

PSP-EAT – ENHANCING A PERSONAL SOFTWARE PROCESS COURSE

Daniela Rosca¹, Chao-Ying Li², Kimberly Moore², Mark Stephan², and Steven Weiner²

Abstract *¾ The main objective of teaching the Personal Software Process (PSP) is to develop in students a professional attitude towards producing software. PSP improves performance in size and effort estimation accuracy, software reusability, product quality while maintaining or increasing overall productivity. This work presents the experience of the first author in teaching PSP at graduate level for three years, and a tool, PSP-EAT, we have built to reduce both student and instructor clerical work in learning and teaching PSP. The tool helps also in increasing the students' data collection accuracy and their receptability to the PSP principles.*

PRESENTATION OVERVIEW

Personal Software Process (PSP) is one of the core courses taught in the Masters in Software Engineering Program at Monmouth University. The choice of PSP as one of the first courses students are required to take, is based on the strong principles of a disciplined approach to software development. These principles help students create or grow a professional attitude towards producing software of the highest quality. PSP teaches students skills for increasing the estimation accuracy of the size of a project and of the time necessary to accomplish it, the value of reuse and the benefits of early defect detection and prevention. All these positive results are expected to occur without loss of overall productivity.

PSP comes with great promises, however it has an obstacle in its use: a manual process for the process data input. Students need to calculate and record an overwhelming number of data values for their course assignments, and the instructor has to verify all these values to ensure a good quality process throughout its incremental application. For example, for a single assignment (that involves PSP2.0) students calculate and record more than 500 values, while for all 10 assignments a student would calculate and record approximately 5000 data values. Multiplying this figure with 21 students (the size of a class) gives a justification for why the instructor's time for grading an assignment for this class varied from two to eight hours.

This presentation will begin with a brief overview of PSP, followed by a description of the first author's

experience in teaching this course for three years. The last part of the presentation will introduce the PSP-EAT tool, and will discuss preliminary results of using the tool in class.

The teacher's experience will be presented in relation to issues in teaching PSP to remote sites, creating the motivation for students to be receptive to PSP principles, introducing an exit essay for students to reflect upon the implications of PSP on one's individual attitude towards producing software. Also the use of historical information on students' common mistakes in increasing the quality of data collection and processing is described.

PSP-EAT uses redesigned Excel worksheets that upgrade the worksheets provided by the author of the textbook. These worksheets automate a series of clerical tasks, saving considerable students' time and also allowing them to focus more on the process and "see the forest from the trees". More specifically, the tool helps the students to reduce the number of data values that need to be calculated and recorded by more than 35%. In general, every new concept taught is manually calculated by students in the first assignment where they need to apply it. However, in subsequent assignments, the tool will automatically perform the same calculation. Other typical tasks performed by PSP-EAT are the transfer of data between forms, clerical computations, or the update of the STU1 form.

For the instructor of the PSP class, PSP-EAT offers 3 features: to add a student, to submit an assignment to be recorded in the tool's database, and to ask for a diagnostic report of a specific assignment.

Preliminary results of the use of PSP-EAT in class show that the time spent by students in the postmortem phase was reduced with 18 to 38 percent. Also, we observed that the ratio of the postmortem phase out of the total time spent in developing a program has decreased with 39 to 64 percent. From the instructor's perspective, the tool has helped reduce the grading time per student, from 10 minutes to 3-5 minutes, depending of how much additional work needs to be done for completing the evaluation of a student assignment.

From the data collected in the first semester of using PSP-EAT, the results are encouraging. We will finely tune the tool and make it publicly available. The authors will welcome any suggestions for improvement.

•

¹ Daniela Rosca, Monmouth , Software and Electrical Engineering Department, West Long Branch, NJ 07764, drosca@monmouth.edu

² Chao-Ying Li, Kimberly Moore, Mark Stephan, Steven Weiner, U.S. Army Communications-Electronics Command, Software Engineer Center, Fort Monmouth, NJ 07703