

Chapter 10

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-II 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.

DOI: 10.4018/978-1-61350-141-2.ch010

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, 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 col-

lecting the PSP data and analyzing the collected data. Since an automated tool cannot collect all necessary data, however, manual data recoding 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

17 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/61216

Related Content

An Approach to Discover the Stable Routes in BGP Confederations

Shipra Shukla and Mahesh Kumar (2017). *International Journal of Information System Modeling and Design* (pp. 134-147).

www.irma-international.org/article/an-approach-to-discover-the-stable-routes-in-bgp-confederations/199007

Peer-Based Collaborative Caching and Prefetching in Mobile Broadcast

Wei Wu and Kian-Lee Tan (2010). *Advanced Operating Systems and Kernel Applications: Techniques and Technologies* (pp. 238-261).

www.irma-international.org/chapter/peer-based-collaborative-caching-prefetching/37952

A Formal Language for XML Authorisations Based on Answer Set Programming and Temporal Interval Logic Constraints

Sean Policarpio and Yan Zhang (2011). *International Journal of Secure Software Engineering* (pp. 22-39).

www.irma-international.org/article/formal-language-xml-authorisations-based/52594

An Empirical Study on Filter Bubbles in the YouTube Comments Network: Using Social Network Analysis

Dukjin Kim, Wooyoung Lee, Dohyung Kim and Gwangyong Gim (2021). *International Journal of Software Innovation* (pp. 52-65).

www.irma-international.org/article/an-empirical-study-on-filter-bubbles-in-the-youtube-comments-network/290434

Object-Aware Business Processes: Fundamental Requirements and their Support in Existing Approaches

Vera Künzle, Barbara Weber and Manfred Reichert (2013). *Frameworks for Developing Efficient Information Systems: Models, Theory, and Practice* (pp. 1-29).

www.irma-international.org/chapter/object-aware-business-processes/76616