Developers' Information Needs on Collaborative Platforms

Master Thesis of Mathias Birrer
Supervised by Pooja Rani

Final Presentation
Developers' Information Needs

Internal Sources

Planning  Implementation  Testing  Releasing  Maintenance

External Sources
External Sources of Information

Sources: *Mailing Lists, Q&A Sites, Bug Trackers, News Sites, ...*

- Unstructured data
- Fragmented developer workflow
- Diversity

Researchers started investigating these sources

- Compare sources / communities
- Uncover evolution
- Reuse datasets

Reproducibility is highly important
Researchers workflow

- Planning
- Implementation
- Testing
- Releasing
- Maintenance
Background: Which data sources are typically analyzed by researchers to understand developers’ information needs?
RQ1: How do researchers analyze developers' information needs on collaborative platforms?
RQ2 (Case Study): What are developers' questions about code comment conventions?
Methodology RQ1

Literature Selection
- 42 papers

Extraction
- Data Source
- Data Extraction
  - Preprocessing
  - Dataset availability
  - Analysis

Analysis
- Background Information
- Aspects of Reproducibility
- Reproducibility
Results RQ1 – Source Categories
Source Categories with Extraction Method

- Q&A
- User Reviews
- Mailinglist
- Newsgroup
- Bug Report

Categories with Extraction Method:

- Data Dump
- API
- Query Tool
- Web Scraping
- Manual
- N/S
Data Preprocessing

- Remove source code
- Remove HTML tags
- Remove Punctuation
- Remove Non-Alpha-Numeric

- Remove Stop words
- Apply Word stemming
- Apply Lemmatization
- Filter Non-English
- Case unification
Common methodology
Reproducibility Aspects

Ratio of Paper considered reproducible

- Dataset availability
- Extraction
- Preprocessing
- Extraction and Preprocessing
Tool Usage for Preprocessing

- No Preprocessing: 31%
- Preprocessing: 69%

---

Tool Usage Preprocessing

- Yes: 34%
- No: 66%

Tool Usage Analysis

- Yes: 31%
- No: 69%
Tools for Preprocessing

• Common Tools
  *Porter Stemmer, Stanford Parser, Python NLTK, Apache OpenNLP*

• Tools for NLP

• **No tool to manage or automate preprocessing workflow**
RQ2 – Case Study

What Developers discuss about “Code Comment Conventions” on Social Media
Code Comment Conventions

• Trustworthy form of documentation
• Basis for documentation tools
• Style & Syntax cannot be enforced
• Conventions for Languages, Companies, Projects, Developers
• Confusion amongst developers
## LDA Topics

<table>
<thead>
<tr>
<th>#</th>
<th>Topic Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Exceptions</td>
</tr>
<tr>
<td>1</td>
<td>Documentation Generation</td>
</tr>
<tr>
<td>2</td>
<td>IDE &amp; Editors</td>
</tr>
<tr>
<td>3</td>
<td>Processing Code Comments</td>
</tr>
<tr>
<td>4</td>
<td>Testing &amp; Naming Conventions</td>
</tr>
<tr>
<td>5</td>
<td>Project Documentation</td>
</tr>
<tr>
<td>6</td>
<td>Project Naming Conventions</td>
</tr>
<tr>
<td>7</td>
<td>Code Entities Naming Conventions</td>
</tr>
<tr>
<td>8</td>
<td>Comments Writing Strategies</td>
</tr>
<tr>
<td>9</td>
<td>Comment Functionality Websites</td>
</tr>
<tr>
<td>10</td>
<td>Comment Framework</td>
</tr>
<tr>
<td>11</td>
<td>Database</td>
</tr>
<tr>
<td>12</td>
<td>Code Conventions</td>
</tr>
<tr>
<td>13</td>
<td>Comments Syntax</td>
</tr>
</tbody>
</table>
# LDA Topics

<table>
<thead>
<tr>
<th>#</th>
<th>Topic Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Exceptions</td>
</tr>
<tr>
<td>1</td>
<td>Documentation Generation</td>
</tr>
<tr>
<td>2</td>
<td>IDE &amp; Editors</td>
</tr>
<tr>
<td>3</td>
<td>Processing Code Comments</td>
</tr>
<tr>
<td>4</td>
<td>Testing &amp; Naming Conventions</td>
</tr>
<tr>
<td>5</td>
<td>Project Documentation</td>
</tr>
<tr>
<td>6</td>
<td>Project Naming Conventions</td>
</tr>
<tr>
<td>7</td>
<td>Code Entities Naming Conventions</td>
</tr>
<tr>
<td>8</td>
<td>Comments Writing Strategies</td>
</tr>
<tr>
<td>9</td>
<td>Comment Functionality Websites</td>
</tr>
<tr>
<td>10</td>
<td>Comment Framework</td>
</tr>
<tr>
<td>11</td>
<td>Database</td>
</tr>
<tr>
<td>12</td>
<td>Code Conventions</td>
</tr>
<tr>
<td>13</td>
<td>Comments Syntax</td>
</tr>
</tbody>
</table>
LDA Topics – Tag Distribution
Problems with tags

- Tags *comments* and *commenting*
- General and ambiguous
- Irrelevant despite large proportion of tags
LDA Technical Details

• MALLET
• Topics $k = 14$
• Hyperparameters
  • $\alpha = 5$
  • $\beta = 0.01$
Code Comment Conventions

Commenting

High-level Code Versioning
Grammar Rules
Syntax & Format
Maintain Comments
Other

Language
Syntax & Format
Using Feature
Asking for Feature
Process Comments
Change comment template
Asking tool existence
Understand Documentation

Tool
Syntax & Format
Using Feature
Asking for Feature
Process Comments
Change comment template
Setup
Error
Report Bug

IDE & Editor
Syntax & Format
Using Feature
Asking for Feature
Process Comments
Change comment template
Tool setup
Shortcut

Other

Manual Analysis
Question Type on Quora vs. Stack Overflow

- Implementation Problems
- Error
- Limitation and Possibilities
- Implementation Strategies
- Best Practice
- Background Information
- Opinion

Stack Overflow vs. Quora: Comparison of question types.

Stack Overflow
- Implementation Problems: 100%
- Error: 90%
- Limitation and Possibilities: 80%
- Implementation Strategies: 70%
- Best Practice: 60%
- Background Information: 50%
- Opinion: 40%

Quora
- Implementation Problems: 90%
- Error: 80%
- Limitation and Possibilities: 70%
- Implementation Strategies: 60%
- Best Practice: 50%
- Background Information: 40%
- Opinion: 30%
Taxonomy – Most discussed features
What happened with Mailing Lists?

• No relevant data found regarding commenting practices
• Despite previous study on documentation issues
Code Comments Conventions – Challenges

- **Generality** and **ambiguity** of topic keywords
- Selection of **relevant tags**
- Selection of **relevant posts**
- **Conclusion**: Very hard to fully automate extraction and classification of “clean” dataset about *Code Comment Conventions*
Makar – Data Management Tool

Data Source Adapters (SO, ML, Github, CSV, JSON)

Transformations (Preprocessing)

Management (Collections, Filters, Views)

Export Adapters (CSV, JSON, Mallet)

User-defined Data Models
Makar - Functionality

• User-Defined Data Models
• Import Adapters (CSV, JSON, Stack Overflow, Github)
• Transformations for Preprocessing
  • Common preprocessing steps built-in
  • Extensible
• Data Management
  • Collections
  • Search Interface
• Export Adapters (CSV, JSON, Mallet)
Main Contributions

- Characteristics and challenges of external data sources
- Analysis of reproducibility aspects
- Multi-source study
- Developer questions about *Code Comment Conventions*
- Development of prototype tool *Makar*
Future Work

Research on Developers’ Information Needs
• Focus on multi-source studies
• Ease of research workflow

Code Comment Conventions
• Investigate more sources (e.g. Github, Jira)

Prototype Tool
• Extend data source adapters
• Analysis and visualization components
• Evaluation against similar tools
Summary

Researchers workflow

Source Categories with Extraction Method

Tool Usage for Preprocessing

LDA Topics

Manual Analysis

Makar – Data Management Tool