A Field Study of Software Functional Complexity Measurement

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

Abstract- **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. To fill this gap, we propose Wood's task complexity model as a theoretical model which will make it possible to both capture and quantify software functional 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 seems to be difficult to capture at this stage, and so has been ignored in our work.

The empirical study in this paper aims to validate our conceptual model for software functional complexity. It also reports an effort estimation model built on 15 software maintenance projects of a telecom company. The result shows that our estimation model, which takes into account both component complexity and coordinative complexity, is better than the effort prediction model built on the linear regression between maintenance effort and COSMIC functional size.

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

I. INTRODUCTION

It is believed that software complexity has a significant impact on software development or maintenance effort and software quality. Nowadays, more than 100 different measures have been proposed to capture many different aspects of software complexity ([14], [16], [32]). 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 [32]. Computational complexity refers to algorithm efficiency in terms of the time and memory needed to execute a program. While 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.

Psychological complexity is normally interpreted as "*the difficulty to maintain, change and understand software*" [32]. Moreover, it is measured via the software characteristics that make the difficulty of software. Henderson-Sellers (1996) states that "*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.*" Our research perspective adopts this definition. It means that we focus on the *intrinsic complexity* 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 [32]. Researchers call for 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 our previous research [30], we have applied a cognitive approach with which the task complexity model of Wood [31] is used as a theoretical guide to establish a software complexity model and to propose some complexity measures. The complexity investigated in our research is functional complexity. It can be interpreted as the difficulty derived from the functionalities of software and it manifests in the effort to maintenance or develop software. Therefore, our measures are likely functional size measures, that is, they take into account the functionalities of software. However, our measures focus on "complexity aspect" rather than "size aspects". Generally, three of the criteria for a software functional size measure defined in ISO/IEC 14143-1 are adopted: (1) the measure is applicable at earlier stages of the development process, (2) the measured values are easy to understand from the customers' perspective, and (3) the measure provides a valid value for estimating efforts and costs. In other words, the complexity measures are expected as indices of effort, that is, the resources needed to build or maintain software.

Our previous research [30] stops at proposing a software functional complexity model. The model has not been validated yet. Therefore, this paper, as a continuity of the previous research, aims to validate empirically that

[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.