“Of course, if your people aren’t smart enough to think their way through their work, the work will fail. No Methodology will help. Worse still, Methodologies can do grievous damage to efforts in which people are fully competent. They do this by trying to force the work into a fixed mold that guarantees a morass of paperwork, a paucity of methods, an absence of responsibility, and a general loss of motivation.”

Thus speaks one major critic of CMM as quoted by Paul Adler and his colleagues in their article “Enabling Process Discipline: Lessons from the Journey to CMM Level 5” in this issue of MISQE. Ah, for the old free-flowing days when programming was an art, artists did their thing, and life somehow felt freer. Those of us who remember back to the 50s and 60s can still reminisce over the good feeling, perhaps even thrill, as a program that we developed finally ran error-free, did the job and was ready to put into production. It was a one-person problem-solving exercise. And, it was fun.

Looked at in one sense, the fifty-year history of computer use is one of increasing constraint of both developers and users of systems. Most recently, the constraints have become ever more evident. For users, forget the old days when IT wrote programs to “fit the needs of their customers.” Today, plain vanilla ERPs (with a few workarounds) are implemented and users change their work processes to fit. In like manner, programmers are increasingly subjected to increased discipline. The apex of this discipline is the SEI Capability Maturity Model (CMM).

It is still a matter of debate as to which of the five levels of CMM is appropriate for particular organizations. In the 1990s, Computer Science Corporation (CSC) made a decision to move to CMM and to move to the top level in parts of the organization. Adler provides an insight into the journey of two CSC organizations as they made their way to CMM Level 5. CSC’s four “success factors” are interesting. All involve people, both management and programmers. The most novel and, according to the authors, the most important is “organizational socialization” of the development staff – the process of learning and inter-