When the CMM Shoe Doesn't Fit: Tools for SPI on Numerous Small Projects

Dr. C. Wrandle Barth
Raytheon ITSS
September 1, 2000
301-614-5472
fax: 301-614-5268
Randy.Barth@gsfc.nasa.gov

Adapting a development organization to the CMM mold can be challenging when the organization does not fit the CMM target environment. The Goddard V0 DAAC's Software Engineering Process Group has been studying for some time how to apply CMM and other software process improvement techniques to the DAAC environment. The DAAC V0 software is fairly mature. Maintenance work is complemented by various new development activities, many of which make extensive use of the existing body of software. Many of the software "heroes" who developed the original code have gone on to other activities, so improving both the state of the current software and the processes applied to new development is critical to ongoing success. Each semester (six month task planning period), the DAAC's Engineering group may work on 20-30 individual jobs, ranging from some simple fixes and extensions to full utilities or new subsystems. However, these average only about six staff-weeks apiece with few going over twelve staff-weeks; most are one- or two-person efforts. A given engineer may work on several jobs during the semester, sometimes overlapping. While software engineering techniques have much to offer, they need to be properly scaled for this environment.

Tracking all of these activities has been a challenge. Most of the problems associated with any software project are here, but are compounded by the fact that solutions appropriate to multi-staff-year projects can easily overburden these short jobs. A quarterly project status report requiring several hours of preparation is not going to work for a six-week job. If status reports are moved to weekly, they must be able to be completed in a matter of minutes, even when the staff member is working on two or three different jobs at the same time.

Fortunately we have been able to exploit the Web and high-level tools to streamline many of the processes needed for software development. These tools have been brought together in a working internal Web site that the engineers can use for planning, tracking, storing, communicating, collaborating, and documenting their needs. Management, both contractor and government, can easily look at the status on demand rather than having reports thrust upon them presumptuously or needlessly. And even these management queries can be tracked and examined for insights into how to evolve the system.

Taking a cue from modern help documentation, the policies, processes, and procedures are stored as short hypertext pages linked together. This makes finding any piece of information simple and avoids forcing engineers to read details about every aspect of the

process before they can begin to apply it. A secondary copy is available as a linear file for walk-away study for those who prefer to read a procedure from end to end. Templates and worksheets are also directly linked to provide convenient starting points. And many of the areas in software development plans are pre-filled with references to standard DAAC procedures that all jobs follow. For example, a standard configuration management scheme is used throughout the DAAC V0 baseline software, so a separate CM plan in each job would be not only wasteful, but also needlessly confusing.

The processes and procedures have been written to foster software process improvement within the DAAC environment. The goals of this improvement are in many ways parallel to those of CMM. There are, however, a number of areas where we have streamlined our procedures to size them to our needs. For example, normal CMM activities that are designed to insure good communication on a project among the programmers, first-line software managers, requirements manager, software development planner, software designers, other software engineers, systems engineers, and project manager become somewhat moot when, in fact, these are all the same person. But since our processes and procedures are designed to reach the same goals as CMM, we will be doing an evaluation of them to see whether sufficient to assess us at level two.

The heart of the planning and tracking procedures is the Dynamic Status Database (DSDB). The status of all DAAC V0 Engineering work is stored here in detail. Mechanisms are provided to make data entry, update, and querying as easy as possible for the DAAC engineers. In addition, metrics can be captured for Performance-Based Contracting evaluation. At the start of each semester, the government ATR assigns an initial set of jobs for work during the semester and assigns point values and deliverables for each. The section leader enters these into the DSDB to form the basis of the work for the semester.

Many of the jobs are like small software development projects. (Others, like systems administration task and workstation support are also handed by the DSDB but will not enter into our discussion.) A job leader is assigned, who then functions similar to a project manager. Since most of our jobs are one-person activities, much of the CMM-style reporting below a project manager is superfluous. We do have a software leader, but he functions above the job leaders to help distribute staff assignments and track progress. What we focus on is aiding the job leader in organizing his planning and reporting, and aiding the section manager and software lead in understanding the progress.

The job lead expands the customer-mandated list of deliverables in the DSDB for a job with any other work products that are meaningful to track. In particular, when multiple people will be working on a job, he can put those things assigned to others in as separate work products for tracking. The work product becomes the primary element of all tracking and reporting. For each work product, information on estimated effort, schedule, peer review plans, and staff assignment are made through Web pages that update the DSDB. Here again, defaults are provided to simplify the process—the job leader is by default assigned as staff on all products. The lead also inserts job-level estimates and

schedule into the DSDB. When complete, a table can be extracted by cut and paste to put into the software development plan template that the lead can pull from the Web. In order to encourage weekly updates, pages are provided to allow an engineer to complete status updates in a matter of minutes. A single page lists all jobs for which the engineer is lead and all work products on other jobs to which the engineer is assigned. Fields are displayed containing various attributes; any can be clicked and updated. In particular, closing out a completed work product allows a quick marking of the item, and the collection of a few pieces of information (such as the URL where the product can be found, though this is defaulted with the most common value).

While requirements-related information is captured in the DSDB, the requirements themselves are not. When a requirements document is marked as completed, the engineer is queried on the number of requirements in the table. If the requirements change, the engineer makes the changes and re-closes the item, when he will be queried for the number of changes, additions, and deletions. The Web page code will calculate a figure representing the percent of the way through the scheduled job that the event occurs, referred to as the "lead factor". The lead factor is also recorded when the job effort and schedule estimates change. These records will allow us to collect metrics over time to view requirements and planning volatility. While very basic, such information has never been collected in our environment due to the overhead that it seemed to impose.

Peer review tracking is also supported by the system. The work products to be reviewed can be marked in the DSDB and the date the review is completed can be entered. Action items can be entered against any work product or against the job as a whole and assigned. Action items assigned to an engineer appear on the same page with the other work assignments and can be easily closed from that page. The Web pages that describe peer reviews also provide a template for recording statistics associated with the review. This is kept in a separate database and contains no record of the item being reviewed to encourage openness in the review process.

Various queries are provided mainly for the section manager and software lead. These include status information on all jobs, outstanding or overdue work products, open action items, and staffing assignments. Several can be sorted in various ways by clicking on column headers. One of the columns, date of last update, allows an easy way to see jobs that have either stagnated or are not being reported regularly. Whenever any of these queries are made, a record of the query and who made it is recorded. This allows tracking of management oversight itself for audit purposes and to understand which reports are most useful.

In addition to providing access to the DSDB, the Web is being used in other collaborative ways. A working directory on a file server provides a repository for all documentation related to a job as an online "software development folder." Rather than passing around a requirements table by email, for example, the working copy is kept on the server. Automatic version control marks all changes as they are made. At the completion of a change cycle, the marks are hidden, the new version is delivered to the customer's ISO

9000 controlled documents directory, and the DSDB entry for the table is marked as completed (or amended). The document in the working directory contains full information about all versions, when every change was made, and by whom. The collaboration area provides a working directory that is accessible to all on the team with sophisticated tools to minimize overhead associated with tracking and control.

At the writing of this abstract we are piloting these processes and procedures with two jobs to fine-tune them. Then we will begin training our full engineering developer staff and deploy them on all appropriate jobs. By the time of the formal presentation, we should have additional results to report, both on the initial practical success of our efforts and on the evaluation of our efforts for CMM certification. This will lead to future work in expanding the DSDB and collaboration techniques, and in addressing additional areas of software engineering to the small project environment.