Practicing What We Preach: Applying Process Improvement to Student Projects (Tutorial)

William R. Nichols, Ph.D. and Marsha M. Pomeroy-Huff, Ed.D.
Carnegie Mellon University - Software Engineering Institute
wrn@ sei.cmu.edu mph@sei.cmu.edu

Abstract

Software projects often fail for non-technical reasons including ill-defined goals, inaccurate estimates, inadequate plans, and weak intra-project communication. Inexperienced developers often plan in insufficient detail to track progress, fail to detect project slippage early enough to take corrective actions, and make multiple false starts. After project challenges are discovered, a common recovery strategy is to redouble effort, essentially applying the same failed techniques, faster and harder. This tutorial begins with a case study examining a troubled studio project in which the graduate student tea attempted recovery by applying the Team Software Process (TSP) after other recovery attempts failed. In TSP, a coach uses a standard framework to guide self-directed teams through the processes of forming goals, developing a strategy, building a detailed plan, then tracking and managing that plan through project completion. Topics examined in this tutorial will include how TSP addressed the team's specific problems and affected team morale, and the results realized in project outcomes. The tutorial will also include a discussion of how TSP processes can be incorporated into student projects at both the undergraduate and graduate levels.

1. Topic, theme, and goals

This case study examines a studio project carried out over four semesters by a team of five graduate students in Carnegie Mellon University (CMU)'s Master of Software Engineering (MSE) program. In addition to dealing with a difficult client situation, the team struggled with issues such as requirements definition, architectural configuration, and project planning/tracking. The team made little progress during the first four months of the project in spite of implementing several mitigation attempts, and soon after the beginning of the second semester, it was clear that the team was in danger of failure. After discussing several potential mitigation strategies with faculty members and the two studio advisors, the team decided on a recovery approach that centered on application of the Team Software Process (TSP) as the approach to planning, implementing, and tracking the development work [1]. TSP uses a standard framework by which an experienced team coach guides self-directed teams through the processes of forming goals, developing a project strategy, building a detailed plan, then tracking and managing that plan through project completion.

This tutorial will include a description of the case and a discussion of how TSP addressed the team's specific problems, how TSP affected team morale, and the results realized in project outcomes. The goal of the tutorial is to provide university faculty members with information on how the TSP processes can be incorporated into student projects at both the undergraduate and graduate levels, and how this inclusion can help students to avoid many of the problems common with long-term development projects in both academic and industrial settings. The tutorial will also include lessons-learned and practical "do's and don'ts" that should be considered to optimize the successful implementation of TSP in any project setting, both in and outside the classroom.

2. Audience

The tutorial is intended for university faculty members who wish to include effective project planning and management techniques in their undergraduate or graduate software engineering courses, tracks, and degree programs. This tutorial will also be beneficial to those who are looking to add or expand TSP instruction to their existing academic programs. No prior knowledge of TSP or advance preparation is required.

3. Activities and format

The tutorial will begin with a didactic presentation of the case study, including a detailed description of the problem, the persons involved, and preliminary mitigation strategies used in an attempt to solve the problem. There will also be a brief description of the TSP methodology (for audience members who may not be familiar with the technology), followed by a description of how the graduate student team implemented TSP techniques and the results achieved. The case study presentation will conclude with a summary of the students' lessons learned. The audience will then be invited to participate in a discussion of the material presented and how the techniques used in TSP might be beneficial for use in student project implementation. The tutorial will conclude with the presenters providing guidelines and suggestions as to how TSP methods can be incorporated into existing software engineering curricula or introduced in the form of new required or elective courses. The presenters will also provide attendees with an overview of available TSP informational and instructional resources [2, 3, 4, 5].

4. Time needed for the tutorial

The tutorial will last two (2) hours. The first hour will include the exposition of the case study and outcomes. The second hour will consist of a discussion session with question/answer period, the presentation of methods and approaches for including TSP as new or supplemental material in software engineering courses and curricula, and improving results for those who may already have TSP elements incorporated in their instructional tracks.

5. References

[1] W. S. Humphrey, The Team Software Process (TSP) (CMU/SEI-2000-TR-023), Carnegie Mellon University, November 2000

[2] W. S. Humphrey, Introduction to the Team Software Process (TSPi), Addison-Wesley, Boston, MA: 1999 (ISBN 020147719X)

[3] W. S. Humphrey, TSP: Coaching a Development Team, Addison-Wesley Professional, Boston, MA: 2006 (ISBN 0201731134)

[4] W. S. Humphrey, T. A. Chick, W. Nichols, and M. Pomeroy-Huff, The Team Software Process (TSP) Body of Knowledge (BOK) (CMU/SEI-2010-SR-001), Carnegie Mellon University, January 2010