Chapter 11
Software Metrics Evaluation Based on Entropy

R. Selvarani
Dayananda Sagar Institutions, Bangalore, India

T.R. Gopalakrishnan Nair
Dayananda Sagar Institutions, India

Muthu Ramachandran
Leeds Metropolitan University, UK

Kamakshi Prasad
JNTU, Hyderabad, India

ABSTRACT

The complexity of modern software, the commercial constraints and the expectation for high quality product demands the accurate fault prediction based on OO design metrics in the class level in the early stages of software development. The object oriented class metrics are used as quality predictors in the entire OO software development life cycle even when a highly iterative, incremental model or agile software process is employed. Recent research has shown some of the OO design metrics are useful for predicting fault-proneness of classes. In this chapter the empirical validation of a set of metrics proposed by Chidamber and Kemerer is performed to assess their ability in predicting the software quality in terms of fault proneness and degradation. The authors have also proposed the design complexity of object-oriented software with Weighted Methods per Class metric (WMC-CK metric) expressed in terms of Shannon entropy, and error proneness.

I. INTRODUCTION

Object-oriented design and development are popular concepts in today’s software development environ-ment, object-oriented (OO) development has proved its value for systems that must be maintained and modified. OO software development requires a different approach from more traditional functional decomposition and data flow development methods.
including the metrics used to evaluate OO software. The concepts of software metrics are well established, and many metrics relating to product design quality have been developed and used. One approach to controlling software maintenance costs is the utilization of software metrics during the development phase, to help identify potential problem areas in the design.

Software design complexity is a highly important factor affecting the cost of software development and maintenance. If we can determine the impact of complexity factors on maintenance effort, we can develop guidelines which will help reduce the costs of maintenance by recognizing troublesome situations early in the development phase. In response to these situations the managers may take appropriate decision to reduce the design complexity of the system, to develop tools that support maintenance of complex modules, to write documentation that helps the developer to manage the complexity better, to allocate the resources to reflect the situation.

This paper presents the empirical evaluation of CK metrics (Chidamber, Shyam & Kemerer 1994) for object oriented design based on measurement theory and ontology. These measures applied in software system could be used to aid management in estimating the cost and schedule of future projects, evaluating the productivity impacts of new tools and techniques, establishing productivity trends over time, improving software quality, forecasting future staffing needs, and anticipating and reducing future maintenance requirements.

A method based on information theory has also been proposed for examining software design complexity using one of the widely accepted OO complexity design metrics in the context of empirical complexity threshold criteria to assess system-wide software degradation. We have considered five C++ projects done by different group of students. The analysis showed that components with high design complexities were associated with more maintenance activities than those components with lower class complexities.

II. METRIC EVALUATION CRITERIA

Metrics are defined by Fenton and Pfleeger in (Fenton & Pfleeger, 1996) as output of measurements, where measurement is defined as the process by which values are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules. Software metrics are the measures of attributes of a software system (Weyuker, 1988). Traditional functional decomposition metrics and data analysis design metrics measure the design structure independently. OO metrics treats function and data as a combined, integrated object (Chidamber, Shyam & Kemerer, 1994). To evaluate a metric’s usefulness as a quantitative measure of software quality, it must be based on the measurement of a software quality attribute. The metrics evaluate the OO concepts such as methods, classes, cohesion, coupling, and inheritance. The metrics focus on internal object structure, external measures of the interactions among entities, measures of the efficiency of an algorithm and the use of machine resources, and the psychological measures that affect a programmer’s ability to create, comprehend, modify, and maintain software.

III. EMPIRICAL LITERATURE ON CK METRICS

There are a number of empirical studies on CK metrics (Chidamber, Shyam & Kemerer 1994; Booch, & Grady, 1994; Bansiya, Davis, & Etzkon, 1999). The existing empirical studies have been compared and the analysis of their results has been reported by Subramanyam and Krishana (Bansiya, Davis, & Etzkon, 1999). To improve the effectiveness of developer interactions in the study, we have adopted a ground theory (can be defined as a systematic qualitative approach to research methodology where research hypothesis and theories can be formulated based on the data collected, (Subramanyam, & Krishnan, 2003)
Related Content

Formalization of Expert Knowledge About the Usability of Web Pages Based on User Criteria Aggregation

Rosetta Composition Semantics
Megan Peck and Perry Alexander (2014). *Advances and Applications in Model-Driven Engineering* (pp. 201-239).
[www.igi-global.com/chapter/rosetta-composition-semantics/78617?camid=4v1a](www.igi-global.com/chapter/rosetta-composition-semantics/78617?camid=4v1a)

Fast Data Processing for Large-Scale SOA and Event-Based Systems
[www.igi-global.com/article/fast-data-processing-for-large-scale-soa-and-event-based-systems/137070?camid=4v1a](www.igi-global.com/article/fast-data-processing-for-large-scale-soa-and-event-based-systems/137070?camid=4v1a)

Specifying Knowledge Graph with Data Graph, Information Graph, Knowledge Graph, and Wisdom Graph
[www.igi-global.com/article/specifying-knowledge-graph-with-data-graph-information-graph-knowledge-graph-and-wisdom-graph/201482?camid=4v1a](www.igi-global.com/article/specifying-knowledge-graph-with-data-graph-information-graph-knowledge-graph-and-wisdom-graph/201482?camid=4v1a)