

DOCTORAL CANDIDATE: Aiko Yamashita
DEGREE: Philosophiae Doctor
FACULTY: Mathematics and Natural Sciences
DEPARTMENT: Department of Informatics
AREA OF EXPERTISE: Empirical Software Engineering
SUPERVISORS: Ole Hanseth and Magne Jørgensen
DATE OF DISPUTATION: 15st of October 2012

DISSERTATION TITLE: *Assessing the Capability of Code Smells to Support Software Maintainability Assessments: Empirical Inquiry and Methodological Approach*

Code smells are design weaknesses that can decrease the maintainability of the source code in a software system. Some code smells can be identified automatically in the source code via software tools. The main goal of this thesis is to investigate, from different perspectives and in a realistic setting, how good code smells are for measuring or evaluating the maintainability of a software system.

A main conclusion derived from our research is that the usefulness of code smells depends on the maintainability perspective involved, i.e., whether the analysis is conducted at system level or at file level, and also depends of how maintainability is measured, e.g., in terms of maintenance effort, occurrences of problems, etc.

When the software was analyzed as a whole, the capacity of code smells to evaluate the maintainability was found to be relatively modest. When maintainability was measured via maintenance effort and defects, code smells present in a system was not a better indicator of maintainability than the simpler measure of system size, measured as lines of code (LOC) of a system. Also, when maintainability was measured through the occurrence of problems during maintenance, the role of code smells on the overall system maintainability was relatively small. For example, of the total set of maintenance problems, only about 30% were related to code containing code smells. When maintainability was measured through a set of system-level characteristics deemed important by software developers, many of these characteristics also did not directly correspond to current definitions of code smells.

When the software was analyzed at file level, code smells were not found useful on pointing out files that are more likely to require more effort than others during maintenance. However, code smells were found to be useful in pointing out files that are more likely to be problematic during maintenance. Also, some important system-level maintainability characteristics were highly related to code smells. "Design consistency," for example, was a characteristic considered highly important by software developers and, it showed high correspondence with several code smells.

Given these results, to achieve more complete evaluations of software maintainability, it is recommended to combine code smells analysis with other approaches such as expert judgment and semantic analysis techniques.

In order to combine expert judgment and code smell analysis (and to consequently support more complete evaluations of maintainability), a technique called concept

mapping was adopted from social research to software engineering. A report was presented on our experiences, in-sights and recommendations from using this technique in a software maintainability evaluation context. The main conclusion is that despite some limitations, concept mapping is a promising approach for maintainability evaluations that need to combine different sources of information, such as code smell analysis and expert judgment.