

Chapter 38

Towards an Integrated Personal Software Process and Team Software Process Supporting Tool

Ho-Jin Choi

*Korea Advanced Institute of Science and
Technology, South Korea*

Sang-Hun Lee

*Korea Advanced Institute of Science and
Technology, South Korea*

Syed Ahsan Fahmi

*Korea Advanced Institute of Science and
Technology, South Korea*

Ahmad Ibrahim

*Korea Advanced Institute of Science and
Technology, South Korea*

Hyun-Il Shin

*Korea Advanced Institute of Science and
Technology, South Korea*

Young-Kyu Park

*Korea Advanced Institute of Science and
Technology, South Korea*

ABSTRACT

Personal Software Process (PSP) and Team Software Process (TSP) have been developed and used to help individual developers and teams make high-quality products through improving their personal and team software development processes. For the PSP and TSP practices, data collection and analysis of software metrics need to be done at fine-grained levels. These tasks are not trivial, requiring tool support. This chapter aims to discuss issues to building such a tool, and introduce our on-going endeavor towards an integrated PSP and TSP supporting tool. In particular, features of sensor-based automated data collection for PSP, utilization of Six Sigma techniques into PSP and TSP activities, and incorporation of electronic process guide will be paid attention.

INTRODUCTION

Continuous process improvement has been regarded as a solid solution to make high-quality products at the personal and team levels as well as at the project and organization levels. The

Personal Software Process (PSP) was developed to help individual developers make high-quality products through improving their personal software development processes. PSP provides a set of methods and practices to assist individual software developers to improve product and process quality such as defined and measurable process, size and effort estimation based on historical data,

DOI: 10.4018/978-1-4666-4301-7.ch038

code and design review, precise designs, process quality measures, detailed plan, and earned value tracking. The Team Software Process (TSP) guides team members and managers in applying process principles to consistently produce high-quality products on planned schedule at the team level. In particular, the TSP, along with the PSP, supports process improvement in small organizations which have difficulty in applying the so-called “heavy” process as defined by the Capability Maturity Model Integration (CMMI) because of limited resources to adopt CMMI. Among these methods and practices, the measurement and analysis is a central and core practice in identifying process deficiencies and providing a focus on process improvements. Sets of historical project data are used to make a reliable estimate on effort and quality.

While the PSP was proved to be an effective way to improve the accuracy of effort estimation and to reduce defects (Abrahamsson & Kautz, 2002; Hayes & Over, 1997; Prechelt & Unger, 2001) the manual, paper-based data recording and process guide has been recognized as the major barriers in using the PSP process (Disney & Johnson, 1998; Johnson et al., 2003). Due to the high-overhead and context-switching problems of manual data recording, developers have difficulties in acquiring reliable data, which can lead to misguided analysis results. These problems can be overcome through an automated tool for collecting the PSP data and analyzing the collected data. Since an automated tool cannot collect all necessary data, however, manual data recording should still be supported as well because data errors can be decreased to a few items. To help developers collect more reliable and necessary data, it is therefore required to develop a tool that can support both automated and manual data collection activities. One of the research objectives in this chapter is to investigate ways to automate the data collection activities to support the PSP. To address the problems of context switching and recording overhead which exists in the manual, time-log recording technique as suggested by the

original PSP, we have developed a PSP supporting tool, called *Jasmine*, which uses various software sensors built in the integrated development environment (IDE) to collect data semi-automatically. Then, we have further investigated a way to utilize speech sensor and other information to collect and reason about activity times, which is a crucial step to PSP time recording.

Another aim of our research is to take this PSP supporting tool and add features to support various activities to support the TSP, leading to an integrated PSP and TSP supporting tool. For this purpose, we have surveyed existing TSP supporting tools and compared their pros and cons. The results of the survey are presented in this chapter. Although our endeavor to build such an integrated PSP/TSP tool is still an on-going business, we have made a few achievements thus far. First, we have developed a framework to utilize Six Sigma techniques in the PSP and TSP practices to support process improvement activities at the individual and team levels. Six Sigma techniques in general provide statistical analysis and decision-making supports that can be employed for data analysis, continuous improvement, and process control. On the other hand, only a few analysis techniques have been suggested for the PSP and TSP practices. Deploying Six Sigma techniques within a PSP/TSP supporting tool, therefore, can become beneficial for quantitative analysis to identify high leverage activities, evaluate the effectiveness of process changes, quantify cost and benefits, and control process performance. Second, we have introduced the notion of electronic process guide (EPG) into the design of our PSP/TSP supporting tool to help developers follow the PSP and TSP practices. The original PSP and TSP define a set of processes and provide materials such as scripts, templates, and checklists in a paper-based form which has problems in the usability and maintenance. To allow easy navigation of the PSP process information and to enable storing additional information, we have developed an EPG that enhances the contents and usability of the paper-based process guide.

16 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/towards-integrated-personal-software-process/77733

Related Content

An Energy Efficient Trust Aware Opportunistic Routing Protocol for Wireless Sensor Network

Nagesh Kumar, Yashwant Singhand Pradeep Kumar Singh (2017). *International Journal of Information System Modeling and Design* (pp. 30-44).

www.irma-international.org/article/an-energy-efficient-trust-aware-opportunistic-routing-protocol-for-wireless-sensor-network/199001

TOE Model Analysis for AI Vision Open API

Hyun-Ho Jang, Huy Tung Phuong, Thi-Thanh-Thao Voand Gwangyong Gim (2022). *International Journal of Software Innovation* (pp. 1-16).

www.irma-international.org/article/toe-model-analysis-for-ai-vision-open-api/297509

Applying Niche Theory to Measure Uses and Gratifications of Social Media in Malaysia

Ha Jin Hwang, Haeng Kon Kim, Monowar Mahmoodand Norazryana Mat Dawi (2022). *International Journal of Software Innovation* (pp. 1-11).

www.irma-international.org/article/applying-niche-theory-to-measure-uses-and-gratifications-of-social-media-in-malaysia/289594

Label Propagation Algorithm for the Slices Detection of a Ground-Glass Opacity Nodule

Weiwei Du, Dandan Yuan, Jianming Wang, Xiaojie Duan, Yanhe Maand Hong Zhang (2019). *International Journal of Software Innovation* (pp. 104-118).

www.irma-international.org/article/label-propagation-algorithm-for-the-slices-detection-of-a-ground-glass-opacity-nodule/217395

Toward an Integrative Model of Application-Software Security

Vijay V. Raghavan (2003). *Practicing Software Engineering in the 21st Century* (pp. 157-163).

www.irma-international.org/chapter/toward-integrative-model-application-software/28116