Decision Support Systems 59 (2014) 15–27

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Organizing knowledge workforce for specified iterative software development tasks

Benjamin B.M. Shao [a], Peng-Yeng Yin [b], Andrew N.K. Chen [c,*]

[a] W. P. Carey School of Business, Arizona State University, Tempe, AZ 85287, United States
[b] College of Management, National Chi Nan University, Nantou, Taiwan
[c] School of Business, University of Kansas, Lawrence, KS 66045, United States

ARTICLE INFO

Article history:
Received 26 November 2012
Received in revised form 20 August 2013
Accepted 3 October 2013
Available online 11 October 2013

Keywords:
Iterative software development
Workforce management
Task assignment
Knowledge management
Simulations
Particle swarm optimization

ABSTRACT

Organizing knowledge workers for specific tasks in a software development process is critical for the success of software projects. Assigning workforce in software projects represents a dynamic and complex problem that concerns the utilization of cross-trained knowledge workers who possess different productivities and error tendencies in coding and defect correction. This complexity is further compounded when the development process follows a software release life cycle and involves major releases of alpha, beta, and final versions in the context of iterative software development. We study this knowledge workforce problem from three essential project management perspectives: (1) timeliness — obtaining shortest development time; (2) effectiveness — satisfying budget constraint; and (3) efficiency — achieving high workforce utilization. We explore ideal workforce composites with two strategic focuses on productivity and quality and with different scenarios of workload ratios. An analytical model is formulated and a meta-heuristic approach based on particle swarm optimization is used to derive solutions in a simulation experiment. Our findings suggest that forming an ideal workforce composite is a non-trivial task and task assignments with divergent focuses for software projects under different workload scenarios require different planning strategies. Practical implications are drawn from our findings to provide insight on effectively planning workforce for software projects with specific goals and considerations.

1. Introduction

Software projects require extensive development efforts and can be quite a challenging undertaking for planning and execution. To cope with complexity, large software projects are typically divided into multiple modules, each of which is individually developed and then tested and corrected through iterations for incremental development and quality improvement [11]. An important aspect in software development is resources that include budget and workforce. The productivities of knowledge workers for different types of tasks are not identical; neither is the cost of employing these people. How to effectively manage knowledge workers (e.g., software developers and testers) for various tasks in the software development process is thus a critical issue for the success of software projects. Workforce management in software projects is a dynamic and complex problem that is concerned with the utilization of cross-trained knowledge workers who possess different productivities and error tendencies in coding and defect correction. This complexity is further compounded when the development process follows a software release life cycle and involves major releases of alpha, beta, and final versions in the context of iterative software development.

Following the "design evaluation" guideline for a design-science study [30,40], we tackle this software project management problem through an experimental paradigm from three essential project management perspectives: (1) timeliness — obtaining shortest development time; (2) effectiveness — satisfying budget constraint; and (3) efficiency — achieving high workforce utilization. We explore ideal workforce composites with two strategic focuses on productivity and quality and with different scenarios of workload ratios. On the workforce management front, we examine how software project managers assemble and allocate workers with different proficiencies in development and testing skills to various tasks in the context of iterative software development (i.e., alpha, beta, and final releases of the software release life cycle) to achieve the shortest total development time, maximize workforce utilization, satisfy budgetary constraint, and meet quality requirements. An analytical model is formulated and a meta-heuristic approach based on particle swarm optimization is used to derive solutions in a simulation experiment.

We explore whether organizations can strike a desired balance by evaluating the tradeoff among abovementioned project management perspectives with appropriate assignment of diverse workers under different strategic focuses and workload scenarios. In doing so, we take the first step to tackle this complicated problem with the belief that effective software project management involves more than deploying workers to tasks by availability but requires a holistic and

* Corresponding author at: Accounting and Information Systems, School of Business, University of Kansas, Summerfield Hall, 1300 Sunnyside Avenue, Lawrence, KS 66045. Tel.: +1 785 864 7529; fax: +1 785 864 5328.
E-mail address: achen@ku.edu (A.N.K. Chen).

0167-9236/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dss.2013.10.002

systematic guideline. Practical implications drawn from our findings can provide insight on effectively planning workforce for software projects with specific goals and considerations.

The intended contributions of this study are as follows. First, while most previous studies look at specific aspects of iterative software development, we consider multiple key factors affecting different metrics of software development performance. We do so by formally modeling a software development problem associated with both project and workforce management. Specifically, our model seeks to minimize the completion time while satisfying budget constraint, meeting quality requirement, and managing worker assignment in the context of iterative software development. Second, we uncover valuable insights of project and workforce management using a contemporary particle swarm optimization (PSO) algorithm. The efficacy of both the proposed model and the PSO algorithm is demonstrated through an extensive simulation. Third, in a larger sense, our endeavor also responds to a recent call for more holistic, multidisciplinary, business-centric, and "macrodesign science" information systems (IS) research that can carry potential high visibility and high impact to organizations and society [16].

The rest of the paper is organized as follows. Section 2 provides background descriptions of iterative software development, project and workforce management, and key factors and perspectives associated with software development. Section 3 explains the context of software development process and our model formulation. Section 4 introduces the particle swarm optimization approach as well as our experiment and simulation settings. Section 5 presents our findings and corresponding implications. Finally, concluding remarks, limitations, and future research directions are offered in Section 6.

2. Background

It is useful to first clearly define some key concepts and terms to delineate the specific context of our study. A *software development process* starts with selecting, designing, and initiating a software project. This process proceeds to the construction and refining iterations and then reaches the release of the final product. In this study, we focus on the phase of "construction and refining iterations" where coding and defect correction occur. A *software development project* encompasses different scales and types that range from a large system development involving conceptual, logical, and physical designs of many interdependent components to producing software modules requiring mostly programming and debugging efforts only. In this study, we refer a software development project to the latter. For example, in the offshore IT outsourcing context, a software project completed by a subcontractor can be only involved in coding and debugging efforts. Another example can be a turn-key application or module which is developed by a consulting company.

In the context of software development, *knowledge workers* refer to developers with different proficiency levels of coding and debugging computer programs. That is, knowledge workers possess different levels of productivities and error tendencies in coding and defect correction, respectively. Therefore, the context of this study lies in knowledge workforce management and project management for software development. Knowledge workers to be assigned are software code producers who follow pre-defined software specifications and instructions to perform specific software development and quality improvement tasks. We examine how software project managers can assemble and allocate workers with different proficiencies in coding and testing skills to a software development project in order to achieve objectives such as the shortest total development time, maximum workforce utilization, satisfying budgetary constraint, and meeting quality requirements.

2.1. Iterative software development

Most modern large-scale software projects are carried out in an incremental and iterative fashion termed the software release life cycle. That is, the incremental construction of software projects proceeds in a series of iterations. Each of the iterations consists of a development phase, followed by a testing–debugging phase [11]. In the development phase, developers work on their assigned modules and perform such tasks as detailed design and coding. In the follow-up testing–debugging phase, developers try to enhance the quality of modules by evaluating the actual output against the expected output and taking corrective actions if any discrepancy occurs in such evaluation. Such an iteration of adding functionalities, inspecting and testing components, and releasing a sequential number of working versions usually repeats for several times before the complete product is finally delivered to the customer or made available on the market [36]. Additionally, Keil [32] emphasized the importance of matching appropriate project management systems to the stages of software projects.

Many technical and managerial challenges exist in developing large-scale software projects that follow the iterative development process. Complexity, innovativeness, and criticality of software systems all contribute to technical challenges [44]. The literature of software engineering and information systems has strongly promoted the use of incremental and iterative development for software projects, as the effectiveness of such development methods has been consistently demonstrated. Practical development paradigms such as iterative process [11], the risk-driven spiral model [10], and rational unified process [38] have been proposed and employed in the industry. Incremental and iterative development facilitates the development of software systems as it reduces the propagation impacts of defective code and produces comparatively stable products in progress that subsequent development can improve upon [44]. On the other hand, managerial challenges arise from the need to coordinate the efforts of project members who possess different skill levels for different tasks. Incremental process models and spiral models based on iterative enhancement have been proposed to manage software development in practice. However, these models provide little guidance for decision makers to effectively manage project and knowledge workforce.

2.2. Software project management and workforce management

Project management refers to the planning and control activities associated with the making of a product or service to ensure that the production process be carried out smoothly to meet certain goals. Managing software projects is a complex undertaking as it involves both technical problems (e.g., data structures, algorithms, program efficiency) and managerial challenges related to workforce, resource allocations, and other administrative aspects [49]. While many new innovative development tools and methods (e.g., agile, object-oriented programming languages, reusable code) have been proposed and employed to enhance the quality and productivity of software development, it is argued that a management-focused approach to improving the practice of software development process is also warranted [4].

Several attributes separate software projects from other types of projects, causing IS practitioners to struggle with effective software project management. First, the dynamics of software construction typically involve iterative and incremental development. These build and test phases can repeat several times, thus complicating its planning and controlling efforts. Second, the lack of effective planning and controlling tools as well as the paucity of accurate timely information exacerbate the dilemma of promoting creativity and exercising disciplined management in software projects. Finally, knowledge workers are oftentimes cross-trained and possess varied levels of skill, knowledge, and experience [15]. Thus, workforce is likely to show different productivity and error tendency in completing different tasks. These productivity and error factors have to be taken into account by project managers, making the planning decisions even more difficult.

Workforce management stems from the human resource management (HRM) literature which encompasses the aspects of practices and strategies for creating, utilizing, and maintaining workforce to