Sistematic Literature Review

Design Patterns, Anti-patterns and Code Smells

Monografia - Engenharia de Software Experimental

Bruno Cardoso

UFMG, Maio 2014

Sistematic Literature Review

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

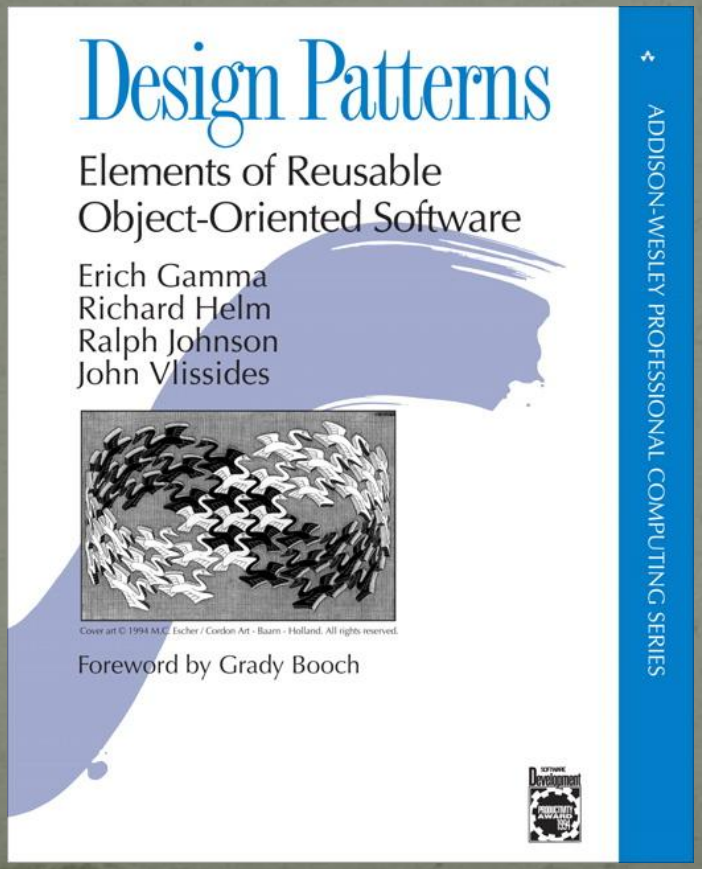
Sistematic Literature Review

- Background

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

Background

- Design Patterns

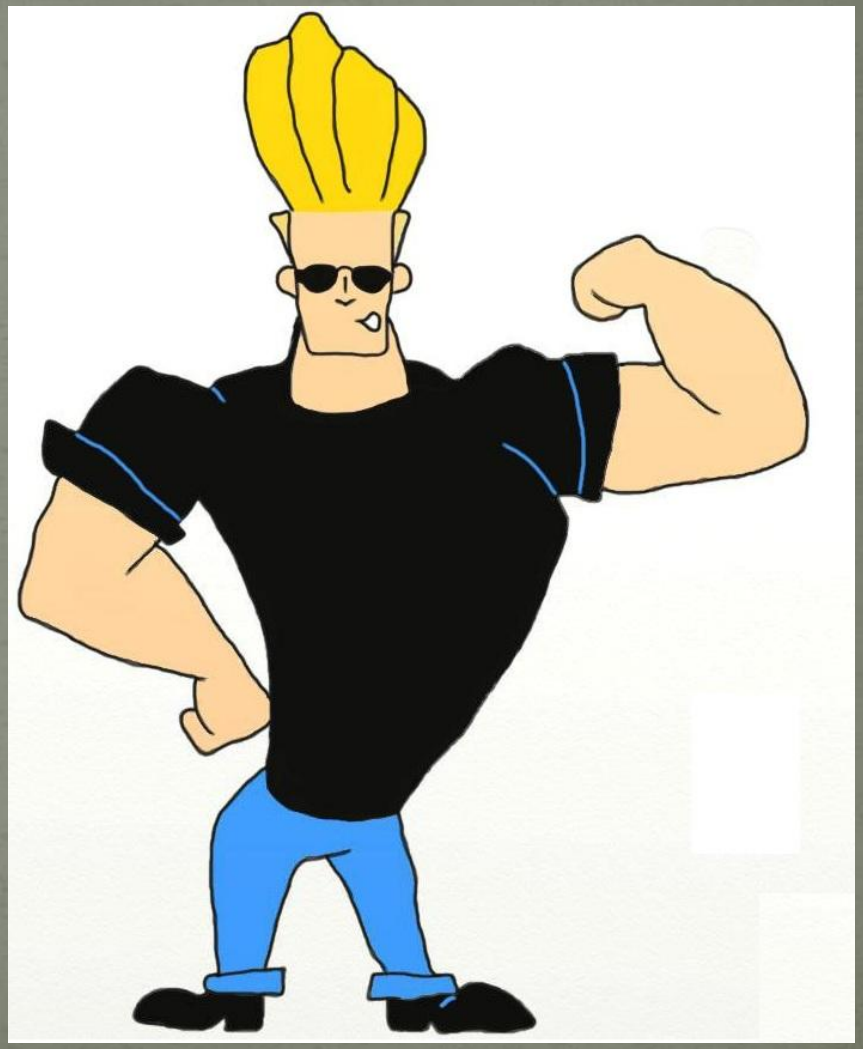


Design Patterns: Elements of Reusable Object-Oriented Software is a software engineering book describing recurring solutions to common problems in software design.

Background

- Design Patterns

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.



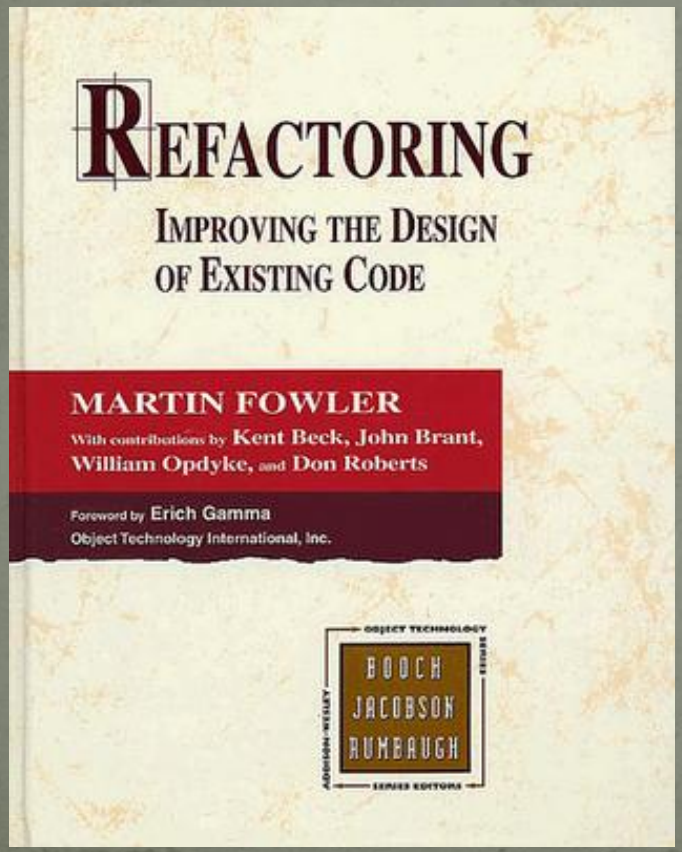
Background

- Design Patterns

- **Pattern Name and Classification**
- **Intent**
- **Applicability**
- **Structure**
- **Known Uses**
- **Related Patterns**

Background

- Code smells



Kent Beck and Martin Fowler have introduced the term "**bad smells**"

Also called **code smells**, **design smells**, among others.

Background

- Code smells

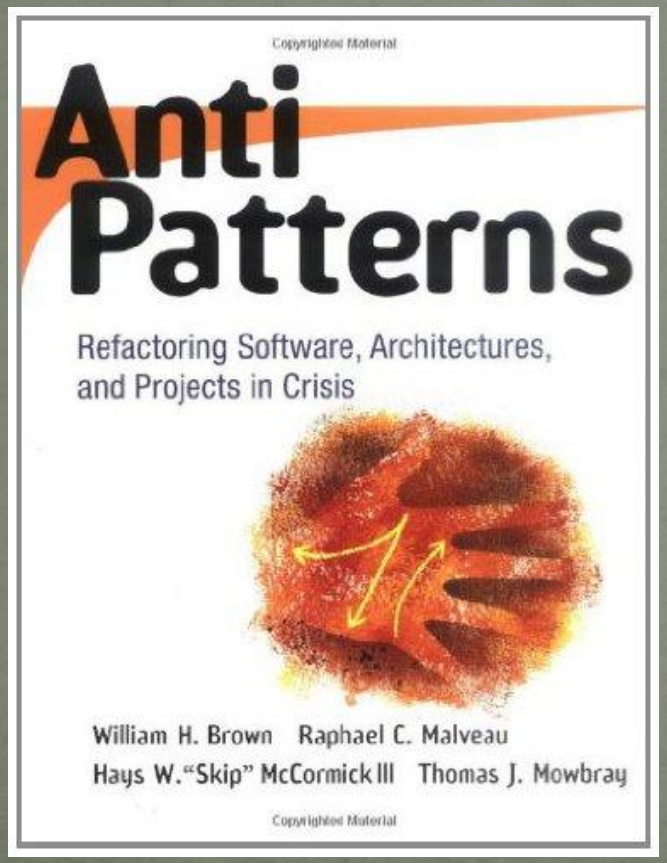
A code smell is a **surface indication** that usually corresponds to a deeper problem in the system.

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

Smells **don't always indicate a problem**. Some long methods are just fine.

Background

- Anti patterns



Anti-patterns are patterns whose purpose is to document common bad practices in software design.

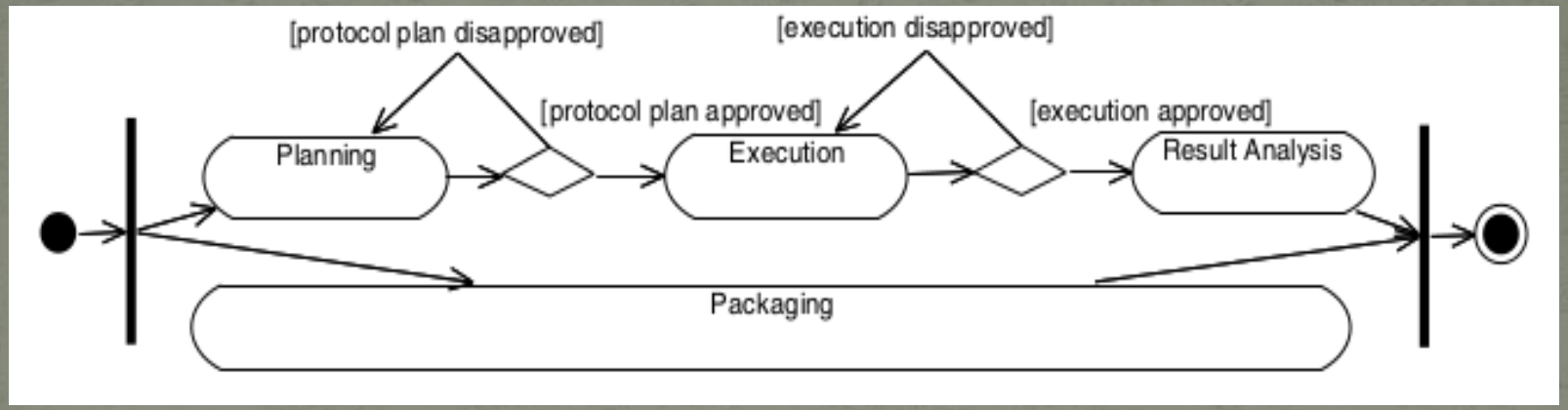
Background

- 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.

Sistematic Literature Review

- Conduction process



Systematic review conduction process.

Sistematic Literature 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

Sistematic Literature Review

- 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

Design Pattern

Code Smell

Sistematic Literature Review

➢Question Formularization

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

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

➢Is there any relation between smells and anti patterns? If there is, which relation is it?

Sistematic Literature Review

- Sources selection

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

Cause –consequence
Structural
Refactoring

Code Smell

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

Phase	Source	Number of studies
Search String	Ieee Xplore	71
	ACM Digital Library	31
Analysing Abstract	Ieee Xplore	23
	ACM Digital Library	12
Scanning	Ieee Xplore	6
	ACM Digital Library	5

Information Extraction

Given the inclusion/exclusion criteria, the studies were extracted directed from the sources

Only studies whose abstracts were considered to fit the inclusion criteria were extracted

Results Summarization

Title	Author	Year
An Ontological Identification of Relationships between Anti-Patterns and Code Smells	Luo, Hoss and Carver	2010
On Extended Similarity Scoring and Bit-vector Algorithms for Design Smell Detection	Polášek et al	2012
SQUAD: Software Quality Understanding through the Analysis of Design	Foutse Khomh	2009
A Logic Based Approach to Locate Composite Refactoring Opportunities in Object-Oriented Code	Jebelean, Chiril and Cretu	2010
Towards Automated Restructuring of Object Oriented Systems	Trifu and Reupke	2007
Software Change in the Solo Iterative Process: An Experience Report	Dorman and Rajlich	2012
Sharing Bad Practices in Design to Improve the Use of Patterns	Bouhours et al	2010
SearchBased Determination of Refactorings for Improving the Class Structure of ObjectOriented Systems	Seng, Stammel and Burkhart	2006
Impact of Refactoring on Quality Code Evaluation	Fontana and Spinelli	2011
Combining Clustering and Pattern Detection for the Reengineering of Component-based Software Systems	Detten and Becker	2011
Perspectives on Automated Correction of Bad Smells	Pérez, Crespo	2009

Results Summarization

Title	Why?
An Ontological Identification of Relationships between Anti-Patterns and Code Smells	Cause-consequence (Smells -> Antipatttern)
On Extended Similarity Scoring and Bit-vector Algorithms for Design Smell Detection	Structure (Design Pattern - Antipattern)
SQUAD: Software Quality Understanding through the Analysis of Design	Structure (AP-DP ; Smells-DP)
A Logic Based Approach to Locate Composite Refactoring Opportunities in Object-Oriented Code	Refactoring (Smells -> DP)
Towards Automated Restructuring of Object Oriented Systems	Refactoring (AP, Smells -> DP)
Software Change in the Solo Iterative Process: An Experience Report	Refactoring (AP, Smells -> DP)
Sharing Bad Practices in Design to Improve the Use of Patterns	Structure (AP-DP ; Smells-DP)
SearchBased Determination of Refactorings for Improving the Class Structure of ObjectOriented Systems	Refactoring (Smells -> DP)
Impact of Refactoring on Quality Code Evaluation	Refactoring (Smells -> DP)
Combining Clustering and Pattern Detection for the Reengineering of Component-based Software Systems	Structure (Smells-DP)
Perspectives on Automated Correction of Bad Smells	Refactoring (Smells -> DP)

Results Summarization

➢Conclusions

➢6 studies concerned on refactoring

➢4 studies stated a paralell between Design Patttern/Antipattern structures or and Design Patttern/Smell structures

➢1 study worked on cause-consequence relation between smells and antipatterns

Results Summarization

➢Conclusions

➢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?