An Empirical Study on Requirements Traceability Using Eye-Tracking

ICSM 2012, Riva Del Garda, Italy

Nasir Ali, Zohreh Sharafi, Yann-Gaël Guéhéneuc, and Giuliano Antoniol

... featuring Bram Adams 😊
Requirements Traceability

Requirements traceability is defined as “the ability to describe and follow the life of a requirement, in both a forwards and backwards direction” [Gotel, 1994]

Program comprehension  Code location of new requirement
What’s Requirements Traceability Good For?

• Ease program comprehension

• Discover what code must change to handle a new requirement

• Help in determining whether a specification is completely implemented
Linking Requirements with Source Code

Requirements

RSS option for an instant messenger

SOURCE CODE

20% of similarity
IR-based Approaches

- Vector Space Model [Antoniol et al. 2002]
- Latent Semantic Indexing [Marcus and Maletic, 2003]
- Jensen Shannon Divergence [Abadi et al. 2008]
- Latent Dirichlet Allocation [Asuncion, 2010]
Linking Requirements with Source Code

```java
public class DndUtils {

    /**
     * Indicates if it is safe to access the system clipboard. Once false,
     * access will never be checked again.
     */
    private static boolean canAccessSystemClipboard = true;

    /**
     * Key used in app context to lookup.
     */
    private static Object clipboardKey = new Object();

    static boolean isLinux = System.getProperty("os.name").startsWith("Linux");

    /**
     * Returns true if this set of DataFlavors might include a FileFlavor.
     */
    // ...
    
    20% of similarity

    RSS option for an messenger

   _requirements

    ICSM 2012

    SOURCE CODE
```
How IR Techniques see Source Code: TF/IDF

```
public class SendEmail {
    SendEmail () {
        ..................
        ..................
    }

    public sendOutEmail(String emailId) {
        EmailAddressFormatChecker emailAddressFormat =
        new EmailAddressFormatChecker();
        int status;

        if(emailAddressFormatChecker.verify(emailId)) {
            ..................
            ..................
        }
    }
}
```

\[ TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \]

**Term Frequency**: importance of term in a document

\[ IDF_i = \log_2 \left( \frac{|D|}{d : |t_i \in d|} \right) \]

**Inverse Document Frequency**: rareness of term across all the documents

**weight of term**
public class SendEmail {

    SendEmail (){  
        ........................
        ........................
    }

    public sendOutEmail(String emailId) {

        EmailAddressFormatChecker emailAddressFormatChecker = new EmailAddressFormatChecker();
        int status;

        if(emailAddressFormatChecker.verify(emailId)){
            ........................
            ........................
        }

    }
}

Developer

What the Developer Really Saw

Class, Method, Variables, Comments

ICSM 2012
Our Conjecture

Understanding how developers verify RT links can allow developing an improved IR-based technique to recover RT links with better accuracy than previous techniques.
Part I: Eye-tracking Study
RQ1: What Source Code Entities (SCEs) do Developers Value the Most?

<table>
<thead>
<tr>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Class Name</td>
</tr>
<tr>
<td>- Method Name</td>
</tr>
<tr>
<td>- Variable Name</td>
</tr>
<tr>
<td>- Comments</td>
</tr>
</tbody>
</table>
Hypotheses for RQ1

• \( H_{01} \) – There is no difference between the importance of SCEs for developers.

• \( H_{02} \) – There is no difference between the importance of domain- and implementation-related SCEs for developers.
Statistical Tests

• **Wilcoxon Rank Sum**
  – It is defined as how many times a Y precedes X in the samples. It is a nonparametric test and an alternative to the two-sample t-test.

• **Kruskual-Wallis**
  – Kruskal-Wallis rank sum test is a nonparametric method for testing the equality of the population medians among different groups.
Approach: Eye-tracking Experiment

This class calculates the area of a circle based on the runtime radius of a circle input.

```java
public class CalculateArea {
    public void calculateCircleArea() {
        int radius = 0;
        System.out.println("Please enter radius of a circle");
        try {
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            radius = Integer.parseInt(br.readLine());
        } catch (NumberFormatException ne) {
            System.out.println("Invalid radius value" + ne);
            System.exit(0);
        } catch (IOException ioe) {
            System.out.println("IO Error :" + ioe);
            System.exit(0);
        }
        double area = Math.PI * radius * radius;
        System.out.println("Area of a circle :" + area);
    }
}
```

eye fixation ==
developer importance
Such Information is Relatively Easy to Obtain!

- Facelab by Seeing Machine
  - Built-in cameras
  - Infrared pad
  - Monitor Screen
# Case Study Setup

## Subject Statistics

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Subjects</strong></td>
<td>26</td>
</tr>
<tr>
<td><strong>Ph.D.</strong></td>
<td>22 (1 rejected: pilot study)</td>
</tr>
<tr>
<td><strong>M.Sc.</strong></td>
<td>4 (1 rejected: glasses)</td>
</tr>
<tr>
<td><strong>Java Experience</strong></td>
<td>On Average 3.39 Years</td>
</tr>
<tr>
<td><strong>Traceability Experience</strong></td>
<td>90% subjects</td>
</tr>
</tbody>
</table>

## Source Code Statistics

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming Language</strong></td>
<td>Java</td>
</tr>
<tr>
<td><strong>Total Source Code Snippet</strong></td>
<td>6</td>
</tr>
<tr>
<td><strong>Total LOC</strong></td>
<td>19, 18, 19, 18, 24, 28</td>
</tr>
</tbody>
</table>
Example Task

```java
// This class calculates the area of a circle based on runtime radius of a circle input

public class CalculateArea {

    public void CalculateCircleArea() {

        int radius = 0;
        System.out.println("Please enter radius of a circle");

        try {
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            radius = Integer.parseInt(br.readLine());
        } catch (NumberFormatException ne) {
            System.out.println("Invalid radius value + ne");
            System.exit(0); 
        } catch (IOException ioe) {
            System.out.println("IO Error + ioe");
            System.exit(0);
        }

        double area = Math.PI * radius * radius;
        System.out.println("Area of a circle is + area");
    }

}
```

This class takes as input the radius of a circle to calculate its area.
Post-experiment Question

“Please rank the source code entities that help you more to comprehend source code and verify a traceability link.”

<table>
<thead>
<tr>
<th>Source Code Entities</th>
<th>Preference Ranks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Name(s)</td>
<td></td>
</tr>
<tr>
<td>Method Name(s)</td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td></td>
</tr>
<tr>
<td>Variable Name(s)</td>
<td></td>
</tr>
</tbody>
</table>

ICSM 2012
Post-experiment Questions (1)

• **Question1**: Was the source code easy to read?
  – **Yes** – Easy to read
  – **No** – Source code font was too small and hard to read

• **Question2**: Were the source code identifiers easy to understand?
  – **Yes**
  – **No**
Post-experiment Questions (1)

• **Question1**: Was the source code easy to read?
  – **Yes** – Easy to read [unanimously]

• **Question2**: Were the source code identifiers easy to understand?
  – **Yes** [unanimously]
Post-experiment Questions (1)

• **Question1**: Was the source code easy to read?
  – **Yes** – Easy to read (**All the subjects answered in Yes**)

• **Question2**: Were the source code identifiers easy to understand?
  – **Yes** (**All the subjects answered in Yes**)
Eye-tracking Experiment Results
### Eye-tracking Experiment Results

<table>
<thead>
<tr>
<th>Source Code Entities</th>
<th>Average Fixation Time (ms) of All Subjects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method Name(s)</td>
<td>5701.10</td>
</tr>
<tr>
<td>Comments</td>
<td>4542.41</td>
</tr>
<tr>
<td>Variable Name(s)</td>
<td>3181.81</td>
</tr>
<tr>
<td>Class Name(s)</td>
<td>2317.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Domain (48% of all terms)</th>
<th>Average Fixation Time (ms) of All Subjects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation (52% of all terms)</td>
<td>1729.80</td>
</tr>
</tbody>
</table>

**p-Value <0.01**
**Post-experiment Question**

“Please rank the source code entities that help you more to comprehend source code and verify a traceability link.”

<table>
<thead>
<tr>
<th>Source Code Entities</th>
<th>Average Rank of All Subjects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method Name(s)</td>
<td>1.44</td>
</tr>
<tr>
<td>Comments</td>
<td>2.00</td>
</tr>
<tr>
<td>Variable Name(s)</td>
<td>2.61</td>
</tr>
<tr>
<td>Class Name(s)</td>
<td>3.87</td>
</tr>
</tbody>
</table>
RQ1: What Source Code Entities (SCEs) do Developers Value the Most?

<table>
<thead>
<tr>
<th>Physical</th>
<th>Conceptual</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Method Name</td>
<td>1. Domain Related Terms</td>
</tr>
<tr>
<td>2. Comments</td>
<td>2. Application Related Terms</td>
</tr>
<tr>
<td>3. Variable Name</td>
<td></td>
</tr>
<tr>
<td>4. Class Name</td>
<td></td>
</tr>
</tbody>
</table>
Part II: Improved Retrieval Technique
RQ2: Can we Improve IR Techniques by making them Aware of the Developers’ Interests?

```java
public class SendEmail {

    SendEmail (){}
        .................
        .................

    public sendOutEmail(String emailId) { 

        EmailAddressFormatChecker emailAddressFormatChecker = new EmailAddressFormatChecker();
        int status;

        if(emailAddressFormatChecker.verify(emailId)){
            .................
            .................
        }

    }

}
```

1. Method
2. Comment
3. Variables
4. Class
Alternative Weighting Scheme 1: SE/IDF

\[ scet_{fi,j} = \]
Alternative Weighting Scheme 2: DOI/IDF

\[ DOIIF_{i,j} = \{ \]
Research Questions for RQ2

• **RQ2.1** – Does using SE/IDF allow developing a RT links recovery technique with a better accuracy than a technique using TF/IDF?

• **RQ2.2** – Does using DOI/IDF allow developing a RT links recovery technique with a better accuracy than a technique using TF/IDF?
Hypotheses for RQ2

• $H_{01}$ – There is no difference between the accuracy of an LSI-based technique using TF/IDF and using the novel weighting scheme, SE/IDF in terms of F-measure.

• $H_{02}$ – There is no difference between the accuracy of an LSI-based technique using TF/IDF and using the novel weighting scheme, DOI/IDF in terms of F-measure.
Case Study Systems

**iTrust:** Medical application

**Pooka:** An email Client

<table>
<thead>
<tr>
<th></th>
<th>Pooka</th>
<th>iTrust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Language</td>
<td>Java</td>
<td>Java</td>
</tr>
<tr>
<td>Version</td>
<td>2.0</td>
<td>10</td>
</tr>
<tr>
<td>Number of Classes</td>
<td>298</td>
<td>526</td>
</tr>
<tr>
<td>Number of Methods</td>
<td>20,868</td>
<td>3,404</td>
</tr>
<tr>
<td>LOC</td>
<td>244K</td>
<td>19,604</td>
</tr>
</tbody>
</table>

ICSM 2012
IR-based Traceability Link Recovery

- **LSI** (Latent Semantic Indexing)
- Processed corpus is transformed into a term-by-document matrix
- The values of the matrix cells represent the weights of the terms in the documents
- Use **TF/IDF**, **SE/IDF**, and **DOI/IDF** weights to link requirements and source code

ICSM 2012
IR Quality Measures

\[
\text{Precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}
\]

\[
\text{Recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}
\]

\[
F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]
Document Preprocessing

• Extract all the source code and requirements terms
• Store the mapping of each source code term

• Filter (#43@$)
• Stop words (the, is, an….)
• Stemmer (attachment -> attach)
Separate Domain and Implementation Terms

- LDA (Latent Dirichlet Allocation)
  - LDA Parameter Settings \(^1\)
    - \(\alpha = 25 \ (50/K)\)
    - \(\beta = 0.1\)
    - \(K = 2\)
- Mallet LDA Implementation

\(^1\) S. Abebe and P. Tonella, “Towards the extraction of domain concepts from the identifiers,” in 18th Working Conference on Reverse Engineering (WCRE), 2011, pp. 77 –86.
**SE/IDF and DOI/IDF Parameter Settings**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
<th>SCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>0.1</td>
<td>Class</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.4</td>
<td>Method</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.2</td>
<td>Variable</td>
</tr>
<tr>
<td>$\delta$</td>
<td>0.3</td>
<td>Comments</td>
</tr>
<tr>
<td>$\gamma'$</td>
<td>0.74</td>
<td>Domain</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>0.26</td>
<td>Implementation</td>
</tr>
<tr>
<td>$\psi$</td>
<td>$\delta/2$</td>
<td>Only in Requirements</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>2</td>
<td>Terms in Source Code and Requirements</td>
</tr>
</tbody>
</table>

*Weighted according to developers’ interest*
Pooka F1 Results: DOI/IDF > SE/IDF > TF/IDF
iTrust F1 Results: DOI/IDF > SE/IDF > TF/IDF
## Average Results across All Thresholds

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Avg. Precision</th>
<th>Avg. Recall</th>
<th>P-Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pooka</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSI (TF/IDF)</td>
<td>14.71</td>
<td>21.07</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>LSI (SE/IDF)</td>
<td>18.30</td>
<td>24.86</td>
<td></td>
</tr>
<tr>
<td>LSI (DOI/IDF)</td>
<td>25.73</td>
<td>26.42</td>
<td></td>
</tr>
<tr>
<td><strong>iTrust</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSI (TF/IDF)</td>
<td>36.43</td>
<td>33.14</td>
<td>&lt; 0.01</td>
</tr>
<tr>
<td>LSI (SE/IDF)</td>
<td>38.66</td>
<td>33.97</td>
<td></td>
</tr>
<tr>
<td>LSI (DOI/IDF)</td>
<td>39.55</td>
<td>35.07</td>
<td></td>
</tr>
</tbody>
</table>
Research Questions for RQ2

• **RQ2.1** – Does using SE/IDF allow developing a RT links recovery technique with a better accuracy than a technique using TF/IDF?

• **RQ2.2** – Does using DOI/IDF allow developing a RT links recovery technique with a better accuracy than a technique using TF/IDF?
Hypotheses for RQ2

- \( H_{01} \) – There is no difference between the accuracy of an LSI-based technique using TF/IDF and using the novel weighting scheme, SE/IDF in term of F-measure
  
  REJECT

- \( H_{02} \) – There is no difference between the accuracy of an LSI-based technique using TF/IDF and using the novel weighting scheme, DOI/IDF in term of F-measure
  
  REJECT
Summary

• Integrating the developers’ knowledge, i.e., SCEs preferences, in an IR technique statistically improves its accuracy

• Different terms should be weighted according to their positions in the source code
**What the Developer Really Saw**

```java
public class SendEmail {
    SendEmail () {
        
    }
    public sendOutEmail(String emailId) {
        EmailAddressFormChecker emailAddressFormChecker = new EmailAddressFormChecker();
        int status = 0;
        if(emailAddressFormChecker.verify(emailId)) {
            
        }
    }
}
```  

**Approach: Eye-tracking Experiment**

**RQ1: What Source Code Entities (SCEs) Value Developers the Most?**

<table>
<thead>
<tr>
<th>Physical</th>
<th>Conceptual</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Method Name</td>
<td>1. Domain Related Terms</td>
</tr>
<tr>
<td>2. Comments</td>
<td>2. Application Related Terms</td>
</tr>
<tr>
<td>3. Variable Name</td>
<td></td>
</tr>
<tr>
<td>4. Class Name</td>
<td></td>
</tr>
</tbody>
</table>

**Hypotheses for RQ2**

- **H₀**: There is no difference between the accuracy of an LSI-based technique using TF/IDF and using the novel weigh: nlg scheme, SE/IDF in term of F measure
  
  **REJECT**

- **H₀**: There is no difference between the accuracy of an LSI-based technique using TF/IDF and using the novel weigh: nlg scheme, DOI/IDF in term of F measure
  
  **REJECT**
Please forward your detailed questions to this guy

nasir.ali@polymtl.ca
RQ1: What Source Code Entities (SCEs) Value Developers the Most?

<table>
<thead>
<tr>
<th>Physical</th>
<th>Conceptual</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Method Name</td>
<td>1. Domain Related Terms</td>
</tr>
<tr>
<td>2. Comments</td>
<td>2. Application Related Terms</td>
</tr>
<tr>
<td>3. Variable Name</td>
<td></td>
</tr>
<tr>
<td>4. Class Name</td>
<td></td>
</tr>
</tbody>
</table>

Hypotheses for RQ2:

- **H₀₁**: There is no difference between the accuracy of an LSI-based technique using TF-IDF and using the novel weigh: ng|scheme, SE/IDF in term of fX measure
  - **REJECT**

- **H₀₂**: There is no difference between the accuracy of an LSI-based technique using TF-IDF and using the novel weigh: ng|scheme, DOI/IDF in term of fX measure
  - **REJECT**