Integrating PSP Log Tool in Software Engineering Curriculum

Wook-Sung Yoo
Software Engineering Program, Fairfield University

Abstract

The Online PSP Log Tool is a web-based educational tool designed to help instructors teach and students learn Personal Software Process (PSP). PSP is a rigorous, measurements-based pro-cess designed to help software engineers continually improve their process of producing high-quality software product. In PSP, however, substantial record keeping is required including trans-ferring data between forms, clerical computations, and the update of PSP data between projects. In the past, students have relied on paper forms or spread-sheets with manual data entry. To alleviate the tedious recordkeeping work and improve students' learning of the PSP, the Online PSP Log Tool has been developed and was integrated into the software engineering curriculum to provide students and instructors with streamlined mechanisms to learn and teach PSP. Experimental data were collected and analyzed.

1. Introduction

Software is expanding into all sectors of our society and has become a critical part of many of the systems on which modern society depends. As software systems are getting larger and more complex, they frequently contain defects that result in a lower quality product. Even software engineers with experi-ence can inject around 100 defects into their code for each 1000 lines of code written [1]. A quarter of these defects may remain in the product until final sys-tem testing [2]. Although most software developing organizations today have software quality standards and establish frameworks and processes to guide their work forces, the message is better understood by project managers, pro-ject leaders and quality teams than by software developing teams or individual developers [3]. To scale down the software improvement process to the level of an individual developer, the Personal Software Process (PSP) was devel-oped at the Software Engineering Institute at Carnegie Mellon University in 1995 as a comprehensive framework designed to train individual software engineers to improve their personal software process and quality [4]. In the PSP, individuals gather measurements related to their own work products and the process by which they were developed, and use these measures to drive changes to their development behavior. PSP focuses on defect reduction and estimation accuracy improvement as the two primary goals of personal pro-cess improvement [5]. Through individual collection and analysis of personal data, the PSP provides a novel example of how empirically-guided software process improvement can be implemented by individuals regardless of the surrounding organizational context and the availability of institutional infra-structure support. The data collected in PSP can be also used to resist unrea-sonable commitment demands in an actual working environment [6]. Since the introduction of PSP, several case studies involving experience with and teach-ing of the PSP have been reported [7-14]

The Software Engineering program at Gannon University, Erie, PA introduced the basic levels of PSP (PSP 0 and 0.1) in an introductory programming and data structure courses in 1999. We expected PSP to provide students with a disciplined approach to software development and allow them to experience the value of disciplined development first-hand from their first programming language course. However, there were several problems with incorporating PSP into existing programming courses. The number of forms to be filled out seemed overwhelming for students just learning how to program and students submitted the forms incorrectly and entered unmatched data among logs. More importantly, students in the introductory programming classes did not understand the value of PSP and strongly disliked the Personal Software Pro-cess as reported by other case studies [7, 8]. PSP came with great promises. However, it has an obstacle in its use, which strongly discourages students from adopting the process: a rather tedious manual process for data input. In advanced level of PSP (PSP 2.0, 2.1, 3.0), students are expected to record and calculate more than 500 entries for a single assignment. For

all 10 as-signments assigned during the course, a student would typically record and calculate approximately 5000 data values, at best entering the data by hand, and using spreadsheet support. Additionally, the instructor would thenverify this data for every student to ensure a good quality process throughout its incremental application. Our process metrics for PSP instruction show an av-erage of two to eight hours of the instructor's time was spent grading each assignment when students collect data using the manual process (spread-sheet/paper). To alleviate these problems, and to attempt to improve the teaching and learning of PSP, Gannon University developed the Online PSP Log Tool in 2001 and used it for a dedicated PSP course offered to advanced undergraduate and graduate students as one of the core courses in the Soft-ware Engineering program at the Department of Computer and Information Science [15]. The Online PSP Log Tool is a web-based application, developed in ASP with an SQL database, used to enhance the teaching and learning of the PSP developed. The design of the tool provides instructors and students with streamlined mechanisms for tracking progress in improving their personal processes. The tool was later redesigned using .NET technology at the Soft-ware Engineering program at Fairfield University, Fairfield, CT to include ad-vanced features which are described in following sections. The tool is current-ly used to support the updated software engineering curriculum of the Soft-ware Engineering program at Fairfield University. This paper describes the PSP and currently existing tools supporting it, explains the technical details of the Online PSP Log Tool and associatedexperimental results, and concludes with a discussion of the future direction for the Online PSP Log tool.

2. Overview of PSP

The PSP is a rigorous, measurements-based process designed to help soft-ware developers continually improve their process of producing high-quality software products. PSP teaches students skills for estimating development time, size, and defects with accuracy, the value of reuse, and the benefits of early defect detection and prevention.

There are seven process levels (Level 0, 0.1, 1.0, 1.1, 2.0, 2.1, 3). Each new level introduces new elements and more complicated material supported by more extensive metrics to monitor the process until the engineers reach the highest level, PSP 3 [16]. PSP 0 is the first level and covers how to record development time and how to log each compile-and-test defect. These meas-urements are used in process analysis and planning and as a benchmark for assessing improvement. PSP 0.1 level adds size measurement and the pro-cess improvement proposal (PIP), a form to record the process problems en-gineers encounter and their ideas. PSP 1 introduces the PROBE method, a regression-based size-estimating method which uses historical data to esti-mate size and determine the accuracy of the estimate. PSP 1.1 adds resource and schedule management estimating and earned-value tracking. Earned-value tracking allows engineers to weight the relative importance of each task and to judge their progress with multiple check points. PSP 2 introduces de-sign and code reviews, as well as quality measurement and evaluation. PSP 2.1 uses design specification techniques and ways to prevent defects. PSP 3 is the highest process level, where software engineers become fully conver-sant in PSP covering design verification techniques and methods for adapting PSP to engineers' working environments. Table 1 shows a summary of the forms and scripts that support the various PSP levels which provide a pro-gressive introduction to the concepts and techniques used in the PSP.

Table 1. PSP form related to PSP level

Forms and Scripts	PSP Level						
	0	0.1	1	1.1	2	2.1	3
Project Plan Summary	X	X	X	X	X	X	X
Time Recording Log	X	X	X	X	X	X	X
Defect Recording Log	X	X	X	X	X	X	X
Process Improvement		X	X	X	X	X	X
Test Report Template			X	X	X	X	X
Size Estimating Template			X	X	X	X	X
Task Planning Template				X	X	X	X
Schedule planning Template				X	X	X	X
Operational Scenario Template						X	X
Functional Specification						X	X
State Specification Template						X	X
Logic Specification Template						X	X
Issue Tracking Log							X

Over the duration of a PSP course, each student is given 10 programming assignments. Each assignment is associated with a particular PSP level using a sequence of seven increasingly sophisticated software development pro-cesses from PSP level 0

to level 3. The forms in each PSP level are used to collect metrics data while the student is completing each assignment. The students record various measurements related to their personal development activities. For example, if the first program assignment is associated with PSP level 0, the time spent in each phase of development is gathered through the Time Recording Log, the numbers of defects injected and removed during each phase is recorded in Defect Recording Log, and the size of the program is entered in the Project Plan Summary. The data gathered through the first project establishes a baseline set of process measures for time, size, and de-fects and later programs extend these baseline process statistics by employ-ing more advanced processes levels. By going through 10 program assign-ments, students gather the data and analyze them to improve the quality of the software process at a personal level. The PSP strategy is to improve the performance of practicing software engineers and to enable individuals to take ownership of the process.

3. Logging PSP Data and Tools

The PSP was originally developed to collect data in a manual process. Users would print out forms to log effort, size, defects and other information. However, the high overhead of data collection and analysis have been an obstacle to the use of PSP. Due to the critical nature and tedium of the data collection processes vital to the teaching and practice of the PSP, several major at-tempts have been made to develop and deploy freeware support tools [17-26]. Some applications use spreadsheet tools to record data, while others use more sophisticated stand-alone applications to keep track of data in the soft-ware development process. A number of key features distinguish the PSP tools available to educators, specifically the type, or architecture, data storage method, platform, user support, form support, purpose, interface, metrics sup-port and instruction support. Table 2 provides a summary of the tools compar-ing features. Commercial tools were excluded from the comparison, as their cost for adoption is prohibitive for most teaching environments.

Table 2. Feature comparison of current PSP tool

	Online PSP Log Tool	PSP Support	PSP Academic Material
Distribution	Web Based	Spread Sheet	Access DB
Data Storage	Web accessible database	Files	Local files
Platform	Windows	All	Windows
User Support	Multiple user	Single user	Single user
PSP Forms	All	All	All
Editable Forms	All	All	All
Chart Flexibility	Yes	N/A	No
Purpose	Teaching	Either	Academic
Metrics Calculation	Automatic	Manual	Automatic
PSP levels	All PSP levels	Basic PSP	Up to 2.1
Instructor Support	Yes	No	No
User Help	Included	Manual Only	Manual Only

Name	PSP Studio	Process Dashboard	Jasmine	Hackystat
Distribution	Stand-alone	Stand-alone	Stand-alone	Stand-alone
Data Storage	Local DB	Local Files	Sensor-Based-Repository	Sensor Base Repository
Platform	Windows	All	MS office , Eclipse	Eclipse/Visual studio
User Support	Single user	Single user	Single user	Single User
PSP Forms	All	All	No	No
Editable Forms	All	All	No	No
Chart Flexibility	No	Yes	No	No
Purpose	Education	Either	Programmer	Research eeducators
Metrics Calculation	Automatic	Automatic	Automatic(sensor-base)	Automatic(sensor-base)
PSP levels	All PSP levels	All PSP levels	All PSP levels	Up to 2.1
Instructor Support	No	No	No	No
User Help	Manual Only	Included	Included	Manual Only

As shown in Table 2, all tools examined provide advantages over the paper process. Most PSP tools

provide the electronic version of PSP forms, allow users to edit the forms, and automate metrics calculation for all of the PSP levels. Several features are particularly distinguishing for effective teaching use, most notably distributed operation, multi-user support, chart flexibility, and instructor support. For example, stand-alone applications are limited in their effectiveness in classroom environment. Students using a stand-alone tool need to retain the file(s) for each assignment and submit it to the instruc-tor. PSP Studio and Process Dashboard were developed to reduce data gath-ering and analysis. Hackystat collects metrics automatically by attaching sen-sors to development tools eliminating the overhead of metrics collection and context switch between working and data input. However, it changes the na-ture of the metrics that are collected, and raises new adoption issues related to privacy. Jasmine is a tool similar to Hachystat. It collects and analyzes software metrics and PSP data, using an architecture based on plug-ins that automatically collects data from development tools. It is integrated into the Eclipse environment and implemented as an Eclipse plug-in providing func-tions similar to the PSP Studio and the Process Dashboard. Such tools are convenient for developers using an Eclipse environment to practice PSP as the developers do not need to switch to another environment to gather metrics and perform data analysis. An Eclipse plug-in for PSP can assist PSP training in universities and industries. However, there is no or little instructor support in most tools and the instructor has to load the files individually, and review and grade the assignment. An undesirable side effect of the standalone/paper da-ta collection/submission processes is that the student data is typically not available for review until the assignment is completed, with the resultant missed opportunity to help the student improve their work prior to submission. These observations led to the development of the Online PSP Log Tool, with the Web chosen as the most effective means of distributing the system to both students and faculty.

4. Online PSP Log Tool

The Online PSP Log Tool is partitioned into five system-level components. Four directly support the PSP for students and faculty (PSP Online Forms, Instructor's Tool, PSP Chart, and Discussion Board), while the fifth serves to integrate the components (Central Database). Figure 1 presents an overview of the system components. These subsystems are described in the subse-quent subsections.

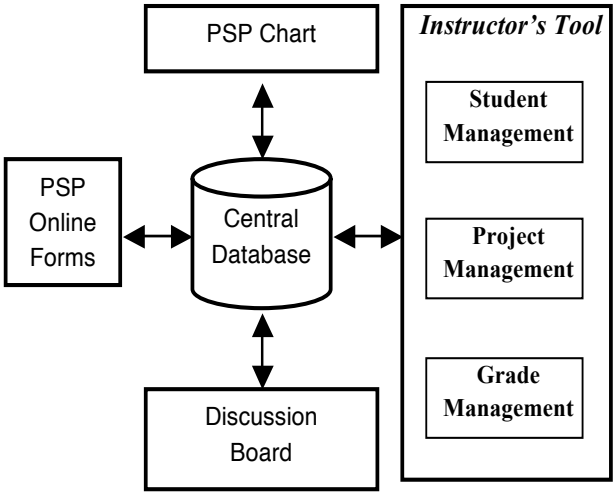
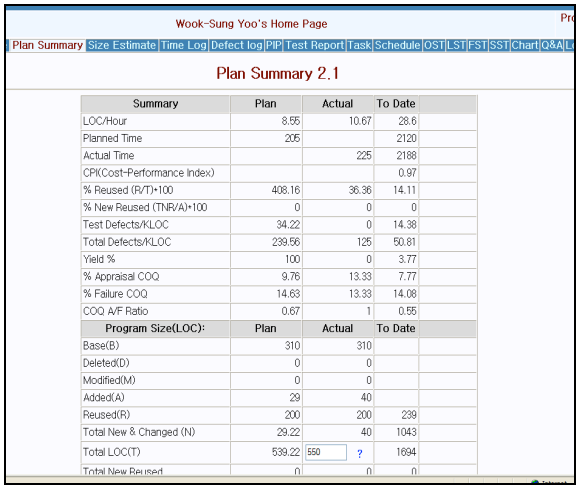


Figure 1. System Overview of PSP Log Tool

4.1. PSP online forms

The PSP Online Forms is the key front-end component used fordata collection that supports the student's PSP work. Implemented as dynamic web forms organized in a single site, this site is password protected so that students can access the PSP form only after a login ID and password are provided by the instructor. Once students log in to the system, they can access the PSP forms for each assignment matching to its corresponding PSP level. A snapshot of the Plan Summary form for PSP Level 2.1 is displayed in Figure 2. This form depicts the planned and actual times, and other computed development met-rics, in the case for PSP Level 2.1. It also shows the tabs to access all of the forms available at this PSP level, for this user. Since the user depicted is an instructor, it also provides a tab link to three additional forms that specifically support teaching and learning: The instructor's tool, charts, and discussion board.



Summary	Plan	Actual	To Date
LOC/Hour	8.55	10.67	28.6
Planned Time	205		2120
Actual Time		225	2188
CPI(Cost-Performance Index)			0.97
% Reused (R/T)*100	408.16	36.36	14.11
% New Reused (TNR/A)*100	0	0	0
Test Defects/KLOC	34.22	0	14.38
Total Defects/KLOC	239.56	125	50.81
Yield %	100	0	3.77
% Appraisal COQ	9.76	13.33	7.77
% Failure COQ	14.63	13.33	14.08
COQ A/F Ratio	0.67	1	0.55
Program Size(LOC):	Plan	Actual	To Date
Base(B)	310	310	
Deleted(D)	0	0	
Modified(M)	0	0	
Added(A)	29	40	
Reused(R)	200	200	239
Total New & Changed (N)	29.22	40	1043
Total LOC(T)	539.22	550 ?	1694
Total New Reused	0	0	0

Figure 2 Snapshot of Plan Summary Form of PSP

Students can update the PSP log data for each assignment multiple times. Once the final submission is made, however, the students cannot modify the data, and the instructor can proceed to grade the assignment. As the tool was designed to support the full set of PSP forms, the online forms and scripts related to seven different PSP levels are available in Online PSP Log Tool. The tool also helps students to understand the PSP better and forces students to follow the proper process. For example, students cannot start to enter the time log or defect log until estimated time is entered in the Plan Summary form. In paper form, students can enter estimated time at any time even after the project is completed. The total time and defect of all phases are automati-cally calculated and updated in the Plan Summary form as students enter the data in the Time Log and Defect Log forms so that students cannot and do not need to enter the time and defect in the Plan Summary form manually. The Online PSP Log Tool also provides various online help such as tooltips. By clicking or hovering the mouse over the "?" icon next to each field in the form, students can see the description of the field and how to fill out the certain field in the form. The newly redesigned Online PSP Log Tool includes features for students to download user's individual data for midterm and final reports and automatic false-data detection function, which helps in reducing overhead and mistakes instudents' work.

4.2. Instructor's tool

The Instructor's Tool was designed to support the instructor with three main functions: Student Management, Project Management, and Grade Manage-ment

4.2.1. Student Management. ith Student Management, the instructor can create, update, and delete stu-dent accounts, and setup passwords. Instructors can create an account for other instructors or teaching assistants to give similar privileges to others sharing in the execution/ administration of the course.

4.2.2. Project management. Project Management allows the instructor to setup different PSP levels for each project. Once a certain PSP level is assigned to each project assign-ment, all forms and scripts related to the PSP level become available for the project automatically to students. Until the instructor assigns the PSP level to the project, students cannot access the forms for that project.

4.2.3. Grade management. rade Management provides a convenient way for instructors to review and grade a student's project. Instructors can view asummary of the current status (completed, in-progress, not started) of a project. Instructors can view whole projects by a certain student or students' project progress on a certain project for the whole class. Then, instructors can go through the details of the PSP forms filled in by individual students on a specific project. Instructors can then assign individual grades and comments for a finished project. Only the instruc-tor and the specific individual student can view the grades and comments.

4.3. PSP chart

Analysis of the data entered by students on each assignment is valuable in-formation for students to understand their performance and to be motivated [4]. However, gathering all data and creating various charts is a very time consuming job for the instructor. The Online PSP Log Tool provides chart tools for instructors to visualize various dimensions of student performance data using graphs. The current implementation of the tool provides 10 differ-ent measurements of the PSP data including time estimate, time actual, de-fect log, %time error, %size error, defect/KLOC, (compile + test) / project, %yield, defects/hour, and COQ A/F ratio. Figure 3 displays a snapshot of sample time estimate chart in instructor's view. The chart shows the class's min, max and average score for each category. The

instructor can review the chart with students in the class, so they can see where they fit in their pro-cess, and learn from the class's collective experience. For each student, the chart will display student's own score with the class's min/max/average score for each category so that the student can privately compare his/her data with other students'. The user can select the line graph, bar graph or 3-D graph to view the data in different way.

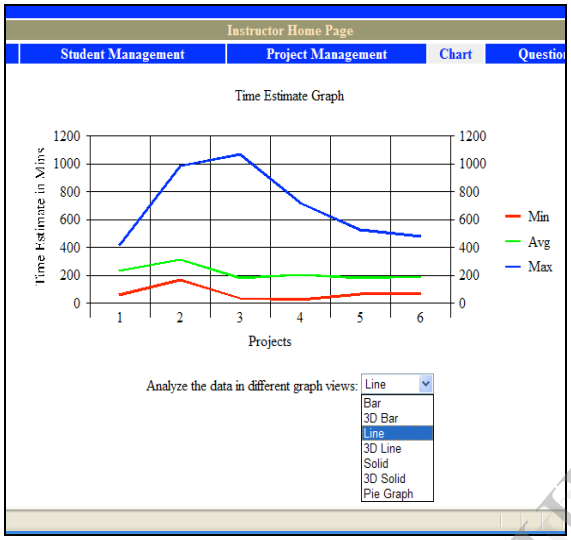


Figure 3 Class-wide Charting Support

One of the features in the updated Online PSP Log Tool developed in .NET framework is the refined charting functionality which is already included in .NET framework to the various programming languages without any extra software.

4.4. Discussion Board

The Discussion Board allows a student to share information, discuss issues, or submit questions to their instructor. The instructors answer students' ques-tions and publish some selected questions to the Q&A section. When students submit any question, it is saved in the database while the instructor receives an e-mail notification of new questions. After the instructor reviews the ques-tion and decides whether or not the question is valuable for the class, or simp-ly warrants a private response. The instructor can edit the question and an-swer, and submit it to the Q&A section. The question and answer will be listed in Q&A section for all students in the class toview it. Like the help feature, the Q&A section helps students understand PSP.

5. Experimental Results

In this section, we will present the experimental results formulated from stu-dent feedback and data gathered from four instances of the PSP class over the school years 2001 – 4 at Gannon University and 2012 – 13 at Fairfield University. These results are compared with the manual data collection pro-cess from control groups. Unfortunately, similar productivity improvement data was not published with information on other tools currently available.

After the Online PSP Log Tool was successfully implemented, the tool wa-sused in Personal Software Process course at Gannon University during the period 2001 – 4. The primary effectiveness measure of the tool's benefit was the time spent in the postmortem phase. To measure this effectiveness, an experiment was developed where the course instance would provide for both an experimental group and a control group. One group used the Online PSP Log Tool and the other group used the paper version of PSP forms as per the text. To be fair, after the completion of the first five out of ten assignments, the group members were swapped so that students who had used Online PSP Log Tool for first five assignments used a paper form for the last five assign-ments. After the tool was redesigned using .NET at Fairfield University, the same experiment was conducted in Software Engineering courses for five projects at PSP level 0 – level 2.0 in 2012-13 to gather the data. Table 3 shows the summary of the average time (minutes) spent in the postmortem phase of ten projects between two groups.

Table 3. Postmortem time comparison (muninutes)

	2001	2002	2003	2004	2012	2013
PSP Tool	23	18	25	17	15	13
Paper	36	34	42	33	27	25
Improvement	36.1 %	47.1 %	40.5 %	48.5 %	44.4 %	48.0 %

The results presented in Table 3 show that the Online PSP Log Tool saves time consistently at the average of 44.1% over paper forms. Qualitative stu-dent feedback confirmed the productivity experienced.

At the end of the semester, we also asked students to complete a survey comparing the paper version vs. the Online PSP Log Tool to evaluate which was preferred. On average, 90% of all students preferred using the Online PSP Log Tool to the paper form.

More importantly, students in recent years (2012-13) showed almost no preference for the paper form. Table 4 presents the summary of tool preference responses.

Table 4. Tool preference responses

	2001	2002	2003	2004	2012	2013
PSP Log Tool	80%	88%	91%	90%	95%	98%
Paper Form	12%	4%	4%	4%	1%	0%
No Preference	8%	8%	5%	6%	4%	2%

In the early 2000s, some students preferred the paper format mainly due to the unavailability of an Internet connection outside of the classroom. Some students had to put the data in paper form and enter PSP data to the online form later. However, with more prevailing Internet connections, Table 4 shows a significant increase in student preference for the online form over paper form. This parallels the continuous updates to the tool based on the students' feedback.From the instructor's perspective, the tool has helped reduce the instructor time significantly. Table 5 shows the summary of instructor's average grading time per student for 10 PSP assignments (set A) in 2001-2004 and 5 PSP assignments (set A) in 2012-2013. This tool reduced the instructor's grading time by 32 % on average and showed more time saving on more projects with higher level of PSP.

Table 5. Instructor's average grading time (minutes)

	2001	2002	2003	2004	2012	2013
PSP Tool	168	115	108	109	86	92
Paper	209	176	182	191	119	127
Improvement	19.6 %	34.7 %	40.7 %	42.9 %	27.8 %	27.6. 7%

Additionally, the reduction in other class preparation time such as analysis of students' data and creating charts is even more significant. These manual activities have been reduced to 0 minutes from 27 minutes on average. The tool virtually eliminated instructor's time of transferring and analyzing students' data.

Besides the tedious and time-consuming process of transferring data, while using paper forms, the student processes were not transparent to the instructor. Since most of the students were not familiar with the process of record keeping of PSP, they were often found to have inserted incorrect log data. The result is unreliable data which then must be corrected for use with the next steps. Sadly, some students tended to be tempted to enter false data because they lacked confidence in the effectiveness of PSP itself. This issue can be resolved by facilitating the interaction between the instructor and students, by monitoring the process of student's work in detail, and the instructor making proper comments in real time. Although not measured directly, anecdotal evidence indicates that the online help embedded in the forms contributed to the reduced error rate experienced by students while filling out the form.

6. Conclusion

The key features of the Online PSP Log Tool have been successfully imple-mented in intranet environment using ASP technology and .ENT framework. The tool has been effectively used in the teaching and learning of the Person-al Software Process, with significant (90%) student preference for tool use. The Online PSP Log Tool not only helps in learning PSP, but also helps in effective budgeting of both the students' time and the instructor's time. The tool also helps in maintaining data quality and data availability. We are plan-ning to gather more data from future software engineering classes at Fairfield University and share the tool with other schools for comprehensive compari-son of current PSP tools and their usage. We also plan to develop an online tool for Team Software Process (TSP) to be integrated with Online PSP Log Tool.

7. Acknowledgement

Special thanks to graduate students who have helped us develop this tool and gathered data..

8. References

[1] Gibson, "Applied Software Process Improvement," Proceedings of the Americas Conference on Information Systems. pp. 596-598, 1997.

[2] S.K. Lisack, "The Personal Software Process in the Classroom: Student Reactions (An Experience Report)," Thirteenth Conference on Software Engineering Education & Training, p.169, 6-8 March 2000.

[3] S.Tadwalkar, "Is your software process staying with you?," Ubiquity, vol. 2002, June 18, 2002.

[4] W.S. Humphrey, A Discipline for Software Engineering. New York: Addi-son-Wesley, 1995.

[5] W.S. Humphrey, PSP(sm): A Self-Improvement Process for Software En-gineers. Addison-Wesley, 2005.

[6] W.S. Humphrey, "Using a defined and measured personal software pro-cess," IEEE Software, vol. 13, pp.77-78, May 1996.

[7] A. Disney & P. Johnson, "Investigating Data Quality Problems in the PSP," Proceedings of the 6th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (SIGSOFT'98), pp.143-152, 1998.
[8] W. Hayes & J. Over "The Personal Software Process (PSP): An Empirical Study of the Impact of PSP on Individual Engineers," CMU/SEI-97-TR-001, Dec.1997.
[9] R.L. Glass, Software Runaways: Lessons Learned from Massive Software Project Failures. Prentice Hall, 1997.
[10] P. Abrahamsson & K. Kautz, "Personal Software Process: Classroom Ex-periences from Finland," Proceedings of the 7th International Conference on Software Quality, p.175-185, June 09-13, 2002.
[11] R. O'Connor & G. Coleman, "Strategies for personal process improve-ment a comparison," Proceedings of the 2002 ACM symposium on Ap-plied computing, pp. 1036-1040, March 11-14, 2002.
[12] P. Ferguson, W.S. Humphrey, S. Khajenoori, S. Macke, and A. Matvya, "Introducing the personal software process: Three Industry Cases," IEEE Computer, vol. 30, pp. 24–31, May 1997.
[13] A.C. Ceberio-Verghese, "Personal Software Process: A User's perspec-tive," Proceedings of the Ninth Conference on Software Engineering Edu-cation, pp. 52-65, 1996.
[14] A.Wesslén, "A Replicated Empirical Study of the Impact of the Methodsin the PSP on Individual Engineers," Empirical Software Engineering, vol. 5, issue 2, pp. 93-123, June 2000.
[15] W. Yoo, S. Frezza, and C. Park, "PSP Web-based Log Tool in a Teaching Environment," Proceeding (470) Computers and Advanced Technology in Education, 2005.
[16] P.Ferguson, et. al, "Results of Applying the Personal Software Process," Computer, vol. 30:5, pp. 24-31, 1997.
[17] M. Postema, M. Dick, J. Miller & S. Cuce, "Tool support for teaching the personal software process," Computer Science Education, vol. 10, no. 2, pp. 179-193, 2000.
[18] J. Henry, ETSU PSP Studio User's Manual. http://www-cs.etsu.edu.1997.
[19] M. Pomeroy-Huff, R. Cannon, T.A. Chick, J. Mullaney &W. Nichols, "The Personal Software Process SM (PSPSM) Body of Knowledge, Version 2.0," http://www.sei.cmu.edu. 2009.
[20] J. Van Buren "STSC A Review of Personal Software Process (PSP) SM and Team Software Process (TSP) SM," http://www.stsc.hill.af.mil. 1999.
[21] M.H.N.M Nasir, A.A. Norman & N.H. Hassan, "Towards a Flexible Tool for Supporting Data Collection & Analysis in Personal Software Process (PSP)," WSEAS Transactions on Information Science and Applications, vol. 5, no. 6, p.1067-1076, June 2008.
[22] M. Maurizio, "Applying the PSP in Industry," IEEE Software, vol. 17 no. 6, pp. 90-95, November 2000.
[23] Inien Syu, et al, "A Web-Based System for Automating a Disciplined Per-sonal Software Process (PSP)," 10th Conference on Software Engineer-ing Education & Training, pp. 86-96, April 1997.
[24] H. Hassan, M.H.N.M. Nasir, S.S.M. Fauzi. "Incorporating software agents in automated personal software process (PSP) tools," 9th International Symposium on Communications and Information Technology, pp. 976-981, 2009.
[25] Towhidnejad, M. ; Hilburn, T., "Integrating the Personal Software Process (PSP) across the undergraduate curriculum," 27th Annual Conference Frontiers in Education Conference, pp. 162 - 168 vol.1, 1997.
[26] I.D. Coman, "An analysis of developers' tasks using low-level, automati-cally collected data," Proceedings of the the 6th joint meeting of the Euro-pean software engineering conference and the ACM SIGSOFT symposi-um on The foundations of software engineering, pp. 579-582, 2007.