Editor: Donald J. Reifer ■ Reifer Consultants ■ d.reifer@ieee.org

XP and the CMM

Donald J. Reifer

During the past year, I have listened to Barry Boehm, Watts Humphrey, and other software engineering sages argue that Extreme Programming and the Software Capability Maturity Model are compatible. They contend that organizations implementing XP practices can easily fit them under the SW-CMM because it represents a framework for self-improvement. Mark Paulk, one of SW-CMM's authors, captures the consensus in this way:

> *When rationally implemented in an appropriate environment, agile methodologies address many SW-CMM Level 2 and 3 practices. The ideas in the agile movement should be carefully considered for adoption where appropriate in an organization's business environment. ("Agile Methodologies and Process Discipline,"* CrossTalk*, Oct. 2002, pp. 15–18)*

So why then can't a firm moving to XP be rated a Level 2 or 3? If XP and the SW-CMM are amenable, why do lead assessors and process proponents frown when the two terms are uttered in the same sentence? Why do XP team members avoid following the plan-driven processes their organizations have inserted that comply with the 318 practices organized in 18 key process areas comprising the SW-CMM? Is there a problem?

Needless to say, I believe there is a problem that organizations must address. Otherwise, sides will form, battles will rage, and nobody will reap the benefits that either technology provides.

Pointing fingers

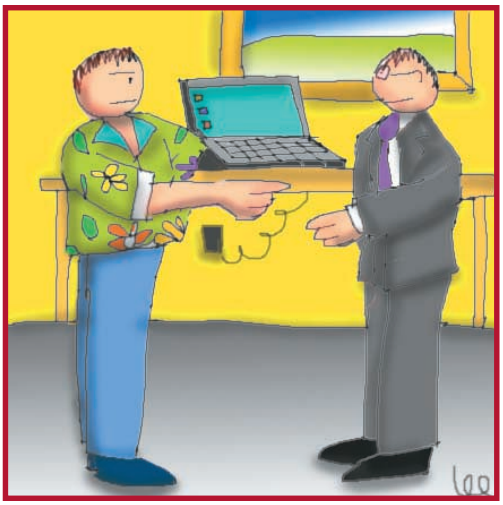
During the past six months, I have worked with two commercial firms on large Web-based projects that are moving to XP. Both have established process organizations and have been rated Level 3 via formal assessments. In both, wars are raging over what each side views as incompatibilities between XP and the SW-CMM.

Any discussion of conflict should begin with intelligence about the combatants. Both projects are large (over 1 million lines of Visual C++ and Java), distributed, and complex. Parts of both are being programmed by teams of teams that are geographically distributed across at least two continents and multiple time zones. Both projects are driven by time-to-market considerations, and neither will have its requirements fully defined until delivery. Both firms view requirements as a learning process and are building applications that are new and volatile. Extreme pressure exists to deliver functionality on time.

Project 1 is building application servers for day traders in the US and Europe. Traders use these analytical programs to profile markets, which vary from country to country. New algorithms provide trending and other information that users value differently according to national preferences. The developers have taken care to scale XP practices such as pair programming for use on this large project. Pairs are staffed daily within geographical hubs; each of which is a center of expertise for a specific market segment. For example, day traders in the power segment employ different analysis techniques than those in the semiconductor industry, so developers for these two applications are not paired together.

This project's developers moved to XP to speed products to market. They resist using processes their process group has established as too formal and involving too much documentation. The business manager has sheltered the XP teams from process group criticism, claiming that this is a pilot project. The process group has elevated the dispute to senior managers, arguing that because their process represents industry best practices, nobody should be exempt.

0740-7459/03/$17.00 © 2003 IEEE



Project 2 is developing Web-hosted applications for use across the enterprise using a newly defined architecture. Spurred on by cost savings realized in putting travel planning and record keeping on the Web, senior management has dictated that all company systems—from accounting to supplier management—must reside on Web servers, located around the world, within the next 18 months. Where the schedule came from, nobody knows. Teams of teams have worked for over a year to satisfy management's requirement.

The developers chose XP because it gave them hope for meeting the schedule. If the firm relied on past processes, their estimating models predicted it would take them at least five years to put these applications on their portal. As expected, when the company's managers heard these numbers, they shut their ears. The mandate is 18 months. So be it.

The developers modified XP practices to address the scaling issues related to large projects, with the major modifications in testing. Each application placed on the Web had to be upward compatible and provide equivalent performance. Independent test teams formed to verify these goals against weekly builds. This in turn forced changes to the XP test-first practice and generated additional test documentation. Nobody on the team complained; the amount of documentation produced using the modified XP practices was considerably less than generated using the organization's Level 3 processes.

However, the quality assurance team raised lack of test documentation as an issue to senior management. Under XP, QA's role was greatly diminished, so they analyzed early builds and pointed to the high error content. To add to the intrigue, the XP manager kicked the QA people off the project because he felt that they were not adding value. Words then got heated, and QA went on the warpath. As a result, the project is at a standstill, with nothing getting done. People are waiting for the open issues to be resolved.

In both organizations, workers have formed camps. Instead of trying to make XP work rationally with the firm's existing processes, each side is pointing fingers at the other. No one seems to be trying to apply XP within the SW-CMM context rationally and profitably as the sages have suggested.

Management pressure to deliver according to an unreasonable schedule seems to be the root of the friction. XP adherents feel that they don't have time for the formality expected when using their existing plan-driven approaches for software development. Process proponents argue that they need to take the time, or else quality will suffer and customer satisfaction will be sacrificed. When you think about it, both arguments contain elements of truth.

Fixing problems

So, how do we fix the problem? Do we force XP engineers and process group members to kiss and make up? How do we implement XP in concert with the SW-CMM?

In answering these questions, principals from these two firms agree that they should try to use the SW-CMM as a framework for embracing XP practices. This will serve two good purposes: The firms can capitalize fully on the process infrastructure they have created and the experience gained as they ramp up their XP projects and transition XP's practices across their organizations. Also, tthey can address the issues of scale in XP in acceptable ways.

In Project 1's case, they could have used many of their existing processes as they addressed the geographical dispersion of talent using dynamic pair programming. For Project 2, they could have modified QA processes to play a more useful role on the project, especially in assuring adherence to the additional testing practices XP added.

Next, their process organizations should provide hard-hitting advice for folding XP practices under the SW-CMM umbrella. Instead of resisting change, they should embrace it, especially if the results reduce time-to-market and cost. Suitably rewritten process manuals should show developers how to put lightweight practices to work in, as Mark Paulk suggested, "appropriate environments." Using this approach, organizations could continue to apply the lessons learned on XP projects across the enterprise.

Finally, the Software Engineering Institute should propagate the lessons learned through their conferences, publications, and Software Process Improvement Network chapters. The SEI could provide leadership and stimulate the process community to develop recommended XP mappings to the SW-CMM. In addition, it could put these mappings on its Web site and highlight them as part of outreach. After all, Level 5 organizations should have a technology change-management process in place to permit them to incorporate innovations such as XP into normal practice. The XP transition could serve as a useful case study for implementing such a process.

Most leaders in the field agree that XP and the SW-CMM are philosophically compatible. However, putting XP to work in an SW-CMM environment is often difficult, because guidance on how to take advantage of existing best practices is not available while transitioning to XP. If the process community provided this guidance, the XP community would gladly use it to smooth this transition.