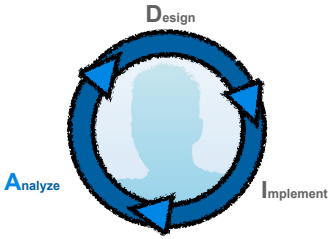
Research in Coding and IDEs

Jan-Peter Krämer
Media Computing Group
RWTH Aachen University

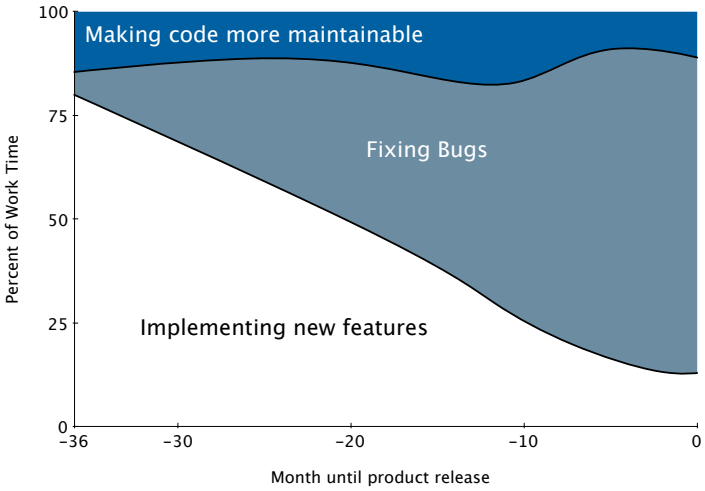
http://hci.rwth-aachen.de/cthci

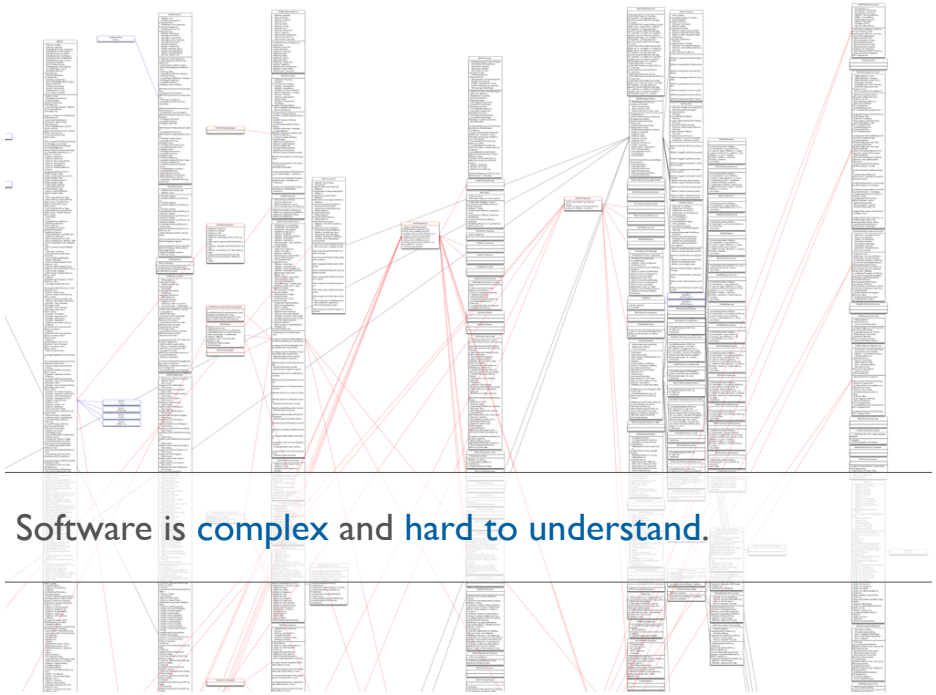
Status Quo



Time in Software Development

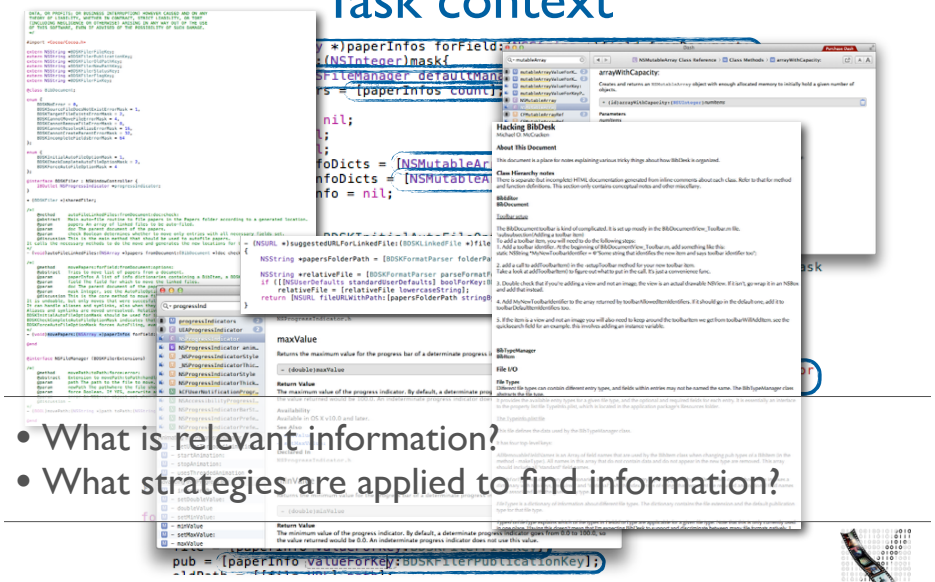
[LaToza2006, Maintaining mental models: a study of developer work habits]





Software is complex and hard to understand.

Task context



- 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]

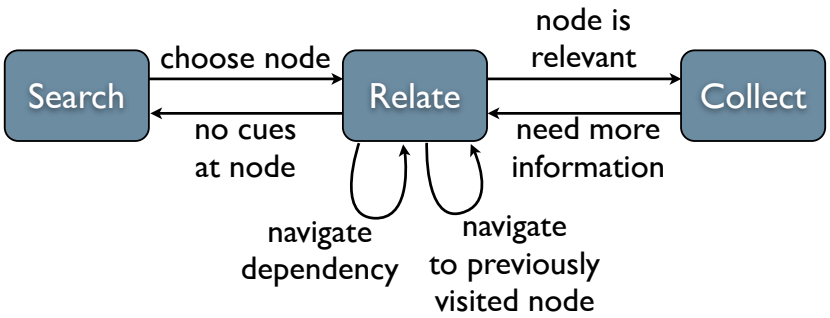
31 Professional Java Developers

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

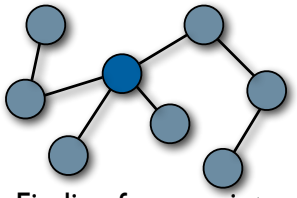
Real world change task

ArgoUML 60k SLOC

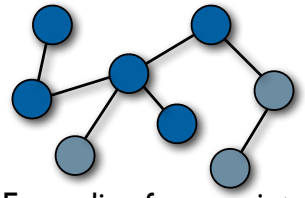
Real world sour code

Models for Developer Strategies

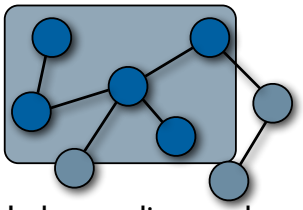
[Sillito2008, Asking and Answering Questions during a Programming Change Task]



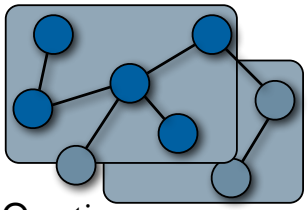
Finding focus points



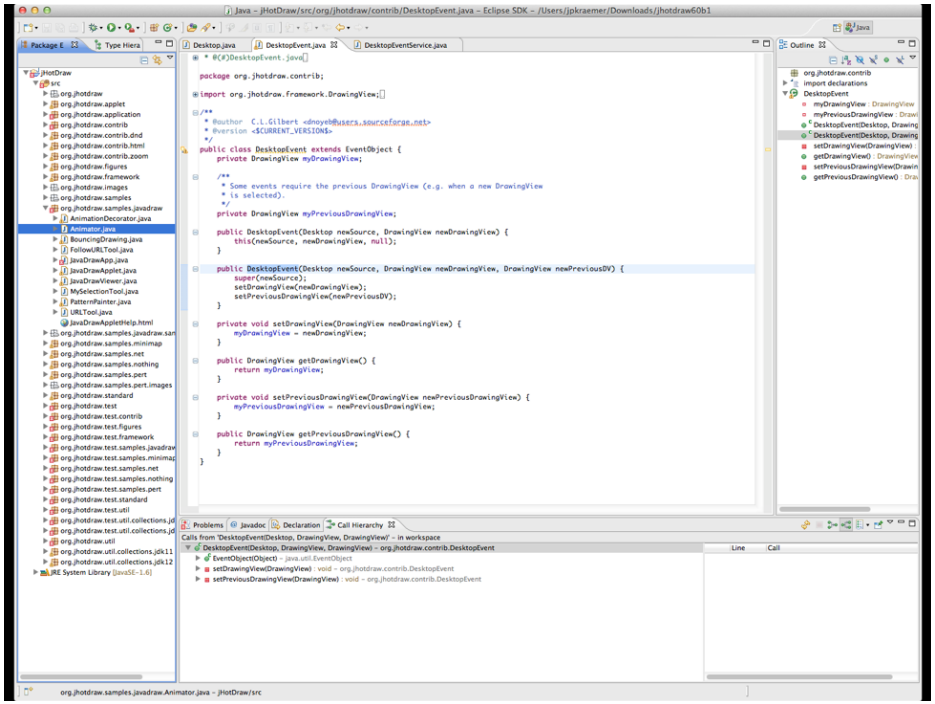
Expanding focus points



Understanding a subgraph

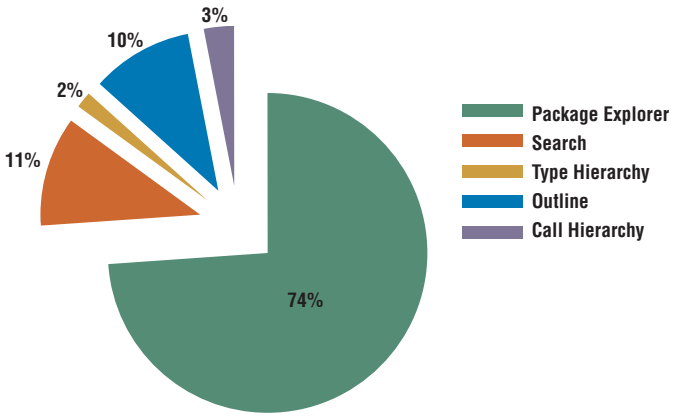


Questions over groups of subgraphs

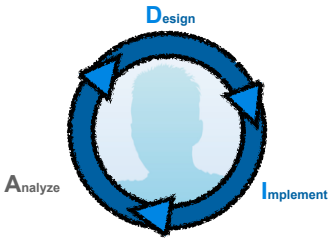


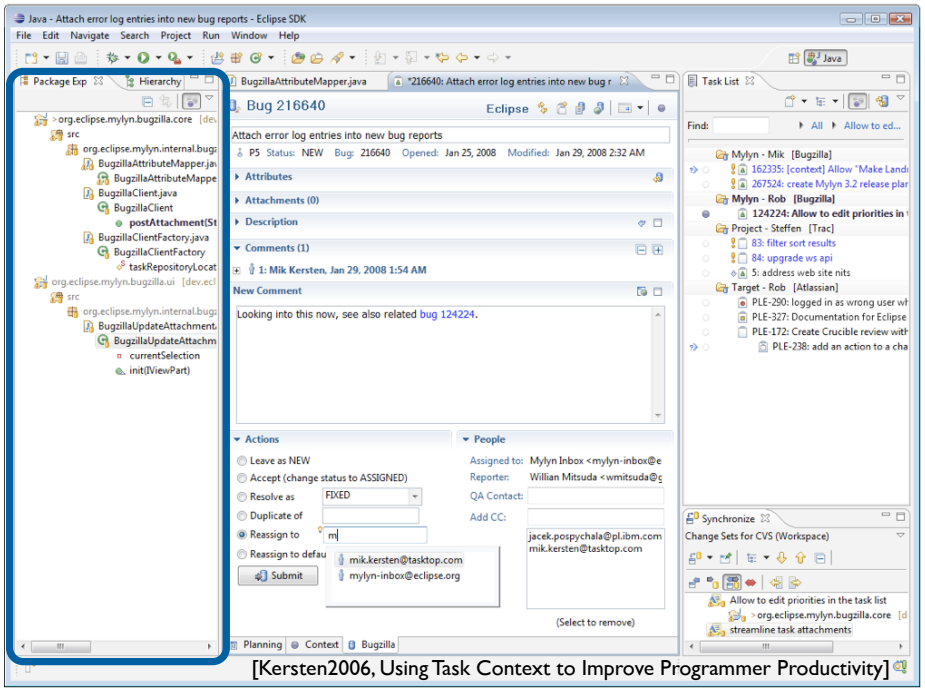
Tools Used in Eclipse

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



Easing Access to Task Context



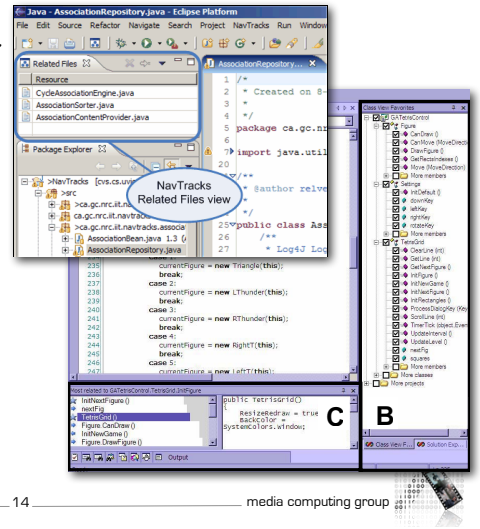


[Kersten2006, Using Task Context to Improve Programmer Productivity]

Recommender Tools

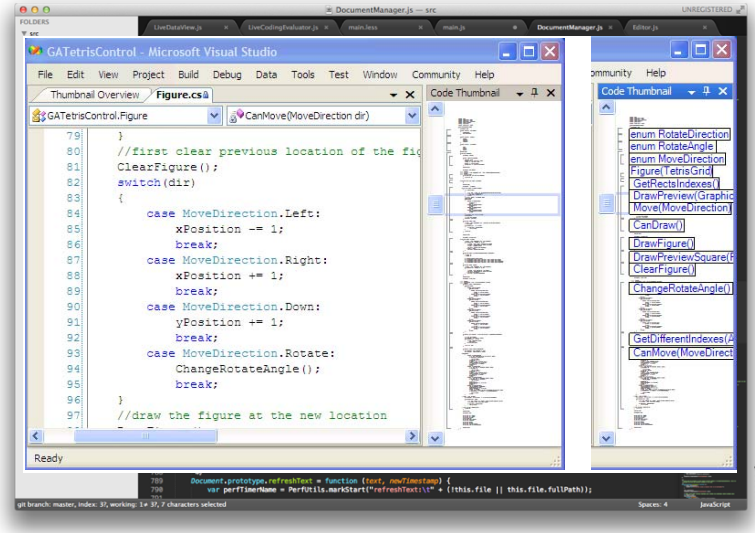
[Singer2005, NavTracks: supporting navigation in software maintenance]
[DeLine2005, Easing program comprehension by sharing navigation data]
[Čubranic´2005, Hipikat: recommending pertinent software development artifacts]

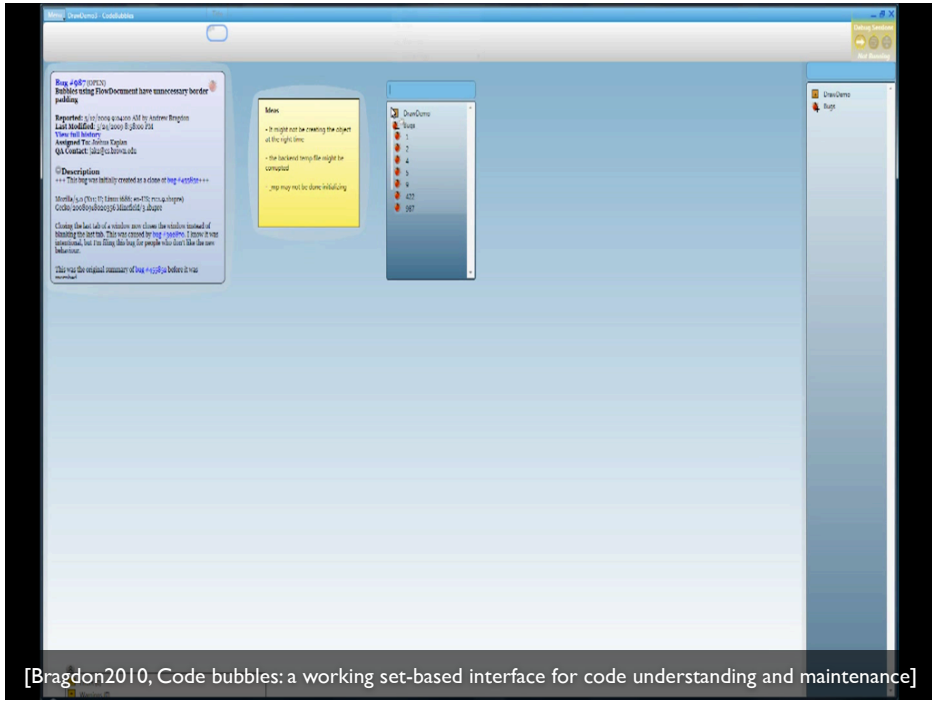
- Calculate a Degree of Interest for source code elements based on:
  - reading history
  - editing history
  - history of other team members
  - information from version control systems
- Remaining Problems:
  - Still only text-based visualization
  - Recommendations for irrelevant code are still irrelevant



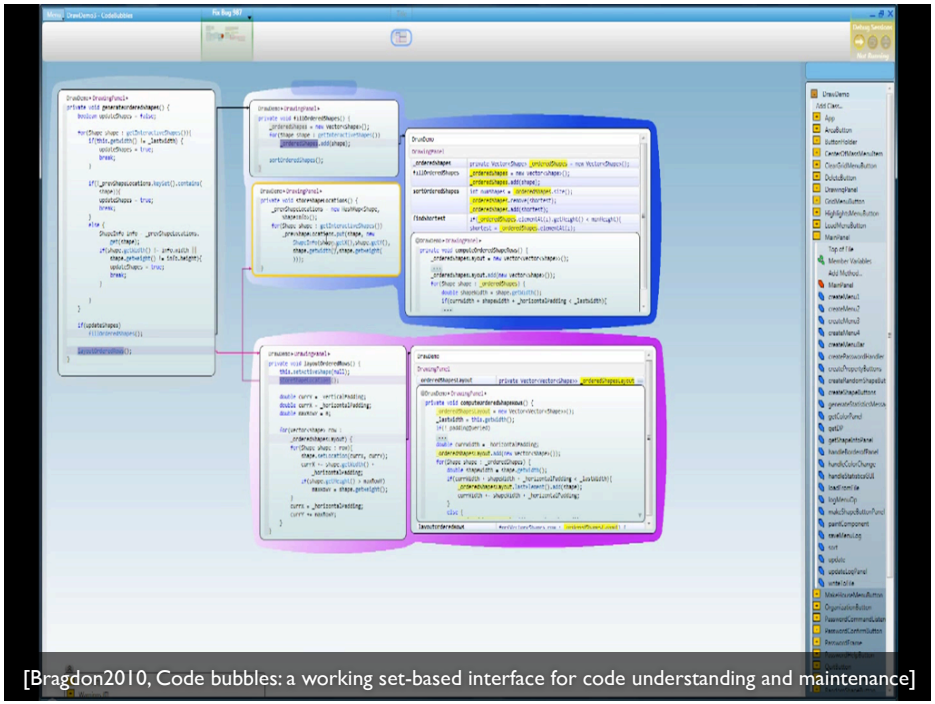
Changing the Presentation

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

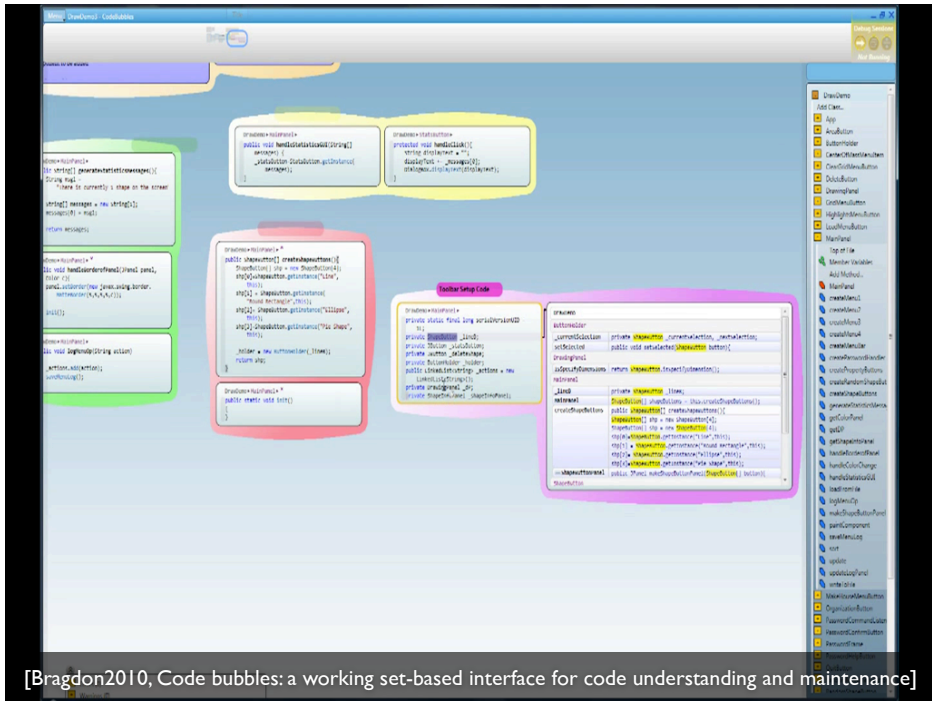




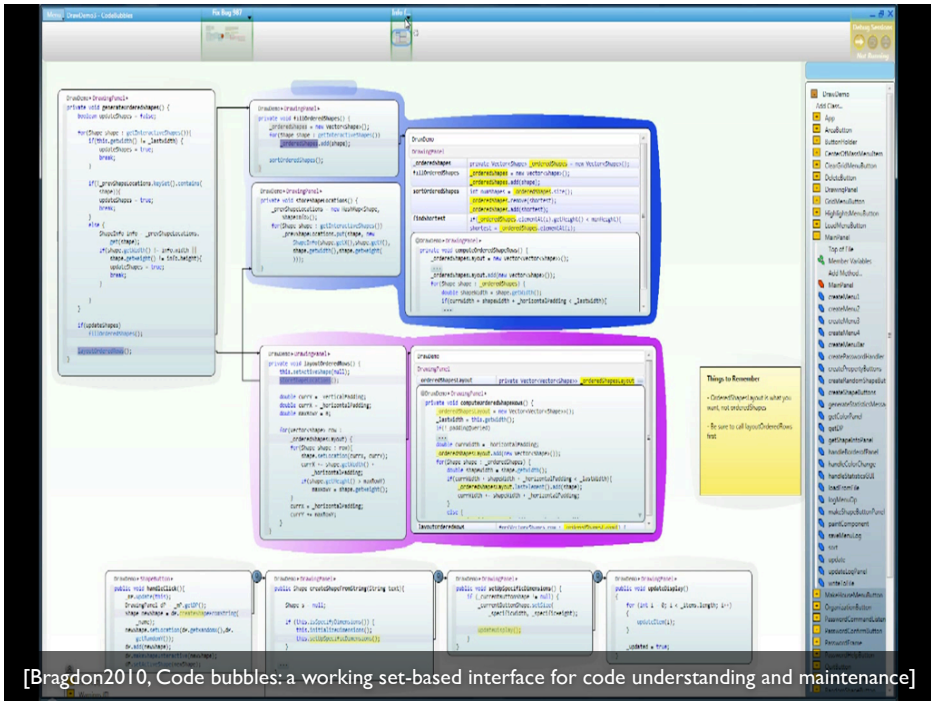
[Bragdon2010, Code bubbles: a working set-based interface for code understanding and maintenance]



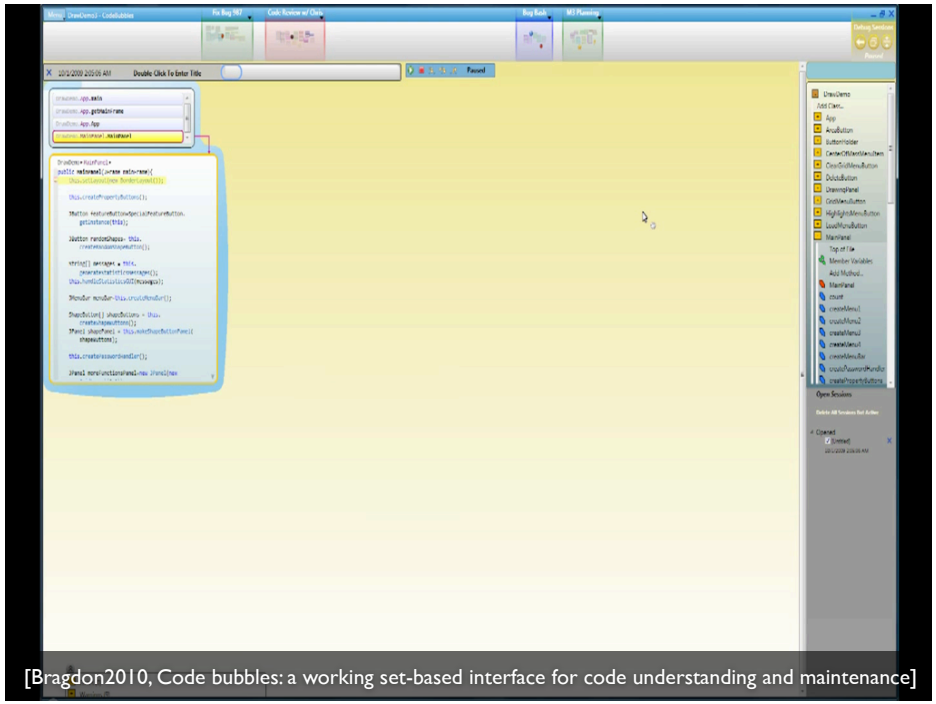
[Bragdon2010, Code bubbles: a working set-based interface for code understanding and maintenance]



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



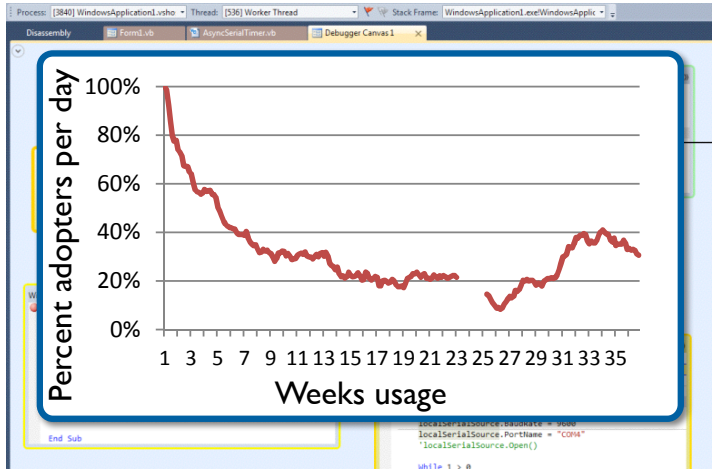
[Bragdon2010, Code bubbles: a working set-based interface for code understanding and maintenance]



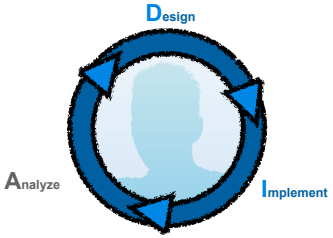
[Bragdon2010, Code bubbles: a working set-based interface for code understanding and maintenance]

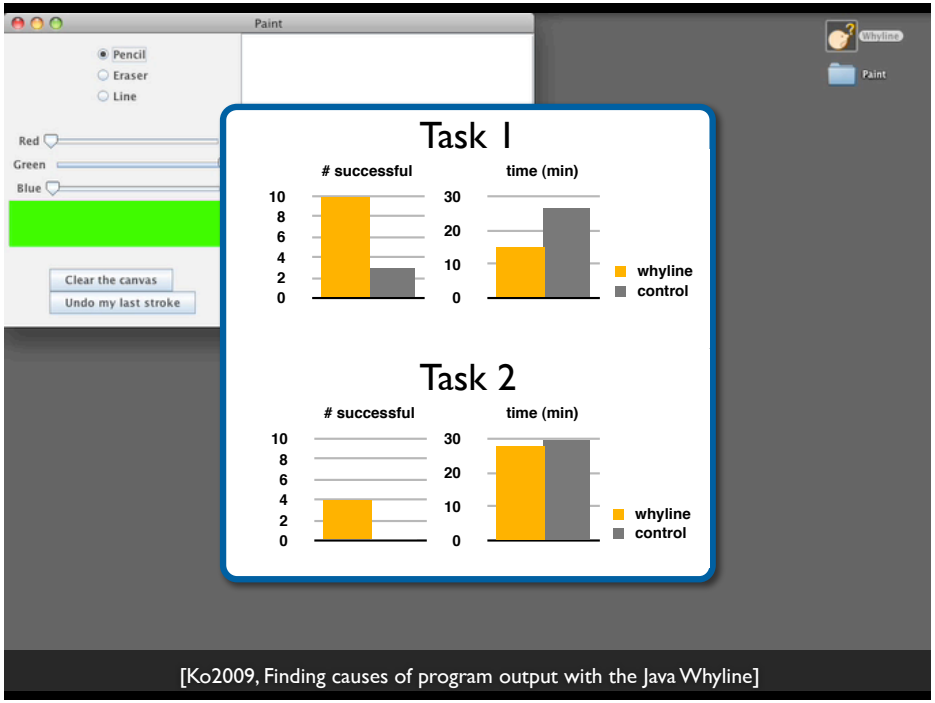
Canvas Interfaces in the Wild

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

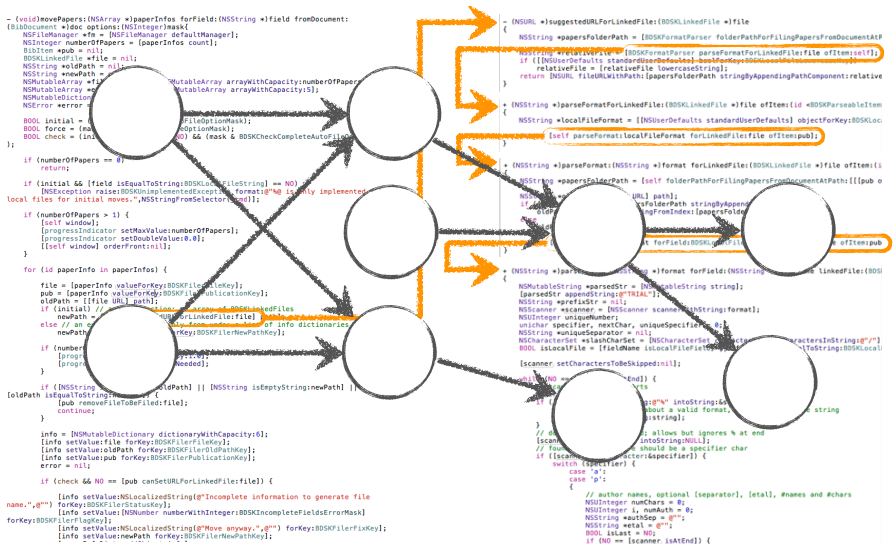


Utilizing the Call Graph





[Ko2009, Finding causes of program output with the Java Whyline]

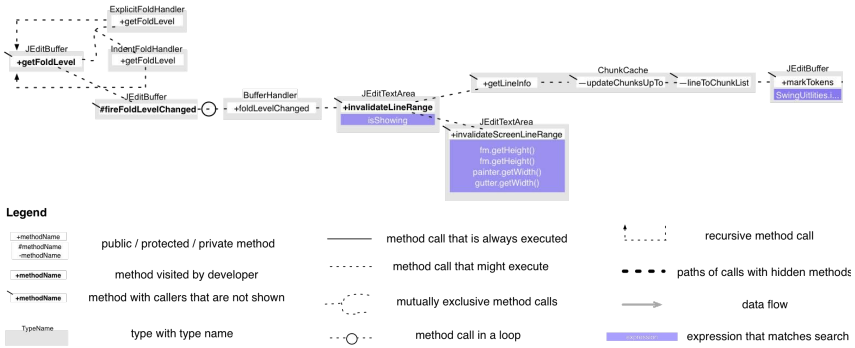


In practice: Feasible paths most interesting

[LaToza2010, Developers ask reachability questions]

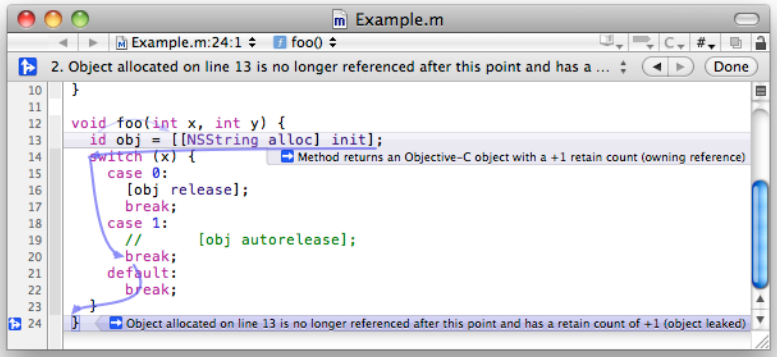
Utilizing Call Graph Information

[LaToza2010, Searching Across Paths]

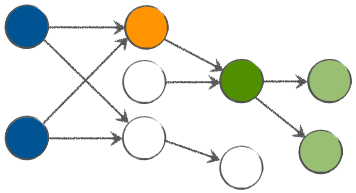


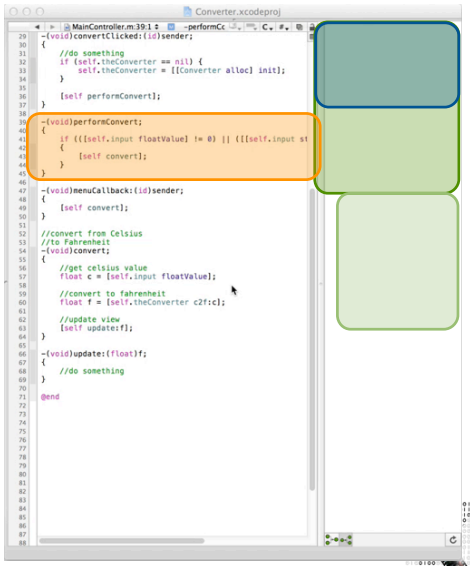
Static Analysis in the Wild

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



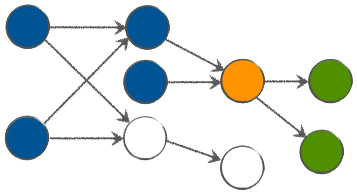
Call Hierarchy

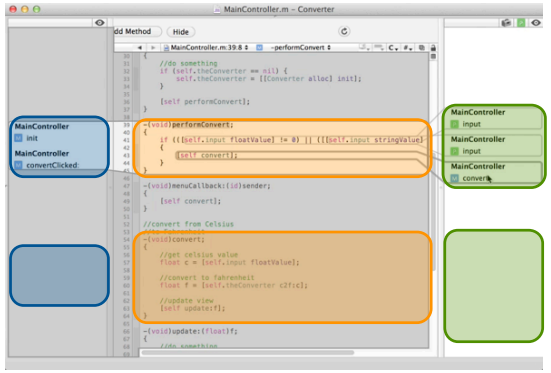




Stacksplorer

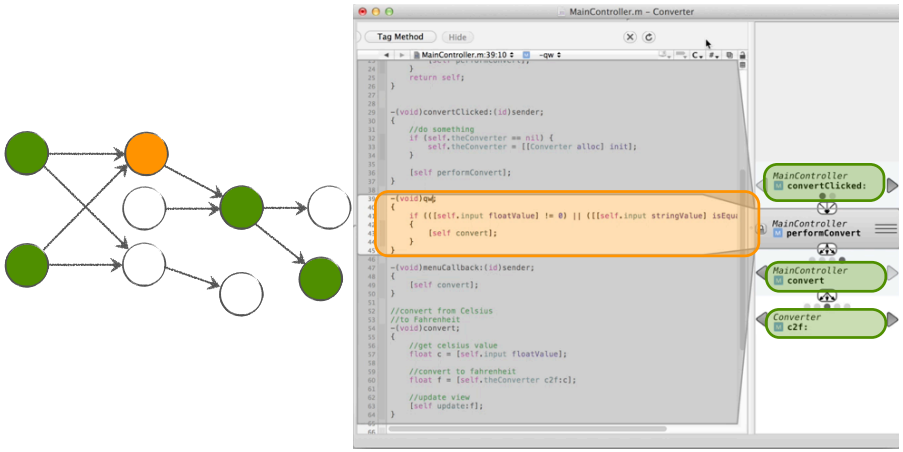
[Karrer2011, Stacksplorer: Call Graph Navigation Helps Increasing Code Maintenance Efficiency]



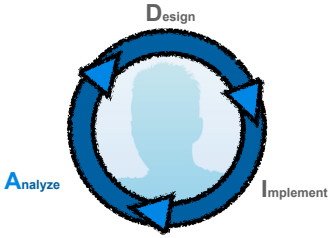


Blaze

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



Analyzing Navigation Behavior



Information Foraging Theory

Predator

Scent

Prey

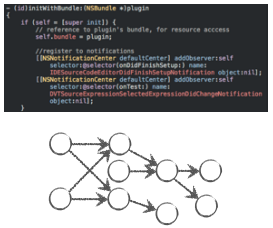
Information Foraging Theory

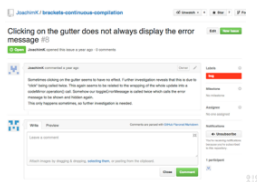
[Lawrance2010, Reactive information foraging for evolving goals]

Predator

Scent

Prey





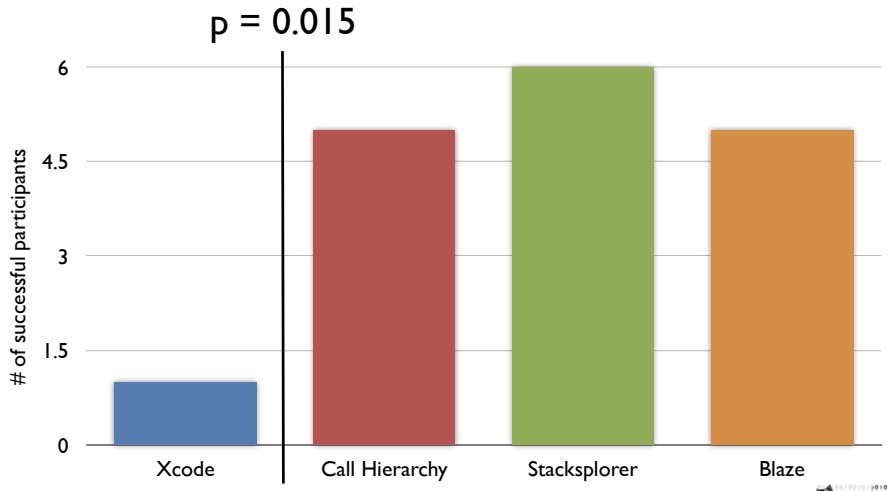
	Xcode	Call Hierarchy	Stacksplorer	Blaze
Find Change Location	Task Success Task Completion Time			
Side Effects of Change				

33 Developers

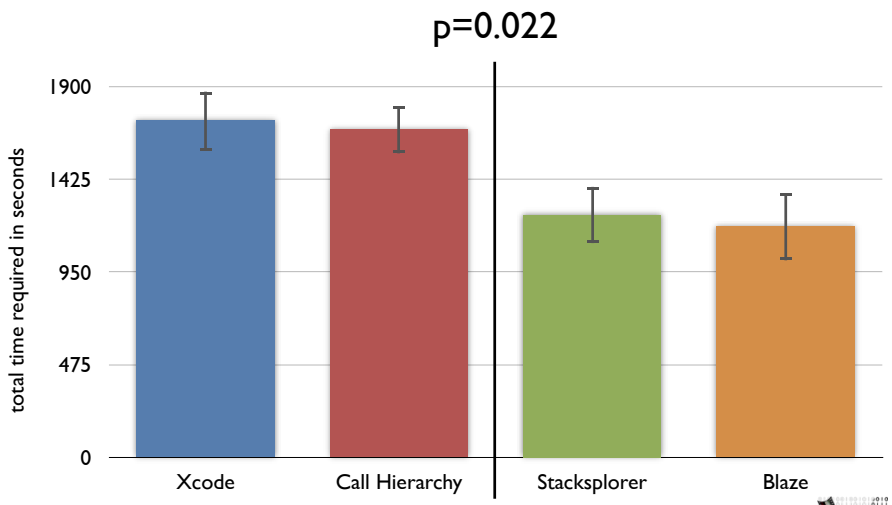
80.000 Lines of Code

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

Task Success



Task Completion Time



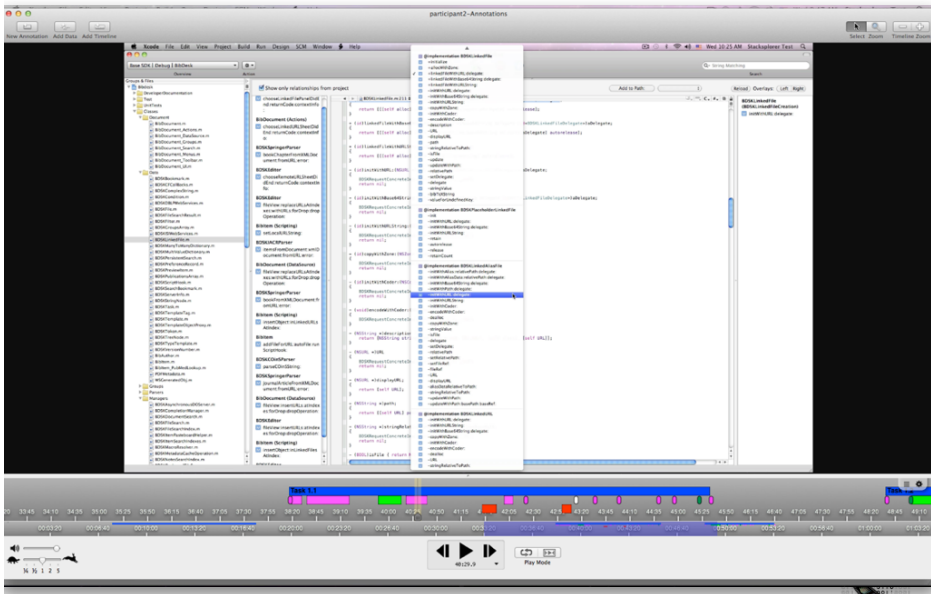
Effectiveness: Xcode | Call Hierarchy | Stacksplorer | Blaze

Efficiency: Xcode | Call Hierarchy | Stacksplorer | Blaze

Why?

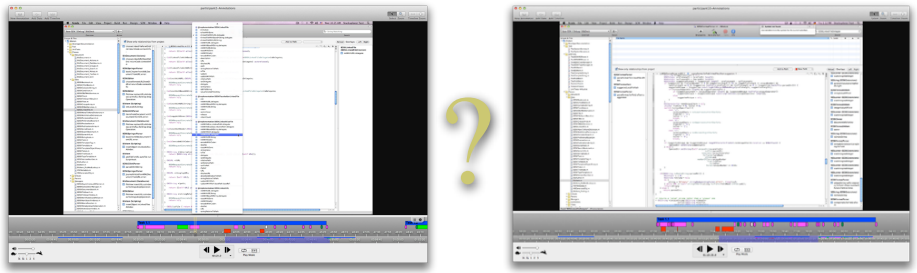
UI Differences

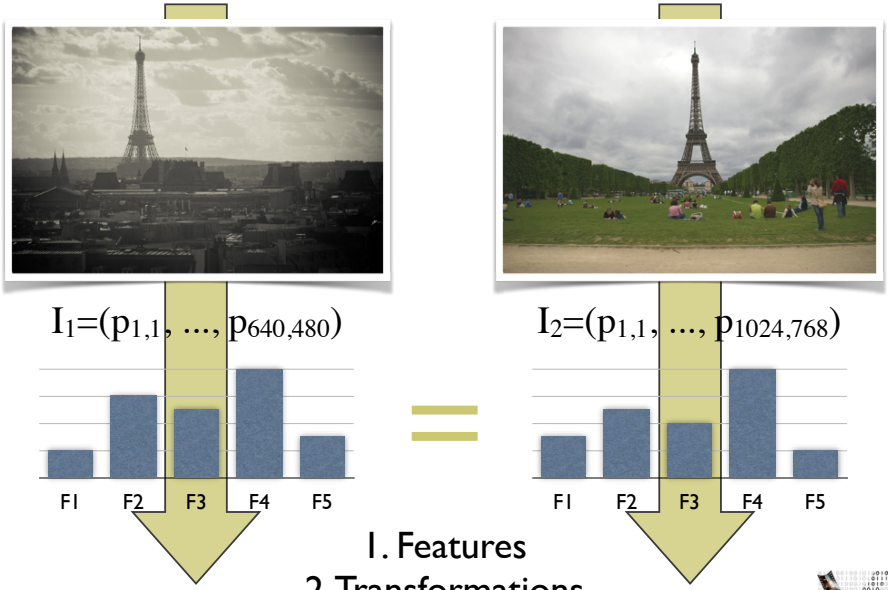
Navigation Behavior



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

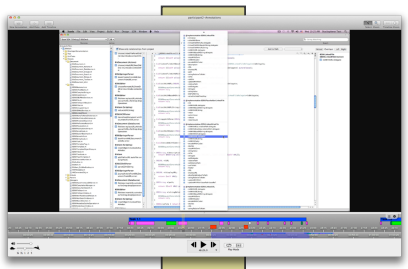
Comparing Navigation Behavior

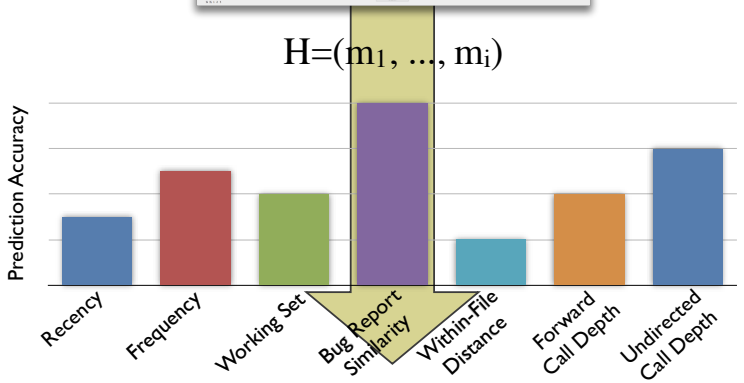




1. Features
2. Transformations

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





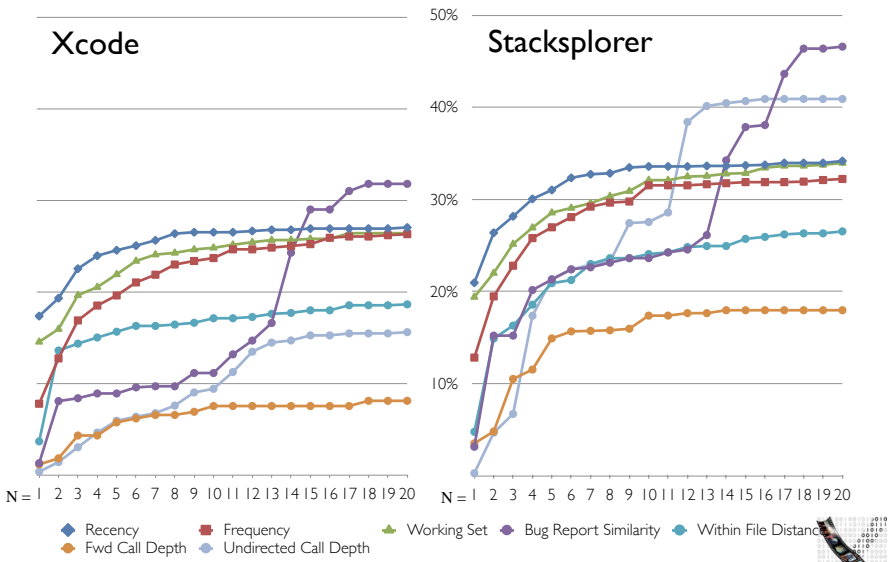
A Predictor

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

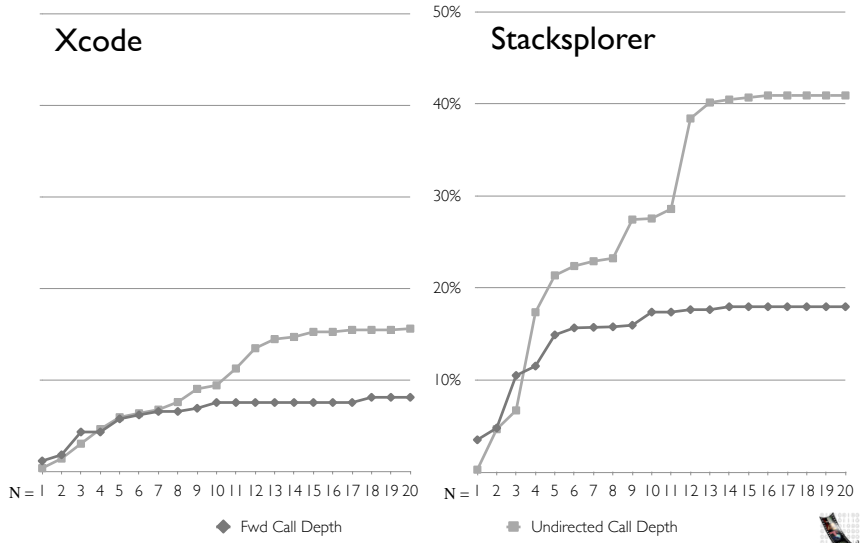
H=(m	Navigation History	H = (a, b, a, d)
M	All methods known to developer at time i	M
A	Activation value for each method in	A A
R	Rank-transformed version of	R R

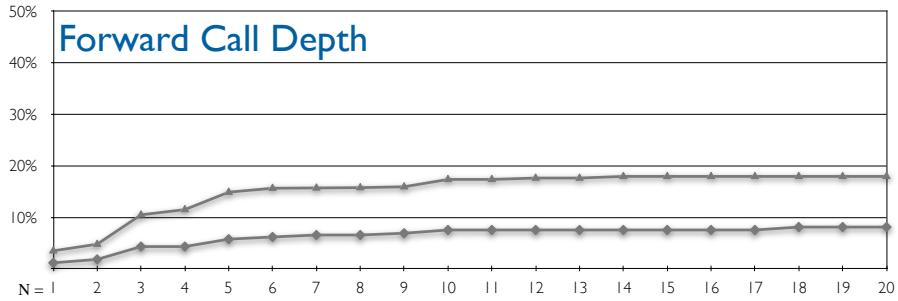
Result: N top-ranked methods

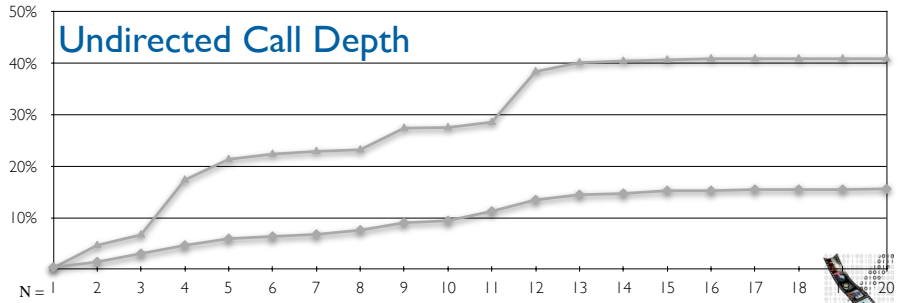
Prediction Accuracy

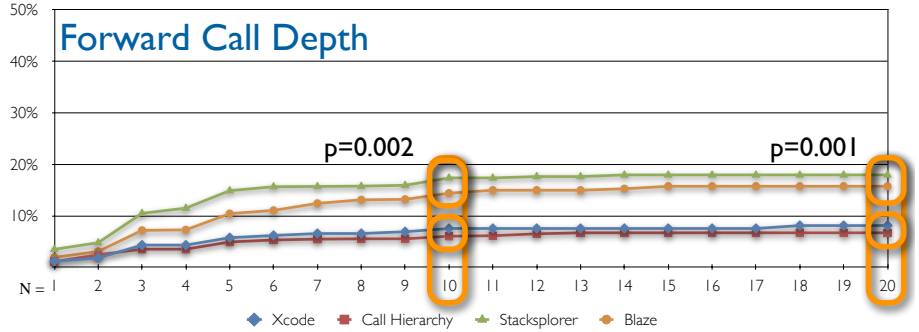


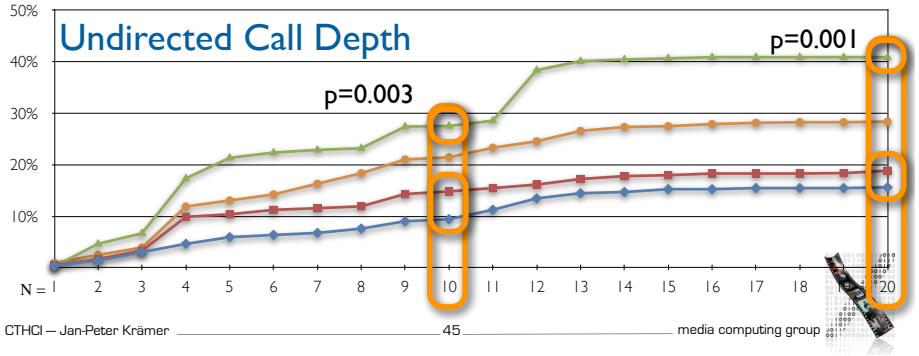
Prediction Accuracy



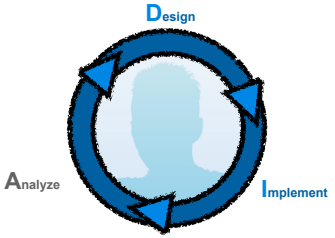


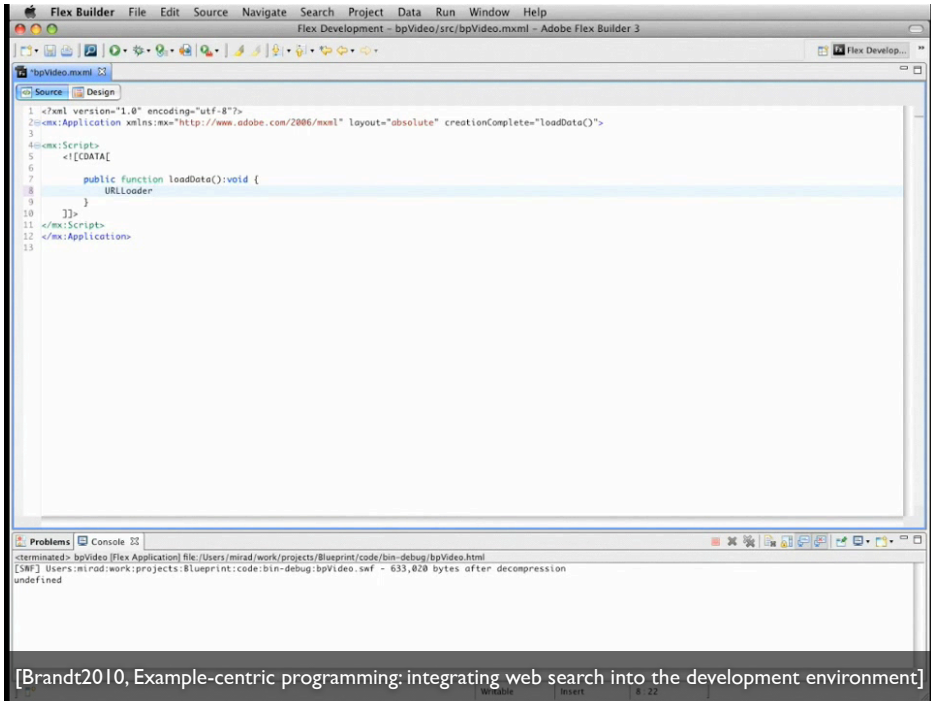






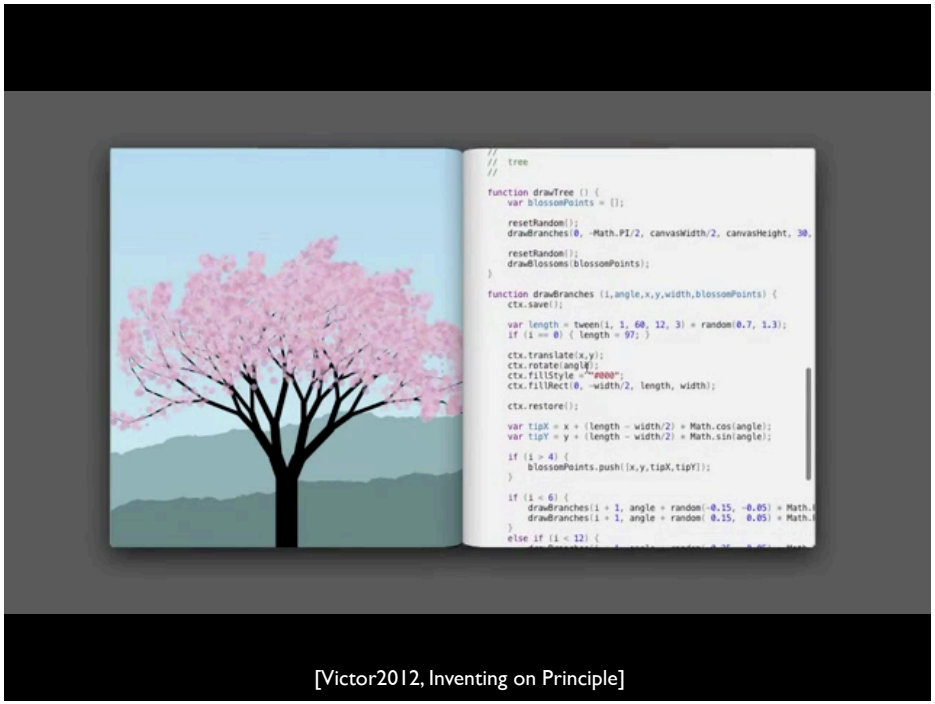
Away from static analysis only





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

[Oney2012, Codelets: Linking Interactive Documentation and Example Code in the Editor]

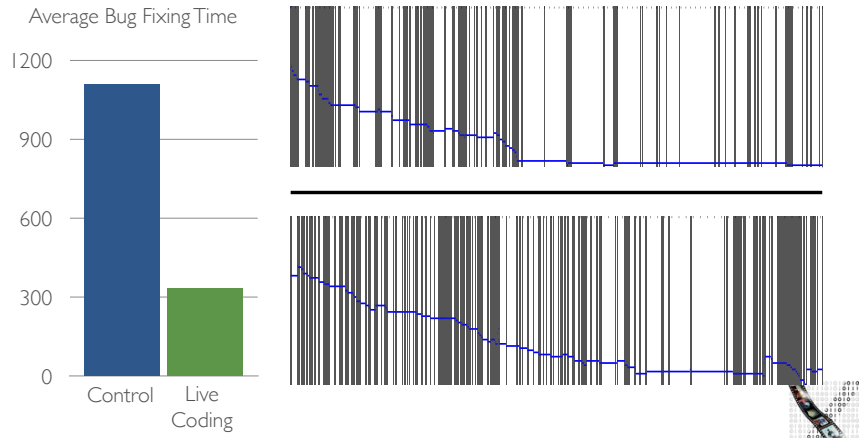


[Victor2012, Inventing on Principle]

Demo

Live Coding Affects Coding Behavior

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



Summary