
Adrián Hernández-López, Department of Computer Science, Universidad Carlos III de Madrid, Madrid, Spain

Ricardo Colomo-Palacios, Østfold University College, Østfold, Norway

Pedro Soto-Acosta, Department of Management & Finance, Universidad de Murcia, Murcia, Spain

Cristina Casado Lumberas, Department of Education, Universidad Internacional de la Rioja, Madrid, Spain

ABSTRACT

Productivity measurement is constructed by the measure of three categories of elements: inputs, outputs and factors. This concept, which started being used in the manufacturing industry, has been also a research topic within Software Engineering (SE). In this area, the most used inputs are time and effort and the most used outputs are source code and functionality. Despite of their known limitations, many of the most used productivity measures are still being used due to the information that they provide for management goals. In order to enable the construction of new productivity measures for SE practitioners, the existence of other inputs apart from time and effort, and other outputs, apart from source code and functionality is analyzed in this paper. Moreover, differences in usage of the inputs and production of the outputs among some SE job positions are analyzed and explained.

Keywords: Job Position, Knowledge Worker, Measurement, Productivity, Software Engineering

1. INTRODUCTION

A wider use of productivity measures started with the industrialization era. The complexity of the measures has increased since those days. And measures have to take into account now more items than just the manufactured objects and the labor effort. These items that compose productivity measures are twofold: inputs (the elements needed in the production process) and outputs (the elements produced in the production process). In addition, factors, which are not considered neither inputs nor outputs, influence the productivity results (Lagerström, Würtemberg, Holm, & Luczak, 2012; Trendowicz & Münch, 2009). A relationship between these elements (inputs and outputs) is commonly established in order to construct a productivity measure. Typically, this relationship is a ratio, i.e. how much output is produced per unit of input.

DOI: 10.4018/IJITSA.2015010103
Software Engineering (SE) is a new industry in comparison with traditional manufacturing sectors where the productivity measurement was born. In this area, productivity measurement has been present in research since the late 70s and beginning of the 80s (Brooks Jr., 1985). The productivity measures used in those years followed a ratio approach between outputs and inputs, and new approaches started to be considered due to the increase on the area research. The most used inputs were time and effort while the most used outputs were source code, and later functionality. Nowadays, the most considered inputs are still the same. Nevertheless, the functionality is more used than the source code, which has been shelved, in the most commonly used measures (Petersen, 2011).

However, SE practitioners can be classified as knowledge workers which are significantly different than the workforce that was employed at the beginning and middle of the last century. These practitioners handle other type of inputs, in addition to the resources given by the manufacturing workers (their skills, strength, and time). They give to the employers their knowledge, experience, and social competences among others. On the other hand, they do not produce a single output, as industrial age workers did. They produce an amalgam of outputs including intangible elements such as quality or added value.

Any worker performs a job position regardless of the industry in which he/she works. Each job position represents a unique definition of a job within an organization. These positions can be generalized in order to make them applicable to other more specific job positions. For example, a definition of the software engineer job could be obtained from O*Net (“Software Engineer (O*Net Definition),” 2010); nevertheless this position has been characterized to each organization in order to adapt it to their reality, missions and goals. Each job position requires a set of inputs in order to produce a set of outputs within a production process and with the interaction of other job positions. So, if the job positions are different among themselves, then the productivity measures considered, and later used, for different jobs can be different. One example within SE is represented by the use of lines of code or functionality as the solo output in productivity measures. So, do all the jobs in SE produce these outputs? Are there any other possible outputs that could be considered when measuring SE practitioners’ productivity?

In addition, the level of measurement has to be taken into account. Within SE, there is a lack of research on the lower levels (e.g., team and individual level) for productivity measures (Petersen, 2011). Also, the same measures are used in various levels (e.g., using a project level measurement to measure a programmer productivity) despite of their differences (Hernández-López, 2012; Hernández-López, Colomo-Palacios, & García-Crespo, 2013). Considering that the granularity of possible measures for inputs and outputs is not the same at different levels of measurement, then using the same measures at different levels do not make sense (e.g., project and worker levels). Thus, using the same measures at job level and in higher levels can be seen as controversial.

In this work the existence of different inputs (other than the time and the effort) and different outputs (other than the source code and the functionality) is analyzed. Moreover, the existence of differences on the usage of the inputs and on the production of the outputs among some SE job positions will be also tackled. For these purposes, the authors developed and executed a method divided into two phases. The first phase explores the goals (and generates the research hypotheses) and the second phase contrasts the stated hypotheses from the previous phase. The remainder of this paper is organized as follows: first the background related to the concepts under study is presented; second, the method used for the research is explained; third, the method results are used to contrast the stated hypotheses; fourth, the results are discussed; and finally, there is a discussion of the results and findings.
Continuum of Context Explication: Knowledge Discovery Through Process-Oriented Portals
[www.igi-global.com/article/continuum-context-explication/2656?camid=4v1a](www.igi-global.com/article/continuum-context-explication/2656?camid=4v1a)

E-Democracy: The Social Software Perspective
[www.igi-global.com/chapter/democracy-social-software-perspective/25446?camid=4v1a](www.igi-global.com/chapter/democracy-social-software-perspective/25446?camid=4v1a)