



Technical University of Vienna
Information Systems Institute
Distributed Systems Group

Improving Evolvability through Refactoring

Jacek Ratzinger, Michael Fischer and
Harald Gall

ratzinger@infosys.tuwien.ac.at

fischer@infosys.tuwien.ac.at

gall@ifi.unizh.ch

TUV-1841-2005-33

March 23, 2005

Refactoring is one means of improving the structure of existing software. Locations where to apply refactoring are often based on subjective perceptions such as "bad smells", which are vague suspicions of design shortcomings. We exploit historical data extracted from repositories such as CVS and focus on change couplings: if some software parts change at the same time very often over several releases, this data can be used to point to candidates for refactoring. We adopt the concept of bad smells and provide additional change smells. Such a smell is hardly visible in the code, but easy to spot when viewing the change history. Our approach enables the detection of such smells allowing an engineer to apply refactoring on these parts of the source code to improve the evolvability of the software. For that, we analyzed the history of a large industrial system for a period of 15 months, proposed spots for refactorings based on change couplings, and performed them with the developers. After observing the system for another 15 months we finally analyzed the effectiveness of our approach. Our results support our hypothesis that the combination of change dependency analysis and refactoring is applicable and effective.

Keywords: software evolution, refactoring, change smells

