A Review of Literature About Models and Factors of Productivity in the Software Factory

Pedro S. Castañeda Vargas, National University of San Marcos, Lima, Peru
David Mauricio, National University of San Marcos, Lima, Peru

ABSTRACT

Software factories seek to develop quality software in a manner comparable to the production of other industrial products, establishing improvements in their production process so as to be more competitive. Productivity, one of the competitiveness pillars, is related to the effort required to fulfill assigned tasks. However, there is no standard way of measuring productivity, making it difficult to establish policies and strategies to improve the factory. In this article, the authors perform a systematic review of the literature on factors affecting productivity of software factories, and models for measuring it. For the period 2005-2017, 74 factors and 10 models are identified. Most of the factors are related to Programming, and a few to Analysis and Design and Testing. Also, most models for measuring productivity only consider activities concerning programming.

KEYWORDS

Analysis, Development, Effectiveness, Efficiency, Performance, Productivity, Software Factory, Testing

INTRODUCTION

Hernández, Colomo and García (2012), indicate that in software engineering, productivity measurement has focused on the productivity of product delivery, perhaps influenced in part by the formulas used to estimate software projects. Software development effort estimation is the process of predicting the most realistic amount of effort required to develop or maintain software, while, according to IEEE (1992), software productivity is defined as the ratio of work product to work effort.

Organizations interested in measuring the productivity of the software factory want a better view of their ability to use available resources to produce profitable goods or services that meet customer needs. Measuring productivity means knowing the performance of an organization, both internal and external, in order to evaluate its progress. The organization measuring productivity has indicators that allow it to compare itself to the market, in order to propose actions that increase its overall efficiency and use all resources in an effective and efficient way. In other words, like everyone, managers need to know how they are doing compared to performance in previous periods, addressing questions such as whether performance is increasing, decreasing, advancing, or retreating; the magnitude of that progress or regress; and program effectiveness.

According to Petersen (2011), the focus of the software improvement process is often on improving productivity, which implies making a proper measurement. It must not only collect information from various sources and then establish results, but also demand greater effort to understand the various elements that comprise it. Figure 1 shows the conceptual model for productivity measurement by

DOI: 10.4018/IJITSA.2018010103

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Arcudia-Abad, Solís-Carcaño, and Cuesta-Santos (2007). The productivity process consists of types of factors (input, output, context, and process) that influence it. Therefore, measurement requires homogenized units for expressing the data obtained, to allow the design of measurement indicators that will provide coherent and comparable results in differing environments.

It is necessary to establish the measurement objective before starting the productivity-measurement process. This will allow setting the level at which to measure—for example, measuring the productivity of the software factory to compare with the market; or of only certain components of the software factory to verify possible flaws that could generate bottlenecks in final production; or of the roles involved in the software development process.

The difficulty in measuring productivity is that there is no consensus about what to measure. It incorporates several factors that in many cases are not taken into consideration (Scacchi, 1995; Asmild, Paradi & Kulkarni, 2006; Moreira, Carneiro, Pires & Bessa, 2010; Yilmaz & O’Connor, 2011; Ondrej, Jiri & Jan, 2012; Cheikhi, Al-Qutaish, & Idri, 2012; Khan, Ahmed & Faisal, 2014). Also, several studies do not define the scope of measurement, making detecting opportunities for improvement in organizations difficult, and raising the research question: “How is productivity measured in software factories?”

This article presents a systematic review of various models that measure productivity in software factories, as well as the factors that affect it. At the same time, these are classified according to the production unit of the software factory involved, in such a way as to enable researchers on the subject to have current information on the studies carried out to date.

This article is organized to begin with an introduction to software factories, followed by presentation of the research methodology. Then, statistics on the publications to date and an analysis of the results of the review lead to an attempt to settle an argument about the facts found in the review of the literature. The final section presents conclusions and suggestions for future research.

**BACKGROUND OF THE SOFTWARE FACTORY**

According to studies by Clements and Northrop (2001), the concept of the software factory brings great advantages to the areas of software development:

- Decrease in production costs per product of up to 60%, saving time to market of up to 98%, and reduction of labor requirement of up to 60%;
Knowledge Worker Profile: A Framework to Clarify Expectations
www.igi-global.com/chapter/knowledge-worker-profile/41691?camid=4v1a

Using Knowledge Management to Create Self-Reliant Communities in Thailand
www.igi-global.com/article/using-knowledge-management-create-self/39091?camid=4v1a