Balancing Discipline and Flexibility
With the Spiral Model, MBASE, and the CMMI

Barry Boehm and Dan Port
USC Center for Software Engineering
February 2001

1. Introduction

In his keynote address at the 1996 International Conference on Software Engineering, Tom De Marco summarized the work of the great military analyst Karl Von Clausewitz on the interplay of armor and mobility in military conflict. At times, armor will dominate mobility, as with heavily armed medieval knights dominating lightly armed peasantry. But if over-optimized, one strategy will lose to advances in the other, as the ponderous French knights found in their inability to dominate the lightly armed and mobile English longbowmen in their watershed loss to the English at Crecy in 1346.

De Marco then drew a parallel between "armor-intensive" software strategies such as the Software Capability Maturity Model and the "mobility-intensive" lightweight processes which were emerging at the time, inferring that the software CMM was too ponderous to cope with the need for rapid development and rapid change characteristic of such sectors as electronic commerce and Web-based systems. In the ensuing discussion, software CMM advocates have cited the high mortality rates of lightweight process organizations, and their frequent inability to cope with success when they need to scale up their process and architectures to deal with more complex services and heavier workloads.

Underlying this point – counterpoint is a key software-engineering question: how much discipline is enough, and how much flexibility is enough?

In our (Boehm and Hansen) previous CrossTalk article, "Understanding the Spiral Model as a Tool for Evolutionary Acquisition" (May 2000, pp. 780), we showed that the risk exposure considerations used as spiral model decision criteria could be used to address "how much is enough?" questions. There, we showed how a "how much testing is enough?" question could be addressed by balancing the risks of doing too little testing (alienating your users) and the risks of doing too much testing (unavailable combat capability; missed market windows).

In this article, we show how the spiral model and its recent extension, Model-Based (System) Architecting and Software Engineering (MBASE), can be used to tailor a project's balance of discipline and flexibility via risk considerations. We also describe and rationalize the major MBASE extensions to the spiral model (model clash avoidance; stakeholder win-win), and provide a counterpart set of essential elements and variants for MBASE. We then relate the MBASE key practices to those of the Integrated Capability Maturity Model (CMMI), and show how both are compatible and can be used to balance discipline and flexibility in a project's system and software engineering approach.

2. MBASE and Model Clash Avoidance
   2.1 Nature of Model Clashes
   - from "Avoiding the Software Model Clash Spiderweb" Computer, Nov. 2000
   2.2 MBASE Model Integration Framework and Process Framework
   - from "Escaping the Software Tar Pit: ..." ACM SEN, Jan. 1999
   2.3 MBASE and Stakeholder WinWin
   2.4 MBASE Essential Elements and Variants (table; short explanation)
   2.5 MBASE Usage Experience
3. MBASE and CMMI
   3.1 MBASE and CMMI key practice mappings (SEPG 2000 tutorial)
   3.2 How MBASE and CMMI balance discipline and flexibility (and support evolutionary acquisition)
   - Example: Schedule as Independent Variable (SAIV) process
4. Conclusions
   - CMM's (particularly CMMI) can balance flexibility and discipline
   - MBASE is a convenient spiral-based approach for transitioning to the CMMI and evolutionary acquisition
5. References