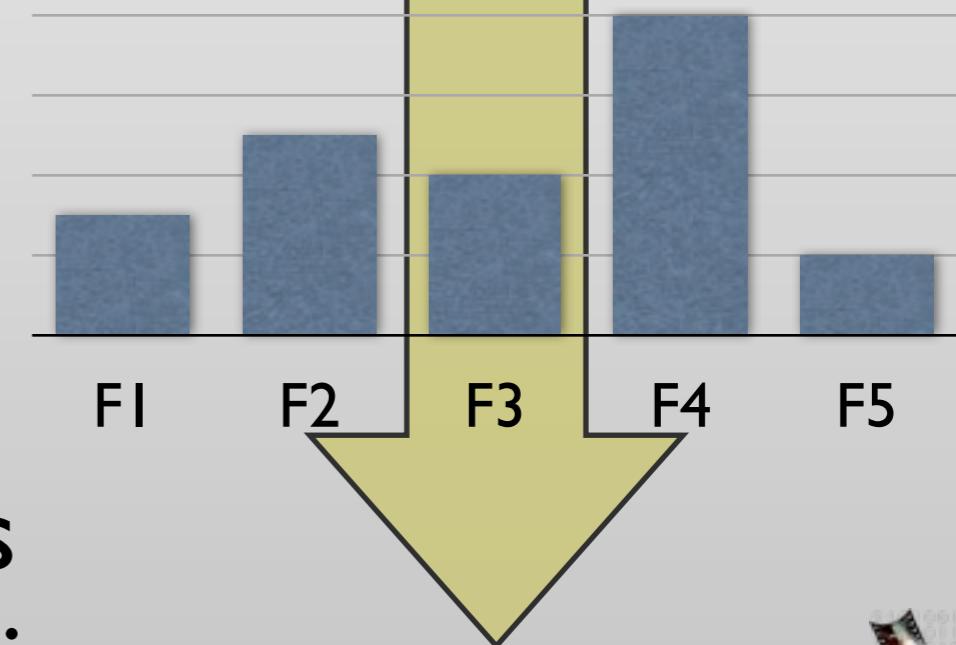




$$I_1 = (p_{1,1}, \dots, p_{640,480})$$



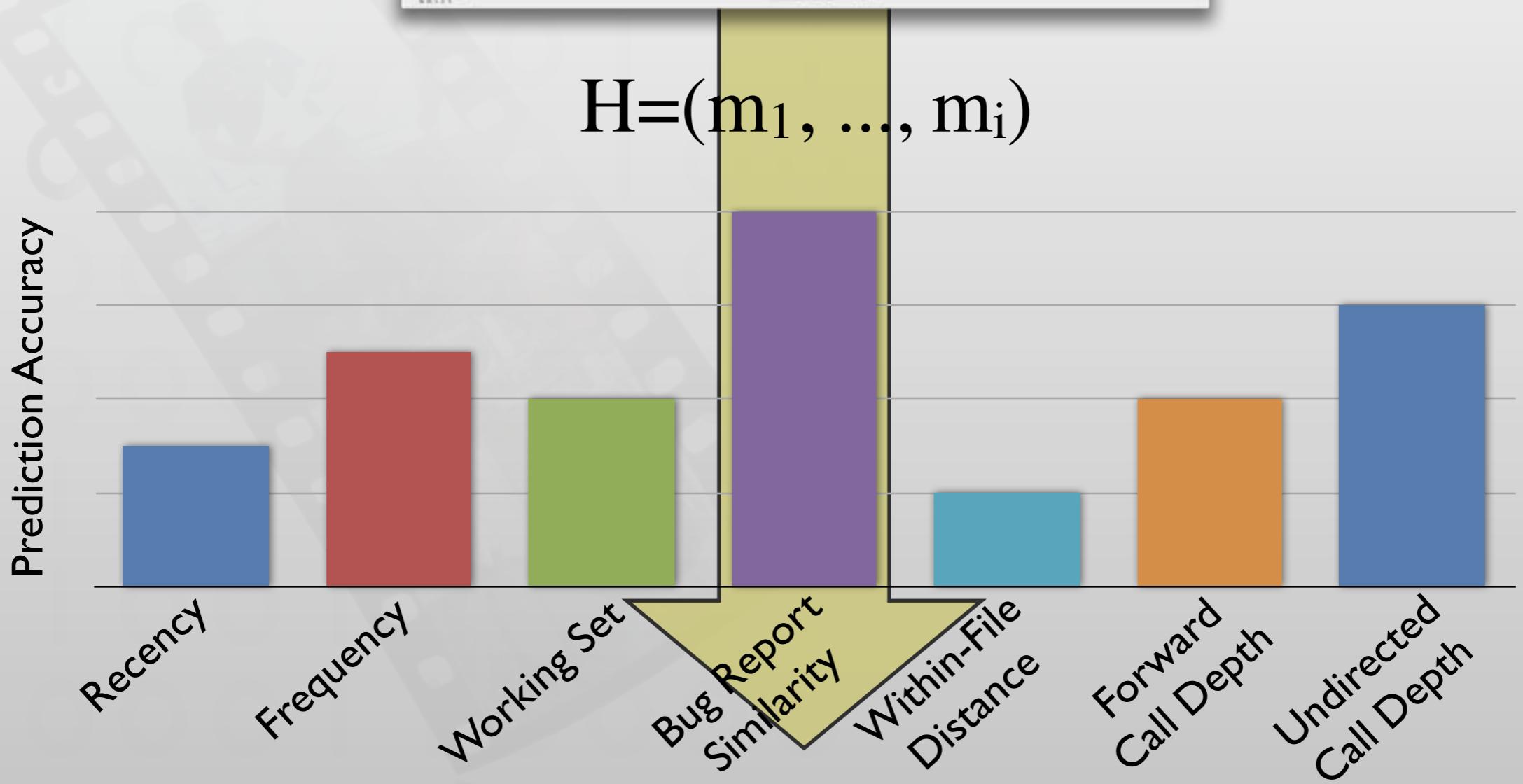
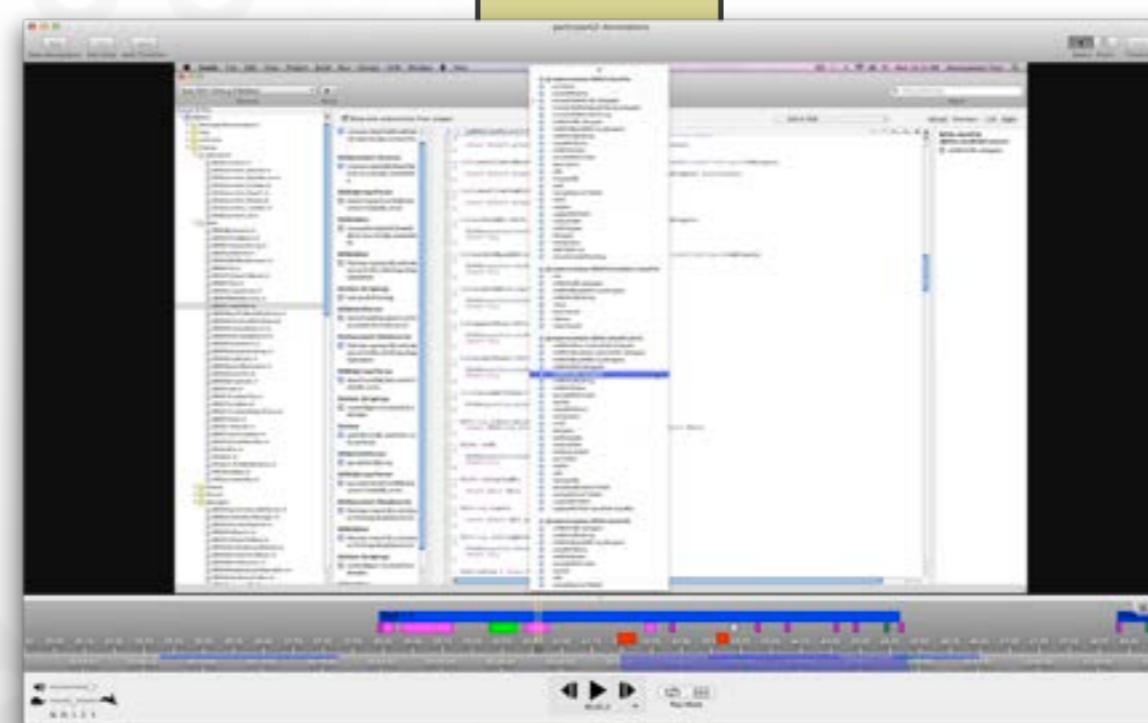
$$I_2 = (p_{1,1}, \dots, p_{1024,768})$$



## I. Features 2. Transformations



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



# 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  $M_i$

$$\begin{aligned} A_4(a) &= 3 \\ A_4(b) &= 2 \end{aligned}$$

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

Rank-transformed  
version of  $A_i$

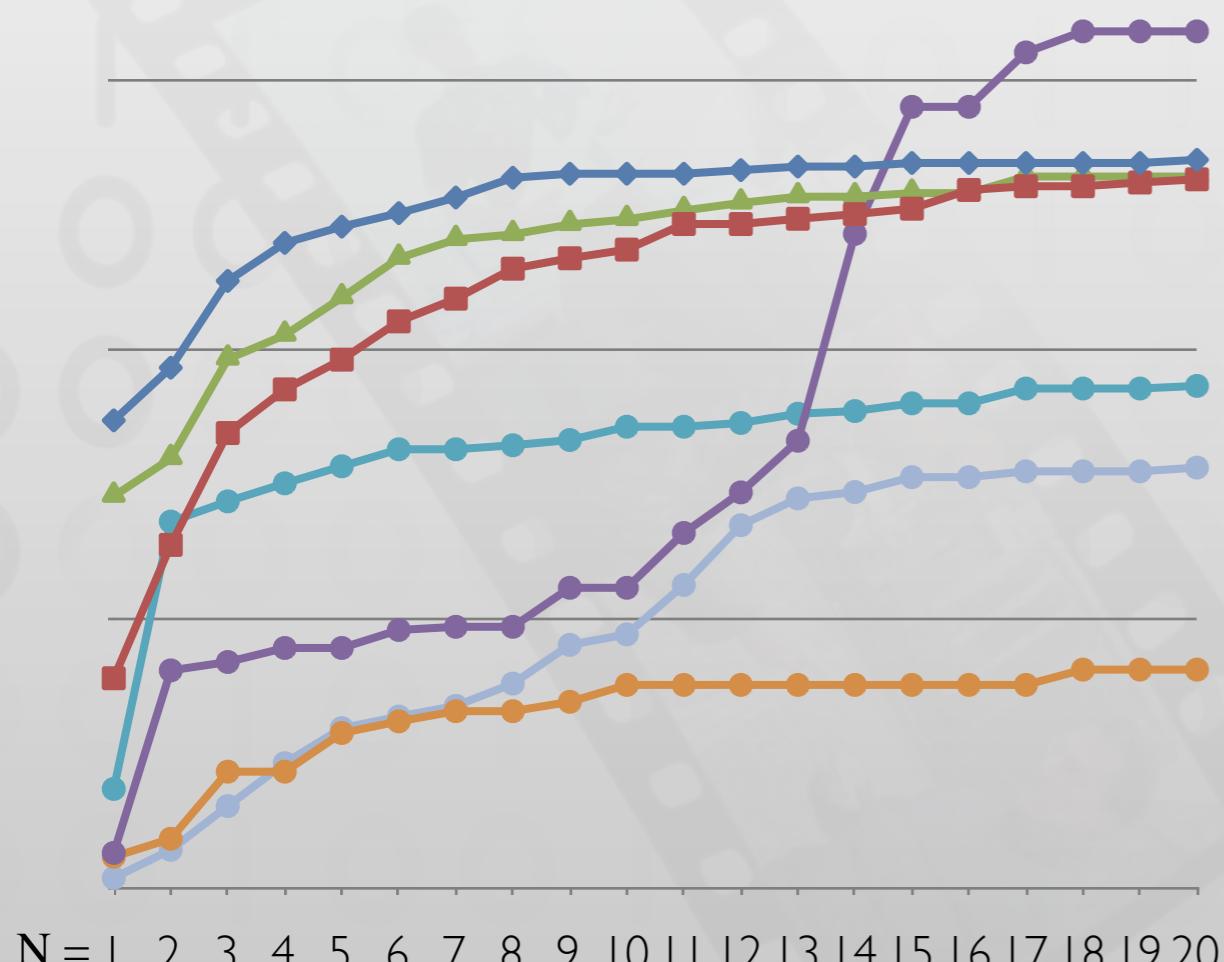
$$\begin{aligned} R_4(a) &= 1 \\ R_4(b) &= 2 \end{aligned}$$

Result: N top-ranked methods

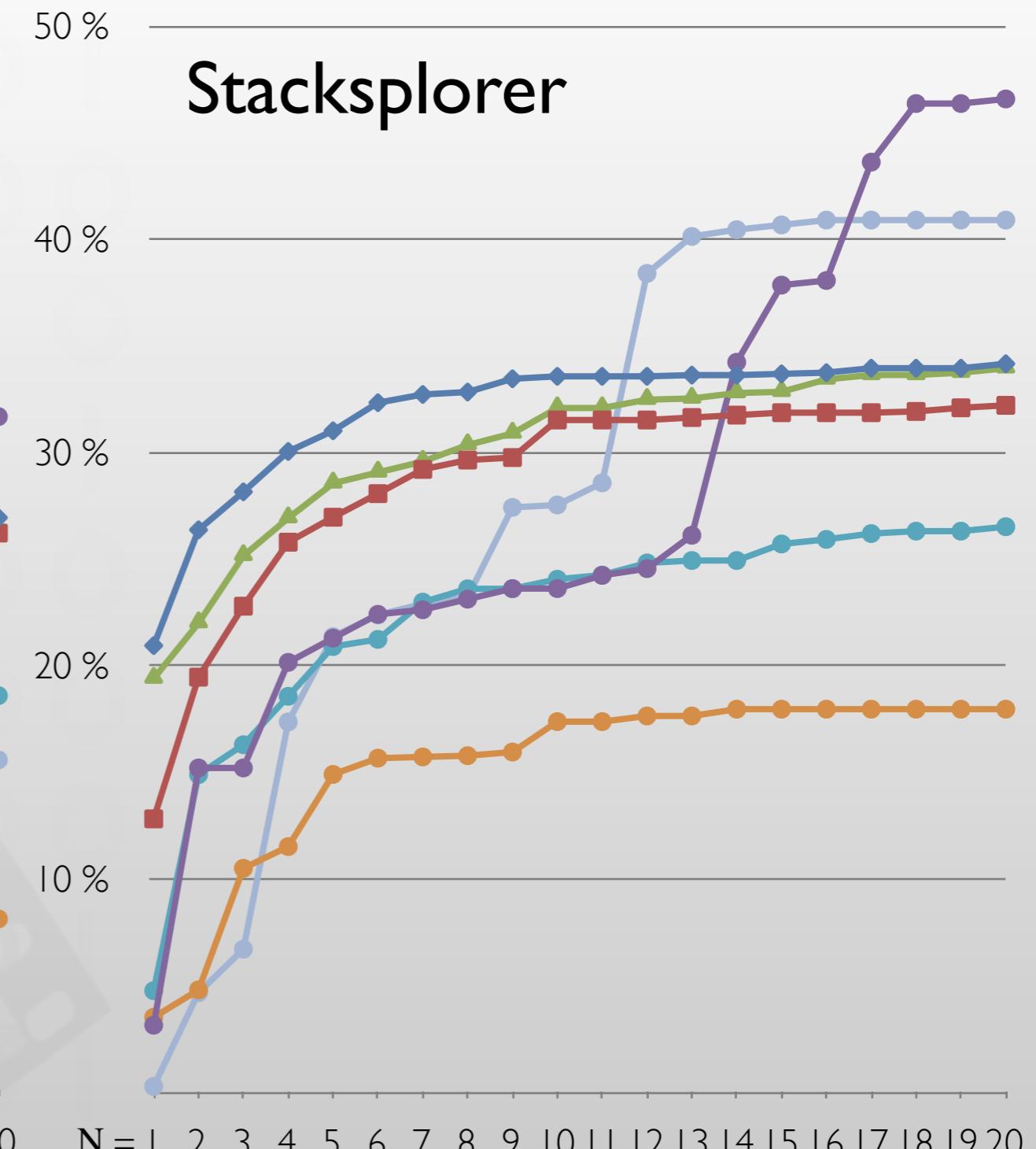


# Prediction Accuracy

Xcode

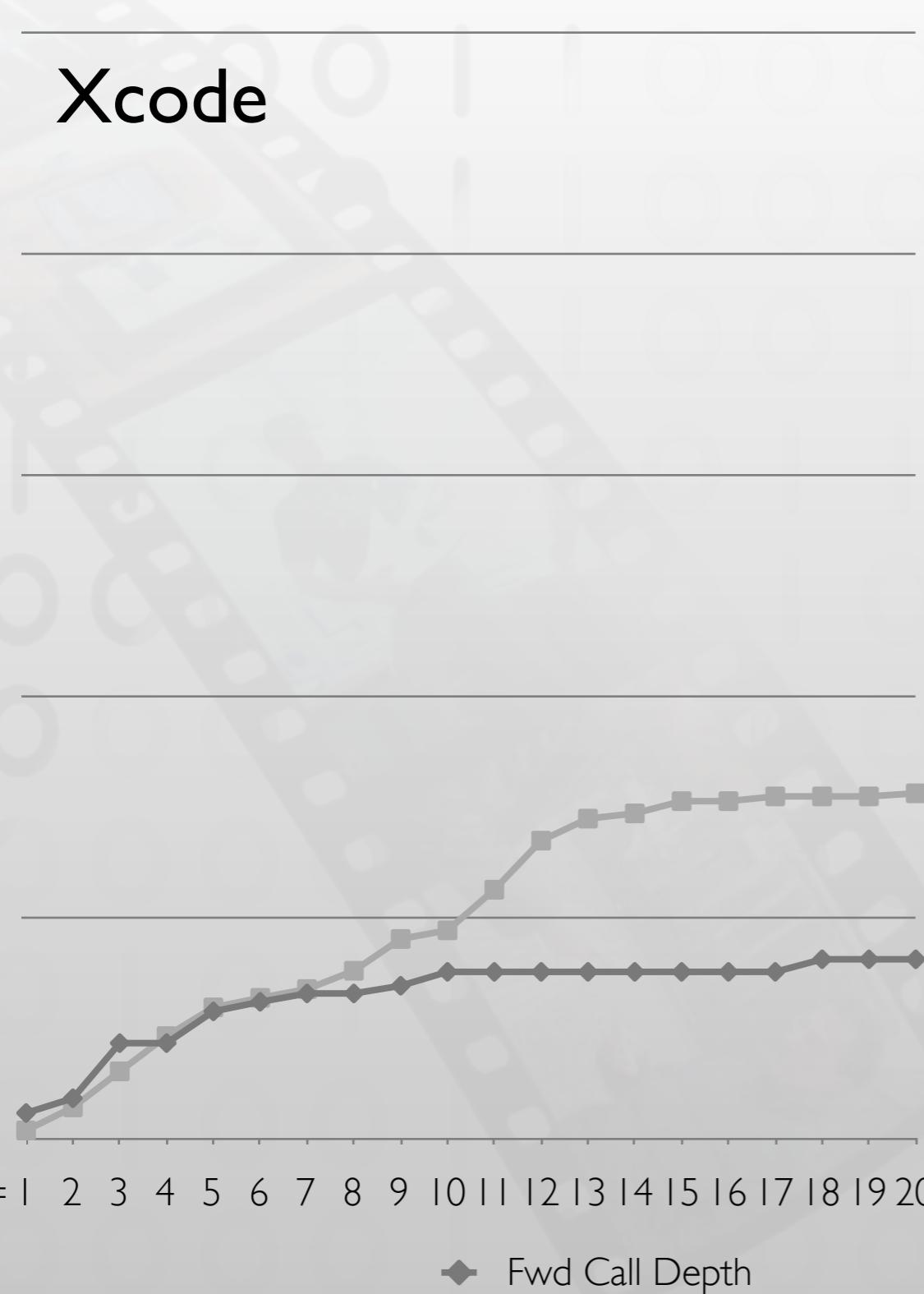


Stacksplorer



# Prediction Accuracy

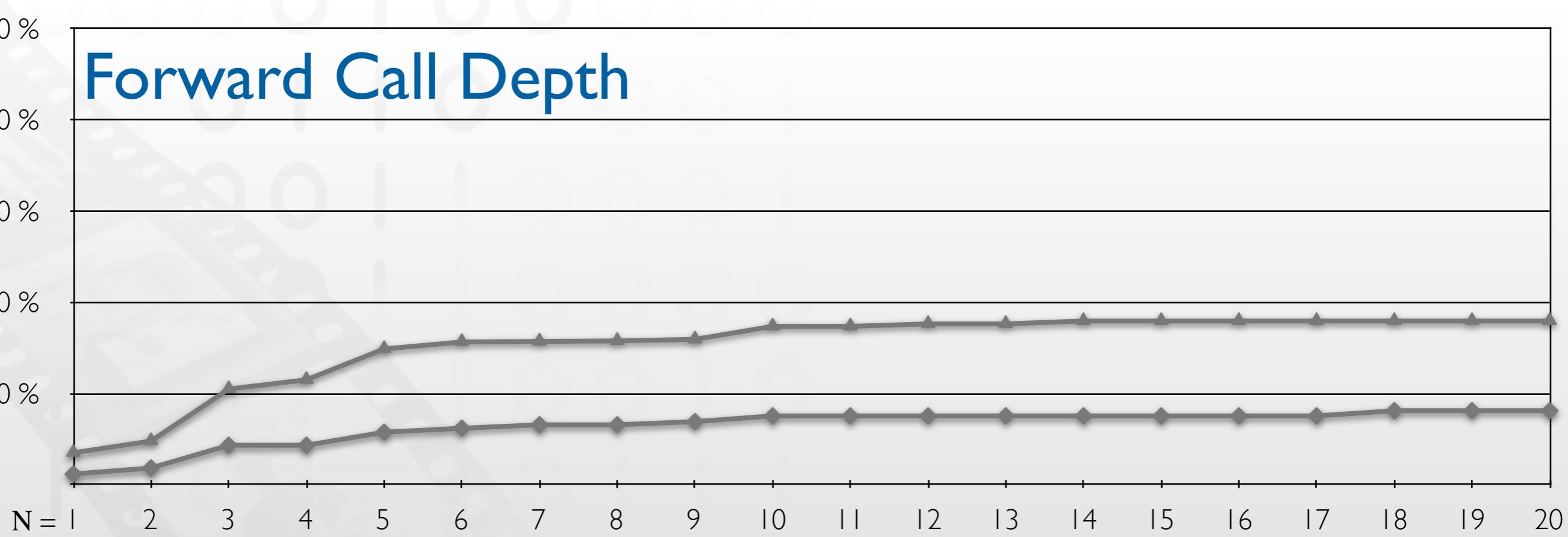
Xcode



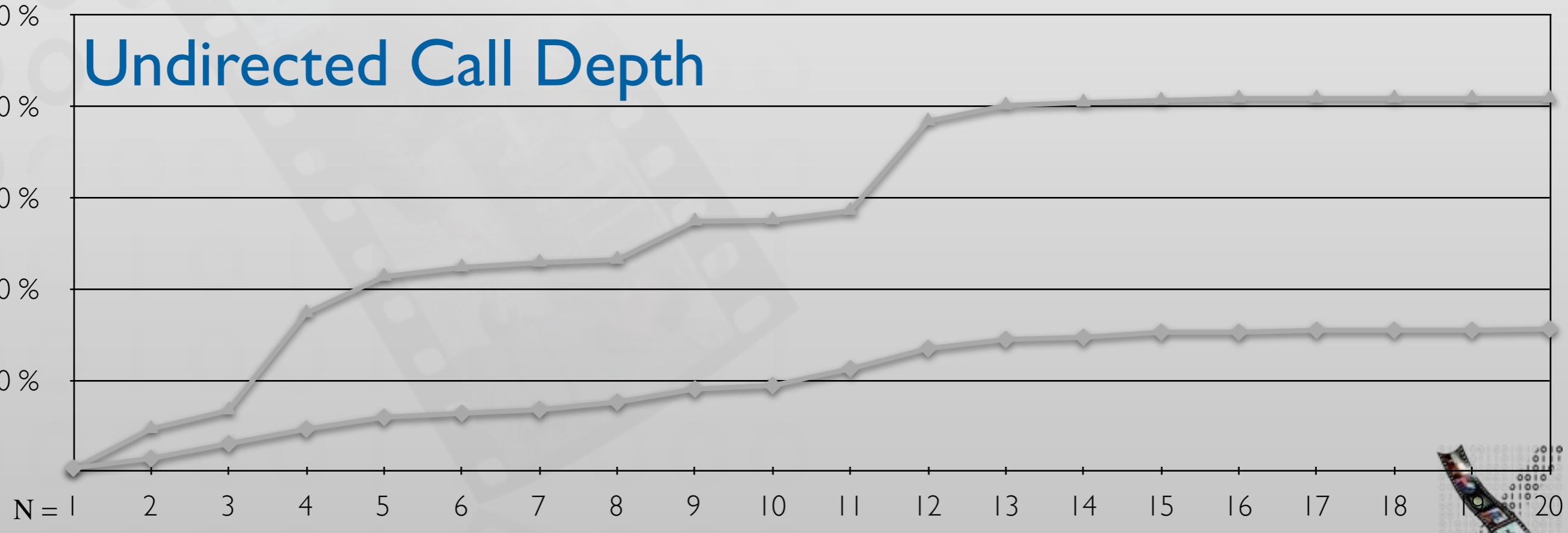
Stacksplorer



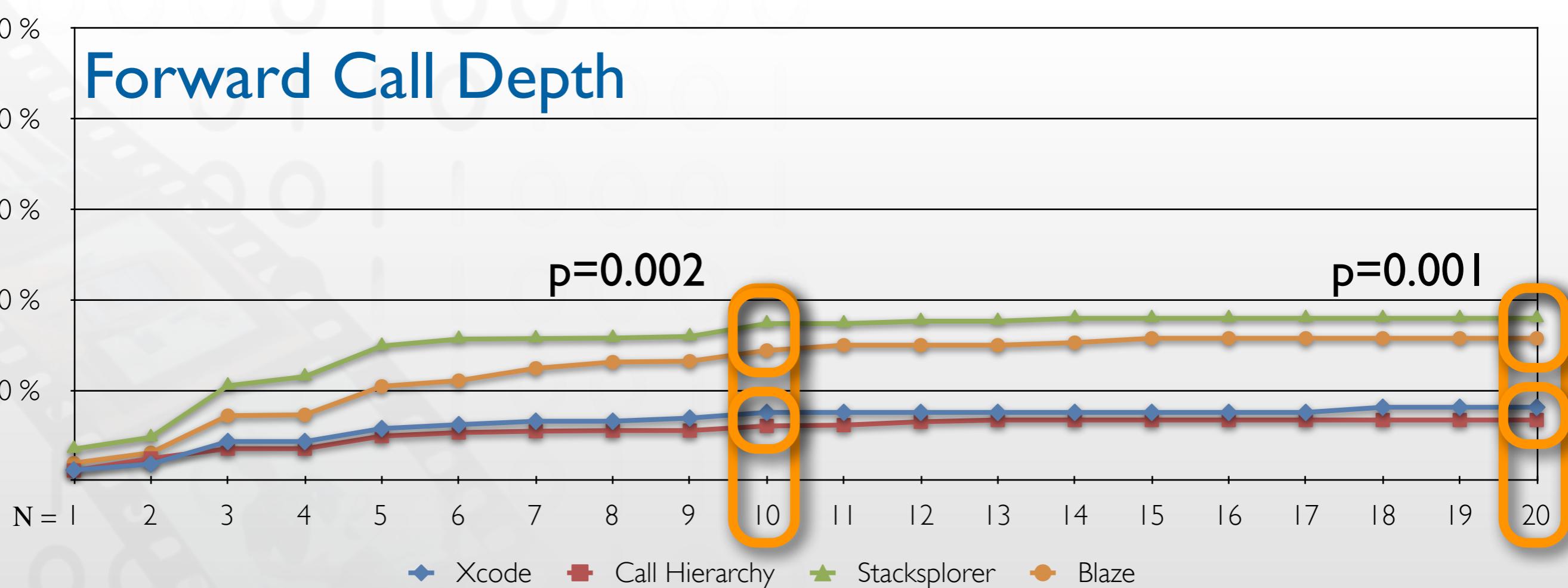
# Forward Call Depth



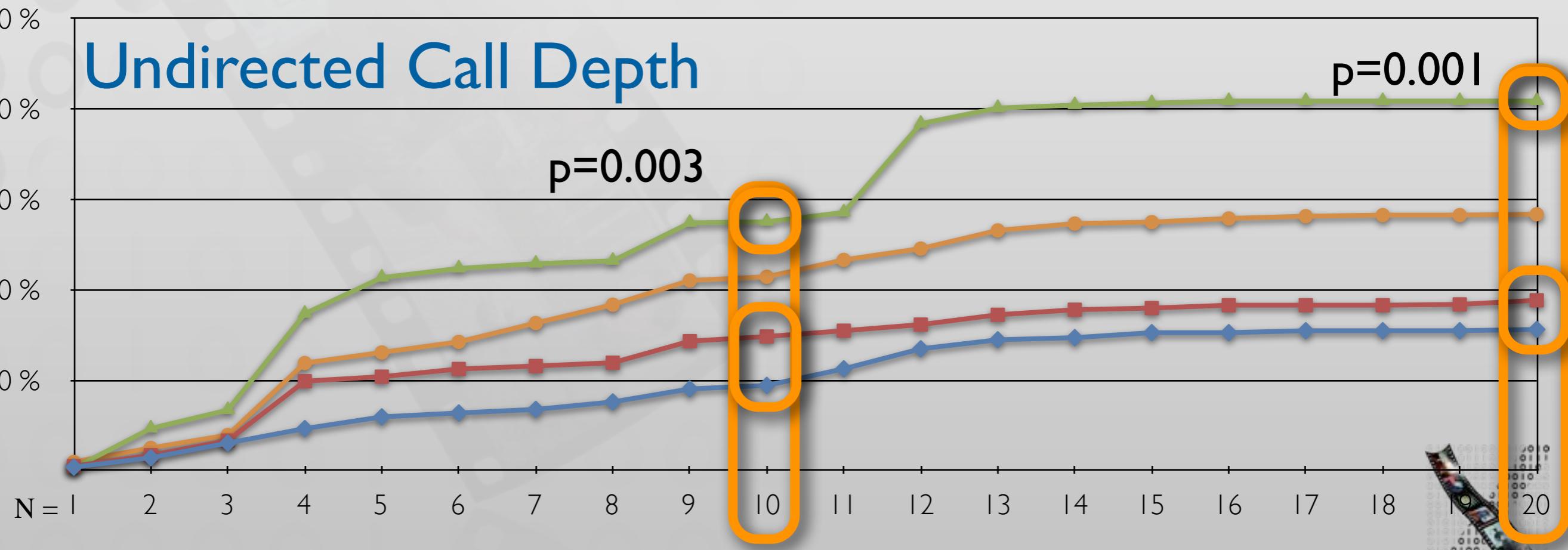
# Undirected Call Depth



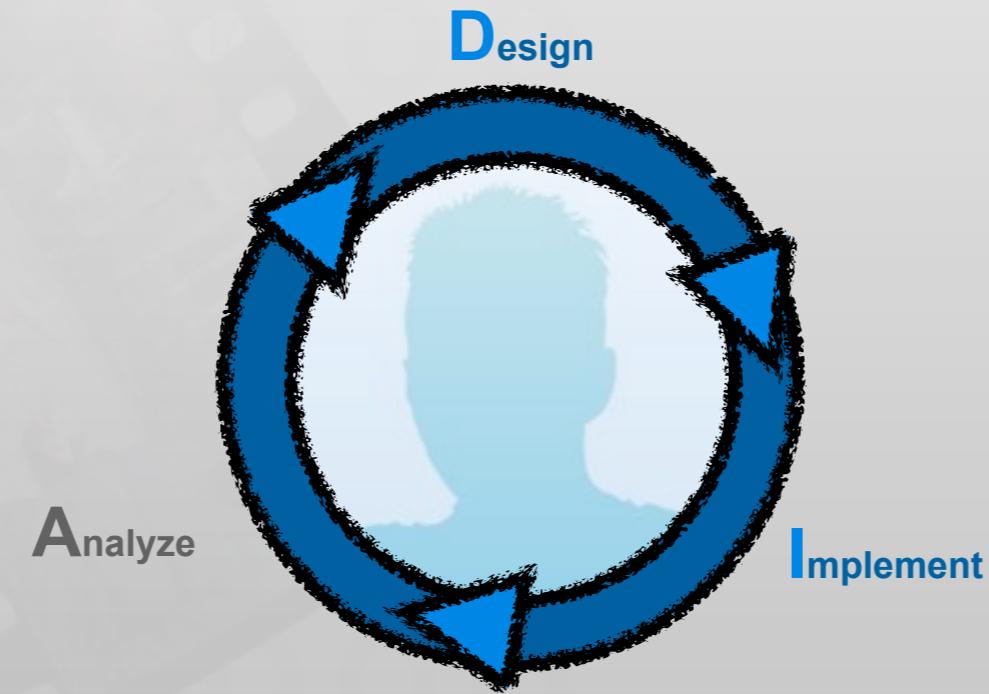
# Forward Call Depth



# Undirected Call Depth



# Outlook



Flex Builder File Edit Source Navigate Search Project Data Run Window Help

Flex Development - bpVideo/src/bpVideo.mxml - Adobe Flex Builder 3

bpVideo.mxml

Source Design

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" creationComplete="loadData()">
3
4 <mx:Script>
5   <![CDATA[
6
7     public function loadData():void {
8       URLLoader
9       }
10    ]]>
11 </mx:Script>
12 </mx:Application>
13
```

Problems Console

```
<terminated> bpVideo [Flex Application] file:/Users/mirad/work/projects/Blueprint/code/bin-debug/bpVideo.html
[SNF] Users:mirad:work:projects:Blueprint:code:bin-debug:bpVideo.swf - 633,020 bytes after decompression
undefined
```

*// 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(i, i, 60, 12, 3) + random(0.7, 1.3);
    if (i == 0) { length = 97; }

    ctx.translate(x,y);
    ctx.rotate(angle);
    ctx.fillStyle = "#000";
    ctx.fillRect(0, -width/2, length, width);

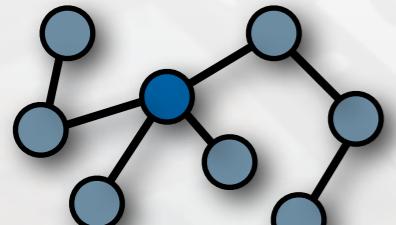
    ctx.restore();

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

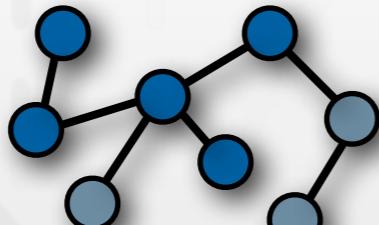
    if (i > 4) {
        blossomPoints.push([x,y,tipX,tipY]);
    }

    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, -0.15) * Math.PI,
        drawBranches(i + 1, angle + random( 0.25, 0.15) * Math.PI);
    }
}
```

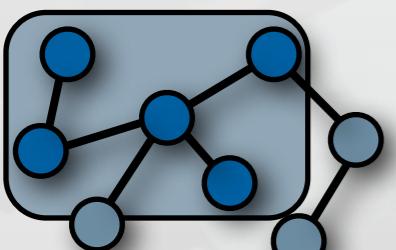
# Summary



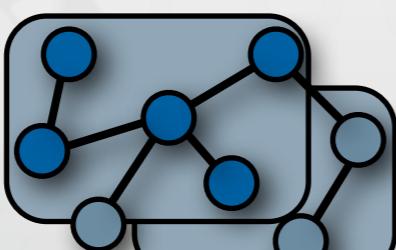
Finding focus points



Expanding focus points



Understanding a subgraph



Questions over groups of subgraphs

