An Information Abstraction Model for End User CASE Support

Shirley A. Becker
The American University

Rick Gibson
The American University

Computer-aided software engineering (CASE) tools have been judged as failures by various, traditional measures of an organization’s success in systems development. We propose to use the Software Engineering Institute Capability Maturity Model (CMM) as a framework for developing a more relevant measure of success. The result, an Information Abstraction Model (IAM), identifies the information required for effective management of software development practices that would support a CMM Level 3 definition. An integrated CASE toolset is presented that would enable the practical use of the IAM in the pursuit of a more mature organization.

With powerful workstations becoming commonplace, Computer-Aided Software Engineering (CASE) tools seem ready to provide invaluable support throughout the systems life cycle. Only CASE tools answer the need for better ways to develop systems with broad opportunities for automation, which also allows for development of applications too complex to develop by traditional methods. Currently a favored approach, promising to improve quality and productivity, CASE tools have become the mainstream information technology that separates the world-class players from the mediocre software shops.

The concept of using computers to automate systems development is not new—in a sense, common language compilers and interpreters can be thought of as CASE tools. Whitten et al. (1994) present the history of CASE tools as dating back to the early 1970s. They point out that early tools allowed for user problems and solution requirements for an information system to be stored in a computerized dictionary. Tools were also developed for analyzing these problems and requirement statements for completeness and accuracy. Unfortunately, few companies could afford to utilize these early CASE tools as they ran on large mainframe computers that consumed precious and expensive machine resources. With the advent of the personal computer, Index Technology (now INTERSOLV) developed a tool called Excelerator in 1984. Its success established CASE as an acronym and an industry.

However, several facts emerge to suggest that the acronym CASE actually stands for Computer-Aided Software Enigma. First, computer professionals have been the last to apply the benefits of computer technology to their own work. Second, it is not clear who is driving the need for CASE, is it the market or the vendors? Misconceptions lead to the creation of “shelfware.” And finally, although Swanson et al. (1991) report productivity improvements attributable to CASE, Yellen (1990) described such gains as elusive and Vessey et al. (1992) suggest lack of structure and training may be to blame for the numerous CASE failures.

Most professionals agree that CASE has not delivered on its promises. Once predicted to be a $1 billion market by 1990, CASE sales were actually only $100 million, with Computer...
Aided Design (CAD) users the leading adopters. This is not surprising since, as Whitten et al. (1994) note, CASE bears a remarkable similarity to the engineering technology aspects CAD/CAM.

In sum, until American firms learn that productivity is the by-product of improved quality, CASE is bound to bring disappointing results in terms of measurements of success.

In this paper, an Information Abstraction Model (IAM) for end user CASE support is presented. Section 2 describes the use of the Capability Maturity Model (CMM) as a framework for establishing information requirements to support organizational maturity. This section also presents the IAM levels and boundaries for effective information management required to achieve a CMM level 3 maturity. Section 3 describes an integrated CASE toolset and user interface that would support the IAM. The paper concludes with a discussion of future research directions.

**Information Abstraction Framework**

Perhaps the real solution to this enigma is closer to Orlikowski’s (1993) argument for adoption of CASE tools conceptually as a form of organizational change in that such a perspective allows us to anticipate, explain, and evaluate different experiences. The implications of her framework further suggest the need to involve the end users (“key players”) by having them state their purpose for adopting CASE tools.

Paulk et al. (1993) observed that after two decades of broken promises regarding software quality and productivity, developers have shifted their attention away from methodologies and technology (including CASE tools) and toward management. They suggest an existing framework for managing the software process known as the Capability Maturity Model (CMM), which was developed in 1987 by the Software Engineering Institute and Mitre Corporation. These software process capability assessment methods are being used by a growing number of organizations. The principle behind the CMM is that the quality of a software product stems, in large part, from the quality of the process used to create it. To consistently improve software products, the process used for developing them should be understood, defined, measured, and improved.

Five basic levels of process maturity have been defined to describe this progression from an ad hoc software process to one that is under statistical control. Increases in maturity are associated with increasing productivity and decreased risk of failure. Paulk et al. (1993) explain that the five-level CMM partitions the software process into management, organizational and engineering subprocesses, each with associated Key Process Areas (KPA) as shown in Table 1.

We propose a framework called the Information Abstraction Model (IAM), shown in Figure 1, that will serve to introduce our organizational approach to using CASE to support this new perspective.

The concentric rings in the IAM represent the boundaries of activities defined within three organizational levels. The central ring depicts Software Engineering activities, perhaps the most formal and best understood in terms of process and practices. The Project Management activities are portrayed as surrounding the central activities to suggest two important concepts—the oversight role managers must play and the need for an interface across the boundary. The outer ring, representing the entire Organization, takes into consideration the maturity of the organization by assessing the successful development of all projects. These boundaries support the subprocesses as defined in the concept of organizational maturity in the CMM.

Early efforts utilizing the CMM focused only on software engineering. It is noteworthy that the higher maturity levels require an infrastructure and corporate culture that supports the concept—thus expanding the scope beyond the boundary of software engineering and into the project management sphere. Humphrey (1989) specifies project management as a key requirement for having a software development organization emerge from its initial (chaotic) stage and move toward maturity. Humphrey et al. (1991) describe the project management improvements that occur when a higher level of maturity is pursued by an organization.

We propose that CASE tools can best serve as this interface across the boundaries that exist between the activities associated with software engineering, project management, and the organization. The CMM provides recommended goals, measurements, and activities that permit an organization to mature, i.e., better define and consistently implement its software development process within and across the IAM layers.

It is important to note, as Paulk et al. (1993) did, that the CMM describes what to do, but not how. We submit that CASE tools provide the how. As an example, compare what Chikofsky and Rubenstein (1988) suggest that CASE offers with the CMM Level 3 definition that then follows:

A key feature of a well developed CASE environment is