Analysis of developer expertise of APIs

Hakan Aksu

Ralf Lämmel
Software Languages Team
University of Koblenz-Landau
Preamble

• MSc thesis
• Initial stage – work in progress
Motivation

• Many software projects in IT companies
• They use:
  – Various languages
  – Various technologies
  – Various problem Domains
  – ...
Motivation

• BUT how does an executive or project manager know which skills a developer has (when hiring or assigning)?
  – Interviews
  – Questionnaires
  – Assignments
  – Publicly available Information
    (e.g. on topcoder or on stackoverflow)
→ „problematic“ methods
Objective

• A new technique to determine the developer skills
  – Leverages previous work experience of developers in a systematic manner
  – We analyze existing evidence for developer expertise based on the version history of existing projects
Milestones (1)

We review related work and best practices of MSR (mining software repositories)


to agree on methods for:

- processing version history

- discovering traceability links between commits, code, and developers.

```java
+ import java.io.FileReader;
+ import java.io.BufferedReader;
... 
  + BufferedReader brTrace;
  + FileReader fReader;
... 
  + fReader = new FileReader(...);
  + brTrace = new BufferedReader(fReader);
  ... 
```
Milestone (2)

We leverage our prior work on API usage analysis and more related work to translate code changes into API usage data.

Prior work:


New Challenge:

API Usage of buildable Java-Projects

Assignment to

Domains

Changed Code in commits
determine

API Usage

Assignment to

Domains
Milestone (3)

We leverage best practices on corpus usage and engineering in MSR to select suitable open-source projects as the corpus to be used in our research.

Challenge: the analysis cannot generally assume all versions to be buildable (resolvable).


Milestone (4)

We identify techniques for

- Summarization and
- Visualization

To derive an

- Understandable and
- Informative

developer profile regarding

- API and
- domain expertise.
Conclusion

• Analysis of ...
  – API usage
  – Change along evolution
  – Developer activity
Conclusion

Sarah

Commit 245

David

Commit 246

... 

import java.io.FileReader; 
import java.io.BufferedReader; 
... 
BufferedReader brTrace; 
FileReader fReader; 
... 
fReader = new FileReader(...); 
brTrace=new BufferedReader(fReader); 
... 
+ import java.awt.*; 
... 
+ GridBagLayout grid = new GridBagLayout(); 
+ Button button; 
... 
+ import javax.swing.*; 
... 
+ JFrame f = new JFrame("Hello World"); 
+ f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); 

API: java.io

Domain: IO

API: AWT & SWING

Domain: GUI
Conclusion

API: java.io
Domain: IO

API: AWT & SWING
Domain: GUI

Change along evolution

Developer activity

Sarah
David

Commit 245
Commit 246

...
References

Questions?