Today's Software Complexity Demands Good CM

Reuel S. Alder, *Publisher*

To address the configuration management (CM) theme of this issue, I will first state unequivocally that documentation is good. A recent experience serves as a vivid demonstration of this point.

I just returned from a trip to Denver that was arranged entirely by electronic means. I had contacted my travel agents, given them my travel plans, and paid with my credit card. All I had to do was go to the airport. No paper was necessary—my ticket was electronic.

There was just one problem. As I entered the terminal, I thought I had stepped into a time warp. All the computers were down, as if the year 2000 problem had just struck. I could not get a boarding pass or seat assignment at the check-in counter. I was sent to the gate with the hope that the computer problems would be solved before I got there. I checked in at the gate, but since the computers were still down, I could not get a seat assignment.

When it was time to board the plane they followed a peculiar process: First, all passengers with seat assignments on their *paper* boarding passes were invited to board the plane. Then, all passengers with paper boarding passes and no seat assignment were invited to board. After numerous checks to see if any seats were left, those with electronic tickets were invited onto the plane; as we boarded, our names were written on a piece of lined paper to keep a record of who boarded the aircraft. I felt like a third-class citizen (not second class, because my ticket was coach). During this process, it was not clear to us electronic dependents if seating would ever be available.

The moral of this story might be that paper is superior to electrons, but that is not the point I want to make; instead, I intend to emphasize the necessity of keeping track through CM. CM is an important job, but apparently, few people want to do it. CM has moved beyond mere clerical work into a highly technical realm, but those with sufficient training to perform CM feel that the job is too mundane and monotonous, that it does not provide a creative outlet. They would prefer to engage in the more "glamorous" aspects of software production, such as design and code.

But CM cannot be neglected; it can mean the difference between chaos and order, between good software and defective software, between precision and error. The complexity of today's software demands that meticulous care be taken to track and record all modifications, deletions, and additions. No human mind is equipped to remember what was changed on dozens of versions.

But even if someone were able to retain that much information, they would not necessarily remain with the project until completion. When people leave organizations, their successors need to pick up where they left off with little delay. They need to know, by consulting the documents left behind, what was done and what was not. Duplication of effort and omitted tasks waste valuable time in projects that all too often are hurtling toward a deadline.

Just as the absence of written records nearly left me waving goodbye to my flight, poor CM in software projects could leave you bidding farewell to months of hard work as you spend an enormous amount of time backtracking, retesting, and recoding to discover why the modification you made did not turn up in the final product. This month's collection of articles will provide assistance to improve any CM effort. ◆

More Advice on CM Tool Acquisition

Ronald Starbuck's article, "Software Configuration Management: Don't Buy a Tool First" (*CROSSTALK*, November 1997), raised several interesting points. But there are three additional concerns that are key to finding the right software configuration management (CM) tool and for the successful implementation of a sound CM process that is valued for its benefits rather than despised for its perceived obstacles.

First, too often, organizations will purchase a CM tool based on its "bells and whistles" without examining whether those features are desirable and, more important, whether the tool rests on a sound CM process foundation. Second, the tool purchase decision makers may fail to view CM as a discipline that involves not only software engineers or developers but also program managers, test engineers, quality assurance specialists, and the customer(s). In so doing, they fail to evaluate a CM tool in light of the comprehensive needs of the organization and of those involved with or impacted by CM.

And third, organizations frequently find themselves purchasing a CM tool from a vendor who does not know CM. Organizations can reap crucial value-added CM expertise when they work with a vendor who has firsthand knowledge of CM and can apply that "lessons-learned" experience to an organization's unique culture and process, as indicated by Starbuck. Teaming with a CM tool vendor who knows and understands CM—the process and the people—can make the purchase of a CM tool one of the most crucial and ultimately successful steps an organization can take.

Debbie J. Stack
Progressive Software Solutions, Inc. (PROSOFT)