



# Università degli Studi di Salerno

Dipartimento di Scienze Aziendali – Management and Innovation Systems

---

Dottorato di Ricerca

## Management and Information Technology

Curriculum in Informatica, Sistemi Informatici e Tecnologie del Software

XV Ciclo

Tesi di Dottorato

## **Code Smells: Relevance of the Problem and Novel Detection Techniques**

**Tutor**

Prof. Andrea De Lucia

**Coordinatore**

Prof. Andrea De Lucia

**Candidato**

Fabio Palomba

Matr. 8887600001

---

Anno Accademico 2015/2016

# Abstract

I sistemi software stanno diventando il cuore delle attività di molte aziende e, per questa ragione, sono sempre più grandi e complessi. Inoltre, sono frequentemente soggetti a modifiche che riguardano l'implementazione di nuove funzionalità o la risoluzione di difetti. In questo contesto, spesso gli sviluppatori non hanno la possibilità di progettare ed implementare soluzioni ideali, introducendo quindi technical debt, ovvero codice non ben progettato, la cui ri-progettazione viene postposta nel futuro.

Un notevole sintomo della presenza di technical debt è rappresentato dai bad code smell, che sono stati definiti da Fowler per indicare scelte di progettazione e/o implementazione sub-ottimali applicati dai programmatori durante lo sviluppo di un progetto software.

Nel recente passato, molti studi hanno dimostrato l'impatto negativo dei code smell sulla manutenibilità e comprensibilità del codice sorgente. Per questa ragione, molte tecniche sono state proposte per l'identificazione di porzioni di codice affetto da problemi di progettazione. Molte di queste tecniche si basano sull'analisi delle propriet strutturali (ad esempio, chiamate a metodi esterni) estraibili dal codice sorgente.

Nonostante lo sforzo che la comunità di ricerca ha profuso negli anni recenti, ci sono ancora limitazioni che precludono l'applicabilità industriale di tool per l'identificazione di smell. Nello specifico, c'è una mancanza di evidenza empirica riguardo (i) le circostanze che portano all'introduzione degli smell, e (ii) il reale impatto degli smell sulla manutenibilità, in quanto studi precedenti hanno focalizzato l'attenzione su un numero limitato di progetti software. Inoltre, le tec-

---

niche esistenti per l'identificazione di smell sono inadeguate per quanto concerne l'identificazione di molti dei code smell definiti in letteratura. Ad esempio, molti code smell sono intrinsecamente caratterizzati da come gli elementi nel codice cambiano nel tempo, piuttosto che da proprietà strutturali estraibili dal codice sorgente.

Nel contesto di questa tesi abbiamo affrontato queste sfide specifiche, proponendo diversi studi empirici su larga scala aventi come obiettivo quello di capire (i) quando e perchè i code smell sono realmente introdotti, (ii) qual è la loro longevità e come gli sviluppatori li rimuovono, (iii) qual è l'impatto degli smell sulla propensione ai difetti e ai cambiamenti, e (iv) come gli sviluppatori percepiscono gli smell.

Allo stesso tempo, abbiamo proposto due nuovi approcci per la rilevazione di code smell che si basano sull'utilizzo di sorgenti alternative di informazioni, ovvero storiche e testuali, e abbiamo valutato e confrontato la loro capacità nella identificazione rispetto alle altre tecniche basate su analisi strutturale.

I risultati riportati in questa tesi contraddicono le aspettative comuni. Ad esempio, abbiamo dimostrato che i code smell sono spesso introdotti durante il primo commit che introduce l'artefatto affetto dal problema di progettazione. Dall'altro lato, abbiamo dimostrato l'utilità dell'analisi storica e testuale come un modo aggiuntivo per migliorare tecniche esistenti con informazioni ortogonali. Inoltre, forniamo un insieme di problemi aperti che necessitano di ulteriore attenzione in futuro, così come una panoramica di ulteriori applicazioni future dei code smell in altri contesti nel campo dell'ingegneria del software.