Abstract

Anti-patterns are the defects which affects the system quality negatively. An indication of the existence of anti-patterns, in the software is known as "Code Smell" which leads to the refactoring of system. Thus the maintenance becomes difficult to manage. More the number of smells more refactoring is needed. Different approaches have been identified for the detection of anti-patterns in the system. The paper aimed at investigating the impact of anti-patterns on classes and what are the certain kinds of anti-patterns that have a higher impact than others Finally the results have been concluded for the future studies in open source systems. The paper is divided in to four sections in which the introduction is followed by the
types of anti-patterns. Furthermore the related work have been examined carefully with a brief conclusion. Thus the paper reveals different approaches for the identification code smells in the software system. Hence the detection of smells will be helpful in providing more reliability during testing and maintenance phases by predicting anti-patterns and faults before the delivery of the product. Moreover the identification of anti-patterns will be of usage to the community of software engineers and managers for improving the software development maintenance activities.

References

- M. Fowler, 1999 "Refactoring—improving the design of existing code," 1st Ed. Addison-Wesley.
- Yann Gael Gueheneuc, Herve Albin-Amiot and Ecole des Mines de Nantes,2001 "Using Design Patterns and Constraints to Automate the Detection and Correction of Inter-class Design Defects," Paper accepted at TOOLS USA.


Naouel Moha, Yann-Gael Gueheneuc, Anne-Francoise Le Meur, Laurence Duchien
and Alban Tiberghien, 2010 "From a Domain Analysis to the Specification and Detection of Code and Design Smells," Springer Verlag (Germany), pp. 345-361.


Index Terms
Computer Science
Information Sciences

Keywords
Anti-patterns  Code Smells  Refactoring And Maintenance.