Personal Software Process (PSP) Concept Applied to Beginning Engineers

Lisa Anneberg, Ph.D.

Department of Electrical Engineering
Lawrence Technological University
Southfield MI, USA 48075
(248) **204-2539**
anneberg@ltu.edu
/
Roger Ferguson, Ph.D.

Computer Science and Information Systems
Grand Valley State University
Allendale, MI USA
ferguson@csis.gvsu.edu

ABSTRACT

The Personal Software Process (PSP) was designed to help Software Engineers produce high quality software [1]. PSP helps in the estimating, planning, development of software systems. PSP shows the Software Engineer how to track performance against other related software systems, and most importantly, it shows the engineer that PSP can guide their work so they can produce quality software. As of today, only experienced practitioners of Software Engineering have used PSP. However, the rigor of PSP should help novice engineers better manage their time as they design, develop, test and maintain projects. This paper reports the results of a four month study conduct by the LTU Department of Electrical Engineering, with help from the GVSU Computer Science and Information Systems Department. The goal of this study was to apply PSP ideas to an introductory engineering course.

1. INTRODUCTION

The Personal Software Process [PSP] is a time management methodology specifically adapted to computer and information processes technologies. Here at LTU and GVSU, we are interested in extending PSP to beginning engineers. The course, Introduction to Engineering, is a basic course for all engineering students. The students are in electrical, mechanical, and civil disciplines, and are in their first semester on campus. This course is hands-on, with two student design projects. The first design project is introduced in

the second week of the course. The students design, develop, and present a small web page. The students must also write up a project description to document and accompany the web page.

In particular, a semester of PSP data for the first and second freshmen projects in fall 1998 was collected. The first project is Design Project in HTML for presentation to the class. The freshmen estimate their time for the three phases: planning, implementation, and testing. The freshmen use these results in the prior design project -- which is not software based. The second project is a three person engineering project. The three students design, build, and test an egg drop and transportation system.

Fall 1998 is the first time that students are required to include a PSP time accounting as part of their paper. In particular, the students will log time values for the following design phases:

1. Design [on paper only]
2. Development [including coding and research]
3. testing/maintenance

Fifty-eight students in Introduction to Engineering (Fall 1998) compiled time data for the three listed phases of the first project. The results of this compilation will be distributed to the class as soon as possible, and before the commencement of the second design project. The compilation provides the students a valuable guide to the time that each phase requires. PSP data is then be required as part of the second design project. Again, the compilation of the times was recorded and distributed. The division of time for the three main design phases (above) for the two projects is monitored. With this information, students are expected to spend enough time properly designing the project in order to make development and testing less onerous and time consuming. The hypothesis is that the PSP data will help students in the time management aspect of projects.

As a footnote, LTU and GVSU students use the Personal Software Process [1] methodologies in all relevant courses. In fact, the use of PSP is required for lower levels of courses in the department of Computer Science and Information Systems.

2. PSP OVERVIEW

The Personal Software Process (PSP) is a new IT technology that brings discipline to the practices of individual software engineers, dramatically improving the quality, predictability, and cycle time for software-intensive systems. PSP makes engineers aware of the processes they use to do their work and the performance of those processes. They learn to set personal goals for improvement, measurement and analysis their work, and then adjust their process to meet their goals. In this way, software engineers develop the ability to predict their performance and manage the quality of the work they produce. It is a strategy for professional self-development and enhanced productivity.

The idea of PSP, when introduced with proper training, gives engineers an opportunity to learn and practice PSP before applying it to their project work. Data from the course provides convincing evidence of the benefits of PSP. Typically, engineers show substantial improvement in several areas as shown in [1]. Our results from a sample of 58 beginning engineers that have completed PSP training showed improvement as well.

ADVANTAGES OF PSP

- Improved size and time estimating accuracy.
- Improvements in software quality
- Median productivity improvement

The engineers learn to use data and analysis techniques to determine their performance and to measure the effectiveness of the methods they use. The basic premise is that improved personal process discipline can help to increase the effectiveness of individual engineers. As individual performance improves, it seems likely that software team and project performance will similarly improve.

Each PSP step includes all the elements of subsequent steps together with one or two additions. Introducing these concepts one by one helps the engineers learn disciplined personal methods.

PSP Steps

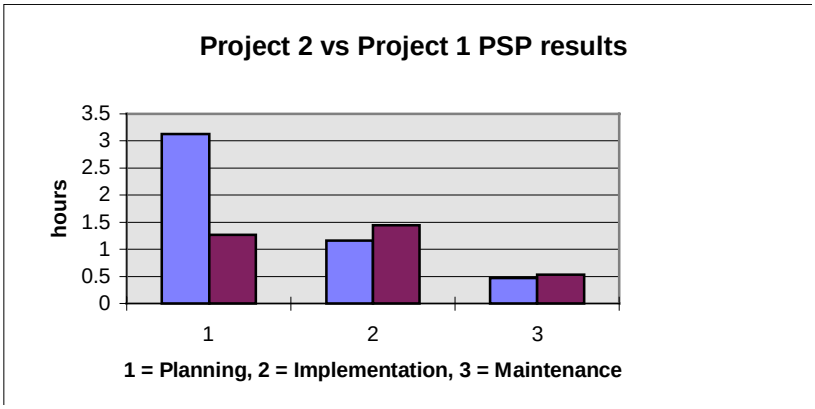
1. Personal Measurement
2. Personal Planning
3. Personal Quality
4. back to step 1

PSP Methods:

- Gathering Data
- Estimating and planning
- Managing Defects
- Managing Yield
- Controlling Cost of Quality
- Understanding Productivity
- Software Design Principles
- Software Process Principles

3. EXPERIMENTAL RESULTS

Fifty-eight LTU freshmen students participated in the first HTML project and recorded the planning, implementation, and maintenance times for the initial project. The fifty-eight students then recorded the planning, implementation, and maintenance times for the second project. The following chart shows the averages of the fifty-eight student's results in hours, comparing Projects 2 and 1:



4. CONCLUSION

PSP's essence is technical time management application to software engineering. This paper extends the usage of PSP to general engineering projects in a freshmen introductory engineering course. The results are very interesting. The students' planning time increased over 100% for the second project, while the implementation and maintenance numbers remained the same. The numerical results of the first project were made available to students. It is likely that the need for proper planning became evident, and students spent more time in the planning stage of the second project. The second project's results were captured on videotape, and show an increased attention to detail. It is hoped that this positive experience will carry through to other projects that the students accomplish. High quality work requires discipline. This small addition to the introductory engineering course will hopefully benefit the students. Future PSP work includes:

1. students' tracking their progress on PSP software integrated in a spreadsheet environment
2. tracking freshmen students for future semesters
3. Incorporating PSP more formally in the entire engineering discipline.
4. assessment of PSP- freshmen in subsequent years versus freshmen with no PSP training

BIBLIOGRAPHIC REFERENCES

[1] W. Humphrey, Introduction to the Personal Software Process, Addison Wesley, Inc., 1997.
[2] N. Leveson, Safeware, System Safety, and Computers, Addison Wesley, Inc., 1995.

Biographic information:
LISA ANNEBERG is an associate professor in electrical and computer engineering at Lawrence Technological University located in Southfield, Michigan. She received her Ph.D. In computer engineering from Wayne State University, Detroit, MI. Her research interests are computer performance models, distributed processing, software engineering, and quality control.

ROGER FERGUSON is an assistant professor in computer science and information systems at Grand Valley State University located in Allendale, Michigan. He received his Ph.D. In computer science from Wayne State University, Detroit Michigan. His research interests are software engineering, software quality models, and software reliability.