



# Co-Evolving Code-Related and Database-Related Changes in a Data-Intensive Software System

Mathieu **Goeminne**, Tom **Mens**  
Service de Génie Logiciel, Université de Mons

FNRS Projet de Recherche “Data-Intensive Software System Evolution”  
in collaboration with A. Cleve and L. Meurice (Université de Namur)

<http://informatique.umons.ac.be/genlog/projects/disse>

# Context

---

- Focus on data-intensive software systems (DISS)
- Expand empirical MSR research to include database-related activities
- Study co-evolution between code and database
- Carry out empirical studies on open source DISS

# Research Questions

---

- RQ1: Is there any relation between how source code files and database-related files evolve?
- RQ2: What is the effect of migrating to new database technology?
- RQ3: How do developers divide their work and how does this evolve over time?

# Case Study: OSCAR

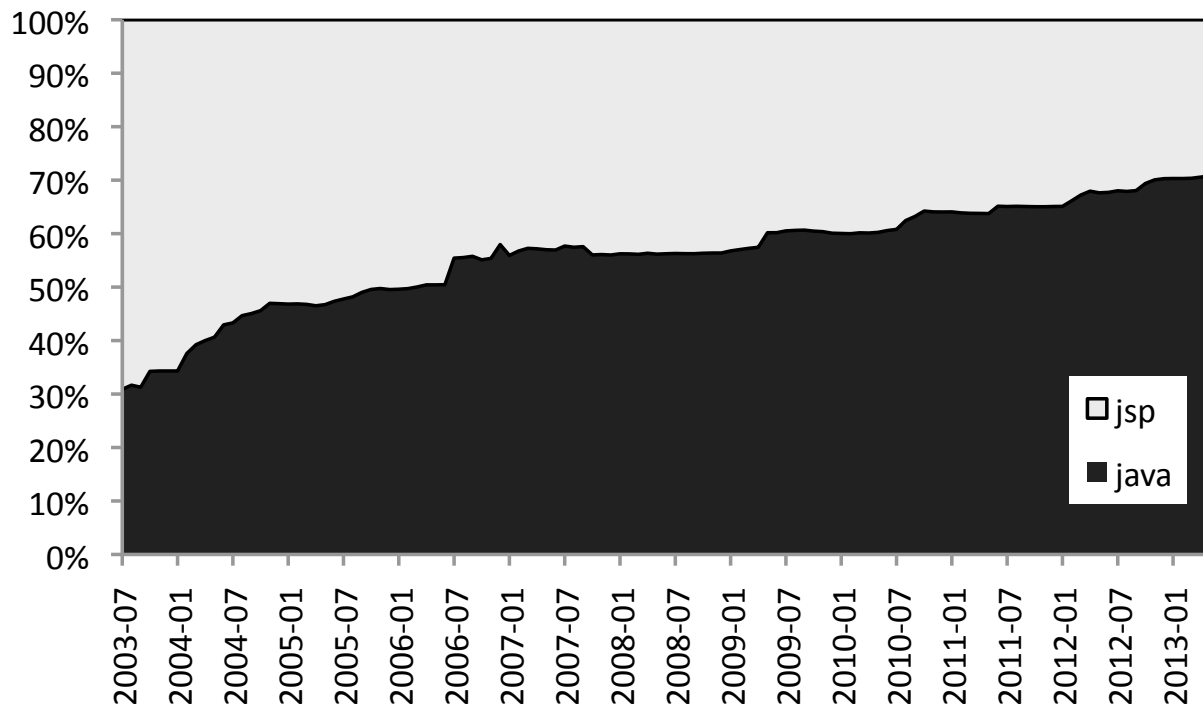
- Canadian research network SCOOP
  - Social Collaboratory for Outcome Oriented Primary care
    - <http://scoop.leadlab.ca>
- Open source tool infrastructure for Electronic Medical Records (EMR)
  - <http://github.com/scoophealth>
- OSCAR: EMR system for healthcare
  - Support for billing, chronic disease management tools, prescription module, scheduling, ...
    - Data available on <https://github.com/scoophealth/oscar.git>

# Case Study: OSCAR

characteristic	value
duration	3,939 days ( > 129 months)
dates	from Nov 2002 till Aug 2013
# of commits	18,727
# of distinct files	20,718 (54% are code files)
# of file touches	93,721
# of distinct developers	100

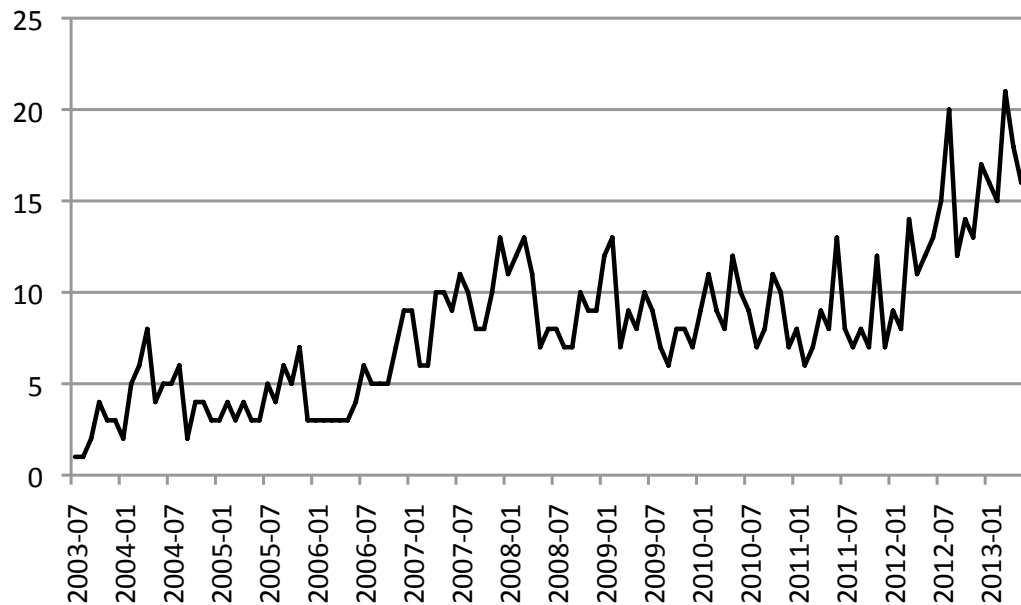
# Evolution of OSCAR

- Monthly aggregated proportion of JSP and Java files in OSCAR



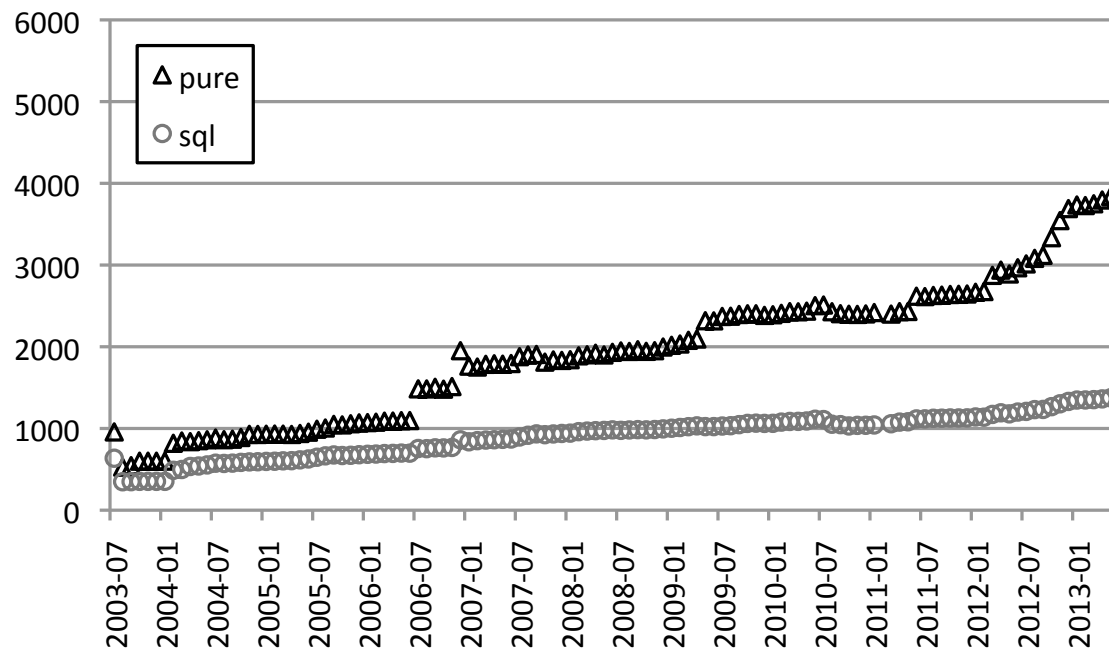
# Evolution of OSCAR - Social Dimension

- Monthly number of distinct active developers for OSCAR



# Evolution of OSCAR

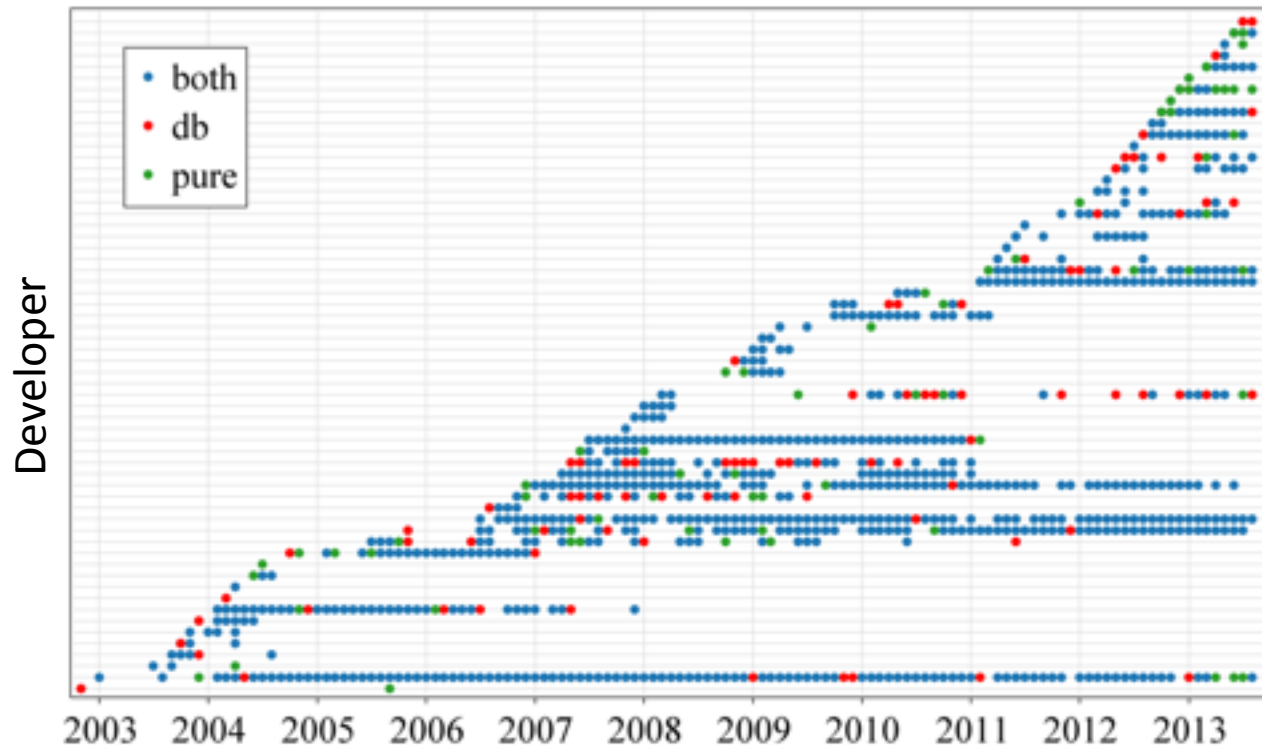
- Growth of source code files and database-related files





# Evolution of OSCAR - Social Dimension

- How does the activity of developers evolve over time?

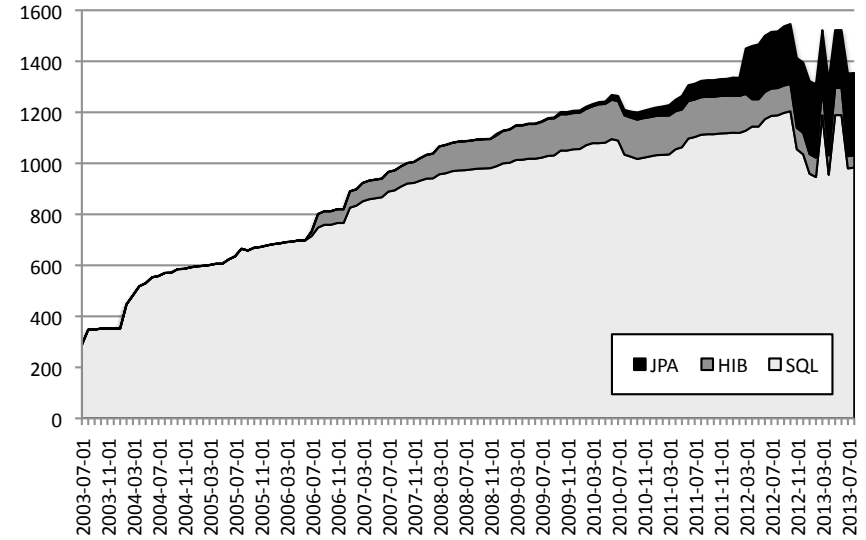
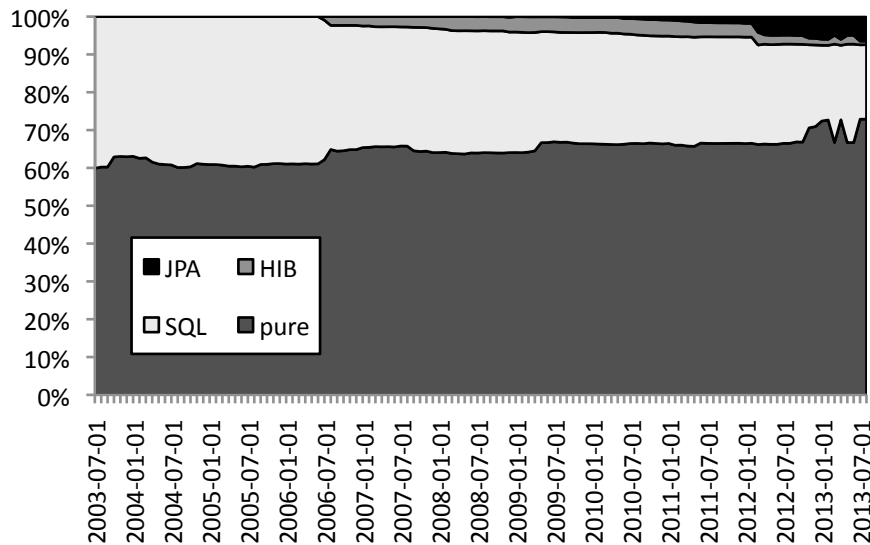


# Introduction of Persistence Provider

- **Hibernate** (introduced in OSCAR since July 2006)
  - Java object-relational mapping (ORM) library
    - XML files map Java classes to database tables and Java data types to SQL data types
    - facilitates data query and retrieval
    - generates SQL calls and relieves the developer from manual result set handling and object conversion
- **JPA** (introduced in OSCAR since July 2008)
  - Java Persistence API
  - Uses Java annotations instead of XML files for ORM

# Introduction of Persistence Provider

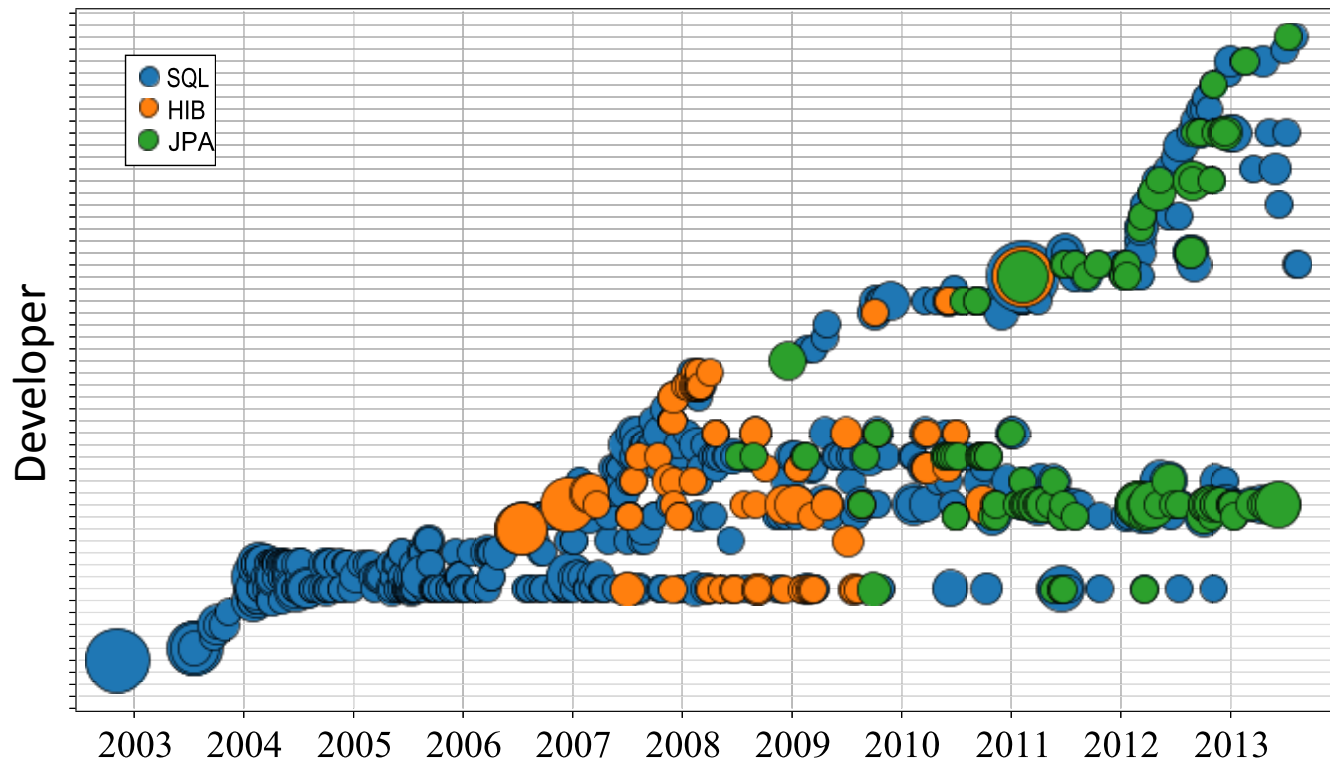
- SQL = code file containing embedded SQL query
- HIB = Java file targeted by Hibernate XML file
- JPA = Java file containing JPA annotation
- pure = code files not containing any of these



Monthly aggregated number of active files

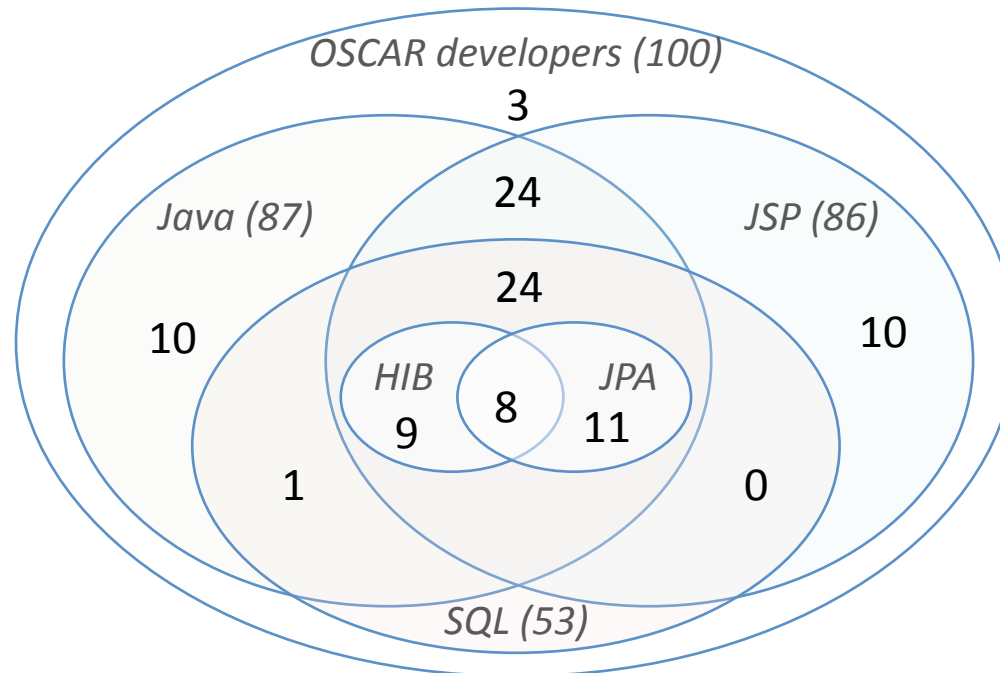
# Introduction of Persistence Provider

- Who is involved in introducing changes in database-related code?



# Evolution of OSCAR - Social Dimension

- How do developers divide their work?



Number of developers that introduce database-related code  
in some file for the first time

# Preliminary Conclusions

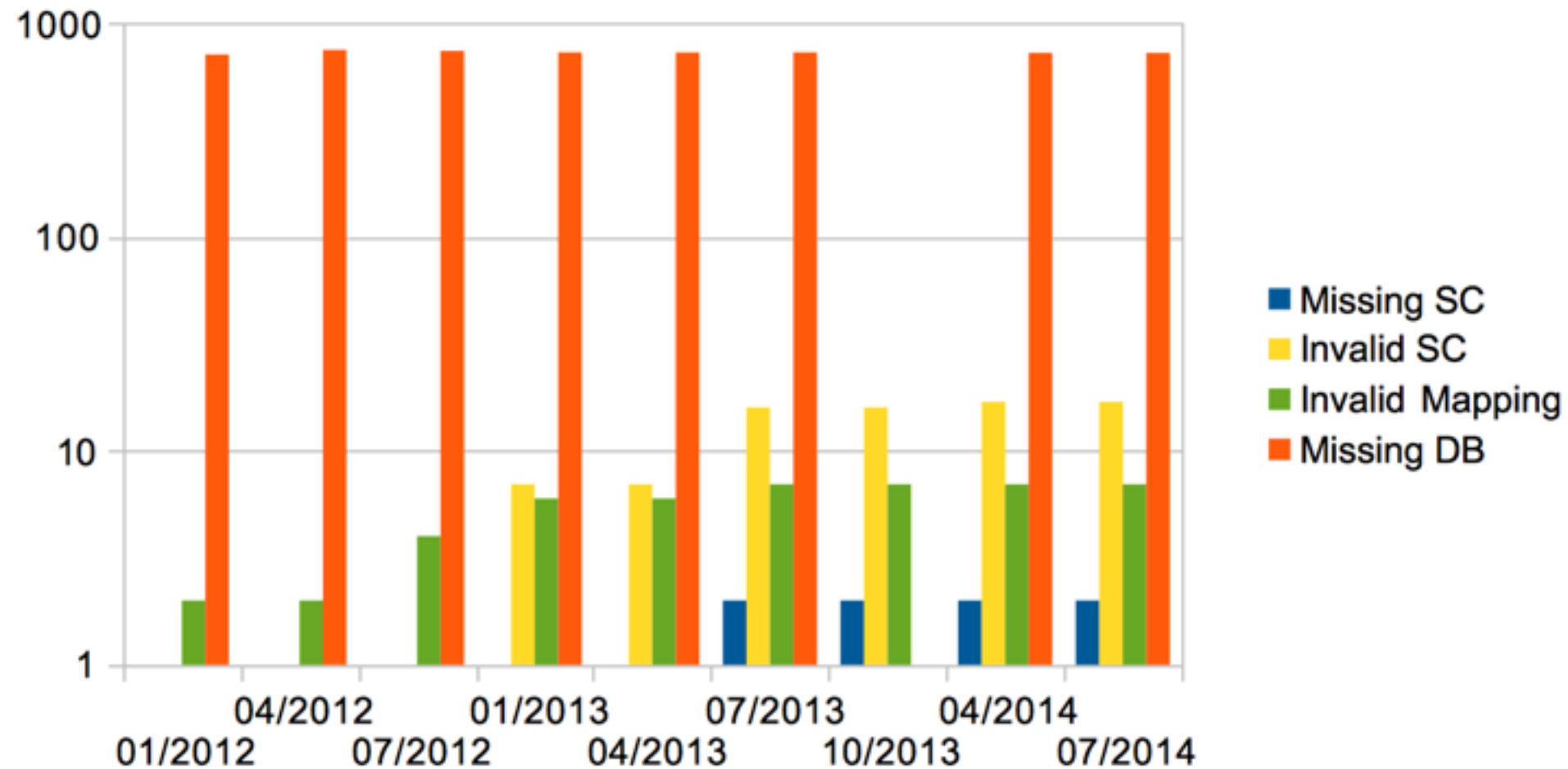
---

- RQ1: Code-related and database-related files evolve together (no “phased” co-evolution)
- RQ2: Migration to Hibernate, then JPA, but embedded SQL still remains important
- RQ3: No clear separation of activities between developers
  - The majority of developers changes both db-related and db-unrelated code
  - No observed periods dedicated to a specific activity

# Using Dali for detecting inconsistencies

- Dali: Eclipse plugin for supporting JPA/Hibernate/... development
- Reports inconsistencies between the source code and the database.
- For each version of considered project
  - Import the source code in Eclipse
  - Create the corresponding database
  - Analyse the errors reported by Dali

# Early results: 9 versions of Oscar





# Future Work

- Continue studying co-evolution between code-related and db-related changes
  - Refine our results by analysing changes at finer granularity
- Analyse database schema changes and their impact on source code (collaboration with UNamur)
  - Detect change patterns in code and database schema
- Study impact of introducing persistence providers
  - Analyse migration patterns in code
  - How do persistence providers reduce impact of changes in database schema?
- Study and compare with other DISS