Using PSP to Evaluate Student Effort in Achieving Learning Outcomes in a Software Engineering Assignment

Brian R. von Konsky   Jim Ivins   Mike Robey

Curtin University of Technology
Department of Computing
GPO Box U1987, Perth 6845
Western Australia

{bvk, jim, mike}@cs.curtin.edu.au

Abstract

The goal of this study was to measure the effort expended by students during a major assignment in a third year software engineering subject. The purpose was to evaluate whether students were expending effort on activities not related to the stated learning outcomes, and to determine whether the assessment pattern and assignment scope were appropriate. The principal learning outcome was the ability to model system state using the Unified Modelling Language, Ward and Mellor Data Flow Diagrams, and Z. Another outcome was the ability to show that system models expressed in these notations were valid and consistent. Students kept Personal Software Process (PSP)[SM] logs to record effort expended on all assignment activities. Student opinions regarding learning outcome attainment and the accuracy of PSP data were evaluated using an anonymous questionnaire. A total of 148 students reported spending an average of 24.9 hours working on the assignment and achieved an average mark of 62.6%. Bachelor of Engineering (Software Engineering) students generally achieved a better mark, while expending less effort than Bachelor of Science students studying Computer Science or Information Technology. Surprisingly, however, there was no correlation between effort and mark. Excessive time recorded in the PSP logs of some students, the large standard deviation (s = 12.6 hours), and the large number of outliers in the data suggest that many students either did not take the PSP seriously, or did not use time efficiently and were distracted by factors unrelated to the intended learning outcomes. Other potentially more efficient modes of assessment and feedback are discussed.

Keywords: Learning Outcomes, PSP, UML, DFD, Z.

1 Introduction

This paper describes recent experience in a third year software engineering subject that expands on modelling notations introduced in second year subjects. The third year subject introduces real-time aspects of UML (Douglas, 2000), Ward and Mellor Data Flow Diagrams (DFD) and State Transition Diagrams (STD) for modelling data flow, state and control (Ward and Mellor, 1985), and Z to facilitate formal reasoning about system models (Nissanke, 1999).

The final mark for the subject was computed from a mid-semester test (20%), an assignment (20%) and a final examination (60%). In the assignment, students were given a natural language specification for a simple problem, for which the model solution consisted of only a small number of states as illustrated in Figure 1. Students were required to produce system models from this specification using UML, Ward and Mellor DFD and STD, and Z. Students were then required to show that models in the various notations were consistent, and to use Z to verify that there were no conditions giving rise to violations of state invariants.

In semester 1, 2004, a total of 162 students enrolled in the third year subject. There were 158 undergraduates from a number of degree programs including 22 BEng (Software Engineering) students, 63 BSc (Computer Science) students, and 65 BSc (Information Technology) students.

Prior to 2004, anecdotal evidence suggested that some students were spending far too much time on an assignment of relatively small scope. A study was therefore undertaken to determine how much time was actually being spent on the assignment. Students were required to log time associated with all assignment activities, following a procedure based on the Personal Software Process (PSP)[SM] (Humphrey, 2000). The goals were to determine how students spent time working on the assignment, to evaluate whether students spent time on activities not related to the stated learning outcomes, and to determine whether a revision to the assessment pattern was warranted.

2 Background

Humphrey (1995, 2000) introduced the PSP as a means of helping students and professional software engineers to manage time and make accurate estimates for the duration of future activities based on historical data that they collect. (For further information about the PSP see also Ferguson *et al.* 1997.) When following the PSP, students record the start time, stop time, and the total duration for each activity, excluding time spent on interruptions (unrelated phone calls, coffee breaks, *etc.*).

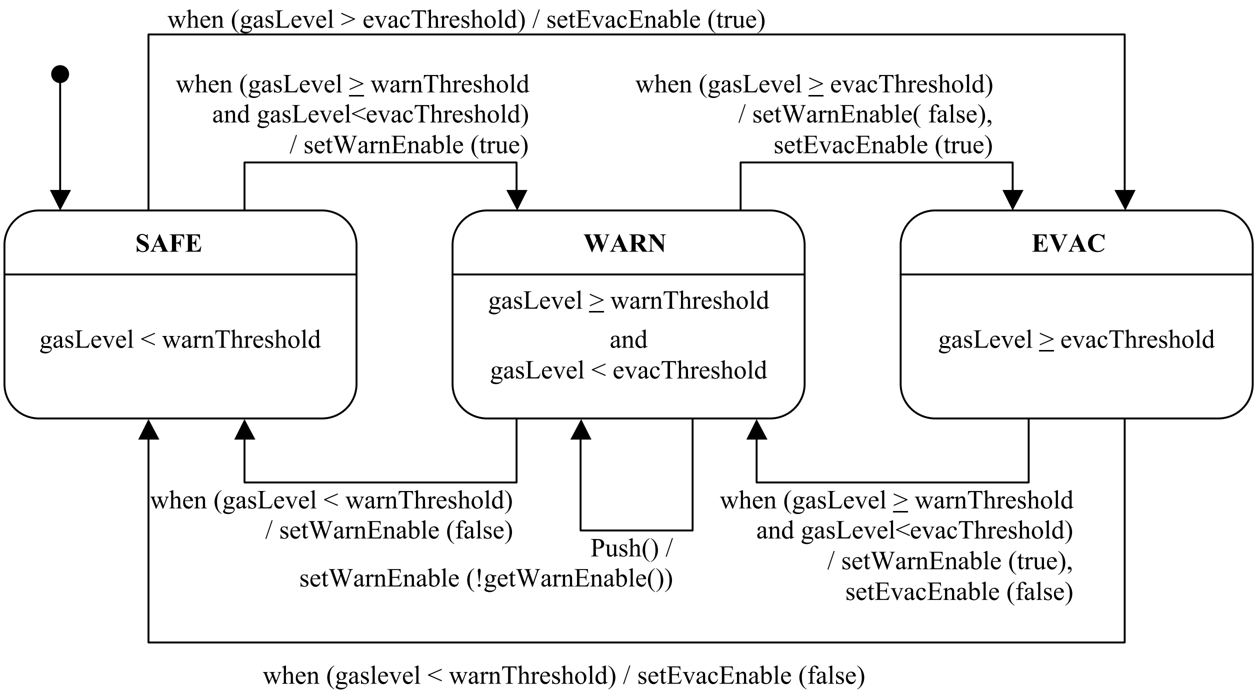

Figure 1. The UML State Chart model solution was of small scope and consisted of only three states. The DFD/STD and Z model solutions were of similar complexity.

If applicable, the size (lines of code written, number of pages read, *etc*.) of each artefact produced or consumed by the activity is also recorded. From the PSP log data, the actual time spent on each activity, and the associated productivity (size/duration) can be calculated. This information is useful in planning how long similar activities will take in the future.

One outcome of following the PSP is that students generate data regarding the amount of time wasted by interruptions and other non-productive activities (Humphrey, 1995, pp. 39–43). However, to completely realize this benefit, interruptions must be measured at the time they occur instead of being estimated after the fact. Unfortunately, this takes discipline that many students lack without sufficient practice or training.

The collection and analysis of PSP data can be tedious and error prone. Researchers have therefore developed automated tools to assist with data entry and analysis (Syu *et al.* 1997; Sillitti 2003). Nevertheless, the introduction of errors during data collection must still be considered (Johnson and Disney 1998).

In addition to tracking effort, those following the PSP collect metrics (Zhong *et al.* 2000) and follow processes shown to improve software quality (Ferguson *et al.* 1997). Consequently, PSP data and metrics have been used in empirical studies before – for example, to show that prior knowledge of the programming language used on a project is not a predictor of software quality in the finished implementation (Wohlin 2002).

This paper describes a study that used PSP to measure the actual time spent (i.e. effort expended) by students working on a major assignment in a software engineering subject. Students recorded the duration and size of modelling and verification activities, following an integrated approach that incorporates multiple modelling notations and formal reasoning (von Konsky *et al.* 2004). In this approach, UML guard and when conditions are modelled in the DFD data flow where events are generated that trigger state transitions in an associated STD. State transitions can generate events that in-turn enable or disable data flow transformations or control external entities. Finally, a Z model is produced to demonstrate that there are no conditions that give rise to violations of a state invariant, and that the model contains no impossible states. The purpose of the integrated approach is to enhance the understanding of each modelling notation and provide a framework for conducting formal reasoning about the model.

3 Method: The Assignment in 2004

The assignment specification required students to generate a UML State Chart, a Ward and Mellor DFD with associated STD, and a Z model for a software-based gas warning and evacuation system controller conforming to the following natural language specification.

- *The software system warns of poisonous gas in a mine.*
- *The software system interacts with a gas sensor capable of detecting the level of gas in the mine.*
- *The software system controls an alarm capable of generating two distinct tones.*

- *The software system sounds an alarm based on the level of poisonous gas detected.*

  Warning Level *– there is no immediate danger, but the software system sounds a warning alarm when this level of gas is detected if the foreman has not disabled it.*

  Evacuation Level *– there is immediate danger so the software system sounds an evacuation alarm when this level of gas is detected. The level of gas detected at the Evacuation Level is greater than the Warning Level.*

- *The mine foreman can push a toggle button to enable/disable the warning alarm.*
- *The mine foreman cannot disable the evacuation alarm.*

The specification is shown here to demonstrate the scope of the assignment. The model solution for the UML State Chart component of the assignment shown in Figure 1 contains only three states. The DFD/STD and Z components of the assignment model the same system, and are therefore of an equivalent scope and complexity.

Students were required to maintain and submit a PSP log recording all assignment related activities. In particular, the assignment specification read as follows.

> *The log should be maintained throughout the time you work on the assignment, rather than being estimated once the assignment has been completed.* ***For each entry you should list one of the following activities in the comments column: Assignment Preparation (including reading and revision), UML Model Development, DFD Model Development, Z Model Development, Process Development, Model Checking/Revision, and Write-up.*** *The log should include columns for the date, the start time, the stop time, length of all interruptions, and the duration excluding interruptions. Duration and interruptions should be expressed in minutes. Log all entries as you work, even if you haven't completed the item. If you forget, log it as soon as you remember, but put a note that you have done this in the comment field along with an indication of when the item was actually added to the log. You will not be penalised for forgetting to log an entry as you go. It is important to log all entries, even if they are estimated later. However, it is important to be able to tell the difference between measured and estimated log entries.*

Students were also required to complete an assignment cover sheet summarising effort associated with each activity listed in bold above. The cover sheet also asked students to summarise metrics regarding the size of each assignment component. Metrics included the number of states, the number of state transitions, and the number of data and event flows.

At the end of the semester, students completed an anonymous questionnaire asking for their opinion regarding the accuracy of the PSP log data, whether they thought the learning outcomes had been achieved, and the general usefulness of the modelling techniques learned over the semester.

4 Results

Overall, students reported spending considerable time on the assignment and on activities not directly related to the defined learning outcomes. At the end of the semester, students thought they had achieved the defined learning outcomes. This opinion was reflected in the marks for the assignment and final examination. These findings are examined in more detail throughout this section.

4.1 Student Demographics

A total of 155 undergraduate students completed the assignment; nine (5.8%) of these students had already attempted the unit at least once. In addition, four postgraduates completed the assignment but were excluded from the analysis. Of the undergraduates, 31 (20.0%) were female and 124 (80.0%) were male, with an average age of 22 years (s = 2.9 years). There were 72 (46.5%) Australians, 12 (7.7%) permanent residents, and 71 (45.8%) international students.

There were 22 (14.2%) BEng (Software Engineering) students, 62 (40.0%) BSc (Computer Science) students, 64 (41.3%) BSc (Information Technology) students, and 7 (4.5%) students from other degree programs. Like the postgraduates, the seven 'miscellaneous' students were excluded from the statistical analysis because of the small number and the diversity of their academic backgrounds.

Unfortunately it was also necessary to eliminate a further seven students because they failed to provide PSP logs. The final sample of 141 students used in the statistical analysis that follows represents 88.7% of the 159 students who attempted the assignment. Several of the remaining students provided incomplete PSP data, so the sample size varies slightly between the statistics reported below.

4.2 Mark and Effort by Degree Program

Overall, students reported working on the assignment an average of 24.9 hours (n = 148, s = 12.6) and achieved an average mark of 62.6% (s = 17.5). Figures 2 and 3 show box-plots for total PSP log time and final mark, broken down by degree program. Note that the 'Other' category represents the students who were subsequently eliminated from the analysis, as described above.

The box-plots show that the 20 BEng (Software Engineering) students scored higher marks than students from other degree programs and expended less effort to do so. They achieved an average mark of 71.6% and reported working on the assignment an average of 19.4 hours.

In comparison, the 61 BSc (Computer Science) students achieved an average mark of 65.0% and reported working on the assignment an average of 26.4 hours. The 60 BSc (Information Technology) students achieved an average mark of 57.0% and reported working on the assignment an average of 25.2 hours.

An analysis of variance (ANOVA) confirmed that the difference in marks between the three groups was highly significant (F = 7.197, d.f. = 138, 2, p = 0.01). The analysis also confirmed that the difference in PSP log hours between the three groups was significant (F = 3.156, d.f. = 138, 2, p = 0.046). These differences are not entirely surprising and appear to be a reflection of the typical admission scores required for entry into the three different degree programs.

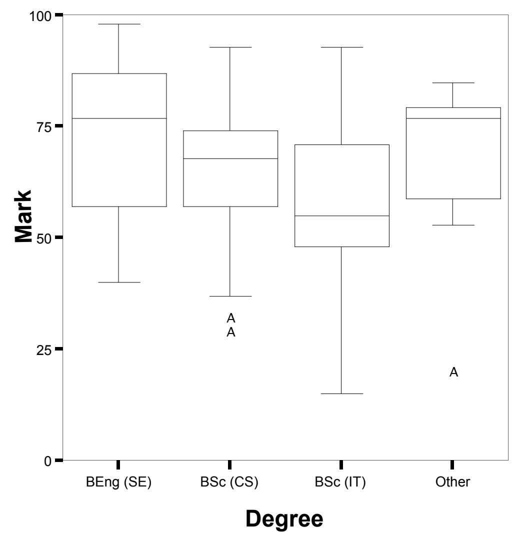

Figure 2. Box-plots of assignment mark by degree.

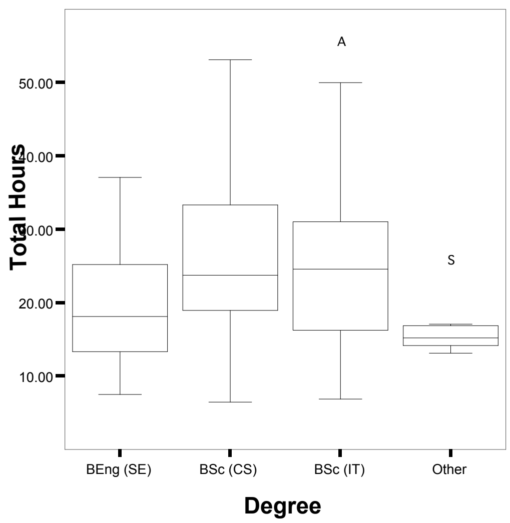

Figure 3. Box-plots of total PSP log hours by degree.

Figure 4 shows a scatter plot of the assignment marks versus total PSP log time, broken down by degree program. Surprisingly, there was no correlation between mark and effort. A Pearson correlation was not significant (r = 0.144, n = 141, p = 0.088). Students who reported spending more time working on the assignment did not necessary get a higher mark. The lack of positive correlation may be a result of the large number of outliers shown in Figure 5 and the accuracy of the PSP logs maintained by students.

4.3 Boundary Cases

The maximum time reported on an assignment cover sheet was 100 hours (this outlier is off the scales used in Figures 3 and 4). However, the corresponding PSP log documented only 30.8 hours spent working on the assignment over a period of 18 days, concluding on the day the assignment was due. The discrepancy between the cover sheet and the PSP log suggests either that the student made a deliberate attempt to inflate the time reported on the cover sheet, or that the student could not be bothered to do the calculations necessary to complete the cover sheet. Despite the discrepancy, the student was awarded a mark of 69%.

The student recording the smallest effort on the PSP log and cover sheet reported working on the assignment for 6.4 hours over a period of 7 days, with a final mark of 50%. This student stopped working on the assignment four days before the due date.

The highest mark awarded was 98%. This student reported spending 30.6 hours on the assignment over a period of 21 days, concluding 3 days before the due date.

The student reporting the earliest start date began working on the assignment 53 days before the due date, logging a total of 25.2 hours during that period. However, the student failed to meet the minimum criteria for the assignment and was only awarded a mark of 37%.

The median start date was 16 days before the due date.

4.4 Activities Not Directly Related to Learning Outcomes

A breakdown of the time spent working on individual assignment activities is shown in Figure 5. Of particular interest is the time spent on activities not directly related to the defined learning outcomes for this subject.

Students reported spending an average of 4.29 hours in preparing to do the assignment (reading, reviewing lecture notes, attending office hours, etc), and an average of 4.97 hours on the assignment write-up. This constitutes an average of 9.26 hours working on items not directly related to defined learning outcomes. Moreover, students tended to spend more time on each of these activities than they did on any activity directly associated with a defined learning outcome.

While it is important to be prepared before commencing any task, other learning experiences including lectures and group-based practical sessions had been designed for this purpose. The additional time students reported spending on preparation for the assignment suggests that they did not completely realise the intended benefit of these preparatory learning experiences. This may be due to a lack of motivation because these were not directly assessed; however, this hypothesis has not been tested in this study.

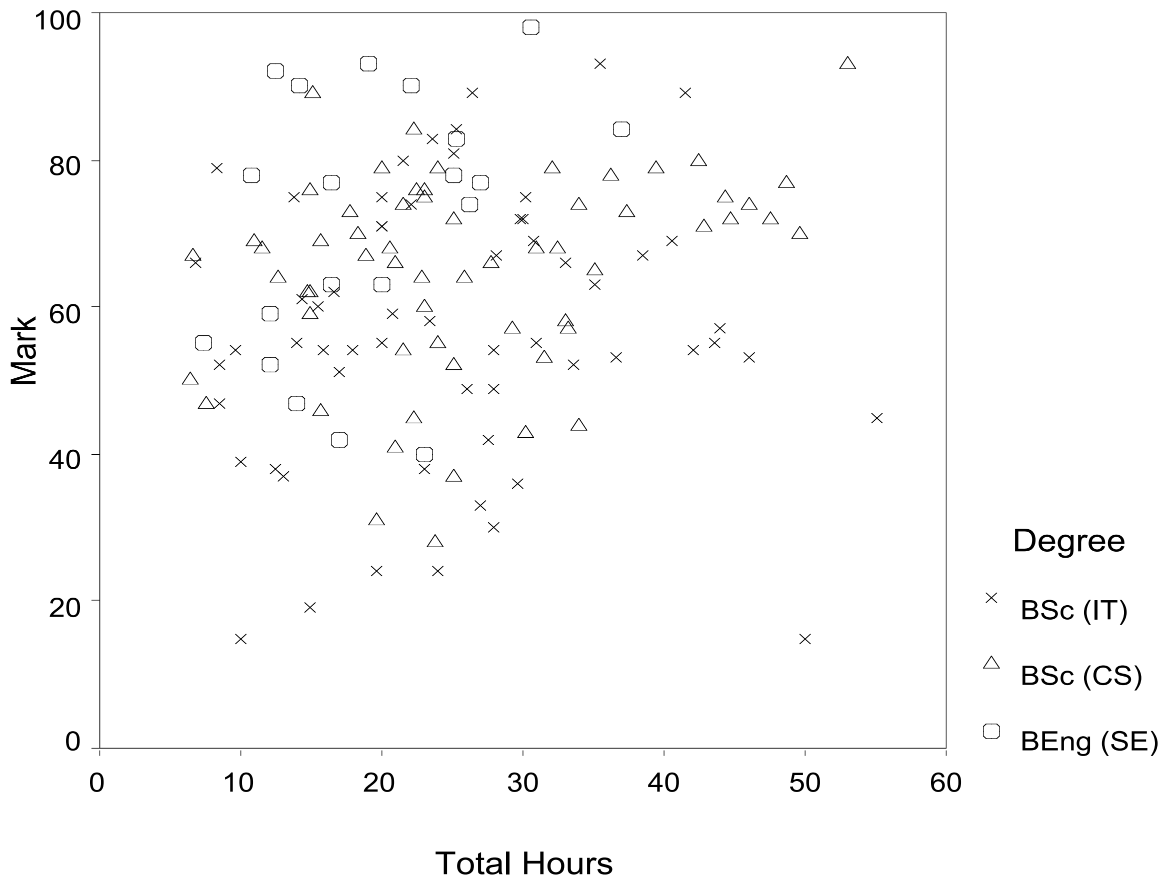

Figure 4. Mark versus total PSP log hours broken down by degree.

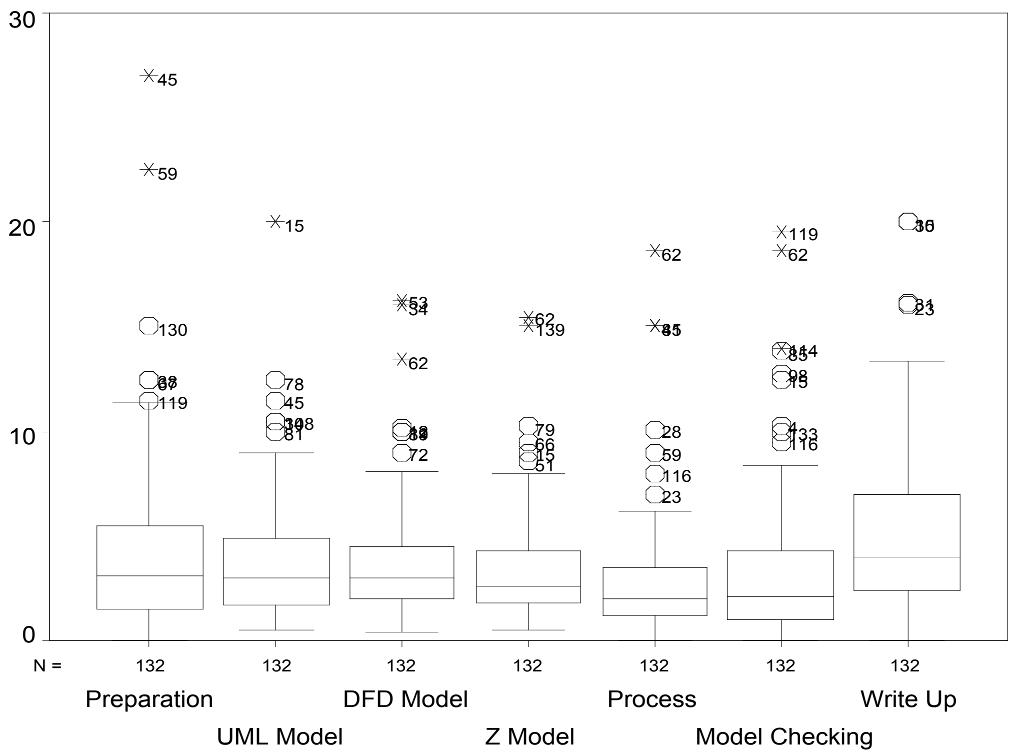

Figure 5. Total PSP log hours broken down by activity. (Numbers on outliers refer to individual students).

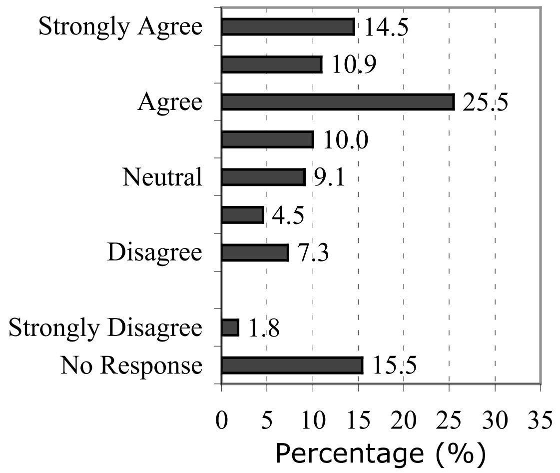

Figure 6. "To the best of my knowledge, the PSP log in my assignment accurately records the amount of time I spent working on the assignment."

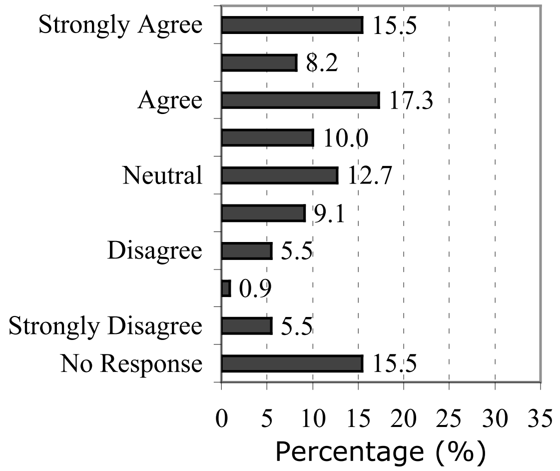

Figure 7. "The PSP log in my assignment contained entries that were generally MEASURED at the time the activity took place."

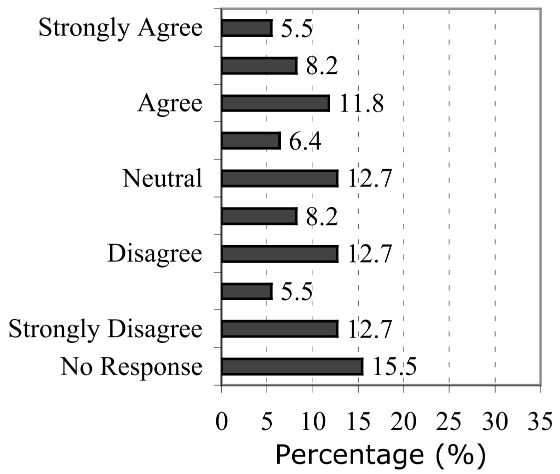

Figure 8. "The PSP log in my assignment contained entries that were generally ESTIMATED well after the activity took place."

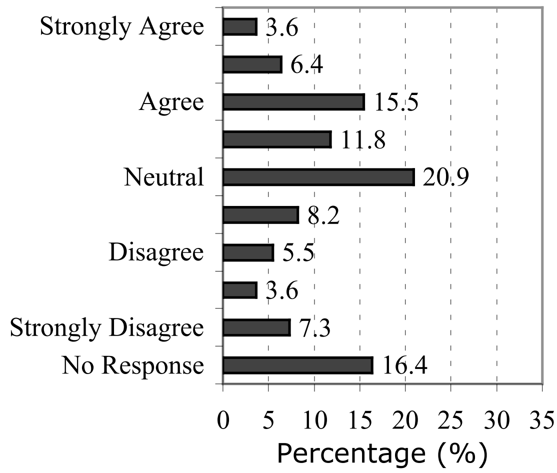

Figure 9. "I found Z to be a useful verification technique, and will continue to use it in the future."

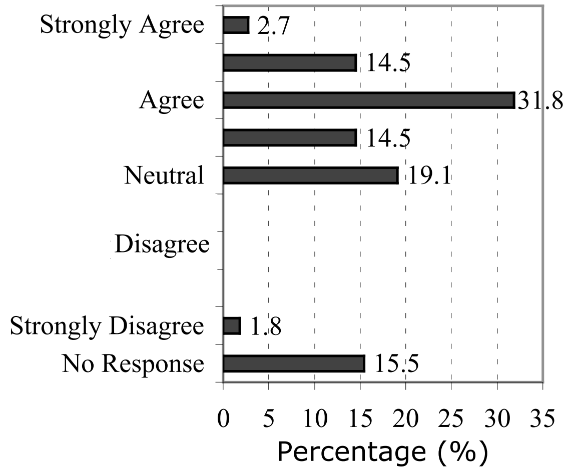

Figure 10. "Co-development in UML and Ward & Mellor DFD has reinforced basic concepts and the way models are represented in the respective notations."

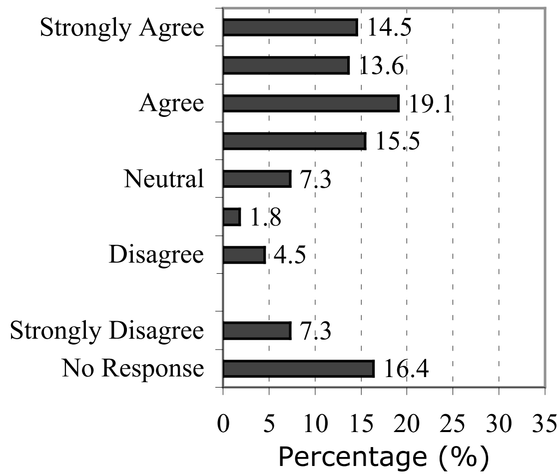

Figure 11. "In my practical session, I found teamwork to be useful to learning."

The extensive effort reported for writing up the assignment is also a concern. While technical communication in both written and oral forms are important skills for professional practice, these are assessed elsewhere in the various degree programs.

It is also worth pointing out the large number of outliers in Figure 5, each of which represents a total of seven hours or more spent on a single activity. The outliers could suggest either that students had difficulty recording PSP information for the correct activities, or that individuals spent a lot of (presumably unproductive) time on particular activities. Of course, another possibility is that the outliers are a side effect of the PSP data being fabricated. This aspect of the PSP approach will require further study.

4.5 Data Accuracy and Attainment of Learning Outcomes

Supplemental questions were included with the standard set of questions normally used in an end-of-semester student feedback questionnaire. The questionnaire used a 9-point scale, where 1 denoted "strongly disagree", 5 denoted "neutral" and 9 denoted "strongly agree" in response to written statements. The supplemental questions were designed to elicit student opinions regarding the accuracy of the PSP logs and the attainment of learning outcomes.

Slightly more than 15% of the respondents answered the standard set of questions, but did not respond to the additional questions that were found on the reverse of the standard questionnaire. Note that end of semester evaluations are anonymous, and there is no way to associate individual responses with specific PSP logs, assignment marks, or final examination results.

Several of the additional questions attempted to evaluate if students thought that PSP logs were accurate, and whether the data in them was generally measured at the time the activity took place, or estimated after the fact. Results are shown in Figures 6, 7 and 8.

As shown in Figure 6, 50.9% of those surveyed fell within the "agree" to "strongly agree" bands when asked if the data recorded in the PSP log accurately reflected the effort associated with the assignment. As shown in Figures 7 and 8, 41% of those surveyed fell within the "agree" to "strongly agree" bands when asked if values recorded in PSP logs were generally measured, and 25.5% fell within the "agree" to "strongly agree" bands when asked if values were generally estimated.

With respect to opinions regarding the attainment of learning outcomes, 25.5% of the students surveyed fell within the "agree" to "strongly agree" bands when asked if they found Z to be a useful verification technique, as shown in Figure 9. Figure 10 shows that 49.0% of the students fell within the "agree" to "strongly agree" bands when asked if co-development in UML and Ward & Mellor DFD with STD had reinforced basic concepts regarding the way models are represented in the respective notations.

During practical sessions, students worked in teams to solve basic modelling problems. Although the ability to work in teams was not a learning outcome that was directly assessed in this subject, 47.2% fell within the "agree" to "strongly agree" bands when asked if they found teamwork in the practical sessions useful, as shown in Figure 11.

Examiners assessing the assignment were asked to look for telltale signs that PSP logs were estimated rather than measured, but not labelled as such. These include suspicious values like all interruptions being multiples of 5 minutes, all activities being multiples of 15 minutes, or the total time expended for the entire assignment summing to a round number like 20 hours.

Examiners rated the accuracy of the data on a scale of 0 to 5, where 0 meant the examiner thought the log was a complete fabrication, and 5 meant the examiner thought the log contained accurate measurements and/or estimates that were clearly labelled as such. Results are shown in Figure 12.

In the assignment specification, students were instructed to record when durations and interruptions were estimated, as opposed to being measured at the time the activity took place. Very few students actually noted that entries in their PSP log were estimates, implying that most students were claiming to report measured values.

Regardless of whether PSP logs contain estimated or measured values, they reflect the student perception that they worked hard on the assignment and put in relatively long hours to complete it. However, this data does not reveal whether students worked efficiently, despite the considerable effort they thought they expended.

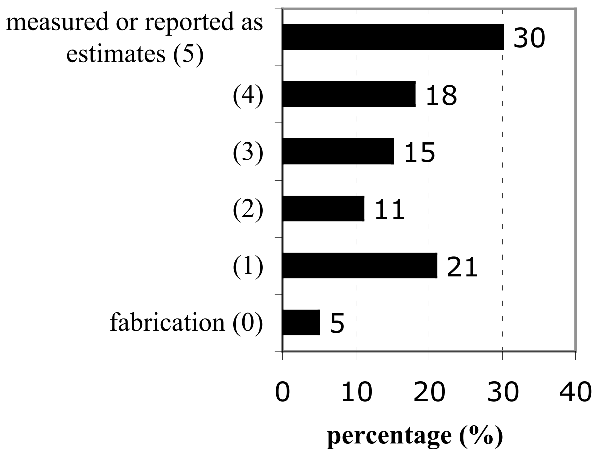

Figure 12. Opinions of examiners regarding the accuracy of PSP data recorded by students.

5 Discussion

During the two-hour final examination, students successfully developed UML, DFD with STD, and Z models that were equivalent in scope and complexity to that of the assignment. As with the assignment, the examination required students to develop consistent models of the same system across the various notations, and to use formal reasoning to verify the underlying state model. This has also been true of final examinations in previous semesters.

Although students were awarded a mark on the assignment, it was principally intended as a mechanism for providing them with feedback prior to the examination. Assuming that an alternative, less taxing mechanism can be found to provide students with feedback, it is difficult to justify redundantly assessing the same learning outcomes in light of the extensive effort expended on the assignment. This time may have been better spent engaged in other learning activities.

The time that students devoted to preparation before starting work on the assignment suggests that practical sessions devoted to the learning outcomes described here should be expanded and restructured. That is, the PSP logs suggest that students felt under prepared for the assignment, and that practical sessions intended for that purpose were ineffective.

During a semester consisting of 12 teaching weeks and containing 11 practical sessions, a total of four sessions were devoted to UML, DFD with STD, Z modelling, and formal verification. One session was devoted to each of these four topics, building upon previous experience in second year subjects. More practical sessions on these topics would likely be of benefit, providing students with additional opportunities to develop the necessary skills and understanding of the new material. However, this would necessarily be at the expense of other material that would need to be abridged or moved to learning experiences conducted outside of structured class-time.

Although most practical sessions in this subject were designed to be conducted in small groups, some material could be moved into private study time. Doing so would provide time for additional practical sessions and group interaction devoted to the learning outcomes described here.

As shown in Figure 11, 47.2% of the students surveyed fell within the “agree” to “strongly agree” bands when asked if group interaction during practical sessions was useful to learning. That students thought this was a positive learning experience suggests that group work should be expanded to provide students with a framework for group interaction outside of class time.

For example, it would be possible to suggest modelling problems that study groups could work on outside class time and then submit as a portfolio to obtain feedback, rather than for assessment. That is, feedback could be decoupled from assessment in a manner that promotes self and peer learning and prepares students for formal assessments. As an added benefit, portfolios submitted for feedback purposes do not necessarily need to have a formal due date. This would make students responsible for managing time commitments across multiple subjects.

PSP logs also show that students spent considerable time writing-up the assignment. The ability to enter engineering drawings and mathematical equations using word processors or Computer Aided Software Engineering (CASE) tools are important skills, but these were not learning outcomes associated with this subject. However, the ability to model systems using a variety of notations, and to check their accuracy are important learning outcomes of the subject. These can be achieved with paper and pencil and do not require sophisticated software for data entry and analysis.

Anecdotally, those students who faithfully logged PSP data did tend to realise the benefit of noting the amount of time spent on unproductive activities. Interruptions unrelated to the assignment were recorded in the PSP logs of a few students. Entries like these begin to reveal a more complete picture of how students actually spend their time. Generally, simple estimates of overall effort do not accurately reflect the duration of time consuming interruptions and other inefficient uses of time.

None-the-less, PSP data collected for this study does support the student perception that they work very hard in pursuit of their studies. Furthermore, considering all subjects taught in the third year of semester 1, 2004, a typical BEng (Software Engineering) student had two tests and a major assignment due in week 9, a test in week 11, two major assignments due in week 12, the major assignment for the subject described in this paper due in week 13, and a major assignment due in week 14. Ensuring that the due dates in various subjects are interleaved is a good first step in managing student workload. However, this study highlights the need for the additional step of formally evaluating the actual effort associated with specific assignments to fully characterise student workload. This effort must be justified against the learning outcomes to ensure that they are balanced with respect to assessment.

6 Guidelines For New Curricula

The observations described above have lead to the following guidelines for the re-design of the subject.

- In addition to setting learning outcomes, subject controllers should also set effort targets for all learning experiences, feedback mechanisms, and assessment instruments.
- Making students track their effort against attainment of learning outcomes is the first step in developing effective time management skills and teaching students to take responsibility for their own education.
- In assessing effort, do not rely solely on student opinions regarding how hard they worked. Put mechanisms in place to measure actual effort. Put in place safeguards to identify data that is likely to have been estimated rather than measured.
- Quantitatively assess the actual effort against targets, and adjust the learning experiences when effort cannot reasonably be justified against the outcomes. This is particularly important when there is no clear relationship between effort and the mark awarded, the latter of which has been the traditional measure of outcome attainment.

- Determine whether any sub-groups within a given student cohort are expending more effort than other subgroups, particularly if that effort does not result in a better outcome. These groups may be under prepared or may not have the same background or prerequisites as others students and hence will require additional structured learning experiences.
- Eliminate time-consuming assignments from the assessment pattern when there are redundant assessment instruments, or when there is a less time consuming way to assess the same thing.
- Separate feedback mechanisms from assessment instruments where possible, and let due dates for feedback mechanisms be as flexible as possible.
- Allow students to submit engineering drawings neatly executed using paper and pencil, assuming that mastery of sophisticated CASE tools is not a designated learning outcome.
- Teach effective time management skills throughout the curriculum for a given degree program, beginning as early as possible.

7 Conclusion

The PSP is a useful tool that enables students and educators to quantitatively evaluate the effort associated with a given learning experience.

Historically, some computing students have regarded working long hours or "pulling an all nighter" as a "badge of honour". Logging actual effort helps students to understand how they are spending their time. For the most part, students do work hard and are happy to tell their peers, parents, and lecturers such. When used correctly, PSP can help students work more efficiently by quantitatively demonstrating how time is wasted and where it can be used more efficiently.

When students use the PSP, educators can evaluate just how hard they actually work. Well-intentioned lecturers who know their subject material may have some idea how long their less experienced students will work to complete an assignment. However, lecturers will not know for sure until effort is actually measured. Doing so enables educators to quantitatively evaluate effort against learning outcomes and adjust the curriculum accordingly.

8 References

Douglass, B.P. (2000): Chapter 1: Introduction to Real-Time Systems and Objects in *Real-Time UML: Developing Efficient Objects for Embedded Systems*, 2nd ed., ISBN 0-201-65784-8, 1–47.

Ferguson, P., Humphrey, W.S., Khajenoor, S., Macke, S. and Matvya, A. (1997): Results of applying the personal software process. *IEEE Computer,* **30**(5): 24–31.

Humphrey, W.S. (1995): *A Discipline for Software Engineering.* Sydney, Addison-Wesley, ISBN 0201546108.

Humphrey, W.S. (2000): *Introduction to the Personal Software Process.* Sydney, Addition-Wesley, ISBN 0201548097.

Johnson, P.M. and Disney, A.M. (1998): The personal software process: a cautionary case study. *IEEE Software,* **15**(6): 85–88.

Nissanke, N. (1999): *Formal Specification: Techniques and Applications.* Springer-Verlag, ISBN 1-85233-002-3.

Sillitti, A., Janes, A., Succi, G. and Veernazza, T. (2003): Collecting, integrating and analyzing software metrics and personal software process data. *Proc. 29th Euromicro Conference,* 336–342.

Syu, I., Salimi, A., Towbidnejad, M. and Hilburn, T. (1997): A web-based system for automating a disciplined personal software process (PSP). *Proc. 10th Conference on Software Engineering Education & Training,* Virginia Beach, USA, 86–96.

von Konsky, B., Robey, M. and Nair, S. (2004): Integrating design formalisms in software engineering education. *Proc. 17th Conference on Software Engineering Education and Training,* Norfolk, USA, 78–83.

Ward, P.T. and Mellor, S.J. (1985): *Structured Development for Real-Time Systems, Volume 1: Introduction and Tools.* Prentice Hall, ISBN 0-13-854787-4.

Wohlin, C. (2002): Is prior knowledge of a programming language important for software quality? *International Symposium on Empirical Software Engineering*, 27–34.

Zhong, X., Madhavji, N.H. and El Emmam, K. (2000): Critical factors affecting personal software processes. *IEEE Software,* **17**(6): 76–83.