

# Chapter 6

## Software Engineering Productivity: Concepts, Issues and Challenges

**Adrián Hernández-López**

*Universidad Carlos III, Madrid, Spain*

**Ángel García-Crespo**

*Universidad Carlos III, Madrid, Spain*

**Ricardo Colomo-Palacios**

*Universidad Carlos III, Madrid, Spain*

**Fernando Cabezas-Isla**

*Universidad Carlos III, Madrid, Spain*

### ABSTRACT

*Software engineering productivity has been widely studied, but there are many issues that remain unsolved. Interesting works related to new metrics and more replications of past productivity analysis have emerged, however, in order to fulfill these unsolved issues, a consensus about influencing factors and well recognized and useful sets of inputs and outputs for using in measurements must be reached. In this regard, a clear state of the art may shed light on further research in software engineering productivity, which remains a promising research area. In this paper, general concepts of software engineering productivity along with general issues and recent challenges that need further attention from the research community are presented.*

### INTRODUCTION

Productivity management continues to prove a challenge for IT projects. While the manufacturing industries have designed and tested methods for determining productivity, IT industry lags behind in terms of methods for evaluating the outcomes and predicting the effort required to complete

projects (Dalcher, 2006). The importance of productivity management in this sector is due to be a key factor for knowledge activities (Ramirez & Nembhard, 2004) and a crucial aspect for decision making in the global market (Ross & Ernstberger, 2006). There are two main areas of production in the IT industry that can coexist in an IT project: software and hardware. Focusing on IT software

DOI: 10.4018/978-1-4666-2800-7.ch006

project, primary types of projects are considered: new development and maintenance, in which productivity measurement seems similar but need to be analyzed separately (Premraj, Shepperd, Kitchenham, & Forselius, 2005).

On the one hand, productivity metrics in IT software projects are mainly based on ratios between the size of delivered software and the effort performed to obtain it (Arnold & Pedross, 1998; MacCormack, Kemerer, Cusumano, & Crandall, 2003). Following this direction, Function Points (FP) and Source Lines Of Code (SLOC) are misleading measurements because the overall development may be unproductive even though programming productivity has increased (Lee & Schmidt, 1997). Furthermore, there are some recent proposals such as measuring productivity based on postmortem Function Points (Asmild, Paradi, & Kulkarni, 2006), using multiple size measures (Kitchenham & Mendes, 2004), and specific methods for methodologies such as Object Oriented (Pendharkar, 2006). On the other hand, the factors that influence IT Productivity are widely accepted, i.e., increasing store constraints, timing constraints, reliability requirements, high level languages, team size, requirements volatility, staff tools skills, staff availability, customer participation, and project duration (see for example, Foulds, Quaddus, & West, 2007; Maxwell & Forselius, 2000). Despite of its acceptance, the influence, positive or negative, of some of these factors in IT Productivity is not clear and may vary due to external factors such as the business sector (Maxwell & Forselius, 2000; Premraj, Twala, Mair, & Forselius, 2004) and outsourcing ratio (Tsunoda, Monden, Yadohisa, Kikuchi, & Matsumoto, 2009).

Moreover, metrics based on code only represent the productivity of coding activities, while other activities such as management, analysis, and design stay out of those metrics scope. Therefore, the relation between productivity and non coding activities in software development projects has

not been properly addressed. Finally, the use of code measures such as SLOC is not sufficient by itself; multidimensional inputs must be included in measuring productivity (Mahmood, Pettingell, & Shaskevich, 1996).

In this scenario, the definition of new models, measures and assessment methods may elucidate productivity in IT projects, especially in software development projects as it has been done for software maintenance projects (Banker, Datar, & Kemerer, 1991). In turn, companies need to establish their own productivity metrics in addition to benchmarking their data (Maxwell & Forselius, 2000). Furthermore, the construction of a state of the art about IT productivity regarding software development may result in a solid contribution to be considered by future researches in this area.

In order to shed some light on this area, we present the state of the art of IT projects performance from different standpoints. Firstly, we introduce the general concepts regarding productivity and software engineering productivity. Secondly, we analyze issues that influence productivity measurement in this industry sector. Thirdly, we present the challenges that have been proposed in the research literature and some new ones that we propose. Finally, we finish this article with some conclusions.

## **CONCEPTS**

First of all, it is important to introduce the main concept of productivity. The origins of the term “productivity” traced back to the eighteen century, and was introduced by Quesnay (1766); however, until the middle of the past century, the definitions were blurred. Traditionally, productivity has been defined as the ratio of outputs produced per unit of input (Jefferys, Hausberger, & Lindblad, 1954). This definition fits well in manufacturing paradigms because is based on quantities of standardized and clearly identified units of measurement,

9 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:  
[www.igi-global.com/chapter/software-engineering-productivity/73228](http://www.igi-global.com/chapter/software-engineering-productivity/73228)

## Related Content

---

### Knowledge Management and New Organization Forms: A Framework for Business Model Innovation

Yogesh Malhotra (2000). *Information Resources Management Journal* (pp. 5-14).

[www.irma-international.org/article/knowledge-management-new-organization-forms/1204](http://www.irma-international.org/article/knowledge-management-new-organization-forms/1204)

### Case-Based Learning in Computer Information Systems

Morgan Jennings, Charles H. Mawhinney and Janos Fustos (2005). *Encyclopedia of Information Science and Technology, First Edition* (pp. 368-372).

[www.irma-international.org/chapter/case-based-learning-computer-information/14264](http://www.irma-international.org/chapter/case-based-learning-computer-information/14264)

### Offshoring Entertainment and Media to India

Alyssa D. Schwender and Christopher J.M. Leet (2008). *Journal of Information Technology Research* (pp. 25-40).

[www.irma-international.org/article/offshoring-entertainment-media-india/3707](http://www.irma-international.org/article/offshoring-entertainment-media-india/3707)

### Motivations and Results for CRM Adoption in Large Companies in Portugal

Maria Manuela Cruz-Cunha, João Varajão and Daniela Santana (2013). *Information Resources Management Journal* (pp. 25-36).

[www.irma-international.org/article/motivations-results-crm-adoption-large/73792](http://www.irma-international.org/article/motivations-results-crm-adoption-large/73792)

### The Resource-Based Theory of the Firm: The New Paradigm for Information Resources Management?

Kalle Kangas (2003). *Business Strategies for Information Technology Management* (pp. 129-148).

[www.irma-international.org/chapter/resource-based-theory-firm/6108](http://www.irma-international.org/chapter/resource-based-theory-firm/6108)