Model for Improving Productivity Without Impacting Quality of Deliverables in IT Projects

Sanjay Mohapatra, Department of Management and Information System, Xavier Institute of Management, Bhubaneswar, India
Sreejesh S, Department of Marketing and Strategy, IBS, IFHE University, Hyderabad, Andhra Pradesh, India

ABSTRACT

This research has developed a theoretical model that will help to improve productivity without reducing quality in software projects. It has been observed that while trying to increase productivity, quality gets hit in software projects. However, customer requirements in these projects demand that the projects be delivered on time without having defects or bugs in the final deliverables. Thus, it becomes important to identify variables that would increase productivity without compromising the quality. In the present study, through literature review different variables were identified that would affect both productivity and quality simultaneously. The data were collected from 36 software projects and were analysed to check the relationships between the identified variables, productivity and quality. Using structured equation modelling it was found that level of application complexity, training, level of client support, reusing existing code and quality of document management system significantly impact productivity without compromising quality. The findings can be used in projects engaged in customized software development as well as in commercial software development.

Key Words: IT Projects, Productivity, Quality, Customer Requirements, Factor Analysis, Structural Equation Modelling

INTRODUCTION

A software project is defined as a set of activities which has a starting date, with specific goals and conditions (Berkun, 2005). The project will also have defined roles and responsibilities, a budget for executing the project, a planning, and a fixed end date. The project will also involve multiple stakeholders. These activities are to be executed using hardware and software resources. Brooks (1995) defines software project management as a sub-discipline of project management in which software projects are planned, monitored and controlled. Thus, there are several aspects of software projects such as project planning, project monitoring and control, managing software requirements, risk management, and software engineering life cycle. A software project can be classified into different types of projects depending on the life cycle stages that are followed by it.
In a study NASSCOM (www.nasscom.in, 2010) had said that by 2015, an increase in productivity in software will be worth $145 billion in the U.S and $290 billion worldwide. The study emphasized that there is a need for studying different variables that helps to improve software productivity. For commercial software development projects where large systems and applications are being developed through vendors, productivity plays an important role for both the vendor and the customer. An improvement in productivity in vendor organization results in faster delivery of the application and also the customer can use the delivered application at the earliest to get business benefits. While improving productivity, a check is also made on the quality of the application as final deliverables with defects will run the risk of not getting accepted by the customer. This will result in rework and loss of productivity. Hence, it becomes a business imperative to understand different variables that help in improving productivity without reducing quality. For this reason, in this study, variables affecting both productivity and quality are discussed.

LITERATURE REVIEW

Boehm (1981), in an earlier study conducted with large government software projects as part of a sponsored government programme, found that cost of software projects is inversely proportional to variables that would help improve productivity. He found that high staff capability and low application complexity would improve productivity. However, no detail explanation was provided for measuring staff capability. Lawrence (1981) conducted a study in medium to large organization in Australia. He found that productivity increased with simpler application and better computing environment (CPU speed) being developed. These observations were similar to findings made by Mohanty (1981) and Kemerer (1987). Thadhani (1984), Conte (1986) and Lambert (1984) observed that good computing speed increases productivity.

In a study Boehm (1981), found that productivity of software development projects was influenced by different variables. The model used for cost estimation was called COCOMO which was used for estimating efforts in the software industry with success. The original COCOMO developed by Boehm was improved upon subsequently by many practitioners and researchers. Thus, COCOMO II had been developed as the improved version. The variables that are considered in COCOMO II model were broadly categorized into:

1. **Personnel Variables**: That included analyst capability, experience in developing applications in a particular domain, capability of a programmer, experience in technology, quality of training provided to increase skill level, attrition rate, and experience in working in a domain.

2. **Product Variables**: Similarly included required reliability of software, database size, complexity of software being developed, availability of reusable modules, and using a systematic process (such as ISO, CMM) so that effective documentation is prepared throughout the life cycle of the software.

3. **Platform Variables**: Comprised of database size, availability of computers, speed of computation and finally

4. **Project Variables**: Involved usage of software testing tools, modern programming practices (using software engineering standards), and requirements volatility.

Vosburg et al. (1984) have made a detail study of large software projects being developed in 17 organizations in nine different countries. The researchers classified variables that affected productivity into ‘product’ and ‘production processes’ related variables. The product related variables were ‘computing resource constraints’, ‘program complexity’, ‘customer participation’, and ‘size of program or application’ being developed. The production process related variables that were found to affect productivity were ‘concurrent hardware-
A Context-Based Approach for Supporting Knowledge Work with Semantic Portals
www.igi-global.com/chapter/context-based-approach-supporting-knowledge/10201?camid=4v1a