FACILITATING INFORMATION MANAGEMENT IN INTEGRATED DEVELOPMENT ENVIRONMENTS THROUGH VISUAL INTERFACE ENHANCEMENTS

Haipeng Cai

Department of Computer Science and Engineering

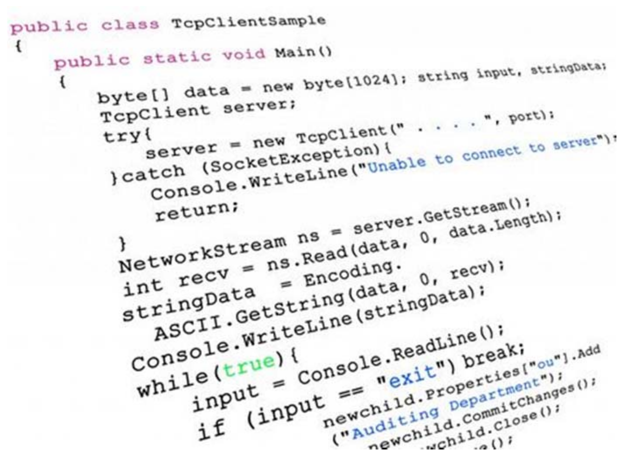
University of Notre Dame

UNIVERSITY OF NOTRE DAME
College of Engineering

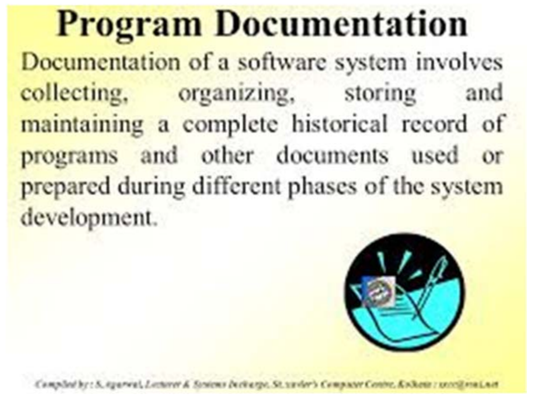
SEKM 2015

Supported by ONR Award N000141410037

Developers deal with deluge of information



```
public class TcpClientSample
 {
    public static void Main()
    {
        byte[] data = new byte[1024]; string input, stringData;
        TcpClient server;
        try{
           server = new TcpClient(" . . . . ", port);
        }catch (SocketException){
           Console.WriteLine("Unable to connect to server");
           return;

        }
        NetworkStream ns = server.GetStream();
        int recv = ns.Read(data, 0, data.Length);
        stringData  = Encoding.
           ASCII.GetString(data, 0, recv);
        Console.WriteLine(stringData);
        while(true){
           input = Console.ReadLine();
           if (input == "exit") break;
                 newchild.Properties["ou"].Add
                 ("Auditing Department");
                 newchild.CommitChanges();
                 newchild.Close();
```



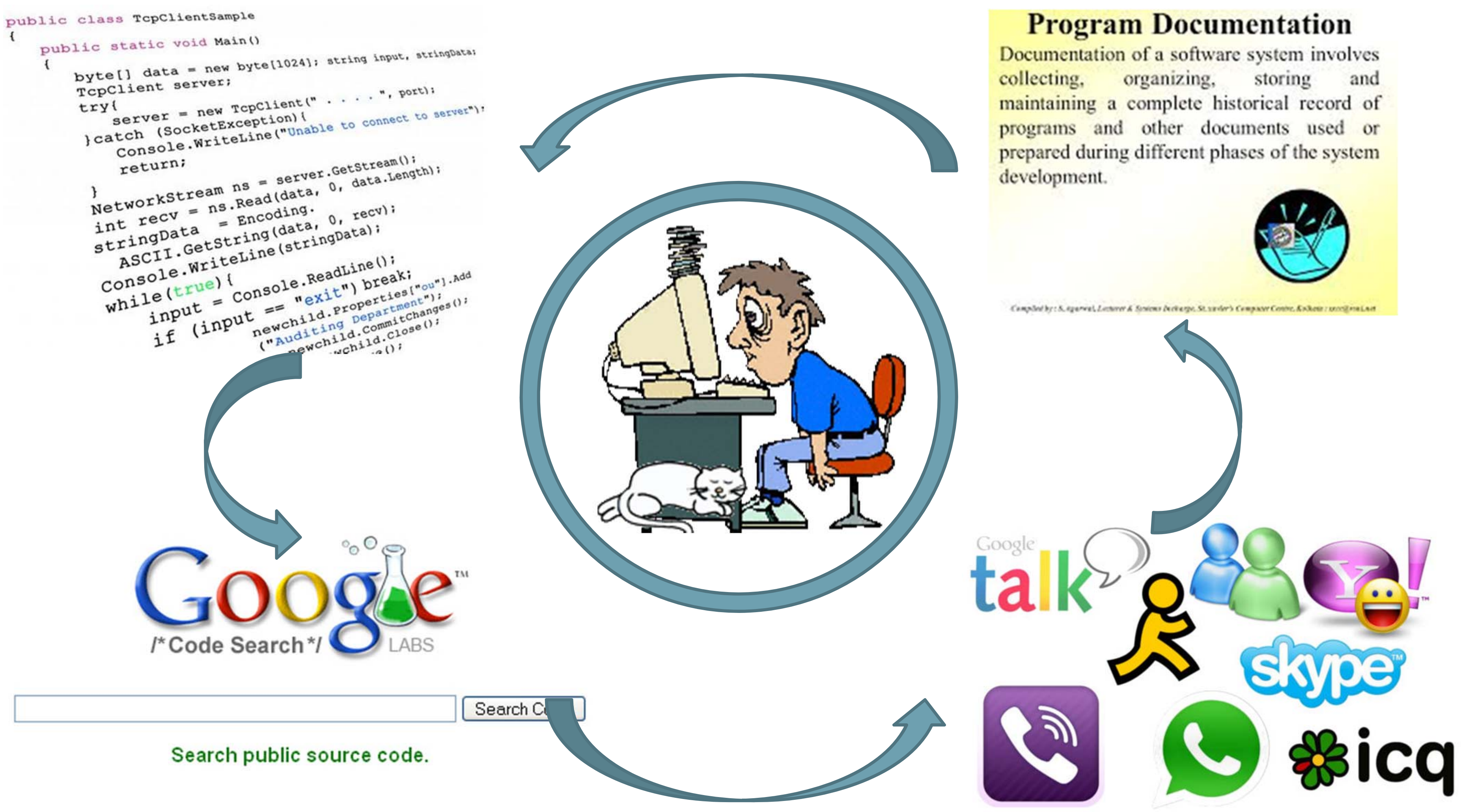
Program Documentation

Documentation of a software system involves collecting, organizing, storing and maintaining a complete historical record of programs and other documents used or prepared during different phases of the system development.

Compiled by : S. Agarwal, Lecturer & Systems Incharge, St. Xavier's Computer Centre, Kolkata : [illegible]

Pictures courtesy of *Google Image Search*

And switch among different information sources



```
public class TcpClientSample
{
    public static void Main()
    {
        byte[] data = new byte[1024]; string input, stringData;
        TcpClient server;
        try{
            server = new TcpClient(" . . . . ", port);
        }catch (SocketException){
            Console.WriteLine("Unable to connect to server");
            return;
        }
        NetworkStream ns = server.GetStream();
        int recv = ns.Read(data, 0, data.Length);
        stringData    = Encoding.
            ASCII.GetString(data, 0, recv);
        Console.WriteLine(stringData);
        while(true){
            input = Console.ReadLine();
            if (input == "exit") break;
                newchild.Properties["ou"].Add
                ("Auditing Department");
                newchild.CommitChanges();
                newchild.Close();
```

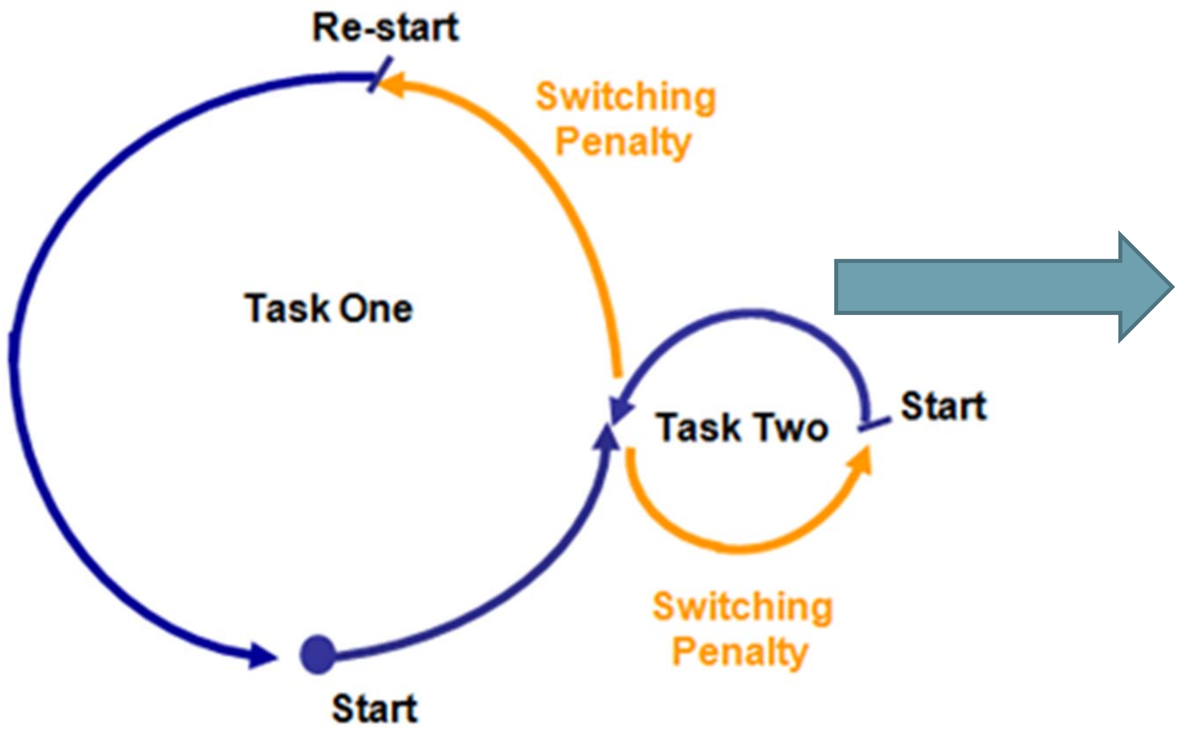
Program Documentation

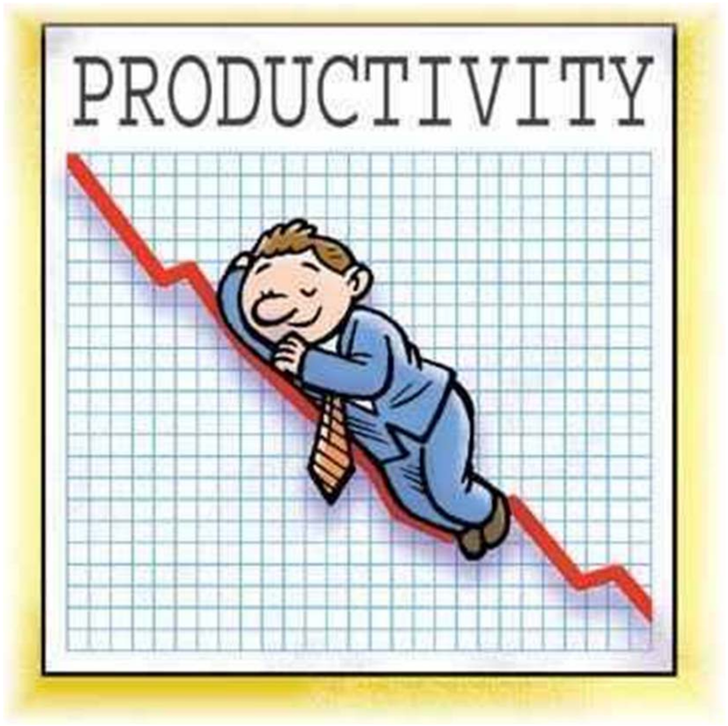
Documentation of a software system involves collecting, organizing, storing and maintaining a complete historical record of programs and other documents used or prepared during different phases of the system development.

Compiled by : S.Agarwal, Lecturer & Systems Incharge, St.Xavier's Computer Centre, Kolkata : [illegible]

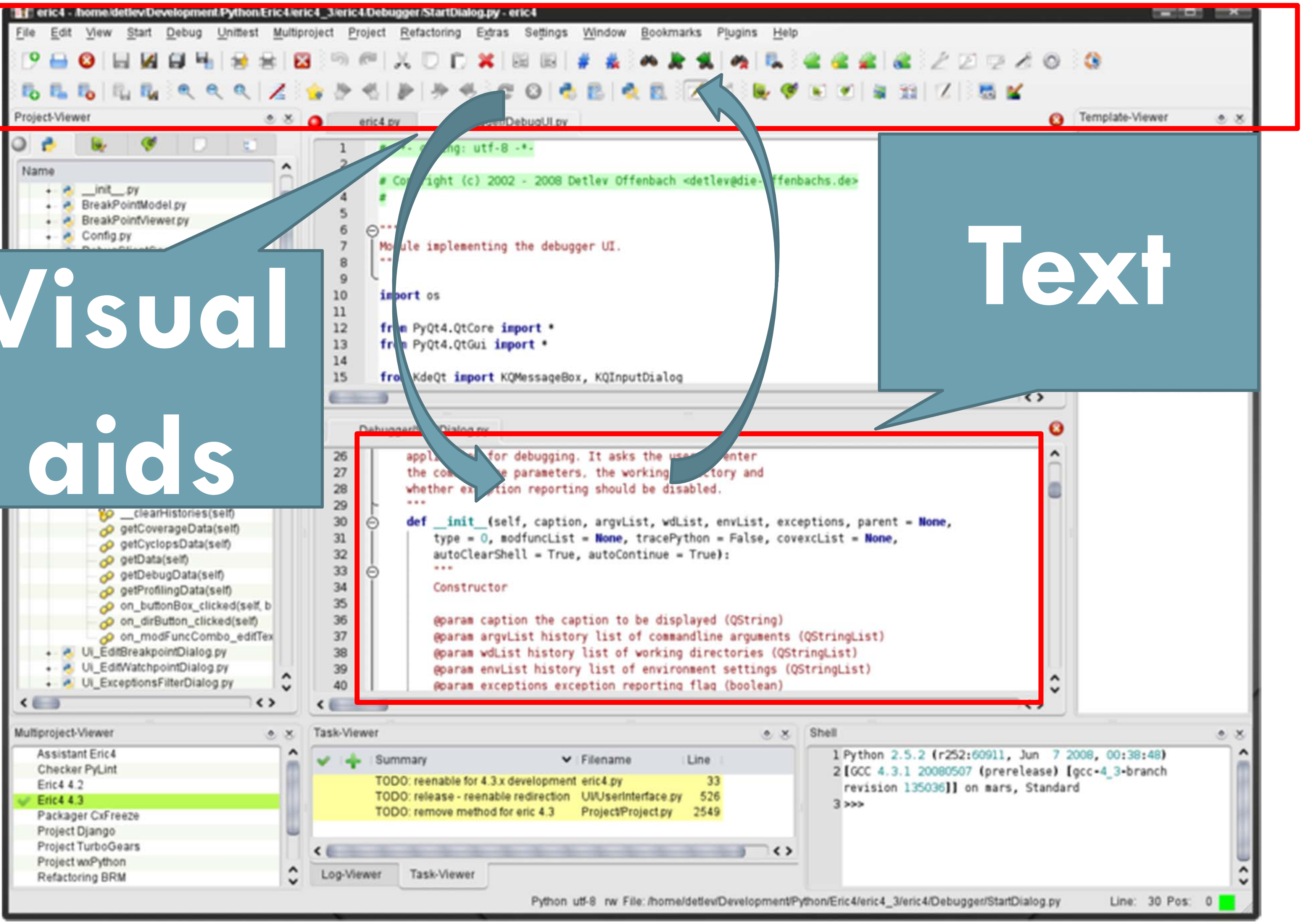
Frequent switches reduce productivity

- Mental-model interruption
- Individual-task latency
- ……





Modern IDEs help yet still suffer


eric4 - /home/detlev/Development/Python/Eric4/eric4_3/eric4/Debugger/StartDialog.py - eric4
File Edit View Start Debug Unittest Multiproject Project Refactoring Extras Settings Window Bookmarks Plugins Help
Project-Viewer
Template-Viewer
eric4.py
DebugUI.py
Name
__init__.py
BreakPointModel.py
BreakPointViewer.py
Config.py
clearHistories(self)
getCoverageData(self)
getCyclopsData(self)
getData(self)
getDebugData(self)
getProfilingData(self)
on_buttonBox_clicked(self, b
on_dirButton_clicked(self)
on_modFuncCombo_editTex
Ui_EditBreakpointDialog.py
Ui_EditWatchpointDialog.py
Ui_ExceptionsFilterDialog.py
Multiproject-Viewer
Assistant Eric4
Checker PyLint
Eric4 4.2
Eric4 4.3
Packager CxFreeze
Project Django
Project TurboGears
Project wxPython
Refactoring BRM
Task-Viewer
Log-Viewer
Shell
Python utf-8 rw File: /home/detlev/Development/Python/Eric4/eric4_3/eric4/Debugger/StartDialog.py Line: 30 Pos: 0

Visual aids

Text

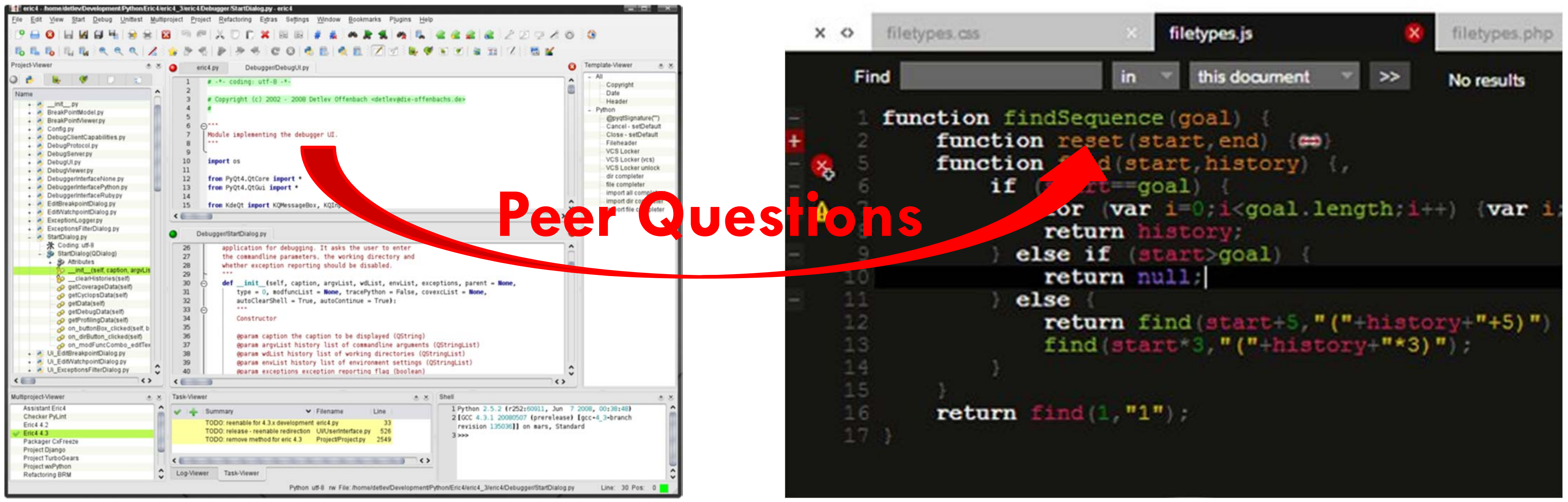
```
#-*- c    ng: utf-8 -*-

# Co    ight (c) 2002 - 2008 Detlev Offenbach <detlev@die-    ffenbachs.de>
#

'''
Mo ule implementing the debugger UI.
''

import os

fr  PyQt4.QtCore import *
fr  PyQt4.QtGui import *

fro KdeQt import KQMessageBox, KQInputDialog
```

```
    appl       for debugging. It asks the use    enter
    the co       e parameters, the working     ctory and
    whether ex    tion reporting should be disabled.
    '''
    def __init__(self, caption, argvList, wdList, envList, exceptions, parent = None,
        type = 0, modfuncList = None, tracePython = False, covexcList = None,
        autoClearShell = True, autoContinue = True):
        '''
        Constructor

        @param caption the caption to be displayed (QString)
        @param argvList history list of commandline arguments (QStringList)
        @param wdList history list of working directories (QStringList)
        @param envList history list of environment settings (QStringList)
        @param exceptions exception reporting flag (boolean)
```

Summary	Filename	Line
TODO: reenable for 4.3.x development	eric4.py	33
TODO: release - reenable redirection	UI/UserInterface.py	526
TODO: remove method for eric 4.3	Project/Project.py	2549

```
1 Python 2.5.2 (r252:60911, Jun  7 2008, 00:38:48)
2 [GCC 4.3.1 20080507 (prerelease) [gcc-4_3-branch
  revision 135036]] on mars, Standard
3 >>>
```

Modern IDEs help yet still suffer



```
# -*- coding: utf-8 -*-

# Copyright (c) 2002 - 2008 Detlev Offenbach <detlev@die-offenbachs.de>
#

"""
Module implementing the debugger UI.
"""

import os

from PyQt4.QtCore import *
from PyQt4.QtGui import *

from KdeQt import KQMessageBox, KQIn
```

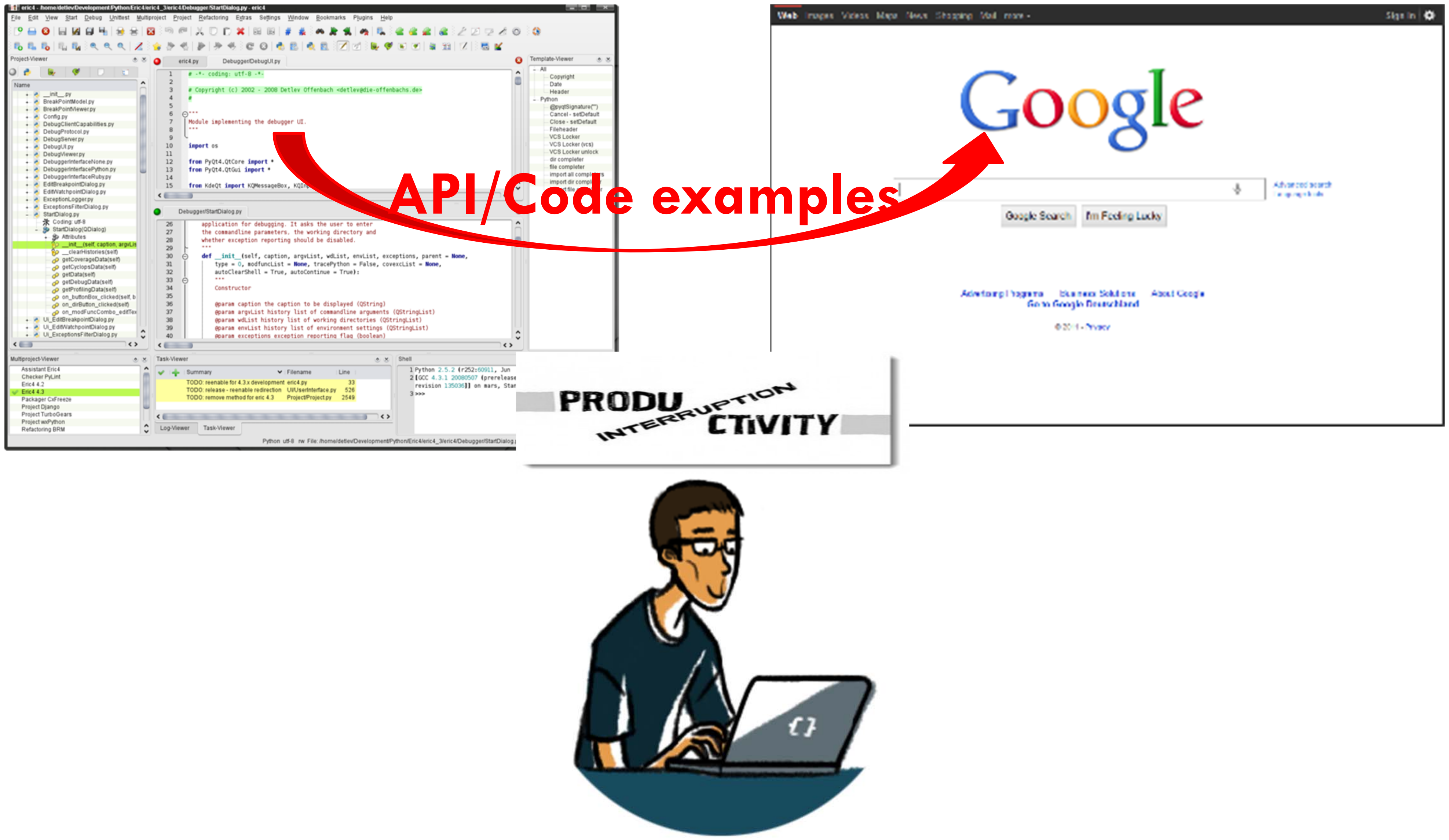
```
    application for debugging. It asks the user to enter
    the commandline parameters, the working directory and
    whether exception reporting should be disabled.
    """
    def __init__(self, caption, argvList, wdList, envList, exceptions, parent = None,
        type = 0, modfuncList = None, tracePython = False, covexcList = None,
        autoClearShell = True, autoContinue = True):
        """
        Constructor

        @param caption the caption to be displayed (QString)
        @param argvList history list of commandline arguments (QStringList)
        @param wdList history list of working directories (QStringList)
        @param envList history list of environment settings (QStringList)
        @param exceptions exception reporting flag (boolean)
```

	Summary	Filename	Line
	TODO: reenable for 4.3.x development	eric4.py	33
	TODO: release - reenable redirection	UI/UserInterface.py	526
	TODO: remove method for eric 4.3	Project/Project.py	2549

```
1 Python 2.5.2 (r252:60911, Jun  7 2008, 00:38:48)
2 [GCC 4.3.1 20080507 (prerelease) [gcc-4_3-branch
revision 135036]] on mars, Standard
3 >>>
```

```
function findSequence(goal) {
  function reset(start,end) {⟺}
  function find(start,history) {,
    if (start==goal) {
      for (var i=0;i<goal.length;i++) {var i;
      return history;
    } else if (start>goal) {
      return null;
    } else {
      return find(start+5,"("+history+"+5)")
      find(start*3,"("+history+"*3)");
    }
  }
  return find(1,"1");
}
```

Modern IDEs help yet still suffer



```
# -*- coding: utf-8 -*-

# Copyright (c) 2002 - 2008 Detlev Offenbach <detlev@die-offenbachs.de>
#

"""
Module implementing the debugger UI.
"""

import os

from PyQt4.QtCore import *
from PyQt4.QtGui import *

from KdeQt import KQMessageBox, KQIn
```

```
    application for debugging. It asks the user to enter
    the commandline parameters, the working directory and
    whether exception reporting should be disabled.
    """
    def __init__(self, caption, argvList, wdList, envList, exceptions, parent = None,
        type = 0, modfuncList = None, tracePython = False, covexcList = None,
        autoClearShell = True, autoContinue = True):
        """
        Constructor

        @param caption the caption to be displayed (QString)
        @param argvList history list of commandline arguments (QStringList)
        @param wdList history list of working directories (QStringList)
        @param envList history list of environment settings (QStringList)
        @param exceptions exception reporting flag (boolean)
```

Summary	Filename	Line
TODO: reenable for 4.3.x development	eric4.py	33
TODO: release - reenable redirection	UI/UserInterface.py	526
TODO: remove method for eric 4.3	Project/Project.py	2549

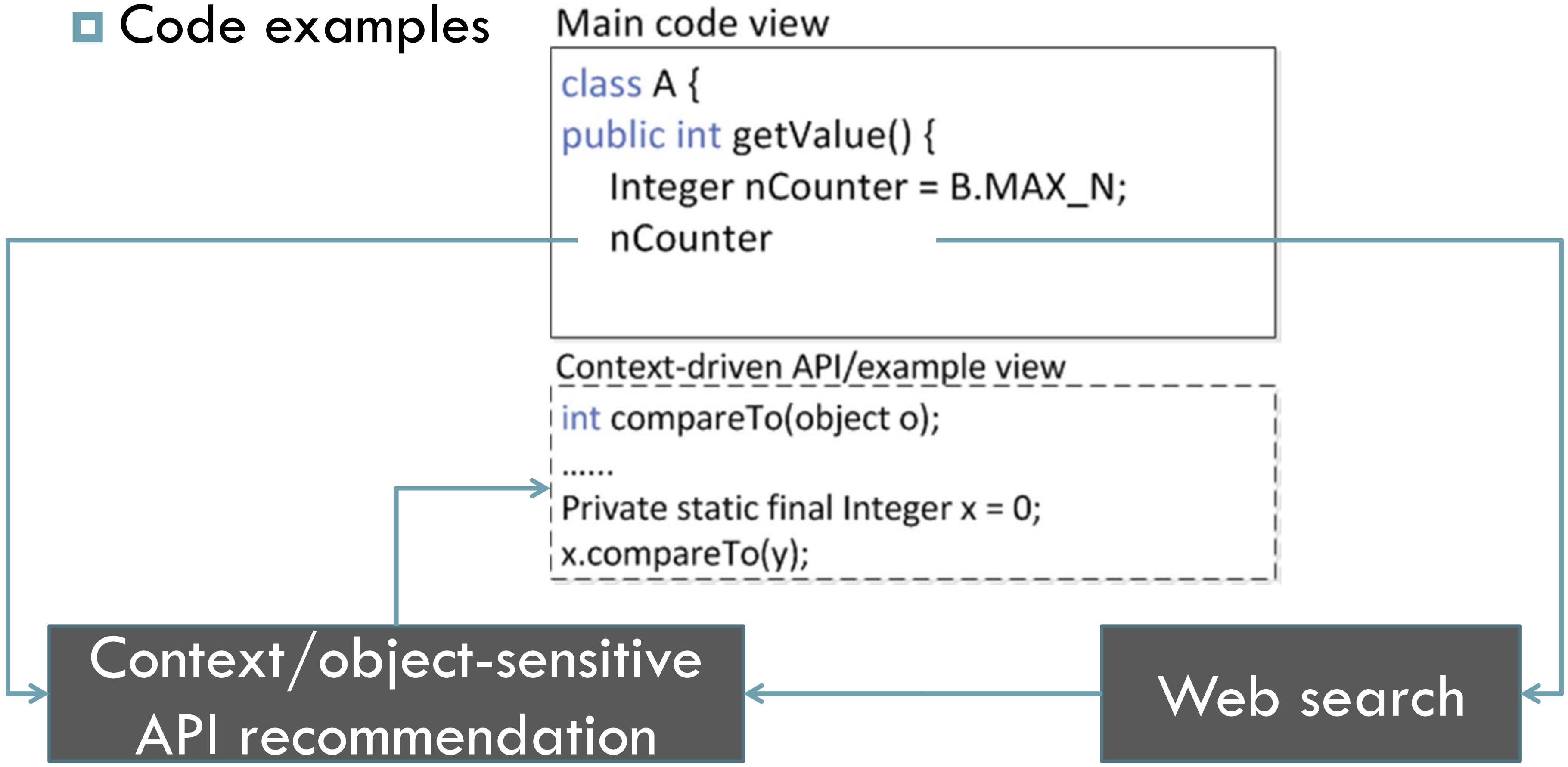
```
Python 2.5.2 (r252:60911, Jun
[GCC 4.3.1 20080507 (prerelease
revision 135036)] on mars, Stan
>>>
```

Facilitating information management in IDEs

- Interface enhancements
  - Context-driven API/code example views
  - Coworker views
  - In-situ interface
- Software visualization
  - Multiple code visualizations
  - Interactive linked visualization

Context-driven API/Code example views

- Automatic context-driven information foraging
  - API usage
  - Code examples

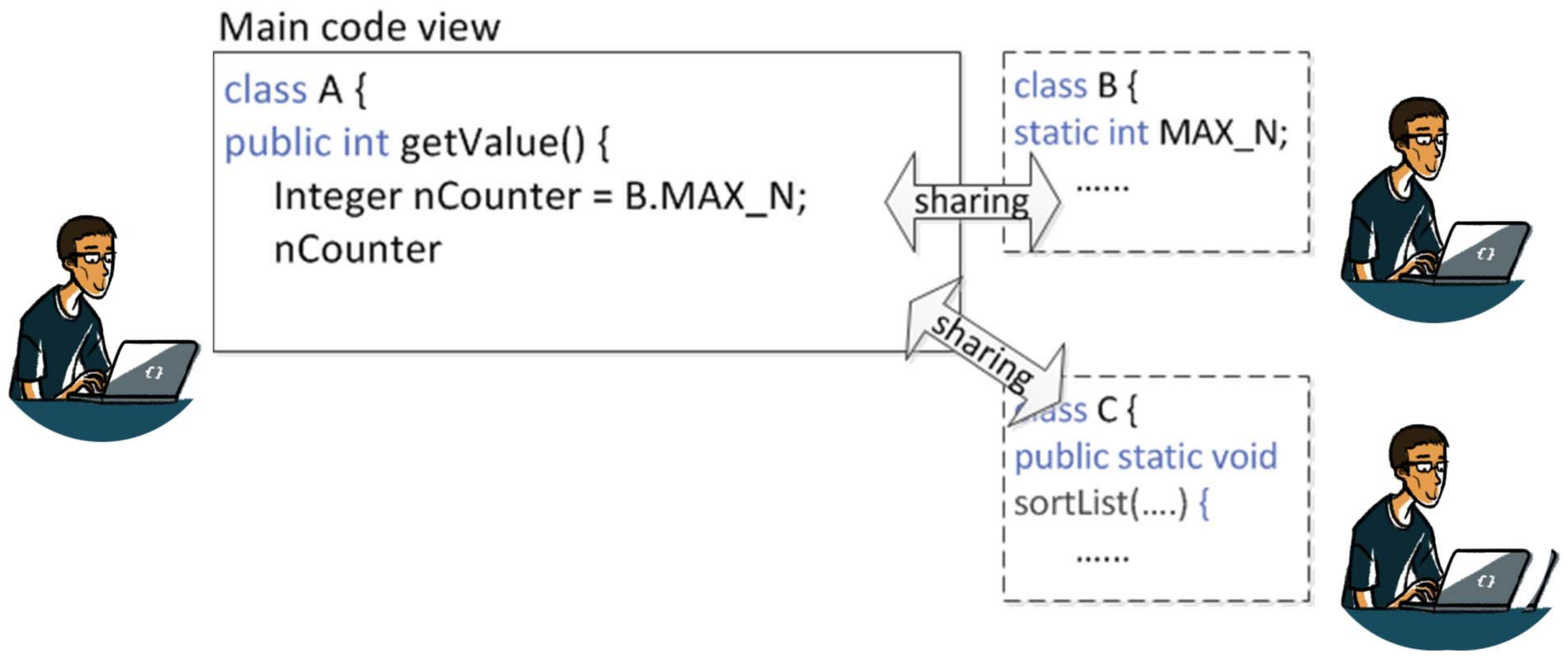

Main code view
Context-driven API/example view
Context/object-sensitive API recommendation
Web search

```
class A {
public int getValue() {
    Integer nCounter = B.MAX_N;
    nCounter
```

```
int compareTo(object o);
……
Private static final Integer x = 0;
x.compareTo(y);
```

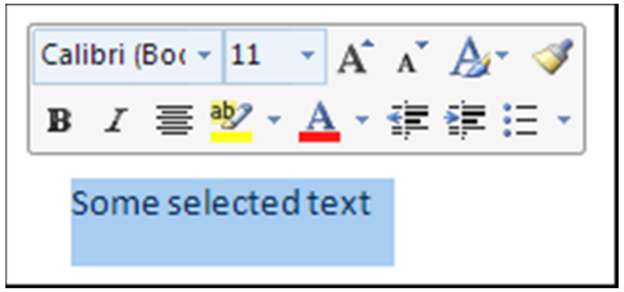
Co-worker views

- On-demand co-worker teaming up
  - Real-time coaching / demonstration
  - Online discussion

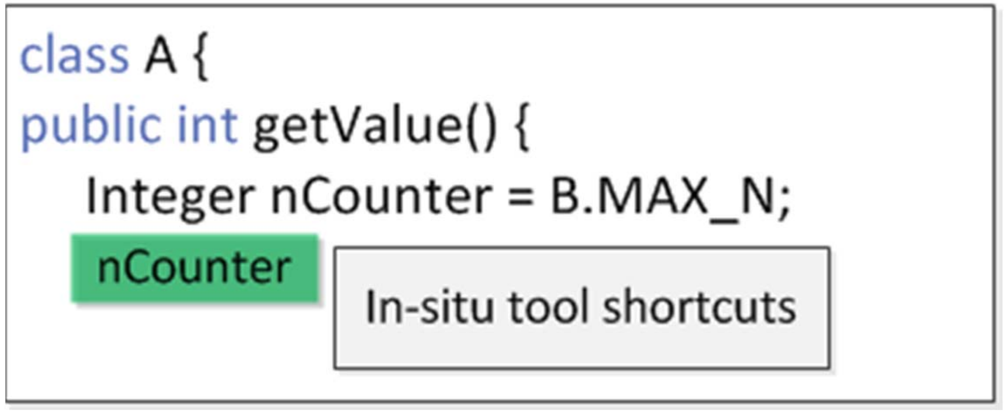


In-situ interface over code editing

- Integrate visual aids with source code editing
  - Automatic push/hiding of commonly used shortcuts
  - Object-sensitive recommendation



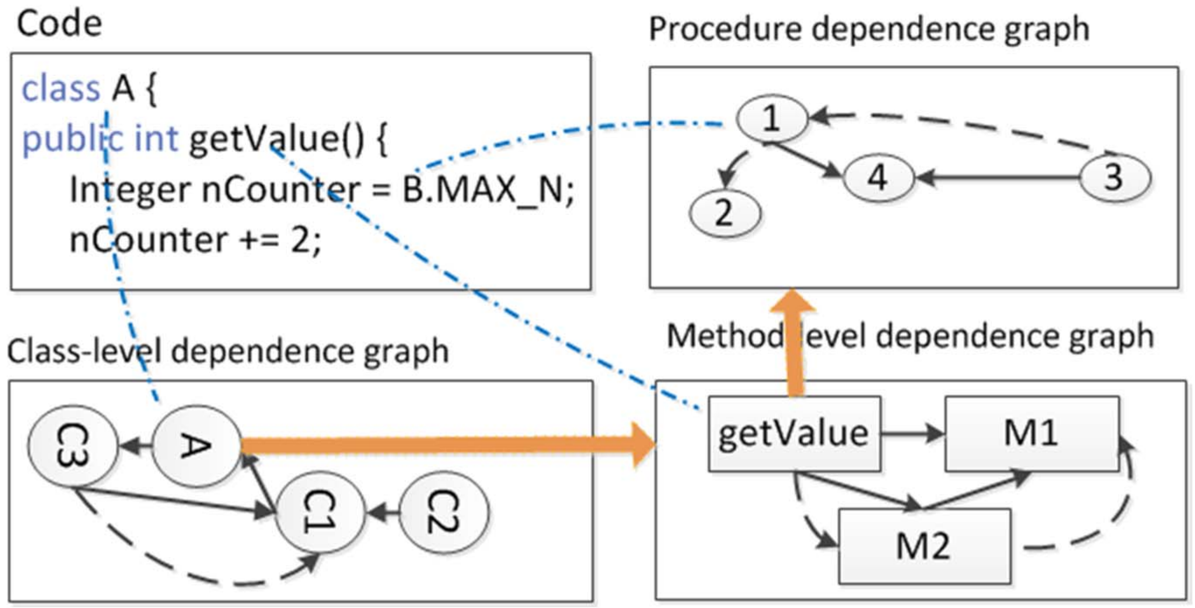
Main code view



```
class A {
public int getValue() {
    Integer nCounter = B.MAX_N;
```

Multiple visualizations of source code

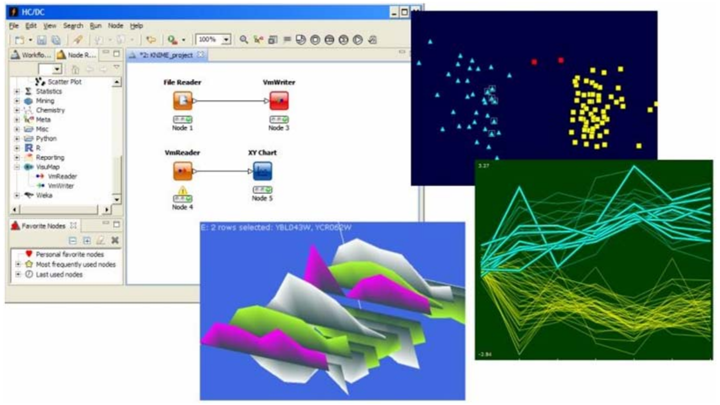
- Different representations of code in separate views
  - Same data
  - Alternative visual depiction (textual and graphical)

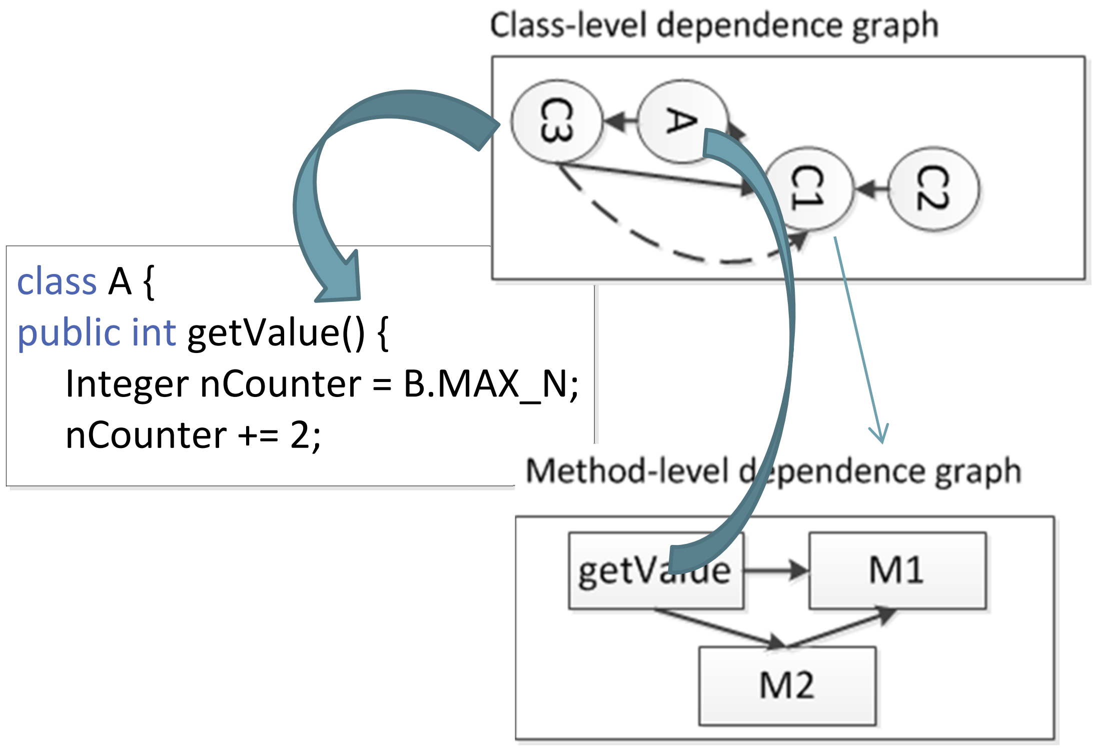

Code
Procedure dependence graph
1
2
3
4
Class-level dependence graph
C3
A
C1
C2
Method level dependence graph
getValue
M1
M2

```
class A {
public int getValue() {
    Integer nCounter = B.MAX_N;
    nCounter += 2;
```

Interactions over linked visualizations

- Linked operations across multiple views
  - Trigger an operation where it is most efficient to do
  - Map the operation to other representations

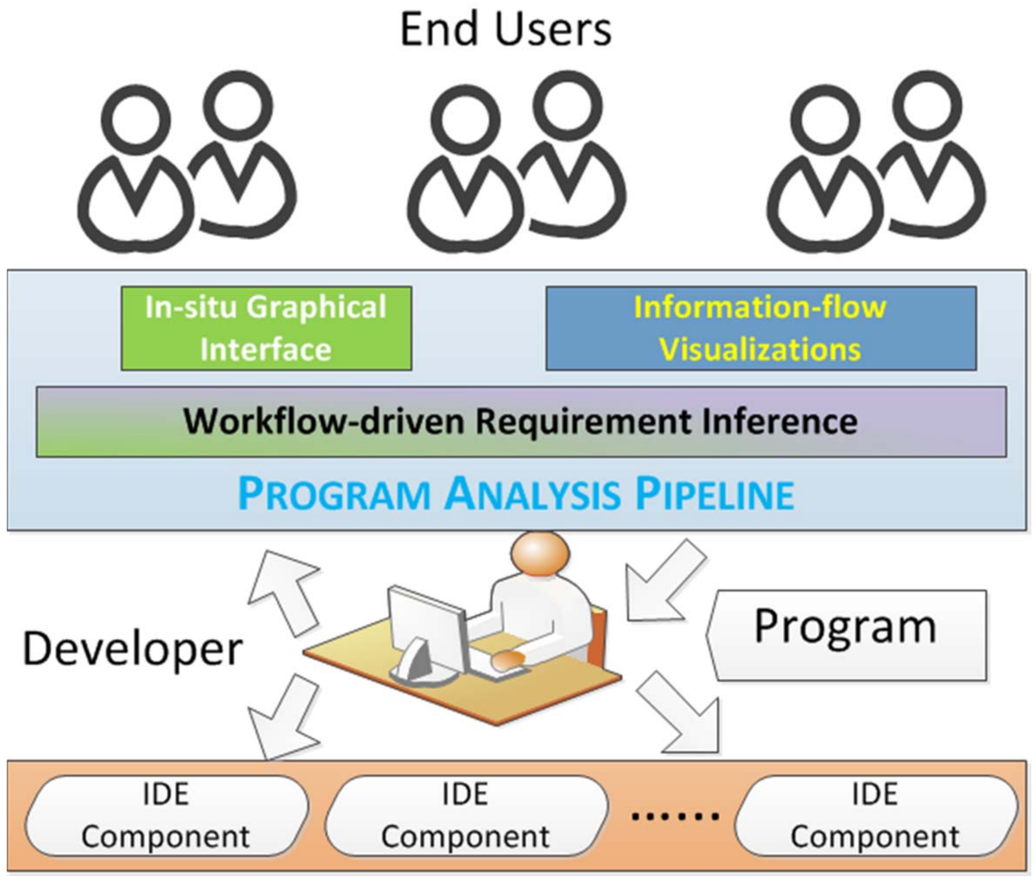




```
class A {
public int getValue() {
    Integer nCounter = B.MAX_N;
    nCounter += 2;
```

Beyond the visual enhancements

- Incorporating program analysis in IDEs
  - Information extracted from programs is more often needed than external sources --- outputs of program analysis


End Users
In-situ Graphical Interface
Information-flow Visualizations
Workflow-driven Requirement Inference
PROGRAM ANALYSIS PIPELINE
Developer
Program
IDE Component
IDE Component
......
IDE Component

Summing up

- Proposal
  - Motivation
    - Reduce context switching in dealing with multiple information needs with modern IDEs
  - Solution
    - Interface enhancements
    - Interactive code visualization
  - Approach
    - Reduce switching within an IDE
      - Multiple-view interactive linked visualization
      - In-situ interface
    - Reduce switching over an IDE
      - Co-worker views
      - API/code example views

Summing up

- Future work
  - Implementation
    - Via IDE plug-ins to start with
  - Evaluation
    - User studies
      - Groups using the enhanced IDE versus a traditional IDE
      - Coding and comprehension tasks
      - Differences in developer performance
      - Quality and time of task completion

Acknowledgements

“Facilitating Information Management in Integrated Development Environments through Visual Interface Enhancements”

Haipeng Cai

http://cse.nd.edu/~hcai/

hcai@nd.edu

Take-away

Three *interface* features and two *visualization* enhancements are proposed to facilitate information management in modern IDEs.