Chapter 1: Basic Concepts and Structure of the Model

> *It is important to keep the distinction between reality and models clear.*
>
> Dr. Joseph F. Traub.[10]

Part 1: Basic Concepts

A "Capability Maturity Model" — what is it?

The term "capability maturity model" stands for a group of related notions. That set of notions is not common and has no standard short name for a bumper sticker and no simple icon for a tattoo whose meaning would be obvious.

Instead of a short common name, we use a long phrase — capability maturity model — and list below the concepts involved in how any company will go about systematically improving the way its work is organize. Such a company is necessarily one which recognizes that process improvement streamlines its business and technical practices to yield competitive advantage. Let's walk through the notions and see how they fit together.

"Capability"

The model document[11] uses "capability" in many places and phrases but does not give a definition. When the model does not define a term in the glossary or elsewhere, it uses the term in the common dictionary sense, and this is the case with "capability." I prefer to give my own reworking of the dictionary definition because the term's use has a history. (See below in this section: Why "capability": a Bit of History.)

For this *Guide*, "capability" means the sustained capacity to produce an agreed result. "Capacity" is the ability and readiness to do something. "Sustained" implies that the kind of capacity in the definition is not a permanent state but that systematic work and effort must be invested and managed lest the given capacity should erode. A "result" is a product or service, the output of the capacity, for a recipient. "Agreed" denotes that the capacity includes figuring out what the recipient is asking for and then obtaining the recipient's concurrence therein. "To produce" is to exercise the capacity which, since it is sustained, is repeatable and reliable. If you buy the product of such a producer, there is a high probability you will get the value you expect.

[10] Traub, 1997, p. C5.
[11] See the "CMMI Comes in a Choice of Formats" in the Introduction.

"Maturity"

"Maturity" here has to do with three other notions. One is "process" or, as I call it, how the work is organized. Another is the idea that process can "evolve" to become more effective, efficient, and streamlined, and that you can control that evolution. The third is that as you control the evolution, your company's process — its routine production steps — will pass through recognizable states. These states are called "levels," and they are roughly analogous to those of a person whose skill level, say, on the trumpet, goes from basic to virtuoso if he or she practices enough.

"Model"

A model is a representation that omits detail or features of something in order to highlight characteristics deemed important for a particular discussion. We usually think of a model as an image, such as an architect's 3-D rendering of a building, or a scale replica, like a ship model. The CMMI is like neither of those examples; it is more like the construction blueprints for a building or the descriptions in an anatomy textbook for organs and parts of a mammal — brain, foot, eye, liver, and so on.

Like the CMMI, the latter representations show separate parts of building or body, not the whole as an operational, real-world system. The CMMI isolates conceptually parts of the operational system.

What is modeled are stages of an engineering production system as it evolves in a whole company (staged representation), or as individual parts of that system evolve separately (continuous representation). The conceptually isolatable parts are called Process Areas, described below in the section of this chapter on structure.

But construction blueprints or illustrations in an anatomy textbook are quite a bit more concrete than the conceptual units in the CMMI. Or you might say, the "model" in CMMI is a whole lot more abstract. Probably the nearest analogy is to a topographical or nautical map, where a structural template is overlaid on some complicated and detailed piece of the real world. The map uses a fairly small set of conventional and generic symbols to model geography.

The model's resemblance to the day-to-day project in a company is about as close as a map of the London Underground, the New York subway, or the Paris Metro is to the stations, streets, and exits the traveler sees. The wide-eyed bewilderment of the stranger stepping out of a train at Victoria station, Times Square, or Les Halles looking for the correct exit or transfer point is evidence of this assertion.

But the analogy with maps breaks down quickly. For the model is not like a map prepared by someone for a journey. It is more like a kit for making a map of a production system.

An appraisal[12] team constructs the map by looking at characteristics of the production system — those characteristics described generically in the Process Areas — and places them on the grid of the model. The map is constructed by the model users — in this case, appraisal teams — by surveying the terrain. The resulting map is highly conventional; it always looks like one of the maturity levels of an organization (staged) or capability levels of a Process Area (continuous), not like what people do on projects. (Of course, models only represent, but do not duplicate, reality.)

"Process"

For a human enterprise like a "company" to produce an outcome, three basic elements are involved: people, process, and technology, or individuals performing tasks with tools. You might say that a fourth element, raw material, like crude oil or iron ore, is required for some product outcomes. I would agree but include those elements under the term "technology," since even raw materials must be "processed" for use. There are whole industries and international enterprises involved in extracting crude oil from under ground and delivering it to refineries where it is consumed as a raw material. In other words, even raw material goes through a people-process-technology triangle.

Capability maturity models focus on the "process" part of the triangle. As we learned in school geometry, the points and sides and angles of a triangle are interrelated — you can't change a single aspect of one without changing all the others in predictable ways. Or in the world of cmms, as the process evolves predictably, the people and the technology will change in more or less predictable ways.

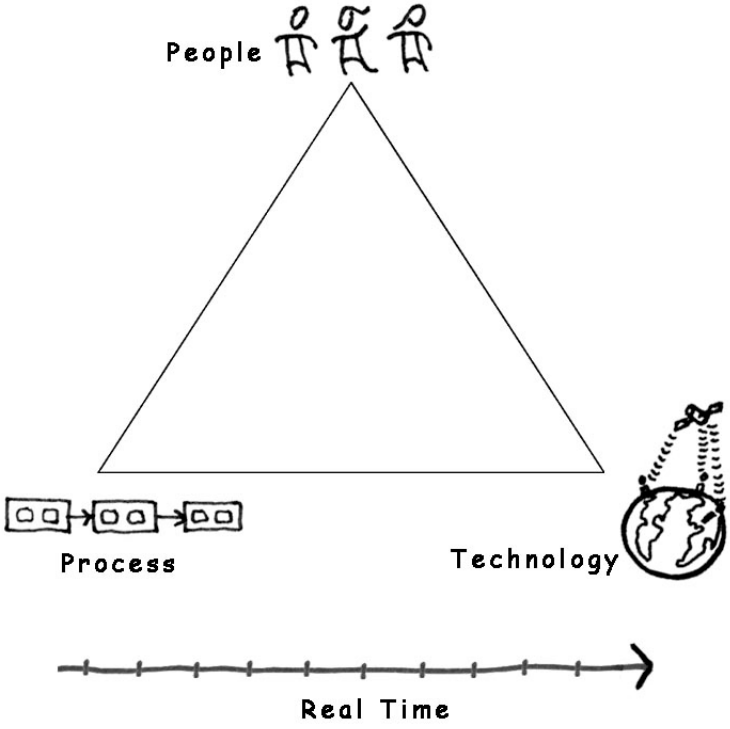
Cmms focus on process because therein lies the greatest opportunity for the kind of evolution leading to more streamlined operation. The effect doesn't necessarily go the other way, from technology or people to process, or at least not so predictably.

Here are the arguments for the focus on process. First for people. Human traits like ingenuity, common sense, and whatever else constitutes the ability to do things, are distributed about the same way in any human population. There may be differences in skills at certain tasks but these differences come largely from the environment. A teenage native of Paris is adept at traveling by Metro. A Xhosa-speaking youth of the Kalahari may grow up knowing how to track a leopard in the countryside. Each could learn the other's skills given time and the right environment — or, you might say, people-process-technology.

For a "company" concerned about capability, part of its process is conveying the skills needed by the workforce. Sometimes you hear the counterargument that "we don't need process because we only hire the best and the brightest." This strategy can only work for a

[12] The SCAMPI[sm] appraisal method for determining degree of implementation of CMMI is beyond the scope of the *Guide*. For a description see SEI 2006b in the Bibliography.

short while, if at all, because it quickly runs up against all kinds of limits. For one, there is the standard distribution of skilled people in any population — there just aren't any more. For another, your competitors are seeking them too and may hire them away.[13] For another, the best and the brightest employees with an entrepreneurial spirit may learn your business, especially your process weaknesses, and start their own companies, becoming your more efficient competitors.



People-Process-
Technology *plus* Time

Figure 1

[13] An example of the hiring frenzy for a very small pool of what I will call, for the sake of argument, "skills," is going on nearby (the Washington, D.C. area) as I write (January 2004, originally, and still true October 2006). Many high tech companies with government contracts are advertising in all media (newspapers, tv, radio, the web) for engineers and programmers who have security clearances to work on the classified projects for countering terrorism. It takes six or more months to get a security clearance, so the ads are aimed at a small population that is continually dwindling as individuals are hired.

Ultimately, process makes the most difference: "The performance of willing workers…was governed totally by the process they worked in." (Deming 1993, p. 173)

Next, why technology has less leverage than process for systematic improvement. Technology advances at its own pace and is not predictable in its effects. For example, what we now call the internet began with the idea of a U.S. military communications network that would still function if part of the country's infrastructure were destroyed by a nuclear attack.[14] A text message entered into the telecommunications system would be segmented into many packets with address headers. The packets would be routed by computers at nodes that would keep checking for next nearest nodes still "alive" and send the packet on quickly, like a "hot potato." The idea goes back at least to the 1960s. The network was operating routinely by the early 1980s in U.S. government circles and a few universities, but the idea of email didn't surface in the popular mind until the early 1990s. The World Wide Web wasn't noticed by the press and most of the world until 1994. And now loads of spam are delivered daily to our email in-boxes.

Besides being unpredictable in its effects, the march of technology tends to raise almost all competitors and at nearly the same rate. Early adopters may benefit first, but their advantage, quickly overtaken by competitors, does not count as "sustained."

The people-process-technology triangle implies the idea that changing any one of the elements involves the other two. It reminds us that introducing new technology affects production steps and that even the best and the brightest — assuming you can find and hire them systematically better than your competitors — have to know what to do next in a production process because others are depending on them.

In a sense, you could say that companies following cmms are becoming better and better at working the people-process-technology triangle.

"Time"

Finally, time governs all. It is under the control of no one and can't be increased or decreased one whit. A company's constant concern should be to optimize its people-process-technology configuration with respect to time, both its cycle time for deploying improvements internally and its capacity to deliver products within schedule constraints.

Nor can you optimize productivity by adding more people to the work in the same time period. Fred Brooks (Brooks 1982 and 1995) makes this point eloquently in his books entitled *The Mythical Man Month*. (See Appendix C, The Laws of Project Thermodynamics, for my attempt at the topic.)

[14] Hafner 1998, Chap. 2.

Optimizing is not working faster, like the Charlie Chaplin figure in the film *Modern Times*. It is a matter of streamlining by removing causes of rework and making one step serve multiple outcomes.

Maturity Levels

Maturity Levels concern the staged representation[15]; their number is five. They are the evolutionary phases that will be traversed by a company systematically improving the way it works — its people-process-technology system through time.

The five maturity levels are designated by number — "1," the least evolved, through "5," the highest — and by a tag phrase:

Level	Tag Phrase
5	Optimizing[16]
4	Quantitatively Managed
3	Defined
2	Managed
1	Initial

My own ideas about what these phrases mean are given in the introductions to Chapters 2 through 5 for the respective levels. (The Initial Level, defined by reference to Level 2, is also described in the Chapter 2 introduction.)

The essence of each level, except for the lowest, is the smooth, sustained, and interconnected operation of the Process Areas involved. My own tag phrases for the essences of each state are Level 1-ness, Level 2-ness, and so on. Level 1-ness is the absence of all or some (Level 2) Process Areas.

Evolution

> *The only things that evolve by themselves in an organization are disorder, friction, and malperformance.*
>
> Peter F. Drucker[17]

Evolution is a foundation concept of cmms. The levels through which capability evolves are meant to be a strategic path traversed by product flows and the organization that contains them, not merely measurement benchmarks. The idea is that capability, sustained capacity to deliver, must be continually improving on the evolutionary scale.

[15] See Part 2 of this Chapter.
[16] Note that the tag phrase for Level 5 is a gerund where the other levels are past participles, indicating something completed, or an adjective ("Initial") specifying sequence. The gerund is a grammatical way (at least in English) to show that Level 5 is a continuous state, not one completed.
[17] Quoted in Feder 2005.

The last statement is a change in my thinking over the last decade and also represents, I believe, a change in the world community interested in cmm-based improvement. In the 1995 *A Guide to the CMM*®, the "four simple concepts" on which the then CMM was based are

1. Evolution is possible and takes time.
2. There are distinguishable stages of process maturity.
3. Evolution implies that some things must be done before others.
4. Maturity will erode unless sustained.

Now in 2006 I would say the first three simple concepts have proven true but the fourth as stated is too weak: it should read

4. Maturity will erode unless sustained by continual improvement.

The world community applying process improvement has also shown continual evolution. Witness the difference in the number of higher maturity organizations between 1995 and 2006. In the mid-90s there were no companies at Level 4 and only one at Level 5 anywhere. Eleven years later, there are more higher maturity organizations outside of the U.S. where the cmm movement began. The majority of all organizations following a cmm path are now commercial firms and not Department of Defense (DoD) suppliers, who were the original, expected theater of process improvement. In my own local experience in the Washington, D.C. area where many of our clients are commercial companies whose customer is the civilian U.S. government, it is common to see government contracts requiring specific project life cycles. That practice was rare a decade ago when such contracts often did not "speak" process.

All this means that the world community applying cmm concepts has raised the bar for everyone and will probably keep raising it. That is the nature of competition fueled by technology innovation, where "technology" in this case includes process.

For one thing the models themselves are evolving and encompass more types of product flow, no longer just software as in the CMM of the 1990s, but also systems as in CMMI currently and soon outsourcing and services.[18] For another, the community is coming to understand the higher reaches of process maturity. There is a growing body of knowledge on just what higher maturity looks like[19] as well as a proposal of how to implement at least some of the practices or subprocesses.[20]

[18] See Appendix E on the CMMI framework.
[19] See any relevant case study in the Proceedings of any recent SEPG conference in the U.S., Europe, and now Latin America, or Australia.
[20] See Humphrey 2006.

Why "Capability": a Bit of History

The idea of capability and a means to measure it — appraisals, formerly called "assessments" — arose together before cmms were even thought of. Both the CMM, predecessor model to the CMMI, and assessment were part of a solution to a problem faced at the time by the U.S. Department of Defense, the huge federal government establishment that maintains the country's military services. Most of the weapons, communication, intelligence, and Information Technology (IT) systems the DoD uses are obtained from commercial companies. The DoD's problem was to sort out from among all the offerers the most capable suppliers of highly expensive, long term and complicated programs, the kind likely to receive legislative scrutiny.

When such a program turned out to be a flop after years of work, followed by well-publicized congressional investigations and hundreds of millions of dollars spent without value received, post mortems often pointed to the software as the culprit. There seemed to be a general inability among the supplier engineering organizations, including the military themselves, to estimate the complexity and cost of software projects and to manage those efforts at least as well as hardware projects, which admittedly in many cases did not always meet the highest standard.

The DoD recognized that the inability was not traceable to anyone's fault nor to causes that anyone knew how to remedy. The usual screening criteria and contract regulations to filter out those suppliers least likely to succeed did not work well enough in the case of software production.

Software is invisible unless documented in some way humans can perceive, such as when source code is printed. Even then there are no standard techniques, easily learned by the non-initiated or agreed on by the experts, to measure the size or complexity or conformance to requirements of software. Inspecting an offerer's software production facility was no help. Every software company in the world looks about the same: you see cubicles — sometimes offices — with persons sitting at keyboards, sometimes typing and with long periods of staring at computer monitors. In fact, the physical environments all look like the place where DILBERT™ works.[21]

When facing this kind of technical problem — how to predict that an offerer will deliver something no one has built before — with no apparent solution, the DoD has a strategy. It establishes a center to discover and try various approaches. Such an organization is called a Federally Funded Research and Development Center or FFRDC. Several such centers arose during the cold war to advance critical defense technology — radar, sonar detection, nuclear research. They are usually run by a university under contract to one of the military

[21] For examples, see just about any DILBERT™ comic strip by Scott Adams (Adams 2006). DILBERT™ works in a company with an inept boss and a preposterously impaired product flow where confusion is the main output and buzzwords reign.

services. Having a university administer the work at one remove from the military avoids the focus on the short term that comes from micro-management by the customer.

Competing against two other universities, Carnegie Mellon University in Pittsburgh, Pennsylvania was awarded the contract in 1984 for the FFRDC now known as the Software Engineering Institute, the SEI.[22] One of its major goals was to "advance the state of the practice of software engineering."

By 1986 one of the SEI approaches to the goal was through process. Others were in education and various software engineering methods — recall the people-process-technology triangle.

The SEI Process Program, then headed by Watts Humphrey, looked at the problem in the large: not at software code but at the companies that produce it. One question the Process Program asked was, How can you make visible the differences among contractors who all say they can do the job you want at the lowest cost?

Their answer was to devise and test a systematic way of gauging the production capability of a software company. Watts Humphrey, Bill Sweet and others with years in the software industry, and who had been on both sides of the customer-supplier frontier, laid out a program.[23] They posited that

"The capability of a contractor to perform software engineering […is] divided into three areas:

1. organization and resource management
2. software engineering process and its management
3. tools and technology"

They defined a method of assessing contractors on these three people-process-technology areas "…to augment the current contractor evaluation methods" used by the DoD. One of their unique ideas was to make this assessment method different from the usual evaluation techniques. The assessment method as well as the criteria for judging capability were to be made public so that contractors could prepare for government evaluations. Contractors could conduct their own internal assessments to identify and remedy weaknesses before such an evaluation. The assessment method's originators also pointed out that customer evaluators should "…consider both current capability and *future plans for software process improvement*." (Italics added. Notice the incipient idea of an industry bootstrapping itself into process maturity. I include "sustained capacity" in my definition of "capability" above because current capability is not guaranteed without continual improvement.)

[22] Barbacci 1985.
[23] Humphrey 1987.

The whole approach as well as the details of the then assessment method are in the 40 pages of the paper by Humphrey and others.[24]

That paper also included, on less than a page, a sketch of five levels of process maturity and, in a few pages of its Addendum B, a list of 85 questions whose answers would be indicators of capacity to deliver. That sketch eventually grew into the *Capability Maturity Model for Software, Version 1.1* in 1991 and into the CMMI in 1998. The list of questions evolved into the assessment method.

Assessments and maturity models go together

This bit of history shows that in the beginning assessment and maturity levels were related as measurement method to measurement scale. What they measured was capability of the software process at an assessed company. The measurement outcome was one of the 5 maturity levels.

Because the maturity level was to be one factor the DoD would consider when selecting contractors, the software process assessment quickly became a concern of companies in the U.S. aerospace sector. The SEI Process Program at the time was training assessment teams in those companies to produce an internal reading of their maturity level as well as teams of government auditors who would perform external evaluations of those companies. As more DoD requests for bids were issued specifying a minimum capability maturity level — always greater than Level 1 — more assessments were done. As more companies were assessed, the assessment method became more familiar. But companies still wanted more details on the maturity levels themselves. The 40 pages and 85 questions in the Humphrey 1987 paper gave an outline but no more.

If you were familiar enough with principles of sound project management and software engineering, it was easy enough to describe the concepts behind the 5 levels and to infer the good practices expected from the 85 questions. If you had experience of the kinds of things that go wrong on technical projects, you could see how the practices would prevent problems and you could therefore make a good case at least for Level 2.

But many companies who were to have their bid proposals judged partly on audits of their process maturity wanted more than a good case. Lack of detail in the model might increase the chance of disputes over whether given practices were in place.

Watts Humphrey's *Managing the Software Pro*cess (Humphrey 1989) addressed the need for more explanation of software engineering process and assessment and of good technical and project management practices. That book is still the best single reference for the concepts behind capability maturity models and for advice on implementing their practices.

[24] The above quotes are from pp. 1 and 3.

But a book by a single author could not be a standard for assessments and audits. Though the maturity model was widely accepted, its details had to be codified for the industry that would use it.

At the SEI Humphrey had started a project with a staff of 4-5 to devise a maturity level framework as a structure for describing good practices as well as a method of working with industry and government stakeholders to identify those practices.

The maturity level framework was the difficult part. How do you organize the many practices involved? Just a list of the practices would not do, as the questions in the 1987 paper showed. Besides describing good practices for a maturity level, the framework also must explain what constitutes being at a maturity level, i.e., keeping those good practices in place. Capability maturity models call this attribute "institutionalization." (This is the idea I try to convey by the term "sustained" in the definition of capability above.)

After considering various architectures, the SEI team codifying the software CMM chose a structure they called a Key Process Area.[25] The same structure has evolved, with a few changes, into the Process Area of the CMMI.

[25] Devising the model architecture was not easy. My colleague, Cynthia Wise, who was part of the team, said they had plenty of debates and considered many alternatives.