Quantifying the Effects of Process Improvement on Effort

Bradford K. Clark, *Software Metrics, Inc.*

When organizations make many improvements concurrently, software project managers have no way of determining how much improvement is due to process maturity versus other factors. Using a 161-project sample, this article isolates the effects on effort of process maturity versus other effects, concluding that an increase of one process maturity level can reduce development effort by 4% to 11%.

According to the Software Capability Maturity Model (SW-CMM), the primary intended long-term benefit of high process maturity is high-quality software meeting customer requirements, delivered on time and within budget.[1] Productivity improvement is another important potential benefit. Because changing the software development process in an organization requires a large investment, many software organizations remain at low levels of maturity.

The effects of increasing process maturity alone are not easy to determine, because organizations generally make concurrent improvements in other areas that also result in benefits to the development organization. So, a clearer assessment of process maturity effects is needed. Anecdotal case study evidence shows many short- and long-term benefits to improving process maturity.[2–9] These studies used different assessment approaches, none of which try to isolate individual factors affecting productivity. Nevertheless, they do indicate that increasing maturity levels generally have positive effects.

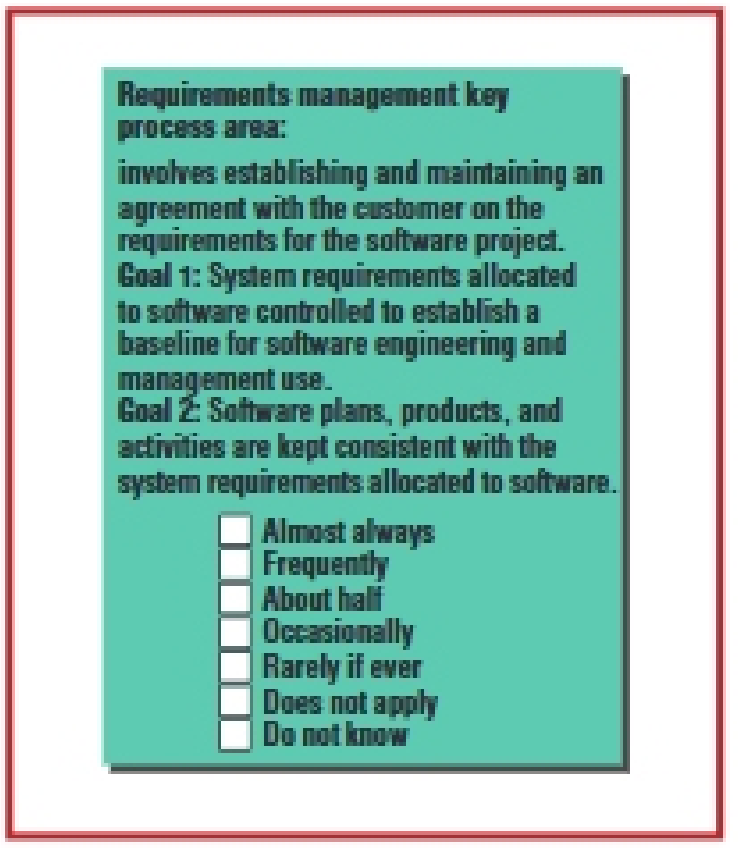
The premise of this article is that increasing process maturity decreases the effort required to develop a software product (effort is a fundamental component of productivity). The challenge is determining the effect of increased process maturity in the context of other influences on software development effort. I have used a mathematical model to segregate process maturity's influence on effort from that of other factors. The model quantifies the magnitude of this influence and the relationship between process maturity and the other factors. The results here are based on previous research that Barry Boehm and I conducted.[10,11]

Data Collection

Four areas generally influence software development effort: product factors, project factors, platform factors, and personnel fac-

0740-7459/00/$10.00 © 2000 IEEE

Figure 1. A Key Process Area data example.



tors. The data we collected on product factors included size, precedentedness, architecture and risk resolution, required reliability, database size, complexity, whether the product was developed for reuse, and documentation match to life-cycle needs. Data collected on platform factors included time constraints, storage constraints, and development platform volatility. Data collected on personnel factors included analyst and programmer capability, personnel continuity, team application experience, and language and tool experience. Data collected on project factors included development flexibility, team cohesion, tool usage, multisite development, schedule compression, and—the factor of interest here—process maturity. The Cocomo II model describes all the factors in detail.[11]

I collected data for the Process Maturity (PMAT) factor using two methods. The first selects an overall maturity level based on either an organized evaluation or a subjective judgment (see the middle column in Table 1). The SW-CMM Level 1 lower half is for organizations that rely on "heroes" to do the job. They don't focus on processes or documenting lessons learned. The SW-CMM Level 1 upper half is for organizations that have implemented most of the Key Process Areas that would satisfy SW-CMM Level 2. It is important to distinguish the groups working their way to a Level 2 rating. A transition from Level 1 lower to Level 1 upper is modeled as a change in a PMAT level. These two Level-1 ratings differ from the SW-CMM's published definition; the remaining levels follow the SW-CMM.

The second method selects a rating, called the Equivalent Process Maturity Level, based on the percentage of compliance for each Key Process Area goal.[1] If the project has undergone a recent CMM assessment, the EPML method uses the percentage compliance for the overall KPA (based on KPA compliance assessment data). If an assessment had not been done, I interviewed someone on the project or familiar with the project. My questions were directed at determining how close a project's processes met the goals for each KPA. I recorded the levels of compliance using the Likert scale. Figure 1 shows that scale along with one KPA.

Although an organization might be rated at a specific SW-CMM level, the respondents were encouraged to use the second method—that is, to answer all KPA questions considering what actually happened on the project.

Each of the 18 KPAs has seven rating levels:

- *Almost Always*: The goals are consistently achieved and are well established in standard operating procedures (over 90% of the time).
- *Frequently*: The goals are achieved relatively often but sometimes omitted under difficult circumstances (about 60% to 90% of the time).
- *About Half*: The goals are achieved about half the time (about 40% to 60% of the time).
- *Occasionally*: The goals are achieved less often (about 10% to 40% of the time).
- *Rarely If Ever*: The goals are rarely if ever achieved (less than 10% of the time).
- *Does Not Apply*: The respondent has the required knowledge about the proj-

Table 1

Process Maturity Rating Levels

Process maturity rating (PMAT)	Overall process maturity level	Equivalent process maturity level
Very low	CMM Level 1 lower half	0
Low	CMM Level 1 upper half	1
Nominal	CMM Level 2	2
High	CMM Level 3	3
Very high	CMM Level 4	4
Extra high	CMM Level 5	5

ect or organization and the KPA but feels that the KPA does not apply to his or her circumstances.
- *Do Not Know*: The respondent is uncertain about how to respond.

I computed an EPML as five times the average compliance level of all *n*-rated KPAs for a single project (Does Not Apply and Do Not Know are not counted, which sometimes makes *n* less than 18). The compliance level for each KPA is represented by a weight assigned to each level of the Likert scale (100 for Almost Always, 75 for Frequently, 50 for About Half, 25 for Occasionally, 1 for Rarely If Ever). I calculated the EPML as