



# Research Overview

Lori L. Pollock, Professor  
Computer and Information Sciences  
University of Delaware

# My Journey

Program Analysis, Software Development & Maintenance Tools,  
Optimizing Compilers

'81 B.S. CS and Econ, Allegheny

'81-'86 PhD in CS, U of Pittsburgh

'86-'90 Assistant Prof, Rice U

'91- Assist, Assoc, Full Prof UD CIS

# What I do here at UD

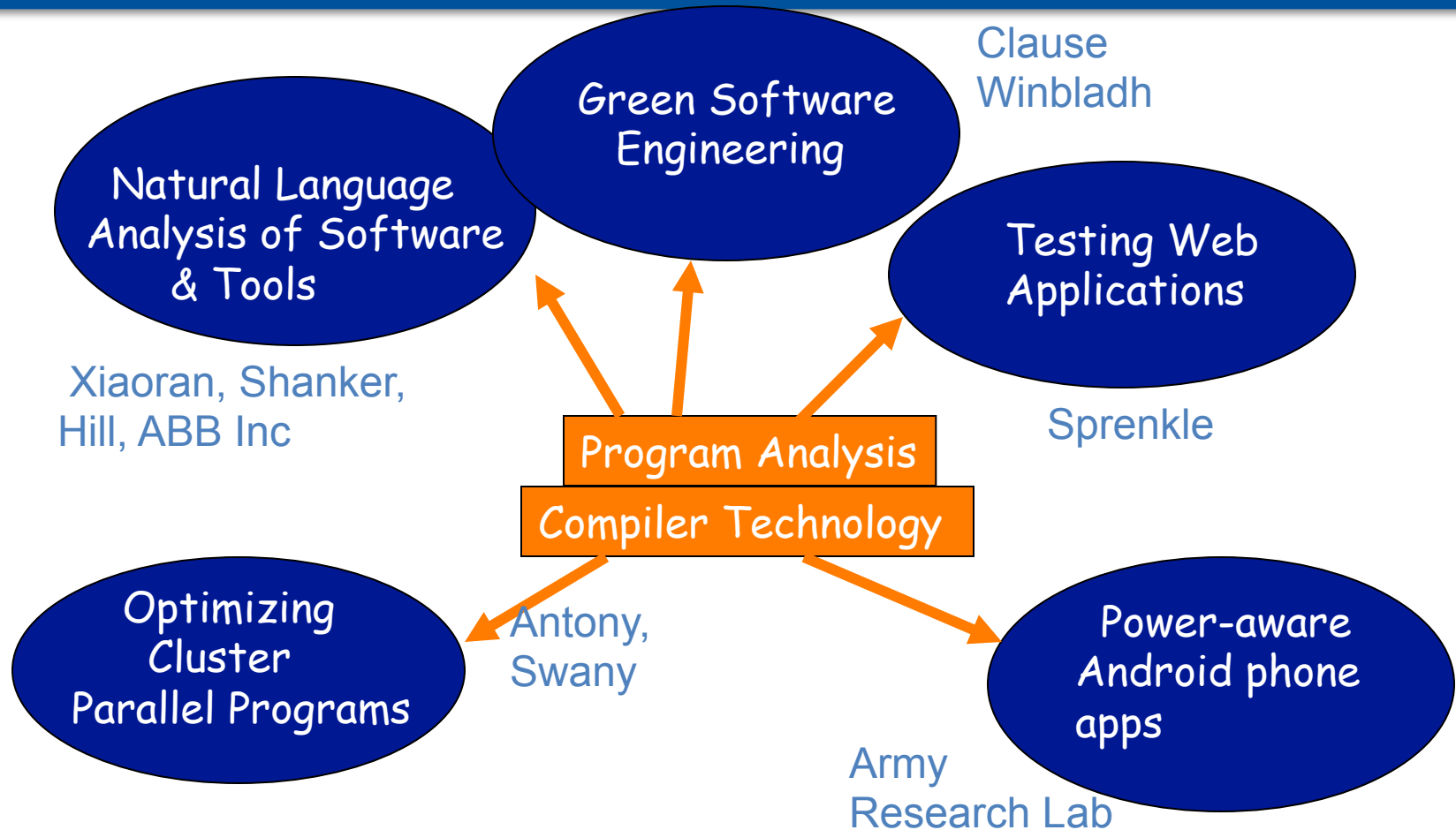
- Research
  - Software Analysis and Compilation Lab
    - 213 Smith Hall
  - Collaborations
    - Vijay Shanker (UD CIS), Terry Harvey (UD CIS), Jim Clause, Kristina Winbladh. ABB Inc, Army Research Lab, others.
- Graduate Teaching
  - CISC 672 Compilers
  - CISC 673 Program Analysis and Transformations
  - CISC 615 Software Testing and Maintenance
  - CISC 879 Software Tools and Environments
- Undergraduate Teaching
  - Learning Game Development (using XOs for middle school)
  - Python programming for non-majors
  - Study abroad programs

# What I do outside UD

- Associate Editor, Transactions on Software Engineering and Methodology (TOSEM)
- Computing Research Association (CRA)'s Committee on the Status of Women in Computing Research **(CRA-W)**
- **Mentoring** workshops
- Program committees,...



# Research Program Overview



Software Tools.....Testing.....Compilers.....Parallel Computing  
Empirical Studies.....Human subjects..... Building helpful tools

# Software is like a car.



It breaks.

# Software is like a car.



We want it to go faster.

# Software is like a car.



We want more features.



# Software is like a car.



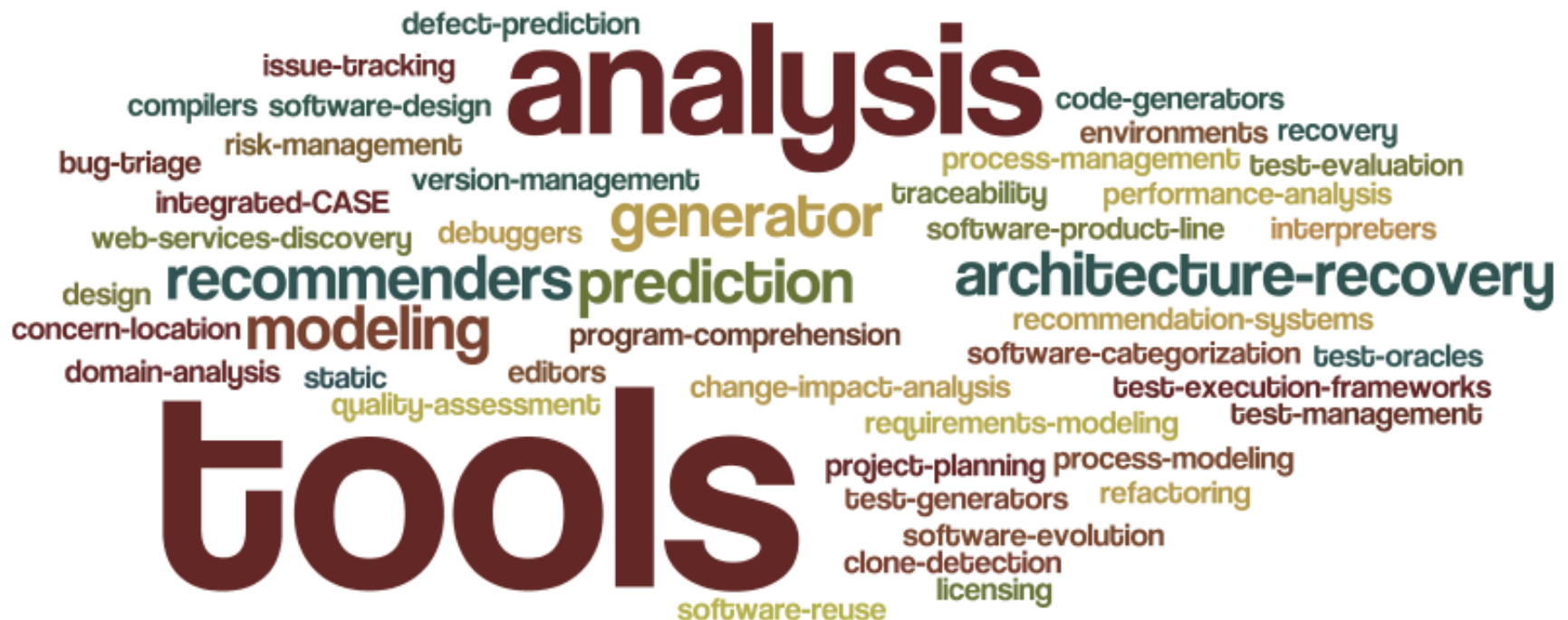
It is increasingly complex under the hood.

# Software is like a car.



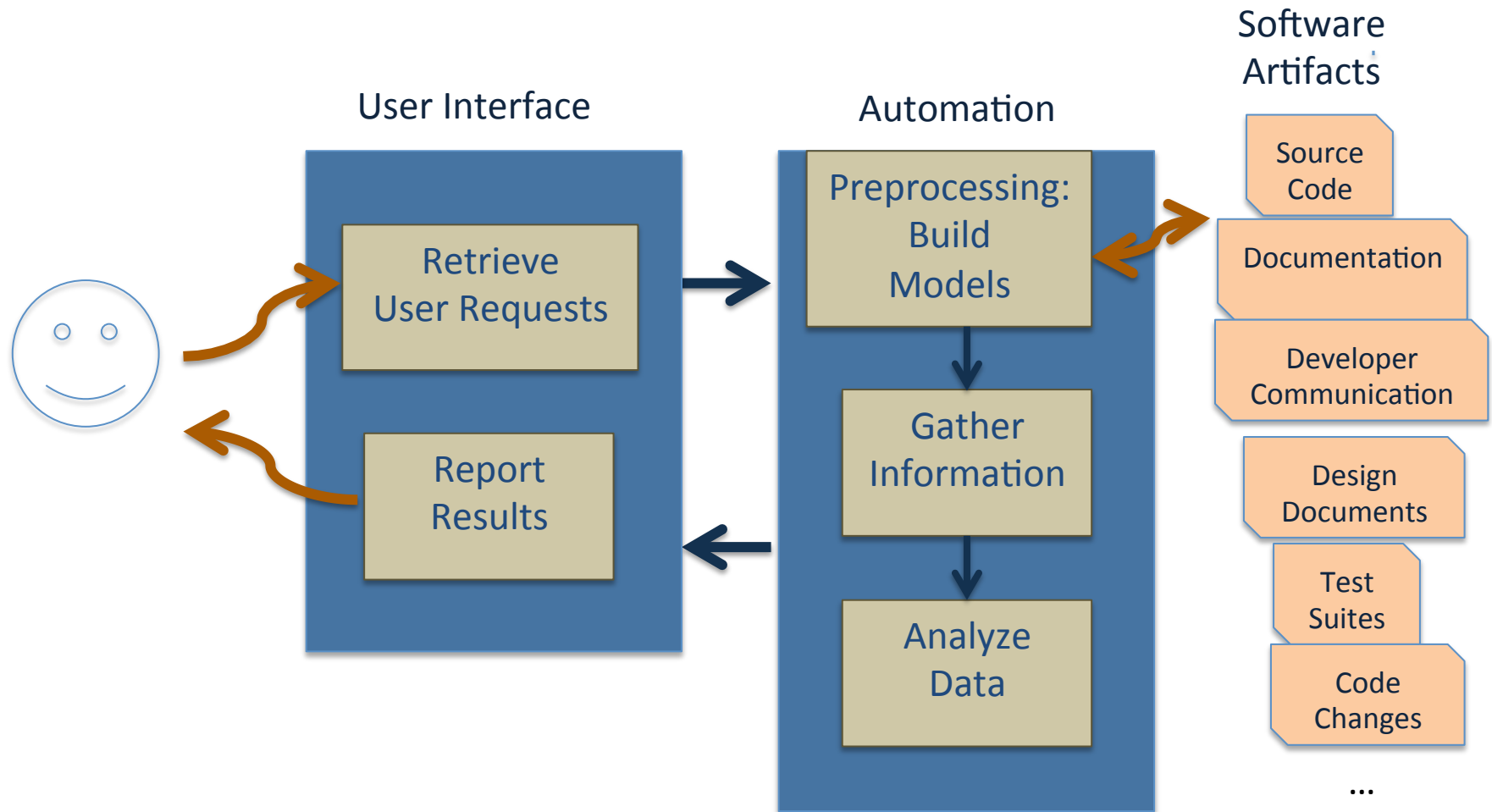
It now requires specialized tools to maintain.

# SE community to the rescue



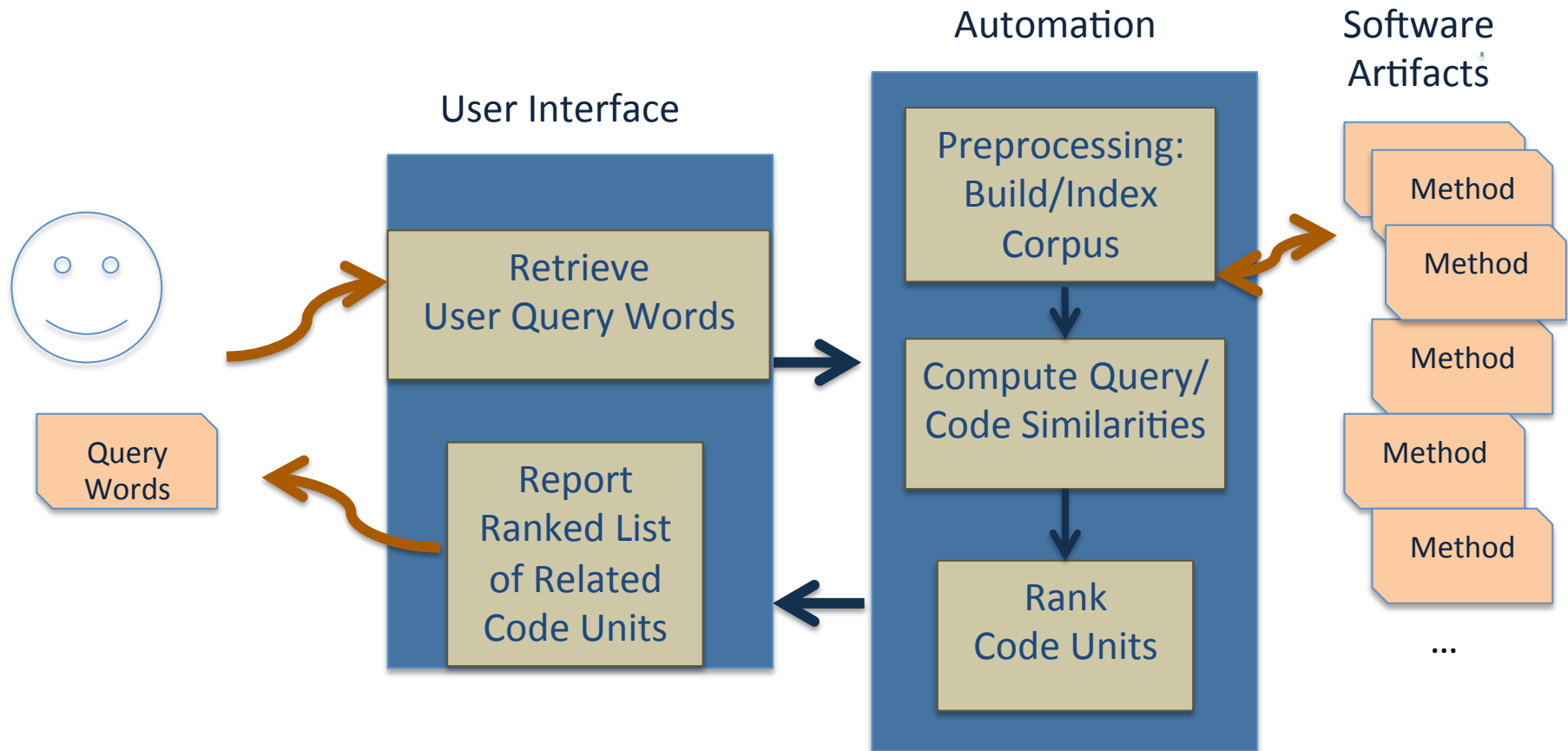


# Power Tools



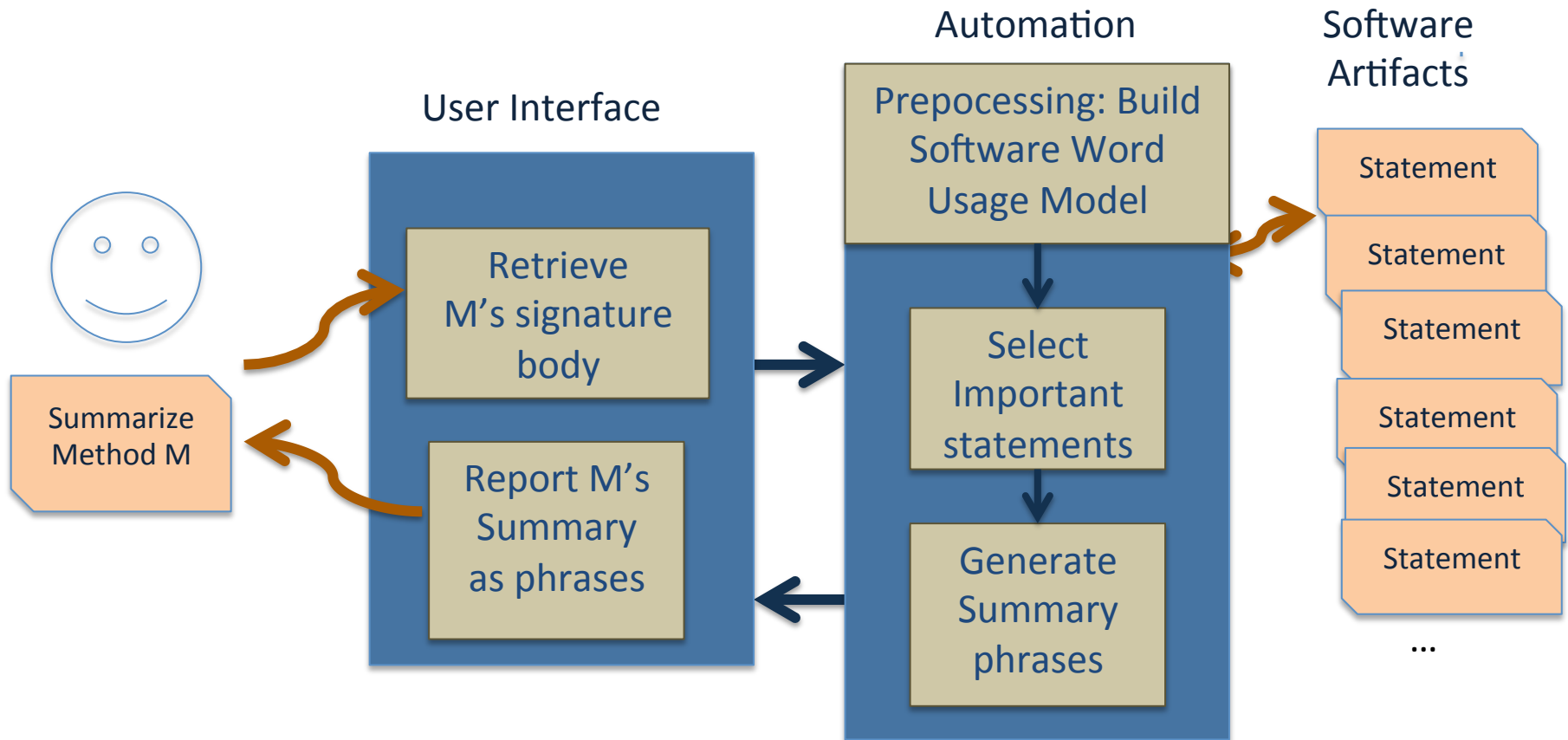


# Example: Code Search Tool





# Example: Method Summarization Tool

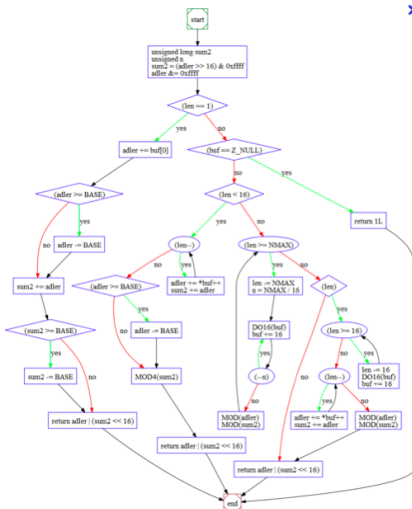




# Power Tools:

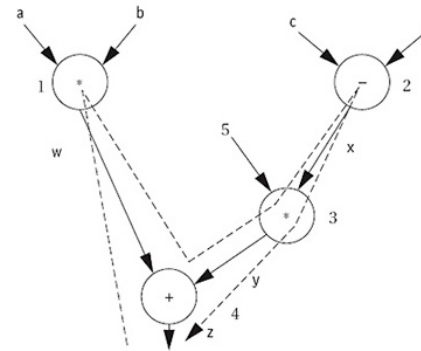
## What Information is used?

### Control Flow

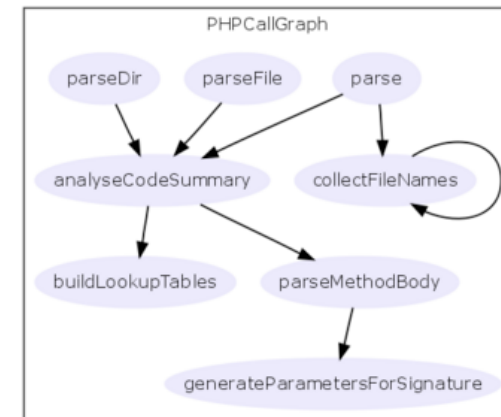


### Structural:

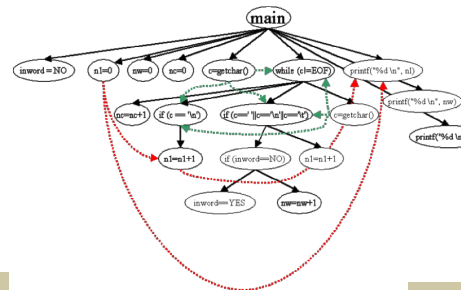
### Data Flow



### Call Graphs



### Program Dependence



- Constants
- Types
- Inheritance

- Dynamic:
  - Frequency/order of execution
- Development Process-related:
  - Change logs, bug reports

# What else is available?

## Consider this code

```
public static int a(int c, int d) {  
    int b;  
    b = c * d;  
    return b;  
}
```

Compute and return a product

```
public static int c(int w, int h) {  
    int a;  
    a = w * h;  
    return a;  
}
```

Compute the area of a rectangle?

```
public static int computeArea(int width, int height) {  
    int area;  
    area = width * height;  
    return area;  
}
```

Given a width and height, compute & return the area of a rectangle, OBVIOUSLY.



# Where is Natural Language in Software?

```
class Player{  
    /**  
     * Play a specified file with specified time interval  
     */  
    public static boolean play(final File file, final float fPosition  
                               final long length) {  
        fCurrent = file;  
        try {  
            playerImpl = null;  
            // make sure to stop non-fading players  
            stop(false);  
            // Choose the player  
            Class cPlayer = file.getTrack().getType().getPlayerImpl();  
            ...  
        }  
    }  
}
```

Class names

Method comments

Method names

Parameter names

Other variables

Internal comments

# How can we leverage the naming?

```
class Player{  
  public static boolean play(final File file, final float fPosition, final long length) {  
    fCurrent = file;  
    try {  
      playerImpl = null;  
      stop(false);  
      class cPlayer = file.getTrack().getType().getPlayerImpl();  
      ...}  
    }  
  }  
}
```

Code Search

Traceability

Code Navigation

Refactoring

Marcus et al. study of  
literature revealed  
25 different SE tasks!

# Many Uses of Text Analysis

Traceability links recovered and maintenance among software engineering artifacts	<i>66 papers</i>
Concept, feature or concern location and aspect mining in source code	<i>50 papers</i>
Change impact analysis in source code	<i>8 papers</i>
Restructuring and refactoring	<i>13 papers</i>
Software reuse	<i>19 papers</i>
Architecture/design recovery	<i>4 papers</i>
Quality assessment and software measurement	<i>21 papers</i>
Defect Prediction	<i>2 papers</i>
Recommending developers	<i>4 papers</i>
Discovery of web services	<i>3 papers</i>
Licensing	<i>4 papers</i>
Requirement Analysis/Engineering	<i>9 papers</i>
Clone detection	<i>1 papers</i>
Program comprehension general	<i>8 papers</i>
Bug triage	<i>8 papers</i>
Software Evolution Analysis	<i>3 papers</i>
Software Categorization	<i>4 papers</i>
Domain Analysis/Software Product Lines	<i>1 papers</i>
Other tasks	<i>3 papers</i>
Software miniaturization	<i>1 papers</i>

Marcus et al.

# So, what is Text Analysis?

*analysis of the natural language used by  
programmers in writing software  
(source code + other software artifacts)*

## Why?

To provide important information  
for building automated and semi-automated  
recommendation systems and analysis tools  
to support SE tasks

# Flavors of Text Analysis

## Information/Text Retrieval (IR/TR)

Given query words, retrieve documents containing unstructured data related to those topics:

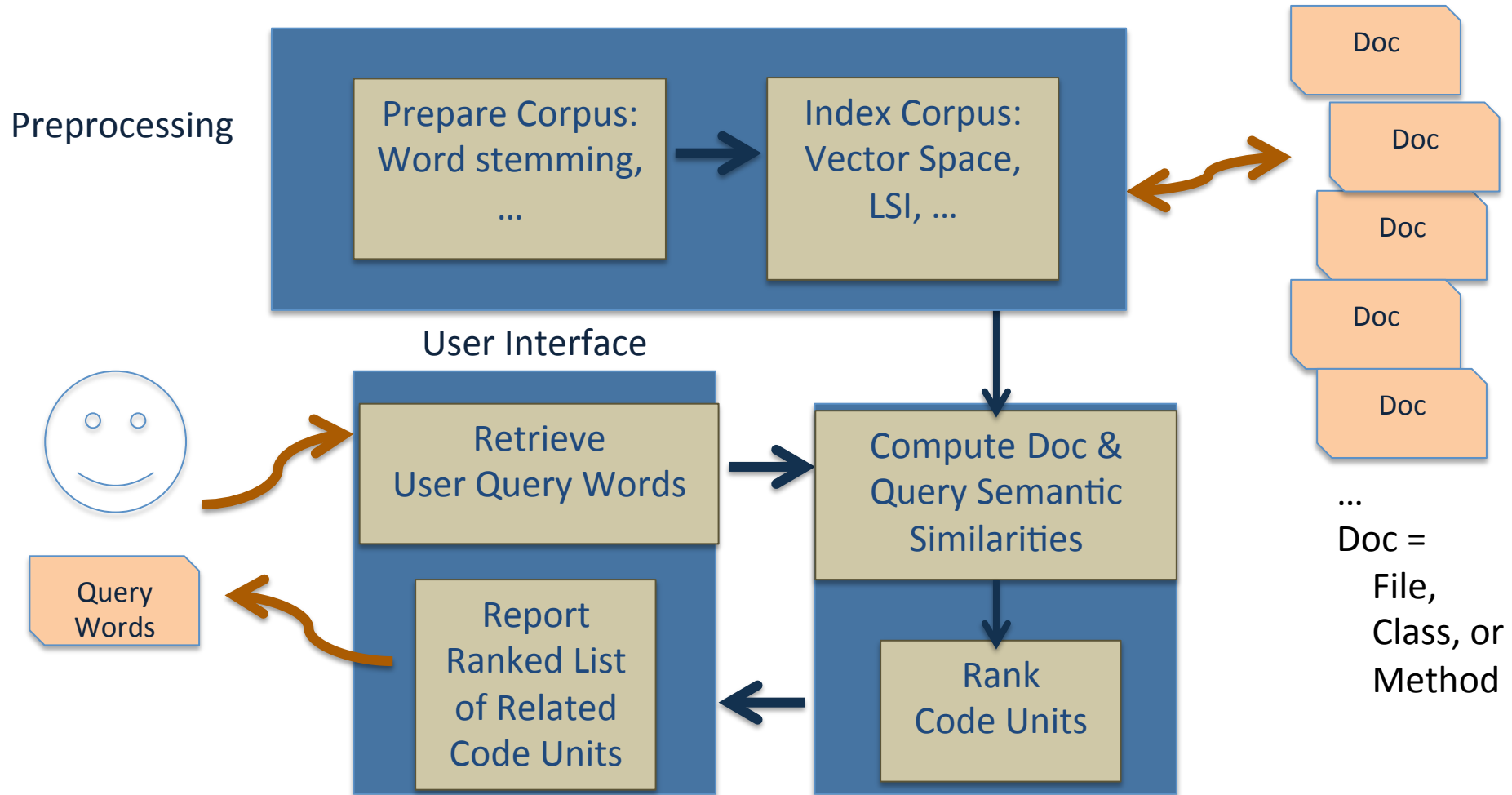
- \* For a known information need, return as many relevant docs as possible
  - \* To enable the user to explore a problem domain
- 

## Natural Language Processing (NLP)

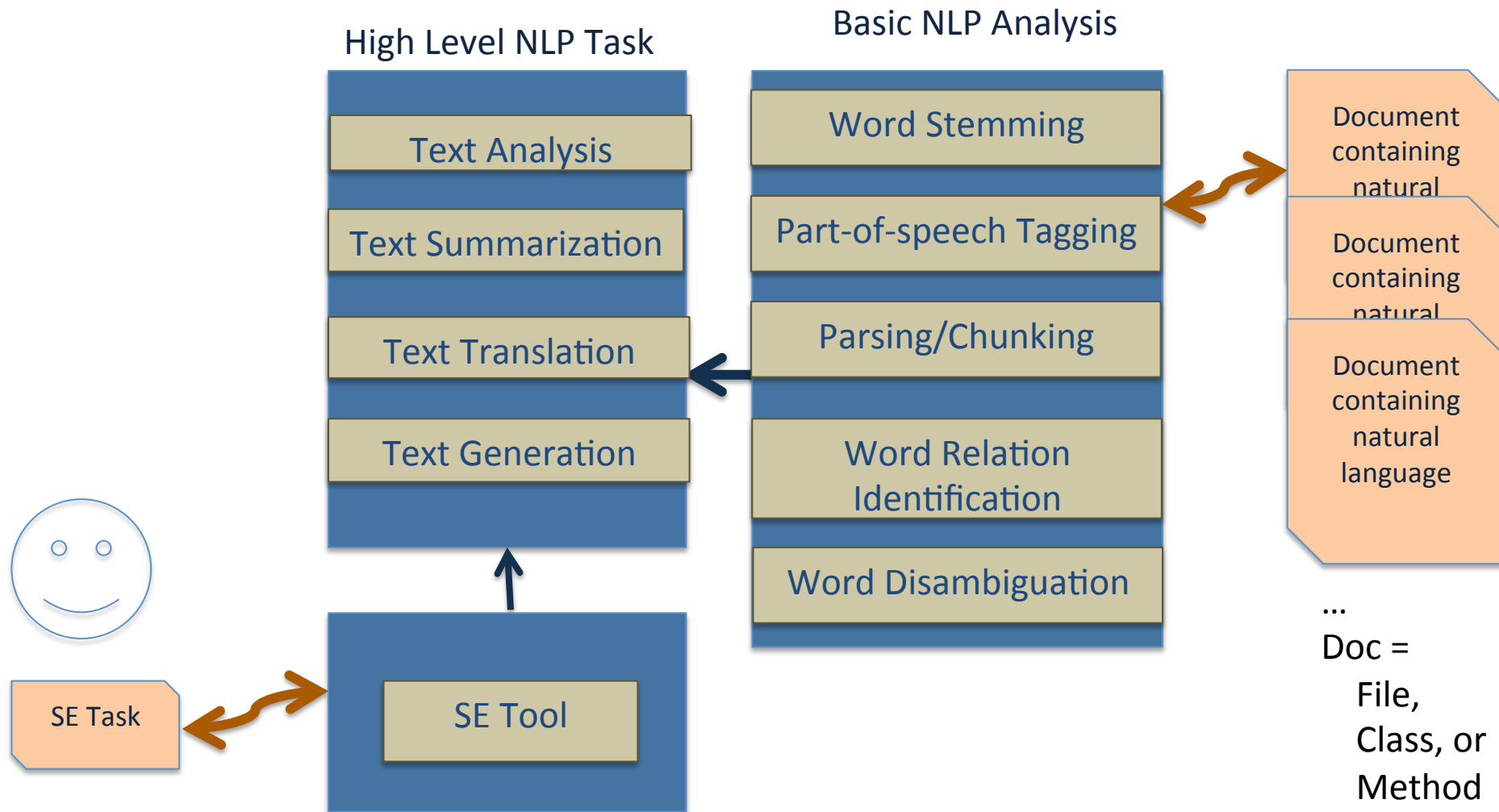
Software that will automatically analyze, understand, and generate languages that humans use naturally (e.g., English)

- \* To know what concepts a word or phrase represents
- \* To know how to link those concepts together in a meaningful way

# Text Retrieval: Overview



# NLP: Overview



# NLP in SE: An Example

1. Split Name into Words
2. Part-of-speech tag method name
3. Chunk method name
4. Identify Verb and Direct-Object (DO)

get User List From File

Split Id

```
public UserList getUserListFromFile( String path ) throws IOException {
```

Tag POS

get <verb> User <adj> List <noun> From <prep> File <noun>

Chunk

get <verb phrase> User List <noun phrase> From File <prep phrase>

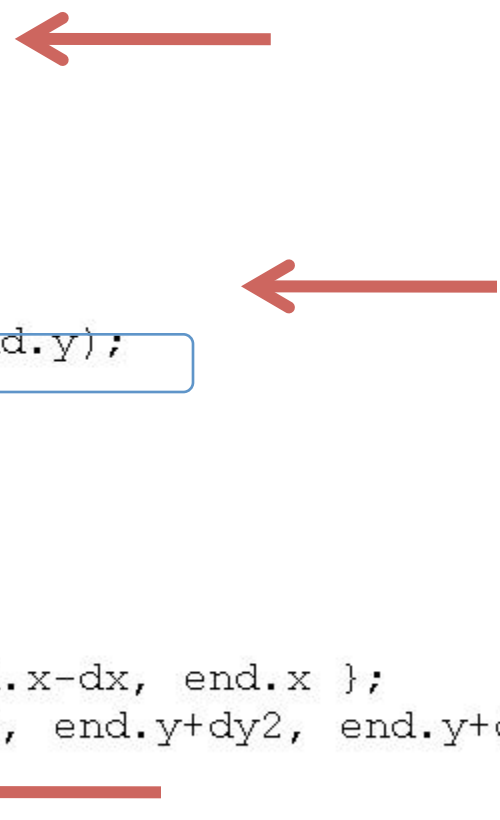
```
throw new IOException( "UserList format issue" + path + " file " + e );
```



## Example Client Tool - UD-Summarize

*/\* Update linear edge view. If width  $\leq 1$ , draw line to given graphics2d,  
else draw polyline to graphics2d \*/*

```
public void paint(Graphics2D g) {  
    update();  
    g.setColor(getColor());  
    g.setStroke(getStroke());  
    setRendering(g);  
    int w=getWidth();  
    if (w<=1) {  
        g.drawLine(start.x,start.y,end.x,end.y);  
    }  
    else {  
        int dx=w/3+1;  
        if(target.isLeft()) dx=-dx;  
        int dyl=getSourceShift();  
        int dy2=getTargetShift();  
        int xs[]={ start.x, start.x+dx, end.x-dx, end.x };  
        int ys[]={ start.y+dyl, start.y+dyl, end.y+dy2, end.y+dy2 };  
        g.drawPolyline(xs,ys,4);  
    }  
    super.paint(g);  
}
```



# Going Forward with Text Analysis

- \* **Apply** text analysis to
  - develop new tools and improve tools
- \* **Combine** information
  - Structure + Text + Dynamic
- **Explore configurations** of analyses
- **Analyze** mixed text documents & link
- \* **Improve Evaluations**
  - Lack of common infrastructure

Source  
Code

Documentation

Developer  
Communication

Design  
Documents

Test  
Suites

Code  
Changes

# Research Opportunities

- If you are intrigued,
    - + inspired to problem solve
    - + ambitious to contribute to research
    - + have background in software engineering
    - + like to think creatively
    - + excited and committed to learn how to conduct high quality research
- THEN Email me. [pollock@cis.udel.edu](mailto:pollock@cis.udel.edu).