Setting Process Improvement Priorities

Mark C. Paulk
Software Engineering Institute
Carnegie Mellon University

The Rocket Scientist Principle – You don't have to be a rocket scientist to understand the Capability Maturity Model® for Software. If something in the CMM® is confusing, it's more likely because we explained the concept poorly than that the concept is extremely sophisticated.

Over the last decade I have had the opportunity to work with many people who were using the Capability Maturity Model (CMM) for Software as the foundation for their process improvement efforts, as well as others with quite different professional philosophies and objectives. Examples of different philosophies include:

- process improvement should focus on the “soft” issues – interpersonal skills, team building, etc.
- process improvement should be measurement-driven – identify your business objectives, define process and product measures that provide insight into achieving those objectives, and continually work to improve performance
- process improvement should be driven by the proven techniques of statistical process control rather than an arbitrary, artificial set of priorities such as those set by the CMM
- “good enough” software, which I would characterize as the philosophy that, in a competitive business world, you have to make pragmatic business decisions about the tradeoff between quality (in the sense of defects), functionality, cost, and schedule – strong feature sets and quick cycle times are the highest priorities for business success

I agree that there is an element of truth in each of these arguments, each of which emphasizes different aspects of “Total” Quality Management. The SEI’s improvement approach is philosophically based on the Total Quality Management (TQM) principle that focusing on quality leads to decreases in cycle time, increases in productivity, greater customer satisfaction, and business success. What distinguishes the CMM from other quality philosophies are the improvement priorities embedded in the five maturity levels.

The CMM improvement emphases move in three “waves.” Level 1 organizations rely on the competence and heroics of the professional staff. Emphasizing the selection, hiring, development, and retention of highly competent employees is crucial to organizational success at level 1 – as it is for all organizations at any maturity. As projects and the organization grow, the ability of people to overcome inadequacies in “the system” are overwhelmed. In the first improvement wave, at level 2, the emphasis on project management is directly aimed at removing barriers by installing basic project management. The second wave, at level 3, moves explicitly toward a learning organization. The third wave systematically applies measurement – and more specifically statistical thinking and an understanding of variation – to controlling and improving software processes at levels 4 and 5.

These priorities have been controversial in some quarters over the years. We have simply observed that these priorities work and work well for most organizations, although they are an 80% solution – most assessments identify non-CMM findings and recommendations for level 1 organizations frequently include establishing a software engineering process group and/or installing peer reviews, which are level 3 priorities. The emphasis on project management in the first improvement wave, however, is what seems to engender the most resistance, because it “ignores the really important engineering processes.” My reply is that much of the “software crisis” is self-inflicted, as when a Chief Information Officer says, “I’d rather have it wrong than have it late. We can always fix it later.”

® Capability Maturity Model and CMM are registered with the U.S. Patent and Trademark Office.
The emphasis in all too many organizations is on achieving cost and schedule goals, frequently at the cost of quality. In his book *Why Does Software Cost So Much?*, DeMarco observes that this situation is the not-surprising result of a combination of factors:

- “People complain to us [software developers] because they know we work harder when they complain.”
- “The great majority [report] that their software estimates are dismal… but they weren’t on the whole dissatisfied with the estimating process.”
- “The right schedule is one that is utterly impossible, just not obviously impossible.”

DeMarco goes on to observe that our industry is over-goaded, and the only real (perceived) option is to pay for speed by reducing quality. This is a management and cultural problem, not an engineering problem. To attack the root of our process problems, we have to address these management issues first – as the CMM recommends.

To quote George Box, “All models are wrong; some models are useful.” Although I believe the CMM is a useful and powerful model, it is neither comprehensive nor perfect, and the issues raised in different approaches should be thoughtfully considered in a CMM-based improvement effort. The fundamental concepts of the staged maturity levels in the CMM have, however, been demonstrated to be effective. Now all we need to discuss are some of the implementation issues…