Software Functional Complexity Measurement with the Task Complexity Approach

De Tran-Cao, Ghislain Lévesque, Jean-Guy Meunier [1]

Abstract- **This research investigates software functional complexity, which is loosely defined as the difficulties arising from the functionalities of software, or *problem complexity*. Software is considered as a task. Therefore, software complexity should be analysed and measured by methods and models of task complexity.**

Numerous measurement methods (metrics) have been proposed to measure software complexity, but these have been criticized for their lack of a theoretical model which would serve as a guide for measurement methods. Researchers are now in urgent need of such a model. To fill this need, we propose Wood's task complexity model in this paper as a theoretical model which will make it possible to both capture and quantify complexity.

Wood's model analyzes task complexity in three dimensions: component complexity, coordinate complexity and dynamic complexity. We use the first two dimensions of the model to analyze software complexity at an early phase of the software lifecycle (analysis phase). The third dimension is proving to be difficult to capture at this stage, and so has been ignored in our model.

Keywords- **Complexity measurement, functional complexity measurement, software complexity, task complexity.**

I. Introduction

Software complexity has been studied for over 25 years now, during which time over 100 different measures have been proposed to capture many different aspects of software complexity (see [13], [15], [29]); however, there is no consensus about what software complexity actually is. What is accepted is that there are two main categories of software complexity: computational and psychological [29].

- Computational complexity refers to algorithm efficiency in terms of the time and memory needed to execute a program.
- Psychological (or cognitive) complexity refers to the human effort needed to perform a software task, or, in other words, the difficulty experienced in understanding or performing such a task.

In the literature, software complexity measures refer to psychological complexity, but there are many different interpretations of their meaning. Zuse (1991) said that "*the term complexity measure is a misnomer. The true meaning of the term software complexity is the difficulty to maintain, change and understand software.*"

Basili (1980) defined software complexity as "*a measure of resources expended by a system (human or other) while interacting with a piece of software to perform a given task.*" Henderson-Sellers (1996) challenged this definition, calling it is too broad, and proposed another definition, as follows: "*The cognitive complexity of software refers to those characteristics of software that affect the level of resources used by a person performing a given task on it.*" This definition puts emphasis on the software characteristics that generate the level of resources needed to perform the task. Our research perspective adopts this definition. In other words, it focuses on the *intrinsic complexity* that is derived from the software characteristics and functionalities.

In spite of the fact that many complexity measures have been proposed, software complexity measurement is still in its infancy. The situation in this field is confusing, and not satisfying for the user [29]. Researchers urgently need a theoretical guidance on software measurement in general, and on software complexity measures in particular. Baker et al. (1990) suggested that, "*for research results to be meaningful, software measurement must be well grounded in theory.*" Kearney (1986) had stated previously that "*successful software complexity measure development must be motivated by a theory of programming behaviour*". In this paper, we attempt to apply a cognitive approach to software complexity by using the task complexity model of Wood [28]. First, a software conceptual model is established, and then its elements are mapped to those of Wood's task model, with the result that software complexity is modeled by Wood's task complexity model. Finally, some measures are proposed for quantifying software complexity.

Generally, in the literature, software complexity measures capture different aspects of software. Complexity measures are not necessarily appropriate to the software development or maintenance effort expended or to software cost. For example, software complexity in terms of the Cyclomatic Number [20] is related to the number of defects rather than to software cost or effort. However, the complexity

[1] De Tran-Cao: Ph.D. student, Computer Science Department, University of Quebec in Montreal.

Ghislain Lévesque : professor, Computer Science Department, University of Quebec in Montreal.

Jean-Guy Meunier: professor, Philosophy Department, University of Quebec in Montreal.