International Journal of Information Technologies and Systems Approach
Volume 13 • Issue 1 • January-June 2020

New Factors Affecting Productivity of the Software Factory

Pedro Castañeda, Universidad Peruana de Ciencias Aplicadas (UPC), National University of San Marcos (UNMSM), Lima, Peru

https://orcid.org/0000-0003-1865-1293

David Mauricio, National University of San Marcos, Lima, Peru

https://orcid.org/0000-0001-9262-626X

ABSTRACT

Productivity is very important because it allows organizations to achieve greater efficiency and effectiveness in their activities; however, it is affected by numerous factors. While these factors have been identified for over two decades, all of the previous works limited the software factory to the programming work unit and did not analyze other work units that are also relevant. 90% of a software factory's effort is absorbed by the software production component, 85% of which is concentrated in the efforts of the analysis and design, programming, and testing work units. The present work identifies three new factors that influence the software factory, demonstrating that the use of rules and events influences analysis & design, team heterogeneity negatively affects analysis and design and positively affects programming; and the osmotic communication affects programming. An empirical study on software factories in Peru, determined that 95% of the influence came from these factors, which corroborated as well that team size and trust within the team influences in software production.

KEYWORDS

Factors, Heterogeneity, Osmotic Communication, Productivity, Rules, Software Factory, Software Production, Trust

INTRODUCTION

Cusumano (1989) defines a software factory as a company whose characteristics include large-scale software production, task standardization, control standardization, labor division, mechanization, automation, and the systematic application of the good practices of software engineering. The software factory offers great advantages, such as the ability to decrease production costs per product up to 60%, the time savings of putting a product on the market up to 98%, labor requirement reductions by up to 60%, the improvement of productivity by approximately 10 times, and the quality of each product with 10 times fewer of the errors. This increases the portfolio of products and services offered and the possibility of winning new markets (Clements & Northrop, 2001).

When measuring productivity, a software factory has indicators that allow for it to be compared in the market in a way that helps with the consideration of actions to increase the overall efficiency,

DOI: 10.4018/IJITSA.2020010101

which will allow for the use of all resources in an effective and efficient way in order to obtain the best possible results. A business needs to know how the organization is performing in relation to previous periods and its competitors and must ask questions such as the following. Is it increasing, decreasing, advancing, or receding? What is the magnitude of this progress or setback? Are the implemented strategies effective?

All models that measure productivity in a software factory consider various elements such as the processes, resources, units of measurement, etc., but often do not consider what affects people and how it impacts on the results of the productivity measurement. For example, the motivation and confidence in the team are factors that could have positive impacts on productivity, while in a demotivated team it could have the opposite effect (Yilmaz & O'Connor, 2011). This is the reason why studies are being conducted to identify the factors that affect productivity. However, the following is evident. (i) The factors identified are oriented to the Programming work unit, which could not be generalized to the software factory, given that it contemplates other work units and each unit has its own particularities. (ii) There are factors that influence productivity in other knowledge domains, but they have not been analyzed in the software factory.

In addition, previous studies have not considered that productivity is dependent upon various factors beyond inputs and outputs that influence the processes and context, among others (Arcudia-Abad, Solís-Carcaño & Cuesta-Santos, 2007; Nomura, Spinola, Hikage & Tonini, 2006).

In this article, the authors introduce new factors that affect productivity in software factories that are supported by theories including language action perspective, transactive memory theory, and good agile practices, such as time-boxing. In addition, we study how these factors affect the work units: Analysis & Design, Programming and Testing. To validate the influence of the introduced factors, 150 responses were collected and assessed of one survey published. The present work is part of a research study on productivity models for software factories.

This article is divided into seven sections. Section 2 presents the literature review about the software factory, productivity, and the factors that affect it. Section 3 details the proposed conceptual model and the elements that comprise it. Section 4 describes the research methodology and includes the strategy applied to obtain the information and analyze the results. Section 5 presents the results of the study. In Section 6, a discussion is established about the findings found in the validation. Finally, the conclusions are presented in Section 7.

LITERATURE REVIEW

Software Factory

Although the term "software factory" began to be used in the 60s and 70s in the United States and Japan, only in the 90s did the specialized literature contain works about its organization. It started with the work of Basili, Caldiera and Cantone (1992), which was the first work that proposed the organization of the software factory that was focused on software development. Li C., Li H. and Li M. (2001) extended the approach of Basili et al. (1992) by considering Capability Maturity Model Integration (CMMI) and International Standarization Organization 9001 (ISO 9001). Kruchten (2004) described the process of software development in terms of disciplines of the Unified Rational Process (RUP). Fernandes and Teixeira (2004) proposed a model that classified the factory according to the scope or phases of development defined in the process, thus offering an idea of what the life cycle of a project should be.

Nomura et al. (2006) proposed a software factory structure based on the reuse concepts of Basili et al. (1992), operational management, organizational division (Fernandes & Teixeira, 2004), the application of software engineering activities cited by Swanson, McComb, Smith and McCubbrey (1991), the improvement of work methods (Cusumano, 1989), and engineering practices cited in the Project Management Body of Knowledge (PMBOK) and RUP. Some authors limit the organization of the software factory to the development of the product, as seen in the works of Basili et al.

An Eco-System Architectural Model for Delivering Educational Services to Children With Learning Problems in Basic Mathematics

Miguel Angel Ortiz Esparza, Jaime Muñoz Arteaga, José Eder Guzman Mendoza, Juana Canul-Reichand Julien Broisin (2019). *International Journal of Information Technologies and Systems Approach (pp. 61-81).*

www.irma-international.org/article/an-eco-system-architectural-model-for-delivering-educational-services-to-children-with-learning-problems-in-basic-mathematics/230305