Sistematic Literature Review
Design Patterns, Anti-patterns and Code Smells

Monografia - Engenharia de Software Experimental
Bruno Cardoso
UFGM, Maio 2014
A systematic literature review is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest.

[Kitchenham and Charters 2007, Almeida Biolchini et al. 2007]
Sistematic Literature Review

Glossary

- **Primary study**: A study investigating a specific research question
- **Secondary study**: A study that reviews all the primary studies relating to a specific research question with the aim of integrating/synthesising evidence related to a specific research question
Sistematic Literature Review

Background

This work aims to verify the relation between design patterns, code smell and anti-patterns.
Background

Design Patterns

Design Patterns: Elements of Reusable Object-Oriented Software is a software engineering book describing recurring solutions to common problems in software design.
If a designer uses the GoF patterns on his design, we hypothesize that he is ensured to select the best known solution to solve his problems.
Background

➤ Design Patterns

➤ Pattern Name and Classification
➤ Intent
➤ Applicability
➤ Structure
➤ Known Uses
➤ Related Patterns
Kent Beck and Martin Fowler have introduced the term “bad smells”

Also called code smells, design smells, among others.
A code smell is a surface indication that usually corresponds to a deeper problem in the system.

Smell is by definition something that's quick to spot.

Smells don't always indicate a problem. Some long methods are just fine.
Anti-patterns are patterns whose purpose is to document common bad practices in software design.
Background

Anti patterns

An anti-pattern explains how a solution that initially appears to be a good choice to solve a specific problem results in the creation of conflicts because of its implementation.
Systematic Literature Review

➢ Conduction process

Systematic review conduction process.
[Biolchini et al. 2007] came up with a model to support sistematic reviews:

- Question Formularization
- Sources selection
- Studies selection
- Information Extraction
- Results Summarization
Goals

To summarise the existing evidence concerning a treatment or technology

To identify any gaps in current research in order to suggest areas for further investigation

To provide a framework/background in order to appropriately position new research activities
Sistematic Literature Review

Goals

To **identify any gaps in current research** in order to suggest areas for further investigation.
Sistematic Literature Review

- Question Formularization

- What is the relation between design patterns and code smells? Can design patterns produce smells?

- What is the relation between design patterns and anti patterns? Can design patterns become anti patterns?

- Is there any relation between smells and anti patterns? If there is, which relation is it?
Sistematic Literature Review

Sources selection

IEEE Xplore® Digital Library

ACM DL DIGITAL LIBRARY

Studies should be in English
Sources Selection

Key words

- code smell (code smells, bad smell, bad smells, design smell, design smells)
- design patterns (design pattern)
- antipattern (antipatterns, anti-pattern, anti-patterns)
Sources Selection

Search string

("code smell" OR "code smells" OR "bad smell" OR "bad smells" OR "design smell" OR "design smells")
AND ("design patterns" OR "design pattern")
AND ("antipattern" OR "antipatterns" OR "anti-pattern" OR "anti-patterns")
Studies Selection

- Inclusion/Exclusion Criteria
- Selection execution
Inclusion Criteria

Design Pattern

Cause –consequence
Structural Refactoring

Code Smell

Cause –consequence
Structural Refactoring

Cause –consequence
Structural Refactoring

Anti Pattern
Exclusion Criteria

Studies whose focus were the analysis/comparison of detection tools were not considered in this literature review.
## Studies Selection

### Protocol

<table>
<thead>
<tr>
<th>Phase</th>
<th>Source</th>
<th>Number of studies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search String</td>
<td>IEEE Xplore</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>ACM Digital Library</td>
<td>31</td>
</tr>
<tr>
<td>Analysing Abstract</td>
<td>IEEE Xplore</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>ACM Digital Library</td>
<td>12</td>
</tr>
<tr>
<td>Scanning</td>
<td>IEEE Xplore</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>ACM Digital Library</td>
<td>5</td>
</tr>
</tbody>
</table>
Given the inclusion/exclusion criteria, the studies were extracted directly from the sources.

Only studies whose abstracts were considered to fit the inclusion criteria were extracted.
<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>An Ontological Identification of Relationships between Anti-Patterns and Code Smells</td>
<td>Luo, Hoss and Carver</td>
<td>2010</td>
</tr>
<tr>
<td>SQUAD: Software Quality Understanding through the Analysis of Design</td>
<td>Foutse Khomh</td>
<td>2009</td>
</tr>
<tr>
<td>A Logic Based Approach to Locate Composite Refactoring Opportunities in Object-Oriented Code</td>
<td>Jebelean, Chiril and Cretu</td>
<td>2010</td>
</tr>
<tr>
<td>Towards Automated Restructuring of Object Oriented Systems</td>
<td>Trifu and Reupke</td>
<td>2007</td>
</tr>
<tr>
<td>Sharing Bad Practices in Design to Improve the Use of Patterns</td>
<td>Bouhours et al</td>
<td>2010</td>
</tr>
<tr>
<td>SearchBased Determination of Refactorings for Improving the Class Structure of ObjectOriented Systems</td>
<td>Seng, Stammel and Burkhart</td>
<td>2006</td>
</tr>
<tr>
<td>Impact of Refactoring on Quality Code Evaluation</td>
<td>Fontana and Spinelli</td>
<td>2011</td>
</tr>
<tr>
<td>Combining Clustering and Pattern Detection for the Reengineering of Component-based Software Systems</td>
<td>Detten and Becker</td>
<td>2011</td>
</tr>
<tr>
<td>Perspectives on Automated Correction of Bad Smells</td>
<td>Pérez, Crespo</td>
<td>2009</td>
</tr>
<tr>
<td>Title</td>
<td>Why?</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>An Ontological Identification of Relationships between Anti-Patterns and Code Smells</td>
<td>Cause-consequence (Smells -&gt; Antipattem)</td>
<td></td>
</tr>
<tr>
<td>On Extended Similarity Scoring and Bit-vector Algorithms for Design Smell Detection</td>
<td>Structure (Design Pattern - Antipattern)</td>
<td></td>
</tr>
<tr>
<td>SQUAD: Software Quality Understanding through the Analysis of Design Structure</td>
<td>Structure (AP-DP ; Smells-DP)</td>
<td></td>
</tr>
<tr>
<td>A Logic Based Approach to Locate Composite Refactoring Opportunities in Object-Oriented Code</td>
<td>Refactoring (Smells -&gt; DP)</td>
<td></td>
</tr>
<tr>
<td>Towards Automated Restructuring of Object Oriented Systems</td>
<td>Refactoring (AP, Smells -&gt; DP)</td>
<td></td>
</tr>
<tr>
<td>Software Change in the Solo Iterative Process: An Experience Report</td>
<td>Refactoring (AP, Smells -&gt; DP)</td>
<td></td>
</tr>
<tr>
<td>Sharing Bad Practices in Design to Improve the Use of Patterns</td>
<td>Structure (AP-DP ; Smells-DP)</td>
<td></td>
</tr>
<tr>
<td><strong>SearchBased Determination of Refactorings for Improving the Class Structure of ObjectOriented Systems</strong></td>
<td>Refactoring (Smells -&gt; DP)</td>
<td></td>
</tr>
<tr>
<td><strong>Impact of Refactoring on Quality Code Evaluation</strong></td>
<td>Refactoring (Smells -&gt; DP)</td>
<td></td>
</tr>
<tr>
<td>Combining Clustering and Pattern Detection for the Reengineering of Component-based Software Systems</td>
<td>Structure (Smells-DP)</td>
<td></td>
</tr>
<tr>
<td>Perspectives on Automated Correction of Bad Smells</td>
<td>Refactoring (Smells -&gt; DP)</td>
<td></td>
</tr>
</tbody>
</table>
Results Summarization

Conclusions

- 6 studies concerned on refactoring
- 4 studies stated a paralell between Design Pattern/Antipattern structures or and Design Pattern/Smell structures
- 1 study worked on cause-consequence relation between smells and antipatterns
None of the studies proposed to deeply search a relation of cause-consequence between Design Patterns and Code Smells

2 studies cited that Design Pattern may cause smells due to misuse or Pattern definition
Threats to validity

- More sources could have been used in order to have a bigger variety of studies to select
- There was only one researcher
Questions?