# Chapter 22
# Measurement of Cognitive Functional Sizes of Software

**Sanjay Misra**
*Atilim University, Turkey*

## ABSTRACT

*One of the major issues in software engineering is the measurement. Since traditional measurement theory has problem in defining empirical observations on software entities in terms of their measured quantities, Morasca tried to solve this problem by proposing Weak Measurement theory. Further, in calculating complexity of software, the emphasis is mostly given to the computational complexity, algorithm complexity, functional complexity, which basically estimates the time, efforts, computability and efficiency. On the other hand, understandability and compressibility of the software which involves the human interaction are neglected in existing complexity measures. Recently, cognitive complexity (CC) to calculate the architectural and operational complexity of software was proposed to fill this gap. In this paper, we evaluated CC against the principle of weak measurement theory. We find that, the approach for measuring CC is more realistic and practical in comparison to existing approaches and satisfies most of the parameters required from measurement theory.*

## INTRODUCTION

The key element of any engineering process is measurement. Engineers use measures to better understand and assess the quality of engineered products or systems that they built. However, absolute measures are uncommon in software engineering. Instead, software engineers attempt to derive a set of indirect measures that provide an indication of quality of some representation of software. Software engineers plan 'how' an information system should be developed in order to achieve its quality objectives. The quality objectives may be listed as performance, reliability, availability and maintainability and are closely related to software complexity. Numbers of researchers have proposed variety of software complexity measures (Halstead, 1997), (Kushwaha and

DOI: 10.4018/978-1-4666-0261-8.ch022

Misra, 2006), (Woodward and Hennel, 1979) and (Wang, 2003, 2009). Out of the numerous proposed measures, selecting a particular complexity measure is again a problem, as every measure has its own advantages and disadvantages. There is an ongoing effort to find such a comprehensive complexity measure, which addresses most of the parameters of software. Further, complexity measurement in computer science is basically related to computational complexities which are computing the time and space complexity. On the other hand in the software engineering, the emphasis is on the functional complexity (Wang, 2003, 2007a, 2009). But in both of case, these measures do not relate the complexity due to the actual human efforts required to comprehend the software. Actually, the one facet of the complexity is the understandability of the software/code. Understandability of the code relates to ease of comprehension. It is a cognitive process. Although, in past some efforts have been done in calculating the cognitive complexity (Rauterberg 1996, Klemola 2003) of a software, but neither they sufficient nor they cover other aspects of the complexity, like internal architecture, operational complexity, etc. (Wang, 2007a).

The cognitive complexity measures are based on cognitive informatics (Wang, 2004), which in turn help in comprehending the software characteristics. Wang and Shao (Wang and Shao, 2003) have proposed cognitive functional size (CFS) measure to calculate the cognitive complexity of a software. The measure defines the cognitive weights for the Basic Control Structures (BCS). Further, they extended and modified the CFS and presented new weight values based on empirical observations (Wang, 2005, 2009).

A newly proposed complexity measure is acceptable, only when its usefulness has been proved by a validation process. It must be validated and evaluated both formally and practically. The purpose of the validation is also to prove the usefulness of attribute, which is measured by the proposed metric. Elements of measurement theory has been proposed and extensively discussed in literature (Briand et al., 1996), (Basili, 2007), (Fenton, 1994; 1991; 1998), (Weyuker, 1998), (Zuse, 1991; 1992) as a means to evaluate the software complexity measures. However, for formal approach of measurement theory, there is a problem: How can we recognize and describe the attribute in the empirical observation domain in order to relate its values to the proposed metric? (Misra and Kilic, 2006). Naturally, the representational measurement theory does not care about the practical difficulty of making empirical observations on the attribute and their identification. These are the underlying reasons for proposal of weak measurement theory by (Morasca, 2003). He has argued that the representation condition is very demanding for state of art of software engineering measurement. Therefore, he proposed for weakening the representation condition and developed the concept of weak measurement theory. We will discuss on it in detail in section 3.

In the present work, we applied measurement theory/weak measurement theory concepts on cognitive complexity measure. This theory for metric evaluation is more practical and useful and encompasses all the factors which are important for the evaluation of any proposed measure. It is worth mentioned that cognitive functional size has already evaluated (Misra 2004) by Weyuker's properties (Weyuker, 1998). However, satisfying Weyuker's properties is necessary, but not sufficient condition for a good complexity measure (Cherniavsky, 1991). These properties are also under several criticisms (Fenton, 1994) and (Zuse, 1991) and still the topic of research (Gursaran, 2001), (Zhang, 2002), (Sharma 2006), (Misra 2006). Further, cognitive functional size measure has been extended and improved by taking empirical observations; it is required to check its validity through measurement theory perspective. As consequences, we validate the cognitive complexity against the principles of measurement theory.

## Related Content