Change-Based Refactoring Mining

Quinten David Soetens
Javier Pérez
Serge Demeyer

June 3rd, 2013
Echelon Formation

Wedge Formation
Identify which refactorings were performed during evolution of a software system.
State of the Art

Warning: All these are Snapshot Based
Snapshots are Coarse Grained
Fine Grained Code Changes
Change-Based Test Selection in the Presence of Developer Tests

Quinten David Soetens, Serge Demeyer and Andy Zaidman

In "Proceedings of 17th European Conference on Software Maintenance and Reengineering, CSMR 2013"

ChEOPSJ: Change-Based Test Optimization

Quinten David Soetens and Serge Demeyer

In "Proceedings of 16th European Conference on Software Maintenance and Reengineering, CSMR 2012"
First Class Change Objects

Changes act on Source Code (FAMIX) Entities
(e.g. AddClassChange, AddMethodChange, etc.)
First Class Change Objects

Changes have Structural Dependencies
(e.g. AddMethod --> AddClass --> AddPackage etc.)
Change Graphs

FromClass.java

```java
package somepackage;
public class FromClass{
}
```

ToClass.java

```java
package somepackage;
public class ToClass{
    public void boo(){
        [... some code...]
    }
}
```
Refactoring Patterns
Results
Future Directions

- Implement and evaluate this approach
- Use Groove as Graph Transformation Engine
- Compare with RefFinder
- Extend with more Refactoring Patterns
- Industrial Case Study
Conclusions

Mining Refactorings

Snapshots are Coarse Grained

Fine Grained Code Changes

Change Graphs

Refactoring Patterns

Questions?