Is This Training?
A Unique Approach to Software Process Training in Industry

Pauline D. Fortin, Christine A. Jeske, John R. Lakey, Kristin B. Urquhart, and Anthony Vea

Texas Instruments Learning Institute
Texas Instruments Incorporated
P.O. Box 650311, MS 3928
Dallas, Texas 75265
Phone: 214-917-1911, email: PFOR@msg.ti.com

Abstract. This paper describes our just-in-time approach to software process training. We explain how our project-specific, software-process workshops are custom tailored and conducted for individual software development projects. We also explain why the workshops succeed at assisting projects to define the detailed plans and procedures that form their software process. We conclude by describing how we deal with some of the unique challenges we face in developing and conducting the workshops.

1 Introduction

Consider the following scene:

A project's software development team gathers around a table to plan and review the activities for the next development stage. Project-specific software plans, standards, and procedures are clearly in view. The team discusses their detailed project schedule. They reach consensus, and everyone present seems to understand what is expected of them and why.

Is this a scene from a fictional project?

No. This scene is becoming routine for software projects attending our Just-in-Time Software Process Workshops.

2 What are Just-in-Time Software Process Workshops?

The Just-in-Time Software Process Workshops provide a forum for an entire software-development project team to come together in a dedicated, project-specific, workshop setting to define the details of their software process. Workshops last one to two days and are held when a project team is ready to launch into a new stage of development. Each workshop combines technical training and objective facilitation with pre-class and in-class activities tailored to meet the needs of a specific software-development project.
2.1 Workshops Currently Available

We currently offer four Just-in-Time Software Process Workshops:

- Requirements Analysis
- Preliminary Design
- Detailed Design
- Implementation (includes Coding, Module Testing, and Software Integration)

2.2 Role and Key Features of the Workshops

The workshops do not replace the overall project planning which generally occurs early in a project's life cycle. Instead, the workshops provide the software team with the means for updating, refining, and adding the necessary detail to the software plans and procedures (e.g., Software Development Plan and Software Standards and Procedures Manual) at a time when the details are best understood. Since the entire project team attends the workshops together, the major outcomes include an understanding of and a commitment to the project's agreed-upon process.

Although the primary focus of the workshops is not formal process theory, we include discussions and presentations that communicate our organization's approved software process along with lessons learned from other projects. The opportunity for software practitioners to view and discuss their own software process within the framework of the organization's software process plays a major role in our organization's continuous software process improvement. Additional training courses are available which cover software engineering fundamentals, computer languages, operating systems, tools, and methodologies.

A fundamental feature of the workshops is their timing. Since the workshops are held *just-in-time*, a software development team can examine the activities and process issues relating to its current stage of the software life cycle. The workshops help to clarify what is to be done, how it will be done, who is responsible, and when it is to be completed. The workshop instructors act as facilitators that assist the software team in making decisions during the workshop.

2.3 History and Background of the Workshops

The workshops have evolved since 1990, when a member of our current software process training team developed a training course to deploy our then-new corporate software process. Project teams found it beneficial when the instructors asked them to review and compare their own software plans and procedures to the corporate process. Feedback from students clearly pointed out the need, however, to break the course into several installments to match more closely the course topics with a project's software life cycle. Adopting a just-in-time approach seemed like the logical way to meet project needs. Additional activities, exercises and presentations were subsequently developed to integrate project-specific materials and further customize each workshop. Project teams liked this